

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

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

До захисту допущено

Завідувач кафедри

(підпис) Віталій РОМАНКЕВИЧ
(ініціали, прізвище)

“ ____ ” _____ 2020 р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Комп'ютерні системи та компоненти»
спеціальності 123 «Комп'ютерна інженерія»**

на тему: **«Автоматизоване робоче місце викладача університету»**

Виконав:

студент IV курсу, групи КВ-62
(шифр групи)

Троян Борис Володимирович
(прізвище, ім'я, по батькові)

(підпис)

Керівник, доц.каф.СПСКС, к.т.н, Петрашенко А.В.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М.
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проєкті немає
запозичень з праць інших авторів без відповідних
посилань.

Студент _____
(підпис)

Київ – 2020 року

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

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерні системи та компоненти»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ
(підпис) (ініціали, прізвище)

«___» _____ 2020 р.

ЗАВДАННЯ

на дипломний проєкт студента

Трояна Бориса Володимировича

(прізвище, ім'я, по батькові)

1. Тема проєкту: Автоматизоване робоче місце викладача університету,

керівник проєкту доц.каф. СПСКС, к.т.н, Петрашенко А.В.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «25» травня 2020 р. № 1181-С

2. Термін подання студентом проєкту _____

3. Вихідні дані до проєкту: див. технічне завдання.

4. Зміст пояснювальної записки: обґрунтування теми дипломного проєкту, обґрунтування вибору засобів для виконання проєкту, методів написання автоматизованої системи, що включає роботу з БД та використання клієнт-серверної архітектури програми, тестування програмного продукту та аналіз його результатів.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо):

Структура БД додатку. Структурна схема;

Обробка сервером запиту до відображення вебхука. Схема алгоритму;

Алгоритм роботи відображення lesson_view. Схема алгоритму;

Формування HTML-сторінки з шаблонів DTL на прикладі сторінки lesson_edit.html. Схема алгоритму.

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., доц.каф.СПСКС,к.т.н.		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
	Видача завдання на дипломне проєктування	04.11.2020	
	Розробка технічного завдання	25.12.2020	
	Аналіз існуючих рішень	15.01.2020	
	Вибір середовища розробки	09.02.2020	
	Розробка програмного продукту	29.03.2020	
	Відлагодження програмного продукту	19.04.2020	
	Підготовка пояснювальної записки	05.05.2020	
	Оформлення матеріалів проєкту	18.05.2020	
	Попередній огляд матеріалів диплому на кафедрі	20.05.2020	

Студент

(підпис)

Борис ТРОЯН

Керівник проєкту

(підпис)

Андрій ПЕТРАШЕНКО

АНОТАЦІЯ

Дипломний проєкт включає пояснювальну записку (64 стор., 30 рис.).

Метою даного бакалаврського проєкту є розробка автоматизованого робочого місця викладача університету, призначеного для підвищення ефективності організації навчального процесу. Розробка має вигляд клієнт-серверної системи, клієнтська частина якої представлена у вигляді сайту для використання викладачами та студентами можливостей, що надає система, реалізовано сучасний адаптивний інтерфейс, пристосований як для мобільних, так і десктопних пристроїв. Серверна частина в свою чергу обробляє запити клієнтів, будує сторінки сайту з шаблонів, а також здійснює обробку форм та взаємодію з базою даних. Система дозволяє ефективно здійснювати відмічання присутності студентів на заняттях, контроль виконання цифрових лабораторних робіт, виставлення оцінок та відслідковування рейтингу студентів. Система орієнтована на подальше розширення та модифікацію для адаптації до специфіки окремих закладів.

Для створення актуальної розробки було проаналізовано основні проблеми впровадження цифрових засобів в галузі освіти, розглянуто існуючі аналоги та обрано підходящі засоби для розробки.

При виконанні роботи також було здійснено тестування програмної розробки та проаналізовані його результати для виявлення можливих недоліків системи та шляхів їх подолання.

Ключові слова: web-додаток, Python, Django, автоматизована система, клієнт-серверна система, база даних, організація освітнього процесу, студентська інформаційна система.

ABSTRACT

The diploma project includes an explanatory note (64 p., 30 fig.).

The goal of this project is development of an automated workspace for university teacher, which is designed to increase the efficiency of the educational process. Developed program constructed as client-server system. The client part is presented in the form of a website that allows teachers and students to use features provided by the system. Website have a modern adaptive interface that works both on mobile and desktop devices. The server part processes user's requests, constructs site pages from templates. Also it processes forms and accomplish interactions with database. The system allows user to effectively mark attendance of students in class, to control the performance of lab works in digital form, to grade and track student ratings. The system is suited for further expansion and modification to adapt to the specifics of individual institutions.

To create a relevant software, were analyzed main problems related to introduction of digital tools in the education, existing analogues were analyzed and suitable tools for development were selected.

During the work, software was also tested and it's results were analyzed to identify potential shortcomings of the system and ways to overcome them.

Keywords: web-application, Python, Django, automated system, client-server system, database, organization of educational process, student information system.

[illegible]

[illegible]

ЗМІСТ

1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ.....	2
3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ	2
4. ДЖЕРЕЛА РОБОТИ	2
5. ТЕХНІЧНІ ВИМОГИ.....	3
5.1. Вимоги до системи, що розробляються.....	3
5.2. Вимоги до апаратного забезпечення.....	3
5.3. Вимоги до програмного забезпечення.....	3
6. ЕТАПИ РОЗРОБКИ	4

					ІАЛЦ.045440.002 ТЗ				
Змін.	Арк.	№ докум.	Підпис	Дата	Автоматизоване робоче місце викладача університету Технічне завдання	Літ.	Аркуш	Аркушів	
Розробив		Троян Б.В.							
Перевірив		Петрашенко А.В.					1	4	
						КПІ ім. Ігоря Сікорського Каф. СПіСКС гр. КВ-62			
Н. контроль		Клятченко Я.М.							
Зав. каф.		Романкевич В.О.							

НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Автоматизоване робоче місце викладача університету».

Галузь застосування: автоматизація організації освітнього процесу через веб-серверний додаток.

ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення системи у вигляді клієнт-серверного додатку для автоматизації організаційних моментів освітнього процесу з використанням сучасних тенденцій цифрових технологій.

ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації в періодичних виданнях та електронні статті у мережі Інтернет.

					ІАЛЦ.045440.002 ТЗ	Арк.
						9
Змін.	Арк.	№ докум.	Підпис	Дата		

ТЕХНІЧНІ ВИМОГИ

- **Вимоги до програмного продукту, що розробляється**
 - сумісність з сучасними мобільними пристроями;
 - зрозумілий пересічному користувачу інтерфейс клієнтської частини;
 - можливість отримувати від системи найактуальнішу інформацію по розкладу та оцінкам;
 - можливість обслуговувати багато користувачів одночасно;
 - наявність механізму відмічання студентів на занятті;
 - можливість підключення вебхука з Github;
 - можливість редагування даних бази даних обслуговуючим персоналом.
- **Вимоги до апаратного забезпечення сервера**
 - Комп'ютер на базі процесора сімейства AMD, A4-4020 та вище або сімейства Intel, G3900 та вище (залежить від цільового навантаження) з сумісною ОС;
 - оперативна пам'ять 1 Гбайт або більше (залежить від цільового навантаження);
 - наявність доступу до мережі Інтернет.
- **Вимоги до програмного та апаратного забезпечення користувача**
 - веб-браузер;
 - наявність доступу до мережі Wi-Fi (IEEE 802.11 b/g/n).

					ІАЛЦ.045440.002 ТЗ	Арк.
						10
Змін.	Арк.	№ докум.	Підпис	Дата		

ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів
1.	Видача завдання на дипломне проєктування	04.11.2019
2.	Вивчення літератури за тематикою роботи	16.11.2019
3.	Розроблення та узгодження технічного завдання	25.11.2019
4.	Розроблення структури додатку	17.01.2020
5.	Розроблення дизайну та графічних елементів	04.02.2020
6.	Програмна реалізація додатку	15.03.2020
7.	Тестування додатку	04.04.2020
8.	Підготовка матеріалів текстової частини проєкту	27.04.2020
9.	Підготовка матеріалів графічної частини проєкту	15.05.2020
10.	Оформлення технічної документації проєкту	18.05.2020

					ІАЛЦ.045440.002 ТЗ	Арк.
						11
Змін.	Арк.	№ докум.	Підпис	Дата		

[illegible]

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	3
ВСТУП	5
РОЗДІЛ 1. АНАЛІЗ ПОСТАВЛЕНИХ ЗАДАЧ ТА ІСНУЮЧИХ РІШЕНЬ..	11
1.1 Аналіз існуючих на ринку рішень SIS	11
1.2 Формування конкретних задач та стандартів для розроблюваного продукту	14
Висновки до розділу	15
РОЗДІЛ 2. ОПИС РОЗРОБЛЕНОЇ СИСТЕМИ АВТОМАТИЗОВАНОГО РОБОЧОГО МІСЦЯ ВИКЛАДАЧА.....	17
2.1 Огляд функціональних можливостей та інтерфейсу отриманої розробки	17
2.1.1 Клієнтська підсистема	18
2.1.2 Серверна підсистема.....	28
2.2 Потенційні шляхи подальшого розвитку розробки	30
Висновки до розділу	31

РОЗДІЛ 3. ТЕХНІЧНІ РІШЕННЯ ВИКОРИСТАНІ В РОБОТІ	33
3.1 Аналіз доступних інструментів та обґрунтування доцільності використання обраних для розробки засобів (Python Django)	33
3.2 Розгляд технологій використаних в розробці та представлення реалізованих технічних особливостей та деталей	35
Висновки до розділу	46
РОЗДІЛ 4. ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ.....	47
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56

ДОДАТКИ

Додаток 1. Копії графічних матеріалів

- ІАЛЦ.045440.005 Д1. Структура БД додатку. Структурна схема;
- ІАЛЦ.045440.006 Д2. Обробка сервером запиту до відображення вебхука. Схема алгоритму;
- ІАЛЦ.045440.007 Д3. Алгоритм роботи відображення lesson_view. Схема алгоритму;
- ІАЛЦ.045440.008 Д4. Формування HTML-сторінки з шаблонів DTL на прикладі сторінки lesson_edit.html. Схема алгоритму.

Додаток 2. Лістинг програми

Додаток 3. Презентація проєкту

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

БД	База даних
ВНЗ	Вищий навчальний заклад
Клік	запит-перехід по вказаному посиланню, симуляція натискання користувача на посилання.
API	Application Programming Interface (укр. Прикладний програмний інтерфейс) - це набір чітко визначених методів для взаємодії різних компонентів програми.
BYOD	Bring your own device - ІТ-політика, згідно з якою співробітникам рекомендується використовувати особисті мобільні пристрої
CSRF	Cross-Site Request Forgery (укр. Міжсайтова підробка запиту) - тип атаки на веб-ресурс
DTL	Django template language (укр. Мова шаблонів Django)
IT	Information technology (укр. Інформаційні технології)
JSON	JavaScript Object Notation - текстовий формат обміну даними між комп'ютерами.
ORM	Object-relational mapping (укр. Об'єктно-реляційна проєкція) - технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування.
SIS	Student Information System (укр. Студентська (учнівська) інформаційна система
SQL	Structured query language (укр. Мова структурованих запитів)
URL	Uniform Resource Locator (укр. Уніфікований локатор ресурсів) - стандартизована адреса ресурсу в інтернеті.

WSGI	Web Server Gateway Interface - стандарт взаємодії між програмою на мові Python, яка виконується на стороні сервера, і самим веб-сервером.
XSS	Cross-site scripting (укр. міжсайтовий скриптинг) - тип атаки на веб-ресурс

					ІАЛЦ.045440.004 ПЗ	Арк.
						16
Змін.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

В наш час цифрові технології вже певною мірою проникли в усі сфери життя як суспільства в цілому, так і окремих осіб. Робота, навчання, всі галузі побуту так чи інакше користуються все зростаючими можливостями, котрі надає їм стрімкий розвиток цифрових технологій - від простого пошуку рецептів до комплексного динамічного планування розпорядку, базуючись на цілому ряді фізіологічних параметрів окремої людини. Перші 20 років XXI століття характеризуються початком четвертої індустріальної революції, «цифровим переходом» від свого роду «аналогових» систем та процесів індустріальної економіки та інформаційного суспільства до «цифрової» економіки та «цифрового» суспільства[1]. З кожним днем інновації все більше змінюють звичний нам порядок речей, і з поглибленням інтеграції цифрових технологій пов'язані свої ризики, так само, як історично складалося з будь-якими іншими інноваціями. Проте задля забезпечення впевненого розвитку індустрії та суспільства інновації є неодмінною умовою, тому не можна дозволяти старим звичкам та ризикам ставати на шляху прогресу.

На відміну від наших попередників ми живемо в епоху вільного доступу до інформації, коли для отримання доступу до необхідних даних немає необхідності шукати документи по різних закладах або намагатись знайти експерта з галузі, яка вас цікавить, щоб щось отримати бажану інформацію, а достатньо провести пошук в інтернеті, доступ до якого є вже практично всюди. «Цифрова» грамотність визнана ЄС однією з 8 ключових компетенцій для повноцінного життя та діяльності[1]. Але незважаючи на це перед нами ще стоїть багато задач на шляху досягнення легкого та ефективного використання цифрових технологій в багатьох галузях як безпосередньо в Україні, так і в світі загалом. Швидкі та глибинні наслідки від переходу на цифрові технології будуть можливими лише тоді, коли «цифрова» трансформація стане основою

					ІАЛЦ.045440.004 ПЗ	Арк.
						17
Змін.	Арк.	№ докум.	Підпис	Дата		

життєдіяльності українського суспільства, бізнесу та державних установ, стане звичним та повсякденним явищем[2].

Однією з вищезгаданих галузей, і далеко не найменш важливою, є освіта. Враховуючи те, що для сучасного покоління школярів інформаційні технології є невід'ємною частиною звичного укладу їх життя, то цілком очевидним буде висновок, що освітня галузь має бути однією з провідних за впровадженням новітніх цифрових технологій та практик.

З деяких причин, серед яких обмеженість фінансування, вірність старим звичкам та застарілість класичних методик викладання, темпи впровадження інновацій зі сфери інформаційних технологій в галузі освіти в різних місцях може кардинально відрізнятись, як в масштабі окремого міста та області, так і України і всього світу. Для повноцінного впровадження цифрових технологій необхідно докладно розібратися в основних галузях та елементах освітнього процесу в яких ці технології мають найбільший вплив.

Ефективний пошук, отримання, обробка та поширення необхідної для навчання інформації та прикладних засобів є однією з найбільш пріоритетних задач в процесі інтеграції цифрових технологій, адже без актуальних, точних та достовірних даних та надійних і зручних засобів для взаємодії з ними освітня галузь не може повноцінно підготувати спеціалістів котрі відповідають актуальним вимогам сучасності. Об'єм інформації в мережі Інтернет на сьогодні є просто колосальним, при цьому вона не надто добре організована, якась частина інформації вже застаріла, а нова може виявитись дубльованою, неповною або неправдивою по причині того, що саморегуляція, як одна із принципів основ Інтернету, за такого масштабу не дає належного результату. Враховуючи вказані нюанси, потребу в якісних засобах обробки даних з Інтернету, тим паче в галузі освіти, важко переоцінити. Прикладами цієї категорії цифрових засобів можна віднести пошукові системи та профільні бази даних.

