

Міністерство освіти та науки України
Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

Кваліфікаційна робота
за освітнім ступенем «бакалавр»
на тему:
«Вебінтерфейс візуалізації освітнього процесу ФМІ»

Виконав:

здобувач IV курсу

групи КН-41

спеціальності 122

«Комп'ютерні науки»

Марон Назарій Дмитрович

Науковий керівник:

д.п.н., професор Шліхта Г.О.

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ	5
1.1 Стан дослідженості проблеми;	5
1.2 Огляд системи розкладу та системи репозитарію;	6
1.3 Обґрунтування підходу;	6
РОЗДІЛ 2. МЕТОДОЛОГІЯ ТА АНАЛІЗ ВИМОГ ДО СИСТЕМИ	8
2.1 Аналіз вимог предметної області;	8
2.2 Підходи до виконання;	9
2.3 Проєктування інтерфейсу;	10
РОЗДІЛ 3. МОДЕЛЮВАННЯ	12
3.1 Постановка задачі;	12
3.2 Опис моделі даних;	13
3.3 Алгоритм вирішення задачі;	14
РОЗДІЛ 4. ПРАКТИЧНА РЕАЛІЗАЦІЯ, РОЗГОРТАННЯ ТА ТЕСТУВАННЯ	16
4.1 Вибір засобів;	16
4.2 Доречність плагіну Code Snippets	17
4.3 Технічні моменти;	18
4.4 Розгортання;	19
4.5 Результати.	35
ВИСНОВКИ	37
СПИСОК ЛІТЕРАТУРИ	39

ВСТУП

На сьогоднішній день більша частина всієї інформації стає цифровою. Навчальні заклади також не є виключенням: вони переходять на електронні журнали, бази даних, курси.

Одним із найважливіших етапів навчання у ВНЗ є наукові роботи студентів, а саме – написання кваліфікаційних робіт. Фінальний результат навчання, який підбиває підсумки, та показує, чого навчився студент за увесь час.

Але, часто буває, що після захистів ці фінальні роботи просто губляться та зникають в паперових архівах та жорстких дисках кафедри. Тому було вирішено створити електронний репозитарій.

Це спеціальні каталоги, де зібрана інформація про всі здані та захищені роботи. Це дозволяє зробити науку відкритою, допомагає іншим студентам знайти приклади для власних робіт та досліджень, та підвищує престиж самого університету в інтернеті.

Актуальність даної теми. Розробка системи репозитарію кваліфікаційних робіт є дуже доречною і необхідною з кількох причин:

Наразі часто облік робіт ведеться застарілими методами – у паперових журналах або в таблицях, які складно об'єднати в один список для пошуку.

Викладачам кафедри постійно треба готувати звіти про здані та захищені роботи. Наявність автоматизованої системи значно спрощує увесь процес, та дозволяє зробити такий звіт за пару секунд замість того, щоб шукати та переглядати переглядати папки та файли за минулі роки.

Також діяльність університетів має бути відкритою та прозорою. Це також стосується і результатів навчання та робіт.

Мета роботи. Головною метою є створення зручної та простої для використання веб інтерфейсу для візуалізації компонентів освітнього процесу для кафедри інформаційних технологій та моделювання. А саме: інтерактивний розклад занять і репозитарій кваліфікаційних робіт. Це повинно дозволяти легко завантажувати дані про роботи, швидко знаходити ці дані за допомогою фільтрів та пошуку, наприклад: за роком, за спеціальністю чи за керівником і переглядати інформацію про кожну роботу, та упростити пошук інформації про освітній процес, а саме – актуальний розклад.

Завданням роботи для досягнення поставленої мети є:

- Проаналізувати системи розкладу та репозитаріїв;
- Вибрати технології для реалізації та розробки, які будуть простими в обслуговуванні та безкоштовними;
- Створити структуру бази даних на основі CSV-файлу, щоб можна було легко вносити нову інформацію без спеціальних знань, а також використати як базу даних репозитарій на GitHub з файлами розкладу у форматі json;
- Написати код програми, який буде читати файл з даними та виводити їх на сайт у вигляді таблиці;
- Впровадити систему на сайт кафедри.

Об'єкт дослідження – процес збору, зберігання та надання відкритого доступу до даних про кваліфікаційні роботи та розкладу в університеті.

Предмет дослідження – програмні засоби та веб технології для побудови інтерфейсу візуалізації освітнього процесу.

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ

1.1 Стан дослідженості проблеми

Цифровізація вищої освіти в Україні регулюється МОН щодо створення єдиного середовища інформації, де візуалізація освітніх даних є критичною для навчального процесу.

Сучасна цифровізація вищої освіти в Україні базується на створенні єдиного інформаційного простору, що вимагає відмови від статичних документів на користь інтерактивних веб інтерфейсів, так як це підвищує ефективність в управлінні навчальним процесом.

Кваліфікаційна робота є обов'язковим етапом завершення навчання здобувачами вищої освіти для бакалаврату та магістратури.

Після успішного захисту перед екзаменаційною комісією, ці роботи повинні зберігатися на кафедрі.

Проте, застарілі підходи до зберігання робіт має низку суттєвих недоліків. На кафедрі створюються списки у звичайних файлах Excel, а самі тексти робіт зберігаються або в паперовому вигляді в архіві, або у вигляді файлів на комп'ютерах представників кафедри.

Публічний доступ до захищених робіт найчастіше відсутній, і отримати інформацію можна лише за спеціальним запитом.

