

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Приладобудівний факультет

Кафедра приладів і систем орієнтації і навігації

«На правах рукопису»
УДК 519.651: 519.654

«До захисту допущено»

Завідувач кафедри

_____ Н. І., Бурау

«___» _____ 20__ р.

Магістерська дисертація

на здобуття ступеня магістра

**зі спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»
(«Комп'ютерно-інтегровані технології оптичних та навігаційних систем»)**

**на тему: «Автоматизація процесу обробки сигналів з коливальними
складовими»**

Виконав:

студент II курсу, групи ПГ- 11мп
Шелемаха В.В.

Керівник:

к.т.н., доцент,
Цибульник С.О.

Консультант з Розроблення стартап-проекту:
завідувач, д.е.н., проф. Бояринова К.О.

Рецензент:

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших
авторів без відповідних посилань.

Студен _____

Київ – 2022 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Приладобудівний факультет

Кафедра комп'ютерно-інтегрованих оптичних та навігаційних систем

Рівень вищої освіти – другий (магістерський)

Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Комп'ютерно-інтегровані системи та технології в приладобудуванні»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Н.І. Бурау

«___» _____ 2022 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Шелемасі Владиславу Валентиновичу

1. Тема дисертації «Автоматизація процесу обробки сигналів з коливальними складовими», науковий керівник дисертації Цибульник Сергій Олексійович, к.т.н., доцент, затверджені наказом по університету від «___» _____ 20__ р. № _____

2. Термін подання студентом дисертації: 06 грудня 2022 року

3. Об'єкт дослідження: процес обробки коливальних сигналів автоматизованими системами.

4. Предмет дослідження: автоматизація процесу оброблення для проведення швидкого первинного оцінювання спектральних характеристик коливальних сигналів; генерування та обробка тестових коливальних сигналів для навчання операторів та формування у них навичок аналізу даних.

5. Перелік завдань, які потрібно розробити:

5.1 Провести огляд стану проблеми.

5.2 Провести огляд існуючих рішень у сфері автоматизованих систем обробки коливальних сигналів.

5.3 Провести огляд існуючих засобів розроблення автоматизованої системи обробки коливальних сигналів.

5.4 Розробити основні модулі та підсистеми.

5.5 Перевірка працездатності системи.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: таблиці, графіки, рисунки, схеми, діаграми тощо

7. Орієнтовний перелік публікацій: 1 стаття в матеріалах конференції

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розроблення стартап-проекту			

9. Дата видачі завдання 15 вересня 2022 року

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Провести огляд стану проблеми	05.10.2022	
2.	Провести огляд існуючих рішень у сфері автоматизованих систем обробки коливальних сигналів	19.10.2022	
3.	Провести огляд існуючих засобів розроблення автоматизованої системи обробки коливальних сигналів	26.10.2022	
4.	Розробити основні модулі та підсистеми	23.11.2022	
5.	Перевірка працездатності системи	30.11.2022	
6.	Оформлення рукопису дисертації	06.12.2022	

Студент

Владислав ШЕЛЕМАХА

Науковий керівник дисертації

Сергій ЦИБУЛЬНИК

РЕФЕРАТ

Питання по первинній обробці сигналу досить часто постає перед дослідниками і інженерами, які вже мають великий досвід по роботі з різними типами сигналів та серед людей, які тільки починають вивчати тему первинного оцінювання спектральних характеристик коливальних сигналів. Область первинної обробки широко використовується у різних областях науки, як у виконанні проектування обладнання, математичних розрахунках, дослідженнях струму чи систем багатокласової діагностики.

У зв'язку з цим виникає потреба у автоматичних системах по обробленню сигналів для оцінки спектральних характеристик коливальних сигналів, які будуть виконувати обробку та графічно показувати результати у вигляді: таблиць, графіків, рисунків, схем, діаграм, з подальшим експортуванням результатів.

Основні потреби, які виникають при обробці сигналу:

- побудова часового графіку сигналу;
- математичний розрахунок величин, а саме: мінімальне значення(по всім вісям), максимальне значення(по всім вісям), медіана, середнє значення, розмах, дисперсія, середньоквадратичне відхилення, математичне сподівання період та частота дискретизації та кількість відліків у сигналі;
- побудова частотних графіків сигналу за різними методами, а саме: швидке перетворення Фур'є, використання різних типів віконних функцій;
- побудова часових та частотних графіків різних типів періодограм;

Отже задача автоматизації процесу оброблення для проведення швидкого первинного оцінювання спектральних характеристик коливальних сигналів наразі стоїть гостро у наші часи.

Метою дисертаційної роботи є розробка автоматизованої системи оброблення для проведення швидкого первинного оцінювання, а також рекомендації щодо його ефективного використання.

Досягнення мети передбачає вирішення наступних задач:

- огляд стану проблеми;
- огляд існуючих рішень первинного оброблення сфері автоматизованих систем обробки коливальних сигналів ;
- огляд існуючих засобів розроблення автоматизованої системи обробки коливальних сигналів;
- розробка основних модулів та підсистем;
- перевірка працездатності системи;

Об'єктом дослідження є процес обробки коливальних сигналів автоматизованими системами.

Предметом дослідження є автоматизація процесу оброблення для проведення швидкого первинного оцінювання спектральних характеристик коливальних сигналів; генерування та обробка тестових коливальних сигналів для навчання операторів та формування у них навичок аналізу даних.

Методи дослідження – для розробки автоматизованої системи використовувались мова програмування Python для обробки даних та мова програмування JavaScript для відображення результатів у вигляді графічного матеріалу.

Наукова новизна дисертації полягає в наступному:

- зменшено час обробки, необхідний для первинного аналізу даних, які містять коливальні процеси, у порівнянні з математичним пакетом MATLAB;
- автоматизовано процес обробки згенерованих коливальних сигналів для підвищення ефективності навчання користувачів основам аналізу даних.

Практичне значення отриманих результатів визначається розробкою алгоритмічного та програмного забезпечення на основі розроблених методик, рекомендацій та математичних моделей, які можуть застосовуватись у навчанні операторів або первинному аналізі різних типів сигналів. Можливе впровадження результатів у навчальний процес або наукову тему.

Апробація результатів дисертації відбулася на наступних конференціях:

- погляд у майбутнє приладобудування, КПІ ім. Ігоря Сікорського, приладобудівний факультет, Київ, 2022;
- ефективність та автоматизація інженерних рішень у приладобудуванні, КПІ ім. Ігоря Сікорського, приладобудівний факультет, Київ, 2022;

Публікації. За матеріалами дисертації було опубліковано 2 тези доповідей на науково-практичній конференції.

Шелемаха В.В., студент гр. ПГ-11мп, к.т.н., доц. Цибульник С.О. КПІ ім. Ігоря Сікорського. Огляд засобів обробки інформації. Погляд у майбутнє приладобудування(22 червня 2022), КПІ ім. Ігоря Сікорського, приладобудівний факультет, Київ, 2022.

Шелемаха В.В., студент гр. ПГ-11мп, к.т.н., доц. Цибульник С.О. КПІ ім. Ігоря Сікорського. Автоматизація процесу обробки сигналів з коливальними складовими. Ефективність та автоматизація інженерних рішень у приладобудуванні(6 грудня 2022), КПІ ім. Ігоря Сікорського, приладобудівний факультет, Київ, 2022.

Структура дисертації. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, переліку посилань. Загальний обсяг дисертації становить 102 сторінок, 27 малюнків, 35 таблиць, 54 положення переліку посилань.

АВТОМАТИЗОВАНА СИСТЕМА ШВИДКОГО ПЕРВИННОГО
ОЦІНЮВАННЯ, ПЕРІОДОГРАМА, ПЕРЕТВОРЕННЯ ФУР'Є,
МАТЕМАТИЧНИЙ АНАЛІЗ СИГНАЛУ

ЗМІСТ

ВСТУП.....	7
1 ОГЛЯД СТАНУ ПРОБЛЕМИ.....	9
1.1 Програмне забезпечення Microsoft Excel	9
1.2 Програмне забезпечення STATTA	10
1.3 Програмне забезпечення SPSS	12
1.4 Мова програмування R.....	15
1.5 Мова програмування SAS	16
1.6 Програмне забезпечення Matlab	18
1.7 Програмне забезпечення Minitab	20
1.8 Програмне забезпечення Scilab	22
1.9 Мова програмування Python	24
1.10 Програмне забезпечення GraphPad Prism	26
1.11 Мова програмування Java.....	27
2 Обґрунтування обраних технологій для автоматизованої обробки сигналів.....	29
2.1 Що таке Python	29
2.2 Переваги та недоліки мови програмування Python	30
2.3 Порівняння мови програмування Python з іншими мовами програмування	33
2.4 Що таке Django web framework	35
2.4.1 Як виглядає код Django.....	38
2.5 Переваги та недоліки Django	39
2.6 Порівняння Django з іншими фреймворками.....	43
2.7 Що таке Pandas.....	47
2.7.1 DataFrame та Series.....	49
2.8 Переваги та недоліки Pandas.....	51
2.9 Порівняння Pandas з бібліотекою NumPy	53

2.10	Опис технологій, які були обрані для розробки веб додатку.....	56
	Висновки до розділу 2.....	57
3	Автоматизована система по обробленю сигналу з подальшим спектральним аналізом	58
3.1	Код для налаштування роботи серверної частини проекту.....	60
3.2	Аналітичний код по обробці даних.....	63
3.3	Код для графічного відображення даних.....	64
3.4	Порівняння з іншими система по обробці сигналів	67
3.4.1	Зручність.....	67
3.4.2	Стабільність.....	68
3.4.3	Надійність.....	69
3.4.4	Швидкодія.....	69
3.5	Приклад роботи з веб додатком.....	71
	Висновки до розділу 3.....	79
4	РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ.....	81
4.1	Опис ідеї проекту.....	81
4.2	Технологічний аудит ідеї проекту.....	84
4.3	Аналіз ринкових можливостей запуску стартап проекту.....	85
4.4	Розроблення ринкової стратегії проекту.....	93
4.5	Розроблення маркетингової програми стартап-проекту.....	96
4.6	Розроблення бізнес-моделі стартап-проекту.....	100
	Висновки до розділу 4.....	101
	ВИСНОВКИ.....	103
	ПЕРЕЛІК ПОСИЛАНЬ.....	105
	Додаток А	110
	Додаток Б.....	168

ВСТУП

Наразі важко уявити проект або бізнес без великих об'ємів даних для подальшої обробки та аналізу даних. Але велика кількість даних та їх обрахування займає все більше часу для збору та аналізу даних. При цьому вибір оптимальних та швидкісних методів по обробці сигналів стає дедалі складним та затратним за ресурсами та грошам. Особливо ці труднощі характерні для BigData проектів та їх похідних.

Тому на перший план виходять системи, які дозволяють швидко проводити первинне оброблення сирих даних сигналів для подальшого оцінювання та подальшої потреби для використання у проекті.

Отже, метою побудови автоматизованого процесу оброблення для проведення швидкого первинного аналізу сигналів у більшості випадків є розуміння чи потрібні ці дані для використання у подальшому у більш складних системах та більш швидке навчання молодого персоналу для роботи та обробки даних. Проте в більшості технічних систем задача пришвидшене навчання операторів є невирішеною, оскільки існуючі рішення потребують великий обсяг математичних та технічних навичок.

Тому розробка нових автоматизованих систем по обробленню коливальних сигналів є надзвичайно важливою і актуальною задачею при навчанні операторів та оцінці спектральних характеристик сигналу.

Метою дисертаційної роботи є розробка автоматизованої системи оброблення для проведення швидкого первинного оцінювання, а також рекомендації щодо його ефективного використання.

Досягнення мети передбачає вирішення наступних задач:

- огляд стану проблеми;
- огляд існуючих рішень первинного оброблення у сфері автоматизованих систем обробки коливальних сигналів ;
- огляд існуючих засобів розроблення автоматизованої системи обробки коливальних сигналів;

- розробка основних модулів та підсистем;
- перевірка працездатності системи;

Об'єктом дослідження є процес обробки коливальних сигналів автоматизованими системами.

Предметом дослідження є автоматизація процесу оброблення для проведення швидкого первинного оцінювання спектральних характеристик коливальних сигналів; генерування та обробка тестових коливальних сигналів для навчання операторів та формування у них навичок аналізу даних.

Методи дослідження – для розробки автоматизованої системи використовувались мова програмування Python для обробки даних та мова програмування JavaScript для відображення результатів у вигляді графічного матеріалу.

Наукова новизна дисертації полягає в наступному:

- зменшено час обробки, необхідний для первинного аналізу даних, які містять коливальні процеси, у порівнянні з математичним пакетом MATLAB;
- автоматизовано процес обробки згенерованих коливальних сигналів для підвищення ефективності навчання користувачів основам аналізу даних.

Практичне значення отриманих результатів визначається розробкою алгоритмічного та програмного забезпечення на основі розроблених методик, рекомендацій та математичних моделей, які можуть застосовуватись у навчанні операторів або первинному аналізі різних типів сигналів. Можливе впровадження результатів у навчальний процес або наукову тему.

Апробація результатів дисертації відбулася на наступних конференціях:

- погляд у майбутнє приладобудування, КПІ ім. Ігоря Сікорського, приладобудівний факультет, Київ, 2022;

Публікації. За матеріалами дисертації було опубліковано 1 тези доповідей на науково-практичній конференції.

РОЗДІЛ 1

ОГЛЯД СТАНУ ПРОБЛЕМИ

В даний час ми живемо в «епоху даних», коли використання програмного забезпечення для статистичного аналізу або аналізу даних неминуче в будь-якій галузі досліджень.

Це значить, що вибір правильного програмного інструменту чи платформи є стратегічним питанням для дослідницького відділу. Тим не менш, у багатьох випадках користувачі, які приймають рішення, не приділяють належної уваги всебічний та належний оцінці того, що пропонує ринок. Справді, вибір, як і раніше, залежить від кількох факторів, таких як, наприклад, особисті переваги дослідника, тобто яке програмне забезпечення використовувалося в університеті або вже відомо самому користувачеві. В принципі, це не так, але в деяких випадках цього цілком недостатньо і може призвести до «тупикової» ситуації, як правило, після кількох місяців або років роботи, зробленої в неправильному програмному забезпеченні. У будь-якому випадку, здається, що немає загальноприйнятої та відомої точки зору на процес оцінки програмного забезпечення з точки зору кінцевого дослідника. Але можна проаналізувати ринок застосунків для статистики та, спираючись на пропозиції виробника, відокремити деякі з них.

1.1. Програмне забезпечення Microsoft Excel

Одна з таких програм Microsoft Excel — це застосунок для роботи з електронними таблицями, призначений для виконання основних та складних математичних обчислень та функцій. Крім виконання арифметичних операцій, Excel дозволяє користувачам створювати графічні відображення даних, форми та зведені таблиці. Програму також можна використовувати для сортування інформації та допомоги користувачам у визначенні тенденцій даних[1]. Дане програмне забезпечення дає користувачам можливість програмувати

підпрограми, які називаються «макросами». Ці програми використовують код Visual Basic для запису вибраних дій користувача з Excel. Потім ці дії можуть бути ініційовані для автоматичного завершення Excel. Досвідчені користувачі навіть можуть запрограмувати Excel для створення інтерактивних електронних таблиць, які реагують на різні зовнішні тригери[2]. Але ця програма має ряд мінусів та плюсів[3].

Цей застосунок дуже зручний для аналізу малих та середніх даних. Має швидке виведення графіків. Також є плюс в тому, що можна просто імпортувати та експортувати дані в інші бази даних (access, sql), і так само з них.

Є сумісність з усіма комп'ютерами. Ще макроси VBA легко програмувати, але занадто багато формул для вибору (також вказано як мінус).

Мінуси Excel є наступні: під час дослідження було видно, що програма зависатиме, якщо набір даних занадто великий, для імпорту всіх даних вихідний файл має бути відкритим та занадто багато формул та синтаксису можуть заплутати.

Використання програмного забезпечення Microsoft Excel може покращити розуміння змісту у межах графічного представлення інформації; додаток забезпечує візуальне подання даних, що спрощує їх аналіз. Excel знижує складність побудови даних та надає користувачам засіб для інтерпретації даних. Також можна звернути назад традиційний процес аналізу даних, надавши учням заповнену діаграму та подивившись, чи зможуть вони відновити базовий робочий лист. Це має велике значення для того, щоб допомогти їм зрозуміти взаємозв'язок між даними та діаграмою.

1.2. Програмне забезпечення STATA

Але Excel не має комплексного набору функцій для більш глибокого аналізу. Користувачі, яким потрібен саме повний, інтегрований статистичний пакет програмного забезпечення, який надає все необхідне для обробки даних, обирають застосунок STATA. Програмне забезпечення Stata – це комплексне

програмне забезпечення для аналізу даних із високотехнологічним набором інструментів, що забезпечує цілісний та сучасний аналіз даних. Всі інструменти програми можуть бути повністю налаштовані та автоматизовані. Застосунок ідеально підходить для підприємств та установ, які зосереджені на даних та статистиці[3]. Він має спеціалізовані інструменти для наступних областей: поведінкові науки, біостатистика, наука про дані, економіка, освіта, епідеміологія, фінанси, маркетинг, дослідження, медицина, політологія, громадська охорона здоров'я, державна політика та соціологія. Stata пропонує надійні та безпечні інструменти для дослідження, обробки та візуалізації даних. Вивчаючи цю програму, було видно, що є майже всі відомі статистичні моделі, функції, методи та аналітику, які дозволяють користувачам повністю використовувати глибину, межі та потужність зібраних даних[4]. Виходячи з особистого досвіду та оцінки програми користувачами, можна виділити кілька хороших і поганих сторін[5].

Серед плюсів даного продукту можна відмітити, що програма має широкий набір функцій. Можна знайти майже всі відомі статистичні методи. Ще програма надає легкий доступ через графічний інтерфейс. Також є можливість застосунку бути автоматизованим та він має сумісність із старими версіями. Ця програма доступна для Windows, MacOS та Linux

Має гарну підтримку з боку спільноти STATA та доступний великий обсяг літератури.

Але, наприклад, з точки зору включення нових методів (оновлень версії) програма стоїть на місці, також немає можливості запрограмувати нові функції в Stata та інтеграція з іншим програмним забезпеченням обтяжлива Комерційна однокористувальна ліцензія (IC) коштує приблизно 730 € і вище. При купівлі кількох ліцензій надаються значні знижки, у сфері освіти діють особливі умови.

У цій програмі [6] доступна корисна функція «Вкажи та клацни»: при її використанні можна отримати доступ до всіх функцій Stata з обробки даних, статистики та візуалізації з меню та пов'язаних діалогових вікон.

Синтаксис Stata інтуїтивно зрозумілий та простий у вивченні та викладанні. Ось приклад простоти коду:

regress y x1 x2

logistic y x1 x2

Безкоштовна технічна підтримка: якщо у користувача виникли питання про використання програми, спеціальний штат Stata швидко відповість. Документація: документація Stata несуперечлива, зв'язкова та проста для розуміння. Stata постачається з 33 томами з більш ніж 17 000 сторінок документації у форматі PDF, що містить розрахункові формули, повністю опрацьовані приклади, посилання на літературу та детальні обговорення.

1.3. SPSS

Але якщо проаналізувати статистику по кількості наукових статей, знайдених у Google Scholar, щодо програмного забезпечення для обробки даних, то можна побачити, що статистичний пакет SPSS займає перше місце.

На рисунку 1 показано кількість статей, знайдених для більш популярних пакетів програмного забезпечення та мов (тих, що мають принаймні 1700 статей) за 2018 рік[7] .

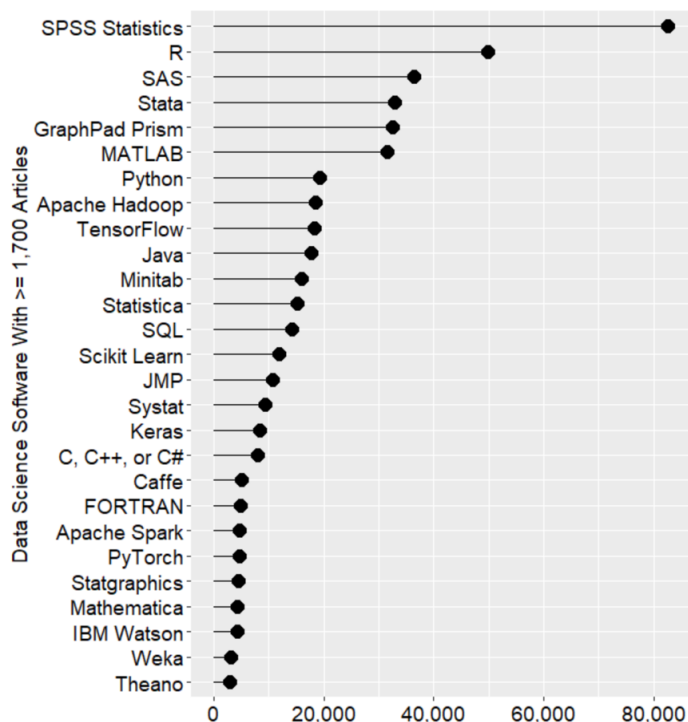


Рисунок 1.1 - Кількість наукових статей за 2018 рік

SPSS є, безумовно, найдомінантнішим пакетом, як це було протягом понад 20 років. Це може бути пов'язано з його балансом між потужністю та простим використанням.

SPSS (Statistical package for the social sciences) це набір програм, об'єднаних в один пакет. Основним застосуванням цієї програми є аналіз наукових даних, пов'язаних із соціальними науками. Ці дані можна використовувати у маркетингових дослідженнях, опитувань, інтелектуального аналізу даних, тощо[8].

За допомогою отриманої статистичної інформації дослідники можуть легко зрозуміти попит на продукт на ринку та відповідним чином змінити свою стратегію. По суті, SPSS спочатку зберігає та систематизує надані дані, а потім компілює набір даних для отримання відповідного висновку. SPSS спроектований таким чином, що може обробляти великий набір форматів змінних даних.

Як SPSS допомагає в дослідженнях та програмах аналізу даних:

SPSS - це революційне програмне забезпечення, яке використовується в основному вченими-дослідниками, яке допомагає їм обробляти важливі дані простими кроками. Робота з даними – складний та трудомісткий процес, але це програмне забезпечення може легко обробляти та оперувати інформацією за допомогою деяких методів. Ці методи використовуються для аналізу, перетворення та створення характеристичного шаблону між різними змінними даними. На додаток до цього висновок може бути отриманий через графічне уявлення, щоб користувач міг легко зрозуміти результат.

Data Transformation. Цей метод використовується для перетворення формату даних. Після зміни типу даних він поєднує дані одного типу в одному місці, і ними стає легко керувати. Є можливість вставляти різні типи даних у SPSS, і додаток змінить свою структуру відповідно до специфікації та вимог системи. Це означає, що навіть якщо зміниться операційна система, SPSS все одно зможе працювати зі старими даними.

ANOVA (дисперсійний аналіз). Це статистичний підхід до порівняння подій, груп чи процесів та виявлення відмінностей між ними. Допомагає зрозуміти, який метод найбільше підходить для виконання завдання. Дивлячись на результат, можна переконатися у доцільності та ефективності того чи іншого методу.

MANOVA (багатомірний дисперсійний аналіз): цей метод використовується для порівняння даних випадкових величин, значення яких невідоме. Техніку MANOVA також можна використовувати для аналізу різних типів населення та того, які фактори можуть вплинути на їхній вибір.

Можна перерахувати плюси цієї програми[9] те, що вона має просту у використанні систему меню для базового аналізу (вихід за рамки основ більшості дослідницьких потреб зазвичай вимагає синтаксису чи іншого пакета), також можна легко отримати висновок та у зручному для читання форматі. До версії 16 легко переміщатися за даними. Але, насамперед, застосунок не спрямований безпосередньо на потреби академічних досліджень. Ще можна відмітити, що

програма надмірно багато коштує для студентів. Наприклад, для базової версії потрібна місячна передплата у розмірі 99 \$, а для професійної версії цінік піднімається до 5800 \$ на рік. Нові версії не повністю об'єднуються по дизайну та типові аналізи з деяких дисциплін не зазнали суттєвих оновлень протягом багатьох років, також, зазвичай розробники випускають із помітними помилками. І в додаток ця програма не є справжньою мовою програмування. Також можна проаналізувати застосунок, який по кількості користувачів не поступається SPSS, це R software.

1.4. Мова програмування R

R — це мова програмування для статистичних обчислень та графіки, яку можна використовувати для аналізу та графічного відображення даних. Програма широко використовується дослідниками з різних дисциплін для оцінки та відображення результатів, а також викладачами статистики для дослідження методів. Софт є безкоштовним, що робить його привабливим варіантом, але для виконання роботи покладається на програмний код — замість меню або кнопок, що розкриваються.

R грає дуже значної ролі у науці про данних[10]. Ось у чому суть.

По перше, можна запускати свій код без будь-якого компілятора - R - це мова, що інтерпретується. R інтерпретує код та спрощує розробку коду. Багато обчислень виконуються з векторами, так як ця програма – векторна мова, можна додавати функції до одного вектора, не зациклюючись. Отже, R потужніший і швидший, ніж інші мови.

По друге, статистична мова - R використовується в біології, генетиці, а також у статистиці. Це мова повного циклу, якою можна виконувати завдання будь-якого типу. R не тільки не допоможе вам у технічних областях, але й стане чудовою підмогою у вашому бізнесі. Тут основна причина полягає в тому, що R має відкритий вихідний код, тому його можна модифікувати та поширювати

відповідно до потреб користувача. Він чудово підходить для візуалізації та має набагато більше можливостей у порівнянні з іншими інструментами.

Плюси та мінуси використання R[11]. R може збирати дані з Інтернету за допомогою веб-скрапінгу та інших засобів. Він також може виконувати очищення даних. Очищення даних – це процес виявлення та видалення/виправлення неточних або пошкоджених записів. Додаток також корисний для обробки даних, тобто процесу перетворення необроблених даних у бажаний формат для зручнішого використання.

R сумісний з іншими мовами, такими як C, C++ та FORTRAN. Інші мови, такі як .NET, Java, Python також можуть безпосередньо маніпулювати об'єктами. Також, R містить кілька пакетів, що дозволяють взаємодіяти з базами даних. Деякі з цих пакетів - Roracle, Open Database Connectivity Protocol), Rmysql тощо. Ще одна відмінність, що R не залежить від машини. Він підтримує кроссплатформенну роботу. Таким чином, його можна використовувати у багатьох різних операційних системах.

З мінусів можна відмітити те, що команди R не пов'язані з керуванням пам'яттю. В результаті R може зайняти весь доступний простір. R — гнучка мова програмування, і немає жодних суворих правил, яким слід дотримуватися. Необхідно підтримувати належні стандарти кодування, щоб уникнути брудного та складного коду. Також, R не має спеціальної групи підтримки, щоб допомогти користувачеві з його питаннями та проблемами. Але спільнота досить велика, тому всі допомагають одна одній.

1.5. Мова програмування SAS

Ще одна подібна до R програма для аналізу даних це SAS.

SAS - це величезний набір програмного забезпечення, починаючи від управління даними, операцій з базами даних, статистики, дослідження операцій, географічних інформаційних систем, контролю якості та багато іншого.

Він використовується в багатьох областях та для багатьох цілей. Його широко використовують багато корпорацій, а також університети, дослідні компанії тощо.

SAS розшифровується як система статистичного аналізу та написана мовою C. SAS використовується в більшості операційних систем. SAS можна використовувати як мову програмування, а також графічний інтерфейс. Він був розроблений Ентоні Джеймсом Барром і може зчитувати дані з електронних таблиць та баз даних. Результат може бути представлений у вигляді таблиць, графіків та документів. SAS використовується для складання звітів, отримання та аналізу статистичних даних, а також для виконання запитів SQL[12].

В основному існує чотири типи ПЗ SAS:

1. SAS для Windows. Великі компанії та навчальні заклади вважають за краще використовувати SAS Windows. Крім того, у SAS Windows є безліч утиліт, які допомагають скоротити час, необхідний для написання коду.

2. SAS Enterprise Miner - це прогресивний аналітичний інструмент інтелектуального аналізу даних, який покликаний допомогти користувачам розробляти описові та прогностичні моделі у процесі інтелектуального аналізу даних.

3. SAS STAT Software. Програмне забезпечення SAS STAT використовується виключно для статистичних процесів. Так само для бізнес-аналітики, прогнозного аналізу, описового та швидкого аналізу тощо.

4. SAS Enterprise Guide (EG) - це інструмент, що працює за принципом «вкажи та клацни», керований за допомогою меню та «майстра», який дозволяє користувачам аналізувати дані та публікувати свої результати. Він забезпечує прискорене навчання для швидкого дослідження даних, створення коду підвищення продуктивності, прискорення розгортання аналізу та прогнозів[13].

Аналізуючи цей застосунок можна побачити багато переваг, наприклад як те, що SAS може зчитувати файли даних, створені другими статистичними пакетами. Крім того, SAS дозволяє включати SPSS, Minitab, Excel, Stata та інші

файли даних безпосередньо в програму SAS або за допомогою програмного забезпечення для перетворення файлів. Основною перевагою вивчення цього застосунку є те, що це мова четвертого покоління. Ще SAS надає простий спосіб використання графічного інтерфейсу та багатьох програм. SAS – дуже зрозуміла мова. Процес налагодження простий. Можна зрозуміти та виправити помилку, про яку ясно йдеться у вікні журналу[14]. Однак, наприклад, R завжди реалізовував новий алгоритм, пов'язаний з машинним навчанням швидше, ніж SAS. Основна причина в тому, що будь-хто може працювати з R, оскільки це відкритий вихідний код, але це не стосується SAS. SAS працює у закритому середовищі.

Таким чином, алгоритми, які містяться у процедурах SAS, не призначені для загального користування. Вони доступні у ліцензійній версії, але недоступні для громадського дослідження. Також, R має більшу доступність для розширеної графіки. Його графічне уявлення набагато яскравіше і сумісніше, ніж у SAS. У ньому більше наочних графіків та діаграм.

1.6. Програмне забезпечення Matlab

Також не менш популярний застосунок для аналізу даних являється програмне забезпечення Matlab.

MATLAB – це аббревіатура від Matrix Laboratory. Це тому, що він був уперше створений як матрична мова програмування. Призначений для інженерів та вчених. Програмне забезпечення Matlab – відома мова програмування четвертого покоління, така як Java, C+. В даний час програма використовується в передових додатках, таких як машинне навчання, глибоке навчання та наука про дані.

Це дуже важлива програма, яку повинні вивчити багато студентів інженерних спеціальностей та інженерів. Matlab допомагає виконувати математичні розрахунки, проектування, аналіз та оптимізацію (структурну та математичну), а також надає швидкість, точність та чіткість результатам[15].