					ІАЛЦ.045440.004 ПЗ	Арк.
						18
Змін.	Арк.	№ докум.	Підпис	Дата		

Цифрові технології є невід'ємним засобом в організації дистанційного навчання. Серед головних переваг безпосередньо дистанційного навчання можна відзначити перш за все його гнучкість в багатьох аспектах освітнього процесу, починаючи від вибору курсів та графіку занять та закінчуючи методами контролю, крім цього середня оцінка світових освітніх систем показує, що повністю дистанційне навчання в середньому на 50% дешевше традиційних форм[3]. Засоби навчання онлайн вже встигли зарекомендувати себе і все ширше впроваджуються як доповнення до традиційних методик в освітніх закладах або самостійні системи для профільного навчання у вигляді різноманітних онлайн-курсів та шкіл. Електронне навчання вже поклало початок смарт-навчанню (Smart education). Смарт-навчання – це гнучке навчання в інтерактивному освітньому середовищі за допомогою контенту з усього світу, що знаходиться у вільному доступі[4]. Актуальність необхідності перегляду встановлених освітніх процесів з урахуванням можливостей дистанційного навчання особливо гостро відчувається саме зараз, під час епідемії, коли очне навчання не є допустимим.

Цифрова грамотність є критично необхідною умовою успішної інтеграції цифрових технологій в освітній процес, адже без відповідних навичок користування навіть найпередовіші технології не зможуть дати очікуваного результату. Саме впровадження цифрової грамотності є першим кроком, котрий дозволить поширити ефективний вплив цифрових технологій в усіх галузях, в тому числі і в освіті. Як підтвердження даної тези можна навести той факт, що в Україні як один з кроків плану заходів щодо реалізації Концепції розвитку цифрової економіки та суспільства України на 2018-2020 роки[5] вже запроваджено державний сайт <https://osvita.diia.gov.ua/>, призначений для набуття громадянами цифрової грамотності, необхідної в сучасних реаліях, і сам по собі є прикладом успішного використання цифрових технологій в навчанні.

					ІАЛЦ.045440.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		19

7 березня 2018 року, було підписано Меморандум про співпрацю між Міністерством освіти і науки України та Асоціацією «Інформаційні технології України» метою якого є впровадження цифрових технологій в галузі освіти та створення сучасного освітнього середовища.

Керування навчальним процесом з використанням цифрових технологій вносить не менший за вищезгадані галузі вплив на освіту за рахунок оптимізації організаційних питань навчального процесу. Впровадження цифрових систем дозволяє спростити та автоматизувати більшість процесів котрі супроводжують процес навчання, значно полегшуючи роботу адміністративного персоналу, дозволяючи викладачам та учням зосередитись на безпосередньо на вивченні навчального матеріалу, не відволікаючись зайвий раз на організаційні моменти. Організаційні елементи на основі цифрових технологій показують значно підвищену ефективність порівняно з традиційними засобами. Для ілюстрації - електронний розклад завжди містить найактуальнішу інформацію щодо проведення занять, електронний журнал дозволяє зекономити час, що витрачається на організаційні питання та значно полегшує роботу по збору та обробці навчальної статистики, керування оцінками онлайн дозволяє викладачам значно підвищити ефективність взаємодії з іншими викладачами, персоналом та, потенційно, батьками, а контрольні заходи в цифровому форматі спростять процес оцінювання знань як для студентів, так і для викладачів. І це далеко не всі інновації, що може запропонувати інтеграція цифрових технологій в навчальний процес.

Наразі повсюди в освітніх закладах, принаймні в Україні, можна помітити приклади нехтування комп'ютерними засобами або недоцільне та неповне використання цифрових засобів в освітньому процесі. Часто виникають ситуації, коли цифрові системи імплементуються в навчальному закладі лише “для вигляду”, що обмежує їх практичне використання по причині недостатнього функціоналу, слабкої інтеграції елементів системи як між собою,

					ІАЛЦ.045440.004 ПЗ	Арк.
						20
Змін.	Арк.	№ докум.	Підпис	Дата		

та невідповідності реальним потребам навчального процесу, що в свою чергу відштовхує потенційних користувачів, котрі не захочуть адаптуватись до системи, що, по суті, не надає їм реальних переваг.

Потреба в ефективних SIS та подібних інформаційних системах найчіткіше помітна серед університетів та подібних закладів вищої освіти. Порівняно з іншими освітніми закладами, більший ступінь свободи в навчанні та викладанні загострює потребу в засобах, що можуть ефективно об'єднувати окремі елементи освітнього процесу в одній системі. Саме цифрові технології можуть забезпечити оптимальне рішення для взаємної інтеграції галузей, згаданих раніше, між собою для створення продуктивного середовища для навчання студентів.

Після аналізу даних складових стає очевидною необхідність впровадження цифрових технологій в освітній сфері та переваги котрі воно надасть. В сучасному суспільстві вже сформувались передумови, необхідні для прийняття, та навіть пришвидшення відповідних змін. Це підтверджують статистичні дослідження по зацікавленості студентів та викладачів в сфері освітніх інновацій та трендів, в першу чергу - з використанням цифрових технологій та інструментів (Рис. 1) [6].

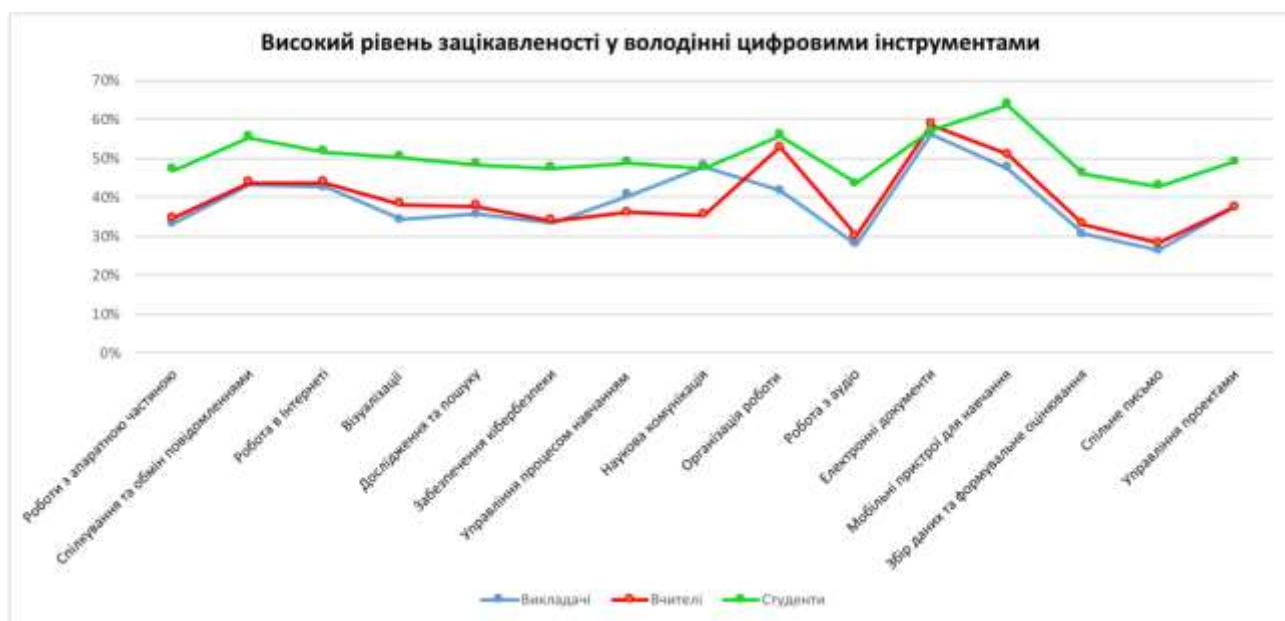


Рисунок 1. Зацікавленість у використанні цифрових інструментів

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045440.004 ПЗ

Арк.
21

Даний дипломний проєкт націлений на сприяння інтеграції цифрових технологій в освітню галузь, а саме створення нових програмних засобів, зв'язаних з керуванням навчальним процесом та його організацією. Для даного проєкту після розгляду вже згаданих потреб і передумов, та аналізу вже існуючих рішень в даній області, прикладів інтеграції цифрових технологій в інших галузях і наявних ресурсів і засобів розробки було вирішено створити базову систему автоматизованого робочого місця викладача для оптимізації взаємодії викладача університету зі студентами та полегшення повсякденних організаційних задач, котра в подальшому може бути легко адаптована та розширена.

					ІАЛЦ.045440.004 ПЗ	Арк.
						22
Змін.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ПОСТАВЛЕНИХ ЗАДАЧ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Аналіз існуючих на ринку рішень SIS

Для виконання даного курсового проєкту перш за все необхідно докладно розібратися, які саме функції є актуальними та які варто реалізувати в програмній розробці даного проєкту, як досягти максимальної ефективності даних функцій в поточних реаліях та які інновації, порівняно з існуючими рішеннями, можна запровадити для покращення досвіду потенційних користувачів розробки.

Для цього в міру моїх можливостей було розглянуто значну кількість учнівських інформаційних систем (SIS - англ. Student Information System), представлених на сучасному ринку.

Більшість великих комерційних рішень, такі як Schoology, Alma, ClassReach, Edsby, Classlink та інші, являють собою великі інтегровані системи, орієнтовані здебільшого на середню освіту. Більшість даних SIS включають такі складові елементи як:

- Структуризація навчальних груп;
- Засоби зберігання та перегляду різноманітної інформації про учнів;
- Менеджмент оцінок та відвідуваності в цифровому вигляді;
- Керування навчальними курсами в цифровому вигляді;
- Організація календарного плану як навчального, так і організаційного процесу;
- Засоби для генерації та друку документів та звітів;
- Колаборація учнів: об'єднання в окремі групи для виконання задач;
- Засоби комунікації учнів та вчителів;
- Засоби для збору статистики, зокрема відвідуваності та успішності;
- Засоби аналітики;
- Засоби для батьків для моніторингу та інтеграції в навчальний процес;

					ІАЛЦ.045440.004 ПЗ	Арк.
						23
Змін.	Арк.	№ докум.	Підпис	Дата		

- Система оголошень та розсилки на e-mail;
- Елементи дистанційного навчання;
- Відеоуроки;
- Організація завдань в електронному вигляді;
- Візуальна персоналізація системи та окремих сторінок;
- Відображення новин та оголошень навчального закладу;
- Оплата послуг;
- Засоби моніторингу шкіл округів.

В технічному плані рішення даного класу зазвичай акцентують увагу на великій кількості окремих функцій, перелічених вище, на безпеці та надійності їх системи та підтримці користувачів в питаннях на кшталт сервісу або персоналізації сайту для окремого закладу. Деякі з них мають власний мобільний додаток.

Можна виділити в окрему категорію інформаційні системи, що базуються на власних розробках окремих освітніх закладів, котрими користуються здебільшого заклади вищої освіти. Залежно від потреб кожного окремого закладу функціонал може відрізнятись, хоча такі істотні елементи як, наприклад, структура та склад груп або організація розкладу мають бути присутні в тому чи іншому вигляді. Порівняно з попередньою категорією в цій не потрібні такі елементи як батьківський контроль або моніторинг інших освітніх закладів. Проте в таких SIS для вищих навчальних закладів є і такі спільні елементи котрих немає у вищезгаданих комерційних систем.

Серед них:

- механізми для обрання власних дисциплін та курсів.
- засоби організації контрольних заходів вищої освіти (екзамени, дипломні роботи, тощо)
- засоби для організації та підтримки наукової роботи студентів та викладачів
- елементи призначені для інтеграції міжвузового та міжнародного співробітництва

					ІАЛЦ.045440.004 ПЗ	Арк.
						24
Змін.	Арк.	№ докум.	Підпис	Дата		

– засоби призначені для роботи приймальної комісії

SIS вищих навчальних закладів можуть значно відрізнятись в залежності від розміру закладу та основного напрямку викладання. ВНЗ котрий готує спеціалістів з інформаційних технологій потребує зовсім інших засобів та методів оцінювання студентів ніж, наприклад, ВНЗ економічного напрямку.

Незважаючи на відмінності вказаних типів SIS обидві категорії мають відповідати сучасним стандартам та потребам. В умовах все більшої доступності мережі Інтернет, зростаючої потужності компактних девайсів та величезній кількості програмних та апаратних рішень в різноманітних сферах ІТ дуже актуальною темою є питання інтеграції цих рішень між собою, в тому числі і інтеграція SIS з зовнішніми сервісами для підвищення її продуктивності. Інтеграція SIS зі сторонніми додатками дозволяє підвищити зручність та ефективність їх використання, а також зменшити навантаження на матеріальну базу закладу за рахунок покладання окремих функцій на інтегровані сервіси. Частина комерційних SIS серед своїх особливостей вказує можливість інтеграції з тими чи іншими популярними сервісами та додатками. Це можуть бути додатки для зберігання даних, проведення відеоуроків, комунікації, збору статистики або вузькопрофільного навчання - на сьогоднішній день в мережі можна знайти сервіси для будь-яких потреб. Прикладами таких сервісів є Google Drive, Google Classroom, Apple School Manager, Dropbox, OpenEd Library, Khan Academy та багато інших.

Після аналізу обраних екземплярів, було зроблено висновок, що вже існує значна кількість достойних рішень для освіти з використанням цифрових технологій і їх кількість та якість зростає, втім присутній ряд недоліків, зокрема в категорії власних SIS закладів вищої освіти по причині того, що вони по суті не конкурують, тому немає значної мотивації для подальшого розвитку. До таких недоліків відносяться недостатня структурна гнучкість багатьох розглянутих систем, що не дозволяє їх ефективно конфігурувати та масштабувати для зменшення вимог до матеріально-технічної бази. Іншим

					ІАЛЦ.045440.004 ПЗ	Арк.
						25
Змін.	Арк.	№ докум.	Підпис	Дата		

виявленим недоліком є той факт, що SIS здебільшого націлені на адаптацію та підтримку традиційних освітніх методик в цифровому форматі, при цьому не надто охоче реалізують функціонал новітніх методик котрі суттєво базуються саме на цифрових технологіях, таких як інтеграція та активне використання принципу BYOD, фокусування на дистанційному навчанні або сфокусована персоналізація навчальних засобів для учнів на основі передових методів серед яких аналітичні методи та нейронні мережі.

1.2 Формування конкретних задач та стандартів для розроблюваного продукту

В даному дипломному проєкті було поставлено за мету за мету створити програмну систему автоматизованого робочого місця викладача, котра б відповідала сучасним стандартам та містила інноваційні рішення, що базуються на активному використанні досягнень сфери ІТ.

Враховуючи наявні ресурси та можливості, розробка обширної програми з реалізацією абсолютно всіх елементів, причетних до навчального процесу, не є можливою. Натомість було вирішено створити базовий комплекс найбільш суттєвих елементів взаємодії викладача та учнів (далі - базова система). До цього комплексу входять:

- Засоби створення та менеджменту базових сутностей, таких як користувачі, дисципліни та інше (детальніше - в розділі 2);
- Система для зручного додавання та редагування занять та відображення актуального розкладу;
- Автоматизована система відмічання присутності студентів з допомогою електронних пристроїв (зокрема смартфонів) з використанням геопозиції;
- Виставлення оцінок за відповідні контрольні заходи;

					ІАЛЦ.045440.004 ПЗ	Арк.
						26
Змін.	Арк.	№ докум.	Підпис	Дата		

- Компактна система сповіщень.

Дана базова система може слугувати основою для подальшого розширення комплексу автоматизованого робочого місця викладача модулями для специфічних потреб та функцій

При розробці крім реалізації базової системи, було вирішено інтегрувати елементи, корисні здебільшого студентам спеціальностей зв'язаних з цифровими технологіями, таких як програмна інженерія, комп'ютерна інженерія або кібербезпека та, відповідно, викладачам, що викладають визначальні для цих спеціальностей дисципліни. Ці елементи утворюють підсистему, призначену для оптимізації перевірки лабораторних робіт що вимагають написання програм. Вона відслідковує відповідні зміни на репозиторії студента в системі Github та повідомляє про це відповідного викладача.