Такий стан речей створює деякі проблеми як для студентів, так і для викладачів. Здобувачі не можуть переглянути тему попередніх років. Викладачам складно формувати звіти для комісій. Тому виникає потреба у створенні відкритого репозитарію, який працюватиме на сайті кафедри.

Інтерактивний веб інтерфейс в свою чергу забезпечує відкритий доступ до розкладу занять, який завжди актуальний та змінюється.

1.2 Огляд системи розкладу та системи репозитарію

Аналіз існуючих рішень, таких, як ASC TimeTables чи FET показує, що вони часто є перевантаженими для потреб окремої кафедри, що робить доречною організацію розробки кастомних модулів.

А системи, такі як DSpace та EPrints є потужними, проте вимагають складного технічного обслуговування та вимагають спеціальних знань.

Наприклад: для нашої мети – такий складний підхід не є доречним. Потрібно було знайти рішення, яке буде максимально швидким, зрозумілим, простим та не вимагаючим складних кроків для персоналу кафедри.

Тому наш підхід спрощує роботу персоналу.

1.3 Обґрунтування підходу

Проаналізувавши недоліки існуючих систем, було прийнято рішення розробити власний модуль, інтегрований у систему управління контентом WordPress, на якій працює сайт кафедри.

Об'єднання розкладу та репозитарію в єдиній системі на сайті кафедри створює можливість для студента, де вся необхідна інформація доступна в браузерному вікні. Це значно знижує технічні питання та витрати на обслуговування платформ.

Тому основою рішення став принцип максимальної зручності. Замість створення багаторівневої бази даних із розкладами та із детальними профілями кожної роботи, було обрано формат інтерактивної таблиці з прямим доступом до файлів робіт та інтерактивного застосунку з інформацією про актуальний розклад.

Головними перевагами такого підходу є:

- Швидкість доступу. В цій системі користувач бачить усю необхідну інформацію одразу в браузері. Також замість переходу на окремі сторінки, користувач просто натискає кнопку «Відкрити», і файл з роботою миттєво відкривається у поточній вкладці браузера. Це робить роботу з репозитарієм швидкою.
З розкладом теж все просто: користувач отримує розклад своєї групи чи викладача прямо під формою.
- Простота керування. Дані для таблиці беруться зі звичайного файлу формату CSV. Не потрібно розбиратися в адмін-панелях, достатньо просто додати новий рядок у файлі та завантажити його на сайт.
- Миттєва фільтрація. Завдяки використанню JavaScript бібліотеки DataTables, пошук та сортування відбувається миттєво, без перезавантаження сторінки.
- Захист даних. Для робіт студентів, які були відраховані, або чий роботи мають обмежений доступ, кнопка відкриття миттєво змінюється на текстове повідомлення «Немає доступу», що гарантує дотримання внутрішніх правил.

Таким чином, цей підхід дозволяє вирішити всі поставлені задачі з мінімальними витратами ресурсів, також додає при цьому швидкий, сучасний та зручні інтерфейси для користувача.

РОЗДІЛ 2. МЕТОДОЛОГІЯ ТА АНАЛІЗ ВИМОГ ДО СИСТЕМИ

2.1 Аналіз вимог предметної області

Перед початком роботи, потрібно було зрозуміти, хто саме і як буде користуватися репозитарієм. Аналіз предметної області показав, що у системи є три основні групи користувачів зі своїми вимогами у кожного:

- Студенти та відвідувачі або гості сайту. Для них головне – це зручність у використанні та швидкість. Вони заходять на сайт, щоб отримати конкретну інформацію максимально зручно та швидко, за ключовим словом або фільтром. Система має працювати миттєво, без завантажень сторінок. Також з можливістю переходу на онлайн пари.
- Представники кафедри та ті, хто керує даними. Це люди, які оновлюють базу даних та щороку додають нові роботи. Зазвичай, вони не мають навичок роботи зі складними базами даних та програмуванням. Тому головна вимога від них – максимальна простота використання та оновлення інформації. Процес має бути таким ж простим, як просте збереження таблиці Excel.
- Для адміністраторів сайту важливо ж, щоб наш репозитарій не створював надмірне навантаження на сервер та не зламав сайт.

Враховуючи ці спеціальні умови, було сформовано головне правило розробки: система має бути легкою, не потребувати встановлення тяжких плагінів і працювати всередині сайту кафедри автоматично.

2.2 Підходи до виконання

Для розв'язання поставлених задач було обрано методологію мінімальної складності. Цей підхід означає, що замість реалізації мега складної архітектури, розробник обирає найпростіший шлях виконання поставленої задачі.

Зокрема, використано такі підходи:

- Відмова від реляційних баз даних на користь плоских файлів. Замість того, щоб створювати окремі таблиці в базах даних сайту, було вирішено використати звичайний файл CSV, який містить інформацію. Це ідеально задовольняє вимогу персоналу. Тобто, вони продовжують вести облік у звичайному Excel, а потім просто зберігають його у форматі CSV, перетягують в медіатеку адмін-панелі, щоб по посиланню підв'язати оновлену таблицю до репозитарію. Також у випадку з розкладом: не треба постійно оновлювати статичні файли, все робиться на GitHub, файли формату json важать небагато, а інформація миттєво структурується в таблицю прямо під формою в залежності від запиту користувача.
- Обробка даних для гостей та відвідувачів. Зазвичай, коли користувач застосовує фільтр на сайті, браузер відправляє запит на сервер, сервер шукає дані та повертає нову сторінку. Тут застосовано інший підхід: сервер лише один раз віддає таблицю повністю, а пошук, сортування та фільтрація відбуваються прямо в браузері користувача за допомогою JavaScript. Пошук та обробка інформації стає миттєвим.