MATLAB особливо використовується в інженерних програмах. Цей софт часто використовується в системному аналізі та математичних розрахунках та їх візуалізації. Основні сфери застосування включають:

1. Обчислення чисельної лінійної алгебри[10]
2. Машинне навчання
3. Наука про дані
4. Моделювання
5. Створення діаграм для великих даних
6. Аналіз та візуалізація даних

Найбільшою перевагою MatLab є той факт, що він полегшує візуалізацію даних для користувача. Замість того, щоб покладатися на деякі фундаментальні знання з кодування або комп'ютерних наук, MatLab дозволяє користувачам отримати більш інтуїтивне уявлення про свої дані. Зручний для читання формат, в якому дані представлені як до, так і після аналізу, дозволяє користувачам зберігати жорсткий контроль над тим, як дані представлені. Це, мабуть, найважливіше тим, хто не має просунутої статистичної освіти; немає необхідності розбиратися в тонкощах процедури, якщо зміни легко видно у таблиці чи графіку. Це також означає, що MatLab може проводити аналізи, зручні для нефахівців, і може зробити великомасштабний аналіз даних зручнішим для бізнесу, який має справу з ним.

Перевага цієї візуалізації не обмежується двовимірним характером інших програм, таких як SPSS або SAS, і, отже, дозволяє краще концептуалізувати дані у реальному сценарії. Плюси та мінуси використання MATLAB.[16] Він простий у використанні завдяки інтерфейсу командного рядка та файлової структури, ще MATLAB не залежить від платформи і, отже, може бути встановлений у різних операційних системах, таких як Windows, Vista, Linux та Macintosh. Також, MATLAB має величезну вбудовану бібліотеку функцій для багатьох визначених завдань. Це робить роботу більш комфортною та економить час. Ці функції доступні як частина різних наборів інструментів, які

включають обробку сигналів, обробку зображень, зв'язок, системи управління, нейронні мережі тощо. MATLAB надає інструмент для розробки програм на основі графічного інтерфейсу користувача, які також можуть ліцензуватися на основі MAC-адреси для клієнтів з використанням кількох рядків коду сценарію. Застосунок пропонує команди, пов'язані з графікою та візуалізацією, які не залежать від пристроїв. Програми MATLAB компілюються в машинно-незалежний p-код. І ще одна відмінність, що помилки легше виправити, оскільки це мова, що інтерпретується.

Однак MATLAB є мовою, що інтерпретується, тому для її виконання потрібно більше часу, ніж для інших скомпільованих мов, таких як C, C++.

Також потрібен швидкий комп'ютер із достатнім обсягом пам'яті. Це збільшує вартість для людей, які бажають використовувати його для програмування. В додаток до цього, річна передплата коштуватиме 940 \$. Довічна - 2350 \$.

Важко розробляти програми реального часу з використанням MATLAB, оскільки програма знаходиться «поверх» вікон.

Тим не менш, Matlab, в першу чергу, призначений для використання в якості математичного інструменту і має перевагу в цій галузі, володіючи набагато кращими можливостями роботи з математичними функціями та великими матрицями. А ось застосунок Minitab має перевагу у статистиці, пропонуючи ширший спектр методів.

1.7. Програмне забезпечення Minitab

Програмне забезпечення Minitab — це хмарний статистичний інструмент, розроблений, щоб допомогти організаціям у різних галузях, таких як виробництво, охорона здоров'я, енергетика, автомобілебудування. Або можна прогнозувати закономірності та візуалізувати дані. Функції включають статистичне моделювання, спільне використання даних, часові ряди, симуляції та розподіл.

Введення даних спрощене, тому його можна легко використовувати для статистичного аналізу, а також це допомагає маніпулювати набором даних. Якщо дано тенденції, закономірності та діаграми, вони легко аналізуються та інтерпретуються, щоб можна було зробити остаточний висновок. Є також кілька функцій, які допомагають користувачам досягти покращення, а також спростити робочі процеси. Спільне використання, розгортання та тиражування проекту спрощується за допомогою Project Roadmaps. Статистику також можна відстежувати в режимі реального часу за допомогою Instant Insights, яка автоматично відображає дані на панелі приладів програми[17].

Проблеми, які виникають під час використання Minitab, можуть бути легко вирішені службою підтримки клієнтів, а також навченими користувачами. Програмне забезпечення також має розраховану на багато користувачів ліцензію, тому компанії можуть дійсно використовувати його в повній мірі.

З корисного функціоналу, застосунок має аналіз системи виміру: MSA — це математичний метод, який реалізований у програмі визначення кількості відхилень, які існують у процесі виміру. Мінливість процесу може впливати на загальну дисперсію процесу. Також, функцію Reliability/Survival – яка дозволяє вибрати найкращий розподіл для моделювання даних. Це допоможе визначити, яка функція найкраще описує ваші дані[18].

Незважаючи на те, що програма відносно проста у використанні, має велику кількість функцій, і користувачеві потрібно багато часу, щоб повністю освоїти це програмне забезпечення. Немає підтримки інших операційних систем, крім Windows.

Ціна ліцензійного продукту вже відносно велика (1610 \$ на рік), для використання ширшого спектра функцій потрібно доплачувати. (До 3000 \$ на рік). Підсумовуючи, програмне забезпечення Minitab це простий статистичний пакет програмного забезпечення, який дійсно допомагає вивчити та зрозуміти дані, які користувач збирає.

Для навчання, наприклад студентам, є безліч корисних вбудованих функцій, таких як:

- Регресія: ця функція дозволяє користувачам знаходити взаємозв'язок між змінними (що є ключовою функцією будь-якого статистичного інструмента). Регресія доступна у вигляді лінійної, нелінійної, порядкової, номінальної тощо.
- Статистичний контроль процесу: ця функція допомагає створювати причинно-наслідкові діаграми, контрольні діаграми змінних, багатоваріантні контрольні діаграми, діаграми зважені за часом і т.д.
- Доступно багато навчальних посібників, які допомагають учням під час використання програмного забезпечення.

1.8. Програмне забезпечення Scilab

Іншою найкращою альтернативою Minitab є Scilab, який використовується для математичних обчислень.

Програмне забезпечення Scilab – це програмне забезпечення для аналізу даних з відкритим вихідним кодом, яке охоплює математику, статистику, оптимізацію та обробку сигналів, а також керує функціями системи. Програма пропонує широкий набір інструментів для візуалізації даних, розробки алгоритмів, розробки програм та чисельного аналізу. Існує безліч способів, якими користувачі можуть представляти свої дані, від 2D до 3D, анімованих фігур, діаграм, графіків і т. д. [19]. Можна завантажити зовнішні модулі для подальшого налаштування їх подання у програмному забезпеченні. Scilab використовує мову програмування, що ідеально підходить для прототипування алгоритмів. Програма може обробляти та створювати складні алгоритми, просто вводячи кілька рядків коду в програмне забезпечення за допомогою Scinotes або редактора змінних. Коли справа доходить до чисельного аналізу, програма пропонує безліч методів для математичних і наукових обчислень.

Програма сумісна з Windows, Mac та Linux.

Проаналізувавши перелік вбудованих функцій, можна назвати основні якості продукту[20].

Стандартна установка програмного забезпечення Scilab постачається з базовим набором функцій:

- основні математичні операції (додавання, віднімання, множення, розподіл тощо).
- тригонометричні функції (синус, косинус, тангенс, котангенс та ін.)
- логічні операції (І, АБО, НЕ та ін.)
- ці базові оператори можуть застосовуватися до різних типів даних (булеві значення, цілі числа, числа з плаваючою комою, рядки тощо)
- елементарні функції (мінімум, максимум, абсолютне значення тощо)
- маніпуляції з матрицями (транспонування, твір та сума елементів масиву тощо)
- обумовлені користувачем функції
- програмування (умови та цикли)

Дослідив додаток, можна зробити підсумок.

Велика перевага даного ПЗ – це те, що по факту програма безкоштовна. Не дивлячись на це, що в ній доступно велика кількість корисних функцій[21]. Скрипт Scilab може містити всередині себе будь-яку кількість функцій (і даних), а виконання скрипта завантажує всі функції та дані в пам'ять. А, наприклад, у співставленні з MATLAB, будь-яка функція повинна бути записана в окремому файлі з таким же ім'ям, як і у функціях, але з розширенням .m. Ще один із можливих плюсів полягає в тому, що програмне забезпечення Scilab з відкритим вихідним кодом. Доступ до вихідного коду дає вам можливість видати код, покращити його та додати нові функції, а також розділити свій вклад з іншими.

Але Scilab іноді гальмує і потребує багато пам'яті. Час від часу він зависає, що теж погано, але для коротких кодів таких проблем немає. Підтримка, однак, навіть не близька до порівняння з конкурентами, набагато складніше

знайти більше внутрішніх функцій і підказок. Менше функцій, чим у приватних програмах типу MatLab.

1.9. Мова програмування Python

Наступний застосунок, який підходить для аналізу даних, це Python. Scilab добре підходить для тих, хто дивиться не на точну конструкцію програми, а на предметну область. Хоч і Python — це мова програмування загального призначення, що вимагає додаткових бібліотек навіть для виконання базових математичних операцій, але все одно можна відокремити корисні сторони даного програмного забезпечення.

Мова програмування Python - це інтерпретована мова високого рівня з об'єктно-орієнтованим підходом. Мова використовується для розробки API, штучного інтелекту, веб-розробки, Інтернету речей і т.д.

Частина того, чому Python став таким популярним, полягає в тому, що він широко використовується серед фахівців за даними. Це одна з найпростіших мов для вивчення, вона має вражаючі бібліотеки та ідеально підходить для будь-якого етапу обробки даних.

Обробка даних та моделювання[22]. На цьому етапі використовуються дві основні бібліотеки: NumPy та Pandas.

NumPy, скорочення від Numerical Python, надає ефективний інтерфейс для зберігання і маніпулювання великими даними мовою програмування Python. NumPy надає функції, які ви можете викликати, що робить його особливо корисним для роботи з даними.

У тому чи іншому сенсі масив NumPy схожий на вбудований у Python тип списку, але масиви NumPy забезпечують набагато ефективніше зберігання та операції з даними під час збільшення їхнього обсягу. NumPy пропонує особливий різновид масивів, що використовує багатовимірні масиви, звані ndarrays, або N-вимірні масиви.

Pandas пропонує дві структури даних: серію (список елементів) та фрейм даних (таблицю з кількома стовпцями). Ця бібліотека перетворює дані в фрейм даних, дозволяючи видаляти або додавати до нього нові стовпці і виконувати різні операції. Pandas надає розширені структури даних для зберігання різних типів маркованих і реляційних даних. Це робить Python дуже гнучким і надзвичайно корисним для очищення та маніпулювання даними.

Pandas має високу гнучкість і надає функції для виконання таких операцій, як злиття, зміна форми, об'єднання і конкатенація даних.

Matplotlib та Seaborn широко використовуються для візуалізації даних Python. Це означає, що вони допомагають перетворювати довгі списки чисел на зрозумілі графіки, гістограми, кругові діаграми, теплові карти тощо. Звичайно, бібліотек набагато більше, ніж було згадано. Python пропонує безліч інструментів для проектів аналізу даних та може допомогти у виконанні будь-якого завдання в рамках процесу[20].

Плюси та мінуси[23]. Велика спільнота. Python існує вже деякий час і об'єднує багатьох розробників Python завдяки його використанню в різних сферах ІТ. Отже, якщо розробник будь-коли застрягне, він, швидше за все, швидко і без зусиль знайде рішення за допомогою спільноти. Python — одна з найпростіших мов для вивчення завдяки ясному синтаксису та зручності читання. Таким чином, можна швидко вивчити мову та зайнятися проектами з аналізу даних. Додатковою перевагою зрозумілого синтаксису та легкої читабельності є швидкість самої розробки: розробнику не потрібно надто багато думати під час написання, а код легше налагоджувати. Дуже багатий перелік різноманітних бібліотек для різних завдань. Також Python можна використовувати в багатьох областях та проектах, він працює швидше завдяки гіпергнучкості і може використовуватися з будь-яким інструментом швидкої розробки програм.

Однак, розробка різних програм спрощується завдяки динамічній типізації, яка відмінно підходить для багатьох цілей Python. Але для аналізу

даних це є недоліком, тому що уповільнює пошук помилок даних, пов'язаних із присвоєнням однієї і тієї ж змінної різних даних. Ще проблема у тому, що споживання пам'яті Python дуже велике. Тому це може бути не найкращий вибір для завдань з інтенсивним використанням пам'яті. Багатопоточність не дуже хороша в Python через глобальне блокування інтерпретатора (GIL). GIL - це просто м'ютекс, який дозволяє одночасно виконувати лише один потік.

В результаті виконання коду в багатопотокових програмах з прив'язкою до процесора може бути повільніше, ніж однопоточних.

1.10. Програмне забезпечення GraphPad Prism

Програмне забезпечення GraphPad Prism поєднує в собі наукову побудову графіків, всебічну апроксимацію кривої (нелінійна регресія), зрозумілу статистику та організацію даних. GraphPad Prism спочатку був розроблений для біологів-експериментаторів у медичних школах та фармацевтичних компаніях, особливо у галузі фармакології та фізіології. Програмне забезпечення тепер набагато ширше використовується біологами всіх видів, а також соціальними та фізичними вченими. Це програмне забезпечення пропонує користувачам повну бібліотеку статистичного аналізу, включаючи ANOVA, непараметричні порівняння, t-тести та таблиці непередбачених обставин. Програмне забезпечення GraphPad Prism Statistical Analysis надає вісім типів відформатованих таблиць даних, спеціально призначених для різного аналізу, щоб забезпечити точне введення даних та звіти. Крім того, користувачі будуть отримувати оновлення в режимі реального часу, коли в їх дані вносяться зміни та результати аналізу[24].

Можна відмітити такі плюси цього застосунку: коли у дані та аналіз вносяться будь-які зміни, ці зміни миттєво відображаються в результатах, графіках і макетах з оновленнями в режимі реального часу. Також користувачі можуть експортувати свої графіки у високій якості та налаштовувати тип файлу, роздільну здатність, прозорість, розміри, колір, простір і т. д. своїх візуалізацій

відповідно до вимог публікації. Щоб заощадити час у майбутньому, користувачі можуть встановити параметри експорту за замовчуванням.

Однак, це програмне забезпечення не пропонує можливості написання скриптів, які так само корисні, як в R або Python. Це програмне забезпечення розроблено так, щоб пропонувати інтерфейс клацання, а не інтерфейс командного рядка. Він призначений для простого використання з інтерфейсом клацання. Було б непогано, якби для цього програмного забезпечення могли співіснувати командний рядок і інтерфейс клацання[25].

Немає допоміжної літератури. Допомоги із зовнішніх джерел дуже мало. Програмне забезпечення вимагає передплати, яка є досить дорогою в порівнянні з іншими конкурентами.

1.11. Мова програмування Java

Також на ринку застосунків для статистичного аналізу є Java.

Мова програмування Java - одна з найбільш затребуваних мов програмування, що використовуються сьогодні. Це незалежна від платформи, корисна і надійна мова. Розробники в усьому світі використовують Java для створення додатків, веб-інструментів і платформ для розробки програмного забезпечення. Java також знаходить широке застосування в машинному навчанні та науці про дані[26].

Java має безліч чудових фреймворків для науки про дані. Ці фреймворки надають розробникам базову функціональність і допомагають їм економити час і гроші. Прикладами популярних фреймворків для машинного навчання є:

Deeplearning4J - це інструментарій глибокого навчання з відкритим вихідним кодом для Java для розгортання нейронних мереж. Він може бути інтегрований з Hadoop і Spark.

ND4J - розшифровується як N Dimension-array objects for Java. Це набір інструментів для наукових обчислень, обробки сигналів і лінійної алгебри. Він має вбудовані бібліотеки, такі як numpy і MATLAB.

Apache Mahout - Це масштабований і розподілений фреймворк алгебри, що масштабується. Він допомагає в класифікації, кластеризації та рекомендаціях.

Java має унікальний синтаксис. Цей синтаксис дозволяє розробникам зрозуміти угоди, вимоги до змінної та методологію кодування. Java сильно типізована - тобто кожен тип даних уже зумовлений у структурі мови, і всі змінні мають бути частиною будь-якого типу даних[27].

Мова програмування Java - це об'єктно-орієнтована, універсальна та унікальна мова, яка пропонує масу функціональних можливостей. Її відмінна продуктивність і швидкість роблять її однією з найбільш затребуваних на ринку. Вона також забезпечує можливості безпеки, програмування, орієнтованого на мережу, і незалежності від платформи.

Java - не найпростіша мова програмування в цій галузі науки про дані. Вона пропонує сторонні бібліотеки з відкритим вихідним кодом, і будь-який java-розробник може впровадити Machine Learning і зайнятися наукою про дані. Однак новачки в галузі науки про дані віддають перевагу таким мовам, як Python і R, оскільки вони відносно простіші, ніж Java.

РОЗДІЛ 2

ОБҐРУНТУВАННЯ ВИБРАНИХ ТЕХНОЛОГІЙ ДЛЯ АВТОМАТИЗОВАНОЇ ОБРОБКИ СИГНАЛІВ

Постійно зростаюча кількість наукових досліджень породжує все більші, складніші та мультимодальні набори даних. Це призводить до того, що завдання аналізу даних стають більш вимогливими. Щоб допомогти впоратися з цими новими проблемами, більше дисциплін тепер повинні включити передові методи візуалізації у свої стандартні дані методи обробки та аналізу. Хоча багато

систем було розроблено, щоб дозволити вченим досліджувати, аналізувати та візуалізувати свої дані, багато з цих рішень є специфічними для домену, що обмежує їх сферу застосування як загальні засоби обробки інформації.

Один із способів підвищити їхню гнучкість – це побудова на основі інтерпретованої мови.

Мова програмування Python забезпечує середовище розробки, придатне як для завдань обчислення, так і для візуалізації. Однією з ключових переваг Python є можливість використання пакетів для розширення мови програмування, щоб забезпечити розширені можливості, такі як маніпулювання масивами та матрицями, обробки зображення, цифрової обробки сигналів та візуалізації.

У даному розділі буде розглянуто чому для виконання автоматизованої системи обробки сигналів була використана мова програмування Python та додаткові до неї бібліотеки Django та Pandas. Також буде проведено аналіз даних технологій та їх переваги і недоліки в порівнянні з іншими доступними технологіями.

2.1 Що таке Python

Python — це інтерпретована об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Його високорівневі вбудовані структури даних у поєднанні з динамічною типізацією та динамічним зв'язуванням роблять його дуже привабливим для швидкої розробки додатків, а також для використання як мови сценаріїв або з'єднувальної мови для з'єднання існуючих компонентів. Простий, легкий для вивчення синтаксис Python підкреслює читабельність і, отже, знижує вартість обслуговування програми. Python підтримує модулі та пакети, що заохочує модульність програми та повторне використання коду. Інтерпретатор Python і обширна стандартна бібліотека доступні у вихідному або двійковому вигляді безкоштовно для всіх основних платформ і можуть вільно поширюватися[28].

Однією з причин, що роблять Python ідеальною мовою для аналізу даних є те, що це мова програмування з відкритим кодом. Окрім цього, Python мають

високу швидкість, і існує велика підтримка, доступна для Python, – це ще кілька причин, які роблять Python улюбленим для багатьох. Насправді люди, які займаються аналізом даних, також мають можливість спробувати багато різних речей. Python також дозволяє розробникам розгортати програми та запускати прототипи, що робить процес розробки набагато швидшим. Після того, як проект на шляху до того, щоб стати аналітичним інструментом або додатком, його можна перенести на більш складні мови, такі як Java або C, при необхідності.

Python є не лише академічною мовою програмування, на прикладі якої можна тільки навчати основам програмування і далі про неї забути. Серед областей, де активно застосовується мова Python, можна виділити: системне програмування, розробку динамічних веб-сайтів, інтеграцію компонентів, розробку програм для роботи з базами даних, швидке створення прототипів, розробку програм для наукових обчислень, навчання, розробку ігор[29].

2.2 Переваги та недоліки мови програмування Python

Серед відомих перспектив мови Python, можна виділити такі переваги як:

- **Простий у використанні та вивченні:** для початківців Python простий у використанні. Це мова програмування високого рівня, а її синтаксис схожий на англійську мову. Ці причини роблять мову легкою для вивчення та адаптації. Порівняно з Java і C, у Python аналогічне завдання можна виконати за допомогою меншої кількості рядків коду. Завдяки легкому освоєнню принципи Python можна виконувати швидше порівняно з іншими мовами.

- **Підвищення продуктивності:** Python є дуже продуктивною мовою. Проста природа Python допомагає розробникам зосередитися на вирішенні проблем. Щоб зрозуміти синтаксис і поведінку мови програмування, користувачам не потрібно витрачати години, які йдуть на виконання більшої кількості роботи.

- **Гнучкість:** ця мова є дуже гнучкою, і тому вона дозволяє користувачеві випробовувати нові речі. Користувачі можуть розробляти нові

види програм, і ставити перед собою різні задачі. Мова не обмежує користувача спробувати щось інше. Інші мови програмування не надають такої гнучкості та свободи, тому Python є кращим у цих питаннях.

- **Велика бібліотека:** Python надає користувачеві величезну відкриту бібліотеку. Стандартна бібліотека Python величезна, і майже кожна функція, яку потрібно виконати, доступна в її бібліотеці. Це вдалось завдяки величезній підтримці спільноти та корпоративного спонсорства. Під час роботи з Python користувачеві не обов'язково використовувати зовнішні бібліотеки.

- **Підтримуюча спільнота:** Мова Python була створена багато років тому, саме з цієї причини вона має зрілу спільноту, яка може підтримувати будь-який тип розробника, починаючи від рівня початківця до рівня експерта. Існує достатньо довідників, посібників і документації щодо мови програмування Python, яка допомагає розробникам швидше та краще розуміти мову. Завдяки підтримуючій спільноті, Python швидко розвивається порівняно з іншими мовами.

Попри всі переваги мови, є недоліки які для деяких розробників можуть стати основними, у виборі мови програмування. Серед основних недоліків можна виділити:

- **Швидкість:** порівняно з Java або C швидкість Python нижча. Python — це інтерпретована мова, яка динамічно типізується. Для виконання коду кожен рядок коду має бути чітко впорядкований, оскільки мова інтерпретується. Це займає багато часу, а отже, уповільнює процес виконання. Динамічна структура Python також уповільнює його швидкість, тому що під час виконання коду необхідно виконати зайву роботу. Тому у тих випадках, коли швидкість виконання програми в пріоритеті, Python використовується не дуже часто.

- **Споживання пам'яті:** Python має дуже високе споживання пам'яті. Причиною для цього послуговує його гнучкість до типів даних. Він використовує великий обсяг пам'яті. Python не є гарним вибором для завдань,

де користувач хоче оптимізувати пам'ять, для цієї мови потребується інтенсивне використання пам'яті.

- **Розробка для мобільних пристроїв:** переваги Python у серверних платформах і настільних комп'ютерах, а отже, це фантастична мова серверного програмування. Але це не підходить для мобільної розробки. Для мобільної розробки Python є досить поганим вибором. Оскільки він не ефективно використовує пам'ять та досить повільна відносно інших мов програмування.

- **Доступ до бази даних:** Python забезпечує просте програмування. Однак, коли він взаємодіє з базою даних, виникають деякі проблеми. У порівнянні з такими технологіями, як JDBC і ODBC, які є досить відомими, рівень доступу до бази даних мови програмування Python є примітивним і недорозвиненим. Великі підприємства, яким зазвичай потрібна плавна взаємодія зі складними застарілими даними, не віддають перевагу використанню Python.

- **Помилки виконання:** користувачі Python помічали різні проблеми, які вони помічали на рівні архітектури мови. Оскільки мова Python динамічно типізована, тип даних змінної може бути змінений у будь-який час. Через це доводиться витратити більше ресурсів і часу на тестування системи, а також в мові є помилки, які відображаються під час виконання.

- **Простота:** Python є простою та легкою у використанні мовою програмування, що також стає недоліком мови. Користувачі Python настільки звикають до його легкого синтаксису та обширної бібліотеки, що стикаються з проблемами під час вивчення інших мов програмування. Тому Python має дуже вразливий характер, і користувачі починають сприймати все легковажно.

2.3. Порівняння мови Python з іншими мовами програмування

Існують нескінченні дебати щодо найкращої мови програмування, C++ проти Java проти Python. В цілому, в цій темі не може бути одного, безпрецедентного переможця. Кожна мова програмування розроблена для вирішення конкретної проблеми та добре підходить для вирішення своєї

проблеми. Отже, важливо спочатку скласти список усіх вимог і пріоритетів, а потім вирішити, який шлях обрати.

C++ є швидкою та скомпільованою мовою програмування, яка стала популярною та є першою мовою програмування, яку вивчає програміст.

Java популярна завдяки своїй незалежності від платформи, і величезна кількість десктопних додатків були створені за допомогою цієї мови.

У наведеній нижче таблиці підсумовано основні відмінності між C++ , Java та Python.

Таблиця 2.1 – Порівняння мов програмування

Тема	C++	Java	Python
Компільована чи інтерпритована	Скомпільована мова програмування	Одночасно компілюється та інтерпретується.	Інтерпретована мова програмування
Залежність від платформи	залежить від платформи	не залежить від платформи	не залежить від платформи
Перевантаження оператора	підтримує перевантаження операторів	не підтримує перевантаження операторів	підтримує перевантаження операторів
Спадкування	забезпечує як одиночне, так і множинне успадкування	єдине успадкування можливе, тоді як множинне успадкування можна досягти за	забезпечує як одиночне, так і множинне успадкування

		допомогою інтерфейсів	
Підтримка потоків	не має вбудованої підтримки потоків; Це залежить від бібліотек	має вбудовану підтримку потоків	підтримує багатопотоковість
Час виконання	дуже швидко. Фактично, це основний вибір програмістів	набагато швидший за Python з точки зору швидкості виконання, але повільніший за C++	Через інтерпретатор Python повільно виконується

Продовження таблиці 2.1

Обробка програми	Функції та змінні використовуються поза класом	Кожна частина коду (змінні та функції) має бути всередині самого класу.	Функції та змінні можна оголошувати та використовувати поза класом
Підтримка бібліотеки	має обмежену підтримку бібліотек	забезпечує бібліотечну підтримку для багатьох концепцій, таких як UI	має величезний набір бібліотек і модулів
Довжина коду	Довжина коду менша, ніж у Java,	довжина коду більша, ніж у Python і C++.	має найменшу довжину коду

	приблизно в 1,5 рази		
--	-------------------------	--	--

Таким чином ми можемо наглядно зрозуміти в чому принципові відмінності і явні переваги та недоліки Python серед інших мов програмування.

2.4. Що таке Django web framework

Фреймворк — це набір модулів або пакетів, які допомагають у написанні веб-програми або допомагають у розробці веб-програми. Під час роботи з фреймворком нам не потрібно турбуватися про деталі низького рівня, такі як протоколи, сокет, керування потоками.

Framework робить життя розробника розумнішим і простішим, надаючи структуру для розробки додатків. Вони автоматизують впровадження звичайних рішень, що дає розробникам можливість зосередитися лише на логіці програми, а не на рутині. Вони надають загальні серверні швидкі, надійні та масштабовані веб-додатки, які легко підтримувати.

Frameworks надає функціональність для виконання наступних операцій для розробки веб-додатків: розробник швидко може збільшувати розмір самого додатку, містить десятки додаткових функцій, які можна використовувати для вирішення поширених завдань веб-розробки, зручна робота з базами даних SQL та NoSql типу.

Django – це високорівневий Python веб-фреймворк, який дозволяє швидко створювати безпечні та підтримувані веб-сайти. Створений досвідченими розробниками, Django бере на себе більшу частину клопоту веб-розробки, тому ви можете зосередитися на написанні свого веб-додатку без необхідності винаходити велосипед. Він безкоштовний і з відкритим вихідним кодом, має зростаючу та активну спільноту, відмінну документацію та безліч варіантів як безкоштовної, так і платної підтримки.

Django допомагає писати програмне забезпечення, яке буде:

- **Повним:** Django слідує філософії «Все включено» і надає майже все, що розробники можуть захотіти зробити «з коробки». Оскільки все, що вам потрібно, є частиною єдиного «продукту», все це бездоганно працює разом, відповідає послідовним принципам проектування та має велику та актуальну документацію

- **Різностороннім:** Django може бути (і був) використаний для створення практично будь-якого типу веб-сайтів – від систем керування контентом та wiki до соціальних мереж та новинних сайтів. Він може працювати з будь-яким клієнтським середовищем і може доставляти контент практично в будь-якому форматі (включаючи HTML, RSS-канали, JSON, XML тощо). Хоча Django надає рішення практично для будь-якої функціональності, яка вам може знадобитися (наприклад, для кількох популярних баз даних, шаблонізаторів тощо), внутрішньо він також може бути розширений сторонніми компонентами, якщо це необхідно.

- **Безпечним:** Django допомагає розробникам уникнути багатьох поширених помилок безпеки, надаючи фреймворк, розроблений для того, щоб «робити правильні речі» для автоматичного захисту сайту. Наприклад, Django надає безпечний спосіб керування обліковими записами користувачів та паролями, уникаючи поширених помилок, таких як розміщення інформації про сеанс у файли cookie, де вона вразлива (замість цього файли cookie містять лише ключ, а фактичні дані зберігаються в базі даних) або безпосереднє зберігання паролів замість хешу пароля.

Хеш пароля — це значення фіксованої довжини, створене шляхом обробки пароля через криптографічну хеш-функцію. Django може перевірити правильність введеного пароля, пропустивши його через хеш-функцію і порівнявши виведення зі збереженим значенням хеш. Завдяки «односторонньому» характеру функції, навіть якщо збережене хеш-значення скомпрометовано, зловмиснику буде складно визначити вихідний пароль. Django, за умовчанням, забезпечує захист від багатьох уразливостей, включаючи

SQL-ін'єкцію, міжсайтовий скриптинг, підробку міжсайтових запитів та клікджекінг.

- **Масштабованим:** Django використовує компонентну “shared-nothing” архітектуру (кожна її частина є незалежною від інших і, отже, може бути замінена або змінена, якщо це необхідно). Чіткий поділ частин означає, що Django може масштабуватися при збільшенні трафіку шляхом додавання обладнання на будь-якому рівні: сервери кешування, сервери баз даних або сервери додатків. Одні з найбільш завантажених сайтів успішно масштабували Django (наприклад, Instagram та Disqus, якщо назвати лише два з них).

- **Зручним у супроводі:** Код Django написаний з використанням принципів і шаблонів проектування, які заохочують створення коду, що підтримується і повторно використовується. Зокрема, у ньому використовується принцип "Don't Repeat Yourself" (DRY, "не повторюйся"), тому немає непотрібного дублювання, що скорочує обсяг коду. Django також сприяє групуванню пов'язаних функціональних можливостей у повторно використовувані «додатки» і, на нижчому рівні, групує пов'язаний код у модулі (відповідно до шаблону Model View Controller (MVC)).

