

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

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

_____ Олександр КОВАЛЬ

«___» _____ 2021 р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Комп'ютерний еколого-економічний
моніторинг процесів та систем»**

спеціальності 122 «Комп'ютерні науки»

**на тему: «Інструментальні засоби формування навчальних планів для програм
подвійного диплому»**

Виконав:

студент ІV курсу, групи ТМ-71
Трамбович Антон Русланович

Керівник:

Доцент, к.е.н.,
Гусєва Ірина Ігорівна

Консультант:

Рецензент:

Доцент, к.т.н.,
Веремійчук Юрій Андрійович

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____

Київ — 2021 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки: 122 Комп'ютерні науки

Спеціалізація: Комп'ютерний моніторинг та геометричне моделювання процесів і систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр КОВАЛЬ

(підпис)

” ____ ” _____ 2021р.

ЗАВДАННЯ

на дипломну роботу студенту

Трамбовичу Антону Руслановичу

(прізвище, ім'я, по батькові)

1. Тема роботи Інструментальні засоби формування навчальних планів для програм подвійного диплому

керівник роботи _____ Гусева Ірина Ігорівна, к.е.н.

(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ”24” травня 2021р. № **1267-с**

2. Строк подання студентом роботи _____ 08.06.2021р.

3. Вихідні дані до роботи мова програмування PHP, фреймворк Bootstrap, середовище розробки Brackets

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) _____ створення веб-додатку для роботи з файлами програми подвійного диплому та формування узгоджених навчальних планів. Проаналізувати існуючі методи роботи з файлами та порівняння даних. Розробити універсальну програму для роботи _____ з _____ таблицями _____ Excel.

5. Перелік ілюстративного матеріалу

Актуальність, Постановка задачі, Бібліотеки для роботи з файлами, Існуючі анаоги, Архітектура системи, Модель бази даних, Діаграма прецедентів, Сторінки реєстрації та авторизації, Інтерфейс сторінки авторизації, Інтерфейс сторінки завантаження, Інтерфейс сторінки порівняння, Інтерфейс сторінки перегляду та видалення, Висновки.

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------|-------------------------------------------|----------------|------------------|
| | | завдання видав | завдання прийняв |
| | | | |

7. Дата видачі завдання ” 8 ” _____ жовтня _____ 2020 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів виконання дипломної роботи | Термін виконання етапів роботи | Примітки |
|-------|-----------------------------------------------------|--------------------------------|----------|
| 1. | Затвердження теми роботи | 15.10.2020 | |
| 2. | Вивчення та аналіз задачі | 16.10.2020- 16.12.2020 | |
| 3. | Розробка архітектури та загальної структури системи | 09.03.2020- 23.03.2020 | |
| 4. | Розробка структур окремих підсистем | 24.03.2020- 10.04.2020 | |
| 5. | Програмна реалізація системи | 14.04.2020- 10.05.2020 | |
| 6. | Оформлення пояснювальної записки | 18.05.2020- 02.06.2020 | |
| 7. | Захист програмного продукту | 11.05.2020 | |
| 8. | Передзахист | 25.05.2020 | |
| 9. | Захист | 15.06.2021 | |

Студент

(підпис)

Трамбович А.Р.

(прізвище та ініціали)

Керівник роботи

(підпис)

Гусєва І.І.

(прізвище та ініціали)

АНОТАЦІЯ

У роботі розглянуті теоретичні та практичні аспекти розробки програмного продукту для формування навчальних планів для програм подвійного диплому.

Метою роботи є створення веб-додатку для роботи з файлами програми подвійного диплому та узгодження навчальних планів. Програма подвійного диплому – це програма академічної мобільності на певному рівні вищої освіти, за якою освітній процес реалізується одночасно двох закладах вищої освіти за погодженими освітніми програмами, яка дає змогу отримати два дипломи.

Для досягнення мети проаналізовано роботу аналогічних програм, розроблено класифікаційну модель, реалізовано алгоритми зчитування, порівняння та безпечного зберігання даних.

Пояснювальна записка складається зі вступу, п'яти розділів, висновку, списку використаних джерел.

Ключові слова: веб-додаток, парсер, порівняння рядків, алгоритм Олівера, XLS таблиця, формування навчальних планів для програм подвійного диплому.

ABSTRACT

The paper considers theoretical and practical aspects of software product development for the formation of curricula for double degree programs.

The purpose of the work is to create a web application for working with double degree program files and coordinating curricula. The double diploma program is a program of academic mobility at a certain level of higher education, according to which the educational process is implemented simultaneously by two higher education institutions.

To achieve this goal, the work of similar programs is analyzed, a classification model is developed, algorithms for reading, comparing and secure data storage are implemented.

The explanatory note consists of an introduction, five chapters, a conclusion, and a list of sources used.

Keywords: web application, parser, row comparison, Oliver's algorithm, XLS table, curriculum development for double degree programs.

ЗМІСТ

| | |
|---------------------------------------------------------------------------------------|----|
| <u>ВСТУП</u> | 7 |
| 1. <u>Постановка задачі формування навчальних планів для подвійного диплому</u> | 8 |
| 2. <u>Опис предметної області</u> | 10 |
| 2.1 <u>Способи роботи з Excel-файлами</u> | 10 |
| 2.2 <u>Способи безпечного зберігання файлів</u> | 12 |
| 2.3 <u>Методи порівняння даних</u> | 16 |
| 2.4 <u>Опис існуючих аналогів</u> | 17 |
| 2.5 <u>Висновки до розділу</u> | 26 |
| 3. <u>Засоби розробки</u> | 27 |
| 3.1 <u>Засоби Front-End розробки</u> | 27 |
| 3.2 <u>Засоби Back-End розробки</u> | 29 |
| 4. <u>Опис програмної реалізації</u> | 36 |
| 4.1 <u>Модель реалізації веб-додатку</u> | 36 |
| 4.2 <u>Структура програмного забезпечення</u> | 37 |
| 4.3 <u>Реалізація алгоритму отримання даних з файлу</u> | 39 |
| 4.4 <u>Реалізація алгоритму порівняння файлів</u> | 40 |
| 4.5 <u>Опис бази даних</u> | 42 |
| 4.5 <u>Діаграма прецедентів системи</u> | 43 |
| 4.6 <u>Діаграма розгортання</u> | 45 |
| 5. <u>Робота користувача з програмною системою</u> | 47 |
| 5.1 <u>Авторизація</u> | 47 |
| 5.2 <u>Модуль завантаження</u> | 48 |
| 5.3 <u>Модуль порівняння</u> | 50 |
| 5.4 <u>Модуль перегляду та видалення</u> | 53 |
| 5.5 <u>Закінчення роботи з програмою</u> | 53 |
| 5.6 <u>Висновки до розділу</u> | 54 |
| <u>Висновок</u> | 55 |
| <u>Список використаних джерел</u> | 56 |

ВСТУП

Методи автоматизації все більше впроваджуються у наше життя. Програми автоматизації призначені для виконання рутинної роботи якісно, швидко, та виключаючи людський фактор. Дуже часто у навчальних закладах викладачі зіштовхуються з великою кількістю документів, які необхідно перевірити, а іноді і порівняти між собою.

Для вирішення цієї проблеми існують синтаксичні аналізатори, або парсери – частини програми, які перетворюють текст з файлів у структурований формат, необхідний для подальшого аналізу отриманих даних. Парсери бувають різних типів та з різною логікою роботи, в залежності від даних, які вони мають зчитувати. Парсер можна інтегрувати в веб-інтерфейс – це дозволить користуватись ним з любого пристрою та з οποї точки планети, маючи лише доступ в інтернет.

Для реалізації такого програмного забезпечення будемо використовувати веб-сервер Apache 2.4, а програмувати його будемо за допомогою мови PHP 7. Мова програмування PHP має великий обсяг готових бібліотек, які допоможуть у роботі з файлами. Зберігати файли будемо за допомогою СУБД MySQL через її швидкість та надійність

Веб-інтерфейс для зручності користувача повинен бути адаптивним. Для швидкої реалізації цієї задачі ідеально підходить фреймворк Bootstrap.

1. ПОСТАНОВКА ЗАДАЧІ ФОРМУВАННЯ НАВЧАЛЬНИХ ПЛАНІВ ДЛЯ ПРОГРАМ ПОДВІЙНОГО ДИПЛОМУ

Метою дипломної роботи є створення системи для формування навчальних планів для програм подвійного диплому. Програма повинна виконувати функцію завантажування даних у базу даних, порівняння їх та видалення за бажанням користувача. Формат та структура вихідних файлів може відрізнятися.

Для реалізації поставленої мети було поставлено наступні задачі:

- проаналізувати можливі варіанти роботи з файлами
- розробити алгоритми роботи з файлами та їх даними
- розробити базу даних
- розробити веб-додаток

Вхідні дані – навчальні плани у вигляді електронної таблиці

Вихідні дані – зведений навчальний план.

Система повинна відповідати усім нормам безпеки особистих даних: усі завантажені файли повинні шифруватись за алгоритмом SHA-3. Доступ до системи має відбуватись тільки після введення логіну та паролю.

Програма повинна використовувати загальнодоступні значки та шаблони інтерфейсу користувача.

Додаток не має перевизначати або некоректно використовувати стандартні шаблони інтерфейсу користувача.

Система повинна виконувати функцію зчитування даних з XLS файлів, враховуючи те, що стовпці з даними можуть знаходитись у довільному порядку. Після отримання даних програмою, користувач повинен мати можливість переглядати, порівнювати та видаляти дані з бази.

Веб-додаток буде розроблятися за допомогою фреймворку Bootstrap 4, для швидшої та стабільнішої роботи клієнтської частини. Для обробки запитів між клієнтом та сервером буде використовуватись мова програмування PHP. Дані будуть зберігатись засобами системи управління базами даних MySQL. Локальний сервер створений засобами OpenServerPanel.

Функції, які має виконувати програма:

- зчитування даних з файлу
- безпечне збереження даних на сервері
- порівняння даних між собою
- формування звіту
- видалення файлів з серверу

Для зручності, програма буде поділена на три сторінки:

- модуль завантаження
- модуль порівняння
- модуль перегляду та видалення

2. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

2.1 Способи роботи з Excel-файлами

Мова програмування PHP широко використовується в усіх сферах програмування та має чимало готових бібліотек. Робота з файлами це основа для багатьох програмних продуктів, тому існує багато способів роботи з ними, кожен з яких має свої переваги та недоліки. Було розглянуто три найпопулярніші бібліотеки:

- PHP-Excel-Reader
- Nuovo Spreadsheet-Reader
- PHPSpreadsheet

PHP-Excel-Reader безпосередньо зчитує двійковий формат файлів XLS і може повертати значення і формати з будь-якого елемента. Його головна особливість – швидке створення об'єкту класу, просте та швидке зчитування з XLS файлу, вивід даних одразу на HTML сторінку із CSS стилями. Недоліками є те, що бібліотека не може працювати з XLSX файлами (Рис 2.1), має проблеми з деякими типами кодувань та не має методів запису даних у файл.

| Format | Reading | Writing |
|-----------------------------------|---------|---------|
| Open Document Format/OASIS (.ods) | ✓ | |
| BIFF 5 (.xls) Excel 95 | ✓ | |
| BIFF 8 (.xls) Excel 97 and above | ✓ | |

Рисунок 2.1 – Перелік доступних для роботи форматів у PHP-Excel-Reader

Nuovo Spreadsheet-Reader – бібліотека для зчитування таблиць, яка відрізняється від інших тим, що основною метою для неї є ефективне отримання даних з об'ємних файлів. Всі дані зчитуються з файлу послідовно, кожен рядок повертається у вигляді числового масиву. Бібліотека вміє працювати з CSV, XLS, XLSX файлами (Рис 2.2). Головною проблемою бібліотеки є те, що вона має проблеми з пам'яттю, оскільки відразу зчитує та зберігає дані, через що нерідко трапляються помилки.

| Format | Reading | Writing |
|----------------------------------------------|---------|---------|
| Open Document Format/OASIS (.ods) | ✓ | ✓ |
| Office Open XML (.xlsx) Excel 2007 and above | ✓ | ✓ |
| BIFF 8 (.xls) Excel 97 and above | ✓ | ✓ |
| BIFF 5 (.xls) Excel 95 | ✓ | |
| CSV | ✓ | ✓ |

Рисунок 2.2 – Перелік доступних для роботи форматів у Nuovo Spreadsheet-Reader

RHPSpreadsheet – багатофункціональна бібліотека, яка на відміну від інших вміє не тільки зчитувати дані в Excel, а й записувати їх. Розроблена як частина великої бібліотеки RHPOffice, за рахунок чого вміє працювати з багатьма типами файлів [8] (Рис. 2.3). Перевагою є те, що RHPSpreadsheet легка у використанні та має власну документацію. Єдиним недоліком є проблеми з пам'яттю, при роботі з великою кількістю даних.

| Format | Reading | Writing |
|-----------------------------------------------------------------------------------------------|---------|---------|
| Open Document Format/OASIS (.ods) | ✓ | ✓ |
| Office Open XML (.xlsx) Excel 2007 and above | ✓ | ✓ |
| BIFF 8 (.xls) Excel 97 and above | ✓ | ✓ |
| BIFF 5 (.xls) Excel 95 | ✓ | |
| SpreadsheetML (.xml) Excel 2003 | ✓ | |
| Gnumeric | ✓ | |
| HTML | ✓ | ✓ |
| SYLK | ✓ | |
| CSV | ✓ | ✓ |
| PDF (using either the TCPDF, Dompdf or mPDF libraries, which need to be installed separately) | | ✓ |

Рисунок 2.3 – Перелік доступних для роботи форматів у PHPSpreadsheet

2.2 Способи безпечного зберігання файлів

Головною проблемою програмного продукту, який працює з файлами є гарантія безпеки завантажених даних. Особливо це стосується веб-додатків, які знаходяться на загальнодоступному хостингу. Якщо не захистити файли належним чином є шанс, що вони будуть завантажені сторонніми обличчями. Перш за все, доступ до завантажених файлів та директорій повинен бути тільки у авторизованих користувачів. Авжеж, це захист базового рівня, тому необхідно реалізувати складніший метод. Для вирішення цієї проблеми було розглянуто декілька варіантів:

- Створення ключу-доступу для кожного файлу
- Хешування назви файлу
- Завантаження файлів у хмару

Метод створення ключу доступу для кожного файлу є надійним та нескладним у реалізації. Суть метода полягає в тому, що при завантаженні файлу на сервер генерується довільний ключ, який закріплюється у базі даних за цим файлом. У подальшому, за допомогою цього ключа можна буде завантажити цей файл. Недоліком цього методу є те, що він тягне за собою необхідність покращити безпеку бази даних, тому що у випадку взлому БД, усі файли можуть бути втрачені. Також, існує імовірність втрати цього ключа користувачем. Роботу цього методу добре реалізує програмне забезпечення BitLocker. Програма створює ключ та шифрує усі данні, які повинні бути під захистом.