Ключовими якостями, що втілювались в даній розробці є:

- Простота використання: система має полегшувати організацію освітнього процесу, а не ускладнювати її;
- Гнучкість: можливість зміни дизайну та модифікації функціоналу під специфічні потреби окремого закладу;
- Модульність: можливість розширення функціоналу системи з допомогою додаткових програмних модулів;
- Спирання на цифрові технології як один з невід'ємних елементів навчального процесу.

Висновки до розділу

Існуючі сучасні рішення в сфері SIS хоч і мають широкий функціонал, проте володіють рядом недоліків, основними з яких є значні вимоги до матеріальної

					ІАЛЦ.045440.004 ПЗ	Арк.
						27
Змін.	Арк.	№ докум.	Підпис	Дата		

та технічної бази та небажання інтеграції здобутків ІТ-галузі замість застарілих традиційних практик. В даному проєкті створено осучаснене автоматизоване робоче місце викладача, в якому була зроблена спроба в міру наявних можливостей подолати ці недоліки та показати актуальність та ефективність інтеграції елементів, що суттєво спираються на сучасні здобутки та тенденції цифрових технологій.

					ІАЛЦ.045440.004 ПЗ	Арк.
						28
Змін.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ОПИС РОЗРОБЛЕНОЇ СИСТЕМИ АВТОМАТИЗОВАНОГО РОБОЧОГО МІСЦЯ ВИКЛАДАЧА

2.1 Огляд функціональних можливостей та інтерфейсу отриманої розробки

Автоматизоване робоче місце викладача університету являє собою веб-ресурс, написаний мовою Python 3 спираючись на засоби веб-фреймворку Django. Клієнти-користувачі отримують доступ до функціоналу системи через сторінку в веб-браузері.

В загальному вигляді ресурс надає користувачам такі функціональні можливості:

- Для всіх користувачів:
 - Перегляд профіля
 - Розклад
- Для викладача:
 - Редагування своїх занять
 - Додавання нових занять в свій розклад
 - Перегляд курсів, до яких викладач має відношення
 - Менеджмент контрольних заходів та оцінок
 - Перегляд оцінок за контрольний захід навчального курсу
 - Редагування оцінок групою та по одній
 - Додавання нових контрольних заходів
 - Редагування існуючих контрольних заходів
 - Ініціація процесу відмічання
 - Отримання повідомлень при комітах в репозиторіях, зв'язаних з курсами даного викладача
- Для студента:
 - Перегляд курсів, контрольних заходів та оцінок групи

					ІАЛЦ.045440.004 ПЗ	Арк.
						29
Змін.	Арк.	№ докум.	Підпис	Дата		

- Можливість підключення репозиторію на Github до навчального курсу
- Відмічання присутності з допомогою геопозиції
- Для персоналу
 - Заповнення розкладу заняттями для певного періоду
 - Засоби керування Django Admin

Система автоматизованого робочого місця викладача має дві сторони:

Клієнтська сторона (підсистема) (front-end), що слугує для комфортного доступу через веб-браузер до функціоналу системи всіх користувачів, навіть з мінімальними навичками роботи з комп'ютером.

Серверна сторона (підсистема) (back-end), котра обробляє всі запити користувачів, здійснює обробку форм та взаємодіє з базою даних, а також обробляє автоматичні запити з репозиторіїв github, що приєднані до системи.

2.1.1 Клієнтська підсистема

Розглянемо детальніше можливості що надає клієнтська підсистема. Вона реалізована у вигляді сайту, що має два основних шляхи: /admin/ та /workspace/.

/admin/ представляє собою автоматично згенеровану сторінку інтерфейсу адміністратора Django - стандартний набір засобів фреймворку призначений, здебільшого, для персоналу для роботи з даними БД: додавання, зміни та видалення екземплярів об'єктів БД. Загальний вигляд сторінки /admin/ представлено на Рис. 2. Так як це стандартний засіб, то його документацію [7] можна знайти на офіційному сайті Django та розглядати його тут далі немає сенсу.

За шляхом /workspace/ знаходиться безпосередньо індексна сторінка основної системи, і з цієї частини починаються URL всіх функціональних сторінок системи.

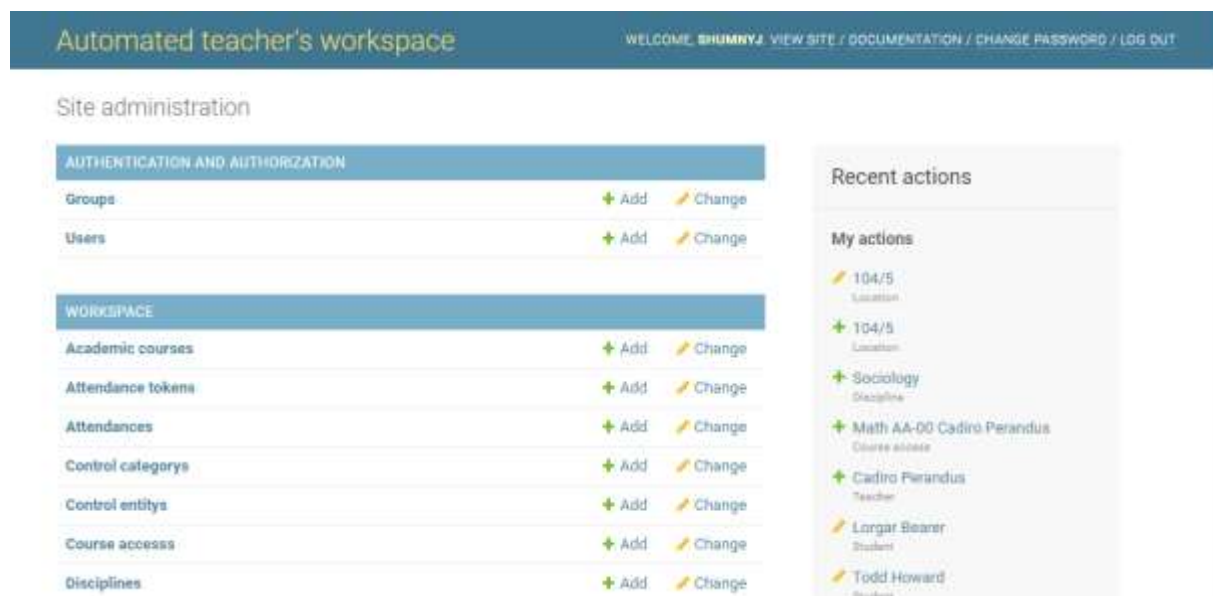


Рисунок 2. Загальний вигляд сторінки /admin/

Для комфортного використання на різних пристроях та полегшення процесу верстання сторінки системи були побудовані з використанням принципів адаптивного дизайну та засобів, що надає front-end фреймворк Bootstrap 4. Сторінки системи побудовані по загальному шаблону, структуру якого розглянемо на прикладі індексної сторінки на Рис. 3 та Рис. 4.

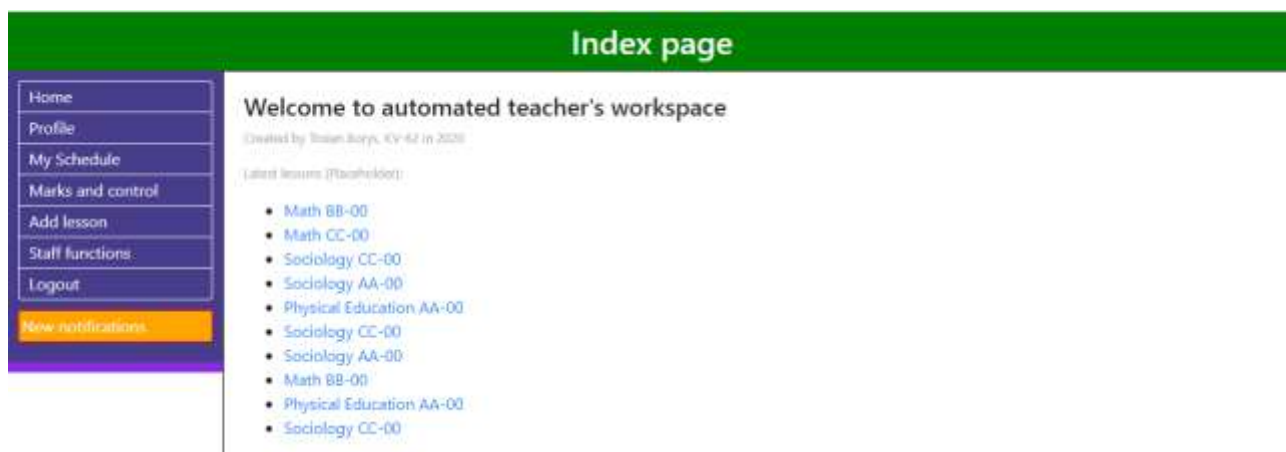


Рисунок 3 - Загальний вигляд сторінки системи

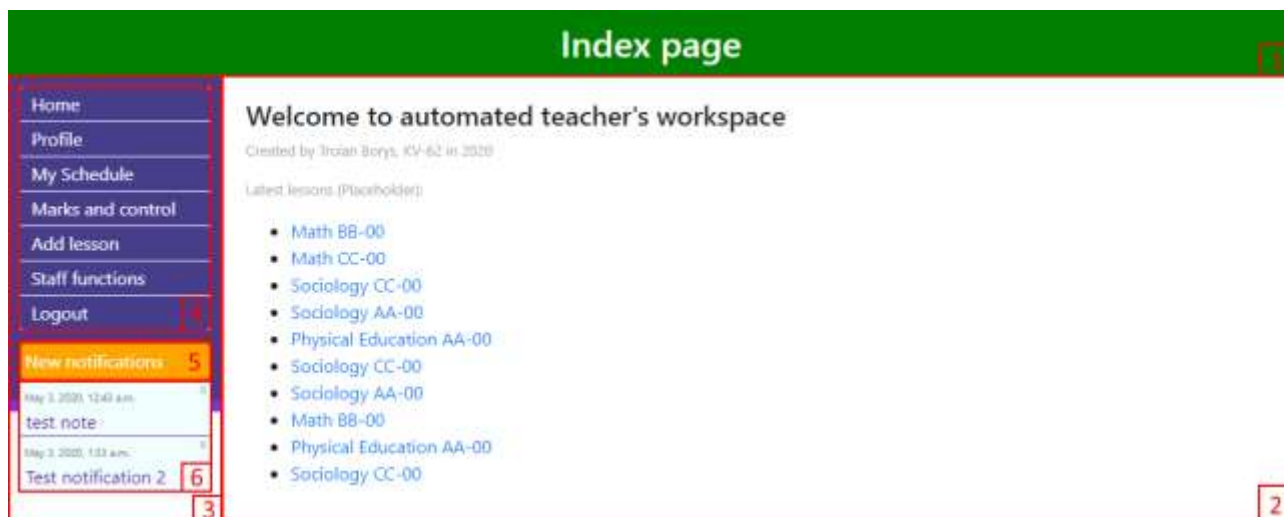


Рисунок 4 - Основні елементи сторінки системи

Основні елементи сторінки (Рис. 4):

1. Заголовок сторінки.
2. Основний вміст сторінки. Містить всі основні функції та інформацію поточної сторінки, наприклад оцінки студентів за певний контрольний захід на сторінці відповідного заходу.
3. Бокова панель.
4. Навігаційне меню. Містить пункти навігації, доступні поточному користувачеві.
5. Індикатор повідомлень. З'являється коли поточному користувачеві надходять сповіщень та при натиску дозволяє їх переглянути
6. Розгорнутий список сповіщень.

Компонування сторінки базується на використанні система сітки (grid) з Bootstrap, що побудована на засобах flexbox.

Вміст деяких елементів сторінок сайту, може змінюватись в залежності від поточного користувача, що забезпечується внутрішньою роботою відображень Django. Приклад такої адаптації наведено на Рис. 5 на прикладі навігаційного меню. а - неавторизований користувач, б - користувач зі штату обслуговуючого персоналу, в - викладач.

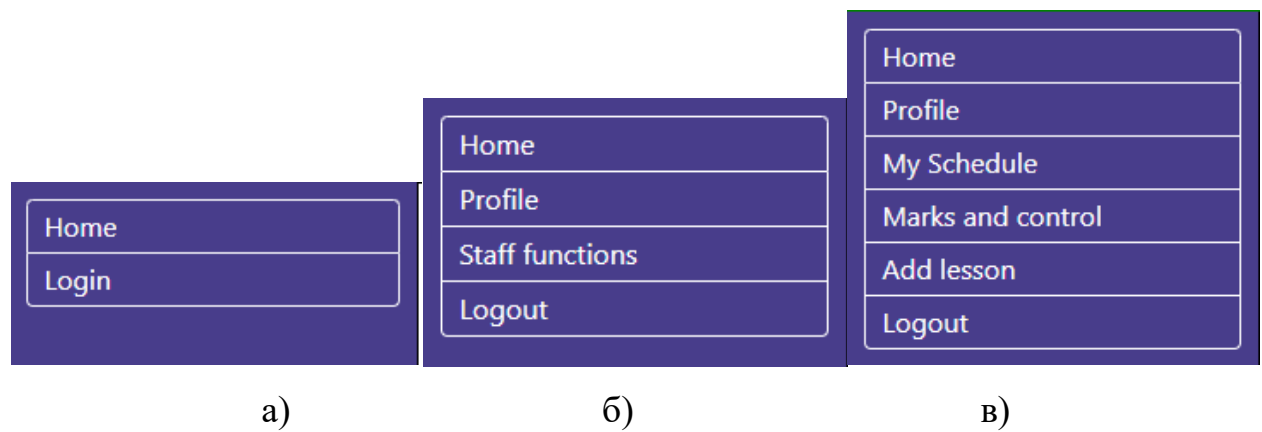


Рисунок 5. Відображення адаптації навігаційного меню;

Окремі динамічні елементи сторінок для роботи використовують функції Javascript, як написані спеціально для додатку (напр. меню сповіщень), так і взяті з фреймворку Bootstrap (напр. вкладки розкладу).

Так як нюанси освітнього процесу в кожному окремому закладі можуть відрізнятись, і такі нюанси при розробці даного проєкту були здебільшого невідомі, то в процесі розробки було зроблено певні припущення та спрощення щодо відповідних нюансів функціонування системи автоматизованого робочого місця викладача, таких як реєстрація, кількість груп та викладачів на уроці, деталі процесу редагування курсу та виставлення оцінок, тощо. Детальніше ці нюанси будуть розглянуті далі, посилаючись на специфіку роботи системи.

Для роботи в системі користувачу необхідно перш за все авторизуватись в свій аккаунт, це можна здійснити на сторінці `/workspace/login`. В будь-який момент можна деавторизуватися. Для авторизації та аутентифікації, розподілу користувачів по групах використовуються функції системи аутентифікації Django. Відповідно до специфіки роботи системи, реєстрація нових користувачів проводиться лише персоналом.

Після авторизації користувач отримує доступ до функціоналу системи.

На сторінці `/workspace/profile` користувач може переглянути дані свого профілю, та перейти на сторінку `/workspace/profile/edit` для редагування окремих параметрів свого профіля (обмежених специфікою роботи системи -

давати змінювати безпосередньо користувачу деякі дані, такі як навчальна група або ім'я та прізвище не є доречним).