2.3 Проектування інтерфейсу

На основі вимог та обраних підходів було розроблено загальну концепцію роботи системи. Складається все з трьох компонентів:

- Сховище даних. Тут ми маємо CSV-файл, що містить чітку структуру колонок, яка вміщає в себе необхідну інформацію, тобто: ПІБ здобувача, рік, ступінь, спеціальність, тема, науковий керівник та пряме посилання на файл роботи.
- Обробник. Обробником в нас є PHP-код. Спеціальний модуль, яка вбудовується в будь-яку сторінку на сайті. Він автоматично відкриває CSV-файл з медіабібліотеки, зчитує рядки з цього файлу і формує з них HTML-таблицю. Також реалізована концепція безпеки: якщо в темі вказано, що студент відрахований – програма автоматично блокує виведення посилання на файл, вставляючи текст «Немає доступу»
- Інтерфейс користувача. Для візуальної частини маємо потужну бібліотеку DataTables. Вона перетворює HTML-таблицю на інтерактивний додаток. Саме вона відповідає за швидкий доступ. Користувачу виводяться зручні списки для фільтрації та поле пошуку.

Також варто звернути увагу на прямий доступ до файлу. У системі присутній більш простий для користувачів підхід: кнопка «Відкрити» знаходиться в таблиці, що дозволяє переглядати роботи в один клік.

Розклад:

Потенціал застосунку полягає саме в зв'язку з віддаленим репозитарієм на GitHub. Оскільки я не створював власну базу даних з розкладом, а використав вже існуючі дані, розташовані там, тому метою створення цього застосунку було миттєве виведення інформації на сторінці.

Місце зберігання даних:

Всі назви груп зберігаються в папці, розміщеній на GitHub.

Програма звертається до репозитарію GitHub, який містить файл з назвами груп та розкладами, після чого дані виводяться на поточній сторінці.

РОЗДІЛ 3. МОДЕЛЮВАННЯ

3.1 Постановка задачі

Головна задача розробленого модуля полягає в автоматичному перетворенні звичайних таблиць в CSV файлах та файлах json у зручний веб-інтерфейс.

Цю задачу можна описати як процес перетворення вхідного масиву даних з файлу на вихідний HTML-код.

- Вхідні дані. Текстовий файл у форматі CSV, де кожен рядок – це запис про якусь одну кваліфікаційну роботу.
- Умови обробки. Програма відсіює перший рядок, потім перевіряє тему роботи. Якщо в темі є слово «Відрахований» або відсутнє посилання, програма блокує доступ до файлу для звичайних користувачів.
- Вихідні дані. Згенерована HTML-таблиця, до якої підключаються JavaScript для миттєвого пошуку та фільтрації.

Розклад:

Структура включає в себе ідентифікатори груп, назви дисциплін, ПІБ, посилання на онлайн заняття та дні тижня.

3.2 Опис моделі даних

Більшість веб додатків використовують складні бази даних. Проте в цій задачі був використаний крок по простіше. Ми використали модель «Пласкої бази даних».

Модель даних репозитарію базується на таблиці, яка містить фіксовану кількість стовпців.

Для коректної роботи алгоритму була затверджена така структура моделі:

- ПІБ студента;
- Рік захисту;
- Освітній ступінь;
- Спеціальність;
- Тема кваліфікаційної роботи;
- Науковий керівник;
- Посилання.

Така структура моделі дозволяє формувати дані у звичайному редакторі Excel, зберігати їх у форматі CSV, і просто замінювати старий файл на відредагований новий. Програма автоматично підтягне все з нового файлу.

3.3 Алгоритм вирішення задачі

Робота системи поділена на два етапи: обробка даних на стороні сервера та взаємодія візуальна з користувачем на сторінці.

Ось загальний алгоритм:

1) Етап перший. Серверна генерація.

- Користувач відкриває сторінку репозитарію на сайті;
- Система управління зустрічає спеціальний спеціальний шорткод і запускає написаний на PHP скрипт;
- PHP звертається до сервера і відкриває файл за посиланням, що вказано в коді за допомогою функції `fopen()`;
- Запускається цикл `while`, який читає файл рядками по черзі за допомогою функції `fgetcsv()`;
- Алгоритм пропускає перші заголовки, бо там ми маємо технічні заголовки;
- Для кожного наступного рядка алгоритм перевіряє стовпець з темою роботи на наявність слова «Відрахований». А для любого регістру використовується функція `mb_stripos`;
- Формується рядок таблиці і якщо перевірка виявила відрахованого студента, або колонка з посиланням порожня – алгоритм генерує «Немає доступу». В іншому випадку генерується кнопка з вбудованим посиланням.
- Після проходження по всіх рядках скрипт закриває файл і відправляє таблицю в браузер.

2) Етап другій. Інтерактивність для клієнтів.

- Браузер користувача завантажує згенеровану HTML-таблицю;
- Автоматично підключається бібліотека jQuery та плагін DataTables.js;
- Плагін сканує колонки таблиці, знаходить унікальні значення та автоматично наповнює ними випадючі списки;
- Коли користувач вводить текст у пошук, чи обирає щось у фільтрах – алгоритм DataTables миттєво приховує зайві рядки за допомогою CSS;
- При натисканні «Скинути» всі фільтри очищуються, і таблиця повертається до початкового стану.

Такий алгоритм дозволяє зняти надлишкове навантаження з бази даних сайту, і таблиця повертається до початкового стану.

В інтерактивній формі з розкладами ж використовується алгоритм завантаження даних через `fetch()`, фільтрацію за групою, ітерацію по днях тижня для побудови графіку занять.