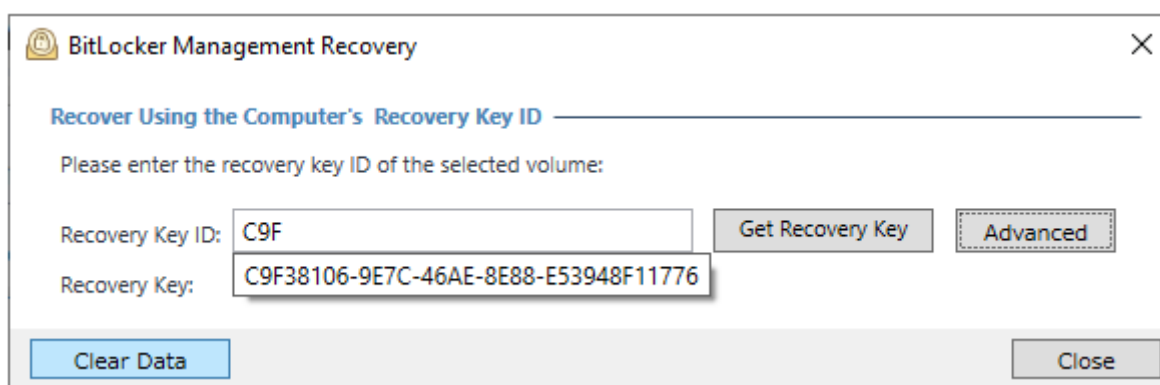


Рисунок 2.4 – Інтерфейс BitLocker

Метод хешування назви файлу полягає в тому, що одразу після завантаження назва файлу змінюється хеш-функцією (Рис 2.5) і зберігається у такому вигляді на сервері. Якщо підібрати правильну хеш-функцію (Рис 2.6), то дізнатись назву файлу буде майже неможливо, через що і доступитись до нього буде майже нереально.

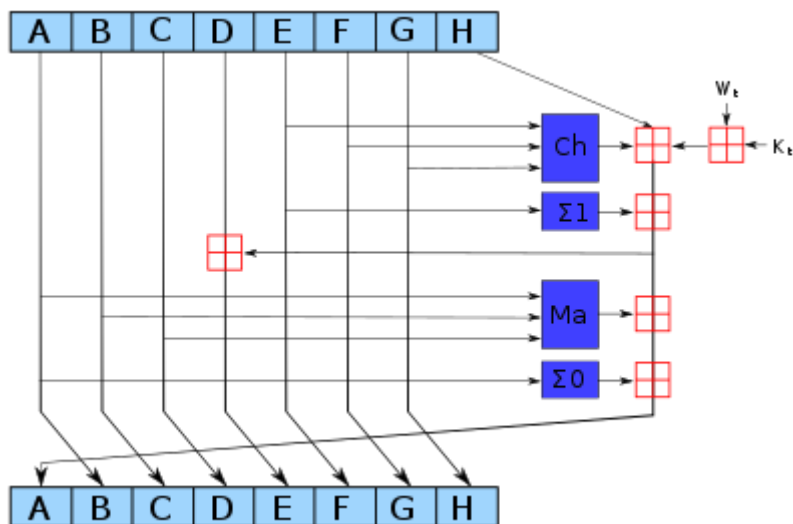


Рисунок 2.5 – Графічне зображення однієї ітерації хешування

Хешування - це алгоритм, який обчислює значення бітового рядка фіксованого розміру. Файл в основному містить блоки даних. Хешування перетворює дані в набагато більш коротке значення фіксованої довжини або ключ, який представляє вихідний рядок. Хороший алгоритм хешування демонструватиме властивість, назване ефектом лавини, при якому результат хешування буде значно або повністю змінюватися навіть при зміні одного біта або байта даних у файлі [14].

| Algorithm | Word Size | Block Size | Output Size | Rounds |
|-----------|-----------|----------------------|---------------------|--------|
| MD-2 | 32 | 128 | 128 | 18 |
| MD-4 | 32 | 512 | 128 | 48 |
| MD-5 | 32 | 512 | 128 | 64 |
| SHA-0 | 32 | 512 | 160 | 80 |
| SHA-1 | 40 | 512 | 160 | 80 |
| SHA-2 | 56/64 | 512/1024 | 224/256 /384/512 | 64/80 |
| SHA-3 | 64 | 1152/1088 832/576 | 224/256/384/512 | 24 |

Рисунок 2.6 – Порівняльна таблиця хеш-алгоритмів

2 жовтня 2012 SHA-3 (Кессак) став переможцем конкурсу криптографічних алгоритмів, проведеного Національним інститутом стандартів і технологій США. 5 серпня 2015 року алгоритм був затверджений та опублікований як стандарт FIPS 202. SHA-3 забезпечує надійну односторонню функцію. Це означає, що не можливо відтворити вхідні дані з виводу хешу, а також не можливо змінити вхідні дані, не змінивши хеш. SHA-3 робить неможливим створення будь-яких двох наборів даних з однаковим хешем.

Сучасним методом збереження файлів є завантаження їх на VPS – віртуальний виділений сервер. Як правило, такі сервери фізично знаходяться у європейських дата-центрах, де вони знаходяться під захистом мережесих екранів та моніторингових систем, а доступ до файлів та директорій відбувається через VPN шлюз. Для обміну даними між клієнтом та сервером можна встановити FTP сервер - інтернет-протокол,

який використовується для передачі файлів через Інтернет з одного комп'ютера на інший (Рис. 2.7). Цей сучасний метод має лише один недолік – висока ціна.

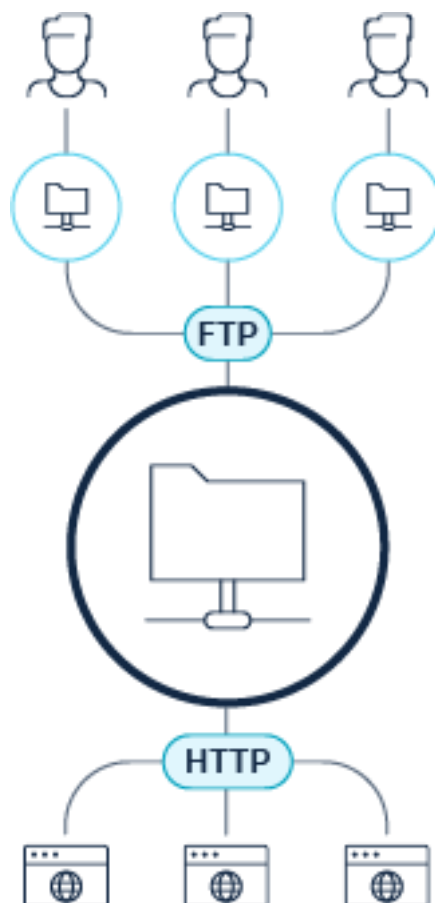


Рисунок 2.7 – Принцип роботи FTP-серверу

2.3 Методи порівняння даних

Метод levenshtein() – це мінімальна кількість вставок, замін і вилучень символів, необхідне для перетворення рядка №1 у рядок №2. Складність алгоритму дорівнює $O(m * n)$, де n і m - довжини рядків №1 та №2. Для розрахунку будується матриця розміром $(n+1)*(m+1)$, де n і m - довжини рядків №1 та №2 [10]. Для побудови матриці використовується рекурсивне рівняння виду:

$$D_{0,0} = 0$$

$$D_{i,j} = \min \begin{cases} D_{i-1,j-1} & +0 \text{ (equal, nochange)} \\ D_{i-1,j-1} & +1 \text{ (replace)} \\ D_{i-1,j} & +1 \text{ (insert)} \\ D_{i,j-1} & +1 \text{ (delete)} \end{cases}$$

Метод `similar_text()` обчислює ступінь схожості двох рядків по алгоритму Олівера, але з деякими правками. Ця реалізація алгоритму не використовує стека, використаного в оригіналі, натомість застосовуються рекурсивні виклики, що в деяких випадках може прискорити процес. Для знаходження застосовуються рекурсивні виклики, що в деяких випадках може прискорити процес. Складність алгоритму становить $O(n^3)$, де n - довжина довшою з двох рядків [9].

$$d = \frac{2 * K_m}{|S1| + |S2|}$$

де K_m - число співпадаючих символів, $|S1|$, $|S2|$ - довжина рядків

2.4 Опис існуючих аналогів

CMS Wordpress – це безкоштовна система управління вмістом сайту з відкритим вихідним кодом, написана на PHP. Головною перевагою Wordpress – є велика кількість безкоштовних плагінів для будь-якого типу сайту. Окрім невеликих плагінів та модулів, які додають необхідний функціонал на сайт, існують і ті, що можуть глобально змінити концепцію сайту [11]. Одним з таких є WooCommerce – він перетворює сайт на інтернет-магазин і додає в адміністративну частину сайту чимало нових можливостей. Головна можливість – управління товарами інтернет-магазину (Рисунок 2.8– Меню товарів WooCommerce).

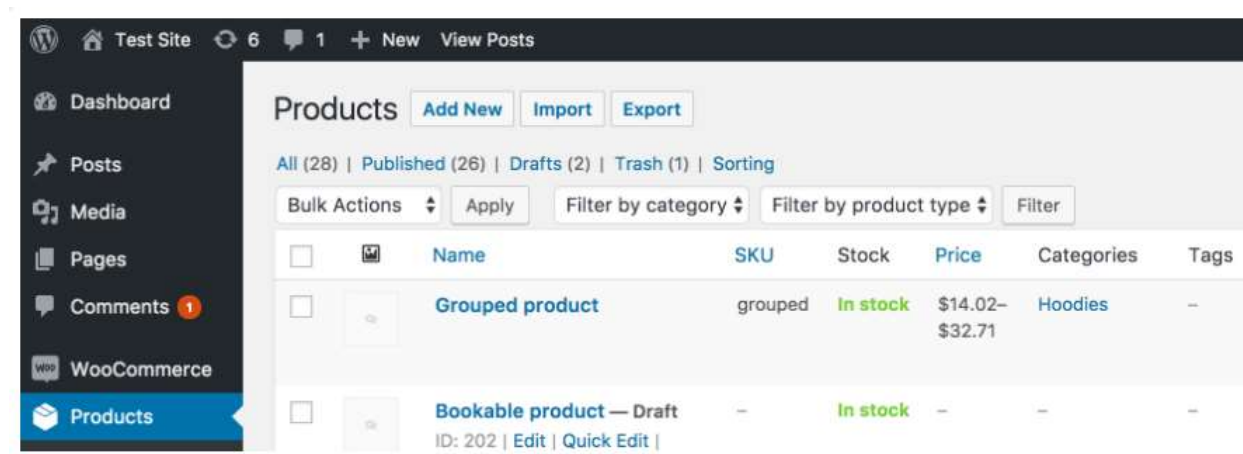


Рисунок 2.8 – Меню товарів WooCommerce

Плагін WooCommerce допомагає створити магазин електронної комерції за допомогою WordPress. Він бере базову операційну систему WordPress і перетворює її в повністю функціонуючий магазин електронної комерції.

WooCommerce дозволяє легко і доступно продавати будь-які товари і послуги на веб-сайті. Він дозволяє продавати цифрові і фізичні продукти, приймати безпечні платежі, управляти запасами і автоматично сортувати всі податки [12].

Модуль імпорту товарів для WooCommerce може зчитувати CSV та XLS файли та автоматично завантажувати на сайт товари, їх опис, зображення, ціну тощо.

Великий плюс полягає в тому, що адміністратор сайту не зобов'язан притримуватись шаблону, достатньо вказати назву кожної з колонок файлу та потім, під час імпорту, вказати за що відповідає кожне зі значень. Якщо якість зі значень імпортувати не потрібно – для цього є спеціальне значення у випадяючому меню (Рис 2.9 – Модуль імпорту товарів WooCommerce).

| Column name | Map to field |
|----------------------------------------------------|--------------|
| ID <i>Sample: 80</i> | ID |
| Type <i>Sample: variation</i> | Type |
| SKU <i>Sample: T-SHIRT-SHIP-YOUR-IDEA-BLACK</i> | SKU |
| Name <i>Sample: Ship Your Idea</i> | Name |
| Published <i>Sample: 1</i> | Published |

Рисунок 2.9 – Модуль імпорту товарів WooCommerce

Після завершення роботи з назвами таблиць файлу адміністратор натискає кнопку «Імпортувати» і модуль самостійно створює товари з усіма необхідними даними, які були вказані (Рис 2.10 – Повідомлення про успішний імпорт товарів).

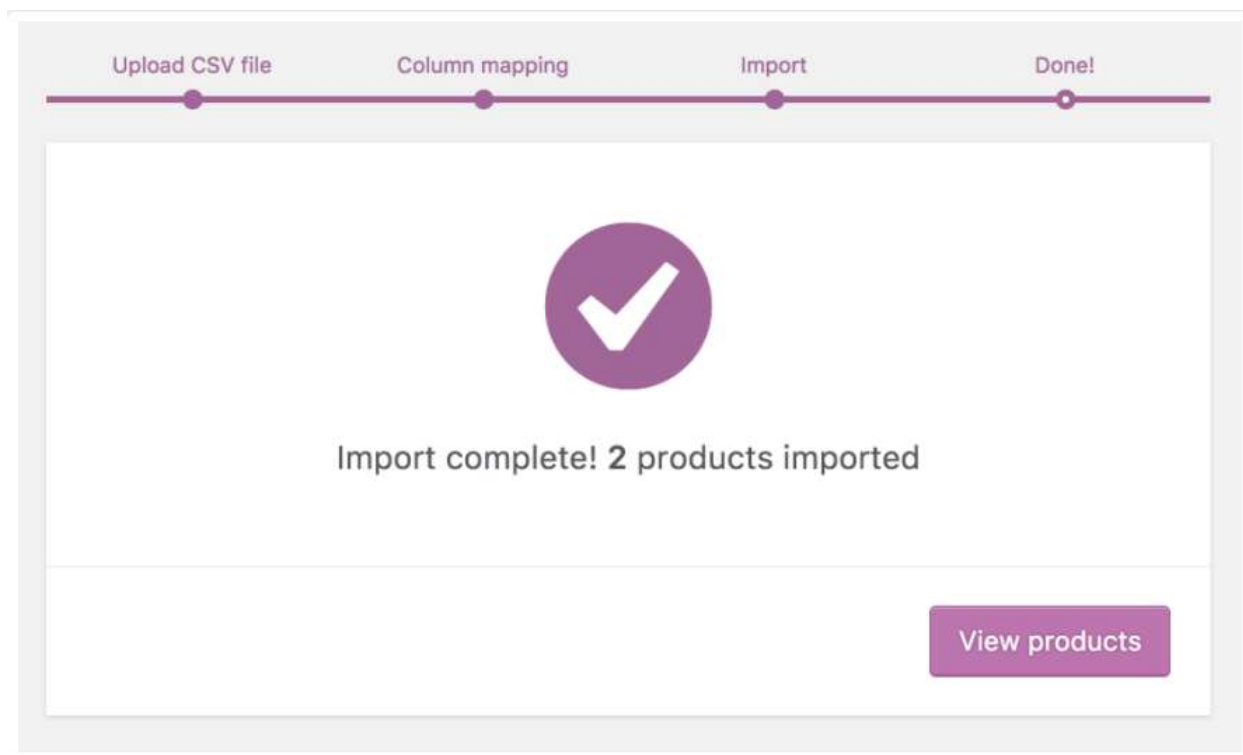


Рисунок 2.10 – Повідомлення про успішний імпорт товарів

Логіка цього модуля націлена на максимальну зручність та швидкість зі сторони користувача, але сам процес відбувається досить довго при великих обсягах даних та не включає людського фактору.

Google Merchant Center - це єдина онлайн-панель управління, де онлайн-компанії управляють своїм зовнішнім виглядом у всіх продуктах електронної комерції Google і при необхідності вносять зміни в свої онлайн-списки.

Основна мета Google Merchant Center - дозволити компаніям підтримувати актуальну інформацію про продукти, включаючи зображення і ціни, для відображення у відповідних результатах пошуку в Google покупки. Google Merchant Center також інтегрується з іншими сервісами Google, такими як Google My Business, щоб забезпечити надійний нагляд і контроль над маркетингом і електронною торгівлею на базі Google [13].