- **Переносним:** Django написано на Python, який працює на багатьох платформах. Це означає, що ви не прив'язані до будь-якої конкретної серверної платформи і можете запускати програми на багатьох версіях Linux, Windows і Mac OS. Крім того, Django добре підтримується багатьма веб-хостингами, які часто надають певну інфраструктуру та документацію для розміщення сайтів Django[29].

2.4.1 Як виглядає код Django?

У традиційному інформаційному веб-сайті, веб-додаток чекає на запити HTTP від веб-браузера (або іншого клієнта). Коли запит отримано, програма розробляє те, що необхідно на основі URL-адреси та, можливо, інформації в POST або GET запитах. Залежно від того, що потрібно, він може читати або

записувати інформацію з бази даних або виконувати інші завдання, необхідні для задоволення запиту. Програма потім поверне відповідь веб-браузеру, часто динамічно створюючи сторінку HTML для відображення браузера, вставляючи отримані дані HTML шаблон.

Веб-застосунки написані на Django зазвичай групують код, який обробляє кожен з цих кроків, в окремі файли:

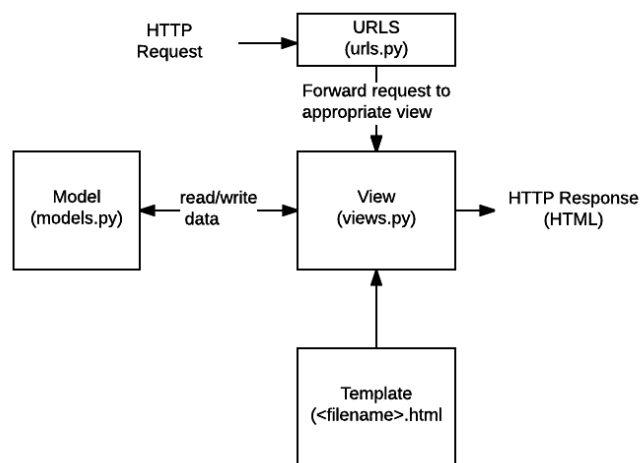


Рисунок 2.1 - MVT (Model View Template) діаграма фреймворку Django

- **URLs:** Хоча можна обробляти запити з кожної URL-адреси за допомогою однієї функції, набагато зручніше писати окрему функцію для обробки кожного ресурсу. URL-mapper використовується для перенаправлення HTTP-запитів у відповідне подання на основі URL-адреси запиту. URL-mapper також може вилучати дані з URL-адреси відповідно до заданого шаблону і передавати їх у відповідну функцію як аргументи.

- **View:** Уявлення (view) - це функція оброблювача запитів, яка отримує HTTP-запити та повертає відповіді. View має доступ до даних через моделі (необхідних для задоволення запитів і делегування відповіді в шаблони).

- **Models:** Моделі являють собою об'єкти Python, які визначають структуру даних програми та надають механізми для управління (додавання, зміни, видалення) та виконання запитів до бази даних.

- **Templates:** Template (шаблон) - це текстовий файл, що визначає структуру або розмітку сторінки (наприклад HTML сторінки), з полями для підстановки, що використовуються для представлення актуального вмісту. View може динамічно створювати HTML сторінки, використовуючи HTML шаблони та заповнюючи їх даними з моделі (model). Шаблон може бути використаний для визначення структури будь-яких типів файлів, не обов'язково HTML[30].

2.5 Переваги та недоліки Django

Як відомо, кожна монета має дві сторони, і Django також не є винятком. Перша сторона Django показує швидку розробку, швидку обробку, масштабованість, тоді як інша сторона розкриває його монолітність і нездатність створювати менші проекти.

Django сам по собі має багато переваг, але ми розглянемо деякі основні, які також відрізняють Django від інших фреймворків:

- **Реалізовано на Python** - Якщо ви вивчали Python або бачили код Python, ви також помітите, що Python дуже легко читати, і це основна філософія дизайну, яка лежить в основі створення Python.

Крім того, Python є досить потужним і використовується для реалізації наукових обчислень і штучного інтелекту високого рівня. Отже, Python і Django Framework, реалізовано з використанням однієї мови, тому це надає величезну підтримку у бекенді, але нічого не ставить під загрозу у фронтенді.

- **Краще підключення CDN і керування вмістом** - Ця функція є однією з причин того, що нові компанії та соціальні мережі, такі як Instagram, використовують Django в першу чергу через наявність більшого контролю над CDN.

CDN — це не що інше, як мережі доставки вмісту, оскільки, судячи з назви, це спеціальні сервери з всіма даними, який такі сайти, як Netflix і Amazon Prime, використовують для потокової передачі. Ці сервери містять лише мультимедіа та ресурси для вашої веб-сторінки, оскільки вони географічно

розташовані поблизу клієнта, тому можуть швидше обслуговувати весь ресурс і, таким чином, підвищувати задоволеність клієнтів.

Django надає бібліотекам і розробникам можливість використовувати його як CMS (систему керування вмістом) завдяки чудовому інтерфейсу адміністратора, який робить досить простим налаштування та запуск CDN. Сайти активно використовують CDN, які надають вам мультимедійний вміст, наприклад YouTube, Instagram, Google тощо.

- **Framework в який все включено** - Django створений веб-розробниками для веб-розробників, тому, звичайно, він вирішить загальні питання та проблеми, з якими стикаються розробники. Django Framework має стільки функціональних можливостей, що вам може навіть не знадобитися створювати нічого, окрім власної унікальної програми, і ось у чому полягає філософія дизайну Django DRY (Don't Repeat Yourself). Всі базові і найчастіше вживані компоненти вже були розроблені спеціалістами, і тому немає необхідності розроблювати один і той самий код двічі.

- **Швидка обробка** - Це одна з найвагоміших переваг в порівнянні з іншими фреймворками, оскільки архітектура Django відрізняється від усіх інших фреймворків у галузі. Це означає, що Django використовує архітектуру MTV, що робить весь процес передачі через Інтернет легшим і швидшим, оскільки ресурси можна розмістити на CDN. Сервер Django продуктивно справляється з проблемами, зберігаючи при цьому швидкість. Архітектура Django має значні відмінності від інших фреймворків.

- **Швидка розробка** - Причиною високої швидкості розробки є те, що архітектура Django MTV реалізує філософію слабозв'язаних компонентів. Це означає, що ми можемо працювати над різними компонентами паралельно, а потім набагато легше їх інтегрувати. Ця функція Django значно відрізняється від інших фреймворків, і наразі Django є найкращим для швидкої розробки в галузі програмування

- **Легко масштабувати** - Django було створено таким чином, що він зможе працювати з будь-якими видами апаратних доповнень. Ця перевага є основною причиною того, чому найзавантаженіші сайти світу, такі як Instagram, Pinterest, Disqus тощо, використовують фреймворк Django.

Django базується на слабозв'язаній архітектурі, яка надає йому можливість додавати функціонал в будь-якому місці компонентів, оскільки вони керуватимуть цією зміною. І ці зміни практично не впливають на інші компоненти, що робить цей фреймворк досить затребуваним в порівнянні з іншими.

- **Безпека** - Фреймворк Django створений кращими світовими веб-розробниками, які мають великий досвід і знання. Саме це робить безпеку Django на найвищому рівні через дуже малу ймовірність проломів у захисті, навіть в системі автентифікації користувачів.

Наприклад Laravel Framework у PHP передає дані через метод GET і навіть паролі, які ми вводимо, видимі, що є дуже високим ризиком, але Django має вбудований функціонал який про це подбає. Він також використовує метод GET для передачі даних, але паролі та вся важлива інформація автоматично шифруються за допомогою довгого ключа безпеки. Навіть у базі даних Django ми не можемо побачити пароль.

Django має весь функціонал, який може знадобитися в Інтернеті. Але все ж кожна технологія має деякі недоліки, і Django не є винятком. Хоча дані недоліки зв'язані не з продуктивністю і швидкодією, скільки з дизайном. Ось декілька із них:

- **Django монолітний** - Що ж, для когось це може стати гарним рішенням, але в більшості випадків це недолік. Django має певний набір файлів і попередньо визначених змінних. І перш ніж почати створювати проект на Django, вам потрібно дізнатись про всі ці файли.

Фреймворк Django має певний спосіб визначення та виконання завдань. Це логічна файлова структура, яку легко освоїти. Але також, це робить

обов'язковою умовою те, що ви не можете використовувати власну файлову структуру. З цієї причини, що фреймворк має широко відомий спосіб розробки як «спосіб Django». Якщо ви не дотримуетесь цих правил, можливо, ви не зможете розгорнути проект за допомогою Django.

Django фреймворк і сервер шукають інформацію в цих зарезервованих файлах і не змінюють її.

Тому багато розробників віддають перевагу flask, а не Django, але для цього фреймворку вам потрібно досить добре знати бекенд. Крім того, вам потрібно багато часу та знань, щоб створити власні структури та серверні шаблони, які не використовуються в Django. Тут вам потрібно лише більше часу приділяти своєму унікальному проекту, а не іншим речам.

- **Не для малих проектів** - Усі функції Django містять багато коду. Це вимагає серверної обробки та часу, що створює деякі проблеми для веб-сайтів низького класу, які можуть працювати навіть із дуже низькою пропускнуою здатністю. Крім того, Django є масштабованим і полегшує роботу розробника. Це означає, що Django має надавати унікальні функції та особливості, інакше яка різниця між Django та іншими фреймворками[31].

Після наведених вище факторів ми можемо зробити висновок, що Django є дуже потужним фреймворком, але інколи бажання зробити розробку більш зручнішою, може стати недоліком. Хоча Django не поступається продуктивністю, надаючи вам усі ці функції, вам просто потрібно вивчити правильні способи Django та те, як їх реалізувати, на відміну від будь-якої іншої структури.

2.6 Порівняння Django з іншими фреймворками

На сьогоднішній день існує безліч фреймворків, для різних типів задач та проектів, надалі буде проведено аналіз найпопулярніших Python фреймворків в порівнянні з Django, для роз'яснення чому він найкраще підходить для поставленої задачі.

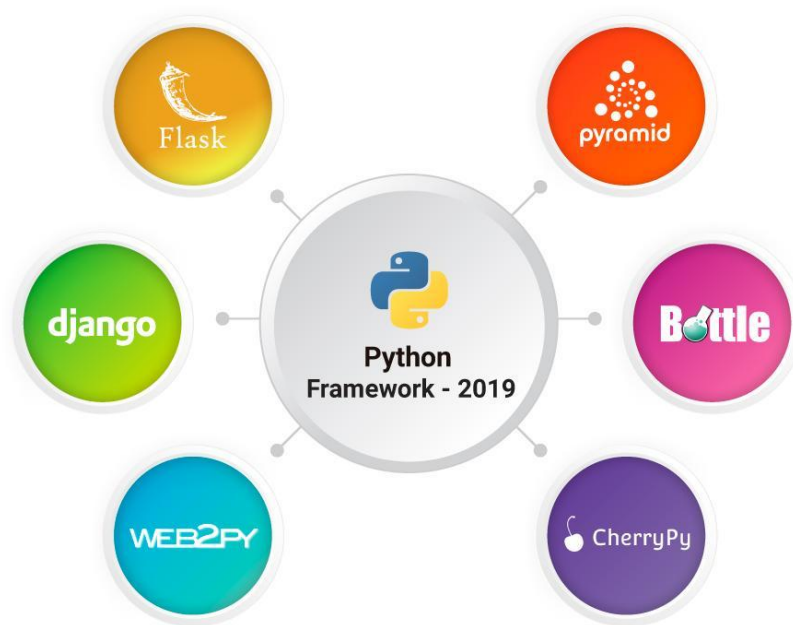


Рисунок 2.2 – Найпопулярніші Python фреймворки

Django:

Він належить до категорії фреймворків із повним стеком, які останнім часом набули популярності та вважаються одним із найкращих веб-фреймворків Python. Він дотримується принципу «Не повторюйся».

У Django є багато вбудованих бібліотек, які допомагають переходити з однієї бази даних в іншу. За замовчуванням Django може працювати з такими базами даних: MySQL, Oracle, PostgreSQL і SQLite, інші бази даних можна використовувати за допомогою сторонніх драйверів. Для відображення об'єктів у таблиці бази даних він використовує ORM.

Ключовий функціонал:

- Безпека є найважливішим пунктом цього фреймворка. У порівнянні з іншими фреймворками Python Django доводить свій найбільший ступінь захисту.
- Він забезпечує URL-маршрутизацію
- Він забезпечує підтримку автентифікації
- Django пропонує функцію міграції схеми бази даних

- Він надає адекватні попередньо зібрані бібліотеки для розробки повного стеку
- Django дотримується архітектури MVC-MVT. У цій архітектурі розробник просто повинен надати модель, уявлення та шаблон, потім користувач виконує відображення цього в URL-адресу, а потім решту завдань виконує сам Django.

Pyramid

Pyramid є частиною проекту Pylon; це одна з найбільш гнучких структур у мові Python. Цей фреймворк дозволяє розробнику вибирати що завгодно - СУБД, структуру URL, стилі шаблонів і багато іншого[32].

Ключовий функціонал:

- Гнучкий і адаптивний.
- Проста структура та легкість у вивченні.
- Прозорість і вимірювана якість.
- Управління безпекою.
- Гнучкі інструменти для аутентифікації та сценарію авторизації.
- Мови шаблонів і механізми, включаючи Jinja2 і Mako.
- Платформа обходу для зіставлення URL-адрес із кодом.
- Кешування HTTP.
- Вбудована підтримка зберігання сеансів.
- Створюйте традиційні програми RDBMS, орієнтовані на маршрути.

Web2Py

Це фреймворк категорії повного стека. Це масштабована структура з відкритим кодом, яка підтримує всі операційні системи. Web2Py має свою веб-базовану IDE, яка має всі функції, які повинна мати IDE, таку як налагоджувач, редактор коду та розгортання одним клацанням миші. Він не може використовувати Python 3.

Ключовий функціонал:

- Існує система квитків, у якій квиток надсилається користувачеві, якщо щось піде не так із фреймворком.
- Фреймворк незалежний від платформи.
- Є зворотна сумісність, яка гарантує, що розробка на новій версії фреймворку не втратиться зв'язок з вже винайденим кодом попередніх версій.
- Читабельність кількох протоколів
- Забезпечує контроль доступу на основі ролей
- Немає жодних передумов для його встановлення чи налаштування.
- Підтримує інтерналізацію

Flask

Ще одним популярним легким і мікро-фреймворком є Flask. Завдяки модульній конструкції його легше адаптувати. Використовуючи цей фреймворк, розробники можуть створювати надійні веб-програми, а створення таких програм полегшує використання будь-якого типу розширення.

Ключовий функціонал:

- Сумісність з Google App Engine
- Допоміжний шаблон jinja2 та інструментарій Werkzeug WSGI
- Вбудований відладчик
- Забезпечення модульного тестування
- Для активації клієнтських сеансів Flask забезпечує підтримку файлів cookie
- Спокійне відправлення запиту
- Іншою важливою особливістю цього фреймворку є те, що він базується на Unicode
- Також підтримується обробка запитів HTTP
- Ця структура також надає можливість підключення будь-якої ORM

Bottle

Цей фреймворк ідеально підходить для невеликих програм і в основному використовується для створення API. Це один із найбільш використовуваних веб-фреймворків Python, оскільки для створення програми не потрібні жодні інші залежності, окрім стандартної бібліотеки Python, програмісти просто можуть працювати з апаратним забезпеченням. Цей фреймворк створює єдиний вихідний файл і відноситься до категорії мікрофреймворків.

Ключовий функціонал:

- Використовуючи цей Фреймворк, ви отримуєте доступ до форм даних, файлів cookie, завантаження файлів та інших пов'язаних з HTTP метаданих.
- Маршрут відправлення запитів є ще одним ключовим елементом цього фреймворку.
- Він пропонує вбудований HTTP-сервер
- Має підтримку плагінів для різних баз даних.
- Також можна використовувати механізми шаблонів сторонніх розробників і сервери WSGI/HTTP

CherryPy

Одним із найстаріших мікрофреймворків є CherryPy. Він має мінімалістичний підхід. CherryPy — це об'єктно-орієнтована структура з відкритим кодом. Для доступу до даних або створення шаблону можна використовувати будь-яку технологію. Програми, створені за допомогою цього фреймворку, є автономними програмами Python, які мають вбудований багатопоточний сервер[33].

Ключовий функціонал:

- За допомогою цього фреймворку можна запускати декілька серверів одночасно

- Незалежний від платформи.
- Покриття, профілювання, тестування – це деякі з вбудованих функції, які підтримуються фреймворком.
- Він працює на Android.
- Він забезпечує хорошу систему конфігурації
- Є веб-сервер з пулом потоків, сумісний з HTTP WSGI.
- Кешування, кодування та автентифікація – ще деякі функції, які надає фреймворк

2.7 Що таке Pandas

Pandas — це пакет Python з відкритим вихідним кодом, який найбільш широко використовується для завдань науки про дані/аналізу даних і машинного навчання. Він побудований на основі іншого пакета під назвою NumPy, який забезпечує підтримку багатовимірних масивів. Будучи одним із найпопулярніших пакетів обробки даних, Pandas добре працює з багатьма іншими data science модулями в екосистемі Python і зазвичай входить до складу кожного дистрибутива Python, від тих, що постачаються з вашою операційною системою, до дистрибутивів комерційних постачальників, таких як ActivePython від ActiveState.

Насправді за допомогою Pandas ви можете робити все, що змушує провідних світових дослідників даних проголосувати за Pandas як за найкращий доступний інструмент аналізу та обробки даних[34].

Pandas має на меті бути основним будівельним блоком високого рівня для практичного аналізу реальних даних у Python. Крім того, він має ширшу мету — стати найпотужнішим і найгнучкішим інструментом для аналізу/маніпулювання даними з відкритим кодом, доступним будь-якою мовою.

Також не можна упустити найважливіші та основні моменти даного фреймворку, такі як:

- Швидкий і ефективний об'єкт DataFrame для обробки даних із вбудованим індексуванням
- Інструменти для читання та запису даних між структурами даних у пам'яті та різними форматами: CSV та текстові файли, Microsoft Excel, бази даних SQL та швидкий формат HDF5
- Інтелектуальне вирівнювання даних і інтегрована обробка відсутніх даних: отримайте автоматичне вирівнювання на основі міток під час обчислень і легко маніпулюйте хаотичними даними, перетворюючи їх в упорядковану форму
- Гнучка зміна форми та поворот наборів даних
- Інтелектуальне нарізання на основі міток, фантастичне індексування та піднабір великих наборів даних
- Стовпці можна вставляти та видаляти зі структур даних для зміни розміру
- Агрегування або перетворення даних за допомогою потужного механізму групування, що дозволяє виконувати операції розділеного застосування та об'єднання наборів даних
- Високопродуктивне злиття та об'єднання наборів даних
- Ієрархічна осьова індексація забезпечує інтуїтивно зрозумілий спосіб роботи з даними великої розмірності в структурі даних меншої розмірності
- Функціональні можливості часових рядів: генерація діапазону дат і перетворення частоти, статистика рухомого вікна, зміщення дати та відставання. Також можливість створювати часові зсуви для домену та об'єднання часових рядів без втрати даних
- Високо оптимізований для продуктивності, з критичними шляхами коду, написаними на Cython або C
- Python with pandas використовується в багатьох академічних і комерційних сферах, включаючи фінанси, нейронауку, економіку, статистику, рекламу, веб-аналітику тощо

2.7.1 DataFrame та Series

Щоб ефективно працювати з pandas, необхідно освоїти найголовніші структури даних бібліотеки: DataFrame та Series. Без розуміння що вони собою представляють, неможливо надалі проводити якісний аналіз.

Структура/об'єкт Series являє собою об'єкт, схожий на одномірний масив (пітонівський список, наприклад), але відмінною його рисою є наявність асоційованих міток, т.зв. індексів, вздовж кожного елемента зі списку. Така особливість перетворює його на асоціативний масив або словник на Python.

```
>>> import pandas as pd
>>> my_series = pd.Series([5, 6, 7, 8, 9, 10])
>>> my_series
0      5
1      6
2      7
3      8
4      9
5     10
dtype: int64
>>>
```

Рисунок 2.3 – Ініціалізація та відображення структури Series

У строковому поданні об'єкта Series індекс знаходиться ліворуч, а сам елемент праворуч. Якщо індекс явно не заданий, то pandas автоматично створює RangeIndex від 0 до N-1, де N загальна кількість елементів. Також варто звернути, що у Series є тип елементів, що зберігаються, в нашому випадку це int64, так як ми передали цілі численні значення.

Об'єкт DataFrame найкраще уявляти у вигляді звичайної таблиці і це правильно, адже DataFrame є табличною структурою даних. У будь-якій таблиці завжди присутні рядки та стовпці. Стовпцями в об'єкті DataFrame виступають об'єкти Series, рядки яких є безпосередніми елементами[35].

DataFrame найпростіше сконструювати з прикладу пітонівського словника:

```
>>> df = pd.DataFrame({
...     'country': ['Spain', 'England', 'USA', 'Ukraine'],
...     'population': [17.04, 143.5, 9.5, 45.5],
...     'square': [2724902, 17125191, 207600, 603628]
... })
>>> df
```

	country	population	square
0	Spain	17.04	2724902
1	England	143.50	17125191
2	USA	9.50	207600
3	Ukraine	45.50	603628

Рисунок 2.4 – Ініціалізація та відображення структури DataFrame

Об'єкт DataFrame має 2 індекси: по рядках і стовпцям. Якщо індекс рядків явно не заданий (наприклад, колонка за якою потрібно їх будувати), то pandas задає цілісний індекс RangeIndex від 0 до N-1, де N це кількість рядків у таблиці.

2.8 Переваги та недоліки Pandas

Бібліотека Pandas є надійним програмним забезпеченням і має багато переваг. Якщо перераховувати їх усі, це займе набагато більше часу, ніж на те, щоб вивчити бібліотеку.

Тут перераховані переваги, які є основою бібліотеки Pandas:

Відмінне відображення даних:

Бібліотека Pandas є ідеальним інструментом для тих, хто хоче заглибитися в науку про дані або аналіз даних, завдяки різним способам представлення та організації даних. Це дуже важлива функція, якою не можна нехтувати, оскільки неможливо проаналізувати або прочитати будь-які дані, якщо вони не представлені достатньо добре.

Добре організований, чистий набір даних важливий, коли дані важко аналізувати та читати.

Менше програмування, більше виконаної роботи:

Написавши 1-2 рядки коду в Pandas, ви можете легко виконувати завдання, які потребуватимуть близько 10-15 рядків коду на C++ або Java,

можливо навіть більше. Це втілює в собі ефективність, яку надає Pandas. У науці про дані є багато чого для практики, і тому це дуже корисна здатність для людей, які тільки починають працювати в цій галузі.

Зменшення непотрібного навантаження на кодування допомагає як ентузіастам, так і професіоналам науки про дані економити багато часу, що є дуже важливим для досліджень, які вони проводять.

Ефективна обробка величезних даних:

Як обговорювалося раніше, час дуже важливий, коли йдеться про data science. Тому надзвичайно важливо, щоб бібліотека, яка використовується, була дуже ефективною в часі. Це напрямок, у якому Pandas досяг найбільших успіхів. Уес МакКінні створив цю бібліотеку з єдиною метою — мати можливість працювати з великими обсягами даних швидше та краще, ніж будь-яка інша бібліотека у світі. Це робить його надзвичайно важливим для аналізу великої кількості даних.

Широкий набір функцій:

Ця бібліотека надає користувачеві великий набір команд і дивовижних функцій, які можна використовувати для аналізу наданих даних з максимальною легкістю.

Pandas допомогли аналізу даних вийти на абсолютно новий рівень. Це допомагає вам фільтрувати дані відповідно до встановлених вами умов, а також відокремлювати та сегментувати ваші дані відповідно до ваших власних уподобань.

Гнучкість даних і легке налаштування:

Pandas надає своїм користувачам величезний список функцій, які вони можуть застосувати до наявних у них даних, щоб редагувати та налаштовувати їх, змінюючи відповідно до власної волі та бажання. Це допомагає їм отримати максимальну віддачу від своїх даних і допомагає їм проаналізувати всю інформацію, яку вони мають у своєму розпорядженні.

З переліком переваг Pandas також має власні обмеження та недоліки, про які не менш важливо знати. Подалі будуть перераховані недоліки бібліотеки Pandas.

Складний синтаксис, який не завжди відповідає Python:

Коли ви використовуєте Pandas, знаючи, що це частина Python, деякі її синтаксичні конструкції можуть бути складними. Це проблема, оскільки багато користувачів не можуть ефективно та плавно перемикатися між звичайним кодом Python і Pandas. Однак така проблема виникає лише тоді, коли ви використовуєте розширені рівні Pandas.

Погана документація:

Це велика проблема з Pandas, особливо для новачків. Якщо будь-яка мова програмування чи бібліотека добре задокументована, це допомагає людям зрозуміти її справжній потенціал і дає уявлення про всі її функції та застосування. Через брак хорошої документації pandas, певним чином, стала ексклюзивною для власних користувачів і людей, які хочуть вивчити її чи спробувати. Хороша документація заохочує все більше користувачів вивчати бібліотеку чи мову.

Погана сумісність з 3D матрицями:

Це один із найбільш помітних недоліків бібліотеки Pandas. Якщо ваша робота стосується двовимірних (2D) матриць, то для вас не може бути нічого кращого, ніж pandas. Однак, як тільки ви оновите свої дані до тривимірної (3D) матриці, pandas не принесе великої користі, і вам доведеться скористатися допомогою інших бібліотек, таких як NumPy[36].

Наостанок хоча фреймворк pandas має декілька недоліків, які на перший погляд здаються досить вагомими і громіздкими, її переваги нівелюють їх, як згадано з порівнянних вище пунктів. Ця бібліотека має достатньо великий потенціал для використання в аналізі даних.

2.9 Порівняння Pandas з бібліотекою NumPy

NumPy — це ще одна потужна програмна бібліотека Python, яка активно використовується протягом останніх кількох років. NumPy — це бібліотека з відкритим кодом, яка має багато учасників. На офіційному сайті згадується, що NumPy є «фундаментальним пакетом для наукових обчислень на Python». Операції над великими багатовимірними масивами та матрицями можна легко виконувати за допомогою NumPy. Крім того, NumPy також надає нам величезну колекцію математичних функцій високого рівня, наприклад, функцію `sin()`, функцію `sort()` тощо для роботи з цими масивами та їхніми елементами. NumPy — це бібліотека Python, яка надає різні похідні об'єкти (наприклад, замасковані масиви та матриці), а також набір процедур для швидшої роботи з масивами. Сюди входять математичні, логічні, маніпуляції формою, сортування, вибір, введення/виведення, дискретне перетворення Фур'є, базова лінійна алгебра, основні статистичні операції, випадкове моделювання та багато інших подібних операцій.

NumPy дуже швидко перетворився на пакет Python, який може дуже ефективно обробляти колосальні обсяги даних разом із підтримкою множення матриць і зміни форми даних. NumPy добре підтримує об'єктно-орієнтований підхід, використовуючи `ndarray`. Іншими словами, `ndarray` — це клас, який складається з багатьох методів і атрибутів. Більшість його методів віддзеркалюються функціями в зовнішньому просторі імен NumPy. Це дозволяє програмісту кодувати в обраній парадигмі. Ця гнучкість дозволила діалекту масиву NumPy і класу NumPy `ndarray` стати де-факто мовою обміну багатовимірними даними, що використовується в Python[37].

Тепер, коли ми чітко розуміємо, що таке Pandas і NumPy, давайте подивимося на основні відмінності між NumPy і Pandas:

Таблиця 2.2 – Порівняння Pandas та NumPy

Параметр порівняння	Pandas	NumPy
---------------------	--------	-------

Основна мета використання	для завдань аналізу даних у Python	для роботи з числовими значеннями, оскільки він полегшує застосування математичних функцій.
Сумісність даних	добре працює для числових, алфавітних і різнорідних типів даних одночасно.	краще працює лише з числовими даними, має ефективне сховище та швидко виконує математичні операції над числовими значеннями на основі масивів і матриць на основі масивів.

Продовження таблиці 2.2

Продуктивність	Якщо кількість рядків у наборі даних перевищує п'ятсот тисяч (500 Кб), то продуктивність Pandas краща, ніж NumPy.	NumPy є швидшим у продуктивності, ніж Pandas, до п'ятдесяти тисяч (50K) рядків і менше набору даних.
Інструменти	DataFrames і Series є найпотужнішими інструментами для Pandas.	Масиви є найпотужнішим інструментом NumPy.
Використання пам'яті	Pandas споживає більше пам'яті порівняно з NumPy.	NumPy споживає менше пам'яті порівняно з Panda.
Об'єкти	DataFrames — це двовимірні (2d) об'єкти, надані Pandas.	NumPy надає n-вимірні масиви, тип даних (dtype) такі ж як об'єкти.

Індексція	Індексування серій pandas значно повільніше, ніж індексування масивів NumPy.	Індексування масивів NumPy відбувається набагато швидше, ніж індексування масивів Pandas.
Використання або застосування в організаціях	Pandas використовується в багатьох популярних організаціях, таких як Trivago, Kaidee, Abeja Inc. та багатьох інших	Instacart, SendGrid, Walmart, Tokopedia та багато інших організацій використовують NumPy.

Продовження таблиці 2.2

Продуктивність	Якщо кількість рядків у наборі даних перевищує п'ятсот тисяч (500 Кб), то продуктивність Pandas краща, ніж NumPy.	NumPy є швидшим у продуктивності, ніж Pandas, до п'ятдесяти тисяч (50K) рядків і менше набору даних.
----------------	---	--

Отже, на завершення можна сказати, що незважаючи на те, що Pandas побудовано на основі NumPy, обидві бібліотеки Python мають значні відмінності. І Pandas, і NumPy спрощують множення матриць і тому активно використовуються в галузі Data Science, особливо в розробці моделей у машинному навчанні. Ці бібліотеки не лише відкривають можливість влаштуватися на роботу в найбільших компаніях світу, але й допомагає стати хорошими експертами з машинного навчання та data science.

2.10. Опис технологій, які були обрані для розробки веб додатку

Для написання веб додатку, було обрано використовувати мову програмування Python для написання Backend частини та JavaScript для відображення ілюстрацій користув на стороні Frontend.

Під час розробки був використаний Framework - Django, який дозволив швидко підняти серверну частину сайту та сконцентрувати увагу на математичній обробці даних. Для обробки даних також був використана бібліотека Pandas, яка дає гнучкі можливості по роботі з великими об'ємами даних та має частково готові часткові рішення. Для частотного аналізу даних була використана бібліотека Scipy.

На стороні Frontend для графічного відображення блоків був використаний Bootstrap та бібліотека Plotly для побудови графіків. Для мережевих запитів на Backend з подальшою обробкою був використаний jQuery.