РОЗДІЛ 4. ПРАКТИЧНА РЕАЛІЗАЦІЯ, РОЗГОРТАННЯ ТА ТЕСТУВАННЯ ПРОДУКТУ

4.1 Вибір засобів

Для того, щоб система могла легко інтегруватися та була надійною, необхідно було підібрати конкретні технології.

Оскільки офіційний сайт кафедри працює на WordPress, розробку було введено туди.

Основні інструменти:

- PHP. Мова програмування, на якій працює сам WordPress, ідеально підійшла для читання файлу CSV;
- HTML та CSS. Мови розмітки та стилей, за допомогою якого створено сам каркас таблиці із зовнішнім виглядом;
- JavaScript та jQuery. Використані для забезпечення інтерактивності;
- Плагін Code Snippets. Спеціальний інструмент, який дозволяє безпечно додавати власний PHP код на сайт без ризику зламати тему оформлення.

Для інтерактивного розкладу ж застосовано JavaScript, CSS та HTML.

Модуль розкладу реалізовано з використанням сітки на основі HTML блоків.

Джерелом даних є зовнішній json файл у репозитарії GitHub, тому розклад можна оновлювати без втручання в код.

4.2 Доречність плагіну Code Snippets

Через цей плагін, код не записується в системні файли сайту, які перезаписуються при кожному оновленні. Замість цього плагін створює окрему таблицю в базі даних, в який WordPress буде

підглядати і виконувати код так, ніби він частина системи, коли сторінка завантажується.

З цим плагіном, якщо в коді є помилка, код просто не запуститься, а сайт буде стабільно та без проблем працювати.

Також код, завдяки плагіну, не прив'язаний до теми сайту. Тобто, якщо раптом буде організовано зміну дизайну сайту, то цей репозитарій нікуди не зникне, так як він живе в плагіні, а не в файлах оформлення.

Також код можна виключити одним кліком, не видаляючи його. Також цей плагін містить в собі десятки різних дрібних покращень.

4.3 Технічні моменти

Процес розробки репозитарію розпочався зі створення спеціального шорткоду.

Шорткод – це маленька команда у WordPress, яку можна вставити на будь-яку сторінку, і для користувача замість неї з'явиться результат роботи скрипта.

Перша частина:

скрипт отримує посилання на завантажений файл з медіатеки сайту. Використовуючи функцію `foren`, скрипт відкриває файл у режимі читання, а функція `fgetcsv` розбиває кожен рядок на окремі змінні. Особлива увага була приділена реалізації функції обмеження доступу: скрипт аналізує колонку з темою роботи, і якщо знаходить там слово «Відрахований», або якщо взагалі посилання відсутнє, він замість активної кнопки «Відкрити» виводить «Немає доступу».

Друга частина:

Після того, як таблиця згенерована, в роботу включається JavaScript код. За допомогою функції `initComplete` бібліотека `DataTables` сканує заповнену таблицю і знаходить усі унікальні роки захисту, освітні ступені, спеціальності та імена наукових керівників. З цих даних скрипт автоматично створює зручні випадаючі списки над таблицею.

Також було запрограмовано загальне поле пошуку, яке реагує на введення тексту з клавіатури і відфільтровує таблицю.

Додатково була реалізована кнопка «Скинути», яка очищає всі фільтри в один клік.

4.4 Розгортання

Процес розгортання як репозитарію кваліфікаційних робіт, так і розкладу відбувався безпосередньо на сайті кафедри Інформаційних технологій та моделювання. Розгортання було виконано в три кроки, оскільки працює він на базі WordPress:

- Підготовка та завантаження бази даних.
Список робіт було оформлено у вигляді таблиці та збережено у форматі CSV. Цей файл був завантажений в медіатеку адмін-панелі сайту.
Це дозволило отримати пряме посилання на дані, з якої програма читає інформацію.