Перед тим як запусити рекламну компанію необхідно спочатку завантажити усі потрібні товари на перевірку, щоб компанія Google упевнилась в тому, що ви не

порушуєте правила розміщення. Для цього був створений окремий модуль, який обробляє завантажені дані (Рис 2.11 – Інтерфейс Google Merchant Center).

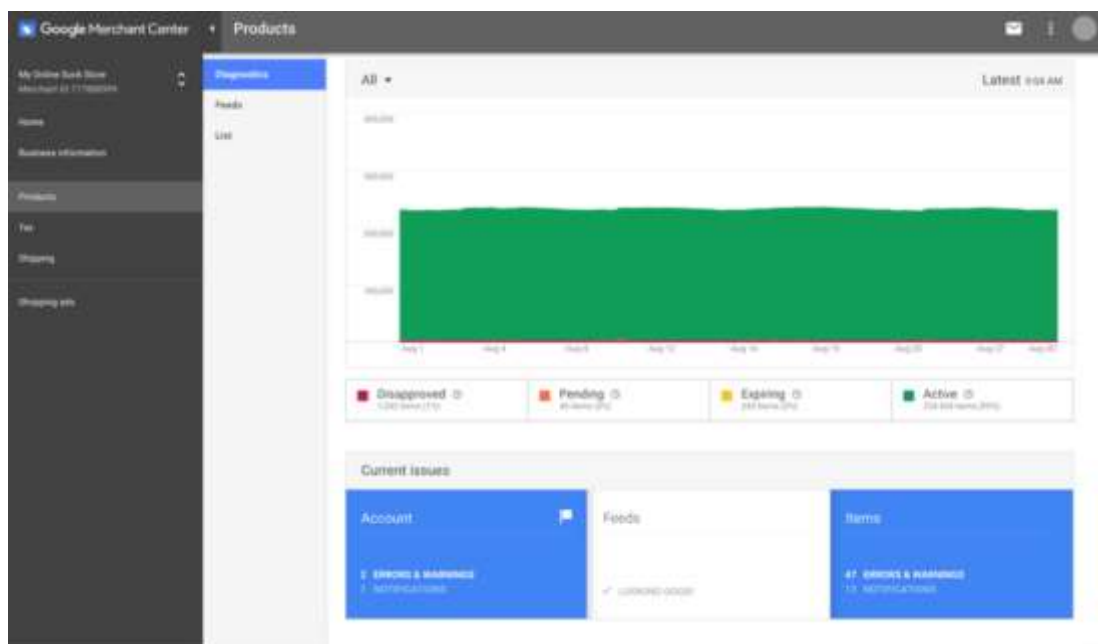


Рисунок 2.11 – Інтерфейс Google Merchant Center

У модуль необхідно завантажити CSV або XLS файл з товарами, їх описом, посиланнями на фото та ціною. Особливістю цього модуля є те, що ці дані перевіряють боти, тому Google зобов'язує притримуватись певного шаблону, умови до якого описані в особистому кабінеті (Рис 2.12 – Умови до шаблону Merchant Center). Якщо у файлі будуть стовпці з незрозумілими для модуля назвами він їх просто пропустить.

| Attribute | Description |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| id [id] | unique product identifier (no more than 50 characters) |
| title [title] | product name (no more than 150 characters) |
| description [description] | product description (no more than 5000 characters) |
| link [link] | link to the landing page of the product (no more than 2000 characters) |
| image_link [image_link] | link to the main product image (no more than 2000 characters) |
| availability [availability] | availability of goods in the store Valid values <ul style="list-style-type: none"> • in_stock [in stock] • out_of_stock [out_of_stock] • preorder [preorder] |
| price [price] | product price (Syntax - number, design in accordance with ISO 4217) |
| condition [state] | product condition |

Рисунок 2.12 – Умови до шаблону Merchant Center

Як не дивно, у такого модуля також існують свої переваги. Не дивлячись на те, що на підготовку файлу втрачається чимало часу, даний метод працює майже без помилок та дозволяє швидше обробляти дані, адже алгоритм спеціально оптимізований для певного шаблону і, що найголовніше, обробляється повністю автоматично. Система самостійно знаходить будь-які порушення у правилах розміщення реклами та повідомляє автора файлу про це.

В разі успішного або неуспішного завантаження даних система повідомить користувача про це у графічному вигляді (Рис 2.13).

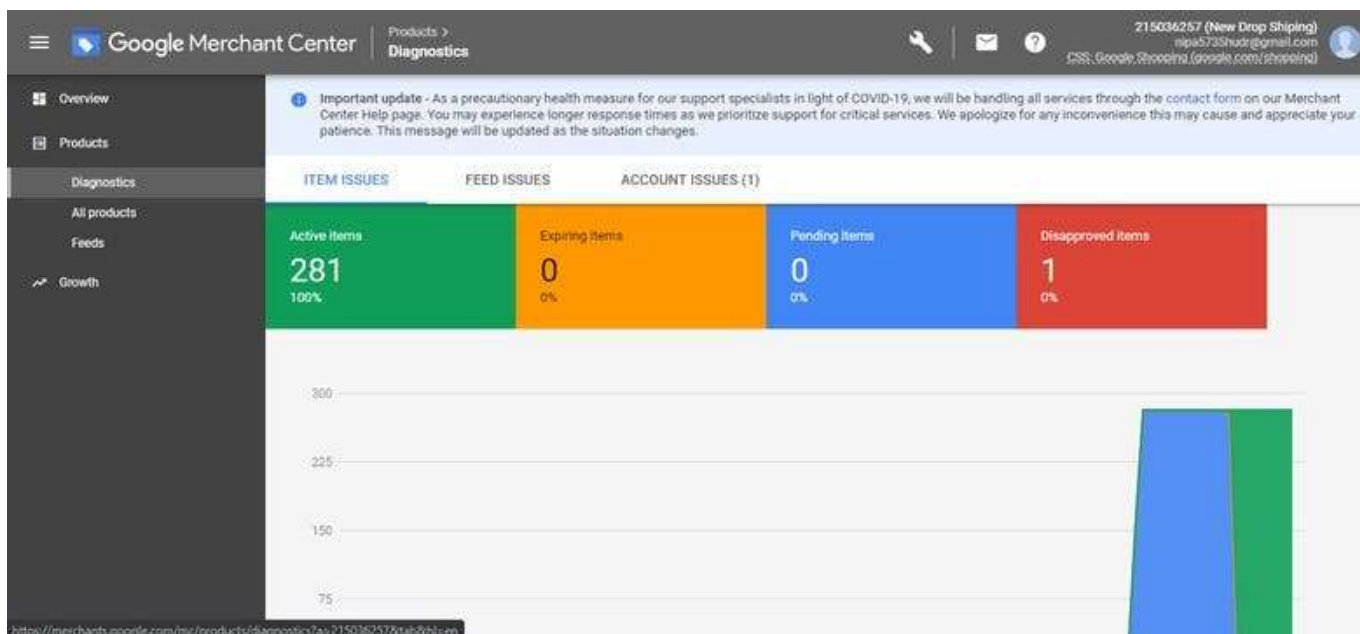


Рисунок 2.13 – Графічне відображення стану завантажених даних

Данна логіка роботи модуля налаштована на максимальну швидкість, автоматизованість зі сторони сервісу та точність у пошуку помилок.

OCR (optical character recognition) – це технологія зчитування зображень друкованого або рукописного тексту та конвертація їх у текстові дані, які використовуються для представлення символів в комп'ютері [15].

Достатньо складний у реалізації, але дуже ефективний спосіб отримання даних з файлів будь-якого типу, адже зчитування відбувається напряму з фотографії, скріншота або PDF-файлу.

Для аналізу роботи даного методу отримання даних був обраний програмний продукт OCR Space, який працює не тільки онлайн, а й ще може бути використаний у власному проекті за допомогою підключення API (Рис 2.14 – Веб-сторінка OCR Space).



Free Online OCR - Convert images and PDF to text (Powered by the [OCR API](#))

Рисунок 2.14 – Веб-сторінка OCR Space

Програма не ставить ніяких умов перед користувачем, все що необхідно – це завантажити файл у потрібному форматі, обрати мову та певні додаткові можливості, якщо вони потрібні (Рис 2.15 – Інтерфейс OCR Space). Після заповнення усіх необхідних полів, програма розіб’є файл на символи та зчитає їх, записавши одразу у текстовому та JSON форматах (Рис 2.16 – Результат роботи програми).

Upload image or PDF file (.png, .jpg, .webp or .PDF)

OR

Paste url to source file (.png, .jpg, .webp or .PDF)

Language

Detect orientation and auto-rotate image if needed

Do receipt scanning and/or table recognition:

Auto-enlarge content (recommended for low DPI)

Create Searchable PDF Just extract text and show overlay (fastest option)
 Create searchable PDF with visible text layer
 Create searchable PDF with invisible text layer

Select OCR Engine to use: Use OCR Engine1 (Default, fast, most OCR languages)
 Use OCR Engine2 (Often better for number and special character OCR)

Рисунок 2.15 – Інтерфейс OCR Space

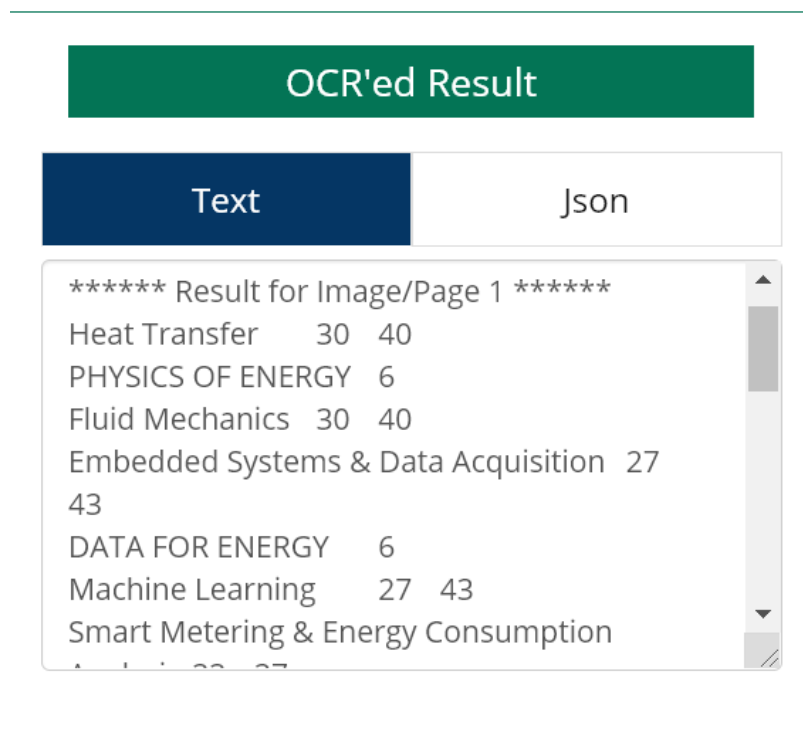


Рисунок 2.16 – Результат роботи програми

У порівнянні з іншими методами, дана технологія дозволяє отримувати дані не прив'язуючись до типу та формату файлу, не потребує вносити якісь зміни зі сторони користувача, але й має свої недоліки. Перш за все, це досить вибагливе до заліза програмне забезпечення, тому для швидкого зчитування великої кількості даних потрібен потужний комп'ютер. По-друге, цей метод потребує дуже ретельного налаштування та тестування, адже не завжди текст отримується у тому ж вигляді, що й був спочатку. Особливо це відноситься до таблиць та даних, де існує одночасно кирилиця та латиниця.

2.5 Висновки до розділу

У даному розділі наведено приклади модулів та парсерів різноманітних типів та алгоритмів, які використовуються у популярних веб-ресурсах. В кожному були знайдені свої переваги та недоліки.

Проаналізовано бібліотеки для роботи з Excel файлами у мові програмування PHP. Розглянуто можливі методи збереження файлів на сервері. Описано вбудовані в PHP методи порівняння даних.

3. ЗАСОБИ РОЗРОБКИ

3.1 Засоби Front-End розробки

Мова розмітки HTML була розроблена британським вченим Тімом Бернерсом-Лі у 1986-1991 роках, а вже у 1995 році був створений перший загальноприйнятий HTML стандарт, відомий як RFC 1866. Головна особливість цієї мови – використання дескрипторів, за допомогою яких можна швидко поділити сторінку на блоки. Ця мова розуміється браузерами, і його елементи служать в якості інструкцій, тому веб-сторінки відображаються однаково в усіх сучасних браузерах [6].

HTML5 - це поточна версія HTML, метою якої є забезпечення більш ефективної взаємодії з користувачем. HTML5 так само є кодом розмітки, як і його попередники. Як і будь-яка нова версія веб-елемента, HTML5 вирішує багато проблем попередніх версій. HTML5 дозволяє веб-сторінкам усувати необхідність в файлах cookie, зберігаючи дані в браузері користувача. Збільшує швидкість завантаження веб-сторінки, підвищуючи адаптивність між браузерами. Усунення необхідності в плагінах підвищило адаптивність браузерів для відео та інших медіа. Ще одна ключова особливість HTML5 - додавання елементів управління відтворення відео та аудіо безпосередньо в браузері.

Каскадні таблиці стилей CSS – це формальна мова опису зовнішнього виду веб-сторінки. Використовується у веб-програмуванні для визначення кольорів, стилей, шрифтів, позиціонування блоків на сторінці тощо. CSS це основний інструмент для створення зовнішнього виду сторінки. Термін «каскадні таблиці стилей» був вперше запропонований Хоконом Лі у 1994 році, який розпочав роботу над CSS разом з Бертом Босом. Вже в кінці 1996 року була видана перша версія CSS1, яку затвердив Консорціум Всесвітньої павутини .

Фреймворк Bootstrap 4 - це frontend фреймворк, створений для швидкого створення адаптивних сайтів. Це найпопулярніших набір інструментів, за допомогою яких було швидко створено клієнтську частину модулю веб-додатку.

Bootstrap - це безкоштовний додаток з відкритим вихідним кодом, заснований на CSS. Його можна використовувати для розробки зовнішнього інтерфейсу мобільних додатків, використовуючи різноманітні шаблони, які в основному створюються на JavaScript та можуть бути змінені [16].

Мова програмування JavaScript – це мова сценаріїв на стороні клієнта, яка дозволяє веб-сторінкам бути динамічними. Це інтерпретована мова і її не потрібно компілювати. JavaScript використовується в основному для поліпшення взаємодії користувача з веб-сторінкою.

JavaScript з'явився на базі мови Cmm, яка була розроблена у 1992 році і повинна була замінити макроси та залишатись схожою на мову Cі. На основі Cmm був створений програмний продукт CEnvі. У 1995 році компанія Nombas розробила версію CEnvі, яка працювала на веб-сторінках. Саме цю подію можна назвати початком історії JavaScript.

У фоновому режимі JavaScript зчитує літери під час введення користувачем, відправляє ці дані на віддалений сервер, а сервер відправляє їх назад. Програмне забезпечення на стороні сервера аналізує слова і запускає алгоритми, щоб передбачити запит користувача. JavaScript на клієнтській машині максимально простий і малий, щоб не сповільнювати взаємодія користувача. Зв'язок між JavaScript і серверної програмою обмежений пропускнуою здатністю користувача [17].

Середовище розробки Brackets – це сучасний текстовий редактор з відкритим вихідним кодом, створений спеціально для веб-розробників. Його основні переваги – безкоштовність та можливість завантажувати сторонні модулі, які покращують роботу з кодом.

Brackets написаний на JavaScript та був випущений у 2014 році компанією Adobe. Нажаль, нещодавно компанія анонсувала закриття проекту у вересні 2021 року [7].