Сторінка розкладу /workspace/schedule (Рис. 6) показує персональний розклад користувача на найближчі два тижні (починаючи з понеділка поточного тижня), розділений на вкладки по дням та дозволяє перейти на сторінки окремих занять. Розклад представлений у вигляді вкладок, вкладки для днів, на які не назначено занять є неактивними.



Рисунок 6 - Вигляд сторінки розкладу

На сторінці заняття /workspace/lesson/<lesson_id> можна переглянути деталі заняття: час початку, аудиторію, викладача, групу. Там же розміщено посилання на положення аудиторії на карті Google Maps. Дане посилання всередині системи організовано як відображення для перенаправлення користувача, що дозволяє легко змінити засоби відображення положення при потребі.

Окрім загальної інформації на сторінці уроку у відповідний час з'являються форма для початку процесу відмічання присутності у викладача (Рис. 7а) та для власне відмічання у студента (Рис. 7б).

a) Minutes for attendance:

б)

Рисунок 7 - Форми сторінки заняття

При початку відмічання на відповідному занятті, студентам надходить відповідне сповіщення. Таким чином процес відмічання автоматизується, дозволяючи відвести більше часу безпосередньо на навчання та прибираючи помилки, що виникають внаслідок людського фактору (не розчув прізвища, запис не в той рядок, тощо). Всі сповіщення мають термін дії та видаляються при його спливанні.

Сторінка `/workspace/marks` відображає список курсів, до яких причетний поточний користувач. З неї можна перейти на сторінки окремих курсів, що мають шлях `/workspace/marks/course/<course_id>/`. Відповідно до специфіки роботи системи переглядати сторінки курсів та оцінок за контрольні заходи можуть лише користувачі, причетні до них, решта користувачів будуть перенаправлені до їх власної сторінки курсів. Загальний вигляд вмісту сторінки курсу представлено на Рис. 8.

Control entities for course Math AA-00:

- [Control work #1](#)
- [Lab Work #1](#)
- [Control work #2](#)
- [Lab Work #2](#)

[Marks table](#)

Рисунок 8 - Загальний вигляд вмісту сторінки навчального курсу

					ІАЛЦ.045440.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		35

На сторінці присутні посилання на сторінки всіх контрольних заходів та зведену таблицю оцінок за цей курс. Сторінка контрольного заходу `/workspace/marks/course/<course_id>/ce/<control_entity_id>` містить загальну інформацію про контрольний захід та оцінки за нього (Рис. 9а). Сторінка `/workspace/marks/course/<int:c_id>/table` містить таблицю всіх оцінок та сумарний рейтинг студентів за курс (Рис. 9б). Для генерації таблиць з оцінками замість звичайних використовуються окремі алгоритми побудови списків.

Така система відображення оцінок дозволяє всім учасникам навчального процесу набагато легше відстежувати поточні оцінки та здачу робіт, без застосування сторонніх засобів.

Marks for Control work #2:

Control work

Deadline: June 1, 2020

Highest mark: 10.0

Study materials:

"Covers themes: A B C"

[<- Back to course](#)

Students	Mark
Bearer Lorgar	7.0
Corax Corvus	7.0
El'Jonson Lion	9.0
Lupercal Horus	5.0
Red Magnus	8.0

[Edit marks](#)

[Edit control entity](#)

а)

Marks table of course Math AA-00:

Student	Control work #1	Lab Work #1	Control work #2	Lab Work #2	Total
Bearer Lorgar	1.0		7.0	3.0	11.0
Corax Corvus	2.0	5.0	7.0	4.0	18.0
El'Jonson Lion	4.0	3.0	9.0	3.0	19.0
Lupercal Horus	1.0	3.0	5.0	5.0	14.0
Red Magnus	3.0		8.0	5.0	16.0

[<- Back to course](#)

б)

Рисунок 9 - Відображення оцінок за курс

Очевидно, що викладачу необхідно так чи інакше керувати своїми заняттями та навчальними курсами. Для цього створені відповідні сторінки з формами (приклади на Рис. 10, 11):

- /workspace/marks/course/<course_id>/ce/<control_entity_id>/edit для зміни виставлення оцінок за контрольний захід (Рис. 11).
- /workspace/marks/course/<course_id>/ce/<control_entity_id>/change для редагування власне контрольного заходу
- /workspace/marks/course/<int:c_id>/add для додавання нових контрольних заходів (Рис. 10а)
- /workspace/lesson/<int:l_id>/edit для редагування існуючих занять, котрі ще не відбулися
- /workspace/lesson/add для додавання занять в свій розклад (Рис. 10б)

Add control entity for course Math AA-00:

Name:

Control type:

Deadline:

Mark max:

Materials:

а)

Add lesson :

Type:

Location:

Date:

Lesson start time:

Discipline:

Student group:

б)

Рис 10 - Форми для додавання нових об'єктів в БД викладачем

Edit marks for Lab Work #2:

Students	Mark
Bearer Lorgar	
Corax Corvus	
El'Jonson Lion	
Lupercal Horus	
Red Magnus	

Submit

Рисунок 12 - Форма для виставлення оцінок за контрольний захід

Редагування певним чином обмежено специфікою роботи системи, так як не всі елементи можуть бути змінені викладачем, наприклад в занятті можуть бути змінені час, дата, місце та тип, але не сам предмет або викладач. Незаповнені поля форм при редагуванні не змінюють відповідні поля об'єктів. Редаговані безпосередньо викладачем заняття відмічені зірочкою в розкладі для зручності. Викладачі, що зв'язані з курсом але не є його куратором можуть лише редагувати оцінки курсу.

Система надає студентам в кожному окремому навчальному курсі (для яких це актуально) посилання вигляду `course/wh/github/<course_id>-<stud_id>/` для прослуховування вебхуків з підключених репозиторіїв Github (Рис 12). Даний контекст введено в сторінку за допомогою функції `CUSTOM_CONTEXT_FUNC`, що буде розглянута в іншому розділі.

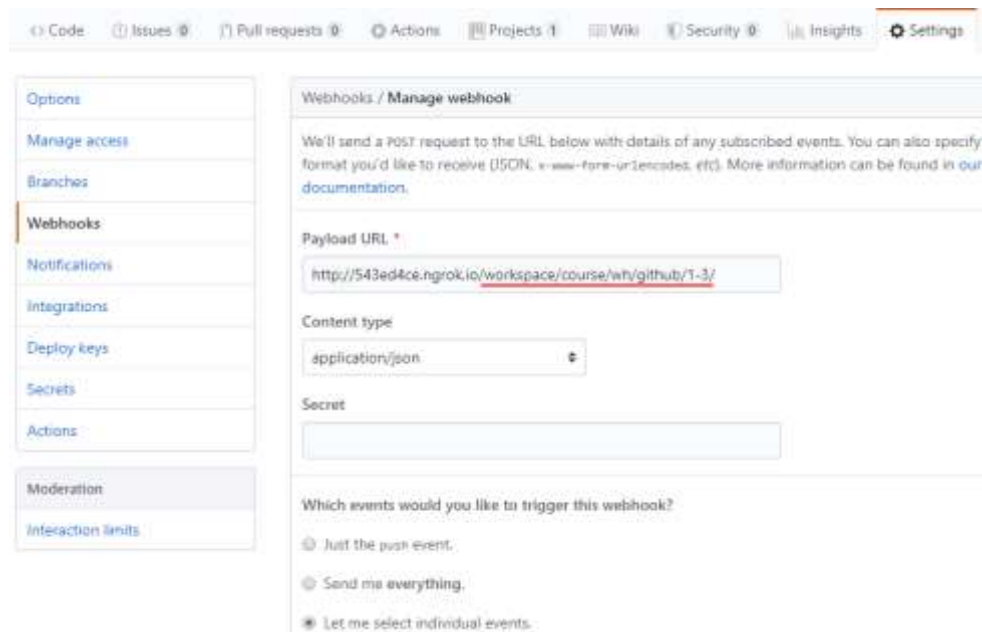


Рисунок 12 - Сторінка налаштування вебхука на репозиторії Github

Створена спеціальна сторінка /workspace/batch_add/ з формою для заповнення базового розкладу - додавання регулярних занять на певний проміжок часу (семестр). При надсиланні даної форми (Рис. 13) кожні 2 тижні (кожного тижня при встановленому полі Each of 2 weeks) на проміжку часу між датами в полях Start date та Last date включно, обраного дня тижня буде створений об'єкт уроку з заданими параметрами.

Discipline:	Mathematical analysis pt.1 ▾
Teacher:	Kayde Six ▾
Student group:	CC-00 ▾
Type:	----- ▾
Location:	----- ▾
Lesson start time:	
Day of the week:	THU ▾
Each of 2 weeks?	☐
Start date (week):	January ▾ 1 ▾ 2020 ▾
Last date (week):	January ▾ 1 ▾ 2020 ▾
Submit	

Рисунок 13 - Форма додавання занять партіями

2.1.2 Серверна підсистема

Тепер розглянемо можливості, що надає серверна підсистема. Вона представлена у вигляді додатку Django з усіма його стандартними можливостями - окрім власне запуску сервера можлива робота в режимі командної оболонки (shell), генерація та застосування міграції БД та використання додаткових команд власної розробки. До функцій працюючого сервера належать:

- Синхронний зв'язок з користувачами;
- Здійснення зв'язку з базою даних: отримання, обробка, збереження, надсилання користувачам та видалення даних з БД. В якості БД використана PostgreSQL;
- Генерація фінальної версії веб-сторінки з шаблону та надсилання користувачу;
- Підтримання системи авторизації;
- Обробка даних з форм.

Для авторизації користувачів використовуються стандартні засоби авторизації Django - додаток `django.contrib.auth`, що містить стандартну модель користувача `User`, систему дозволів, систему груп користувачів та налаштування хешування паролів. На додаток до цього в проєкті базова модель `User` розширюється в моделі-профілі `Teacher` та `Student` за допомогою відношень `OneToOne` та, на додаток до цього, з допомогою відповідних користувацьких груп.

Очевидно, що розроблений додаток є інтерактивним, при цьому практично вся взаємодія з користувачами відбувається за синхронною моделлю обміну даними (Рис 14), тому форми є невід'ємною складовою даної розробки.

					ІАЛЦ.045440.004 ПЗ	Арк. 40
Змін.	Арк.	№ докум.	Підпис	Дата		

```
[10/May/2020 21:59:34] "GET /workspace/marks/course/7/table HTTP/1.1" 200 2665
[10/May/2020 21:59:36] "GET /workspace/marks/course/7/ HTTP/1.1" 200 2226
[10/May/2020 21:59:53] "GET /workspace/marks/course/7/ce/28/ HTTP/1.1" 200 2724
[10/May/2020 21:59:55] "GET /workspace/marks/course/7/ce/28/edit/ HTTP/1.1" 200 2977
[10/May/2020 21:59:58] "POST /workspace/marks/course/7/ce/28/edit/ HTTP/1.1" 302 0
[10/May/2020 21:59:58] "GET /workspace/marks/course/7/ce/28/ HTTP/1.1" 200 2727
[10/May/2020 22:00:01] "GET /workspace/marks/course/7/ HTTP/1.1" 200 2226
```

Рисунок 14 - Звіт з консолі сервера про синхронні запити

Вони використовуються для редагування/додавання будь-яких об'єктів, серед яких заняття, оцінки та контрольні заходи, крім того на них спирається система відмічання, тому обробка форм становить значну частину функціоналу сервера. При заповненні форм можлива ситуація, коли не потрібно заповнювати або змінювати всі поля форми (або відповідного форму об'єкта), в такому випадку сервер, залежно від ситуації, проігнорує незаповнені поля або встановить туди значення по замовчуванню для уникнення технічних конфліктів. Механізм обробки форм сервером даного проєкту спирається на структуру форм Django, котра дозволяє значно спростити як процедуру попереднього заповнення та обмеження порожніх форм, так і процес отримання відповідних даних з запиту.

Система відмічання на занятті даного проєкту базується на наявності відповідного запису в БД - токена відвідування, котрий створюється при відправці викладачем у відповідний час форми (Рис. 7а) зі сторінки заняття. Після отримання форми викладача сервер створює токен для цього заняття, в якому вказує його час спливання, та записує його до БД, а також формує сповіщення та розсилає його учням, котрі входять до групи що відвідує дане заняття. В свою чергу учні при наявності відповідного токена учням на сторінці уроку відображається форма з Рис. 7б, при надсиланні якої сервер отримує дані про місцезнаходження учня та записує їх до запису відвідування, а учню надсилає відповідь, чи він був він успішно записаний.

Сервер має окремий набір шляхів `/workspace/course/wh/github/` для отримання та обробки повідомлень з підключених репозиторіїв Github за допомогою технології вебхук. При отриманні пакету з підключеного репозиторію сервер перевіряє його вміст, при отриманні коректних даних формуючи сповіщення для відповідного викладача. При спробі перегляду користувачем сторінки вебхука він буде перенаправлений сервером на початкову сторінку.

На додаток до основного функціоналу сервера для спрощення обслуговування БД адміністратором була розроблена користувацька команда Django для даного додатку під назвою **cleanup**. Вона дозволяє очистити БД від тимчасових записів, а саме токенів відвідування та сповіщень, термін дії котрих закінчився, але вони з певних причин залишилися в БД. Приклад роботи команди на Рис. 15.

```
C:\Users\shumnyj\source\PycharmProjects\TeacherWorkSpace>python manage.py cleanup
Successfully cleaned 3 notifications and 0 expired attendance tokens
```

Рис 15. Результат виклику команди cleanup

2.2 Шляхи подальшого розвитку розробки

Різні навчальні заклади мають різні потреби та можливості. Відповідно до них має змінюватись функціонал SIS, що буде актуальною для них. Наприклад, сільська школа на 300 осіб переслідує зовсім інші цілі у виборі інформаційної системи, ніж багатотисячний університет. Ціллю розробки було створення продукту, котрий за рахунок можливості модифікації потенційно зможе підійти для будь-якого закладу. За рахунок гнучкості окремих елементів програми (url, шаблонів) можливе досить значне налаштування вже існуючої розробки, а за рахунок пакетної системи Python та методології написання повторно

					ІАЛЦ.045440.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		42

використовуваних додатків Django на базі даної розробки можуть бути відносно легко створені додатки-доповнення, що розширюють та доповнюють функціонал базової системи відповідно до потреб. Найбільш перспективними напрямками для розвитку додатку є:

- інтеграція з популярними сервісами та додатками для підвищення ефективності навчального процесу, наприклад розсилка сповіщень через бота в Telegram з персоналізацією налаштувань або інтеграція Google Docs для зберігання навчальних матеріалів та спільного редагування документів;
- збір та обробка статистики по успішності та відвідуваності учнів;
- система для проходження тестування/опитувань на сайті з автоматизацією перевірки;
- модуль для розсилки важливих повідомлень на e-mail;
- модуль гнучкого розкладу та факультативів
- підсистема для бібліотеки;
- підсистема для роботи приймальної комісії.

І це далеко не всі перспективи, з розвитком цифрових технологій та появою нових програмно-інформаційних рішень можливості розвитку та модернізації лише зростатимуть.

Висновки до розділу

Було розроблено програмну базу для побудови SIS та на її основі створено робочий додаток у вигляді автоматизованого робочого місця викладача. Цей додаток має клієнтську частину зі зручним адаптивним веб-інтерфейсом, котрий не вимогливий до апаратних засобів користувача, та легку в використанні та обслуговуванні серверну підсистему. Як база, так і сам

					ІАЛЦ.045440.004 ПЗ	Арк.
						43
Змін.	Арк.	№ докум.	Підпис	Дата		

додаток мають значні перспективи для застосування та подальшого вдосконалення.