ПІБ студента	Рік	Освітній ступінь	Форма навчання	Спеціальність	Тема кваліфікаційн	Керівник	Кафедра	Переглянути
Довгаль Владислав	2023/2024	Бакалавр	Денна	122 Комп'ютерні на	Інформаційна сист	доц. Крайчук Сергій	Інформаційних тех	http://repository.rshu
Китайло Денис Ан	2023/2024	Бакалавр	Денна	122 Комп'ютерні на	Моделювання інфо	доц. Сінчук Алеся М	Інформаційних тех	http://repository.rshu
Король Максим Ми	2023/2024	Бакалавр	Денна	122 Комп'ютерні на	Інформаційна сист	доц. Шевцова Ната	Інформаційних тех	http://repository.rshu
Косарєв Дмитро Се	2023/2024	Бакалавр	Денна	122 Комп'ютерні на	Веб-додаток для пл	доц. Бабич Степані	Інформаційних тех	http://repository.rshu
Лаврушкін Сергій Ві	2023/2024	Бакалавр	Денна	122 Комп'ютерні на	Забезпечення безпе	доц. Шинкарук Наз	Інформаційних тех	http://repository.rshu
Назарчук Богдан Гр	2023/2024	Бакалавр	Денна	122 Комп'ютерні на	Створення та оптим	доц. Мороз Ігор П'єт	Інформаційних тех	http://repository.rshu
Семенюк Руслан Ве	2023/2024	Бакалавр	Денна	122 Комп'ютерні на	Лендінг на основі ін	доц. Шевцова Ната	Інформаційних тех	http://repository.rshu
Токар Роман Іванов	2023/2024	Бакалавр	Денна	122 Комп'ютерні на	Електронний журна	доц. Сінчук Алеся М	Інформаційних тех	http://repository.rshu
Добряля Михайло Г	2023/2024	Бакалавр	Денна	121 Інженерія прог	Мобільний додаток Відрахований	ст. викл. Ляшук Тар	Інформаційних технологій та моделюва	
Доронін Віталій Ол	2023/2024	Бакалавр	Денна	121 Інженерія прог	Чат-бот для автомо	доц. Гаврилюк Волс	Інформаційних тех	http://repository.rshu
Ільюк Ілля Ігорович	2023/2024	Бакалавр	Денна	121 Інженерія прог	Комп'ютерна гра н	доц. Петренко Серг	Інформаційних технологій та моделюва	
Король Назар Ігор	2023/2024	Бакалавр	Денна	121 Інженерія прог	Система менеджме	доц. Петренко Серг	Інформаційних тех	http://repository.rshu
Маткобожик Олег М	2023/2024	Бакалавр	Денна	121 Інженерія прог	Моделювання інфо відрахований	доц. Сінчук Алеся М	Інформаційних технологій та моделюва	
Соколькова Анастас	2023/2024	Бакалавр	Денна	121 Інженерія прог	Економічний аналіз	доц. Паламарчук Ол	Інформаційних тех	http://repository.rshu
Тимошук Олександр	2023/2024	Бакалавр	Денна	121 Інженерія прог	Порівняльний аналі	доц. Батишкіна Юлі	Інформаційних тех	http://repository.rshu
Троневиц Віталій С	2023/2024	Бакалавр	Денна	121 Інженерія прог	Конструктор-інтерп Відрахований	доц. Сяський Волод	Інформаційних технологій та моделюва	
Шеремет Олена П'є	2023/2024	Бакалавр	Денна	121 Інженерія прог	Організація захисту	доц. Кіндрат Павло	Інформаційних тех	http://repository.rshu
Якимчук Ілля Юрій	2023/2024	Бакалавр	Денна	121 Інженерія прог	Сучасні технології	ст. викл. Кирик Тет	Інформаційних технологій та моделюва	
Гуменюк Тарас Ігор	2023/2024	Бакалавр	Денна	113 Прикладна мат	Вирішення проблем	доц. Мороз Ігор П'єт	Інформаційних тех	http://repository.rshu
Яковчук Володимир	2025	Бакалавр	Денна	122 Комп'ютерні на	Інформаційний web	доц. Петренко Сергій Вікторович	Інформаційних тех	http://repository.rshu
Лазарчук Максим П'є	2025	Бакалавр	Денна	122 Комп'ютерні на	Е-журнал академічн	доц. Петренко Серг	Інформаційних тех	http://repository.rshu
БанацькийШумансь	2025	Бакалавр	Денна	122 Комп'ютерні на	Адаптація алгоритм нечітко визначених	доц. Сяський Волод	Інформаційних тех	http://repository.rshu

Рис.4.1. База даних

- Написання коду.

Увесь код був написаний у плагіні Code Snippets.

На цьому етапі код був синхронізований з завантаженням файлом CSV та було з нього отримано спеціальну команду шорткод. Дані в середині файлу можна редагувати, просто скачавши файл, відкривши його в Google Таблицях,

завантаживши відреагований файл у форматі CSV, і додавши в медіатеку, замінити посилання старого файлу в коді на нове.

```
// Основна функція репозитарію
function build_custom_diploma_repo() {
```

```

        // Підключаємо зовнішні бібліотеки. Підключаємо стилі
бібліотеки DataTables для таблиці
        wp_enqueue_style('datatables-css',
        'https://cdn.datatables.net/1.13.6/css/jquery.dataTables.min.css');
        // Підключення бібліотеки
        wp_enqueue_script('datatables-js',
        'https://cdn.datatables.net/1.13.6/js/jquery.dataTables.min.js',
        array('jquery'), null, true);

        // Джерело даних, шлях до файлу
        $csv_url = 'https://kitm.rshu.edu.ua/wp-
content/uploads/2026/05/Репозитарій-Репозитарій-1.csv';
        // Буферизація, щоб було повернено код HTML через шорткод
        ob_start();
        ?>

```

Стилізація та інтерфейс. Для сучасного інтерфейсу було створено блок стилів. Також додано Flexbox, що дозволяє керування елементам нормально відображатися на різних пристроях.

```

<style>
    /* Оформлення панелі з фільтрами*/
    .repo-controls {
        background-color: #f8f9fa; border: 1px solid #e9ecef;
border-radius: 8px;
        padding: 20px; margin-bottom: 20px; display: flex; flex-
wrap: wrap; gap: 15px; font-family: sans-serif;
    }
    /* Стили для окремих фільтрів*/
    .filter-group { flex: 1 1 200px; display: flex; flex-
direction: column; }
    .filter-group label { font-size: 11px; text-transform:
uppercase; color: #6c757d; font-weight: 600; margin-bottom: 5px;
letter-spacing: 0.5px; }
    .filter-group input, .filter-group select {
        width: 100%; padding: 8px 12px; border: 1px solid
#ced4da; border-radius: 4px; font-size: 14px; color: #495057;
background-color: #fff; box-sizing: border-box;

    }
    .filter-group input:focus, .filter-group select:focus
{ outline: none; border-color: #80bdff; box-shadow: 0 0 0 0.2rem
rgba(0,123,255,.25); }
    .btn-reset {