Висновки по розділу 2

У даному розділі наведені теоретичний опис мови програмування Python. Оглянуті сфери застосування мови програмування, Оглянуті основні недоліки та переваги використання Python.

Оглянуто web framework Django, який написаний з використанням мови програмування Python. Наведені основні можливості даного веб фреймворка, та паттерн за яким потрібно використовувати Django. Проведений порівняльний аналіз Django з інші веб фреймворками на основі Django.

Проведено огляд бібліотеки Pandas. Наведені плюси та мінуси даної бібліотеки. Показані переваги у застосуванні. Проведений порівняльний огляд бібліотеки Pandas з NumPy.

Додатково описані основні мови програмування, фреймворки та бібліотеки, які використовувалися для розробки веб додатку.

РОЗДІЛ 3

Автоматизована система по обробленню сигналу з подальшим спектральним оцінюванням

Веб додаток написаний декількома мовами програмування, а саме Python та JavaScript. Сам проект складається з 31 файлу, з який 7 файлів(рис. 3.1) з розширенням .html, які містять верстку сайта та код, який відпрацьовує на стороні клієнта. 3 файли для конфігурації системи, а саме requirements.txt, Procfile, .gitignore. Та 21 файл(рис. 3.2), які мають розширення .py, які містять серверну логіку та основну частину по математичній обробці сигналу.

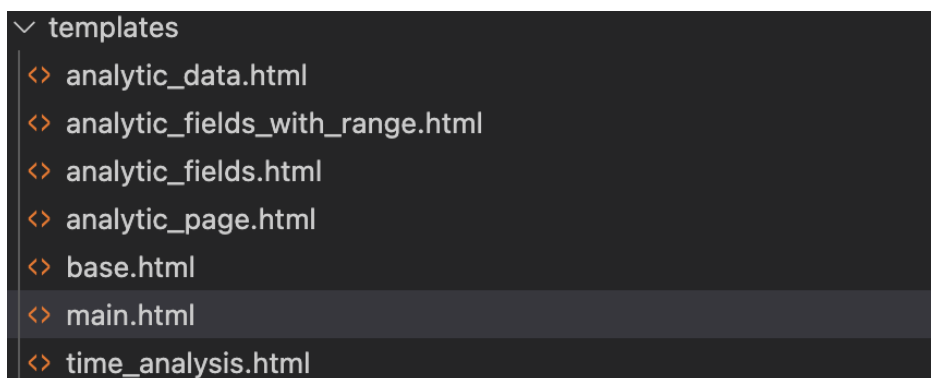


Рисунок 3.1. Файли з розширенням .html

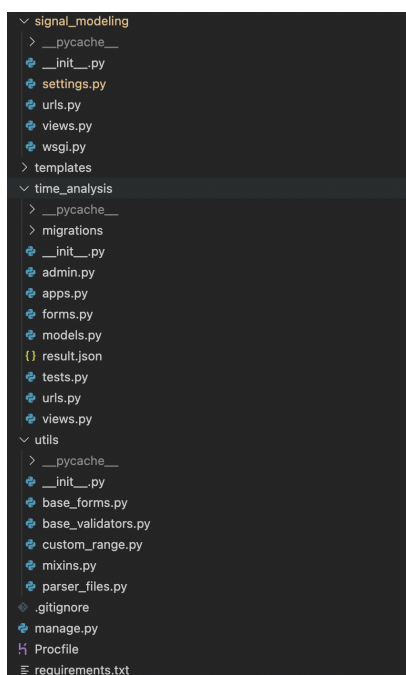


Рисунок 3.2. Файли з розширенням .py

Для того щоб запустити програму локально, потрібно зробити наступні кроки:

1. Встановити Python версії 3.9.10 або вище.
2. Встановити усі бібліотеки, які потрібні для роботи програми. Усі бібліотеки знаходяться у файлі requirements.txt. Які можна встановити за допомогою команди:

```
python -m pip install -r requirements.txt
```

3. Після встановлення необхідних бібліотек, потрібно запустити локальний сервер за допомогою команди:

```
python manage.py runserver
```

4. У браузері відкрити сайт за посиланням, яке напише сервер. Приклад на рисунку 3.3.

```
November 19, 2022 - 14:14:56
Django version 4.0.6, using settings 'signal_modeling.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Рисунок 3.3. Посилання на веб додаток

5. Наступним кроком, буде безпосередня взаємодія з сайтом(рис. 3.4).

Автоматизована система обробки сигналів

Шановний користувач, Ви знаходитесь на початковому екрані Автоматизованої системи

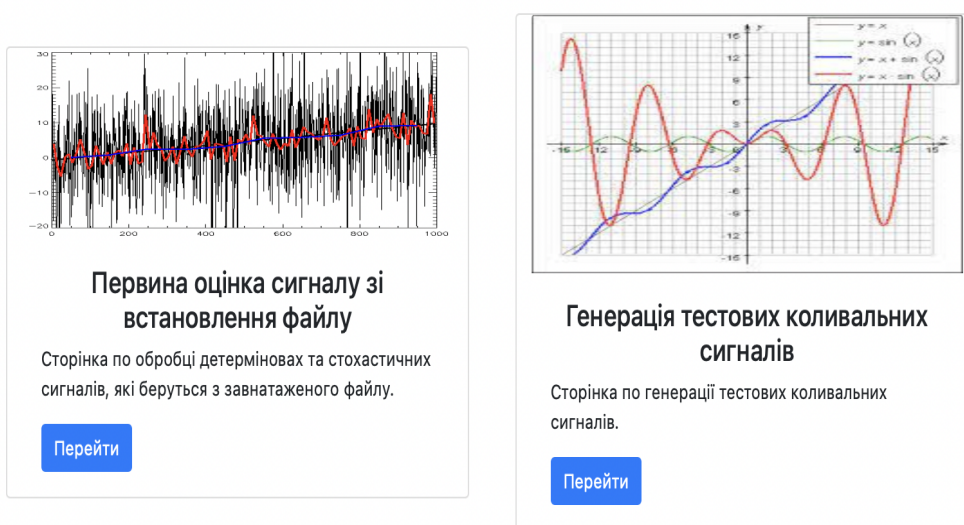


Рисунок 3.4. Головна сторінка сайту

Розглянемо більш детально програмну частину веб додатку, для більш зручного розуміння проекту розділимо огляд на 3 основні частини:

1. Код для налаштування роботи серверної частини проекту;
2. Аналітичний код по обробці вхідних даних;
3. Код для графічного відображення даних;

3.1. Код для налаштування роботи серверної частини проекту

У файлі `requirements.txt` знаходяться версіоновані бібліотеки, які потребує додаток, а саме:

- `acoustics==0.2.6`
- `asgiref==3.5.2`
- `contourpy==1.0.6`
- `cycler==0.11.0`
- `Django==4.0.6`
- `django-crispy-forms==1.14.0`
- `django-multiselectfield==0.1.12`
- `fonttools==4.38.0`
- `kiwisolver==1.4.4`
- `matplotlib==3.6.2`
- `numpy==1.23.1`
- `packaging==21.3`
- `pandas==1.4.3`
- `Pillow==9.3.0`
- `pyarsing==3.0.9`
- `python-dateutil==2.8.2`
- `pytz==2022.6`
- `scipy==1.9.0`
- `six==1.16.0`
- `sqlparse==0.4.3`
- `tabulate==0.9.0`

Файл `Procfile` містить шлях до конфігурації `gunicorn` для даного проекту. Ці налаштування необхідні при деплої коду на сервер і автоматичного оновлення кода, код:

web: gunicorn signal_modeling.wsgi

Файл `manage.py` є головним файлом для роботи з Django. Через цей файл можливо у ручному режимі налаштовувати та використовувати Django через термінал, код знаходиться у додатку.

У папці `signal_modeling` знаходяться основні файли, які містять конфігурацію, налаштування та регулювання шляхів до сторінок усього застосунка, а саме:

1. Файл `settings.py` містить константи, які виконують роль налаштування сервісної частини проекту. Розберемо детальніше кожен константу:
 - 1.1. `BASE_DIR` - шлях до головної папки в якій знаходиться проект
 - 1.2. `SECRET_KEY` - ключ, який необхідний для більш захищеного хешування даних
 - 1.3. `DEBUG` - флаг, який вказує у якому стані запущений проект, у стані розробки чи проект знаходиться на продакшене
 - 1.4. `ALLOWED_HOSTS` - константа, в якій зберігаються хости від яких Django може приймати запити
 - 1.5. `INSTALLED_APPS` - шлях до згрупованих кусків коду, які необхідні для проекту
 - 1.6. `CRISPY_TEMPLATE_PACK` - визначає який фреймворк використовувати для відображення форм
 - 1.7. `MIDDLEWARE` - шлях до хуків в Django для запити та відповідей
 - 1.8. `ROOT_URLCONF` - шлях до головного файлу з посиланнями на сторінки
 - 1.9. `TEMPLATES` - конфігурація темплейтів.

Код файлу у додатку.

3. `wsgi.py` - файл налаштування `gunicorn`:

```
import os
from django.core.wsgi import get_wsgi_application
os.environ.setdefault('DJANGO_SETTINGS_MODULE',
'signal_modeling.settings')
```

```
application = get_wsgi_application()
```

4. `views.py` - файл, який містить код до головної сторінки:

```
from django.views.generic.base import TemplateView
class MainPage(TemplateView):
    template_name = 'main.html'
```

У папці `time_analysis` знаходяться файли, які відповідають за обробку вхідних та вихідних даних. Розглянемо кожен файл окремо.

Файл `__init__.py` необхідний, щоб Python розглядав каталоги як пакети; це робиться для того, щоб запобігти каталогам із загальним ім'ям, наприклад `string`, від ненавмисного приховування допустимих модулів, які відбуваються пізніше (глибше) на шляху пошуку модуля. Сам файл пустий та не містить коду.

Файл `apps.py` зберігає об'єкти конфігурації програми та метадані програми, які належать тільки до папки в якій знаходяться(рис. 3.5.)

```

apps.py ×
time_analysis > apps.py > ...
1  from django.apps import AppConfig
2
3
4  class TimeAnalysisConfig(AppConfig):
5      name = 'time_analysis'
6

```

Рисунок 3.5. Код у файлі apps.py

Файл forms.py містить конфігурації форм, які бачить клієнт на сайті та пост обробка даних які ввів користувач. На рисунку 3.6. - приклад форми яку бачить користувач, на рисунку 3.7. - приклад конфігурації самої форми.

Type of signal
sin

SNR
1

Шуми
Nothing selected

Середнє значення
3

Розмах
4

☒ Обраховувати через кількість періодів

Кількість періодів
2

Кількість точок
22

Частота дискретизації
500

Період дискретизації
0.002

Частота
99

рисунку 3.7. - Приклад форми яку бачить користувач

```

class CustomAnalyticForm(AnalyticBaseForm):
    class SignalType:
        COS = 'cos'
        SIN = 'sin'

    SIGNAL_TYPE = (
        (SignalType.COS, 'cos'),
        (SignalType.SIN, 'sin')
    )

    class Noises:
        WHITE = 'white'
        PINK = 'pink'
        BLUE = 'blue'
        BROWN = 'brown'
        VIOLET = 'violet'

    NOISES = [
        (Noises.WHITE, 'Білий шум'),
        (Noises.PINK, 'Розовий шум'),
        (Noises.BLUE, 'Синій шум'),
        (Noises.BROWN, 'Коричневий шум'),
        (Noises.VIOLET, 'Фіолетовий шум'),
    ]

    noises = MultipleChoiceField(label='Шум', choices=NOISES, required=False)
    snr = FloatField(label='SNR', min_value=0.00001, required=False)
    type_of_signal = ChoiceField(required=False, choices=SIGNAL_TYPE)
    mean = FloatField(label='Середнє значення', required=True)
    scope = FloatField(label='Розмах', required=True)
    checker_count_of_dot_or_period_sampling = BooleanField(label='Обраховувати через кількість періодів', required=False)
    count_of_periods = IntegerField(label='Кількість періодів', required=False)
    count_of_dots = IntegerField(label='Кількість точок', required=False)
    period_sampling = FloatField(label='Період дискретизації', required=True)
    frequency_sampling = FloatField(label='Частота дискретизації', required=True)
    frequency = IntegerField(label='Частота', required=True)

```

рисунку 3.7. Приклад конфігурації форми.

Увесь код, який міститься у файлі forms.py можна знайти у додатку.

Файл result.json містить дефолтні значення для генерації сигнали на сторінці зі штучною генерацією сигналу.

Файл urls.py містить шлях до сторінок з сторінками по обробці сигналу з файлу та з генерації штучного сигналу, код файлу:

```

from django.urls import path

from time_analysis.views import CustomAnalyticView, TimeAnalysisView

urlpatterns = [
    path('time_analysis', TimeAnalysisView.as_view(), name='time-analysis'),
    path('custom_time_analysis', CustomAnalyticView.as_view(),
name='custom-time-analysis'),
]

```

Файл views.py містить класи по обробці вхідних та вихідних даних, код програми можна знайти у додатку.

3.2. Аналітичний код по обробці даних

У папці utils знаходиться основна логіка по проведенню швидкого перетворення для спектрального аналізу. В цій папці знаходиться логіка по роботі з файлами, математичні розрахунки які пов'язані з вхідним сигналом, а також методи які часто використовуються у інших місцях проекту. Розглянемо кожен файл окремо.

Файл __init__.py пустий, необхідний щоб Python розумів, що з цими файлами необхідно працювати у даному проекті.

Файл `base_forms.py` має основний аналітичний код, який оброблює вхідний сигнал для подальшого графічного відображення. У самому файлі проводиться частотний аналіз, аналіз за допомогою різних видів періодограм та математичні розрахунки, а саме: максимальне та мінімальне значення, дисперсія, середнє квадратичне відхилення сигналу, дисперсія, розмах та математичне сподівання. Увесь код файлу можна знайти у додатку

Файл `custom_range.py` містить код генератора, який буде рівновідалені точки між собою, код файлу можна знайти у додатку.

Файл `mixins.py` містить міксін, який дозволяє експортувати дані з проекту, код файлу можна знайти у додатку.

Файл `parser_files.py` містить два класи, один з яких перетворює різні типи файлів у зручний для роботи формат, а саме `DataFrame`. Та другий клас, який перетворює вхідний `DataFrame` у файл типу ІО, код файлу можна знайти у додатку.

3.3. Код для графічного відображення даних

Файли, які потрібні для графічного відображення проекту, знаходяться у папці `templates`. Розглянемо кожний файл окремо.

Файл `base.html` містить основні файли, які імпортуються на сторінки проекту, код файлу можна знайти у додатку.

Файл `main.html` відповідає за відображення головної сторінки проекту, код файлу можна знайти у додатку.

Файл `analytic_page.html` містить верстку форм та графічного відображення даних у вигляді таблиць та графіків. Код файлу можна знайти у додатку.

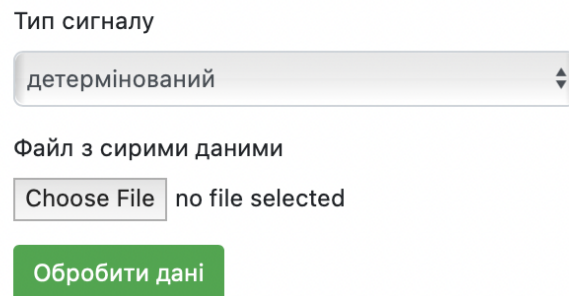
Файл `time_analysis.html` імпортує код з файлу `analytic_page.html`, код файлу:

Файл `analytic_fields.html` будує форму через яку можна імпортувати файл (рис. 3.8), код програми:

```
{% load crispy_forms %}
```

```
{% csrf_token %}
{% for field in form %}
  <div class="row">
    <div class="col-12">
      {{ field|as_crispy_field }}
    </div>
  </div>
{% endfor %}

<button type="submit" class="btn btn-success">{{ button_name }}</button>
```



Тип сигналу

детермінований

Файл з сирими даними

Choose File no file selected

Обробити дані

Рисунок 3.8. Форма для імпортування файлу

Файл `analytic_fields_with_range.html` будує форму через яку можна створити штучний сигнал (рис. 3.9), код файлу можна знайти у додатку.

Type of signal

sin

SNR 1

Шуми

Nothing selected

Середнє значення 3

Розмах 4

☒ Обрахувати через кількість періодів

Кількість періодів 2

200

2 52 101 151 200

Кількість точок 22

32 768

2 8 194 16 385 24 577 32 768

Частота дискретизації 500

10 000

0 2 500 5 000 7 500 10 000

Період дискретизації 0.002

1

0 0.25 0.5 0.75 1

Частота 99

400

0 100 200 300 400

Рисунок 3.9. Форма для кастомного генерації сигнала

Файл `analytic_data.html` відповідає за відображення графіків та таблиць на сторінках(рис. 3.10), код файлу можна знайти у додатку.

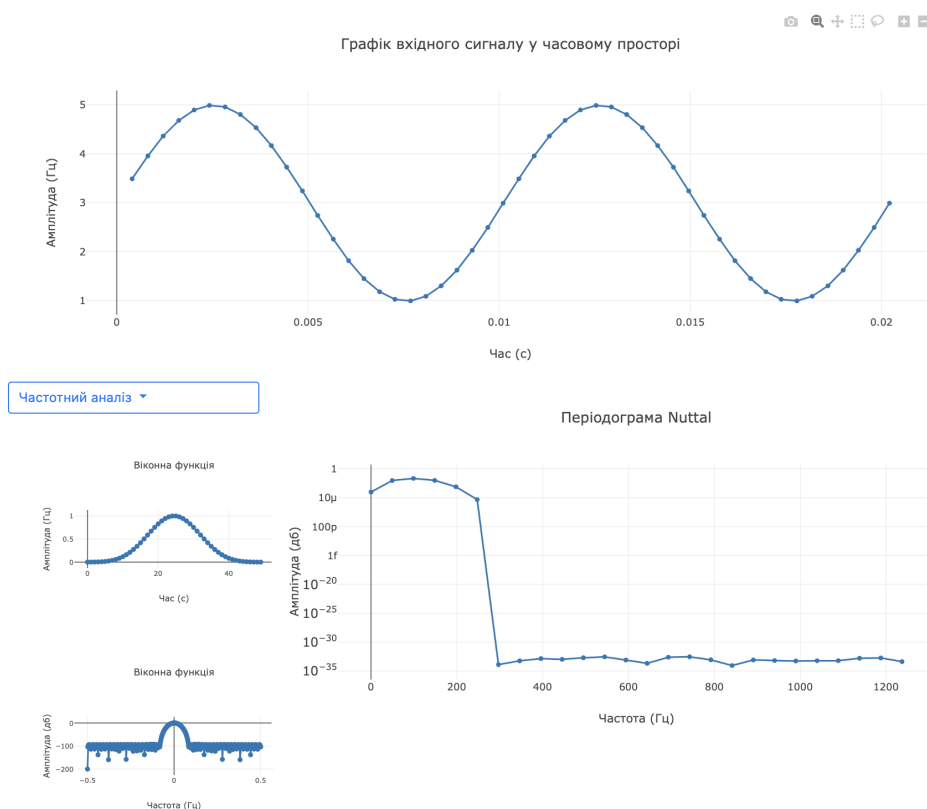


Рисунок 3.10. Графічне відображення даних

3.4. Порівняння з іншими система по обробці сигналів

Для побудову такого роду автоматизованих систем наразі не існує рішень. Оператору доведеться писати власноруч написаний код для його аналізу та обробки. Тож для порівняння візьмемо пакетне рішення Matlab, воно найбільше підходить для вирішення даних проблем.

Порівняння двох систем буде зроблена за наступними параметрами:

1. Зручність;
2. Стабільність;
3. Надійність;
4. Швидкодія.

3.4.1. Зручність

Наразі, той функціонал, який пропонує даний застосунок є повністю безкоштовним та показує увесь спектр для первинної оцінки сигналу у частотній площині. Система відразу пропонує рішення, які можуть дати обґрунтовану

оцінку сигналу. Крім цього, дані з таблиць користувач, може завантажити у довільному форматі, а графіки завантажуються у форматі .png(рис 3.11) . Що дає можливість користувачу у подальшому використовувати дані у своїх матеріалах.

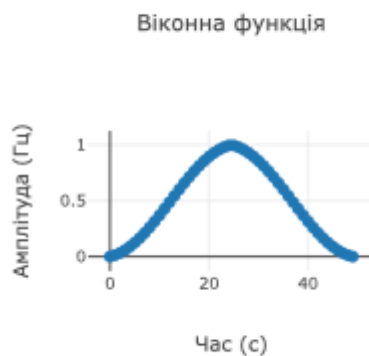


Рисунок 3.11. Приклад завантаженої віконної функції

У свою чергу матлаб не пропонує готових рішень, а потребує від оператора навичок та знань написання коду, розуміння фізичних процесів сигналу та розуміння які методи та функції потрібно використовувати для проведення операцій над даними.

3.4.2. Стабільність

Система працює стабільно та підтримується на різних пристроях: ПК, телефони, планшети. Застосунок не потребує ресурсів пристрою через який використовується, що дає проводити аналіз на великих даних, незважаючи на ресурси, які є у пристроя з якого вводяться дані. Matlab у свою чергу потребує ресурси пристрою в якому запускається та має робочі версії лише на персональному комп'ютері.

Веб додаток працює цілодобово та відразу може використовуватися багатьма користувачами. Має доступ для усього світу. Matlab можна використовувати лише за допомогою маніпуляцій з локальними мережами, що може зайняти більше часу ніж сам огляд сигналу. Matlab залежить від кількості

пам'яті яку може дати комп'ютер, та не завжди адекватно працює з великою кількістю даних.

3.4.3. Надійність

Веб додаток працює з користувачами по SSL з'єднанню що гарантує захищеність даних. Сам застосунок на своєму боці не зберігає ніякі дані по користувачу та його даним, які він вводить. Дані з якими проводиться математична обробка після відправки на сторону клієнта видаляються та не зберігаються у виглядів хешів. Matlab має резервне копіювання даних після виконання операцій та не гарантує безпеку ваших даних, якщо шахрай захоче вкрати дані користувача.

3.4.4. Швидкодія

Для оцінки швидкодії використаємо сторінку генерації штучного сигналу. Значення вхідних параметрів, які будуть змінюватися: тип сигналу(синус, косинус, наявність шумів в сигналі), кількість періодів, кількість точок у сигналі та частота.

1. Зробимо заміри по типу сигналу. Дані, які беруться за константу:

Частота - 100, Кількість періодів - 2, Середнє значення - 3, Розмах - 4

Тип сигналу	Веб додаток	Matlab
sin	0.05с	7с
cos	0.068с	7с
sin та білий шум	0.079с	7с
sin та фіолетовий шум	0.081с	7с
sin, рожевий та коричневий шум	0.085с	7с

Таблиця 3.1. Аналіз швидкодії з різними типами сигналів

2. Зробимо заміри за кількістю періодів у сигналі. Дані, які беруться за константу: частота - 100, тип сигналу - $\sin$, середнє значення - 3, розмах - 4

Кількість періодів	Веб додаток	Matlab
1	0.038с	7с
10	0.047с	7с
50	0.12с	7с
100	0.2с	7с
200	0.373с	7с

Таблиця 3.2. Аналіз швидкодії з різною кількістю періодів

3. Зробимо заміри за кількістю точок у сигналі. Дані, які беруться за константу: частота - 100, тип сигналу - $\sin$, середнє значення - 3, розмах - 4

Кількість точок у сигналі	Веб додаток	Matlab
128	0.072с	7с
256	0.077с	7с
512	0.052с	8с
1024	0.121с	7с
2048	0.17с	8с
4096	0.31с	8с
8192	0.552с	8с
16384	0.976с	9с

Таблиця 3.3. Аналіз швидкодії з різною кількістю точок у сигналі

4. Зробимо заміри за розміром частоти у сигналі. Дані, які беруться за константу: тип сигналу - $\sin$, середнє значення - 3, розмах - 4, кількість періодів - 4

Кількість частот	Веб додаток	Matlab
1	0.064с	13с
10	0.031с	7с
100	0.065с	6с
200	0.066с	6с
250	0.066с	7с
300	0.071с	8с
350	0.066с	7с
400	0.003с	7с

Таблиця 3.4. Аналіз швидкодії з різним значенням частоти у сигналі

3.5. Приклад роботи з веб додатком

Для роботи з веб додатком, потрібно мати файл у якому будуть дані сигналу по двом величинам, а саме: час та амплітуда. Для тестового використання змодельємо штучний сигнал за допомогою скрипта:

```
import numpy as np
```

```
f=99
```

```
T=1/f
```

```
fd=25*f
```

```
p=2
```

```
Td=1/fd
```

```
t = np.arange(Td,p*T,Td)
```

```
y = 3+2*np.sin(2*np.pi*t*f)
```

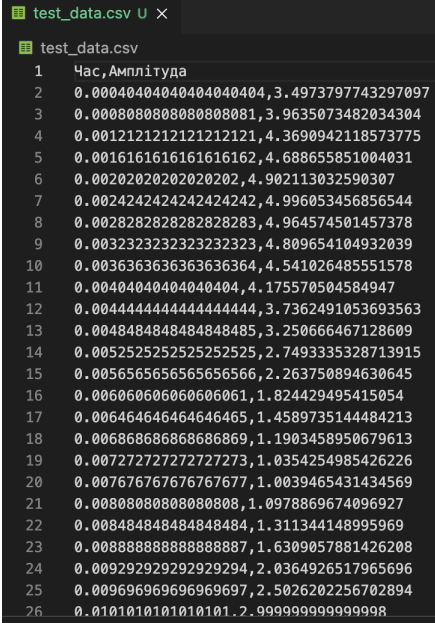
У даному прикладі, ми змодельювали синусоїду з частотою, яка дорівнює 99, кількість періодів - 2, з середнім значенням - 3 та розмах, який дорівнює двом.

Веб додаток може приймати на імпорт три типи файлів, тому розглянемо кожний з них:

1. .csv, для того щоб зберегти дані у файлі з розширенням .csv додамо до скрипта наступний код:

```
import pandas as pd
df = pd.DataFrame({'Час': t, 'Амплітуда': y})
df.to_csv('test_data.csv', index=False)
```

Дані які повинні отримати зображені на рисунку 3.12.



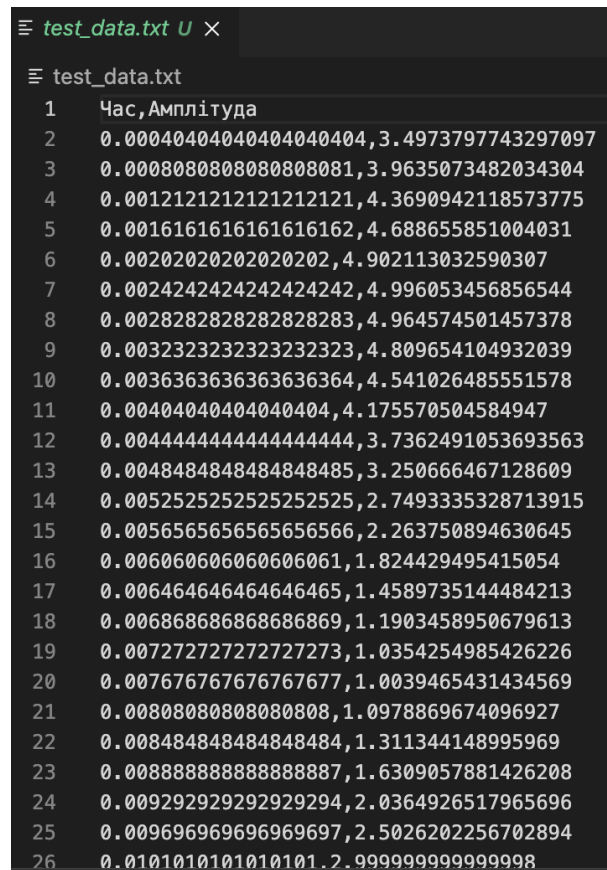
	Час, Амплітуда
1	0.000404040404040404, 3.4973797743297097
2	0.0008080808080808081, 3.9635073482034304
3	0.0012121212121212121, 4.3690942118573775
4	0.0016161616161616162, 4.688655851004031
5	0.0020202020202020202, 4.902113032590307
6	0.0024242424242424242, 4.996053456856544
7	0.0028282828282828283, 4.964574501457378
8	0.0032323232323232323, 4.809654104932039
9	0.0036363636363636364, 4.54102648551578
10	0.0040404040404040404, 4.175570504584947
11	0.0044444444444444444, 3.7362491053693563
12	0.0048484848484848485, 3.250666467128609
13	0.0052525252525252525, 2.7493335328713915
14	0.0056565656565656566, 2.263750894630645
15	0.0060606060606060606, 1.824429495415054
16	0.0064646464646464646, 1.4589735144484213
17	0.0068686868686868689, 1.1903458950679613
18	0.0072727272727272727, 1.0354254985426226
19	0.007676767676767677, 1.0039465431434569
20	0.0080808080808080808, 1.0978869674096927
21	0.0084848484848484848, 1.311344148995969
22	0.0088888888888888887, 1.6309057881426208
23	0.0092929292929292929, 2.0364926517965696
24	0.009696969696969697, 2.5026202256702894
25	0.010101010101010101, 2.9999999999999998
26	

Рисунок 3.12. Приклад даних, які повинні зберегтися у файлі з розширенням .csv

2. .txt, для того щоб зберегти дані у файлі з розширенням .txt додамо до скрипта наступний код:

```
import pandas as pd
df = pd.DataFrame({'Час': t, 'Амплітуда': y})
df.to_csv('test_data.txt', header=['Час', 'Амплітуда'], index=None, sep=',')
```

Дані які повинні отримати зображені на рисунку 3.13.



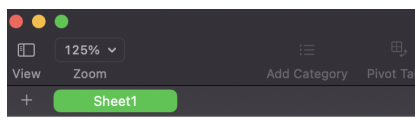
	Час, Амплітуда
2	0.000404040404040404, 3.4973797743297097
3	0.000808080808080808, 3.9635073482034304
4	0.001212121212121212, 4.3690942118573775
5	0.001616161616161616, 4.688655851004031
6	0.002020202020202020, 4.902113032590307
7	0.002424242424242424, 4.996053456856544
8	0.002828282828282828, 4.964574501457378
9	0.003232323232323232, 4.809654104932039
10	0.003636363636363636, 4.541026485551578
11	0.004040404040404040, 4.175570504584947
12	0.004444444444444444, 3.7362491053693563
13	0.004848484848484848, 3.250666467128609
14	0.005252525252525252, 2.7493335328713915
15	0.005656565656565656, 2.263750894630645
16	0.006060606060606061, 1.824429495415054
17	0.006464646464646465, 1.4589735144484213
18	0.006868686868686869, 1.1903458950679613
19	0.007272727272727273, 1.0354254985426226
20	0.007676767676767677, 1.0039465431434569
21	0.008080808080808080, 1.0978869674096927
22	0.008484848484848484, 1.311344148995969
23	0.008888888888888887, 1.6309057881426208
24	0.009292929292929294, 2.0364926517965696
25	0.009696969696969697, 2.5026202256702894
26	0.0101010101010101, 2.999999999999998

Рисунок 3.13. Приклад даних, які повинні зберегтися у файлі з розширенням .txt

3. .xls/.xlsx, для того щоб зберегти дані у файлі з розширенням .xls/.xlsx додамо до скрипта наступний код:

```
import pandas as pd
df = pd.DataFrame({'Час': t, 'Амплітуда': y})
df.to_excel("test_data.xlsx")
```

Дані які повинні отримати зображені на рисунку 3.14.



	Час	Амплітуда
0	0,0004040	3,4973797
1	0,0008080	3,9635073
2	0,0012121	4,3690942
3	0,0016161	4,6886558
4	0,0020202	4,9021130
5	0,0024242	4,9960534
6	0,0028282	4,9645745
7	0,0032323	4,8096541
8	0,0036363	4,5410264
9	0,0040404	4,1755705
10	0,0044444	3,7362491
11	0,0048484	3,2506664
12	0,0052525	2,7493335
13	0,0056565	2,2637508
14	0,0060606	1,8244294
15	0,0064646	1,4589735
16	0,0068686	1,1903458
17	0,0072727	1,0354254
18	0,0076767	1,0039465
19	0,0080808	1,0978869
20	0,0084848	1,3113441
21	0,0088888	1,6309057
22	0,0092929	2,0364926
23	0,0096969	2,5026202
24	0,0101010	3
25	0,0105050	3,4973797
26	0,0109090	3,9635073
27	0,0113131	4,3690942
28	0,0117171	4,6886558
29	0,0121212	4,9021130
30	0,0125252	4,9960534
31	0,0129292	4,9645745
32	0,0133333	4,8096541
33	0,0137373	4,5410264
34	0,0141414	4,1755705
35	0,0145454	3,7362491
36	0,0149494	3,2506664
37	0,0153535	2,7493335

Рисунок 3.14. Приклад даних, які повинні зберегтися у файлі з розширенням .xls/.xlsx

Отримані тестові дані можна використовувати для подальшого використання у веб додатку. Розглянемо покроково як отримати дані для аналізу:

1. Переходимо у веб додаток та потрапляємо на головний екран (рис. 3.15)

Автоматизована система обробки сигналів

Шановний користувач, Ви знаходитесь на початковому екрані Автоматизованої системи

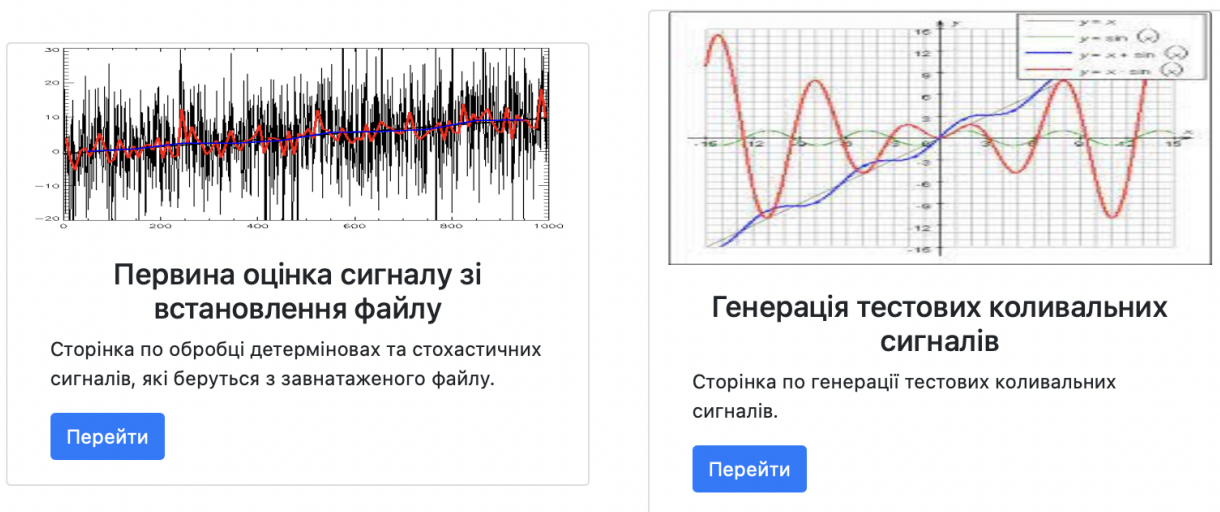


Рисунок 3.15. Головний екран веб додатку

2. Переходимо у вкладку “Первинна оцінка сигналу зі встановлення файлу”.
3. З лівого боку екрану можна побачити форму, у яку можна імпортувати файл(рис. 3.16)

[<- Головна сторінка](#)

Тип сигналу

детермінований

Файл з сирими даними

Choose File

no file selected

Обробити дані

Рисунок 3.16. Форма для створення частотного аналізу по файлу, який імпортується

4. У даному прикладі розглядаємо детермінований тип сигналу.
5. Натискаємо “Choose File” для того щоб обрати файл, який згенерували за допомогою скрипта вище(рис. 3.17).

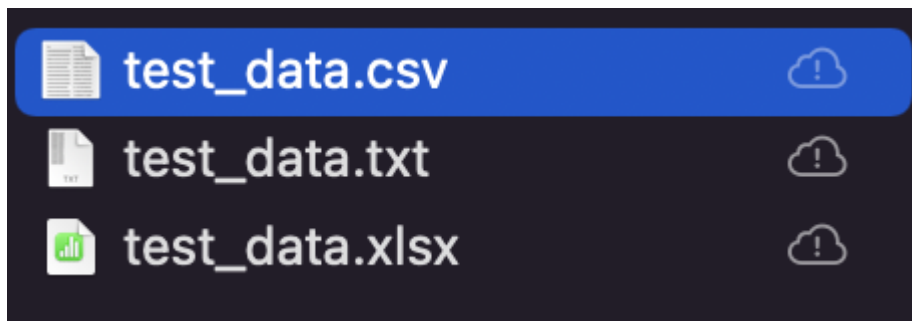


Рисунок 3.17. Файл з сигналом

6. Натискаємо кнопку “Обробити дані”.
7. Після натискання кнопки, сайт обробляє вхідний файл та на основі цього файлу будує часовий та частотні графіки(рис. 3.18).

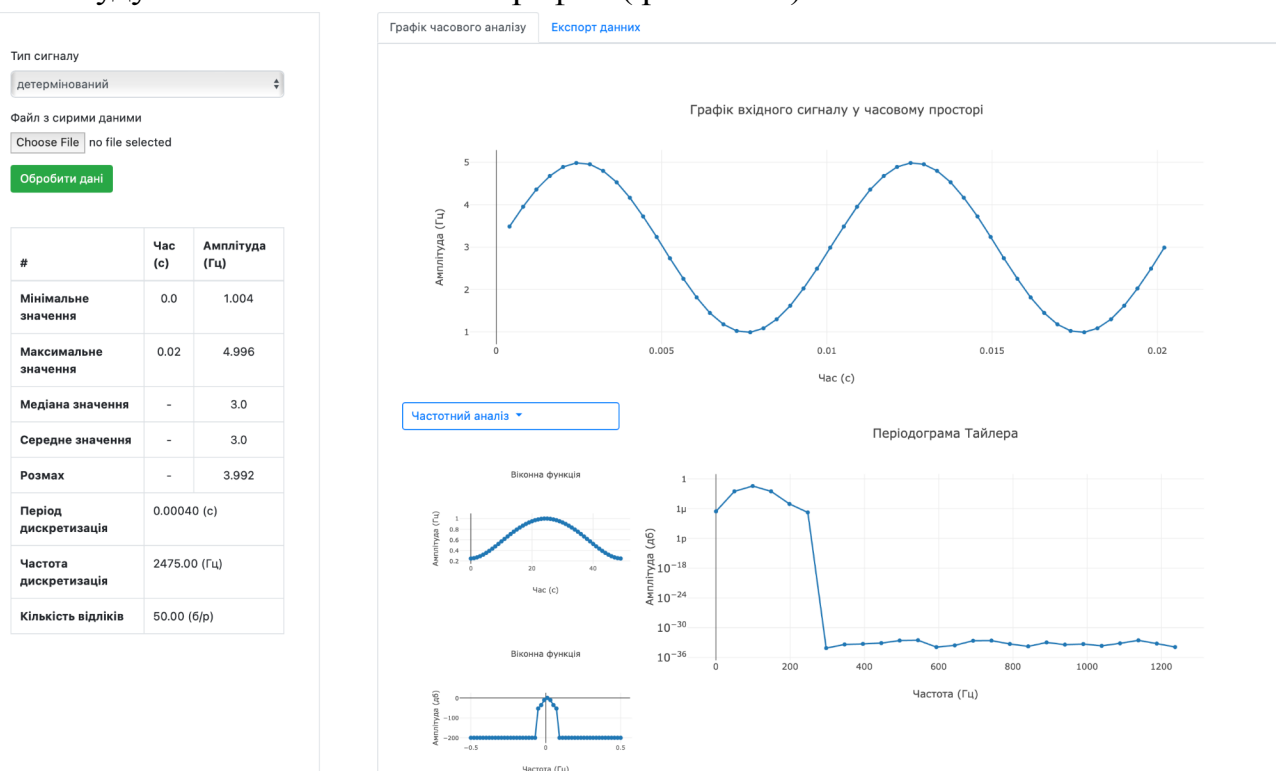


Рисунок 3.18. Згенероване графічне відображення файлу у вигляді таблиці та графіків

Для побудови стохастичного сигналу, використаємо скрипт, який ми використовували вище та додамо шумову складову, скрипт:

```
import numpy as np
f=99
T=1/f
fd=25*f
```

p=2

Td=1/fd

t = np.arange(Td,p*T,Td)

y = 3+2*np.sin(2*np.pi*t*f)

Для генерації шумової складової використаємо snr = 13 та шум білого типу, скрипт:

```
from acoustics.generator import noise as noise_generator
```

```
snr = 13
```

```
noise = noise_generator(len(y), "white")
```

```
koef = np.sqrt( np.mean( np.power(y,2) ) / ( np.power(10,(snr/10)) *  
np.mean(np.power(noise,2)) ) )
```

```
noise = noise * koef
```

Також окрім білого шуму у метод noise_generator можна передати інші назви шумів для їх генерації, а саме:

1. white - білий
2. pink - рожевий
3. blue - синій
4. brown - коричневий
5. violet - фіолетовий

Додамо до сигналу шумову складову:

```
y += noise
```

Збережемо сигнал у форматі .txt:

```
import pandas as pd
```

```
df = pd.DataFrame({'Час': t, 'Амплітуда': y})
```

```
df.to_csv('test_data_with_white_noise.txt', header=['Час', 'Амплітуда'], index=None,  
sep=',')
```

На рисунку 3.19 приклад даних, які потрібно отримати.

```

test_data_with_white_noise.txt
1  Час, Амплітуда
2  0.000404040404040404, 2.6032166830204635
3  0.000808080808080808, 2.9632811279835596
4  0.001212121212121212, 6.37989730740056
5  0.001616161616161616, -2.0483801455757717
6  0.00202020202020202, 6.568302539009067
7  0.002424242424242424, 3.923699997430815
8  0.002828282828282828, 2.869541722190318
9  0.003232323232323232, 5.243404393062382
10 0.003636363636363636, 4.077360149899359
11 0.00404040404040404, 8.217942231697897
12 0.004444444444444444, -4.521580549994788
13 0.004848484848484848, -0.9499366315718469
14 0.005252525252525252, 4.762740529399703
15 0.005656565656565656, 2.0992155685251106
16 0.006060606060606061, -0.27797237413168996
17 0.006464646464646465, -5.257154366730239
18 0.006868686868686869, -5.078106865538196
19 0.007272727272727273, 0.42830296684158475
20 0.007676767676767677, -1.158843744107687
21 0.00808080808080808, 9.740029196539314
22 0.008484848484848484, 6.314675194889414
23 0.008888888888888888, 2.383984049548222
24 0.009292929292929294, 0.25103950457674995
25 0.009696969696969697, 5.775371553092837
26 0.0101010101010101, 0.22213274157652707

```

Рисунок 3.19. Дані, які користувач отримує після генерації сигналу з білим шумом

Для обробки сигналу з шумом у формі потрібно обрати тип сигналу - стохастичний(рис. 3.20)

детермінований

✓ стохастичний (випадковий)

Файл з сирими даними

Choose File no file selected

Обробити дані

Рисунок 3.20. Перемикач типу сигналу на стохастичний
Натискаємо на кнопку “Choose File” та обираємо файл з генерованим сигналом та шумовою складовою(рис. 3.21).

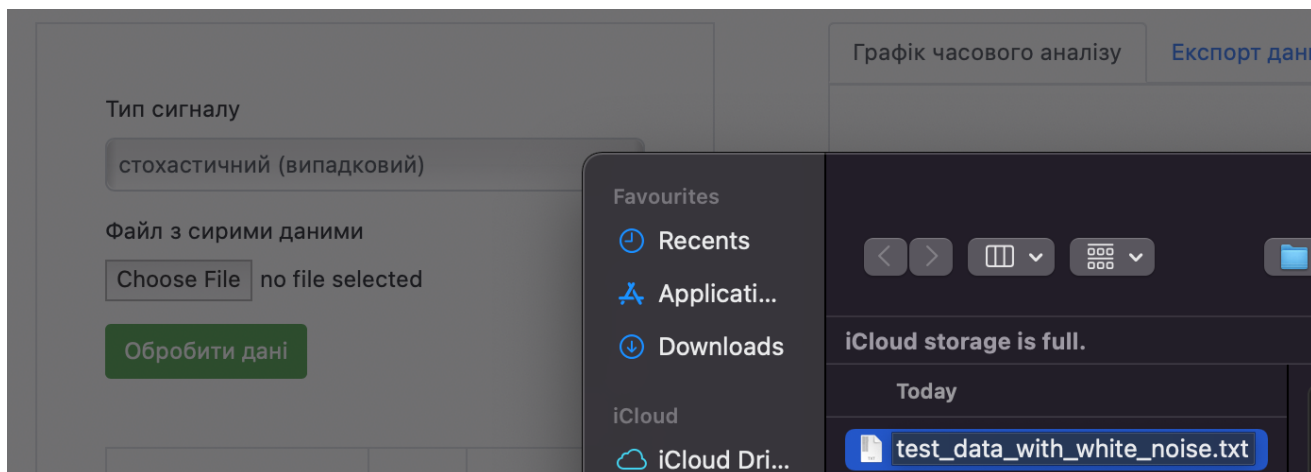


Рисунок 3.21. Обрання файлу з сигналом та шумовою складовою
Після натискання на кнопку “Обробити дані”, веб додаток обробить імпортований файл та покаже результат у вигляді таблиць та графіків (рис. 3.22).



Рисунок 3.22. Згенероване графічне відображення файлу у вигляді таблиці та графіків

Висновок до розділу 3:

У даному розділі був розроблений веб додаток, який автоматизує частотний аналіз вхідних сигналів та має два види прийняття даних: через файл та за допомогою генерації даних через форму.

Також проведено порівняння системи з програмний забезпеченням Matlab за кількома параметрами, а саме: швидкодія, стабільність, зручність та

надійність. Після проведення аналізу порівняння двох систем можна зробити висновок, що веб додаток є набагато швидший, стабільніший, зручний та надійніший за багатьма параметрами у порівнянні з Matlab.

Для використання веб додатку був проведений приклад по генерації детермінованого та стохастичних сигналів та зберігання сигналів у різних форматах файлів(.csv, .txt, .xlsx). Проведений огляд по генерації графічного відображення даних через імпортування файла у додаток.

РОЗДІЛ 4

РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ АВТОМАТИЗОВАНИЙ ВЕБ-ДОДАТОК ДЛЯ ШВИДКОГО ЧАСТОТНОГО АНАЛІЗУ СИГНАЛУ

4.1 Опис ідеї проекту

Через великий об'єм даних у теперішній час постає питання у використанні систем з готовим рішенням для первинної оцінки сигналів. Завдання швидкого аналізу сигналу досить часто постає перед дослідниками та інженерами у різних областях науки та техніки. Область, де можна застосувати досить очевидна, це першочерговий аналіз даних на першому етапі оцінки придатності даних для подальшої обробки та аналізу. Такі задачі загалом постають при виконанні математичних розрахунків, математичного моделювання; при проектуванні комунікаційного обладнання, систем технічного зору, високоякісного звуковідтворювального та медичного обладнання; застосовуються в економіці та соціології, а також у технічній діагностиці та швидкому навчанню операторів.

Цей застосунок дозволяє вченим, оператором та бізнесу на ранніх етапах зрозуміти цілесобразність проекту завдяки швидкому аналізу через цей застосунок. Крім цього початкові оператори можуть використовувати цей застосунок для генерації штучних сигналів з різними складовими для подальшого аналізу та оцінювання.

У даній главі проаналізовано та подано у вигляді таблиць: зміст ідеї; можливі напрямки застосування; основні вигоди, що може отримати користувач товару та чим відрізняється від існуючих аналогів та замінників.

В таблиці 4.1 наведено цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати потрібні групи потенційних клієнтів.

Таблиця 4.1 Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Використання автоматизованої системи частотно аналізу сигналу у різних областях техніки та науки.	1. Будівництво	Попередження зносу конструкції, поломки або її руйнуванню на ранніх етапах. Висока точність отриманих даних
	2. Економіка	Прогнозування зростання чи спадання економічних показників Зменшення витрат на обслуговування Визначення ймовірності процесу Висока точність отриманих даних
	3. Медицина	Первинна оцінка результату лікування. Оцінка тренду процесу. Висока точність отриманих даних
	4. Політика	Політичні прогнози(результати виборів, міжнародні відношення) Визначення ймовірності процесу Висока точність отриманих даних
	5. Військова справа	Прогнозування поведінки супротивника чи потенційного супротивника Визначення ймовірності процесу Висока точність отриманих даних

Продовження таблиці 4.1.

	6. Психологія	Прогнозування стійкості подружніх відносин Визначення ймовірності процесу Висока точність отриманих даних
	7.Ресурсозабезпеченість	Прогнозування споживання електроенергії, води, корисних копалин і т.п. Зменшення витрат на обслуговування Визначення ймовірності процесу Висока точність отриманих даних

Висновок: В більшості технічних систем задача швидкої оцінки сигналу є доволі необхідною, оскільки існуючі рішення не дають такого швидкого результату частотної оцінки сигналу. Тому знаходження нових підходів є актуальним на ринку послуг. Головними напрямками застосування є будівництво, політика, економіка.

Аналіз потенційних техніко-економічних переваг ідеї порівняно із пропозиціями конкурентів наведено в таблиці 4.2.

Таблиця 4.2 Визначення сильних, слабких та нейтральних характеристик

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Excel	Matlab	STAT TA			
1	Низькі витрати	12	20	25	15			+
2	Точність	висока	низька	висока	висока		+	

Продовження таблиці 4.2

3	Швидкість проведення результату	висока	низька	низька	низька			+
4	Гнучкість	+	-	-	-			+
5	Простота	94%	50%	20%	10%			+

Висновки: Як видно з таблиці головною перевагою проекту для користувачів являється простота та гнучкість, яка дозволяє користувачеві швидко навчитися користуватися проектом та почати працювати з ним. Не менш важливим параметром є низкість витрат на користування даним застосунком.

4.2 Технологічний аудит ідеї проекту

Технологічний аудит – це спосіб перевірки технологічного стану підприємства, рівня організації всіх важливих процесів за допомогою встановлених критеріїв, виявлення сильних і слабких сторін виробництва[39].

У таблиці 4.3 проводиться аудит технології, за допомогою якої можна реалізувати ідею проекту. Параметри, які були взяті як ідея проекту:

Таблиця 4.3 Технологічна здійсненність ідеї проекту

1	2	3	4	5
№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Швидкий частотний аналіз сигналу	Використання веб додатку	Наявна	Доступна

Продовження таблиці 4.3

1	2	3	4	5
---	---	---	---	---

2	Штучне моделювання сигналу	Впровадження алгоритмів моделювання сигналу за різними параметрами	Наявна	Доступна
3	Використання застосунку для навчання операторів	Складання навчального процесу за рахунок використання веб додатку	Наявна	Доступна
4	Інтеграція алгоритму в реальні прилади	Створення API для обробки та моделювання сигналу	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: Створення нових алгоритмів на платформі по оптимізації сигналів та їх прогнозування				

Висновки: Проаналізувавши таблицю, можна зробити висновок, що даний проект можна використовувати як у реальних наукових проектах так і у бізнес задач, а також впровадження освітнього процесу використовуючу дану платформу.

4.3 Аналіз ринкових можливостей запуску стартап проекту

Для аналізу ринкових можливостей, які можна використати під час аналізу запуску оцінюються по основним параметрам: кількість гравців, загальний обсяг продажу, динаміка ринку, рентабельність та специфічних умов ліцензування. Ці дані описані у Таблиці 4.4.

Таблиця 4.4 Попередня характеристика потенційного ринку
стартап-проекту

1	2	3
№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	10
2	Загальний обсяг продаж, грн/ум.од	94000
3	Динаміка ринку (якісна оцінка)	зростає

Продовження таблиці 4.4

4	Специфічні вимоги до стандартизації та сертифікації	відсутні
---	---	----------

5	Середня норма рентабельності в галузі (або по ринку), %	64%
---	---	-----

Висновки: головними перевагами показників стану ринку є те, що динаміка стрімко зростає на попит даного продукту, це призведе до підвищення обсягу продаж. Присутні також фактори загрози для виходу на ринок. Такими факторами є висока кількість конкурентів та рівень знань у спеціалістів, що виконують правильне введення даних та реалізацію алгоритму. Середня норма рентабельності на ринку є досить високою – 64%, це означає, що проект рентабельний для входження на ринок.

Для отримання більш чіткої картини ситуації на ринку визначимо потенційні групи клієнтів (табл.4.5), їх характеристики, сформуємо орієнтовний перелік вимог до товару для кожної групи.

Таблиця 4.5 Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Попередження пошкодження та руйнуванню об'єкту під час нормативного строку експлуатації	Будівельні компанії	Особливості, викликані зі специфікою проведення дослідів, необхідність модифікування програмно-алгоритмічного забезпечення, інтеграція з існуючими системами	Швидкість роботи Можливість комплексування з іншими системами Висока точність результатів
2	Прогнозування стану здоров'я людини	Медицина		

Продовження таблиці 4.5

3	Швидкий аналіз економічних показників	Економіка		
---	--	-----------	--	--

Висновки: Проаналізувавши таблицю, виділимо дві головні потреби, що формує ринок. Перша це: швидкий аналіз даних, друге - навчання молодих операторі.

Проведемо аналіз факторів ринкового середовища що сприяють ринковому впровадженню (табл. 4.6.) проекта, та факторів, що йому перешкоджають (табл. 4.7.). Результати наведено у таблицях 4.6-4.7.

Таблиця 4.6 Фактори можливостей

№ п/ п	Фактор	Зміст можливості	Можлива реакція компанії
1	Збільшення попиту	Різке збільшення зацікавленості до системи	Збільшення функціоналу застосунка
2	Необхідність до інтеграції	Відкрити API	Дати можливість компаніям користуватись відкритим API
3	Навчання	Навчання молодих спеціалістів первинної оцінки сигналу	Створення навчального процесу для підвищення кваліфікації спеціалістів
4	Індивідуальне замовлення	Клієнт потребує додання нових специфічних додатків	Валідація ідей клієнтів для додавання нового функціоналу на сайт
5	Інтеграції	Давати можливість клієнтам використовувати застосунок для своїх систем	Оцінки можливих ризиків

Висновки: Такі системи використовуються у нових та прогресивних компаніях та стартапах починаючи від навчання персоналу та повноцінної оцінки

сигналу за допомогою графічного інтерфейсу сайту . Це досягається шляхом реклами та введення нових можливостей на веб додаток.

Таблиця 4.7 Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Ринок наповнений великими гравцями	Бажання клієнтів використовувати перевірені застосунки часом
2	Економічний	Не бажання платити за додатковий функціонал	Пошук нової цільової аудиторії
3	Якість	Збої у системі аналізу сигналу	Відмова від продукту
4	Некомпетентність робітників	Неточність інженерних розрахунків	Відмова від продукту

Висновки: Головним фактором загроз є конкуренція, тому що користувач може обрати перевірену часом систему та згоден витратити більше часу на її освоєння. Також конкуренти мають достатню частку ринку для його демпінгування.

Надалі проведемо аналіз пропозиції (табл. 4.8.), де визначимо загальні риси конкуренції на ринку.

Таблиця 4.8 Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції - чиста конкуренція	Відсутні конкуренти, які мають онлайн додаток по аналізу сигналу	Поява нових гравців

Продовження таблиці 4.8

2. За рівнем конкурентної боротьби - міжнародний	Відсутні конкуренти, які мають онлайн додаток по аналізу сигналу	Поява нових гравців
--	--	---------------------

3. За галузевою ознакою - міжгалузева	Використання у різних галузях	Робота менеджменту і реклами по залученню клієнтів
4. Конкуренція за видами товарів: - товарно-видова	Товар, що пропонується є специфічний та цілонапрямний	Клієнтно-орієнтована стратегія розвитку Адаптивність до умов ринку
5. За характером конкурентних переваг - цінова	Вартість залежить від специфічних запитів клієнтів	Пошук великих клієнтів, які потребують швидкий аналіз великого об'єму даних
6. За інтенсивністю - марочна	Вибір серверів на яких буде знаходитися застосунок	Розкрутка бренду, його рекламування

Висновки: На ринку відсутня чиста конкуренція через те, що окремі гравці не можуть впливати на ціну товару. За рівнем конкурентної боротьби – міжнародна з міжгалузевою ознакою, так як можливі іноземні замовники. Конкуренція за видами товарів – видова. Вартість товару на пряму залежить від комплектації та функціоналу, тому за характером конкурентних переваг – цінова.

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі за моделлю п'яти сил М. Портера (табл. 4.9.)

Таблиця 4.9 Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
------------------	---------------------------	-----------------------	---------------	---------	------------------

	-	Excel, Matlab	Heroku	Building company Клініки Держав ні установ и	Алгоритм Excel, Matlab
Висновки: ринок не є насиченим, але конкуренція висока, оскільки компанії які являються конкурентами мають високу репутацію. Завадою являється необхідний високий рівень капіталовкладень	-	Вихід на ринок відносно простий. Наявні потенційні конкуренти	Постачальники мають не диктувати ціни на ринку	Клієнти не диктують умови через велику зацікавленість до товару	Обмежень практично немає, оскільки існують різні обмеження по використанню

На основі аналізу конкуренції, проведеного наведеного в табл. 4.9, а також із урахуванням характеристик ідеї проекту, що наведені в табл. 4.2, вимог споживачів до товару (табл. 4.5.) та факторів маркетингового середовища (табл. 4.6, 4.7) визначимо та обґрунтуємо перелік факторів конкурентоспроможності. Аналіз наведено в таблиці 4.10.

Таблиця 4.10 Обґрунтування факторів конкурентоспроможності

№ п/ п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Надійність	Використання алгоритму з високою точністю. Чим вище точність розрахунку, тим вище попит на товар

2	Простота	Простота у використанні для користувача робить систему привабливою для клієнта.
3	Адаптивність	Можливість гнучкого налаштування під конкретні задачі
4	Готовий алгоритм	Зменшення часу очікування для покупця
5	Обслуговування	Потреба в консультації з технічних питань

Висновки: Для підвищення надійності системи необхідно використовувати алгоритми з високою точністю. Простота у використанні та налаштуванні приладу робить його більш привабливим для клієнта через те, що необхідно тратити мінімум часу на налагодження та запуск. Адаптивність та простота відкривають широкий спектр використання приладу у поєднанні із іншими системами.

За визначеними факторами конкурентоспроможності (табл. 4.10) проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

Таблиця 4.11 Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів						
			-3	-2	-1	0	+1	+2	+3
1	Надійність	18							+3
2	Простота	20							+3
3	Адаптивність	17							+3
4	Готовий алгоритм	16				0			
5	Обслуговування	18						+2	

Висновки: спираючись на фактори конкурентоспроможності (Таблиця 4.11) та підсумовуючи рейтинг товару відносно головного конкурента, запропонована система має більший рейтинг відносно прямих конкурентів. Дана таблиця показує якими саме особливостями розроблена система відрізняються від аналогів та в яку саме сторону. Детальний аналіз показує, що сильними сторонами є простота, можливість до адаптації з роботою із іншими системами.

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities), що наведено в таблиці 4.12. на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін в таблиці 4.11.

Таблиця 4.12 SWOT- аналіз стартап-проекту

Сильні сторони: 1. простота використання; 2. адаптивність; 3. стабільна робота; 4. універсальність; 5. портативність; 6. можливість до комплексування	Слабкі сторони: 1. низька репутація підприємства на початку впровадження проекту в життя; 2. обмежений функціонал
Можливості: 1. вихід на міжнародний ринок; 2. збільшення попиту; 3. необхідність до інтеграції; 4. освоєння нових сфер; 5. індивідуальне замовлення; 6. співпраця з конкурентами	Загрози: 1. поява схожих застосунків 2. кваліфікованість спеціалістів;

На основі SWOT-аналізу розробимо альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів (табл. 4.13).

Таблиця 4.13 Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Індивідуалізм (максимізація власного виграшу)	Середня	2міс
2	Кооперація (максимізація спільного виграшу)	Висока	3міс
	Суперництво	Нижче середнього	5міс.

Висновки: в результаті аналізу обрано кооперацію, як альтернативну ринкову поведінку через те, що за відносно короткий термін існує велика ймовірність отримання ресурсів.

4.4 Розроблення ринкової стратегії проекту

Розробка ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 4.14).

Таблиця 4.14 Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачі в сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Будівництво інженерних споруд(будинків, мостів)	+	+	низька	+
2	Державні установи	-	+	низька	-
3	Медичні центри діагностики	+	+	низька	+

Продовження таблиці 4.14

4	Національний банк України	+	+	висока	+
5	Центри психології	+	+	низька	-
Які цільові групи обрано: під час вибору цільової групи до уваги бралось в першу чергу готовність споживача сприйняти продукт. Будівельні компанії потребують більш популярних роз'яснень щодо можливостей використання таких систем. У випадку успішної реклами є можливість зайняти нішу.					

Висновки: За результатами оцінки були обрані групи, які потребують швидкого реагування та відповідно швидкого оцінювання даних для подальшого

впровадження тих чи інших дії. З таблиці 4.14 можна побачити що обрані клієнти потребують дані рішення, але ще є небажання витратити на це гроші та нерозуміння для чого їм це.

Для роботи в обраних сегментах ринку необхідно сформулювати базову стратегію розвитку (табл. 4.15.).

Висновки: Кожне підприємство потребує швидкого реагування на проблеми, які можна побачити на ранніх етапах дослідження або вивчення нового матеріалу.

Таблиця 4.15 Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Індивідуалізм	Стратегія недиференційованого маркетингу	Адаптація до вимог ринку Використання новацій Генерування ноу-хау	Стратегія спеціалізації

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 4.16).

Таблиця 4.16 Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Є першопрохідцем	Шукати нових споживачів із поступовим	Компанія буде вдосконалюва	Стратегія виклику лідера

		переманюванням від конкурентів	ти наявний застосунок	
--	--	-----------------------------------	--------------------------	--

Висновок: оскільки проект є першопрохідцем та має суттєві переваги по відношенню до свого прямого конкурента, можливо обрати стратегію виклику лідеру. Це є можливим на фоні використання інноваційних методів підвищення точності алгоритмічного забезпечення, впровадження інновацій і технологій для вдосконалення продукту з метою ускладнення задачі конкурентів;

На основі вимог споживачів з обраних сегментів до постачальника та до продукту, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розробимо стратегію позиціонування (табл. 4.17).

Таблиця 4.17 Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформулювати комплексну позицію власного проекту (три ключових)
1	Швидкість роботи	Стратегія спеціалізації	Дослідження та розвиток	Високошвидкісна, легкість, надійність
2	Можливість відкритого API	Стратегія спеціалізації	Обслуговування	Здатна до відкритого API
3	Висока стабільність результатів	Стратегія спеціалізації	Якість	високоточна

Висновки: Дана стратегія є ключовою для подальшого розвитку проекту та отримання прибутку для реінвестування у проект та купівлю нових технологій з подальшою інтеграцією у проект.

4.5 Розроблення маркетингової програми стартап-проекту

Маркетингова концепція товару є першим кроком до формування ідеї яку отримує споживач, наведено у таблиці 4.18.

Таблиця 4.18 Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Необхідність швидкого спектрального аналізу сигналу	Відносно дешева та надійна система	Ціна, надійність, простота, підтримка, обслуговування, універсальність
2	Навчання операторів спектральної оцінки сигналу	Повний функціонал та гнучкість системи генерувати сигнали	Ціна, надійність, простота, підтримка, обслуговування, універсальність

Висновки: в результаті оцінки маркетингової концепції є можливість створити дві цільові аудиторії для подальшого створення рекламних кампаній.

Розробимо трирівневу маркетингову модель товару (табл. 4.19).

Таблиця 4.19 Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Засіб дає можливість користувачеві швидко оцінити отримані дані сигналу та зробити звітність на основі цього. Та є можливість штучної генерації даних

Продовження таблиці 4.19

II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Доступність	онлайн	Тх
	Швидкість роботи	1-2 с	Тх
	Точність	0,1° СКВ	Тх
	Надійність	IP67	Тх
	Вартість	12	Е
	Якість: сертифікати відповідності якості ISO, сертифікація для використання в медичних закладах, сертифікація IEEE.		
Пакування: -			
Марка: -			
III. Товар із підкріпленням	Додавання нового функціоналу для користувача		
	Після оформлення підписки отримання нового функціоналу, цілодобова технічна підтримка		
Захист товару проводиться шляхом постійного додавання нового функціоналу та захоплення нових ніш ринку			

Висновки: основними засобами захисту застосунку є постійне оновлення та додавання нового функціоналу для конкурентоспроможного застосунку. Також постійний аналіз та викуп новітніх застосунків для подальшої інтеграції їх у систему.

Наступним кроком визначимо цінові межі (табл. 4.20).

Таблиця 4.20 Визначення меж встановлення ціни

№ п/ п	Рівень цін на товари-замі ники	Рівень цін на товари-анал оги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	2350 у.о.	2000 у.о.	10000 у.о.	50-3000 у.о.

Висновки: Для даного товару планується зробити систему підписок для більш гнучкої системи для користувача, який готовий платити лише за окремий функціонал.

Визначимо оптимальну систему збуту, в межах якого будуть прийматися ті чи інші рішення(табл. 4.21).

Таблиця 4.21 Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Підписка	Оформлення підписки для доступу до повного функціоналу товару	Нульового рівня	Безпосередній (прямий)

Висновки: Ефективною вважається система по оформленні підписки, яка допозже отримувати кошти регулярно та більш гнучкі для самого користувача. Через що виникає відносини він-він де кожен обирає той рівень підписки, якій йому потрібен.

Таблиця 4.22 Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Обставини змушують клієнта шукати швидкі оцінки сигналу для	Реклама SMM	Висока швидкість роботи Легкість Надійність Зручність Функціональність	Показати швидкість оцінки результатів	Показ зручності та можливості роботи з отриманими даними

	мінімізації витрат				
--	-----------------------	--	--	--	--

Висновки: Маркетинг проходить через різні канали розміщення рекламних об'яв, а саме Facebook Ads, Google Ads, TikTok, LinkedIn. Цільова аудиторія повинна побачити швидкість та можливості програми менше ніж за 15 секунд.

4.6. Розроблення бізнес-моделі стартап-проекту

Бізнес-модель стартап-проекту поділено на декілько пунктів, а саме: ключові партнери, ключова діяльність, ключові ресурси, ціннісна пропозиція, відносини з клієнтами, канали, сегменти користувачів, структура витрат, джерела доходів[40]. Розглянемо кожний з пунктів окремо.

Ключові партнери стартап-проекту є компанії по розробці хмарних сховищ для зберігання та швидшого оброблення даних. Наприклад: Amazon Web Service, Google Cloud і т.д. Від цього партнерства можна зменшити кількість витрат на сервері та збільшити кількість ресурсів самого сервера. Для партнера вигідно з нами працювати для подальшого захоплення ринку даних аналітики та оплата за предоставлені послуги.

Ключовою діяльністю стартап-проекту є створення веб додатку, який дозволяє швидко оцінити сирі дані сигналу та провести частотний аналіз сигналу, а також розвинення оцінювання частотних характеристик для операторів-початківців. Канали продажу є email розсилки та приватні зустрічі з CEO підприємств для подальшої точкової угоди. Розглядається можливість прямої інтеграції інших підприємств через API.

Ключовими ресурсами є люди, які мають технічну освіту в області обробки сигналів та їх аналізу, а також знання мов програмування для автоматизації системи. Також виникає потреба у системах з великим об'ємом оперативної пам'яті для більш швидкої обробки інформації.

Ціннісна пропозиція у стартапі є більш швидке первинне оброблення сигналу ніж у конкурентів, що дозволяє на ранніх етапах оцінювати сигнал для

подальших дій. А також формування навичок аналізу для операторів за допомогою генерації штучних сигналів. Також є пропозиція по підключенню через відкрите API для роботи напряду через софт компанії, це дозволяє швидше інтегрувати систему у екосистему компанії. Дизайн та сам софт більш зручний та інтуїтивно зрозуміліше ніж у конкурентів.

Відносини з клієнтами будуються на основі зв'язків та дружніх відносин, адже програма потребує точених рішень для масштабування проекту під кожен окрему компанію, що дає можливість збільшити ціну за послуги. Також для покращення впізнаваності проекту серед спеціалізованого середовища будуть створені спеціалізовані івенти для збільшення кількості клієнтів.

Сегменти стартап проекту діляться на дві групи: компанії у яких є потреба у підвищенні кваліфікації своїх співробітників у аналізі сигналів та первинний частотний аналіз сигналу за рахунок імпортування сирих даних сигналу. Ринок у якому планується розвиватися стартап є нішевим.

Канали з прямими клієнтами є email спілкування та прямі зустрічі за келихом вина або за чашкою кави.

Структурою витрат стартап-проекту є високваліфіковані інженери, які можуть проводити масштабування та розвиток веб додатку та сервіси на яких будуть знаходитися програма по обробці даних.

Джерелами доходів є навчання операторів-новачків по навчанню їх аналізувати та генерувати сигнали, також продаж готових рішення які дозволяють автоматизувати частотний аналіз сигналу через API або веб додаток.

Висновок: проект націлений на нішеву аудиторію з подальшим обговоренням бажань клієнт по інтеграції софту у інфраструктуру компанії для її збільшення можливостей. Це дозволяє змінювати ціни у залежності від потреб компаній для інтеграції продукту.

Висновки до розділу 4

Даний розділ описує стратегію по створенню стартап-проекту. Для подальшого збуту готового рішення по ефективній обробці сигналу, які дозволять

підприємству швидко приймати рішення та навчати персонал для подальшого отримання грошей за рахунок зниження витрат. Тому розроблений програмний застосунок має кумулятивний ефект для підприємства та може бути продана зацікавленим особам.

Для освітлення ідеї проекту була описана інформація у таблиці 4.1., у якій приведено назву проекту, можливості котрі може використати клієнт та які ризики він несе при використанні даного застосунку. Також було проведено аналіз слабких та сильних сторін, порівняння продукту не з прямими конкурентами. Проведений аналіз свідчить про відсутність прямих конкурентів у даній ніші. Також сильними сторонами проекту стали швидкість, зручність та інтуїтивно зрозумілий інтерфейс застосунку, що дає максимально понурення клієнта.

Під час технічного аудиту проекту були отримане розуміння кращої технології виконання продукту. Виявлено, що технології вже існують, але не є інтуїтивно зрозумілі та потребує високого рівня розуміння середовища у якому використовують обробку сигналів. Загалом, створення проекту можливо за рахунок двох шляхів - це подальший розвиток у навчальних закладах та інтегрування у навчальну програму, або часні пропозиції до компаній для опису можливостей, які вони отримують після оформлення підписки.

Зважаючи на розвиток технологій та об'єм даних, які наразі повинен обробляти бізнес для роботи. Він потребує у створенні програм, які швидко дозволяють проводити аналіз даних та подальшу роботу з обробленими даними.

Розроблена бізнес-модель стартап-проекту що показує вигідність інвестуванню у проект, по причині нишевості проекту та відсутності прямих конкурентів по інтеграції готової систему у екосистеми існуючих компаній.

ВИСНОВКИ

Використання швидких автоматизованих систем для частотної оцінки сигналів у системах є важливим і актуальним завданням у зв'язку з великим обсягом даних та потребою швидкого реагування на проблеми на початковому етапі обробки сигналу. Тому у дисертації було створено веб додаток, який автоматизує роботу по частотній обробці сигналу та отримано наступні науково-технічні результати:

1. Проведено огляд сучасних рішень по обробці сигналів та їх реалізації. Дослідження плюси та мінуси з кожного з рішень для подальшого використання та оптимізації процесів. Показано, що використання Python та використання його бібліотек та фреймворків дозволяє побудувати великий проект по аналітиці та обробці великих обсягів даних.

2. Проведений огляд ефективності використання фреймворка Django для побудови веб додатку з подальшим масштабуванням. Описані основні патерни роботи Django. Проведений порівняльний аналіз з іншими рішеннями, які оцінювалися при дослідженні.

3. Проведене дослідження по використанню Pandas для побудови високонавантажених систему з великим обсягом даних. Проведений порівняльний аналіз з бібліотекою Numpy. Дана оцінка та порівняльні характеристики бібліотеки Pandas з іншими рішеннями.

4. Спроектований веб додаток, який дозволяє аналізувати сигнал у двох форматах: за допомогою імпортування файлу з довільним розширенням та генерації штучного сигналу для навчання персоналу. При розробці системи було приділено увагу безпеці, швидкості, зручності та надійності системи.

5. Проведений порівняльний аналіз з пакетним рішенням Matlab за параметрами: швидкодія, стабільність, зручність, надійність. Під час аналізу були виявлено, що веб додаток працює швидше у 7-8 разів у порівнянні з Matlab. Дає можливість безкоштовно використовувати функціонал та використовувати його з різних застосунків онлайн. Надійність гарантується SSL підключенням між

сервером та клієнтом. Дизайн зроблений інтуїтивно зрозуміла, що дає швидке розуміння як працює сайт. Сайт працює цілодобово та підтримує роботу одночасно багатій кількості людей.

6. Надані рекомендації щодо того як працює система, та пояснена кодова база самого проекту для подальшого використання у інших проектах.

1. Microsoft Excel®: Is It An Important Job Skill for College Graduates? [Электронный ресурс] // Information Systems Education Journal (ISEDJ). – 2017. – Режим доступа до ресурсу: <https://files.eric.ed.gov/fulltext/EJ1140801.pdf>.
2. Staff W. What Is the Purpose of Microsoft Excel? [Электронный ресурс] / Writer Staff // Reference. – 2020. – Режим доступа до ресурсу: <https://www.reference.com/world-view/purpose-microsoft-excel-d7cf0fb45c44039e>.
3. Lavery S. Importance of Microsoft Excel [Электронный ресурс] / Shea Lavery // ItStillWorks – Режим доступа до ресурсу: <https://itstillworks.com/importance-microsoft-excel-1716.html>.
4. Statistical software for data science [Электронный ресурс] – Режим доступа до ресурсу: <https://soware.ru/products/stata>.
5. What advantages and disadvantages does R have over Stata, i.e., what is R's ability to quickly put together a picture of the development of a firm, country, city, or of other macro-data versus Strata's capability to do the same? [Электронный ресурс] // Quora – Режим доступа до ресурсу: <http://surl.li/djlcq>
6. Stata for Students: Using Stata [Электронный ресурс] // SSCC(social science computing cooperative). – 2011. – Режим доступа до ресурсу: <https://www.ssc.wisc.edu/sscc/pubs/4-30.htm>.
7. Muenchen R. The Popularity of Data Science Software [Электронный ресурс] / Robert Muenchen. – 2010. – Режим доступа до ресурсу: <https://r4stats.com/articles/popularity/>.
8. IBM SPSS Review [Электронный ресурс] – Режим доступа до ресурсу: <https://comparecamp.com/ibm-spss-review-pricing-pros-cons-features/>.
9. What is IBM SPSS? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.trustradius.com/products/ibm-spss/reviews?qs=pros-and-cons#overview>.

10. Weston S. Why You Should Become a UseR: A Brief Introduction to R [Электронный ресурс] / S. Weston, D. Yee. – 2017. – Режим доступа до ресурсу: <https://www.psychologicalscience.org/observer/why-you-should-become-a-user-a-brief-introduction-to-r>.

11. What are the pros and cons of the R programming language? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.quora.com/What-are-the-pros-and-cons-of-the-R-programming-language>.

12. Johnson D. SAS Tutorial for Beginners: What is & Programming Example [Электронный ресурс] / Daniel Johnson. – 2022. – Режим доступа до ресурсу: <https://www.guru99.com/sas-tutorial.html>.

13. SAS [Электронный ресурс] – Режим доступа до ресурсу: <https://support.sas.com/en/support-home.html>.

14. Advantages of SAS | Disadvantages of SAS Programming [Электронный ресурс] – Режим доступа до ресурсу: <https://data-flair.training/blogs/disadvantages-and-advantages-of-sas/>

15. Advantages of MATLAB | Disadvantages of MATLAB [Электронный ресурс] – Режим доступа до ресурсу: <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-MATLAB.html>.

16. Pedamkar P. Uses of Matlab [Электронный ресурс] / Priya Pedamkar – Режим доступа до ресурсу: <https://www.educba.com/uses-of-matlab/>.

17. Benefits of minitlab statistical software [Электронный ресурс] – Режим доступа до ресурсу: <https://find-study-now.com/benefits-of-minitab-statistical-software/>

18. Minitab Review [Электронный ресурс] – Режим доступа до ресурсу: <https://www.capterra.com/p/109731/Minitab-Statistical-Software/reviews/>.

19. Scilab Review [Электронный ресурс] – Режим доступа до ресурсу: <https://comparecamp.com/scilab-review-pricing-pros-cons-features/>.

20. About Scilab [Электронный ресурс] – Режим доступа до ресурсу: <https://www.softwareadvice.co.uk/software/292956/scilab>.

21. What are the disadvantages of Scilab? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.quora.com/What-are-the-disadvantages-of-Scilab>.

22. Karczewski D. Why Is Python a Great Choice For Data Analysis? [Электронный ресурс] / Dawid Karczewski. – 2021. – Режим доступа до ресурсу: <https://www.ideamotive.co/blog/python-for-data-analysis>.

23. What are some pros and cons of using Python for data science? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.quora.com/What-are-some-pros-and-cons-of-using-Python-for-data-science>.

24. GraphPad Prism [Электронный ресурс] – Режим доступа до ресурсу: <https://thinkbiganalytics.com/graphpad-prism>.

25. GraphPad Prism Reviews & Product Details [Электронный ресурс] – Режим доступа до ресурсу: <https://www.g2.com/products/graphpad-prism/reviews>.

26. Role of Data Science in Java: 10 Critical Aspects [Электронный ресурс] // HEVO. – 2021. – Режим доступа до ресурсу: <https://hevodata.com/learn/data-science-in-java/#m4>.

27. Top 6 Reasons Data Scientists Should Know Java [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kdnuggets.com/2020/06/top-6-reasons-data-scientists-know-java.html>.
28. What is Python? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.python.org/doc/essays/blurb/>.
29. Костюченко А. О. ОСНОВИ ПРОГРАМУВАННЯ МООВОЮ PYTHON / А. О. Костюченко. – Чернігів, 2020. – 180 с.
30. Django introduction [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction#:~:text=Django%20is%20a%20high%2Dlevel,needing%20to%20reinvent%20the%20wheel..>
31. News aggregator web application using Django. // INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS. – 2021. – №2320. – С. 6
32. Django Advantages and Disadvantages – Why You Should Choose Django? [Електронний ресурс] – Режим доступу до ресурсу: <https://data-flair.training/blogs/django-advantages-and-disadvantages/>.
33. Python, Django top frameworks and it's comparison [Електронний ресурс] – Режим доступу до ресурсу: <https://micropyramid.com/blog/python-django-top-frameworks-and-its-comparison/>.
34. Top 11 Python Frameworks You Must Know in 2022 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.interviewbit.com/blog/python-frameworks/>.
35. What Is Pandas In Python? Everything You Need To Know [Електронний ресурс] – Режим доступу до ресурсу: <https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/>.
36. Введение в pandas: анализ данных на Python [Електронний ресурс] – Режим доступу до ресурсу: <https://khashtamov.com/ru/pandas-introduction/>.

37. Pros and Cons of using Pandas [Електронний ресурс] – Режим доступу до ресурсу: <https://www.studytonight.com/pandas/pros-and-cons-of-using-pandas>.

38. Pandas Vs NumPy: What's The Difference? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.interviewbit.com/blog/pandas-vs-numpy/>.

39. Технологічний аудит підприємства. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.interviewbit.com/blog/pandas-vs-numpy/>.

40. Макет(канва) бізнес моделі. [Електронний ресурс] - Режим доступу до ресурсу :
<https://www.kiwka.com/maket-kanva-biznes-modeli-pereklad-ukrajinskoyu/> .

Додаток А

Код веб додатку

Файл main.py:

```
#!/usr/bin/env python
```

```
"""Django's command-line utility for administrative tasks."""
```

```

import os
import sys

def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE',
'signal_modeling.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
Файл settings.py:
import os
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
SECRET_KEY =
'q@$@+aietc2^e#3!k4e+_%)gtffhd@xcdbidrmegf5t#bg-@=8'
DEBUG = False
ALLOWED_HOSTS = ['masterdegreeproject1234.herokuapp.com']
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'crispy_forms',
    'time_analysis',
]
CRISPY_TEMPLATE_PACK = 'bootstrap4'
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',

```

```

'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
ROOT_URLCONF = 'signal_modeling.urls'
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
WSGI_APPLICATION = 'signal_modeling.wsgi.application'

```

2. urls.py - головний файл з посиланням на сторінки, код файлу:

```

from django.contrib import admin
from django.urls import path, include

from signal_modeling.views import MainPage

```

```

urlpatterns = [
    path("", MainPage.as_view(), name='main-page'),
    path("", include('time_analysis.urls')),
]

```

Файл forms.py:

```

from utils.custom_range import crange
import numpy as np
from pandas import DataFrame
from utils.base_forms import AnalyticBaseForm

```

```

from utils.parser_files import convertor_file_to_df
from django.forms import FileField, FloatField, IntegerField, ChoiceField,
BooleanField, MultipleChoiceField
from acoustics.generator import noise as noise_generator
from utils.base_validators import validate_file_extension

class TimeAnalyticForm(AnalyticBaseForm):

    class SignalType:
        DETERMINATION = 'determination'
        STOCHASTIC = 'stochastic'

    SIGNAL_TYPE = (
        (SignalType.DETERMINATION, 'детермінований'),
        (SignalType.STOCHASTIC, 'стохастичний (випадковий)')
    )

    signal_type = ChoiceField(choices=SIGNAL_TYPE, label='Тип сигналу',
required=False)
    input_data_file = FileField(label='Файл з сирими даними', required=False,
allow_empty_file=False, validators=[validate_file_extension])

    def calculation_data(self, df: DataFrame) -> dict:
        signal_type_calculation = {
            self.SignalType.DETERMINATION: self._determination_data,
            self.SignalType.STOCHASTIC: self._stochastic_data
        }
        analytics_data =
signal_type_calculation[self.cleaned_data['signal_type']](df)
        graphs_data = self._get_graphs_data(df.copy())
        return {
            'analytics_data': analytics_data,
            'graphs_data': graphs_data
        }

    def is_valid(self) -> bool:
        is_valid = super().is_valid()
        if self.cleaned_data['input_data_file'] is None:
            self.errors['input_data_file'] = ['Будь ласка, оберіть файл', ]
            return False

```



```

    return is_valid

def get_dataframe_from_file(self) -> DataFrame:
    file = self.cleaned_data['input_data_file'].file
    filename = self.cleaned_data['input_data_file'].name

    return convertor_file_to_df.convert(file, filename)

class CustomAnalyticForm(AnalyticBaseForm):
    class SignalType:
        COS = 'cos'
        SIN = 'sin'

    SIGNAL_TYPE = (
        (SignalType.COS, 'cos'),
        (SignalType.SIN, 'sin')
    )

    class Noises:
        WHITE = 'white'
        PINK = 'pink'
        BLUE = 'blue'
        BROWN = 'brown'
        VIOLET = 'violet'

    NOISES = (
        (Noises.WHITE, 'Білий шум'),
        (Noises.PINK, 'Рожевий шум'),
        (Noises.BLUE, 'Синій шум'),
        (Noises.BROWN, 'Коричневий шум'),
        (Noises.VIOLET, 'Фіолетовий шум'),
    )

    noises = MultipleChoiceField(label='Шуми', choices=NOISES,
required=False)
    snr = FloatField(label='SNR', min_value=0.00001, required=False)
    type_of_signal = ChoiceField(required=False, choices=SIGNAL_TYPE)
    mean = FloatField(label='Середнє значення', required=True)
    scope = FloatField(label='Розмах', required=True)

```

```

        checker_count_of_dot_or_period_sampling =
BooleanField(label='Обраховувати через кількість періодів', required=False)
        count_of_periods = IntegerField(label='Кількість періодів',
required=False)
        count_of_dots = IntegerField(label='Кількість точок', required=False)
        period_sampling = FloatField(label='Період дискретизації',
required=True)
        frequency_sampling = FloatField(label='Частота дискретизації',
required=True)
        frequency = IntegerField(label='Частота', required=True)

    def calculation_data(self, df: DataFrame) -> dict:
        analytics_data = self._stochastic_data(df) if
self.cleaned_data.get('noises') else self._determination_data(df)
        graphs_data = self._get_graphs_data(df.copy())

        return_data = {
            'analytics_data': analytics_data,
            'graphs_data': graphs_data,
        }
        if self.cleaned_data.get('noise_data'):
            return_data['noise_data'] = self._get_noise_data()
        return return_data

    def _get_noise_data(self) -> dict:
        noise_data = self.cleaned_data['noise_data']
        calculation_data = self._get_calculation_data(noise_data['y'])
        return {
            'calculation_data': calculation_data,
            'plot_data': {
                'y': list(noise_data['y']),
                'x': list(noise_data['x'])
            }
        }

    def _get_calculation_data(self, noise_data) -> dict:

        return {
            'snr': round(self.cleaned_data['snr'], 3),
            'min': round(np.min(noise_data), 3),

```

```

        'max': round(np.max(noise_data), 3),
        'mean': round(np.mean(noise_data), 3),
        'std': round(np.std(noise_data), 3),
        'dispersion': round(np.var(noise_data) , 3),
    }

def get_dataframe(self) -> DataFrame:
    f = self.cleaned_data['frequency']
    T=1/f
    fd=25*f
    Td=1/fd

    type_of_signal = self.cleaned_data['type_of_signal']
    rozmah = self.cleaned_data['scope']
    mean = self.cleaned_data['mean']
    p = self.cleaned_data['count_of_periods']
    count_of_dots = self.cleaned_data['count_of_dots']
    checker_count_of_dot_or_period_sampling =
self.cleaned_data['checker_count_of_dot_or_period_sampling']
    noises = self.cleaned_data['noises']

    if checker_count_of_dot_or_period_sampling:
        t = crange(Td,p*T,Td)
    else:
        t = np.linspace(Td, count_of_dots*Td, count_of_dots)
    z = 2*np.pi*t*f
    zz = rozmah/2*getattr(np, type_of_signal)(z)
    y = float(mean)+ zz
    if noises:
        snr = self.cleaned_data['snr']

        noise_data = np.zeros(len(y))
        for noise in noises:
            noise_data += noise_generator(len(y), noise)

        koef = np.sqrt( np.mean( np.power(y,2) ) / ( np.power(10,(snr/10)) *
np.mean(np.power(noise_data,2)) ) )
        noise_data = noise_data * koef
        self.cleaned_data['noise_data'] = {
            'y': noise_data,

```

```

        'x': t
    }

    self.cleaned_data['signal_type'] =
TimeAnalyticForm.SignalType.STOCHASTIC

    y += noise_data

    return DataFrame({'t': t.tolist(), 'y': y.tolist()})

def is_valid(self) -> bool:
    is_valid = super().is_valid()
    frequency = self.cleaned_data['frequency']
    frequency_sampling = self.cleaned_data['frequency_sampling']
    period_sampling = self.cleaned_data['period_sampling']

    if not ( frequency * 2 <= frequency_sampling) :
        self.errors['frequency'] = ['Частота та частота дискретизації не
співпадають за теоремою Найквіста Коперніка', ]
        return False

    if not ( round(1/frequency_sampling, 10) == round(period_sampling, 10)
):
        self.errors['period_sampling'] = ['Період дискретизації та частота
дискретизації взаємопов'язані за формулою період дискретизації = 1/частота
дискретизації', ]
        self.errors['frequency_sampling'] = ['Період дискретизації та частота
дискретизації взаємопов'язані за формулою період дискретизації = 1/частота
дискретизації', ]
        return False

    return is_valid
Файл views.py:
import json
from typing import Any, Dict

from django.http import JsonResponse
from utils.mixins import ExportFileViewMixin
from utils.base_forms import DownloadAnalyticFilesForm
from django.views.generic.edit import FormView
from pandas import DataFrame

```

```

from time_analysis.forms import CustomAnalyticForm, TimeAnalyticForm

class TimeAnalysisView(ExportFileViewMixin, FormView):
    template_name = "time_analysis.html"
    form_class = TimeAnalyticForm

    def get_context_data(self, **kwargs: Any) -> Dict[str, Any]:
        context_data = super().get_context_data(**kwargs)
        if kwargs:
            context_data['export_file_form'] =
DownloadAnalyticFilesForm(data={'context_data_field': kwargs.get('data',
{'').get('calculated_data_json'), 'file_type':
DownloadAnalyticFilesForm.FileType.TXT})
        else:
            context_data['export_file_form'] = DownloadAnalyticFilesForm
        return context_data

    def form_valid(self, form):
        income_dataframe: DataFrame = form.get_dataframe_from_file()
        data = {}
        data['headers'] = income_dataframe.columns.tolist()
        data['calculated_data'] = form.calculation_data(income_dataframe)
        data['calculated_data_json'] = json.dumps(data['calculated_data'])
        income_dataframe = income_dataframe.round(5)
        data['data_for_graph'] = income_dataframe.to_dict('list')
        context = self.get_context_data(data=data, form=form)
        return self.render_to_response(context)

class CustomAnalyticView(ExportFileViewMixin, FormView):

    template_name = "time_analysis.html"
    form_class = CustomAnalyticForm

    def default_data(self):

        with open('./time_analysis/result.json', 'r') as fcc_file:
            data = json.load(fcc_file)

```

```

return {
    'data': data,
    'form': self.form_class(
        data = dict(
            type_of_signal = self.form_class.SignalType.SIN,
            mean = 3,
            scope = 4,
            snr=1,
            count_of_periods = 2,
            frequency_sampling = 500,
            period_sampling = 0.002,
            frequency = 99,
            count_of_dots=22,
            checker_count_of_dot_or_period_sampling=True
        )
    )
}

```

```

def get_context_data(self, **kwargs: Any) -> Dict[str, Any]:
    if not kwargs:
        kwargs = self.default_data()
    context_data = super().get_context_data(**kwargs)
    if kwargs:

```

```

        context_data['export_file_form'] =
DownloadAnalyticFilesForm(data={'context_data_field': kwargs.get('data',
{}).get('calculated_data_json'), 'file_type':
DownloadAnalyticFilesForm.FileType.TXT})

```

```

    else:
        context_data['export_file_form'] = DownloadAnalyticFilesForm
    return context_data

```

```

def form_valid(self, form):
    income_dataframe: DataFrame = form.get_dataframe()
    data = {}
    data['headers'] = income_dataframe.columns.tolist()
    data['calculated_data'] = form.calculation_data(income_dataframe)
    data['calculated_data_json'] = json.dumps(data['calculated_data'])
    income_dataframe = income_dataframe.round(5)
    data['data_for_graph'] = income_dataframe.to_dict('list')
    context = self.get_context_data(data=data, form=form)

```

```

        return JsonResponse(context['data'])
Файл base_forms.py:
import math
from typing import List, Union
import numpy as np

from pandas import DataFrame
from scipy.fft import fft, fftshift
from scipy.signal import periodogram
from scipy.signal.windows import (
    hann, taylor, boxcar, triang, blackman, hamming, bartlett, flattop,
    parzen, bohman, blackmanharris, nuttall, barthann, cosine,
    exponential, tukey
)

from django.forms.fields import ChoiceField
from django.forms import Form, ChoiceField, HiddenInput, JSONField
from pandas import DataFrame

class AnalyticBaseForm(Form):

    def _get_graphs_data(self, df: DataFrame) -> dict:
        kilkist_vidlikiv = self._get_kilkist_vidlikiv(df)

        period_descritiatcii = self._get_period_descritiatcii(df)
        chastota_descritiatcii = self._get_chastota_descritiatcii(df)

        fft_data = self._get_fft_data(df)
        periodograms_without_signal = self._get_periodograms_without_signal(df)
        periodograms_without_signal_frequency = self._get_periodograms_without_signal_frequency(periodograms_without_signal)
        return {
            'period_descritiatcii': period_descritiatcii,
            'kilkist_vidlikiv': kilkist_vidlikiv,
            'chastota_descritiatcii': chastota_descritiatcii,