					ІАЛЦ.045440.004 ПЗ	Арк.
						44
Змін.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ТЕХНІЧНІ РІШЕННЯ ВИКОРИСТАНІ В РОБОТІ

3.1 Аналіз доступних інструментів та обґрунтування доцільності використання обраних для розробки засобів (Python Django)

Хоч на сьогоднішній день існує безліч різних мов, на яких можна створювати мережеві додатки, для розробки даного дипломного проєкту було обрано об'єктно-орієнтовану мову Python 3. Вона дозволяє швидко вести розробку завдяки простому синтаксису та динамічній типізації, а також інтерпретованій природі програм, що дозволяє спростити випробування окремих елементів під час розробки. Також Python має обширну бібліотеку стандартних модулів та можливість легко завантажувати додаткові сторонні модулі та фреймворки.

Для виконання мережевих функцій та роботи з базою даних серед можливих мережевих рішень для Python, таких як Flask, Pyramid, Bottle та інші, було обрано веб-фреймворк Django 3.0. Джанго використовує ORM для роботи з базами даних, що не лише зручно для розробників, що надають перевагу об'єктно-орієнтованому стилю програмування, а ще й дозволяє віддалятися від структури власне БД, і за необхідності один і той же код буде працювати з різними БД. За замовчанням Django підтримує PostgreSQL, MySQL, MariaDB, SQLite й Oracle, проте існують сторонні розробки для роботи з іншими БД. Хоч і деякі з альтернатив є більш гнучкими та модульними, проте Django є full-stack фреймворком, котрий по замовчуванню містить багато потужних рішень та засобів для виконання поширених задач в веб-розробці, таких як авторизація та аутентифікація, адміністрування веб-ресурсів та забезпечення базових засобів безпеки за замовчуванням, серед яких захист від міжсайтового скриптингу (XSS), міжсайтової підробки запиту (CSRF), SQL-ін'єкцій та деяких інших загроз[8]. При цьому фреймворк залишається достатньо гнучким для цілей даного курсового проєкту. На додачу до цього,

					ІАЛЦ.045440.004 ПЗ	Арк. 45
Змін.	Арк.	№ докум.	Підпис	Дата		

Django існує вже досить давно, тому у нього наявна обширна документація, а в інтернеті накопичилась значна кількість корисної для розробника інформації про фреймворк.

У якості бази даних для цього дипломного проєкту було використано PostgreSQL 10. В теорії робота додатку можлива з будь-якою з БД, згаданих в попередньому абзаці, серед них було обрано саме PostgreSQL, так як це не комерційний продукт, але при цьому він володіє рядом потужних функцій. Ця БД є повноцінним рішенням для зберігання даних як в тестовому оточенні при розробці, так і в робочому оточенні. Для взаємодії з PostgreSQL Django спирається на засоби `psycopg2`.

Для створення сторінок сайту, крім стандартних засобів у вигляді мови розмітки HTML та стилів CSS, були використані також вбудована мова шаблонів Django (DTL), front-end фреймворк Bootstrap 4 та незначна кількість власних функцій написаних мовою Javascript. Шаблони Django дозволяють змінювати результуючий вміст сторінки в залежності від контексту, переданого з функції відображення, що дозволяє будувати динамічні сторінки замість створення надлишкових HTML файлів, зокрема це зручно при наявності різних груп користувачів. Бібліотека Bootstrap містить значну кількість стилів, шаблонів та інших front-end засобів, призначених для побудови веб-сторінок з динамічною розміткою. Використання Bootstrap дозволяє значно зменшити кількість роботи, необхідної для побудови клієнтської частини системи, при цьому отримуючи достойний вигляд сторінок на будь-яких девайсах.

3.2 Розгляд технологій використаних в розробці та представлення реалізованих технічних особливостей та деталей

Даний курсовий проєкт був розроблений та протестований на комп'ютері з системою Windows 10, інтерпретатором Python версії 3.6 та Django версії 3.0.6.

Файлова структура програми всього проєкта розробки та власне додатку Django відповідає стандартній структурі директорій розробки Django (Рис. 16).

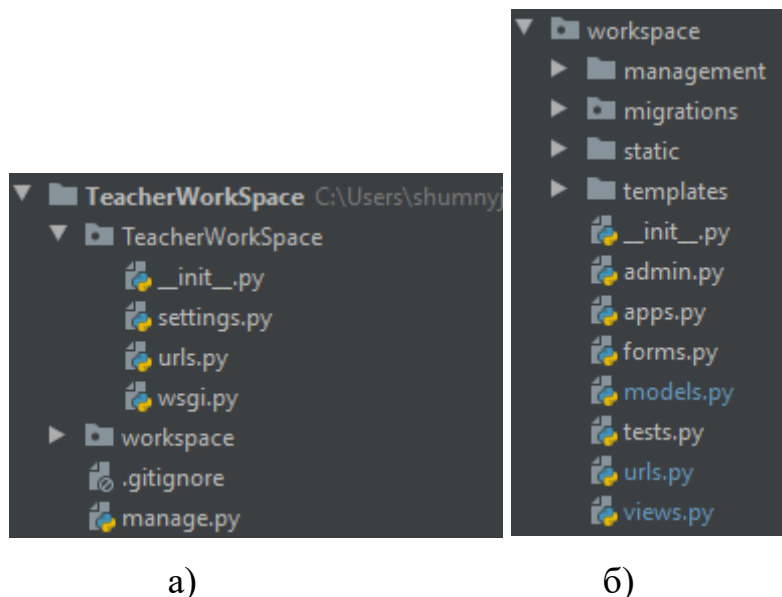


Рисунок 16 - Структура файлів та директорій в проєкті

В структурі загального проєкту (Рис. 16а) є такі елементи:

- manage.py - утиліта командного рядка джанго, використовується для запуску та адміністративних задач (командна оболонка, міграції, тощо);
- TeacherWorkSpace - ключова директорія програмного проєкту, містить в собі файли загальні для всіх додатків:
 - __init__.py - службовий файл для визначення директорії як звичайного (регулярного) пакету Python; порожній;

- settings.py - файл з загальними налаштуваннями програмного проєкту, такими як підключені додатки, налаштування БД, розміщення статичних файлів, middleware та багато інших;
- urls.py - файл для асоціації рядкових посилань (URL) з відповідними відображеннями програми. В даній папці здебільшого слугує для включення додатків в загальний кореневий шлях;
- wsgi.py - файл налагодження зв'язку між програмою та веб-сервером - платформою WSGI, мінімальна конфігурація генерується при створенні програмного проєкту;
- workspace - директорія додатку Django. Саме в ній містяться всі функціональні елементи розробки.

В одному програмному проєкті Django може міститися декілька додатків, але в даному випадку було зроблено рішення, що розділяти функціонал даної розробки на менші модулі недоцільно, тому додаток лише один - workspace.

Структура директорії додатку workspace (Рис 166):

- models.py - файл з моделями - об'єктно-орієнтованими представленнями даних БД у вигляді класів з полями, що відповідають стовпцям таблиць;
- views.py - файл з відображеннями Django - функціями або класами, що обробляють запити, що надходять на сервер;
- urls.py - файл для асоціації URL з відповідними відображеннями додатку. Даний файл підключається до файлу urls.py проєкту;
- forms.py - файл з реалізацією веб-форм у вигляді класів на основі Django;
- admin.py - файл для реєстрації моделей розробленого додатку в панелі адміністратора Django та модифікації панелі адміністратора;
- apps.py - файл для включення конфігурації метаданих додатку;
- tests.py - файл для створення тестів Django;
- __init__.py - аналогічно до такого ж файлу в TeacherWorkSpace;

- management - директорія, в якій міститься директорія commands, в якій містяться файли з користувацькими командами адміністратора Django (в даному випадку - один, cleanup.py);
- migrations - директорія в якій містяться файли міграцій БД. Слугують для створення та зміни таблиць БД відповідно до потреб додатку;
- static - директорія що містить статичні файли додатку, такі як файли стилів, файли з функціями javascript, зображення та іконки, тощо;
- templates - директорія що містить шаблони веб-сторінок додатку - .html файли;

Моделі, визначені у вигляді класів, що наслідують клас `django.db.models.Model` є головним засобом ORM Django. Кожен клас визначає окрему сутність - таблицю в БД, а поля класів визначають назву та тип даних стовпців відповідних таблиць. Приклад структури моделі на Рис.17.

```
class Attendance(models.Model):
    """
    Stores a mark for attendance on the lesson, related to
    :model:`workspace.Student` and :model:`workspace.Lesson`
    """
    lesson = models.ForeignKey(Lesson, on_delete=models.CASCADE)
    student = models.ForeignKey(Student, on_delete=models.CASCADE)
    lat = models.FloatField(blank=True)
    lon = models.FloatField(blank=True)

    def __str__(self):
        return self.lesson.discipline.name + ' ' + str(self.student)
```

Рисунок 17 - Код окремого класу-моделі Attendance

В додатку workspace визначено 15 моделей:

- Student - профіль учня/студента, базується на моделі `django.auth.User` за допомогою поля відношення один-до-одного, та містить інформацію про учня, таку як номер студентського білета або групу;
- Teacher - профіль викладача, базується на моделі `django.auth.User` за допомогою поля відношення один-до-одного та містить інформацію про викладача, таку як вчене звання та кваліфікація;

- StudentGroup - модель що представляє навчальну групу студентів/учнів;
- AcademicCourse - модель, що визначає навчальний курс - сутність, що поєднує викладача, навчальну групу та навчальну дисципліну. Вона є основою для контролю та оцінювання, дозволяючи відрізнати навчальний процес з одного предмету у різних груп;
- CourseAccess - модель, що визначає доступ додаткових викладачів (асистентів, лаборантів, тощо) до курсу, викладачі, зв'язані через CourseAccess мають дещо інші права доступу до контрольних заходів та оцінок, ніж у головного викладача курсу, тому було зроблено рішення на користь такої реалізації замість відношення багато-до-багатьох з викладачами в навчальному курсі;
- Discipline - модель, що визначає навчальну дисципліну. Визначає довідкову таблицю;
- LessonType - модель, що визначає тип заняття або освітнього заходу - лекція, лабораторна робота, екзамен, тощо. Визначає довідкову таблицю;
- Location - модель, що визначає локацію - місце проведення заняття, включно з його координатами;
- Lesson - модель, що визначає окреме заняття або подібний навчальний захід. Не зв'язана з навчальним курсом, натомість має зв'язок з викладачем та дисципліною для більш високої потенційної гнучкості розкладу;
- Attendance - модель, що визначає факт відвідування студентом відповідного заняття;
- AttendanceToken - модель, що визначає тимчасовий об'єкт, відносно якого реєструється відвідування відповідного заняття;
- ControlCategory - модель, що визначає тип контрольного заходу - контрольна робота, лабораторна робота, залік, тощо;

					ІАЛЦ.045440.004 ПЗ	Арк.
						50
Змін.	Арк.	№ докум.	Підпис	Дата		

- ControlEntity - модель, що визначає окремий контрольний захід в межах навчального курсу. Зв'язана з відповідним курсом і є основою для виставлення оцінок;
- Mark - модель, що визначає оцінку. Оцінка виставляється окремому студенту за окремий контрольний захід;
- Notification - модель, що визначає сповіщення. На кожну подію, що створює сповіщення створюється лише один екземпляр, котрий зв'язується з необхідною кількістю користувачів зв'язком багато-до-багатьох через проміжну таблицю.

Детальніше розглянути структуру БД мого додатку можна на Додатку 1 Д1. Крім моделей мого додатку проєкт також створює декілька таблиць для моделей аутентифікації підключених стандартних додатків Django - користувачів, дозволів, дозволів користувачів, користувацьких груп, дозволів груп, а також таблицю з моделями `django_content_type`, таблицю з застосованими міграціями БД `django_migrations`, таблицю що містить користувацькі сесії `django_session` та таблицю з логом адміністратора `django_admin_log`. Варто зазначити, що до кожної таблиці Django додає цілочисельне поле `id`, яке виступає в ролі первинного ключа, якщо при формуванні моделі та таблиці не було вказано діяти інакшим чином.

Обробка синхронних запитів в проєкті базується на системі відображень (`views`), коли функція-відображення або клас, котрий наслідує відповідний базовий клас відображення, ставиться у відповідність деякому URL на сервері. При надходженні запиту за певним шляхом сервер викликає відповідну функцію-відображення або метод `.as_view()` відповідного класу, котрий відповідним чином обробляє запит, і на основі даних з нього повертає відповідну відповідь, частіше за все - веб-сторінку, проте залежно від ситуації може бути повернута помилка, відбутись переадресація на іншу сторінку (відображення) або може бути повернутий інший тип даних, не HTML. Згадані в пункті 2.2.1 сторінки всі мають свої власні відображення:

					ІАЛЦ.045440.004 ПЗ	Арк.
						51
Змін.	Арк.	№ докум.	Підпис	Дата		

- index_view - індексна сторінка;
- MyLoginView(auth_views.LoginView) - сторінка авторизації;
- profile_view - сторінка профіля;
- edit_profile_view - сторінка редагування профіля;
- schedule_view - сторінка розкладу;
- lesson_view - сторінка заняття (Додаток 1 ДЗ);
- lesson_edit_view - сторінка редагування заняття;
- lesson_add_view - сторінка додавання нового заняття;
- marks_menu_view - сторінка зі списком курсів;
- marks_entities_view - сторінка зі списком контрольних заходів курсу;
- marks_add_entities_view - сторінка для додавання контрольного заходу;
- marks_change_ent_view - сторінка для зміни контрольного заходу;
- marks_detail_view - сторінка окремого контрольного заходу з оцінками;
- marks_table_view - сторінка з табличним відображенням сторінки;
- batch_add_view - сторінка для додавання занять партіями.

Крім них в додатку присутні відображення, котрі не повертають користувачу власного вмісту, а лише здійснюють дії на сервері (наприклад, видалення даних з БД) та перенаправляють користувача на іншу сторінку. До цих відображень відносяться:

- logout_view - відображення для деавторизації;
- marks_rm_ce_view - відображення для видалення окремого контрольного заходу, викликається викладачем відповідною кнопкою з вікном підтвердження на сторінці контрольних заходів (Рис. 18);

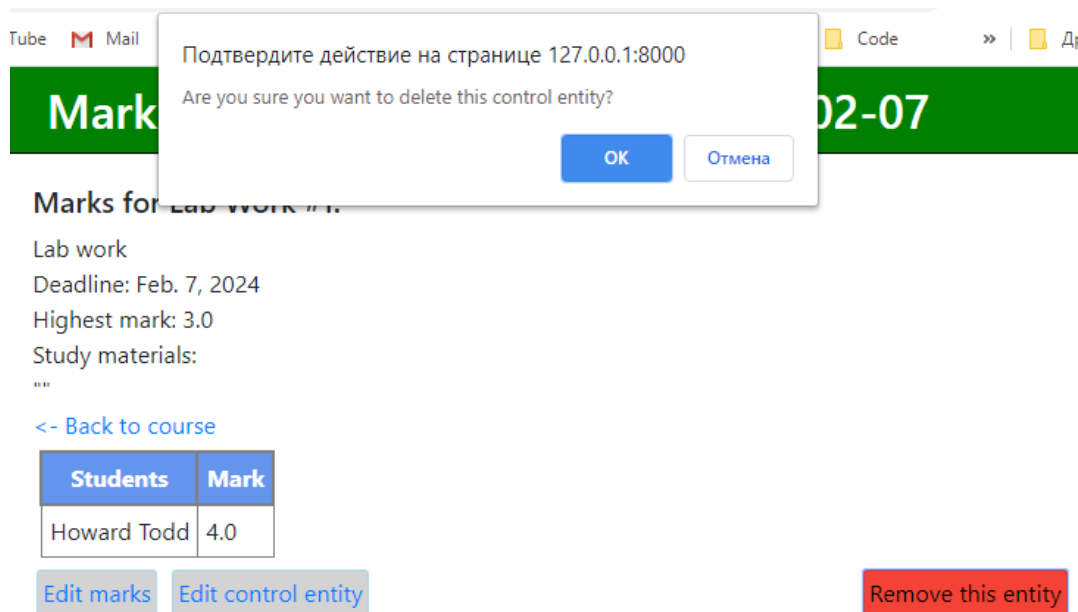


Рисунок 18 - Підтвердження перед звертанням до marks_rm_se_view

- rm_notification - відображення для видалення сповіщення у окремого користувача - видаляється зв'язок користувача зі сповіщенням з проміжною таблиці. Якщо час дії сповіщення закінчився або його видалив у себе останній користувач, до якого мало відношення це сповіщення, то воно видаляється повністю;
- webhook_github_view - відображення, створене для отримання запитів з вебхука Github та генерації відповідних сповіщень (Додаток 1 Д2);
- ShowLocationView - відображення, що наслідує клас RedirectView та слугує для перенаправлення запитів на відображення локації на відповідне зовнішнє посилання. Дозволяє мати єдине посилання для цієї задачі, котре не зміниться при зміні сервісу відображення положення.