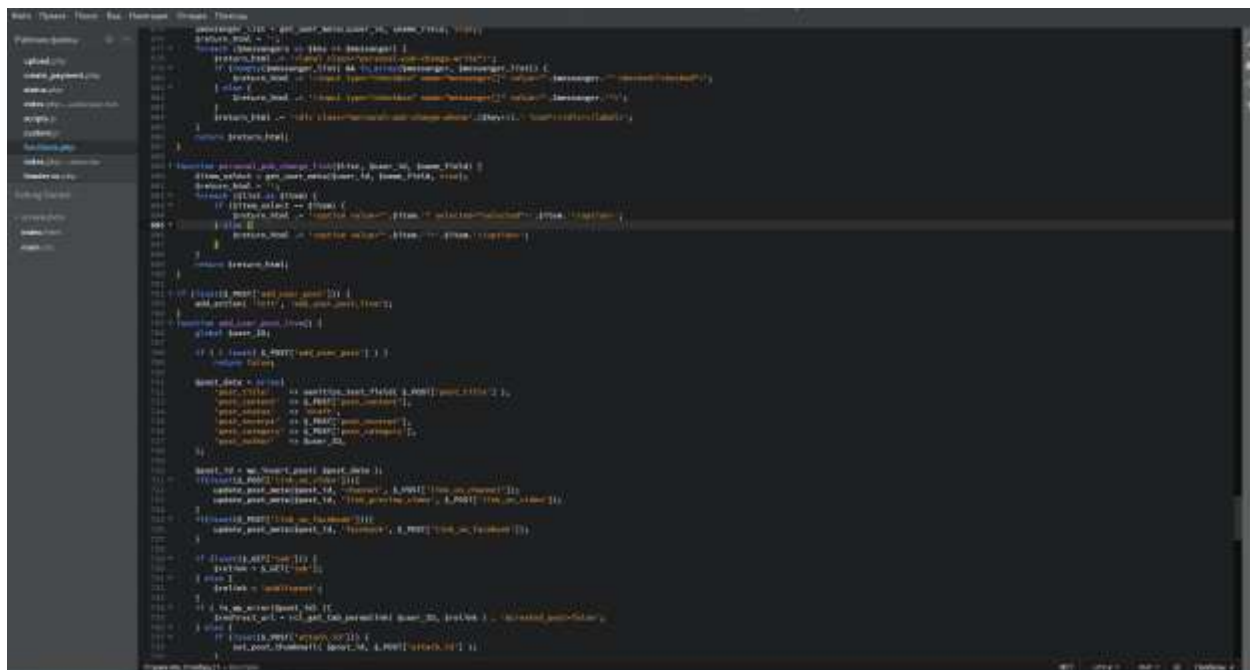


Рисунок 3.1 – Інтерфейс Brackets

3.2 Засоби Back-End розробки

Open Server Panel – це портативний локальний WAMP/WNMP сервер, який підтримує велику кількість інструментів та компонентів для веб-розробки.

WAMP - це аббревіатура, що позначає Windows, Apache, MySQL та PHP. Це програмний стек, який означає, що установка WAMP встановлює Apache, MySQL та PHP безпосередньо в операційну систему Windows. Також існує LAMP і головною різницею від WAMP є те, що такий стек встановлюється на Linux. WNMP, у свою чергу, відрізняється тим, що замість Apache серверу у програмному стеку розгорнуто Nginx [18].

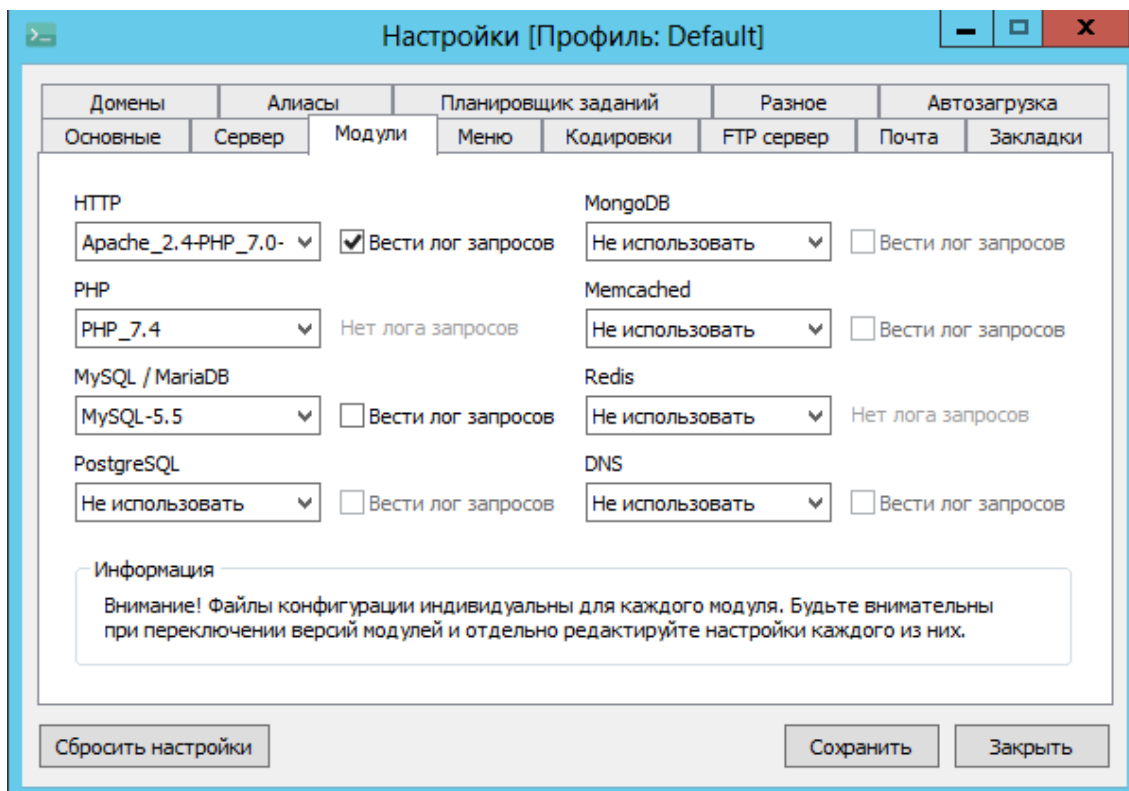


Рисунок 3.2 – Налаштування OpenServerPanel

Open Server має дуже зручний інтерфейс налаштувань, дає можливість обирати версії HTTP, PHP, СУБД, підтримує FTP та SSL, зручне створення та налаштування локальних доменів.

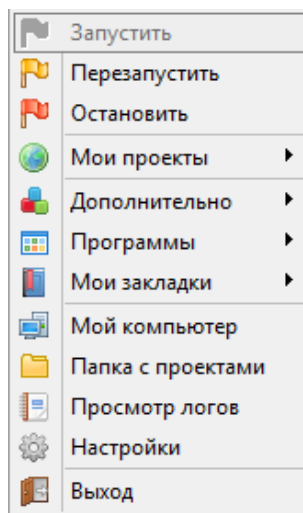


Рисунок 3.3 – Панель швидкого доступу OpenServerPanel

Запуск проекту відбувся першого серпня 2010 року та являється професійним інструментом, який постійно оновлюється.

Для розробки модулю був використаний локальний веб-сервер Open Server Panel, який працює на базі Apache 2.4. Основною мовою програмування для серверної частини була обрана PHP, а для адміністрування СУБД MySQL використовувався веб-додаток PhpMyAdmin.

Apache HTTP Server – це програмне забезпечення з відкритим вихідним кодом. Головні переваги: швидкість, надійність, гнучкість у налаштуванні. Apache може бути налаштований під будь-які потреби програміста завдяки підтримці різноманітних розширень та модулів.

Сервер був написаний на початку 1995 року розробниками з компанії Apache Software Foundation. З квітня 1996 року і до сьогодні Apache – це найпопулярніший HTTP сервер у світі [5].

Nginx - це програмне забезпечення з відкритим вихідним кодом для веб-обслуговування, зворотного проксінг, кешування, балансування навантаження, потокової передачі мультимедіа і багато чого іншого. Він починався як веб-сервер, розрахований на максимальну продуктивність і стабільність.

Ігор Сисоєв почав розробку в 2002 році. Восени 2004 року вийшов перший публічно доступна реліз. З липня 2011 робота над nginx триває в рамках компанії Nginx.

| Apache | Nginx |
|---------------------------------------------------------------------|--------------------------------------------------------------------|
| Виконує функції веб-серверу | Виконує функції як веб-серверу, так і проксі-серверу |
| Не може обробляти багато запитів одночасно через великий веб-трафік | Може обробляти велику кількість запитів, але з обмеженням ресурсів |
| Підтримує багатопоточність | Керує подіями без багатопоточності |
| Модулі можуть завантажуватись та видаляться динамічно | Модулі завантажуються не динамічно, а через ядро ПЗ |
| Підтримує динамічний контент | Не підтримує динамічний контент |

Рисунок 3.4 – Таблиця порівняння Apache та Nginx

Мова програмування PHP 7 – це мова програмування сценаріїв загального призначення, яка в першу чергу використовується для веб-розробки. Швидка та гнучка PHP використовується на багатьох сучасних сайтах та підтримується усіма хостингами. Головна перевага PHP – вона створена для роботи з HTML та її код може бути написаний безпосередньо в HTML.

Крім управління вмістом сторінок, PHP також може відправляти заголовки HTTP. Вона дозволяє встановлювати файли cookie, управляти аутентифікацією і перенаправляти користувачів. PHP пропонує відмінне підключення до багатьох баз даних, а також інтеграцію з різними зовнішніми бібліотеками, які дозволяють робити все, від створення документів PDF до аналізу XML.

Завдяки тому, що мова PHP була створена більше 20 років тому, величезною перевагою є детальна документація, що допомагає реалізувати будь-яку поставлену задачу.

Історія PHP розпочалась у 1994 році, коли датський програміст Расмус Лердорф розробив на мові Сі набір CGI-інструментів для свого онлайн-резюме. У 1995 році вийшла перша версія PHP Tools, написана на Сі. На сьогоднішній день актуальною є восьма версія PHP, яка була випущена літом 2020 року.

Система управління базами даних MySQL – це найпопулярніша система управління реляційними базами даних з відкритим вихідним кодом. Головними перевагами MySQL є швидкість та надійність. СУБД MySQL є ідеальним вибором у роботі з PHP, тому що забезпечує дуже швидке з'єднання між сервером та базою даних, що дає змогу швидко записувати та передавати данні між клієнтом та сервером [2].

Для доступу до бази даних MySQL існує кілька API-інтерфейсів PHP – це розширення MySQLi та PDO_MySQL (Рис 3.9.1).

Перевага MySQLi - це використання підготовлених операторів. Підготовлений оператор - це шаблон оператора для системи баз даних. На відміну від звичайних операторів, вони містять наповнювачі замість значень параметрів. Якщо оператор повинен виконуватися кілька разів (наприклад, всередині циклу) в системі бази даних з різними параметрами, використання підготовлених операторів може збільшити швидкість, оскільки оператор уже попередньо переведений в системі бази даних і його потрібно виконувати тільки з новими параметрами.

PDO надає різні способи роботи з об'єктами і витягує підготовлені оператори, які значно спрощують роботу. Це інструмент доступу до бази даних на PHP, за допомогою якого ми забезпечуємо однаковий доступ до кількох баз даних. Цей метод дозволяє порівняно плавно перемикатися між різними базами даних і платформами, що можна легко зробити, змінивши рядок підключення. Він не підтримує синтаксис, специфічний для бази даних.

| Особливості | PDO | MySQLi |
|-----------------------|----------------------|----------------------|
| Підтримка БД | 12 різних драйверів | MySQL |
| Підключення | Легке | Легке |
| API | ООП | ООП + Процедури |
| Продуктивність | Швидко | Швидко |
| Названий параметр | Підтримує | Не підтримує |
| Відображення об'єктів | Підтримує | Підтримує |
| Збережена процедура | Підтримує | Підтримує |
| Підготовлені запити | Підтримує | Не підтримує |
| Безпека | Безпечніше за MySQLi | Менш безпечно за PDO |

Рисунок 3.5 – Порівняльна таблиця MySQLi та PDO

Веб-застосунок PhpMyAdmin – це веб-застосунок, написаний на PHP із графічним веб-інтерфейсом, який використовується для адміністрування MySQL та MariaDB.

Додаток дозволяє маніпулювати базою даних безпосередньо через браузер: виконувати запити, переглядати та редагувати вміст бази.

PhpMyAdmin був розроблений у 1998 році німецьким розробником на базі застосунку MySQL-WebAdmin. Проект локалізовано на 85 мов та використовується багатьма провайдерами світу.

4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

4.1 Модель реалізації веб додатку

Реалізація та тестування модулю проводилось у портативному програмному середовищі Open Server Panel. Для написання програмного коду було використано середовище Brackets.

Клієнтська частина створена на базі загального шаблону веб-додатка, який реалізований на базовій frontend-технології HTML5, CSS3, JS.

За серверну частину відповідає мова програмування PHP та бібліотека PhpSpreadsheet. Усі данні зберігаються за допомогою системи управління базами даних MySQL.

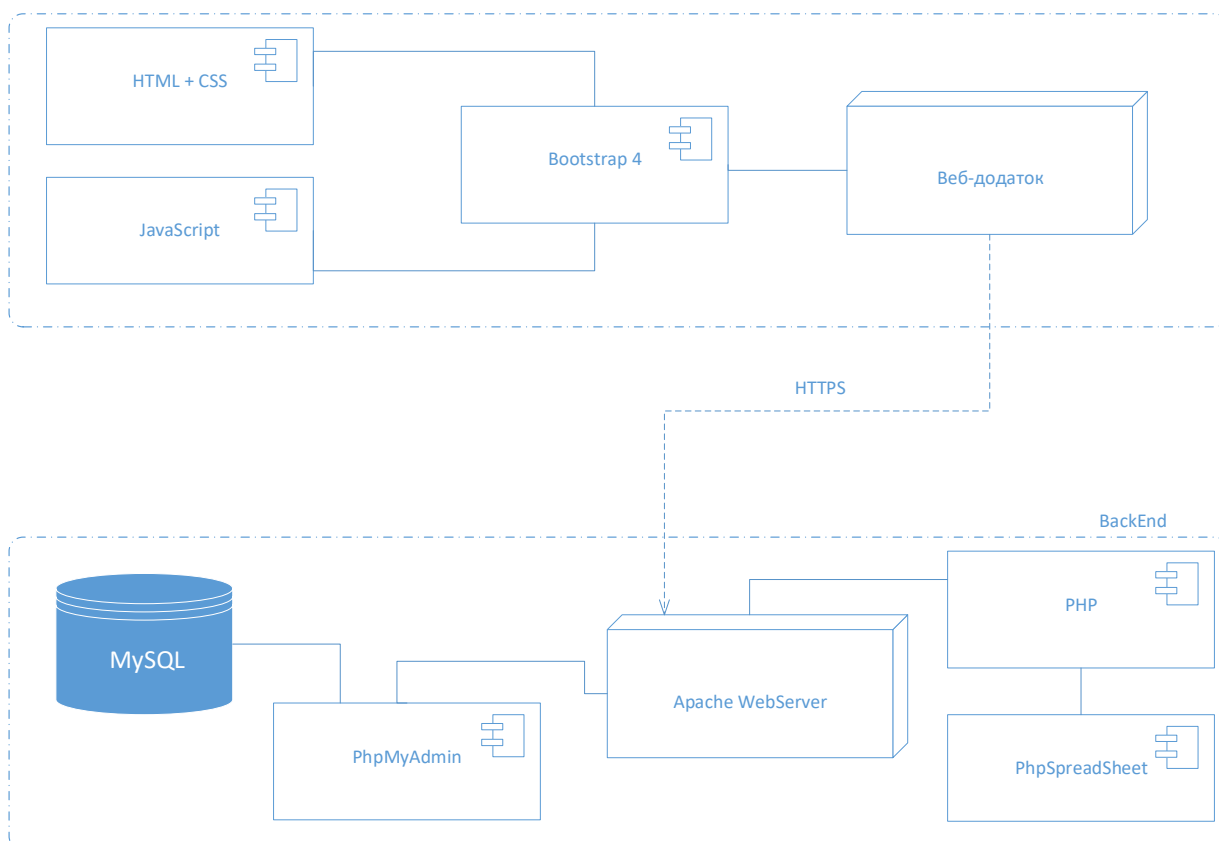


Рисунок 4.1 – Модель реалізації веб-додатку

4.2 Структура програмного забезпечення

Основою системи є комплекс WAMP – Windows, Apache, MySQL, PHP, який реалізований завдяки програмному забезпеченню OpenServerPanel. Комплекс повністю реалізує логіку програми: завантаження файлів, обробку запитів у базу даних, роботу з формами, вивантаження даних у файл.