```

```

        align-self: flex-end; padding: 8px 15px; background-color:
#e2e6ea; border: 1px solid #dae0e5;
        border-radius: 4px; color: #333; cursor: pointer; font-size:
14px; font-weight: 500; height: 37px;
    }
    /* Кнопка скидання*/

    .btn-reset:hover { background-color: #d3d9df; }

/* Оформлення таблиці*/
    #myRepoTable { width: 100%; border-collapse: collapse; font-
family: sans-serif; font-size: 14px; }
    #myRepoTable thead th { background-color: #fff; color: #333;
padding: 15px 10px; border-bottom: 2px solid #dee2e6; text-align:
left; font-weight: 600; }
    #myRepoTable tbody td { padding: 12px 10px; border-bottom:
1px solid #f1f3f5; vertical-align: middle; }
    #myRepoTable tr:hover { background-color: #f8f9fa; }
    /* Кнопка Відкрити та статус без доступу*/

    .btn-open { display: inline-block; padding: 6px 12px;
background-color: #0056b3; color: #fff !important; text-decoration:
none; border-radius: 4px; font-size: 13px; font-weight: 500;
transition: 0.2s; }
    .btn-open:hover { background-color: #004494; text-
decoration: none; }
    .status-badge { background-color: #e9ecef; color: #6c757d;
padding: 4px 8px; border-radius: 4px; font-size: 12px; font-weight:
500; }

/* Приховуємо загальний пошук*/
    .dataTables_filter { display: none; }
</style>

<div class="repo-controls">
    <div class="filter-group" style="flex: 2 1 300px;">
        <label>Пошук за темою або ПІБ</label>
        <input type="text" id="global-search" placeholder="">
    </div>
    <div class="filter-group">
        <label>Пік</label>
        <select id="filter-year"><option value="">Усі
роки</option></select>
    </div>
    <div class="filter-group">
        <label>Ступінь</label>

        <select id="filter-degree"><option value="">Усі
ступені</option></select>
    </div>

```

```

<div class="filter-group">
    <label>Спеціальність</label>
    <select id="filter-spec"><option value="">Усі
спеціальності</option></select>
</div>
<div class="filter-group">
    <label>Науковий керівник</label>
    <select id="filter-supervisor"><option value="">Усі
керівники</option></select>
</div>
<button class="btn-reset" id="reset-
filters">Скинути</button>
</div>

<table id="myRepoTable" class="display">
    <thead>
        <tr>
            <th>ПІБ студента</th>
            <th>Рік</th>
            <th>Ступінь</th>
            <th>Спеціальність</th>
            <th style="width: 35%;">Тема кваліфікаційної
роботи</th>
            <th>Науковий керівник</th>
            <th>Посилання</th>
        </tr>
    </thead>
    <tbody>

```

Серверна частина коду написана на мові PHP. Цей блок потоково зчитує даних із файлу, також завданням його є фільтрація та формування рядків таблиці HTML.

```

<?php
// Обробка файлу та відкриття для читання//
if (($handle = fopen($csv_url, "r")) !== FALSE) {
    $row = 0;
    while (($data = fgetcsv($handle, 1000, ",")) !==
FALSE) {
        // Пропускаємо перший рядок
        if ($row == 0) { $row++; continue; }

        // Екранування теми роботи для безпеки//

```

```

$topic = esc_html($data[5]);
    // Перевірка на слово "Відрахований"
    $is_expelled = mb_stripos($topic,
'відрахований') != false;

        echo "<tr>";
        echo "<td>" . esc_html($data[0]) . "</td>"; //
ПІБ
        echo "<td>" . esc_html($data[1]) . "</td>"; //
Рік
        echo "<td>" . esc_html($data[2]) . "</td>"; //
Ступінь
        echo "<td>" . esc_html($data[4]) . "</td>"; //
Спеціальність
        echo "<td><strong>" . $topic . "</strong></td>";
// Тема
        echo "<td>" . esc_html($data[6]) . "</td>"; //
Керівник
    // Перевірка прав доступу//
    if ($is_expelled || empty($data[8])) {
        echo "<td><span class='status-badge'>Немає
доступу</span></td>";
    } else {

        echo "<td><a class='btn-open' href='" .
esc_url($data[8]) . "'>Відкрити</a></td>";
    }
    echo "</tr>";
    $row++;
}
fclose($handle);
}
?>
</tbody>
</table>

```

Для миттєвого пошуку без перезавантаження сторінки використано бібліотеку DataTables. Логіка для клієнтів збирає значення з колонок та формує випадуючі списки для фільтрації.

```

<script>
    jQuery(document).ready(function($) {
        // Ініціалізація DataTables//
        var table = $('#myRepoTable').DataTable({
            "language": { "url": "///cdn.datatables.net/plug-
ins/1.13.6/i18n/uk.json" },
            "pageLength": 20, // Кількість рядків на сторінці//
            "dom": '<"top">rt<"bottom">ip<"clear">',
            // Заповнюємо випадаючі списки//
            initComplete: function () {
                var api = this.api();

                // Для кожної колонки з фільтрами
                api.column(1).data().unique().sort().each(function
(d) { if(d) $('#filter-year').append('<option value="' + d + '"'>' +
d + '</option>'); });
                api.column(2).data().unique().sort().each(function
(d) { if(d) $('#filter-degree').append('<option value="' + d + '"'>'
+ d + '</option>'); });
                api.column(3).data().unique().sort().each(function
(d) { if(d) $('#filter-spec').append('<option value="' + d + '"'>' +
d + '</option>'); });
                api.column(5).data().unique().sort().each(function
(d) { if(d) $('#filter-supervisor').append('<option value="' + d +
'">' + d + '</option>'); });
                }
            });

            // Обробка глобального пошуку//
            $('#global-search').on('keyup', function()
{ table.search(this.value).draw(); });
            // Налаштування фільтрації//
            $('#filter-year').on('change', function()
{ table.column(1).search(this.value ? '^' +
$.fn.dataTable.util.escapeRegex(this.value) + '$' : '', true,
false).draw(); });
            $('#filter-degree').on('change', function()
{ table.column(2).search(this.value ? '^' +
$.fn.dataTable.util.escapeRegex(this.value) + '$' : '', true,
false).draw(); });
            $('#filter-spec').on('change', function()
{ table.column(3).search(this.value ? '^' +
$.fn.dataTable.util.escapeRegex(this.value) + '$' : '', true,
false).draw(); });
            $('#filter-supervisor').on('change', function()
{ table.column(5).search(this.value ? '^' +
$.fn.dataTable.util.escapeRegex(this.value) + '$' : '', true,
false).draw(); });
            // Скинути //
            $('#reset-filters').on('click', function() {
                $('#global-search, #filter-year, #filter-degree,
#filter-spec, #filter-supervisor').val('');
                table.search('').columns().search('').draw();
            });
        });
    });