Функції відображень крім запиту можуть приймати в якості параметрів іменовані фрагменти відповідного URL - конфертери шляху (path converters), визначені відповідним шаблоном в urls.py, далі ці параметри відіграють свою роль в коді відображення. Як приклад роботи цього механізму можна навести шлях /workspace/lesson/<int:l_id>/, з нього в відображення lesson_view передається параметр l_id, що визначає id заняття, що розглядається. Таким

чином можуть використовуватись стандартні конвертери Django, користувацькі конвертери, котрі можна створити самостійно або регулярні вирази.

Всі запити, що містять в собі дані форм використовують метод POST, так як наразі в додатку немає ситуацій де було б доречно використовувати метод GET - всі форми в додатку наразі націлені на здійснення більш-менш постійних змін в системі, тому було вирішено користуватись більш безпечним методом передачі даних в тілі запиту POST, крім того POST-запит здатний передавати більшу кількість даних, що важливо зокрема для запиту з вебхука Github та форми виставлення оцінок. В додатку при роботі з формами замість прямої роботи з даними з тіла запиту використовуються форми Django. Використання системи форм Django значно полегшує роботу розробника як при відображенні форм, так і обробці отриманих даних. Форми додатку представлені у вигляді класів, що наслідують відповідні базові класи `django.forms.Form` або `django.forms.ModelForm`, та містять поля, що відповідають полям результуючої веб-форми. В них можливе налаштування початкових даних форми, допустимих для вибору значень, типу віджета для заповнення (текстове поле, список, тощо), обов'язковості заповнення поля, підписів полів та інших важливих параметрів форми. В процесі розробки було надано перевагу здебільшого наслідуванню від звичайних форм (`django.forms.Form`), котрі хоч і потребують більшого об'єму написаного коду для обробки та збереження, проте є більш гнучкими в роботі. В свою чергу модельні форми (`django.forms.ModelForm`) зв'язуються з конкретною моделлю, та можуть містити лише поля, аналогічні полям відповідної моделі, що пришвидшує процес зберігання, але дещо обмежує можливості обробки таких форм. Варто згадати про можливість наслідування класів форм, при якому можна створити новий клас форми, додаючи, видаляючи або змінюючи поля класу форми, котрий наслідується. Наслідування форм було використано, наприклад, при створенні форм для редагування профіля: форми для студента та для викладача наслідують базову форму користувача (Рис 19).

					ІАЛЦ.045440.004 ПЗ	Арк.
						54
Змін.	Арк.	№ докум.	Підпис	Дата		

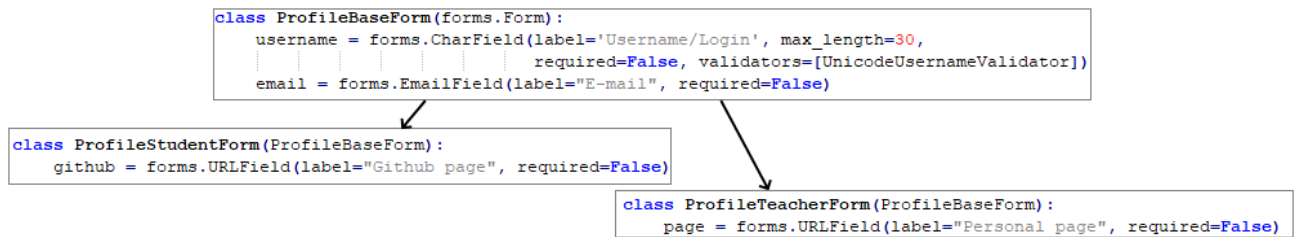


Рисунок 19 - Наслідування форм редагування профіля

Крім того є можливість перезаписати успадковані методи для пристосування форм до певних цілей, наприклад змінювати список доступних для вибору варіантів певного поля залежно від певної ознаки за допомогою перезаписування методу `__init__()`, як це зроблено в формі `MarkForm` в даному проєкті (Рис 20).

```

class MarkForm(forms.Form):
    student = forms.ModelChoiceField(label='Student', queryset=wsm.Student.objects.none(),
                                     empty_label=None, widget=forms.HiddenInput)
    mark = forms.DecimalField(label='Mark', max_digits=4, decimal_places=1, required=False)

    def __init__(self, *args, **kwargs):
        try:
            qs = kwargs.pop('student_group')
        except KeyError:
            qs = None
        finally:
            super().__init__(*args, **kwargs)
            if qs:
                self.fields["student"] = forms.ModelChoiceField(label='Student', queryset=qs, empty_label=None,
                                                                widget=forms.HiddenInput)

```

Рисунок 20 - Внутрішня реалізація форми `MarkForm`

Ще одним інструментом системи форм є набори форм (`formsets`). Вони дозволяють здійснювати аналогічним чином обробку відразу кількох однакових форм в одному запиті з однієї сторінки, при цьому можна змінювати кількість активних форм на сторінці в заданих заздалегідь межах. В додатку з даного дипломного проєкта набори форм використовуються для виставлення оцінок з використанням форми `MarkForm` (Рис 21).

```

if request.method == 'POST':
    MarkFormSet = formset_factory(wsf.MarkForm, extra=len(a_students))
    formset = MarkFormSet(request.POST, form_kwargs={"student_group": a_students})
    if formset.is_valid():
        # Valid forms
        t = datetime.date.today()
        for form in formset:
            if form.cleaned_data["mark"] is not None:
                # if new mark is set
                mrk, created = a_entity.mark_set.get_or_create(student=form.cleaned_data["student"],
                                                                defaults={"reason": a_entity})
                mrk.mark = form.cleaned_data["mark"]
                mrk.date = t
                mrk.save()
            context["c_message"] = "Successfully saved"
            return redirect('workspace:marks_detail', c_id=a_course.id, e_id=a_entity.id)
        else:
            context["c_message"] = "Error in form"
            context["student_forms"] = formset
    else:
        MarkFormSet = formset_factory(wsf.MarkForm, extra=len(a_students))
        formset = MarkFormSet(form_kwargs={"student_group": a_students})
        i = 0
        for form in formset:
            form.initial = {"student": a_students[i]}
            i += 1
        context["student_forms"] = formset

```

Рисунок 21 - Створення та використання наборів форм при отриманні даних в запиті зі сторінки та при ініціалізації сторінки з формою

Форми та набори форм надають спеціальні методи для побудови їх відображення з шаблону, такі як `.as_ul`, `.as_p` та `.as_table` для відображення всіх полів відповідно у вигляді списку, абзаців або таблиці, проте якщо потрібна інша схема розміщення елементів форми на шаблоні, то це також досить легко здійснити.

Як вже було сказано, більшість відображень повертають відповідь у вигляді HTML-сторінки в `HttpResponse`, проте в додатку для цього в більшості випадків використовується не власне повернення `HttpResponse`, а результат виклику допоміжної функції `render()`. Дана функція комбінує існуючий шаблон сторінки з контекстом, переданим із відображення, повертаючи `HttpResponse` з готовою HTML сторінкою. Шаблони є одним з ключових елементів ефективної розробки клієнтської частини додатку. Мова шаблонів Django (DTL) надає значну кількість стандартних тегів та фільтрів для шаблонів, за допомогою яких модифікується відображення контексту та іншого вмісту відповідної веб-сторінки, серед яких теги для контролю уникнення небезпечних символів

(autoescaping), блоки що реалізують наслідування та включення інших шаблонів, організації умовних конструкцій та циклів, форматування виводу елементу, тощо. Також є можливість створювати власні теги та фільтри, хоч в цьому додатку вона не знадобилась. Шаблони Django, на додаток до роботи зі структурами даних Python, оптимізовані для роботи з рештою елементів фреймворку, такими як моделі, форми, набори QuerySet, що вигідно відрізняє його від аналогів. Передача в шаблон контексту, утвореного змінними різних складних типів на зразок словників, об'єктів класів або моделей є цілком доречним, так як в DTL присутній механізм пошуку через оператор “.” (крапка). Коли система шаблонів зустрічає крапку, вона намагається знайти поле з відповідним іменем в змінній контексту:

1. Як елемент за ключем в словнику
2. Як атрибут класу;
3. Як індекс списку.

Цей механізм широко використовується в даному додатку, наприклад при використанні виразу “`{{auth_student.student.group.name}}`” в шаблоні система вставить в сторінку атрибут *group* атрибуту *student* контекстної змінної *auth_student*, що являє собою об'єкт моделі User. При цьому, так як обробка шаблону відбувається на сервері, об'єм даних, що надсилається користувачу з відповіддю по суті ніяк не залежить, від того у якому вигляді були передані контекстні змінні. При цьому не варто покладатись лише на вищезгадані елементи системи шаблонів, в даному додатку є спеціальна функція *student_list_with_marks*, котра повертає список з кортежів (*студент, оцінка*), залишаючи оцінку порожнім полем, якщо вона не виставлена. Відображення такого списку за допомогою стандартних QuerySet є вельми неефективним, при цьому відображення такого списку на сторінці є набагато більш наглядним для викладача, ніж список лише зі студентів, котрі мають оцінку (який можна легко отримати у вигляді QuerySet).

Так як контекст є основним критерієм, що впливає на остаточний вигляд шаблону, то модифікація контексту є одним із ключових моментів в забезпеченні гнучкості додатку. В розробленому додатку для більшого контролю над ходом обробки даних хоч і було використано функціональні відображення, котрі загалом менш гнучкі, проте метою саме змінної CUSTOM_CONTEXT_FUNC та функції get_custom_context було на прикладі показати одне з допустимих рішень для розширення існуючих функціональних відображень.

Також в розробленому додатку активно використовується механізм наслідування шаблонів, що полягає в заміні окремих визначених блоків шаблону, що наслідується, вмістом блоків з такою самою назвою шаблону, що наслідує. Це дозволяє повторно використовувати код шаблонів для ще більшого підвищення ефективності розробки клієнтської підсистеми. Детальніше ознайомитися з механізмом наслідування можна на прикладі з Додатку 1 Д4.

Висновки до розділу

Обрані мною засоби для створення даного дипломного проєкту надають значну кількість інструментів, що пришвидшують процес розробки та заохочують написання коду, котрий можна використати повторно. В процесі розробки здебільшого дотримувались цих самих принципів. Під час розробки якомога ширше використовувався згаданий в розділі арсенал наданих засобів, але при цьому не приносилась в жертву ефективність роботи готового додатку.

					ІАЛЦ.045440.004 ПЗ	Арк.
						58
Змін.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4. ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ

Всі тести проводились на машині з процесором AMD Ryzen 2600X Six-core 3.6 GHz, об'ємом оперативної пам'яті 16 Гб під системою Windows 10 в DEBUG режимі роботи сервера.

Для виявлення можливих слабких місць системи було проведено стрес-тестування зі значним навантаженням на сервер. Тестування проводилось з використанням програми Webserver Stress Tool шляхом симуляції значної кількості кліків-переходів на декілька сторінок сайту від значної кількості емульованих користувачів. Перед тестуванням максимальну кількість з'єднань з БД було встановлено на значення 300. Звернення до сервера відбувалось в режимі локального хоста (хост сайту - 127.0.0.1:8000).

В даному випадку тестування було використано схему зі зростаючим навантаженням - від 0 до 400 користувачів протягом 3 хвилин, з умовно-випадковою частотою кліків-переходів з приблизним середнім значенням затримки між кліками в 2 секунди по 4 різним URL додатку.

За графіками на Рис. 22, 23 бачимо, що при наближенні до 400 одночасних користувачів, починають виникати помилки сервера. При розгляді логів було виявлено, що основною причиною помилок на сервері є неможливість створити нове з'єднання з БД через обмеження на максимальну кількість з'єднань. Так як користувачі здійснюють зв'язок з БД не зовсім в один і той самий момент, то така поведінка не з'являється раніше, в районі 300 користувачів.

					ІАЛЦ.045440.004 ПЗ	Арк. 59
Змін.	Арк.	№ докум.	Підпис	Дата		

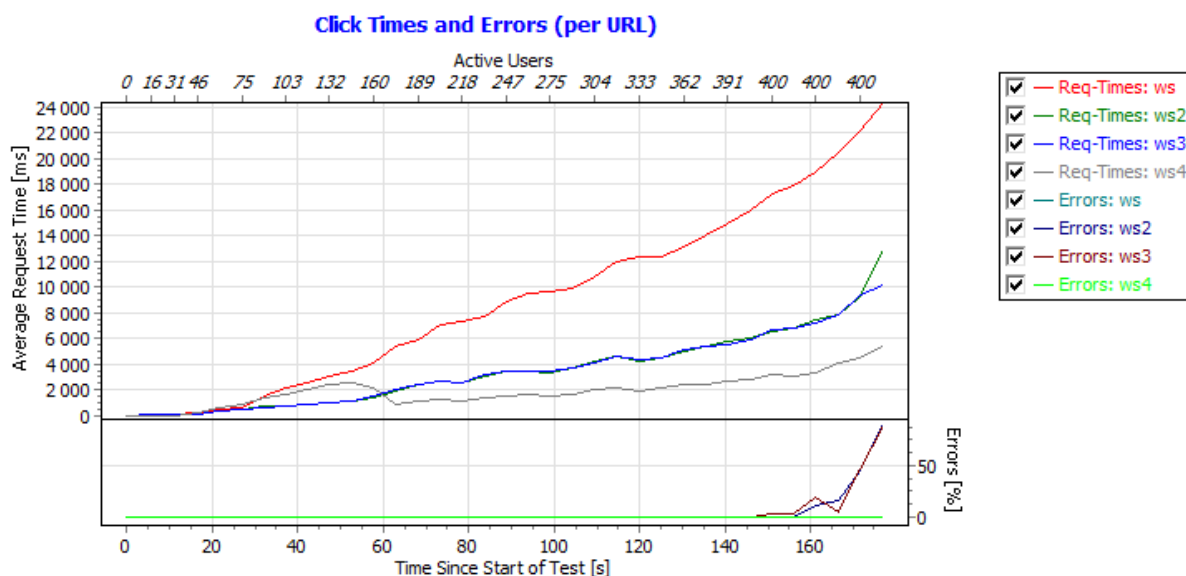


Рисунок 22 - Графік часу, що витрачається на обробку одного переходу та відсоток помилок при переходах при відповідному навантаженні