На рисунку 4.1 зображено файлову структуру веб-додатку.

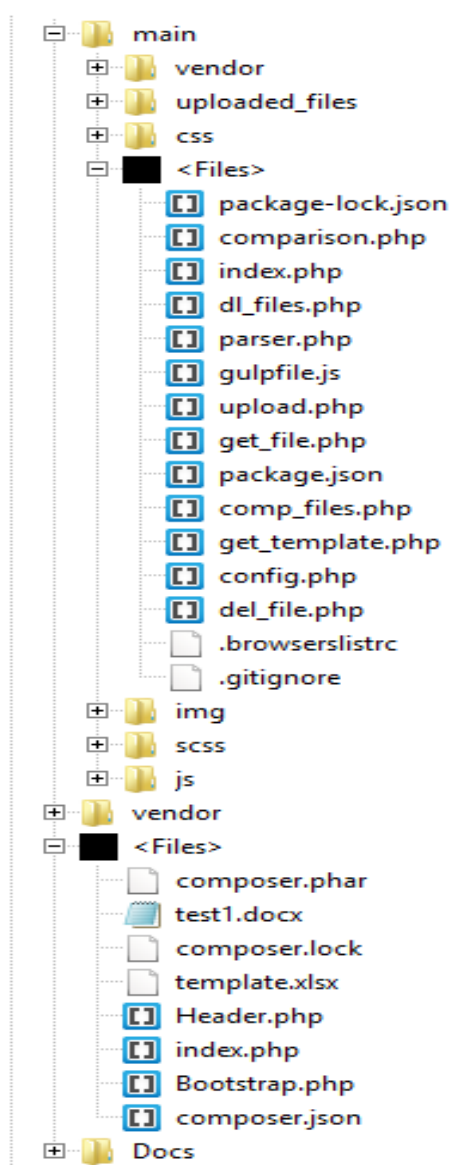


Рисунок 4.1 – файлова структура веб-додатку

- vendor – папка зовнішніх бібліотек
- uploaded_files – папка із завантаженими файлами
- css – папка стилів
- main – основна папка
- img – папка з іконками
- scss – папка стилів формату sass
- js – папка зі скриптами

Програмний продукт складається з веб-додатку та бази даних і побудований на клієнт - серверній архітектурі. Програма передбачає лише один рівень доступу, але дані, які звантажує користувач, доступні тільки йому. Стандартний рівень доступу передбачає:

- Можливість завантажити файл
- Можливість порівняти між собою файли, завантажені самим користувачем
- Можливість переглядати та видаляти дані з бази даних, завантажені самим користувачем
- Можливість сформуванати звіт за підсумками зрівняння

Логіку роботи веб-застосунку зображено на Рис 4.2.

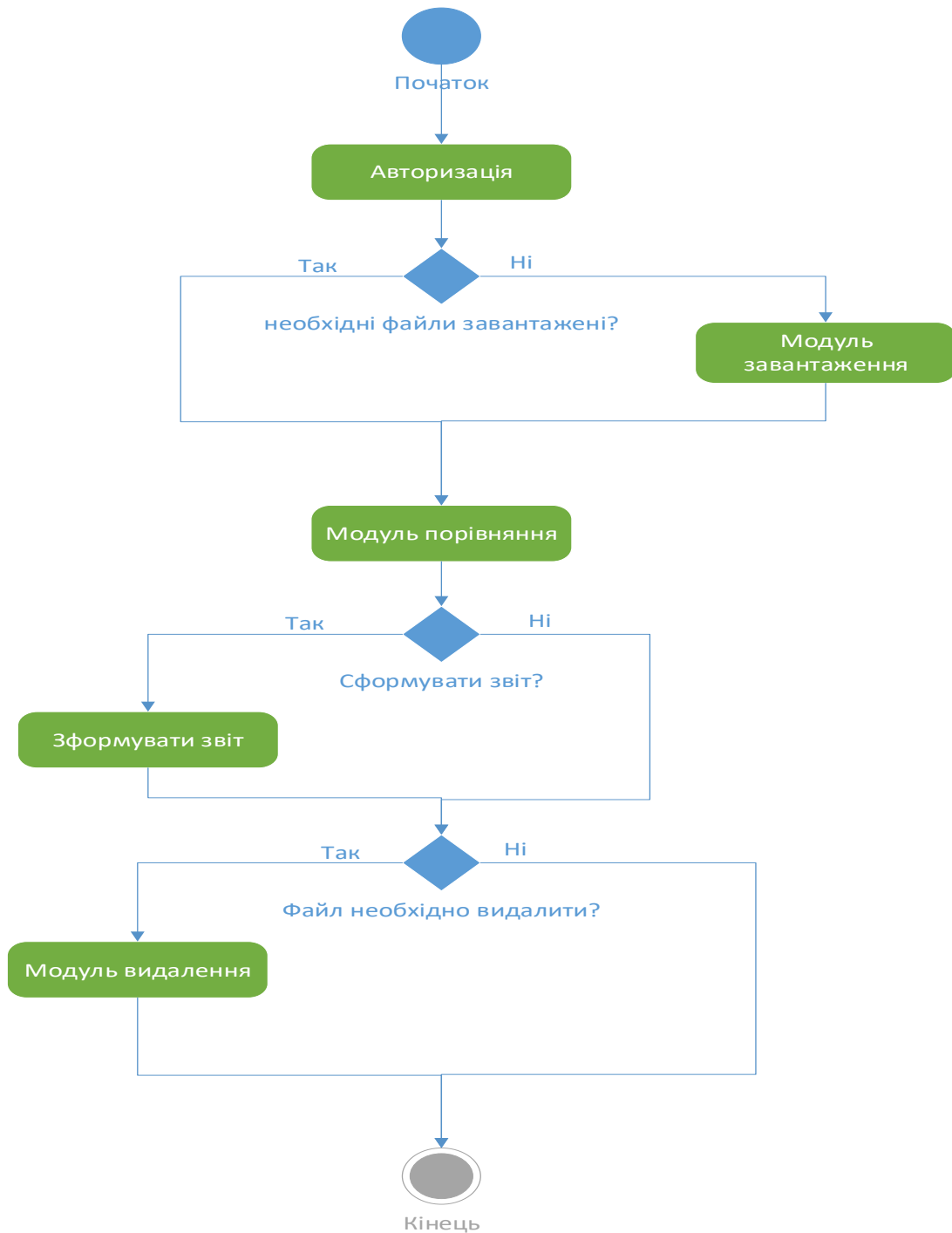


Рисунок 4.2 – Логіка роботи програми

4.2 Реалізація алгоритму отримання даних з файлу

Завдяки бібліотеці PhpSpreadSheet був реалізований основний функціонал модулю отримання даних з Excel-файлу (Рис 4.3). Під час зчитування формується

масив даних, з ключами які відповідають назві комірки. Це дозволяє ідентифікувати дані та працювати з ними у подальшому. Сервер отримує файли за допомогою вбудованої у PHP змінної файлів `$_FILES`, а введені користувачем дані відправляється через HTTP змінну `$_POST`. Метод POST, на відміну від GET, не має обмеження по об'єму даних та гарантує безпеку відправлених даних.

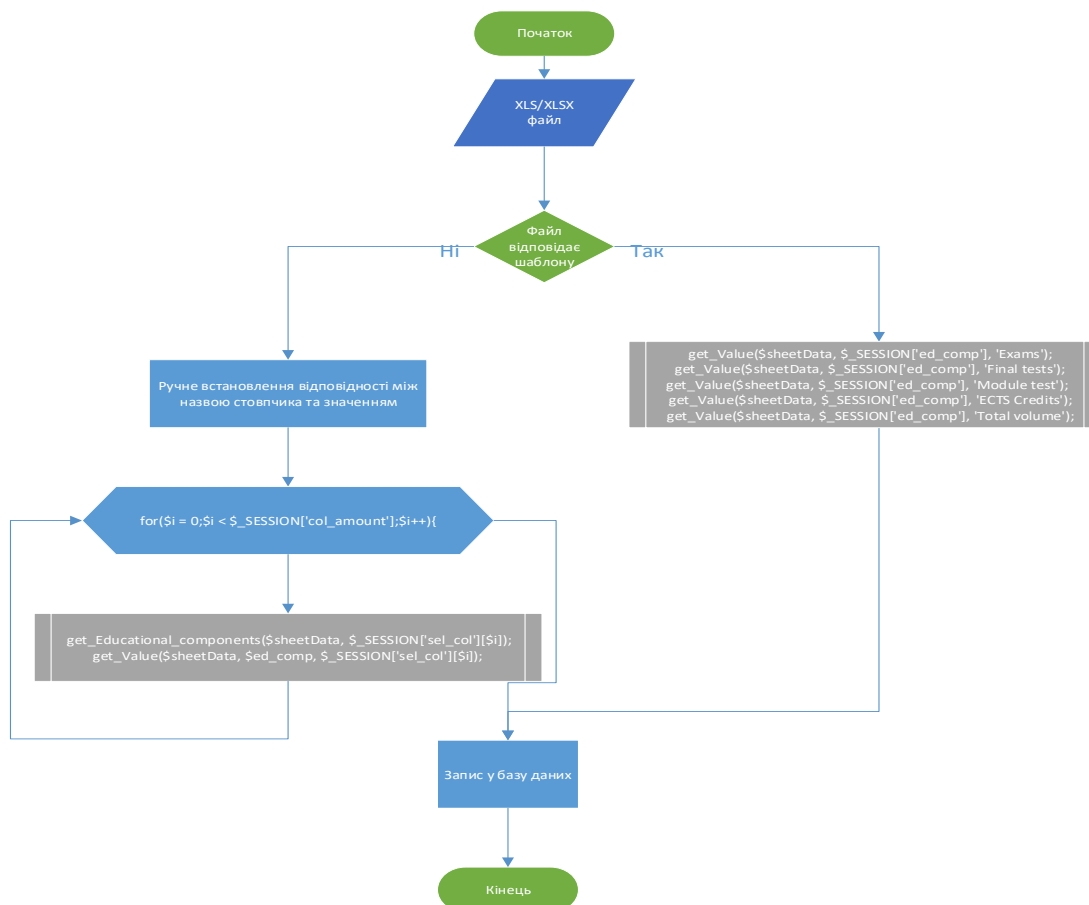


Рисунок 4.3 – Алгоритм отримання даних з файлу

4.3 Реалізація алгоритму порівняння файлів

Для реалізації алгоритму був використаний вбудований у PHP метод `similar_text()`, який працює на основі алгоритму Олівера (Рис 4.4): процес пошуку починається з порівняння кожного елемента першого рядка з кожним елементом другого рядка. Цей

процес відбувається двічі. У першому випадку перший рядок береться за еталон, у другому – другий. У процесі порівняння рахується кількість збігів та загальну кількість розглянутих випадків. Завдяки цьому вдається знайти значення схожості одного рядка з іншим у відсотках. Обрані користувачем дані беруться з бази даних, розшифровуються с JSON у звичайний текстовий масив. Таким чином, вони потрапляють у метод `similar_text()`, де обчислюється відсоток схожості рядків. Якщо відсоток схожості вище за 70%, то дані цих рядків будуть виведені у таблицю, після чого можна буде імпортувати їх у звіт.

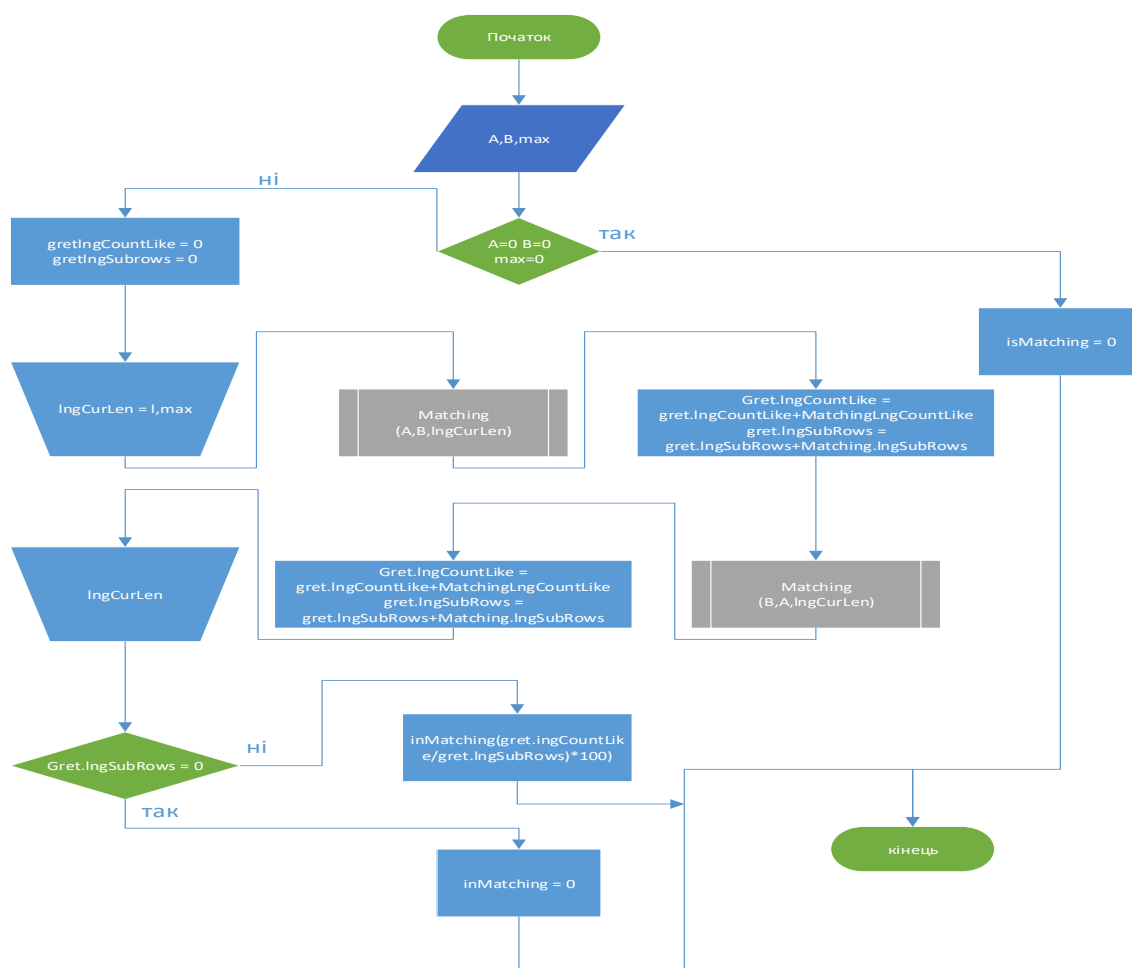


Рисунок 4.4 – Алгоритм Олівера

4.4 Опис бази даних

Для роботи програмного продукту було створено дві таблиці бази даних (Рис 4.5 – Структура бази даних). У першій, після зчитування файлу, програма зберігає отримані дані в БД, для подальшого порівняння їх з іншими файлами.