            'fft': fft_data,
            'periodogram': self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['periodogram']),

```

```

        'triangle_periodogram': self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['triangle_periodogram']),
        'hann_periodogram': self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['hann_periodogram']),
        'blackman_periodogram': self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['blackman_periodogram']),
        'hamming_periodogram': self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['hamming_periodogram']),
        'bartlett_periodogram': self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['bartlett_periodogram']),
        'flattop_periodogram': self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['flattop_periodogram']),
        'parzen_periodogram': self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['parzen_periodogram']),
        'bohman_periodogram': self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['bohman_periodogram']),
        'blackmanharris_periodogram':
self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['blackmanharris_periodogram']),
        'nuttall_periodogram': self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['nuttall_periodogram']),
        'barthann_periodogram': self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['barthann_periodogram']),
        'cosine_periodogram': self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['cosine_periodogram']),
        'exponential_periodogram':
self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['exponential_periodogram']),
        'tukey_periodogram': self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['tukey_periodogram']),
        'taylor_periodogram': self._get_periodogram_data_by_widnow(df,
periodograms_without_signal['taylor_periodogram']),

        'periodograms_without_signal': periodograms_without_signal,
        'periodograms_without_signal_frequency':
periodograms_without_signal_frequency,
    }

def _stochastic_data(self, df: DataFrame) -> dict:
    data = {}

```



```

data['min'] = self._get_min(df)
data['max'] = self._get_max(df)
data['median'] = self._get_median(df)
data['mean'] = self._get_mean(df)
data['quantile'] = self._get_quantile(df)
data['dispersion'] = self._get_dispersion(df)
data['std'] = self._get_std(df)
data['mathematical_expectation'] =
self._get_mathematical_expectation(df)
return data

```

```

def _determination_data(self, df:DataFrame) -> dict:
    data = {}
    data['min'] = self._get_min(df)
    data['max'] = self._get_max(df)
    data['median'] = self._get_median(df)
    data['mean'] = self._get_mean(df)
    data['quantile'] = self._get_quantile(df)
    return data

```

```

@staticmethod
def _get_min(df: DataFrame) -> dict:
    min = df.min().round(3)
    _min = min.to_dict()
    k0 = list(_min.keys())[0]
    k1 = list(_min.keys())[1]
    return {
        'label': 'Мінімальне значення',
        'value': {'x_min': round(_min[k0], 3), 'y_min': round(_min[k1], 3)}
    }

```

```

@staticmethod
def _get_max(df: DataFrame) -> dict:
    max = df.max().round(3)
    _max = max.to_dict()
    k0 = list(_max.keys())[0]
    k1 = list(_max.keys())[1]
    return {
        'label': 'Максимальне значення',

```

```

        'value': {'x_max': round(_max[k0], 3), 'y_max': round(_max[k1], 3)}
    }

```

```
@staticmethod
```

```

def _get_median(df: DataFrame) -> dict:
    median = df.median().round(3)
    _median = median.to_dict()
    k1 = list(_median.keys())[1]
    return {
        'label': 'Медіана значення',
        'value': {'x_median': '-', 'y_median': round(_median[k1], 3)}
    }

```

```
@staticmethod
```

```

def _get_mean(df: DataFrame) -> dict:
    mean = df.mean()
    _mean = mean.to_dict()
    k1 = list(_mean.keys())[1]
    return {
        'label': 'Середнє значення',
        'value': {'x_mean': '-', 'y_mean': round(_mean[k1], 3)}
    }

```

```
@staticmethod
```

```

def _get_quantile(df: DataFrame) -> dict:
    headers = df.columns.tolist()
    y_rozmah = df[headers[1]].max() - math.fabs(df[headers[1]].min())

    return {
        'label': 'Розмах',
        'value': {'x_quantile': '-', 'y_quantile': round(y_rozmah, 3)}
    }

```

```
@staticmethod
```

```

def _get_dispersion(df: DataFrame) -> dict:
    dispersion = df.var()
    _dispersion = dispersion.to_dict()
    k1 = list(_dispersion.keys())[1]
    return {
        'label': 'Дисперсія',

```

```

        'value': {'x_dispersion': '-', 'y_dispersion': round(_dispersion[k1], 3)}
    }

```

```

@staticmethod
def _get_std(df: DataFrame) -> dict:
    std = df.std()
    _std = std.to_dict()
    k1 = list(_std.keys())[1]
    return {
        'label': 'Середньоквадратичне відхилення',
        'value': {'x_std': '-', 'y_std': round(_std[k1], 3)}
    }

```

```

@staticmethod
def _get_mathematical_expectation(df: DataFrame) -> dict:
    headers = df.columns.tolist()
    val2 = (df[headers[1]] * df[headers[0]]).sum() / df[headers[0]].sum()

    return {
        'label': 'Математичне сподівання',
        'value': {'x_mathematical_expectation': '-',
        'y_mathematical_expectation': round(val2, 3)}
    }

```

```

@staticmethod
def _get_kilkist_vidlikiv(df: DataFrame) -> float:
    X_header_name = df.columns.tolist()[0]
    return len(df[X_header_name].to_list())

```

```

def _get_chastota_descritiacii(self, df: DataFrame) -> float:
    return 1 / self._get_period_descritiacii(df)

```

```

@staticmethod
def _get_period_descritiacii(df: DataFrame) -> float:
    X_header_name = df.columns.tolist()[0]
    x = df[X_header_name].to_list()
    return math.fabs(x[1] - x[0])

```

```

def _get_fft_data(self, df: DataFrame) -> dict:

```

```

Y_header_name = df.columns.tolist()[1]
y = df[Y_header_name].to_list()
fd = self._get_chastota_descritiacii(df)
N = len(y)
yf = fftshift(np.abs(fft(y)))/N
start = -fd/2
end = fd/2
step = fd/N
xf = np.arange(start, end, step)

return DataFrame({'y': list(yf), 'x': list(xf.round(0))}).to_dict('list')

def _get_periodogram_data_by_widnow(self, df: DataFrame, window:
Union[str, List[float]]) -> dict:
    Y_header_name = df.columns.tolist()[1]
    y = df[Y_header_name].to_list()
    fd = self._get_chastota_descritiacii(df)
    xp, yp = periodogram(y, fd, window)
    xp = xp.round(5)
    return DataFrame({'y': list(yp), 'x': list(xp)}).to_dict('list')

def _get_periodograms_without_signal(self, df: DataFrame) -> dict:
    Y_header_name = df.columns.tolist()[1]
    y = df[Y_header_name].to_list()
    N = len(y)
    return {
        'periodogram': list(boxcar(N)),
        'triangle_periodogram': list(triang(N)),
        'hann_periodogram': list(hann(N)),
        'blackman_periodogram': list(blackman(N)),
        'hamming_periodogram': list(hamming(N)),
        'bartlett_periodogram': list(bartlett(N)),
        'flattop_periodogram': list(flattop(N)),
        'parzen_periodogram': list(parzen(N)),
        'bohman_periodogram': list(bohman(N)),
        'blackmanharris_periodogram': list(blackmanharris(N)),
        'nuttall_periodogram': list(nuttall(N)),
        'barthann_periodogram': list(barthann(N)),
        'cosine_periodogram': list(cosine(N)),
        'exponential_periodogram': list(exponential(N)),
    }

```

```

        'tukey_periodogram': list(tukey(N)),
        'taylor_periodogram': list(taylor(N)),
    }

    def _get_periodograms_without_signal_frequency(self,
periodograms_without_signal: dict) -> dict:

        return {

                                'periodogram':
self._get_periodogram_freq(periodograms_without_signal['periodogram']),
                                'triangle_periodogram':
self._get_periodogram_freq(periodograms_without_signal['triangle_periodogram']),
                                'hann_periodogram':
self._get_periodogram_freq(periodograms_without_signal['hann_periodogram']),
                                'blackman_periodogram':
self._get_periodogram_freq(periodograms_without_signal['blackman_periodogram']),
                                'hamming_periodogram':
self._get_periodogram_freq(periodograms_without_signal['hamming_periodogram']),
                                'bartlett_periodogram':
self._get_periodogram_freq(periodograms_without_signal['bartlett_periodogram']),
                                'flattop_periodogram':
self._get_periodogram_freq(periodograms_without_signal['flattop_periodogram']),
                                'parzen_periodogram':
self._get_periodogram_freq(periodograms_without_signal['parzen_periodogram']),
                                'bohman_periodogram':
self._get_periodogram_freq(periodograms_without_signal['bohman_periodogram']),
                                'blackmanharris_periodogram':
self._get_periodogram_freq(periodograms_without_signal['blackmanharris_periodogra
m']),
                                'nuttall_periodogram':
self._get_periodogram_freq(periodograms_without_signal['nuttall_periodogram']),
                                'barthann_periodogram':
self._get_periodogram_freq(periodograms_without_signal['barthann_periodogram']),
                                'cosine_periodogram':
self._get_periodogram_freq(periodograms_without_signal['cosine_periodogram']),
                                'exponential_periodogram':
self._get_periodogram_freq(periodograms_without_signal['exponential_periodogram']
),
                                'tukey_periodogram':
self._get_periodogram_freq(periodograms_without_signal['tukey_periodogram']),

```

```

                                'taylor_periodogram':
self._get_periodogram_freq(periodograms_without_signal['taylor_periodogram']),
    }

```

```

def _get_periodogram_freq(self, window: np.arange) -> dict:
    A = fft(window, 512) / (len(window)/2.0)
    freq = np.linspace(-0.5, 0.5, len(A))
    response = np.abs(fftshift(A / abs(A).max()))
    response = 20 * np.log10(np.maximum(response, 1e-10))
    return {
        'freq': list(freq),
        'response': list(response)
    }

```

```

class DownloadAnalyticFilesForm(Form):

```

```

    class FileType:
        TXT = 'txt'
        CSV = 'csv'
        XLS = 'xls'
        XLSX = 'xlsx'

```

```

    FILE_TYPE_CHOICE = (
        (FileType.TXT, 'txt'),
        (FileType.CSV, 'csv'),
        (FileType.XLS, 'xls'),
        (FileType.XLSX, 'xlsx')
    )

```

```

        file_type = ChoiceField(choices=FILE_TYPE_CHOICE, required=True,
label='Тип файла')

```

```

        context_data_field = JSONField(widget = HiddenInput(), required = True)

```

```

Файл custom_range.py:

```

```

import numpy as np

```

```

def cust_range(*args, rtol=1e-10, atol=1e-10, include=[True, False]):
    """

```

```

    Combines numpy.arange and numpy.isclose to mimic
    open, half-open and closed intervals.

```

Avoids also floating point rounding errors as with

```
>>> numpy.arange(1, 1.3, 0.1)
array([1. , 1.1, 1.2, 1.3])
```

```
args: [start, ]stop, [step, ]
      as in numpy.arange
rtol, atol: floats
      floating point tolerance as in numpy.isclose
include: boolean list-like, length 2
      if start and end point are included
"""
# process arguments
if len(args) == 1:
    start = 0
    stop = args[0]
    step = 1
elif len(args) == 2:
    start, stop = args
    step = 1
else:
    assert len(args) == 3
    start, stop, step = tuple(args)

# determine number of segments
n = (stop-start)/step + 1

# do rounding for n
if np.isclose(n, np.round(n), rtol=rtol, atol=atol):
    n = np.round(n)

# correct for start/end is excluded
if not include[0]:
    n -= 1
    start += step
if not include[1]:
    n -= 1
    stop -= step

return np.linspace(start, stop, int(n))
```

```

def crange(*args, **kwargs):
    return cust_range(*args, **kwargs, include=[True, True])
Файл mixins.py:

import json
from django.http import HttpResponse
from utils.base_forms import DownloadAnalyticFilesForm
from utils.parser_files import convertor_df_to_file
from pandas import DataFrame

class ExportFileViewMixin:
    export_form = DownloadAnalyticFilesForm

    CONTENT_TYPE = {
        'txt': 'text/txt; charset=UTF-8',
        'csv': 'text/csv',
        'xls': 'application/ms-excel',
        'xlsx': 'application/ms-excel',
    }

    def export_file(self, file_type, context_data):
        df = DataFrame(json.loads(context_data) or None)
        return convertor_df_to_file.convert(df, file_type)

    def post(self, request, *args, **kwargs):
        """
        Handle POST requests: instantiate a form instance with the passed
        POST variables and then check if it's valid.
        """
        if request.GET.get('export'):
            file_type = request.POST['file_type']
            io_file = self.export_file(file_type, request.POST['context_data_field'])
            response = HttpResponse(io_file
, content_type=self.CONTENT_TYPE[file_type])
            response['Content-Disposition'] = f'attachment;
filename=export_file.{file_type}'
            return response
        form = self.get_form()
        if form.is_valid():
            return self.form_valid(form)

```



```

else:
    return self.form_invalid(form)

```

Файл parser_files.py:

```

from io import BytesIO
from typing import Any, Callable, Optional, IO
from pandas import DataFrame, ExcelWriter, read_excel, read_csv

```

```

class ConvertFileDataToDataFrame:

```

```

    """

```

```

        This class represent fabric of methods
        of the parcing files which extensions
        txt, csv, xls, xlsx
    """

```

```

    """

```

```

        def convert(self, file: IO, filename:str, delimetr: Optional[str]=None) ->
DataFrame:

```

```

        convertor = self._get_convert_method(filename)
        df = convertor(file, delimetr)
        return df

```

```

        def _get_convert_method(self, filename: str) -> Callable:

```

```

            convertors = {
                'txt': self._txt_convertor,
                'csv': self._csv_convertor,
                'xls': self._xls_convertor,
                'xlsx': self._xls_convertor
            }
            return convertors.get(filename.split('.')[1])

```

```

        def _xls_convertor(self, file: IO, delimetr: str) -> DataFrame:

```

```

            return read_excel(file)

```

```

        def _csv_convertor(self, file: IO, delimetr: str) -> DataFrame:

```

```

            return read_csv(file, sep=',')

```

```

        def _txt_convertor(self, file: IO, delimetr: str) -> DataFrame:

```

```

            return read_csv(file, delimiter=delimetr)

```

```

class ConvertDataFrameToFileData:
    """
        This class represent fabric of methods
        of the parcing dataframe to file which extensions
        txt, csv, xls, xlsx
    """

    def _get_convert_method(self, type_to_convert: str) -> Callable:
        convertors = {
            'txt': self._txt_convertor,
            'csv': self._csv_convertor,
            'xls': self._xls_convertor,
            'xlsx': self._xlsx_convertor
        }
        return convertors.get(type_to_convert)

    def _xls_convertor(self, df: DataFrame) -> IO:
        io = BytesIO()
        writer = ExcelWriter(io)
        df.to_excel(writer, 'Дані', index=False)
        writer.save()
        return io.getvalue()

    def _csv_convertor(self, df: DataFrame) -> IO:
        io = BytesIO()
        df.to_csv(path_or_buf=io)
        return io.getvalue()

    def _txt_convertor(self, df: DataFrame) -> IO:
        return self._csv_convertor(df)

    def convert(self, df: DataFrame, type_to_convert: str) -> IO:
        convertor = self._get_convert_method(type_to_convert)
        return convertor(df)

convertor_file_to_df = ConvertFileDataToDataFrame()
convertor_df_to_file = ConvertDataFrameToFileData()
Файл base.html :

```

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>{{ title|default:'Автоматизована система обробки сигналів' }} |
Vlad Shelemakha</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css"
integrity="sha384-xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLM
wNLD69Npy4HI+N" crossorigin="anonymous"> </head>
  <link
href="https://cdnjs.cloudflare.com/ajax/libs/ion-rangeslider/2.1.4/css/ion.rangeSlider.s
kinFlat.css" rel="stylesheet" />
  <link
href="https://cdnjs.cloudflare.com/ajax/libs/ion-rangeslider/2.1.4/css/ion.rangeSlider.m
in.css" rel="stylesheet" />
  <body>
    <script
src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbV
YUew+OrCXaRkfj" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8
D6OP9dO5Vg3Q9ct" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbV
YUew+OrCXaRkfj" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-9/reFTGAW83EW2RDu2S0VKA1zap3H66lZH81PoYlFhbGU+6BZ
p6G7niu735Sk7lN" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.min.js"
integrity="sha384-+sLIOodYLS7CIrQpBjl+C7nPvqq+FbNUBDunl/OZv93DB7Ln/53
3i8e/mZXLi/P+" crossorigin="anonymous"></script>

```

```

    <link
href="https://cdn.jsdelivr.net/npm/bootstrap-select@1.13.14/dist/css/bootstrap-select.min.css">

```

```

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap-select@1.13.14/dist/js/bootstrap-select.min.js"></script>

```

```

    <script
src="//cdnjs.cloudflare.com/ajax/libs/numeral.js/2.0.6/numeral.min.js"></script>

```

```

    <script    src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KChRr/rE9/Qpg6aAZGJwFDMVN
A/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

```

```

    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fak
FPskvXusvfa0b4Q" crossorigin="anonymous"></script>

```

```

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyj
Uar5+76PVCmYl" crossorigin="anonymous"></script>

```

```

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>

```

```

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/ion-rangeslider/2.1.4/js/ion.rangeSlider.min.js"></script>

```

```

    <script src="https://cdn.plot.ly/plotly-2.16.1.min.js"></script>

```

```

    {% block script %}

```

```

    {% endblock script %}

```

```

    {% block main %}

```

```

    {% endblock main %}

```

```

</body>

```

```

</html>

```

```

Файл main.html:

```

```

{% extends "base.html" %}

{% block script %}
<script type="text/javascript">
    $(window).on('load', function() {
        if (sessionStorage.getItem("loaded") === null) {
            sessionStorage.setItem("loaded", "loaded");
            $('#infoModal').modal('show');
        }
    });
</script>

{% endblock %}

{% block main %}

<div class="modal show" id="infoModal" tabindex="-1"
aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-dialog-scrollable">
        <div class="modal-content">
            <div class="modal-body">
                <div class="container-fluid">
                    <div class="row">
                        <button type="button" class="close offset-11 col-1"
data-dismiss="modal" aria-label="Close">
                            <span aria-hidden="true">&times;</span>
                        </button>
                        <p class="text-justify">
                            Інформаційне модальне вікно, яке буде заповнено після
сформування кінцевого
                            результату програми. Наразі цей текст написаний для того щоб,
людина
                            яка подивиться цей сайт зрозуміла нащо я потрібен. Бо всі
думають, що
                            мене потрібно видалити!
                        </p>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </div>
    </div>
</div>
</div>
</div>

<div class="d-flex justify-content-center">
    <div class="col-12 d-flex justify-content-center flex-column text-center">
        <p class="h1 text-center">Автоматизована система обробки
сигналів</p>
        <p class="text-justify text-center">Шановний користувач, Ви
знаходитеся на початковому екрані Автоматизованої системи</p>
    </div>
</div>

<div class="d-flex align-items-center justify-content-center">
    <div class="col-3 card" style="width: 13rem;">
        
        <div class="card-body">
            <p class="h4 text-center card-text">
                Первина оцінка сигналу зі встановлення файлу
            </p>
            <p class="card-text">
                Сторінка по обробці детермінових та стохастичних сигналів, які
беруться з завнатаженого файлу.
            </p>
            <a href="{% url 'time-analysis' %}" class="btn btn-primary
stretched-link">Перейти</a>
        </div>
    </div>

    <div class="col-3 card ml-5" style="width: 13rem;">
        
        <div class="card-body">
            <p class="h4 text-center card-text">

```

```

        Генерація тестових коливальних сигналів
    </p>
    <p class="card-text">
        Сторінка по генерації тестових коливальних сигналів.
    </p>
    <a href="{% url 'custom-time-analysis' %}" class="btn btn-primary
stretched-link">Перейти</a>
</div>
</div>
</div>
{% endblock main %}
Файл analytic_page.html:

{% extends "base.html" %}
{% load crispy_forms_tags %}

{% block main %}

    <div class="col-12 ml-md-4 mt-2">
        <a class="btn btn-outline-primary" href="{% url 'main-page' %}"
role="button"><- Головна сторінка</a>
    </div>
    <div class="vh-100 d-flex justify-content-around my-3">
        <div class="row m-0 p-0 col-md-3 col-12">
            <div class="border p-5 col-12">
                {% block form %}

                    <form method="post" id="data-form"
enctype="multipart/form-data" action="{% url request.resolver_match.url_name %}>
                        {% if request.resolver_match.url_name == "time-analysis" %}
                            {% include 'analytic_fields.html' with form=form
button_name='Обробити дані' %}
                        {% else %}
                            {% include 'analytic_fields_with_range.html' with
form=form button_name='Обробити дані' %}
                        {% endif %}
                    </form>
                {% endblock form %}

            </div>
        </div>
    </div>

    {% if data %}

```

```

<table class="table table-bordered my-5">
  <thead>
    <tr>

      <th scope="col">#</th>
      {% for column_name in data.headers %}
        <th scope="col">{{column_name}} {% if forloop.counter
== 1 %} (с) {% else %} (ГП) {% endif %}</th>
      {% endfor %}
    </tr>
  </thead>
  <tbody>
    {% for key, values in data.calculated_data.analytics_data.items
%}

    <tr id="{{key}}">
      <th scope="row">{{values.label}}</th>
      {% for k, v in values.value.items %}
        <td class="text-center" id="{{k}}">{{v}}</td>
      {% endfor %}
    </tr>
    {% endfor %}

    <tr id="dispersion" class="{% if 'dispersion' not in
data.calculated_data.analytics_data %} d-none {% endif %}">
      <th scope="row">Дисперсія</th>
      <td class="text-center" id="x_dispersion">-</td>
      <td class="text-center" id="y_dispersion">-</td>
    </tr>

    <tr id="std" class="{% if 'std' not in
data.calculated_data.analytics_data %} d-none {% endif %}">
      <th scope="row">Середньоквадратичне відхилення</th>
      <td class="text-center" id="x_std">-</td>
      <td class="text-center" id="y_std">-</td>
    </tr>

    <tr id="mathematical_expectation" class="{% if
'mathematical_expectation' not in data.calculated_data.analytics_data %} d-none {%
endif %}">
      <th scope="row">Математичне сподівання</th>
      <td class="text-center"
id="x_mathematical_expectation">-</td>

```



```

id="y_mathematical_expectation">-</td>
</tr>

<tr>
<th scope="row">Період дискретизація</th>
<td colspan="2" id="period_descritiacii"> {{
data.calculated_data.graphs_data.period_descritiacii|floatformat:5 }} (с) </td>
</tr>
<tr>
<th scope="row">Частота дискретизація</th>
<td colspan="2" id="chastota_descritiacii"> {{
data.calculated_data.graphs_data.chastota_descritiacii|floatformat:2 }} (Гц) </td>
</tr>
<tr>
<th scope="row">Кількість відліків</th>
<td colspan="2" id="kilkist_vidlikiv"> {{
data.calculated_data.graphs_data.kilkist_vidlikiv|floatformat:2 }} (б/п)</td>
</tr>
</tbody>
</table>
{% endif %}
</div>
</div>

<div class="col-md-8 col-12 m-0 p-0">
{% block analytics %}
{% include "analytic_data.html" with data=data %}
{% endblock analytics %}
</div>
</div>
{% endblock main %}

{% block script %}

{% endblock script%}
Файл analytic_fields_with_range.html:
{% load crispy_forms_tags %}
{% csrf_token %}

```

```

<div class="row">
  <div class="col-12">
    {{ form.type_of_signal|as_crispy_field }}
  </div>
</div>

<div class="form-group row">
  <label class="col-sm-4 col-form-label" for="{{ form.snr.auto_id }}">{{
form.snr.label }}</label>
  <div class="col-sm-8">
    <input type="number" name="{{ form.snr.name }}"
value="{{ form.snr.value }}" step="any" class="numberinput form-control" required=""
id="{{ form.snr.auto_id }}">
  </div>

</div>

<div class="row">
  <div class="col-12">

    <label for="{{ form.noises.auto_id }}" class="col-12 px-0 mx-0">
      {{ form.noises.label }}
    </label>
    <select name="{{ form.noises.name }}" id="{{ form.noises.auto_id }}"
class="selectpicker col-12 px-0 mx-0 mb-2" multiple data-live-search="true">
      <option value="white">Білий шум</option>
      <option value="pink">Рожевий шум</option>
      <option value="blue">Синій шум</option>
      <option value="brown">Коричневий шум</option>
      <option value="violet">Фіолетовий шум</option>
    </select>
  </div>
</div>

<div class="form-group row">
  <label class="col-sm-4 col-form-label" for="{{ form.mean.auto_id }}">{{
form.mean.label }}</label>
  <div class="col-sm-8">
    <input type="number" name="{{ form.mean.name }}"
value="{{ form.mean.value }}" step="any" class="numberinput form-control"
required="" id="{{ form.mean.auto_id }}">
  </div>

```

```
</div>
```

```
</div>
```

```
<div class="form-group row">
  <label class="col-sm-4 col-form-label" for="{{ form.scope.auto_id }}">{{
form.scope.label }}</label>
  <div class="col-sm-8">
    <input type="number" name="{{form.scope.name}}"
value="{{form.scope.value}}" step="any" class="numberinput form-control"
required="" id="{{form.scope.auto_id}}">
  </div>
</div>
```

```
<div class="form-check">
  <input class="form-check-input" type="checkbox"
name="{{form.checker_count_of_dot_or_period_sampling.name}}" id="{{
form.checker_count_of_dot_or_period_sampling.auto_id }}" {% if
form.checker_count_of_dot_or_period_sampling.value %} checked {% endif %}>
  <label class="form-check-label" for="{{
form.checker_count_of_dot_or_period_sampling.auto_id }}" >
    {{form.checker_count_of_dot_or_period_sampling.label}}
  </label>
</div>
```

```
<div class="form-group row">
  <label class="col-sm-4 col-form-label" for="{{
form.count_of_periods.auto_id }}">{{ form.count_of_periods.label }}</label>
  <div class="col-sm-8">
    <input type="number" min=2 name="{{form.count_of_periods.name}}"
value="{{form.count_of_periods.value}}" step="any" class="numberinput
form-control" required="" {% if not
form.checker_count_of_dot_or_period_sampling.value %} readonly {% endif %}
id="{{form.count_of_periods.auto_id}}">
  </div>
  <div class="col-sm-12">
    <input type="range" value="{{form.count_of_periods.value}}"
class="slider" id="count_of_periods_range">
  </div>
</div>
```

```

<div class="form-group row">
  <label class="col-sm-4 col-form-label" for="{{ form.count_of_dots.auto_id
  }}">{{ form.count_of_dots.label }}</label>
  <div class="col-sm-8">
    <input type="number" min=2 name="{{form.count_of_dots.name}}"
    value="{{form.count_of_dots.value}}" step="any" class="numberinput form-control"
    {% if form.checker_count_of_dot_or_period_sampling.value %} readonly {% endif
    %} required="" id="{{form.count_of_dots.auto_id}}">
  </div>
  <div class="col-sm-12">
    <input type="range" value="{{form.count_of_dots.value}}"
    class="slider" id="count_of_dots_range">
  </div>
</div>

<div class="form-group row">
  <label class="col-sm-4 col-form-label" for="{{
  form.frequency_sampling.auto_id }}">{{ form.frequency_sampling.label }}</label>
  <div class="col-sm-8">
    <input type="number" name="{{form.frequency_sampling.name}}"
    value="{{form.frequency_sampling.value}}" step="any" class="numberinput
    form-control" required="" id="{{form.frequency_sampling.auto_id}}">
  </div>

  <div class="col-sm-12">
    <input type="range" value="{{form.frequency_sampling.value}}"
    class="slider" id="frequency_sampling_range">
  </div>

</div>

<div class="form-group row">
  <label class="col-sm-4 col-form-label" for="{{
  form.period_sampling.auto_id }}">{{ form.period_sampling.label }}</label>
  <div class="col-sm-8">
    <input type="number" name="{{form.period_sampling.name}}"
    value="{{form.period_sampling.value}}" step="any" class="numberinput
    form-control" required="" id="{{form.period_sampling.auto_id}}">
  </div>

```

```

        <div class="col-sm-12">
            <input type="range" value="{{form.period_sampling.value}}"
class="slider" id="period_sampling_range">
        </div>

</div>

<div class="form-group row">
    <label class="col-sm-4 col-form-label" for="{{ form.frequency.auto_id
}}">{{ form.frequency.label }}</label>
    <div class="col-sm-8">
        <input type="number" name="{{form.frequency.name}}"
value="{{form.frequency.value}}" step="any" class="numberinput form-control"
required="" id="{{form.frequency.auto_id}}">
    </div>

    <div class="col-sm-12">
        <input type="range" value="{{form.frequency.value}}" class="slider"
id="frequency_range">
    </div>
</div>

{% if form.errors %}
    {% for messages in form.errors.values %}
        {% for message in messages %}
            <div class="alert alert-danger">
                {{message}}
            </div>
        {% endfor %}
    {% endfor %}
{% endif %}

    <button type="submit" class="btn btn-success d-none" id="butt">{{
button_name }}</button>

<script>

    var time_reload = null;

```

```

let timerId = setInterval(
  () => {
    if(
      time_reload && Date.now() - time_reload > 1500
    ){

      $("#butt").click();
      time_reload = null;
    }
  },
  350);

```

```

const freq_range_updater = (N) => {
  var slider = $("#frequency_sampling_range").data("ionRangeSlider");

  if(N < 0){
    N = 0;
    $('#id_frequency_sampling').val(N);
  }
  update_data = {
    from: N,
  }
  slider.update(update_data)
};

```

```

const f_range_updater = (N) => {
  var slider = $("#frequency_range").data("ionRangeSlider");
  update_data = {
    from: N
  }
  slider.update(update_data)
};

```

```

const period_range_updater = (N) => {
  var slider = $("#period_sampling_range").data("ionRangeSlider");
  update_data = {
    from: N
  }

```

```

    }
    slider.update(update_data)
  }

$("#id_count_of_periods").on('input', (e) => {
  time_reload = Date.now();
});

$("#id_snr").on('input', (e) => {
  time_reload = Date.now();
});

$("#id_scope").on('input', (e) => {
  time_reload = Date.now();
});

$("#id_mean").on('input', (e) => {
  time_reload = Date.now();
});

$("#id_type_of_signal").change((e) => {
  time_reload = Date.now();
});

$("#id_noises").change((e) => {
  time_reload = Date.now();
});

$( "#id_frequency_sampling" ).on('input', (e) => {
  var fd = $('#id_frequency_sampling').val();
  if(fd < 0){
    fd = 0;
    $('#id_frequency_sampling').val(fd);
  }
  $("#id_period_sampling").val(1/fd);
  $("#id_frequency").val(parseInt(fd/255));
  freq_range_updater(fd);
  period_range_updater(1/fd);
  f_range_updater(parseInt(fd/255));
  time_reload = Date.now();
});