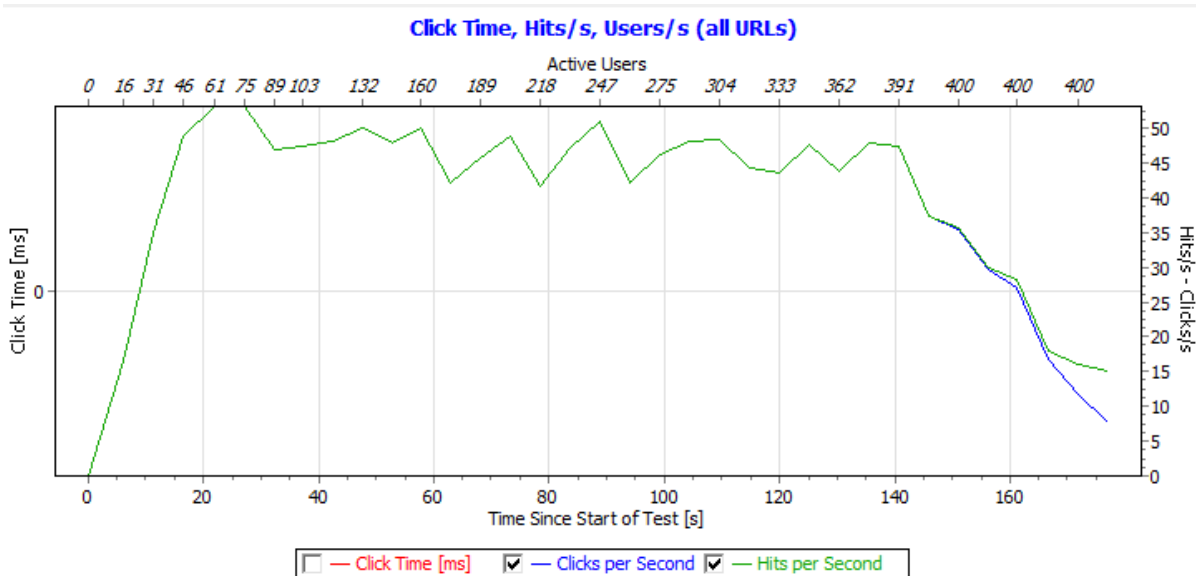


Рисунок 23 - Відповідний обернений графік, що показує сумарну кількість переходів за секунду при відповідному навантаженні

Поява помилок в свою чергу значно знижує кількість дійсних кліків та ще більше підвищує середній час запиту, котрий і так стабільно зростає через звичайне збільшення користувацького навантаження на сервер. На Рис. 24, 25, 26 видно злами на графіках на моменті коли починають з'являтися помилки (~391 користувач).

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045440.004 ПЗ

Арк.
60

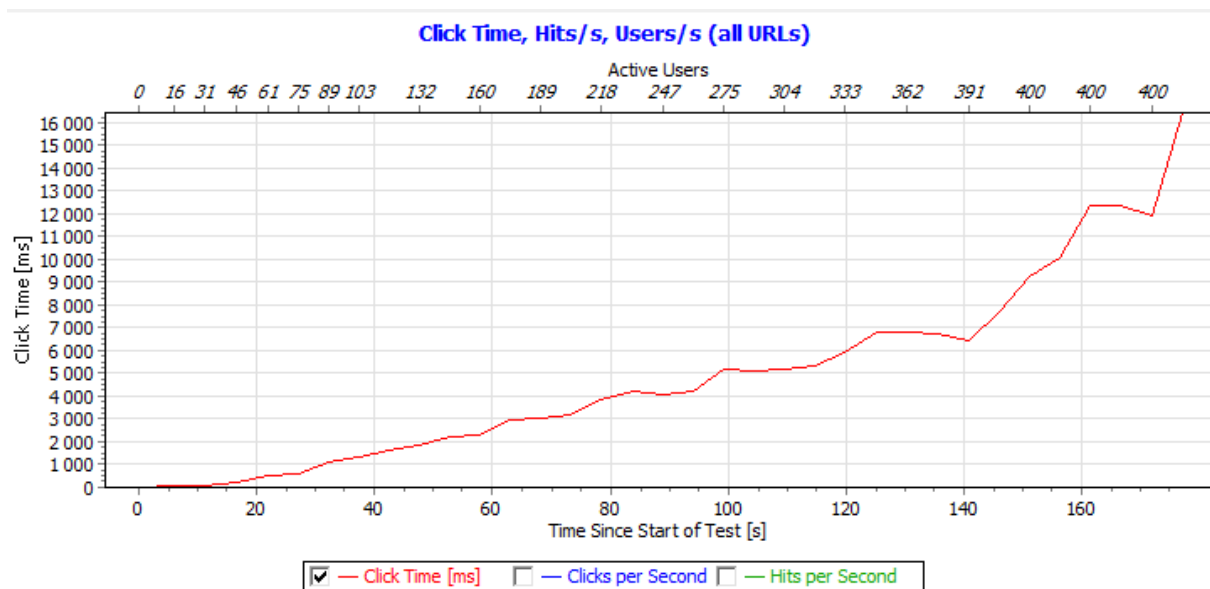


Рисунок 24 - Залежність часу на клік від кількості користувачів

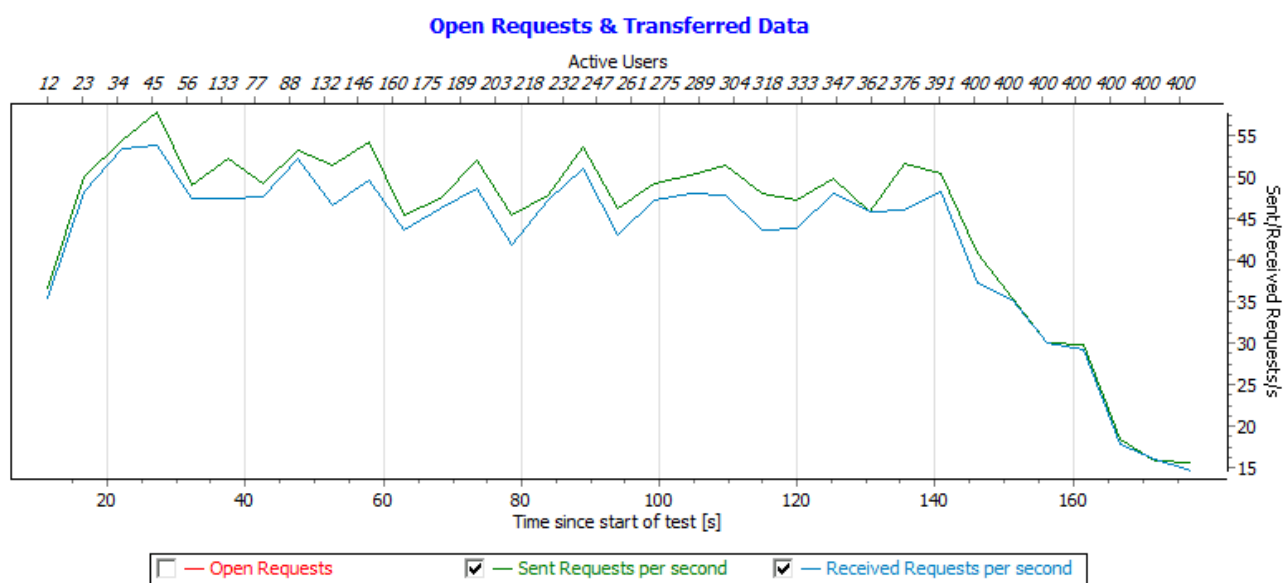


Рисунок 25 - Залежність кількості надісланих/отриманих за секунду запитів від кількості користувачів

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045440.004 ПЗ

Арк.
61

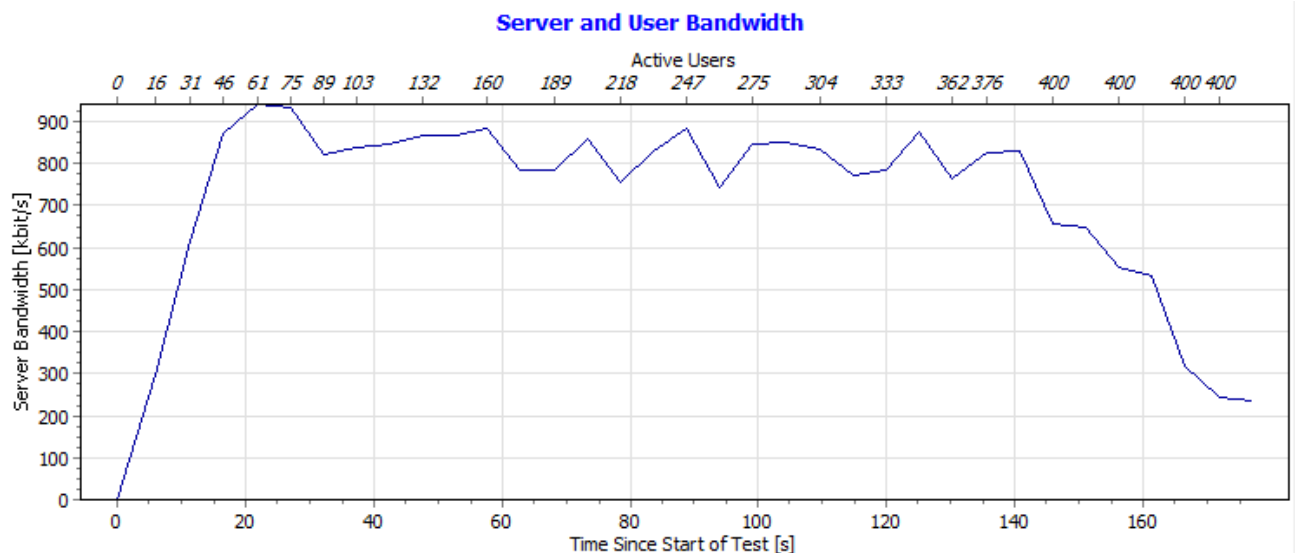


Рисунок 26 - Графік залежності вихідної пропускної здатності сервера від кількості користувачів

Для уникнення подібної ситуації важливо при розміщенні додатку на сервері правильно налаштувати сам додаток та БД. Помилок, що відносилися б до проблем внутрішньої реалізації мого додатку або фреймворку під час даного стрес-тесту виявлено не було виявлено не було.

Так як пропускна спроможність будь-якого сервера обмежена, то зі збільшенням кількості одночасних користувачів сервер змушений розділяти канал передачі трафіку між ними, відповідно знижуючи швидкість передачі даних окремим користувачам, як зображено на графіках з Рис 28, 29. Це призводить до пропорційного поступового зростання середнього часу на обробку окремого запиту як зображено на Рис 23, 24 на ділянці до виникнення помилок.

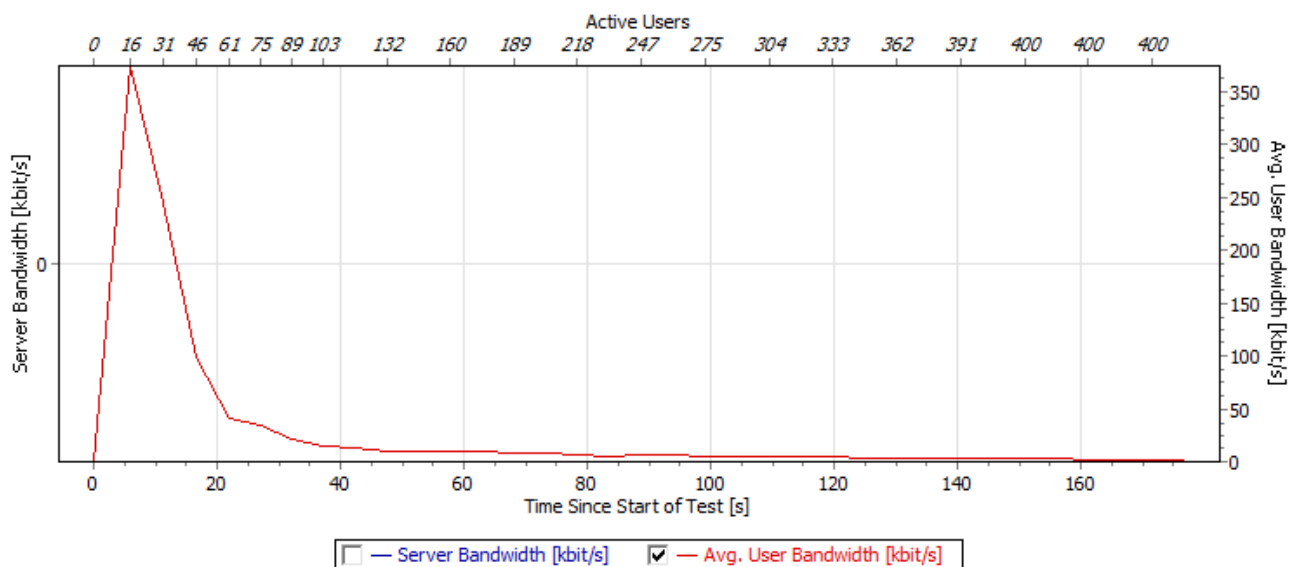


Рисунок 27 - Графік залежності середнього трафіку, виділеного на користувача від кількості користувачів

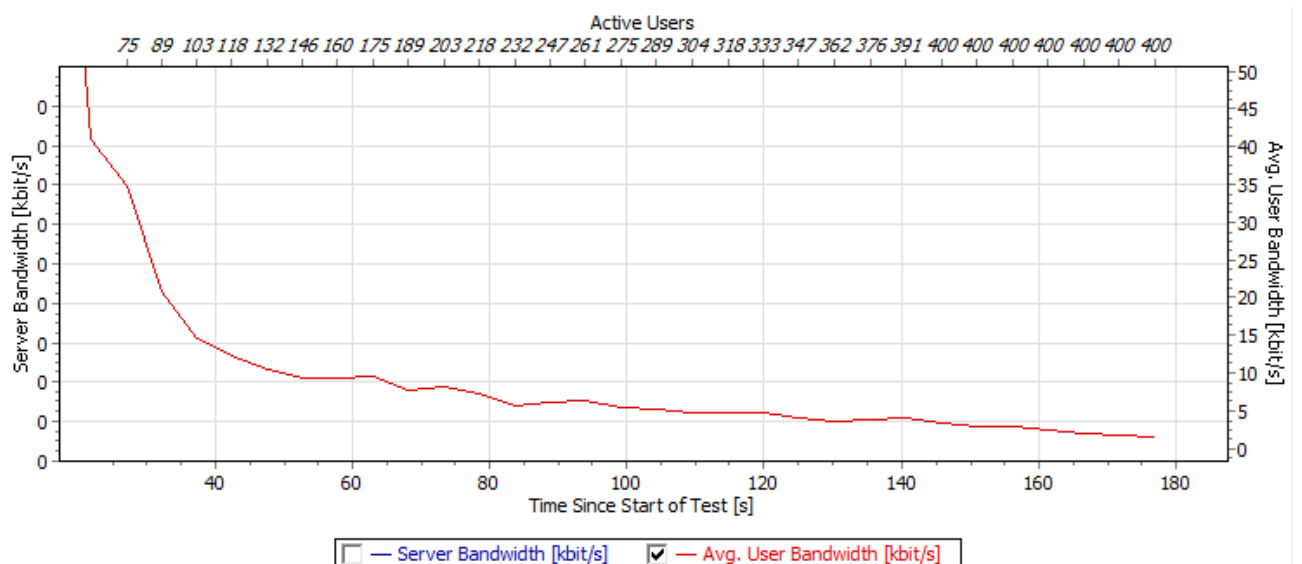


Рисунок 28 - Аналогічний до Рис 27 графік, без пікових значень

На Рис. 27 приведена зведена діаграма статистики часу завантаження сторінок у користувачів протягом часу тестування. Схожа тенденція буде спостерігатись у будь-яких серверних додатків, тому при розміщенні додатку на сервері для подальшої роботи важливо правильно визначити цільову кількість одночасних користувачів та обрати технічне забезпечення хостингу, що буде задовольняти відповідні потреби.