- Id – унікальний ідентифікатор файлу
- Name – назва файлу
- Components – назви дисциплін
- Exams – екзамени
- Final_tests – заліки
- Module_tests – модульні контрольні
- Credits – кредити
- Total – усього

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|----------------|---------------|--------------------|----------|------|--------------|-------------|----------------|----------|
| <input type="checkbox"/> | 1 id | int(11) | | | Нет | Нет | | AUTO_INCREMENT | |
| <input type="checkbox"/> | 2 name | text | utf8mb4_unicode_ci | | Нет | Нет | | | |
| <input type="checkbox"/> | 3 components | text | utf8mb4_unicode_ci | | Нет | Нет | | | |
| <input type="checkbox"/> | 4 exams | varchar(1024) | utf8mb4_unicode_ci | | Нет | Нет | | | |
| <input type="checkbox"/> | 5 final_tests | varchar(1024) | utf8mb4_unicode_ci | | Нет | Нет | | | |
| <input type="checkbox"/> | 6 module_tests | varchar(1024) | utf8mb4_unicode_ci | | Нет | Нет | | | |
| <input type="checkbox"/> | 7 credits | varchar(1024) | utf8mb4_unicode_ci | | Нет | Нет | | | |
| <input type="checkbox"/> | 8 total | varchar(1024) | utf8mb4_unicode_ci | | Нет | Нет | | | |

Рисунок 4.5 – Структура бази даних db_planmodule

Друга база даних створена для збереження даних користувачів (Рис 4.6).

- User_id – унікальний ідентифікатор користувача
- User_login – ім'я користувача в системі
- User_password – пароль користувача

– Module – список файлів, завантажених користувачем

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|-----|---------------|--------------|--------------------|------|--------------|-------------|----------------|----------|
| <input type="checkbox"/> | 1 | user_id | int(11) | | Нет | Нет | | AUTO_INCREMENT | |
| <input type="checkbox"/> | 2 | user_login | varchar(32) | utf8mb4_unicode_ci | Нет | Нет | | | |
| <input type="checkbox"/> | 3 | user_password | varchar(32) | utf8mb4_unicode_ci | Нет | Нет | | | |
| <input type="checkbox"/> | 4 | module | varchar(512) | utf8mb4_unicode_ci | Нет | Нет | | | |

Рисунок 4.6 – Структура бази даних db_useraccounts

Користувач має доступ тільки до тих файлів, які він сам і завантажив до системи.

Для цього реалізовано зв'язки між двома таблицями.

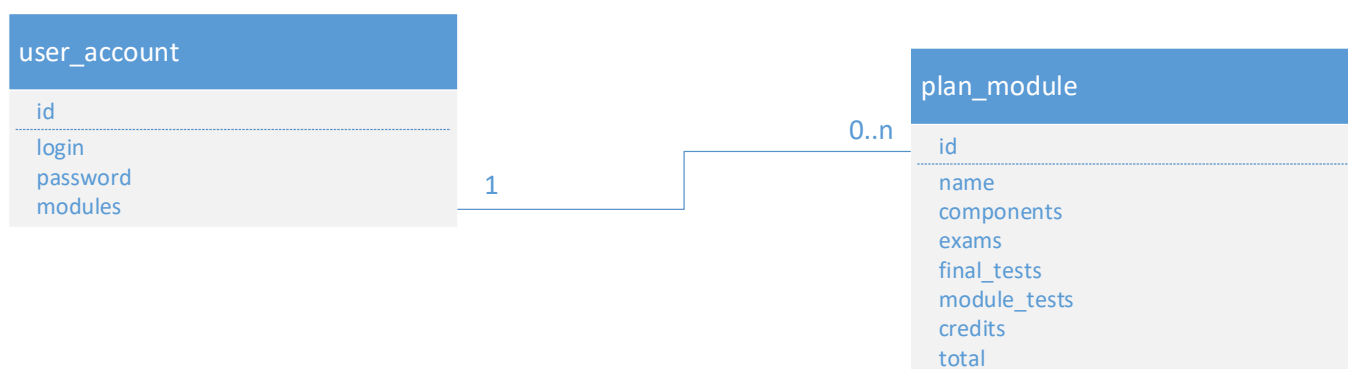


Рисунок 4.7 – Модель бази даних

4.5 Діаграма прецедентів системи

Діаграма прецедентів має ряд особливостей, а саме: система, що проектується представляється у вигляді сутностей або акторів, які взаємодіють із системою за допомогою варіантів використання. Варіант використання служить для опису сервісів, які система надає актору. Діаграма прецедентів програмного продукту представлена на Рис 4.7

Діаграма прецедентів застосовується для моделювання виду системи з точки зору зовнішнього спостерігача.

Актор — це будь-яка сутність, що взаємодіє з розробленою системою ззовні. Актор може бути людиною, іншою системою, підсистемою або класом.

Прецедент — це опис набору послідовних подій, що можуть виконуватись Актором в системі.

Для опису взаємодії акторів та прецедентів на діаграмі відображаються відносини.

Стандартні види відносин між акторами і подіями:

- асоціації (association);
- розширення (extend);
- узагальнення (generalization);
- включення (include).

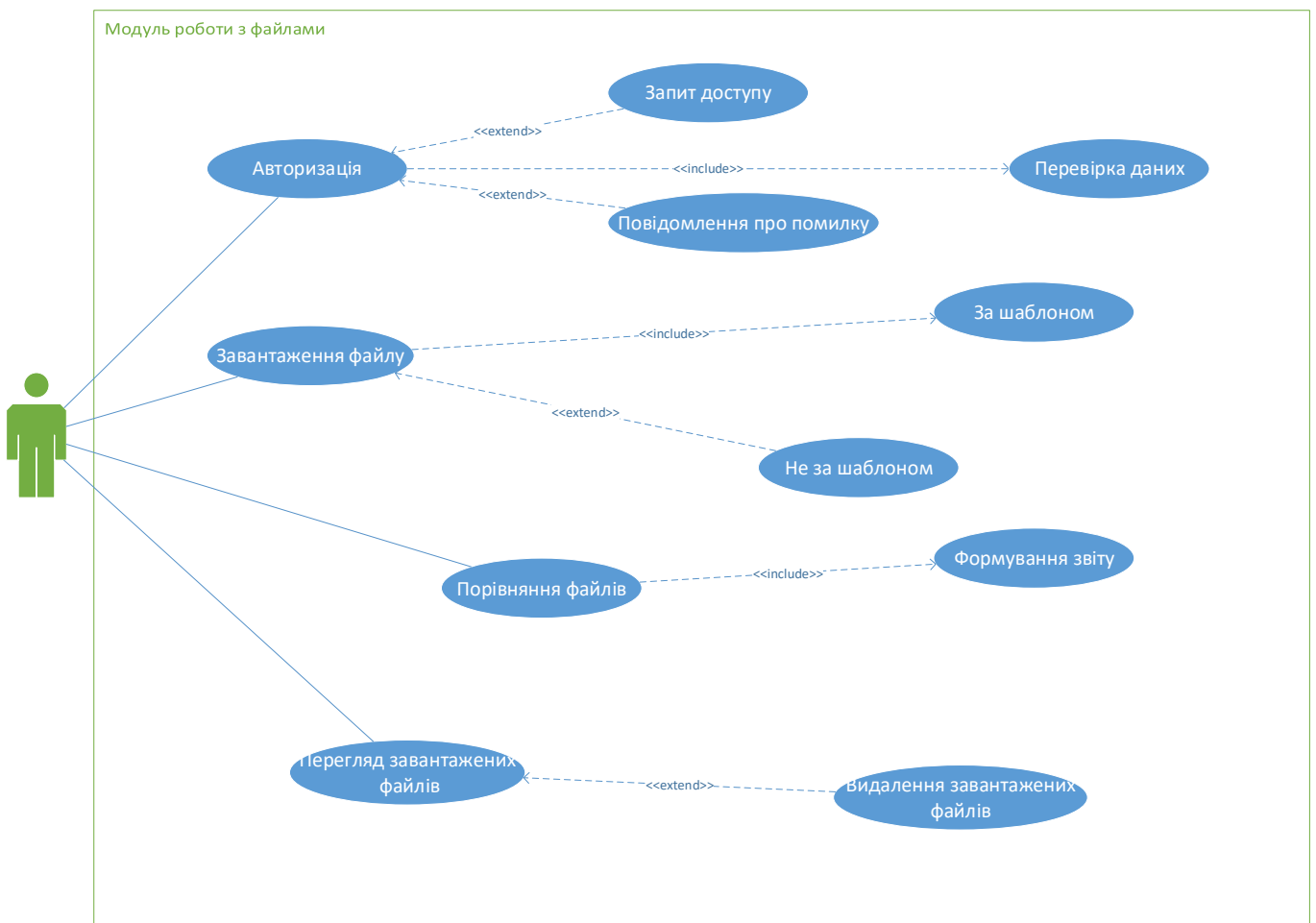


Рисунок 4.7 – Діаграма прецедентів

Головним актором є Користувач, який може авторизуватись чи відправити запит на доступ до веб-додатку. Після авторизації користувачу доступні три сторінки, кожна з яких відповідає за певний функціонал. Головний актор може завантажувати файли, порівнювати файли, переглядати та видаляти їх.

4.6 Діаграма розгортання

В UML діаграмі розгортання моделюють фізичну архітектуру системи. В ній відображаються відносини між програмним забезпеченням та апаратними компонентами в системі, а також фізичний розподіл обробки.

Діаграма розгортання показує фізичне розташування вузлів розподіленої системи, артефакти – програмні компоненти, які зберігаються на кожному вузлі, а також компоненти та інші елементи, які реалізують артефакти. Вузли представляють собою апаратні пристрої, такі як комп'ютери, датчики і принтери, а також інші пристрої, які підтримують середу виконання системи.

Діаграми розгортання ефективні для візуалізації, визначення та документування таких типів систем:

Вбудовані системи, що використовують обладнання, кероване зовнішніми стимулами

Клієнт-серверні системи, які зазвичай розрізняють призначений для користувача інтерфейс і постійні дані системи.

Розподілені системи, які мають кілька серверів і можуть містити кілька версій програмних артефактів, деякі з яких можуть навіть переноситися з вузла на вузол.

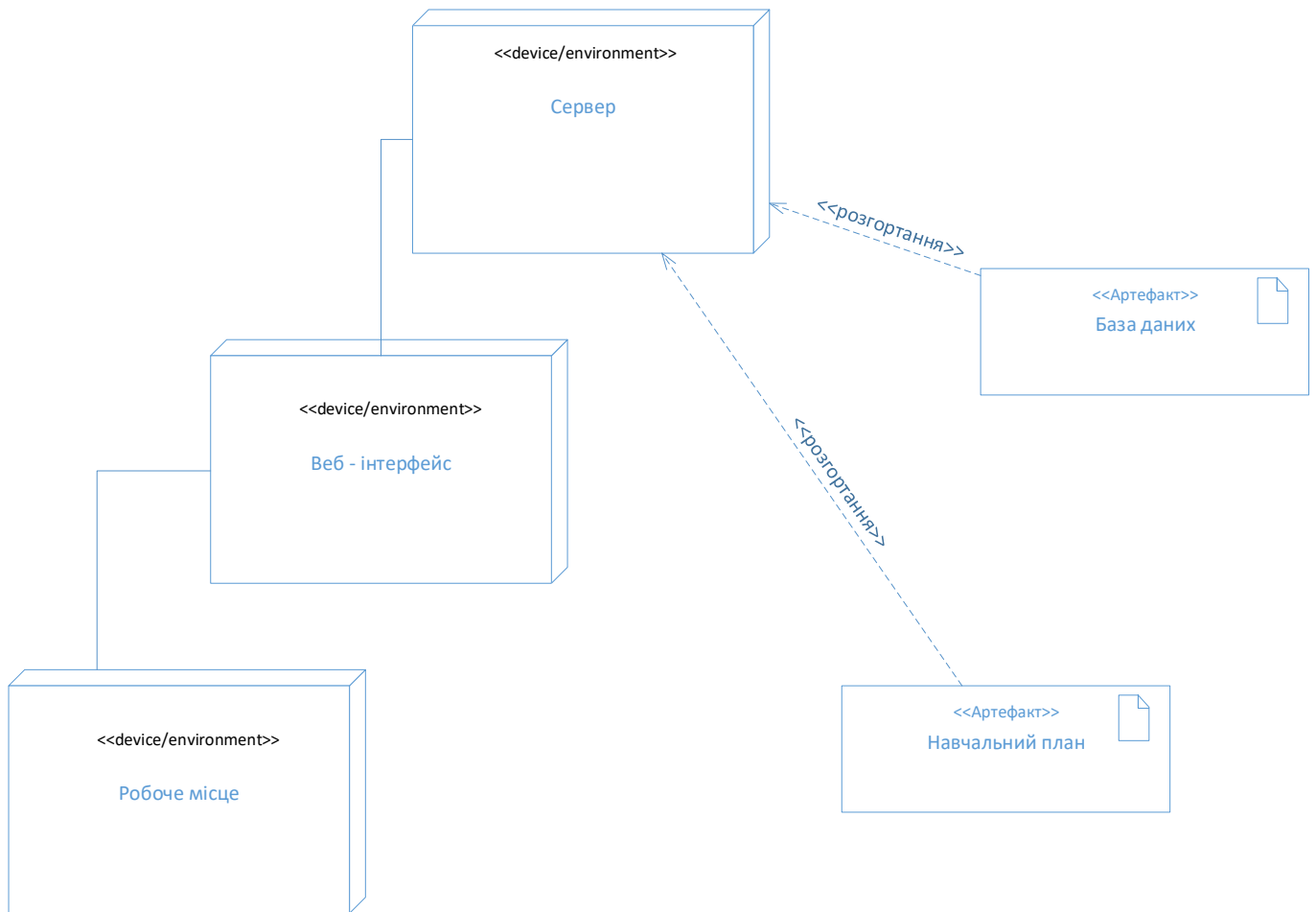


Рисунок 4.8 – Діаграма розгортання

5. РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

5.1 Авторизація

Робота з програмним забезпеченням починається з вікна авторизації (Рис 5.1 – Вікно авторизації). Неавторизовані користувачі не мають доступу до роботи з програмою, через наявність персональних даних.