```

```
});
```

```
$( "#id_period_sampling" ).on('input', (e) => {
  var td = $('#id_period_sampling').val();
  if(td < 0){
    td = 0;
    $('#id_period_sampling').val(td);
  }
  $("#id_frequency_sampling").val(1/td);
  $("#id_frequency").val(parseInt((1/td)/25));
  freq_range_updater(1/td);
  period_range_updater(td);
  f_range_updater(parseInt((1/td)/25));
  time_reload = Date.now();
});
```

```
$( "#id_frequency" ).on('input', (e) => {
  var f = $('#id_frequency').val();
  if(f < 0){
    f = 0;
    $('#id_frequency').val(f);
  }
  var fd = $('#id_frequency_sampling').val();
  $("#id_frequency_sampling").val(f*25);
  freq_range_updater(f*25);
  period_range_updater(1/(f*25));
  $("#id_period_sampling").val(1/(f*25));
```

```
  f_range_updater(f);
  time_reload = Date.now();
});
```

```
$( "#id_count_of_periods" ).on('input', (e) => {
  var N = $('#id_count_of_periods').val();
  var slider = $("#count_of_periods_range").data("ionRangeSlider");

  if(N < 1){
    N = 1;
    $('#id_count_of_periods').val(N);
```



```

    }

    update_data = {
        from: N,
    }
    slider.update(update_data);
    time_reload = Date.now();
});

$("#id_checker_count_of_dot_or_period_sampling").click( (e) => {
    if($("#id_checker_count_of_dot_or_period_sampling").is(':checked')){
        $( "#id_count_of_dots" ).prop( "readonly", true );
        var slider = $("#count_of_dots_range").data("ionRangeSlider");
        update_data = {
            to_fixed:true,
            from_fixed:true,
        }
        slider.update(update_data)
        $( "#id_count_of_periods" ).prop( "readonly", false );
        var slider = $("#count_of_periods_range").data("ionRangeSlider");
        update_data = {
            to_fixed:false,
            from_fixed:false,
        }
        slider.update(update_data)
    }else{
        $( "#id_count_of_periods" ).prop( "readonly", true );
        var slider = $("#count_of_periods_range").data("ionRangeSlider");
        update_data = {
            to_fixed:true,
            from_fixed:true,
        }
        slider.update(update_data)

        $( "#id_count_of_dots" ).prop( "readonly", false );
        var slider = $("#count_of_dots_range").data("ionRangeSlider");
        update_data = {
            to_fixed:false,
            from_fixed:false,
        }
    }

```

```

    }
    slider.update(update_data)
  }
} )

```

```

$( "#id_count_of_dots" ).on('input', (e) => {
  var N = $('#id_count_of_dots').val();
  var slider = $("#count_of_dots_range").data("ionRangeSlider");

```

```

  if(N < 1){
    N = 1;
    $('#id_count_of_dots').val(N);
  }

```

```

  update_data = {
    from: N,
  }
  slider.update(update_data);
  time_reload = Date.now();
});

```

```

$("#count_of_periods_range").ionRangeSlider({
  grid: true,
  skin: "big",
  type: "single",
  from: `{{form.count_of_periods.value}}`,
  min: 2,
  max: 200,
  {% if not form.checker_count_of_dot_or_period_sampling.value %}
  to_fixed:true,
  from_fixed:true,
  {% endif %}
  onChange: function(data) {
    if($("#id_checker_count_of_dot_or_period_sampling").is(':checked')){
      $("#id_count_of_periods").val(data.from);
      time_reload = Date.now();
    }
  }
});

```

```

$("#count_of_dots_range").ionRangeSlider({
  grid: true,
  skin: "big",
  type: "single",
  from: `{{form.count_of_dots.value}}`,
  min: 2,
  max: 32768,
  {% if form.checker_count_of_dot_or_period_sampling.value %}
  to_fixed:true,
  from_fixed:true,
  {% endif %}
  onChange: function(data) {

if(!$("#id_checker_count_of_dot_or_period_sampling").is(':checked')){
    $("#id_count_of_dots").val(data.from);
    time_reloud = Date.now();
  }
}
});

```

```

$("#frequency_sampling_range").ionRangeSlider({
  grid: true,
  skin: "big",
  type: "single",
  from: `{{form.frequency_sampling.value}}`,
  min: 0,
  max: 10000,
  onChange: function(data) {
    $("#id_frequency_sampling").val(data.from);
    $("#id_period_sampling").val(1/data.from);
    $("#id_frequency").val(parseInt(data.from/25));

    period_range_updater(1/data.from);
    f_range_updater(parseInt(data.from/25));
    time_reloud = Date.now();
  }
});

```

```

$("#period_sampling_range").ionRangeSlider({
  grid: true,
  skin: "big",
  type: "single",
  from: `{{form.period_sampling.value}}`,
  min: 0,
  step: 0.000001,
  max: 1,
  onChange: function(data) {
    $("#id_period_sampling").val(data.from);
    $("#id_frequency_sampling").val(1/data.from);
    $("#id_frequency").val(parseInt((1/data.from)/25));

    freq_range_updater(1/data.from);
    f_range_updater(parseInt((1/data.from)/25));
    time_reload = Date.now();
  }
});

```

```

$("#frequency_range").ionRangeSlider({
  grid: true,
  skin: "big",
  type: "single",
  from: `{{form.frequency.value}}`,
  min: 0,
  max: 400,
  onChange: function(data) {
    $("#id_frequency").val(data.from);

    var fd = $('#id_frequency_sampling').val();
    $("#id_frequency_sampling").val(data.from*25);
    freq_range_updater(data.from*25);
    period_range_updater(1/(data.from*25));
    $("#id_period_sampling").val(1/(data.from*25));
    time_reload = Date.now();
  }
});

```

```
</script>
```

Файл analytic_data.html:

```
{% if data.calculated_data %}
<ul class="nav nav-tabs" id="myTab" role="tablist">
  <li class="nav-item">
    <a class="nav-link active" id="graph_time-tab" data-toggle="tab"
href="#graph_time" role="tab" aria-controls="graph_time" aria-selected="false">
      Графік часового аналізу
    </a>
  </li>
  <li class="nav-item">
    <a class="nav-link" id="analysis-tab" data-toggle="tab" href="#analysis"
role="tab" aria-controls="analysis" aria-selected="false">Експорт даних</a>
  </li>
</ul>
<div class="border tab-content" id="myTabContent">
  <div class="tab-pane {% if data.calculated_data == None %} fade {% else
%} active {% endif %} p-5" id="graph_time" role="tabpanel"
aria-labelledby="graph_time-tab">

    <div id="main-graph" class="row p-0 m-0">
      <div id="graph-plt" class="col-12 m-0 p-0"></div>
    </div>

    <div class="row">

      <div class="col-3 p-0">
        <div class="list-group" id="list-tab" role="tablist">

          <div class="btn-group">
            <button type="button" class="btn btn-outline-primary
dropdown-toggle text-left" data-toggle="dropdown" aria-haspopup="true"
aria-expanded="false">
              Частотний аналіз
            </button>
            <div class="dropdown-menu">
              <a class="dropdown-item freq-graph active" id="list-fft-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Швидке перетворення Фер'є</a>
```

```

        <a class="dropdown-item freq-graph" id="list-periodograma-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Періодograma</a>
        <a class="dropdown-item freq-graph" id="list-triangle-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Трикутна періодograma</a>
        <a class="dropdown-item freq-graph" id="list-hann-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Періодograma Ханна</a>
        <a class="dropdown-item freq-graph" id="list-blackman-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Періодograma Blackman</a>
        <a class="dropdown-item freq-graph" id="list-hamming-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Періодograma Hamming</a>
        <a class="dropdown-item freq-graph" id="list-bartlett-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Періодograma Barlett</a>
        <a class="dropdown-item freq-graph" id="list-flattop-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Періодograma Flattop</a>
        <a class="dropdown-item freq-graph" id="list-parzen-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Періодograma Parzen</a>
        <a class="dropdown-item freq-graph" id="list-bohman-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Періодograma bohman</a>
        <a class="dropdown-item freq-graph" id="list-blackmanharris-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Періодograma Blackmanharris</a>
        <a class="dropdown-item freq-graph" id="list-nuttall-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Періодograma Nuttal</a>
        <a class="dropdown-item freq-graph" id="list-barthann-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Періодograma Barthann</a>
        <a class="dropdown-item freq-graph" id="list-cosine-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Періодograma Cosine</a>

```

```

        <a class="dropdown-item freq-graph" id="list-exponential-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Експотенціальна Періодограма</a>
        <a class="dropdown-item freq-graph" id="list-tukey-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Періодограма Tukey</a>
        <a class="dropdown-item freq-graph" id="list-taylor-list"
data-toggle="list" href="#list-periodograma" role="tab"
aria-controls="periodograma">Періодограма Тайлера</a>
    </div>
</div>

<div class="col-12 d-none p-0 mt-3" id="periodograma_div">
    <div id="only-periodograma-graph-plt"
style="height:250px;width:400px;" class="col-12 m-0 p-0"></div>
    <div id="only-periodograma-frequency-graph-plt"
style="height:250px;width:400px;" class="col-12 m-0 p-0"></div>
</div>
</div>
<div class="col-9">
    <div class="tab-content" id="nav-tabContent">
        <div class="tab-pane fade show active" id="list-periodograma"
role="tabpanel" aria-labelledby="list-periodograma-list">
            <div id="periodogram-graph-plt" style="height: 439px; width:
878.5px;"></div>
        </div>
    </div>
</div>
</div>
<div class="tab-pane fade" id="analysis" role="tabpanel"
aria-labelledby="analysis-tab">
    <form method="post" class="col-4 m-5" action={% url 'time-analysis'
%}?export=true>
        {% include 'analytic_fields.html' with form=export_file_form
button_name='Експорт даних' %}
    </form>
</div>

</div>

```

```

{% endif %}
<script>

{% if request.resolver_match.url_name == 'custom-time-analysis' %}

$("#data-form").submit(function (e) {
    e.preventDefault();

    window.serializedData = $(this).serialize();
    $.ajax({
        type: 'POST',
        url: "{% url 'custom-time-analysis' %}",
        data: serializedData,
        success: function (response) {
            data_for_graph = response.data_for_graph;
            data = response.calculated_data;

            if(Object.keys(response.calculated_data).includes('noise_data')){
                $('#graph-plt').removeClass('col-12');
                $('#graph-plt').addClass('col-9');
                $( "#noise_data" ).remove();
                $('#main-graph').prepend(`
<div class="col-3 m-0 p-0" id="noise_data">
    <div style="height:250px;width:400px;" id="noise-graph-plt"
class="col-12 m-0 p-0">
</div>
<table class="table table-bordered col-12 my-2" style="font-size:
12px">

    <thead>
    <tr>

        <th scope="col">#</th>
        <th class="px-0">Амплітуда(Гц)</th>
    </tr>
    </thead>
    <tbody>
    <tr id="noise_snr">
        <th scope="row">SNR</th>

        <td class="text-center
px-0">${response.calculated_data.noise_data.calculation_data.snr}</td>

```



```

</tr>
<tr id="noise_max">
  <th scope="row">Мінімальне значення</th>
  <td class="text-center
px-0">${response.calculated_data.noise_data.calculation_data.min}</td>
</tr>
<tr id="noise_min">
  <th scope="row">Максимальне значення</th>
  <td class="text-center
px-0">${response.calculated_data.noise_data.calculation_data.max}</td>
</tr>
<tr id="noise_mean">
  <th scope="row">Дисперсія</th>
  <td class="text-center
px-0">${response.calculated_data.noise_data.calculation_data.mean}</td>
</tr>
<tr id="noise_dispersion">
  <th scope="row">Дисперсія</th>
  <td class="text-center
px-0">${response.calculated_data.noise_data.calculation_data.dispersion}</td>
</tr>
<tr id="noise_std">
  <th scope="row">Середньоквадратичне відхилення</th>
  <td class="text-center px-0"
>${response.calculated_data.noise_data.calculation_data.std}</td>
</tr>
</div>`)

```

```

let noise_trace1 = {
  x: response.calculated_data.noise_data.plot_data['x'],
  y: response.calculated_data.noise_data.plot_data['y'],
  type: 'scattergl',
  mode: 'lines+markers',
  name: 'Шум',
};
let noise_data_for_graph = [noise_trace1];
let noise_layout = {
  title: 'Шумова складова',
  xaxis: {
    title: `${keys[0]} (c)`,

```

```

        showgrid: true,
        zeroline: true
    },
    yaxis: {
        title: `${keys[1]} (Γπ)`,
        showgrid: true,
        zeroline: true
    },
    showlegend: false,
};
Plotly.newPlot('noise-graph-plt', noise_data_for_graph,
noise_layout);

    }else{
        $('#graph-plt').addClass('col-12');
        $('#graph-plt').removeClass('col-9');
        $('#noise_data').remove();
    }
    set_variables(data, data_for_graph);

$('#period_descritiatcii').text(data.graphs_data.period_descritiatcii.toFixed(5));

$('#chastota_descritiatcii').text(data.graphs_data.chastota_descritiatcii.toFixed(2));

$('#kilkist_vidlikiv').text(data.graphs_data.kilkist_vidlikiv.toFixed(2));

        for (let [key, _value] of
Object.entries(response.calculated_data.analytics_data)){
            for(let [key2, _value2] of Object.entries(_value)){
                if( key2 === 'value'){
                    for(let [key3, _value3] of Object.entries(_value2)){

                        $('#${key3}`).text(_value3);
                    }
                }
            }
        }
        let ad_keys = Object.keys(response.calculated_data.analytics_data);
        for (let stochastic_value of ["mathematical_expectation", "std",
"dispersion"]){

```

```

        if (ad_keys.includes(stochastic_value))
            $('#{stochastic_value}`).removeClass('d-none');
        else
            $('#{stochastic_value}`).addClass('d-none');
        }
        console.log(response.calculated_data)

    }
    })
    {% endif %}
    data_for_graph = JSON.parse('{{data.data_for_graph|safe}}'.replace(/'/g,
    ''));

    data = JSON.parse('{{data.calculated_data|safe}}'.replace(/'/g, ''));
    var headers = JSON.parse('{{data.headers|safe}}'.replace(/'/g, ''));

    window.set_variables = (data, data_for_graph) => {
        window.data_for_fft_graph = data['graphs_data']['fft']
        window.data_for_periodogram_graph = data['graphs_data']['periodogram'];
        window.data_for_triangle_periodogram_graph =
data['graphs_data']['triangle_periodogram'];
        window.data_for_hann_periodogram_graph =
data['graphs_data']['hann_periodogram'];
        window.data_for_blackman_periodogram_graph =
data['graphs_data']['blackman_periodogram'];
        window.data_for_hamming_periodogram_graph =
data['graphs_data']['hamming_periodogram'];
        window.data_for_bartlett_periodogram_graph =
data['graphs_data']['bartlett_periodogram'];
        window.data_for_flattop_periodogram_graph =
data['graphs_data']['flattop_periodogram'];
        window.data_for_parzen_periodogram_graph =
data['graphs_data']['parzen_periodogram'];
        window.data_for_bohman_periodogram_graph =
data['graphs_data']['bohman_periodogram'];
        window.data_for_blackmanharris_periodogram_graph =
data['graphs_data']['blackmanharris_periodogram'];
        window.data_for_nuttall_periodogram_graph =
data['graphs_data']['nuttall_periodogram'];

```

```

        window.data_for_barthann_periodogram_graph      =
data['graphs_data']['barthann_periodogram'];
        window.data_for_cosine_periodogram_graph        =
data['graphs_data']['cosine_periodogram'];
        window.data_for_exponential_periodogram_graph   =
data['graphs_data']['exponential_periodogram'];
        window.data_for_tukey_periodogram_graph         =
data['graphs_data']['tukey_periodogram'];
        window.data_for_taylor_periodogram_graph        =
data['graphs_data']['taylor_periodogram'];

        window.periodogram_without_signal_periodogram   =
data['graphs_data']['periodograms_without_signal']['periodogram']
        window.triangle_periodogram                    =
data['graphs_data']['periodograms_without_signal']['triangle_periodogram']
        window.hann_periodogram                        =
data['graphs_data']['periodograms_without_signal']['hann_periodogram']
        window.blackman_periodogram                    =
data['graphs_data']['periodograms_without_signal']['blackman_periodogram']
        window.hamming_periodogram                     =
data['graphs_data']['periodograms_without_signal']['hamming_periodogram']
        window.bartlett_periodogram                    =
data['graphs_data']['periodograms_without_signal']['bartlett_periodogram']
        window.flattop_periodogram                     =
data['graphs_data']['periodograms_without_signal']['flattop_periodogram']
        window.parzen_periodogram                      =
data['graphs_data']['periodograms_without_signal']['parzen_periodogram']
        window.bohman_periodogram                      =
data['graphs_data']['periodograms_without_signal']['bohman_periodogram']
        window.blackmanharris_periodogram              =
data['graphs_data']['periodograms_without_signal']['blackmanharris_periodogram']
        window.nuttall_periodogram                     =
data['graphs_data']['periodograms_without_signal']['nuttall_periodogram']
        window.barthann_periodogram                    =
data['graphs_data']['periodograms_without_signal']['barthann_periodogram']
        window.cosine_periodogram                      =
data['graphs_data']['periodograms_without_signal']['cosine_periodogram']
        window.exponential_periodogram                  =
data['graphs_data']['periodograms_without_signal']['exponential_periodogram']

```

```

                                window.tukey_periodogram           =
data['graphs_data']['periodograms_without_signal']['tukey_periodogram']
                                window.taylor_periodogram          =
data['graphs_data']['periodograms_without_signal']['taylor_periodogram']
                                window.freq_periodogram_without_signal_periodogram =
data['graphs_data']['periodograms_without_signal_frequency']['periodogram']
                                window.freq_triangle_periodogram    =
data['graphs_data']['periodograms_without_signal_frequency']['triangle_periodogram']
                                window.freq_hann_periodogram        =
data['graphs_data']['periodograms_without_signal_frequency']['hann_periodogram']
                                window.freq_blackman_periodogram    =
data['graphs_data']['periodograms_without_signal_frequency']['blackman_periodogram']

                                window.freq_hamming_periodogram     =
data['graphs_data']['periodograms_without_signal_frequency']['hamming_periodogram']

                                window.freq_bartlett_periodogram     =
data['graphs_data']['periodograms_without_signal_frequency']['bartlett_periodogram']
                                window.freq_flattop_periodogram      =
data['graphs_data']['periodograms_without_signal_frequency']['flattop_periodogram']
                                window.freq_parzen_periodogram       =
data['graphs_data']['periodograms_without_signal_frequency']['parzen_periodogram']
                                window.freq_bohman_periodogram       =
data['graphs_data']['periodograms_without_signal_frequency']['bohman_periodogram']
                                window.freq_blackmanharris_periodogram =
data['graphs_data']['periodograms_without_signal_frequency']['blackmanharris_periodogram']

                                window.freq_nuttall_periodogram      =
data['graphs_data']['periodograms_without_signal_frequency']['nuttall_periodogram']
                                window.freq_barthann_periodogram     =
data['graphs_data']['periodograms_without_signal_frequency']['barthann_periodogram']

                                window.freq_cosine_periodogram       =
data['graphs_data']['periodograms_without_signal_frequency']['cosine_periodogram']
                                window.freq_exponential_periodogram  =
data['graphs_data']['periodograms_without_signal_frequency']['exponential_periodogram']

                                window.freq_tukey_periodogram        =
data['graphs_data']['periodograms_without_signal_frequency']['tukey_periodogram']

```

```

window.freq_taylor_periodogram =
data['graphs_data']['periodograms_without_signal_frequency']['taylor_periodogram']
    delete data;
    window.keys = Object.keys(data_for_graph)
    window._l = data_for_graph[keys[0]].length;
    window.trace1 = {
        x: data_for_graph[keys[0]],
        y: data_for_graph[keys[1]],
        type: 'scattergl',
        mode: 'lines+markers',
        name: 'Вхідний сигнал',
    };
    window._data_for_graph = [trace1];
    window.layout = {
        title: 'Графік вхідного сигналу у часовому просторі',
        xaxis: {
            title: `${headers[0]} (с)`,
            showgrid: true,
            zeroline: true
        },
        yaxis: {
            title: `${headers[1]} (Гц)`,
            showgrid: true,
            zeroline: true
        },
        showlegend: false,
    };
    Plotly.newPlot('graph-plt', _data_for_graph, layout);
    delete data_for_graph;

    window.fft_keys = Object.keys(data_for_fft_graph)
    window.trace1 = {
        x: data_for_fft_graph[fft_keys[1]],
        y: data_for_fft_graph[fft_keys[0]],
        mode: 'markers',
        type: 'scattergl',
        error_y: {
            type: 'data',
            symmetric: false,
            arrayminus: data_for_fft_graph[fft_keys[0]],

```

```

    array: new Array(data_for_fft_graph[fft_keys[1]].length).fill(0),
    width: 1
  },
  name: 'Швидке перетворення Фур\'є',
};
window._data_for_graph = [trace1];
window.layout_fft = {
  title: 'Швидке перетворення Фур\'є',
  xaxis: {
    title: `Частота (Гц)`,
    showgrid: true,
    zeroline: true
  },
  yaxis: {
    title: `Амплітуда`,
    showgrid: true,
    zeroline: true,
  },
  showlegend: false,
};
Plotly.newPlot('periodogram-graph-plt', _data_for_graph, layout_fft);

window.periodograma_matcher = {
  "list-periodograma-list": {
    'data': data_for_periodogram_graph,
    'label': 'Періодограма',
  },
  "list-triangle-list": {
    'data': data_for_triangle_periodogram_graph,
    'label': 'Трикутна періодограма',
  },
  "list-hann-list": {
    'data': data_for_hann_periodogram_graph,
    'label': 'Періодограма Ханна',
  },
  "list-blackman-list": {
    'data': data_for_blackman_periodogram_graph,
    'label': 'Періодограма Blackman',
  },
  "list-hamming-list": {

```

```

'data': data_for_hamming_periodogram_graph,
'label': 'Періодограма Hamming',
},
"list-bartlett-list": {
'data': data_for_bartlett_periodogram_graph,
'label': 'Періодограма Bartlett',
},
"list-flattop-list": {
'data': data_for_flattop_periodogram_graph,
'label': 'Періодограма Flattop',
},
"list-parzen-list": {
'data': data_for_parzen_periodogram_graph,
'label': 'Періодограма Parzen',
},
"list-bohman-list": {
'data': data_for_bohman_periodogram_graph,
'label': 'Періодограма Bohman',
},
"list-blackmanharris-list": {
'data': data_for_blackmanharris_periodogram_graph,
'label': 'Періодограма Blackmanharris',
},
"list-nuttall-list": {
'data': data_for_nuttall_periodogram_graph,
'label': 'Періодограма Nuttal',
},
"list-barthann-list": {
'data': data_for_barthann_periodogram_graph,
'label': 'Періодограма Barthann',
},
"list-cosine-list": {
'data': data_for_cosine_periodogram_graph,
'label': 'Періодограма Cosine',
},
"list-exponential-list": {
'data': data_for_exponential_periodogram_graph,
'label': 'Експотенціальна Періодограма',
},
"list-tukey-list": {

```



```

        'data': data_for_tukey_periodogram_graph,
        'label': 'Періодограма Tukey',
    },
    "list-taylor-list": {
        'data': data_for_taylor_periodogram_graph,
        'label': 'Періодограма Тайлера',
    },
}

window._x = [...Array(_l).keys()];
window.periodogram_data = {
    labels: _x,
    datasets: [{
        label: 'Віконна функція',
        backgroundColor: 'rgb(255, 99, 132)',
        borderColor: 'rgb(255, 99, 132)',
        data: periodogram_without_signal_periodogram,
    }
    ]
};

window.periodogram_matcher = {
    "list-periodograma-list": periodogram_without_signal_periodogram,
    "list-triangle-list": triangle_periodogram,
    "list-hann-list": hann_periodogram,
    "list-blackman-list": blackman_periodogram,
    "list-hamming-list": hamming_periodogram,
    "list-bartlett-list": bartlett_periodogram,
    "list-flattop-list": flattop_periodogram,
    "list-parzen-list": parzen_periodogram,
    "list-bohman-list": bohman_periodogram,
    "list-blackmanharris-list": blackmanharris_periodogram,
    "list-nuttall-list": nuttall_periodogram,
    "list-barthann-list": barthann_periodogram,
    "list-cosine-list": cosine_periodogram,
    "list-exponential-list": exponential_periodogram,
    "list-tukey-list": tukey_periodogram,
    "list-taylor-list": taylor_periodogram,
}

window.periodogram_trace = {
    x: _x,

```

```

y: periodogram_without_signal_periodogram,
type: 'scatter',
mode: 'lines+markers',
};
window._data_for_periodogram = [periodogram_trace];
window.periodogram_layout = {
  title: 'Віконна функція',
  font: {
    size: 8
  },
  showlegend: false,
  xaxis: {
    title: `Час (с)`,
    showgrid: true,
    zeroline: true,
    autorange: true,
  },
  yaxis: {
    title: `Амплітуда (Гц)`,
    showgrid: true,
    zeroline: true,
    autorange: true,
  }
};
Plotly.newPlot('only-periodograma-graph-plt', _data_for_periodogram,
periodogram_layout);

```

```

window.freq_periodogram_matcher = {
  "list-periodograma-list": freq_periodogram_without_signal_periodogram,
  "list-triangle-list": freq_triangle_periodogram,
  "list-hann-list": freq_hann_periodogram,
  "list-blackman-list": freq_blackman_periodogram,
  "list-hamming-list": freq_hamming_periodogram,
  "list-bartlett-list": freq_bartlett_periodogram,
  "list-flattop-list": freq_flattop_periodogram,
  "list-parzen-list": freq_parzen_periodogram,
  "list-bohman-list": freq_bohman_periodogram,
  "list-blackmanharris-list": freq_blackmanharris_periodogram,
  "list-nuttall-list": freq_nuttall_periodogram,
  "list-barthann-list": freq_barthann_periodogram,

```

```

    "list-cosine-list": freq_cosine_periodogram,
    "list-exponential-list": freq_exponential_periodogram,
    "list-tukey-list": freq_tukey_periodogram,
    "list-taylor-list": freq_taylor_periodogram,
  }

  window.periodogram_trace_freq = {
    x: freq_periodogram_without_signal_periodogram['freq'],
    y: freq_periodogram_without_signal_periodogram['response'],
    type: 'scatter',
    mode: 'lines+markers',
  };
  window._data_for_periodogram_freq = [periodogram_trace_freq];
  window.periodogram_layout_freq = {
    title: 'Віконна функція',
    font: {
      size: 8
    },
    showlegend: false,
    xaxis: {
      title: `Частота (Гц)`,
      showgrid: true,
      zeroline: true,
      autorange: true,
    },
    yaxis: {
      title: `Амплітуда (дБ)`,
      showgrid: true,
      zeroline: true,
      autorange: true,
    }
  };

  Plotly.newPlot('only-periodograma-frequency-graph-plt',
    _data_for_periodogram_freq, periodogram_layout_freq);

}

set_variables(data, data_for_graph);

$( ".freq-graph" ).click((e, t) => {
  $(".freq-graph").removeClass("active");

```

```

window.target_id = e.target.id;
console.log(1234);
if(target_id === "list-fft-list"){
    $("#periodograma_div").addClass("d-none");
    window.trace1 = {
        x: data_for_fft_graph[fft_keys[1]],
        y: data_for_fft_graph[fft_keys[0]],
        mode: 'markers',
        name: 'Швидке перетворення Фур\'є',
        type: "scattergl",
        error_y : {
            type: 'data',
            symmetric: false,
            arrayminus: data_for_fft_graph[fft_keys[0]],
            array: new Array(data_for_fft_graph[fft_keys[1]].length).fill(0),
            width: 1
        },
    };
    window._data_for_graph = [trace1];
    window.layout_fft = {
        title: 'Швидке перетворення Фур\'є',
        xaxis: {
            title: `Частота (Гц)`,
            showgrid: true,
            zeroline: true
        },
        yaxis: {
            title: `Амплітуда`,
            showgrid: true,
            zeroline: true,
        },
        showlegend: false,
    };
    Plotly.react('periodogram-graph-plt', _data_for_graph, layout_fft);

} else {
    $("#periodograma_div").removeClass("d-none");

    window.per_data = periodograma_matcher[target_id];
    window.per_data_keys = Object.keys(per_data['data'])

```

```

window.per_data_trace = {
  x: per_data['data'][per_data_keys[1]],
  y: per_data['data'][per_data_keys[0]],
  type: 'scattergl',
  mode: 'lines+markers',
};
window._data_per = [per_data_trace];
window._periodogram_layout_freq = {
  title: per_data['label'],
  font: {
    size: 12
  },
  showlegend: false,
  xaxis: {
    title: `Частота (Гц)`,
    showgrid: true,
    zeroline: true,
    autorange: true,
  },
  yaxis: {
    title: `Амплітуда (дБ)`,
    showgrid: true,
    zeroline: true,
    autorange: true,
    type: 'log',
  }
};

Plotly.react("periodogram-graph-plt", _data_per,
_periodogram_layout_freq);

let _periodogram_trace = {
  x: _x,
  y: periodogram_matcher[target_id],
  type: 'scatter',
  mode: 'lines+markers',
};
let __data_for_periodogram = [_periodogram_trace];

```

```

Plotly.react("only-periodograma-graph-plt",
__data_for_periodogram,periodogram_layout);

freq_graph_data = freq_periodogram_matcher[target_id];

window._periodogram_trace_freq = {
  x: freq_graph_data['freq'],
  y: freq_graph_data['response'],
  type: 'scatter',
  mode: 'lines+markers',
};
window.__data_for_periodogram_freq = [_periodogram_trace_freq];
Plotly.react('only-periodograma-frequency-graph-plt',
__data_for_periodogram_freq, periodogram_layout_freq);

}
});

</script>

```

Додаток Б

Ф. № 3.3

СПИСОК НАУКОВИХ ПРАЦЬ Шелемаха Владислав Валентинович
(прізвище, ім'я, по батькові)

№ з/п	Найменування праць	Назва видавництва, журналу (номер, рік) або номер авторського свідоцтва, номер диплома на винахід	Кількість друкованих аркушів або сторінок разом	Прізвища співавторів праць
1	2	3	4	5
1	Автоматизація процесу обробки сигналів з коливальними складовими	Збірник праць XVII всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Ефективність та автоматизація інженерних рішень у приладобудуванні» – Київ, КПІ ім. Ігоря Сікорського, 6 грудня. – 2021.	5	Цибульник С.О.

2	ОГЛЯД ЗАСОБІВ ОБРОБКИ ІНФОРМАЦІЇ	Збірник праць XV всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Погляд у майбутнє приладобудування» – Київ, КПІ ім. Ігоря Сікорського, 14-15 червня. – 2022.	5	Цибульник С.О.
---	--	--	---	----------------

Автор _____ Шелемаха В.В.
(підпис) (прізвище, ім'я, по батькові)

Зав. каф. КІОНС _____ Надія БУРАУ
(підпис)