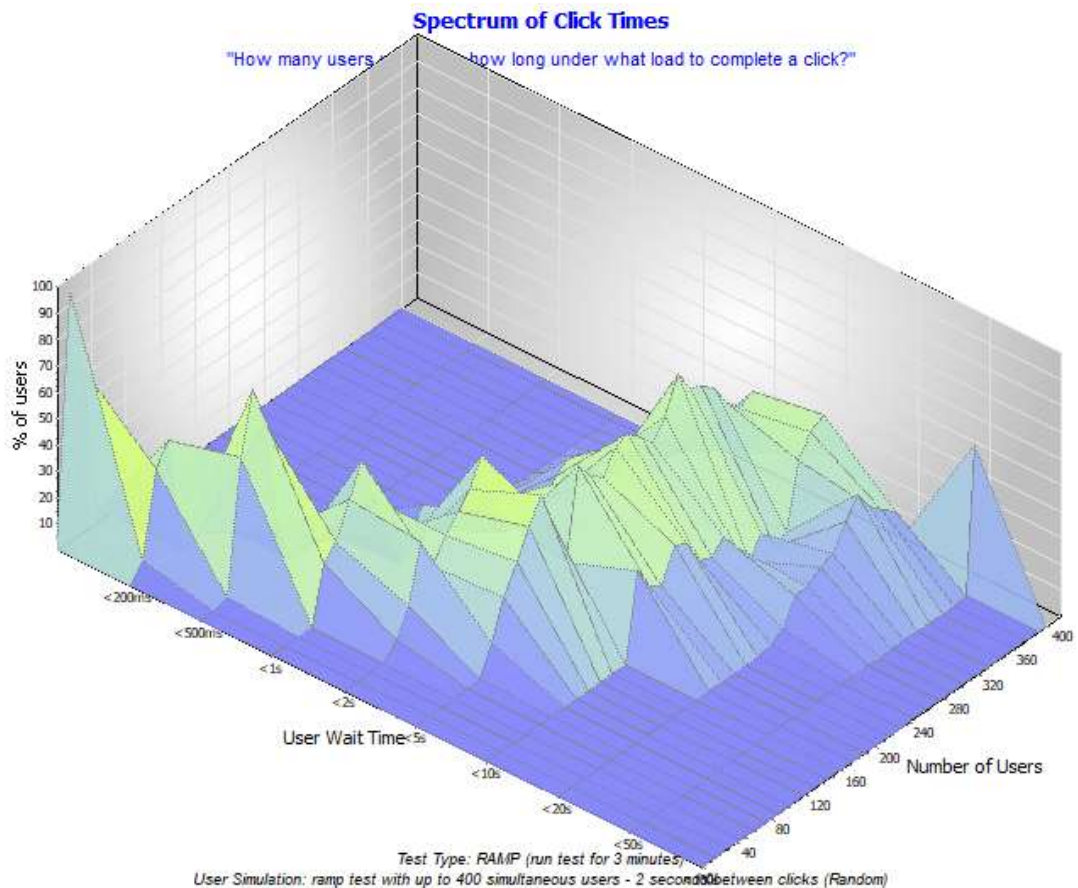


Рисунок 29 - Діаграма розподілу користувачів за часом очікування завантаження сторінки при відповідному навантаженні

Було вирішено провести деяке тестування швидкодії сайту. Для оцінки було зібрано значення базових параметрів швидкодії сторінки сайту, таких як час FCP (first contentful paint) та час події onLoad (L) р засобів відлагодження браузера Google Chrome. Отримані результати були порівняні з аналогами (сторінками з подібним змістом). На Рис 30 приведено результати тестування при звертанні до додатку через тунель сервісу ngrok, доступ до якого здійснюється в регіоні US (Сполучених Штатах Америки), а для мінімізації впливу сторонніх факторів для порівняння були обрані схожі за вмістом сторінки, розміщені на серверах в тому ж регіоні.

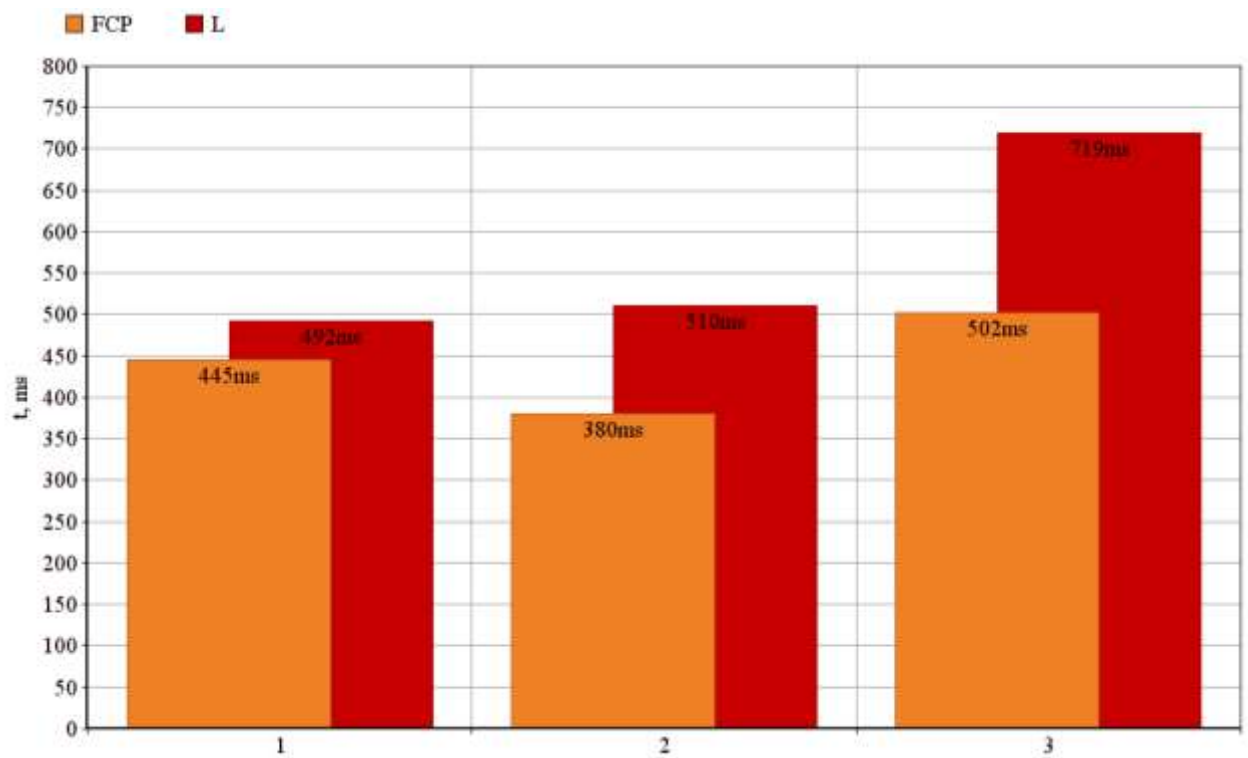


Рисунок 30 - Швидкодія сторінок на регіоні US

Використані сторінки:

1. <http://4292e723.ngrok.io/workspace/>
2. https://www.uchicago.edu/about/non_discrimination_statement/
3. <https://www.usnx.com/helpdesk/helpful-links>
4. <https://ecampus.kpi.ua/home>
5. <http://127.0.0.1:8000/workspace/>

На Рис. 31 приведено результати аналогічного тестування, проте звертання до мого додатку відбувається через localhost, а для порівняння взятий схожий ресурс, розміщений на якомога ближчому сервері - Кампус КПІ.

Результати тестів на швидкодію та подальше порівняння показали, що швидкодія сайту мого додатку знаходиться в межах норми, навіть якщо врахувати неточності, що можуть бути спричинені різницею в наповненні сторінки та точному місці знаходження сервера.

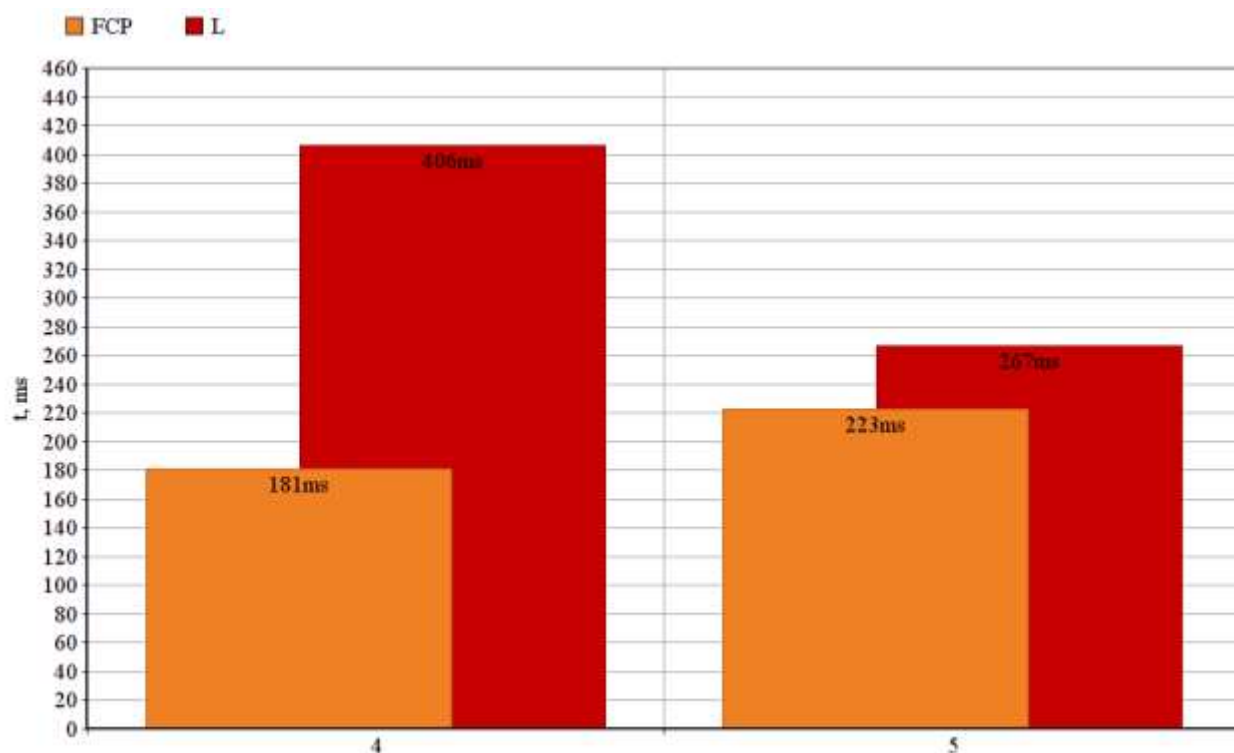


Рисунок 30 - Швидкодія сторінок в локальному регіоні

Висновки до розділу

Хоч власне додаток показує задовільну роботу під час стрес-тестів, але для розміщення на “живих” серверах для роботи в реальних умовах для мого додатку та подібних йому необхідно використовувати апаратні засоби, що будуть витримувати відповідне навантаження. Не менш важливим є налаштування додатку та БД, з якою він взаємодіє, до можливостей відповідних апаратних засобів.

ВИСНОВКИ

При виконанні даного дипломного проєкту було розглянуто існуючі сучасні рішення в сфері учнівських інформаційних систем та цифрових засобів для контролю над освітнім процесом для навчальних закладів. Після проведення цього дослідження та аналізу зібраних даних, було зроблено висновок, що наразі можливості, що надають інформаційні технології, використовуються в даній сфері неповноцінно.

Було створено автоматизоване робоче місце викладача університету, що містить рішення для виконання повсякденних освітньо-організаційних задач на основі методів та функцій, що базуються на найновіших тенденціях в сфері цифрових технологій, таких як повсюдне використання смартфонів, ноутбуків та інших компактних цифрових пристроїв, імплементація принципу BYOD та інтеграція додатків та сервісів з різноманітних сфер між собою для підвищення продуктивності роботи користувача.

При розробці додатку, представленого в даному дипломному проєкті, я, як розробник, орієнтувався на відповідні сучасні тенденції розробки веб-додатків, ключовими з яких в даному проєкті є кросплатформеність вкупі з адаптивним дизайном та елементи модульної структури додатку, що сприяє потенційному розширенню та адаптації розробки.

Отримана система має досить зручний дизайн та цілком успішно виконує поставлені перед нею задачі, за умов, коли їй надані технічні ресурси, що відповідають попиту у користувачів. Це приводить нас до висновку, що подальша інтеграція елементів, що базуються на найновіших здобутках ІТ-сфери є цілком доцільною та продуктивною.

					ІАЛЦ.045440.004 ПЗ	Арк.
						67
Змін.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ляхощка Л.Л. Роль цифрових технологій в освітньому процесі закладу післядипломної освіти // П'ята міжнародна науково-практична Інтернет-конференція «Регіональні культурні, мистецькі та освітні практики» 14-15 березня 2018 року. м. Переяслав-Хмельницький: тези доповідей.
2. Цифрова адженда України – 2020 («Цифровий порядок денний» – 2020) Концептуальні засади. Першочергові сфери, ініціативи, проекти «цифровізації» України до 2020 року : URL : <https://uccr.org.ua/uploads/files/58e78ee3c3922.pdf> (дата звернення 25.04.2020).
3. Попова Н. Е., Чикова О. А. ТЕХНОЛОГИИ ДИСТАНЦИОННОГО ОБУЧЕНИЯ КАК ИННОВАЦИЯ В ПРОЦЕССЕ РЕАЛИЗАЦИИ ОБРАЗОВАТЕЛЬНЫХ СТАНДАРТОВ НОВОГО ПОКОЛЕНИЯ. *Вестник Новосибирского государственного педагогического университета*. 2014. №2 (18). С. 17-26.
4. Гінкул А., Магурян Я. SMART-ОСВІТА ТА ОСОБЛИВОСТІ ЇЇ ВПРОВАДЖЕННЯ В УКРАЇНІ. *Smart-освіта: ресурси та перспективи* : матеріали III Міжнар. наук.-метод. конф. :ф. (Київ, 7 грудня 2018 р.) : тези доповідей.— К. : Київ. нац. торг.-екон. ун-т, 2018. С. 18-20.
5. Гройсман В. Концепція розвитку цифрової економіки та суспільства України на 2018-2020 роки. Схвалено розпорядженням Кабінету Міністрів України від 17 січня 2018 р. № 67-р
6. O. Dziabenko, N. Morze 3D MAPPING OF UKRAINIAN EDUCATION SYSTEM. *Modernization of Pedagogical Higher Education by Innovative Teaching Instruments (MoPED)*. Borys Grinchenko Kyiv University. 15 липня 2018.

					ІАЛЦ.045440.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		68

7. The Django admin site. *Django documentation*. URL:

<https://docs.djangoproject.com/en/3.0/ref/contrib/admin/> (дата звернення 2.05.2020).

8. Security in Django. *Django documentation*. URL:

<https://docs.djangoproject.com/en/3.0/topics/security/> (дата звернення 2.05.2020).

					ІАЛЦ.045440.004 ПЗ	Арк.
						69
Змін.	Арк.	№ докум.	Підпис	Дата		