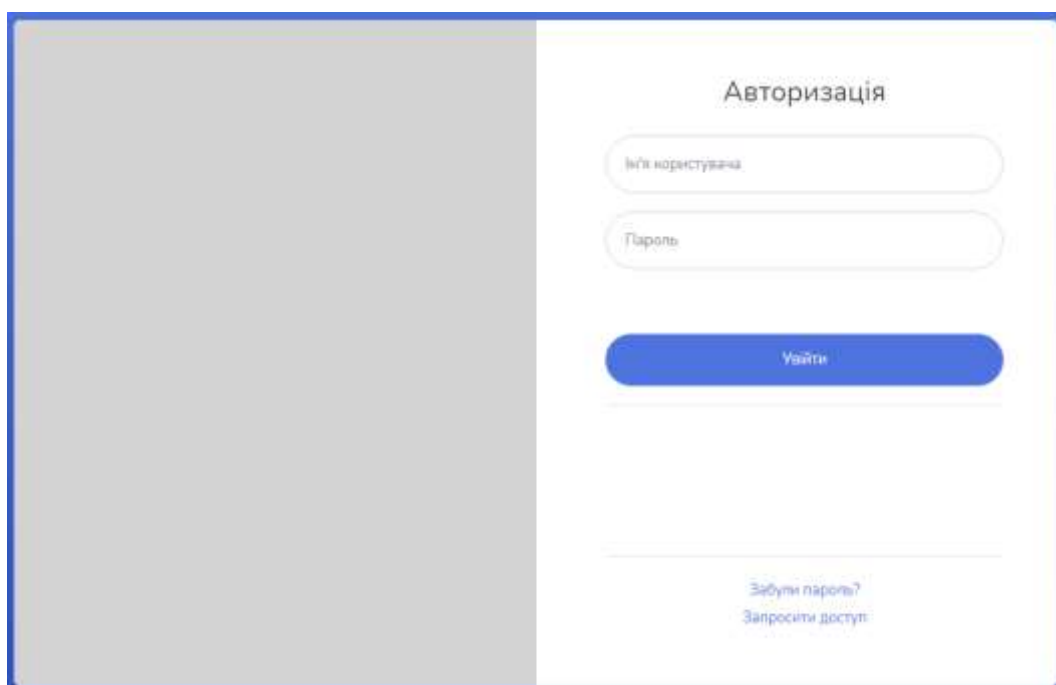


Рисунок 5.1 – Вікно авторизації

Програмне забезпечення формування навчальних планів для програм подвійного диплому для зручності користувача було поділено на три модулі:

- Модуль завантаження
- Модуль порівняння
- Модуль перегляду та видалення

5.2 Модуль завантаження

Для роботи модуля завантаження був розроблений спеціальний алгоритм, який би виконував усі поставлені перед програмний продуктом задачі. Проаналізувавши усі аналоги було використано найбільш вдалі

Модуль завантаження має два режими роботи: завантаження за шаблоном та довільне завантаження. Завантажувати файли можна тільки у форматі XLS або XLSX.

У режимі завантаження за шаблоном модуль може зчитувати дані, які строго структуровані у файлі. Таким чином, зчитування проходить значно швидше та без втручання користувача у сам процес (Рис 5.2 – Завантаження за шаблоном).

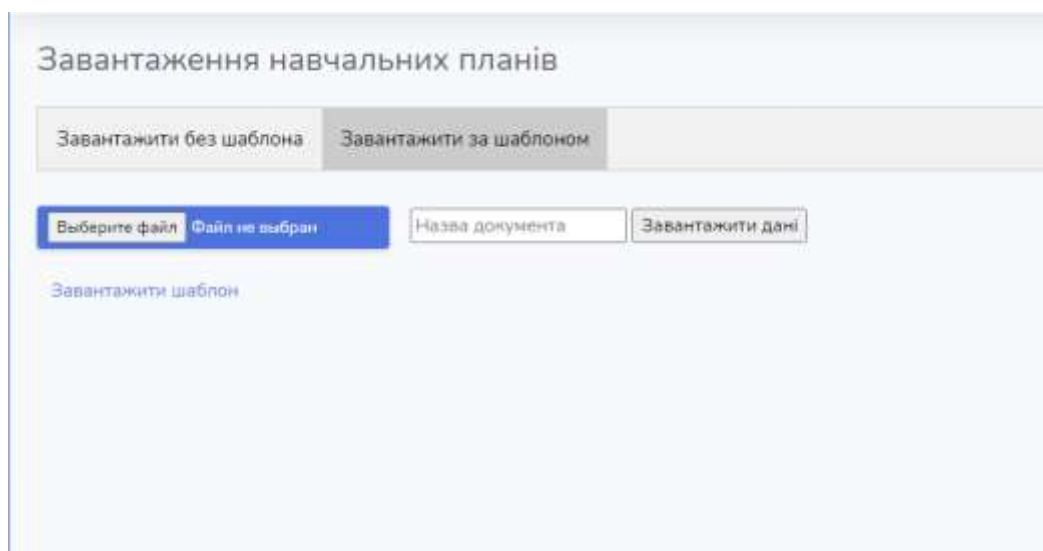


Рисунок 5.2 – Завантаження за шаблоном

Програмою передбачено перевірку файлу на відповідність до шаблону. У разі невідповідності, користувач буде повідомлений про те, що файл оформлений не за шаблоном і буде запропоновано завантажити пустий шаблон для заповнення (Рис 5.3 – Повідомлення про невідповідність шаблону).

Завантаження навчальних планів

Завантажити без шаблону Завантажити за шаблоном

Выберите файл Файл не выбран Назва документа Завантажити дані

Завантажити шаблон

Файл не відповідає шаблону!

Рисунок 5.3 – Повідомлення про невідповідність шаблону

У режимі довільного завантаження дані у файлі можуть бути розташовані у довільному порядку, але користувач повинен підписати кожен стовпчик, який він хоче завантажити, безпосередньо у інтерфейсі. Ті дані, що не потрібні, користувач може не відмічати (Рис 5.4 – Довільне завантаження).

Завантажити без шаблону Завантажити за шаблоном

Виберіть файл Файл не вибран Завантажити файл

Назва документа

Пропустити ▼ Semestre 7

Пропустити ▼ PHYSICS OF ENERGY

Пропустити ▼ 6

Предмети ▼ Heat Transfer

Модульні ▼ 30

Кредити ▼ 40

Завантажити дані

Рисунок 5.4 – Довільне завантаження

5.3 Модуль порівняння

Модуль порівняння працює з даними, які вже завантажені у базу даних. Користувач може обрати ті файли, які він хоче порівняти та отримає результат у вигляді таблиці (Рис 5.5 – Результат роботи модуля порівняння). Модуль розраховано на те, що назва полів двох файлів може відрізнитись один від одного, але буде мати однакове значення. Тому користувач може задати відсоток цієї похибки.

Порівняння навчальних планів

Обрати файл ▾ | Обрати файл ▾ | Завантажити файли

| Назва | Призначення | Екзамени | Залики | Модульні | Кредити | Загальна сума |
|-------|----------------------------------------------------------------------------------|----------|--------|----------|---------|---------------|
| File4 | Development of Internet of Things and networks applications | 0 | 2 | 2 | 3 | 90 |
| File1 | Development of Internet of Things and sensor networks applications | 1 | 0 | 1 | 4 | 120 |
| File4 | Development of Internet of Things and networks applications | 0 | 2 | 2 | 3 | 90 |
| File1 | Coursework on Development of Internet of Things and sensor networks applications | 0 | 3 | 0 | 1 | 30 |
| File4 | Software Innovation management | 0 | 3 | 2 | 2 | 60 |
| File1 | Sustainable innovative development | 0 | 3 | 1 | 2 | 60 |
| File4 | Software Innovation management | 0 | 3 | 2 | 2 | 60 |
| File1 | Software Innovation management | 0 | 3 | 2 | 3 | 90 |
| File4 | GRID technologies | 0 | 2 | 2 | 3 | 90 |
| File1 | Cloud and GRID technologies | 3 | 0 | 3 | 4 | 120 |

Рисунок 5.5 – Результат роботи модуля порівняння

Після завершення роботи модулю, користувач може сформувати звіт, натиснувши відповідну кнопку (Рис 5.6).

| | | | | | | |
|-------|------------------------------------|---|---|---|---|-----|
| File4 | GRID technologies | 0 | 2 | 2 | 3 | 90 |
| File1 | Cloud and GRID technologies | 3 | 0 | 3 | 4 | 120 |
| File4 | Sustainable Development | 0 | 3 | 2 | 2 | 60 |
| File1 | Sustainable innovative development | 0 | 1 | 1 | 2 | 60 |

Рисунок 5.6 – Функція формування звіту

Програма імпортує всі дані зі сторінки у XLSX файл для подальшої роботи з ними, додатково позначивши значення полів, які співпадають, а які відрізняються.

| | A | B | C | D | E | F | G | H |
|----|-------|-------------------------------------|---|---|---|---|-----|---|
| 10 | | | | | | | | |
| 11 | File4 | Software Innovative managemen t | 0 | 3 | 2 | 2 | 60 | |
| 12 | File1 | Software Innovative managemen t | 0 | 2 | 2 | 3 | 90 | |
| 13 | | | | | | | | |
| 14 | File4 | GRID technologies | 0 | 2 | 2 | 3 | 90 | |
| 15 | File1 | Cloud and GRID technologies | 3 | 0 | 3 | 4 | 120 | |
| 16 | | | | | | | | |
| 17 | File4 | Sustainable Developmen t | 0 | 3 | 2 | 2 | 60 | |
| 18 | File1 | Sustainable innovative developmen t | 0 | 1 | 1 | 2 | 60 | |
| 19 | | | | | | | | |

Рисунок 5.7 – Звіт у форматі XLSX

5.4 Модуль перегляду та видалення

Модуль перегляду та видалення створений для маніпуляцій з базою даних. Користувач може переглянути усі завантажені ним файли та за бажанням видалити їх (Рис 5.6 – Результат роботи модуля перегляду та видалення).

Перегляд навчальних планів

| Назва | Предмети | Екзамени | Залики | Модальні | Кредити | Загальна сума |
|-------|----------------------------------------------|----------|--------|----------|---------|---------------|
| File: | Heat Transfer | 30 | 40 | | | |
| | Fluid Mechanics | 30 | 40 | | | |
| | Embedded Systems & Data Acquisition | 27 | 43 | | | |
| | Machine Learning | 27 | 43 | | | |
| | Smart Metering & Energy Consumption Analysis | 33 | 37 | | | |
| | Building Information Management | 27 | 43 | | | |
| | Computational Fluid Dynamics | 33 | 37 | | | |
| | Power Grids | 27 | 43 | | | |
| | IoT for Smart City | 27 | 43 | | | |
| | Building Modeling | 27 | 43 | | | |
| | Renewable Energies | 36 | 34 | | | |

Рисунок 5.6 – Результат роботи модуля перегляду та видалення

5.5 Закінчення роботи з програмою

Після закінчення роботи з програмою користувач може вийти із свого облікового запису (Рис 5.7). Усі дані про поточну сесію будуть видалені.

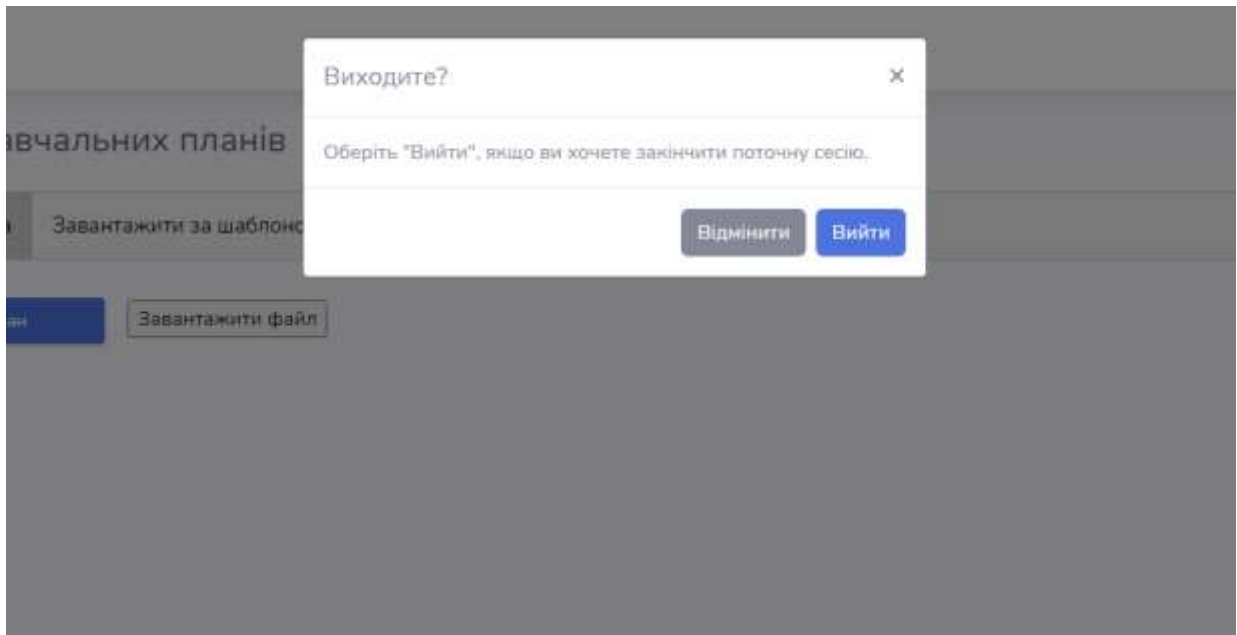


Рисунок 5.7 – Повідомлення при закінченні сесії

5.6 Висновки до розділу

У даному розділі описаний повний функціонал веб-додатку та принципи роботи з програмою. Наведено всі варіанти використання розробленої програмної системи користувачем та можливості кожного з модулів.

ВИСНОВКИ

Під час виконання дипломної роботи була досліджена технологія зчитування та обробки даних зі сторонніх файлів.

Проаналізовано існуючі рішення для роботи з файлами та їх даними. Подібні веб-додатки широко використовуються у багатьох сферах. Перевагами розглянутих рішень є коректність, швидкість у роботі та вміння працювати з великою кількістю даних. Недоліками є те, що багато аналогів використовують зчитування даних за шаблоном, що ускладнює роботу користувача з цим додатком.

Розроблено алгоритми роботи з файлами та їх даними, за якими система може зчитувати дані з файлу у двох режимах: за шаблоном та без шаблону. При зчитуванні з файлу не за шаблоном користувач може сам обрати які дані завантажити у базу даних, а які ні. Після цього файл зберігається на сервері, а його назва хешується для більшої надійності.

Розроблено модель бази даних, яка включає в себе дві таблиці: для збереження особистих даних користувачів та для збереження отриманих даних з файлу.