```

```

});
</script>

<?php
    // Повертаємо контент шорткоду//
    return ob_get_clean();
}
    // Реєструємо шорткод//
add_shortcode('diploma_repo', 'build_custom_diploma_repo');

```

- Створення сторінки. В адмін-панелі було створено окрему сторінку, додано тільки заголовок та тей самий шорткод, який запустив результат роботи коду на цій ж сторінці.

Репозиторій кваліфікаційних робіт

[diploma_repo]

Рис.4.2. Шорткод та заголовок

Щодо розкладу занять: увесь код поділений на два блоки для зручності. Один блок – чистий HTML та CSS, інший блок – JavaScript.

В цьому блоці ми маємо «Каркас» HTML з випадаючими списками та стилі:

```

<div id="uni-schedule-wrapper" style="font-family: sans-serif; max-width: 800px; margin: 20px 0; border: 1px solid #ddd; padding: 20px; border-radius: 8px; background: #fff;">

<div style="text-align: center; margin-bottom: 20px;">

<h3 style="margin-top:0;">Розклад занять</h3>

```

```

<div style="display: flex; flex-wrap: wrap; justify-content: center;
background: #f0f0f1; padding: 5px; border-radius: 5px; gap: 5px;">

<label style="padding: 10px 15px; cursor: pointer; border-radius: 4px;
background: #fff; box-shadow: 0 1px 2px rgba(0,0,0,0.1); flex: 1; min-
width: 200px; text-align: center;" id="lbl-student">

<input type="radio" name="sch_mode" value="student" checked
onchange="toggleMode()" style="display:none;"> Пошук розкладу по групах

</label>

<label style="padding: 10px 15px; cursor: pointer; border-radius: 4px;
color: #555; flex: 1; min-width: 200px; text-align: center;" id="lbl-
teacher">

<input type="radio" name="sch_mode" value="teacher"
onchange="toggleMode()" style="display:none;"> Пошук розкладу викладачів

</label>

</div>
</div>

<div id="panel-student">

<label style="display:block; margin-bottom:5px; font-weight:bold;">Оберіть
групу:</label>

<div style="display: flex; gap: 10px;">

<select id="uni-student-group" style="flex: 1; padding: 10px; border: 1px
solid #ccc; background-color: white;">

<option value="">Завантаження...</option>

</select>

<button onclick="searchByGroup()" style="background: #0073aa; color:
white; border: none; padding: 10px 20px; cursor: pointer; border-radius:
4px; font-weight: bold;">

Знайти

</button>

</div>
</div>

<div id="panel-teacher" style="display: none;">

<label style="display:block; margin-bottom:5px; font-weight:bold;">Оберіть
викладача:</label>

<div style="display: flex; gap: 10px;">

<select id="uni-teacher-name" style="flex: 1; padding: 10px; border: 1px
solid #ccc; background-color: white;">

<option value="">Список викладачів</option>

<option value="Шліхта">Шліхта</option>

```

```

<option value="Сяський">Сяський</option>
<option value="Бабич">Бабич</option>
<option value="Петренко">Петренко</option>
<option value="Шинкарчук">Шинкарчук</option>
<option value="Шевцова">Шевцова</option>
<option value="Ляшук">Ляшук</option>
<option value="Кирик">Кирик</option>
<option value="Турбал">Турбал</option>
<option value="Кіндрат">Кіндрат</option>
</select>

<button id="btn-scan" onclick="scanGlobalTeacher()" style="background:
#d63384; color: white; border: none; padding: 10px 20px; cursor: pointer;
border-radius: 4px; font-weight: bold;">

Знайти

</button>

</div>

<div id="scan-progress" style="display:none; margin-top: 15px;">

<div style="font-size: 12px; margin-bottom: 5px;">Пошук по групах: <span
id="scan-text">0%</span></div>

<div style="width: 100%; background: #eee; height: 8px; border-radius:
4px; overflow: hidden;">

<div id="scan-bar" style="width: 0%; height: 100%; background: #d63384;
transition: width 0.2s;"></div>

</div>

</div>

</div>

<div id="uni-error" style="margin-top: 15px; color: red; font-weight:
bold; display: none;"></div>

<div id="uni-results" style="margin-top: 20px;"></div>

</div>