Розроблено клієнтську частину веб-додатку, яка забезпечує зручне користування як зі стаціонарного комп'ютера, так і з смартфона. Для зручності користувача функціонал був розділений на три вкладки, а дизайн веб-додатку зроблений у мінімалістичному стилі, для доступу навіть з поганою швидкістю інтернету.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PHP [Електронний ресурс] – Режим доступу: <https://www.php.net/docs.php>
2. Шварц Б. MySQL по максимуму. Оптимизация, репликация, резервное копирование / БэрнШварц., 2018. – 864 с. приложений / Ян Ф. Дарвин., 2019.
3. Фленаган Д. JavaScript. Карманный справочник / Девід Фленаган., 2020. – 320 с.
4. Apache [Електронний ресурс] – Режим доступу: <https://httpd.apache.org/>
5. HTML: Популярный самоучитель. А. А. Чиртик -Питер // СанктПетербург; 2008
6. Brackets [Електронний ресурс] – Режим доступу: <http://brackets.io/>
7. PHPSpreadsheet [Електронний ресурс] – Режим доступу: <https://phpspreadsheet.readthedocs.io/en/latest/>
8. Programming Classics: Implementing the World's Best Algorithms by Oliver / Oliver Y. 1994. – p.345-361
9. Fowler M. UML Distilled. A Brief Guide to the Standard Object Modeling Language. / Addison-Wesley, Pearson Education 2004. — P.155-180.
10. Levenstein method [Електронний ресурс] – Режим доступу - <https://www.php.net/manual/ru/function.levenshtein.php>
11. Wordpress [Електронний ресурс] – Режим доступу - <https://codex.wordpress.org/>
12. Woocommerce [Електронний ресурс] – Режим доступу - <https://docs.woocommerce.com/>
13. Merchant Center [Електронний ресурс] – Режим доступу - <https://docs.commercetools.com/merchant-center>
14. The Hash Function BLAKE. / Aumasson, J.-P., Meier, W., Phan, R., Henzen, L 2014. — P.85-130
15. OCRSpace [Електронний ресурс] – Режим доступу - <https://ocr.space/>
16. Bootstrap [Електронний ресурс] - <https://getbootstrap.com/docs/5.0/getting-started/introduction/>

17. JavaScript: The Definitive Guide: Activate Your Web Pages / David Flanagan 2011.

18. WAMPServer [Электронный ресурс] – Режим доступа -

<https://www.wampserver.com/en/>

ДОДАТОК А

Інструментальні засоби формування навчальних планів для програм
подвійного диплому

Специфікація

УКР.НТУУ"КП" _ТЕФ_АПЕПС_ТМ-71176_21Б

Аркушів 2

Київ — 2021

| Позначення | Найменування | Примітки |
|-----------------------------------------------|--------------|------------------------------------------------------|
| Документація | | |
| УКР.НТУУ"КП" _ТЕФ_АПЕПС_ ТІ6291_20Б | Записка.docx | Текстова частина дипломної роботи |
| Компоненти | | |
| УКР.НТУУ"КП" _ТЕФ_АПЕПС_ ТМ-71176_21Б 12-1 | parser.php | Компонент зчитування даних з XLS/XLSX файлу |
| УКР.НТУУ"КП" _ТЕФ_АПЕПС_ ТМ-71176_21Б 12-2 | upload.php | Компонент завантаження файлу на сервер |

ДОДАТОК Б

Інструментальні засоби формування навчальних планів для програм
подвійного диплому

Текст програми

УКР.НТУУ"КП" _ТЕФ_АПЕПС_ ТМ-71176_21Б 12-1

Аркушів 8

Київ — 2021

parser.php

```

<?php
require_once 'config.php';
require '../vendor/autoload.php';
use PhpOffice\PhpSpreadsheet\Spreadsheet;
use PhpOffice\PhpSpreadsheet\Writer\Xlsx;
use PhpOffice\PhpSpreadsheet\IOFactory;
function get_Value($sheetData, $ed_comp, $type){

    foreach($sheetData as $data){
        $key_temp = array_search($ed_comp[0], $data);
        if($key_temp == TRUE){
            break;
        }
    }

    foreach($sheetData as $data){
        $key = array_search($type, $data);
        if($key == TRUE){
            break;
        }
    }

    $exam = array();
    for($i = 0;$i <= sizeof($sheetData); $i++){

        if($sheetData[$i][$key] == NULL && is_numeric(array_search($sheetData[$i][$key_temp],
$ed_comp))){
            array_push($exam, 0);
        }

        elseif (!is_numeric($sheetData[$i][$key])){
            continue;
        }

        elseif($sheetData[$i][$key] == NULL && !is_numeric(array_search($sheetData[$i][$key_temp],
$ed_comp))){

```

```

        break;
    }
    else{
        array_push($exam, $sheetData[$i][$key]);
    }
}
return $exam;
}
function get_Educational_components($sheetData, $sed_comp_tmp){
    foreach($sheetData as $data){
        $key = array_search($sed_comp_tmp, $data);
        if($key == TRUE){
            break;
        }
    }
    if ($key == NULL){
        foreach($sheetData as $data){
            foreach($data as $d){
                preg_match("/[\d]+/", $d,$match);
                if (!empty($match)){
                    $temp_data = $data;
                    break 2;
                }
            }
        }
    }
    $sed_comp = array();
    for($i = 0;$i <= sizeof($sheetData); $i++){
        if($sheetData[$i][$key] == NULL){
            continue;
        }
    }
}

```

```

    if ($sheetData[$i][$key] == 'Educational components' || $sheetData[$i][$key] == 'V. Plan of
Educational process' || $sheetData[$i][$key] == '1. NORMATIVE educational components' ||
is_numeric($sheetData[$i][$key])

```

```

    || $sheetData[$i][$key] == '1.1. General training cycle' || $sheetData[$i][$key] == '1.2. Vocational
training cycle' || $sheetData[$i][$key] == 'Research (scientific) component' || $sheetData[$i][$key] == '2.
ELECTIVE educational components' || $sheetData[$i][$key] == '2.1. Vocational training cycle (Optional
subjects from Faculty catalogue)'){

```

```

        continue;

```

```

    }

```

```

    if($sheetData[$i][$key] == 'TOTAL'){

```

```

        array_push($ed_comp, $sheetData[$i][$key]);

```

```

        break;

```

```

    }

```

```

    else{

```

```

        array_push($ed_comp, $sheetData[$i][$key]);

```

```

    }

```

```

}

```

```

return $ed_comp;

```

```

if (isset($_SESSION['file'])){

```

```

    if ($_SESSION['file']['type'] == 'application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet'){

```

```

        $inputFileType = 'Xlsx';

```

```

    }

```

```

    $inputFileName = 'uploaded_files/' . $_SESSION['file']['name'];

```

```

    $reader = IOFactory::createReader($inputFileType);

```

```

    $reader->setReadDataOnly(true);

```

```

    $spreadsheet = $reader->load($inputFileName);

```

```

    $sheetData = $spreadsheet->getActiveSheet()->toArray(null, true, true, true);

```

```

if (isset($_SESSION['col_amount']) && isset($_POST[0])){

```

```

    for($i = 0;$i < $_SESSION['col_amount'];$i++){

```

```

        if($_POST[$i] == 'ed_comp'){

```

```

        $ed_comp = get_Educational_components($sheetData, $_SESSION['sel_col'][$i]);
    }
}
if($ed_comp != NULL){
    for($i = 0;$i < $_SESSION['col_amount'];$i++){
        if($_POST[$i] == 'none'){
            continue;
        }
        elseif($_POST[$i] != 'ed_comp'){
$_SESSION[$_POST[$i]]['name'] = get_Value($sheetData, $ed_comp, $_SESSION['sel_col'][$i]);
        }
        }
        }
        $_SESSION['ed_comp'] = $ed_comp;
    }
}

if(isset($_SESSION['file_template'])){
    if ($_SESSION['file_template']['type'] == 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet'){
        $inputFileType = 'Xlsx';
    }
    $inputFileName = 'uploaded_files/' . $_SESSION['file_template']['name'];

    $reader = IOFactory::createReader($inputFileType);
    $reader->setReadDataOnly(true);
    $spreadsheet = $reader->load($inputFileName);

    $sheetData = $spreadsheet->getActiveSheet()->toArray(null, true, true, true);
    $_SESSION['ed_comp'] = get_Educational_components($sheetData, 'Educational components');
    if (sizeof($_SESSION['ed_comp']) > 0){
        $_SESSION['col_name'][0] = get_Value($sheetData, $_SESSION['ed_comp'], 'Exams');
        $_SESSION['col_name'][1] = get_Value($sheetData, $_SESSION['ed_comp'], 'Final tests');
    }
}

```

```

$_SESSION['col_name'][2] = get_Value($sheetData, $_SESSION['ed_comp'], 'Module test');
$_SESSION['col_name'][3] = get_Value($sheetData, $_SESSION['ed_comp'], 'ECTS Credits');
$_SESSION['col_name'][4] = get_Value($sheetData, $_SESSION['ed_comp'], 'Total volume');

$data = $pdo->prepare('INSERT INTO ed_plan (name, components, exams, final_tests,
module_tests, credits, total) VALUES (:name, :components, :exams, :final_tests, :module_tests, :credits,
:total)');

$data->bindParam(':name', $_SESSION['name_template']);

$data->bindParam(':components', json_encode( $_SESSION['ed_comp']));

$data->bindParam(':exams', json_encode($_SESSION['col_name'][0]));

$data->bindParam(':final_tests', json_encode( $_SESSION['col_name'][1]));

$data->bindParam(':module_tests', json_encode($_SESSION['col_name'][2]));

$data->bindParam(':credits', json_encode($_SESSION['col_name'][3]));

$data->bindParam(':total', json_encode($_SESSION['col_name'][4]));

$data->execute();
}
else{
    $_SESSION['template'] = 1;
}
}

```

upload.php

```

<?php
session_start();

$message = "";
if (isset($_POST['uploadBtn'])){
    if (isset($_FILES['uploadedFile']) && $_FILES['uploadedFile']['error'] === UPLOAD_ERR_OK){
        // get details of the uploaded file
        $fileTmpPath = $_FILES['uploadedFile']['tmp_name'];
        $fileName = $_FILES['uploadedFile']['name'];
        $fileSize = $_FILES['uploadedFile']['size'];
        $fileType = $_FILES['uploadedFile']['type'];
        $fileNameCmps = explode(".", $fileName);
    }
}

```

```

$fileExtension = strtolower(end($fileNameCmps));

// sanitize file-name
$newFileName = md5(time() . $fileName) . '.' . $fileExtension;

// check if file has one of the following extensions
$allowedfileExtensions = array('xls', 'doc', 'xlsx', 'docx');

if (in_array($fileExtension, $allowedfileExtensions)){
    // directory in which the uploaded file will be moved
    $uploadFileDir = 'uploaded_files/';
    $dest_path = $uploadFileDir . $newFileName;

    if(move_uploaded_file($fileTmpPath, $dest_path)) {
        $message = 'File is successfully uploaded.';
    }
    else{
        $message = 'There was some error moving the file to upload directory. Please make sure the upload
directory is writable by web server.';
    }
}
else{
    $message = 'Upload failed. Allowed file types: ' . implode(',', $allowedfileExtensions);
}

}
else{
    $message = 'There is some error in the file upload. Please check the following error.<br>';
    $message .= 'Error:' . $_FILES['uploadedFile']['error'];
}

$_SESSION['message'] = $message;
$_FILES['uploadedFile']['name'] = $newFileName;

```

```

$_SESSION['file'] = $_FILES['uploadedFile'];
header("Location: index.php");
}

if (isset($_POST['uploadBtn2'])){
    $_SESSION['name_template'] = $_POST['name_template'];
    if (isset($_FILES['uploadedFile_tmp']) && $_FILES['uploadedFile_tmp']['error'] ===
    UPLOAD_ERR_OK){

        // get details of the uploaded file
        $fileTmpPath = $_FILES['uploadedFile_tmp']['tmp_name'];
        $fileName = $_FILES['uploadedFile_tmp']['name'];
        $fileSize = $_FILES['uploadedFile_tmp']['size'];
        $fileType = $_FILES['uploadedFile_tmp']['type'];
        $fileNameCmps = explode(".", $fileName);
        $fileExtension = strtolower(end($fileNameCmps));

        // sanitize file-name
        $newFileName = md5(time() . $fileName) . '.' . $fileExtension;

        // check if file has one of the following extensions
        $allowedfileExtensions = array('xls', 'doc', 'xlsx', 'docx');

        if (in_array($fileExtension, $allowedfileExtensions)){
            // directory in which the uploaded file will be moved
            $uploadFileDir = 'uploaded_files/';
            $dest_path = $uploadFileDir . $newFileName;

            if(move_uploaded_file($fileTmpPath, $dest_path)) {
                $message ='File is successfully uploaded.';
            }
        }
        else{

```

```
$message = 'There was some error moving the file to upload directory. Please make sure the upload directory is writable by web server.';
```

```
}
```

```
}
```

```
else{
```

```
    $message = 'Upload failed. Allowed file types: ' . implode(',', $allowedfileExtensions);
```

```
}
```

```
}
```

```
else{
```

```
    $message = 'There is some error in the file upload. Please check the following error.<br>';
```

```
    $message .= 'Error: ' . $_FILES['uploadedFile_tmp']['error'];
```

```
}
```

```
    $_SESSION['message'] = $message;
```

```
$_FILES['uploadedFile_tmp']['name'] = $newFileName;
```

```
$_SESSION['file_template'] = $_FILES['uploadedFile_tmp'];
```

```
header("Location: index.php");
```

```
}
```

ДОДАТОК В

Інструментальні засоби розпізнавання поведінки людини з
використанням сенсорів мобільного телефону

Опис програми

УКР.НТУУ"КПІ" _ТЕФ_АПЕПС_ ТМ-71176_21Б 13-1

Аркушів 8

Київ — 2021

АНОТАЦІЯ

Додаток містить опис веб-застосунку для формування навчальних планів для програм подвійного диплому, що виконує деякі завдання, визначенні в розділі 1, а саме:

- зчитування даних з файлу
- безпечне збереження даних на сервері
- порівняння даних між собою
- формування звіту
- видалення файлів з серверу

Веб-застосунок розроблений мовами програмування HTML, CSS, JS, PHP з використанням платформи WAMP у редакторі вихідного коду Brackets.

ЗМІСТ

| | |
|------------------------------------------|----|
| 1. ЗАГАЛЬНІ ВІДОМОСТІ..... | 71 |
| 2. ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ | 72 |
| 3. ОПИС ЛОГІЧНОЇ СТРУКТУРИ | 73 |
| 4. ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ | 74 |
| 5. ВХІДНІ ДАНІ | 75 |
| 6. ВИХІДНІ ДАНІ..... | 76 |

ЗАГАЛЬНІ ВІДОМОСТІ

У додатку міститься опис веб-додаток для формування навчальних планів для програм подвійного диплому, що виконує деякі завдання, визначенні в розділі 1. Додаток Б містить програмний код цих компонентів.

Для роботи веб-додатку необхідний бути встановлений хоча б один із веб-браузерів: Chrome, Opera, Safari, Microsoft Edge, Internet Explorer 10. У випадку розгортання програми не на локальному сервері необхідний доступ до мережі інтернет.

Веб-застосунок розроблений мовами програмування HTML, CSS, JS, PHP з використанням платформи WAMP у редакторі вихідного коду Brackets.

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Розроблений веб-додаток надає користувачу наступний функціонал:

- Реєстрація/авторизація в системі.
- Зчитування даних з файлу.
- Безпечне збереження даних на сервері.
- Порівняння даних між собою.
- Формування звіту.
- Видалення файлів з серверу.

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Згідно з архітектурою, веб-додаток складається з трьох модулів: модуль завантаження, модуль порівняння, модуль перегляду та видалення.

Програма виконує функцію завантажування даних у базу даних, порівняння їх та видалення за бажанням користувача. Формат та структура вихідних файлів може відрізнятися.

Система відповідає усім нормам безпеки особовистих даних: усі завантажені файли повинні шифруватись за алгоритмом SHA-3. Доступ до системи відбувається тільки після логіну та паролю.

Система виконує функцію зчитування даних з XLS файлів, враховуючи те, що стовпці з даними можуть знаходитись у довільному порядку. Після отримання даних програмою, користувач може переглядати, порівнювати та видаляти дані з бази.

ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ

Для роботи веб-додатку необхідний бути встановлений хоча б один із веб-браузерів: Chrome, Opera, Safari, Microsoft Edge, Internet Explorer 10.

Файли додатку повинні розміщуватись або на хостингу, або на локальному сервері. У випадку розгортання програми не на локальному сервері необхідний доступ до мережі інтернет.

ВХІДНІ ДАНІ

Вхідною інформацією для формування навчальних планів для програм подвійного диплому є навчальні плани у вигляді електронної таблиці. У випадку, коли файл з електронною таблицею вже знаходиться на сервері – дані з бази даних.

ВИХІДНІ ДАНІ

Вихідною інформацією є HTML таблиця з результатами порівняння двох файлів. За бажанням користувача цю таблицю можна завантажити у форматі XLSX для подальшої роботи з даними.