```

А в цьому блоці розташований сам «Мозок» інтерактивного застосунку зі всією логікою:

```

<script>
document.addEventListener("DOMContentLoaded", function() {
    var sg = document.getElementById('uni-student-group');
    fetch('https://raw.githubusercontent.com/eRSHU-
org/schedule/main/fmi/schedule/all_years.json')
        .then(function(r){ return r.json(); })
        .then(function(d){
            sg.innerHTML = '<option value="">Оберіть групу</option>';
            var a = [];
            if (d.all_years && Array.isArray(d.all_years)) {
                d.all_years.forEach(function(year) {
                    if (Array.isArray(year)) {
                        year.forEach(function(group) {
                            if (group && group.specialty_name) {
                                var name = group.specialty_name.toUpperCase();
                                if (name.includes('ІПЗ') ||
name.includes('KH') || name.includes('ПМ') || name.includes('М-')) {
                                    a.push(group.specialty_name);
                                }
                            }
                        });
                    }
                });
            }
            var uniqueGroups = Array.from(new Set(a)).sort();
            uniqueGroups.forEach(function(n) {
                var o = document.createElement('option');
                o.value = n; o.innerText = n;
                sg.appendChild(o);
            });
        })
        .catch(function(){ sg.innerHTML = '<option value="">Помилка
завантаження</option>'; });
});

```

```

function toggleMode() {
    var mode =
document.querySelector('input[name="sch_mode"]:checked').value;

    document.getElementById('panel-student').style.display = (mode ===
'student') ? 'block' : 'none';

    document.getElementById('panel-teacher').style.display = (mode ===
'teacher') ? 'block' : 'none';

    var ls = document.getElementById('lbl-student'), lt =
document.getElementById('lbl-teacher');

    if(mode === 'student') {

        ls.style.background = '#fff'; ls.style.boxShadow = '0 1px 2px
rgba(0,0,0,0.1)'; ls.style.color = '#000';

        lt.style.background = 'transparent'; lt.style.boxShadow = 'none';
lt.style.color = '#555';

    } else {

        lt.style.background = '#fff'; lt.style.boxShadow = '0 1px 2px
rgba(0,0,0,0.1)'; lt.style.color = '#000';

        ls.style.background = 'transparent'; ls.style.boxShadow = 'none';
ls.style.color = '#555';

    }

    document.getElementById('uni-results').innerHTML = '';
    document.getElementById('uni-error').style.display = 'none';
}

function makeLinks(text) {
    var urlRegex = /(https?:\/\/\[^\s]+\)/g;
    return text.replace(urlRegex, function(url) {
        return ' <a href="' + url + '" target="_blank" style="color:green;
font-weight:bold; text-decoration:none;">[Bxiд]</a>';
    });
}

function searchByGroup() {
    var grp = document.getElementById('uni-student-group').value;

    var res = document.getElementById('uni-results'), err =
document.getElementById('uni-error');

    res.innerHTML = '<div style="color:blue;">Завантаження...</div>';
err.style.display = 'none';
}

```

```

if(!grp) {
    err.innerText = 'Оберіть групу!';
    err.style.display = 'block';
    res.innerHTML='';
    return;
}

var crsMatch = grp.match(/\d/);
if(!crsMatch) {
    err.innerText = 'Помилка курсу';
    err.style.display = 'block';
    res.innerHTML='';
    return;
}

var crs = parseInt(crsMatch[0]);
var fn = grp.toUpperCase();

if (fn.startsWith('M-')) {
    crs = crs + 4; // 1 курс магістрів стає 5-м, 2 курс – 6-м
}

if(!fn.endsWith('.json')) fn += '.json';

fetch('https://raw.githubusercontent.com/eRSHU-
org/schedule/main/fmi/schedule/' + crs + '/' + encodeURIComponent(fn))
    .then(function(r){ if(!r.ok) throw new Error(); return r.json(); })
    .then(function(d){ renderTable(d.week); })
    .catch(function(){ res.innerHTML = ''; err.innerText = 'Розклад не
знайдено'; err.style.display = 'block'; });
}

async function scanGlobalTeacher() {
    var tnm = document.getElementById('uni-teacher-
name').value.toLowerCase();

```

```

    var res = document.getElementById('uni-results'), err =
document.getElementById('uni-error');

    var bar = document.getElementById('scan-bar'), txt =
document.getElementById('scan-text'), prg = document.getElementById('scan-
progress'), btn = document.getElementById('btn-scan');

    if(!tnm) { err.innerText = 'Оберіть викладача!'; err.style.display =
'block'; return; }

    btn.disabled = true; prg.style.display = 'block';

    try {

        var r = await fetch('https://raw.githubusercontent.com/eRSHU-
org/schedule/main/fmi/schedule/all_years.json');

        var list = await r.json(), urls = [];

        list.all_years.forEach(function(yg, i){

            if(Array.isArray(yg)) yg.forEach(function(g){

                if(g && g.specialty_name &&
(g.specialty_name.toUpperCase().includes('ІПЗ') ||
g.specialty_name.toUpperCase().includes('КН') ||
g.specialty_name.toUpperCase().includes('ПМ') ||
g.specialty_name.toUpperCase().includes('М')) {

                    urls.push({ u:
'https://raw.githubusercontent.com/eRSHU-org/schedule/main/fmi/schedule/'
+ (i+1) + '/' + encodeURIComponent(g.specialty_name.toUpperCase()) +
'.json'), n: g.specialty_name });

                }

            });

        });

        var done = 0, allFound = [];

        for(let target of urls) {

            try {

                let resp = await fetch(target.u);

                if(resp.ok) {

                    let d = await resp.json();

                    if(d.week) d.week.forEach(function(day, di){

                        if(day) day.forEach(function(l){

                            if(l && l.lesson_name &&
l.lesson_name.toLowerCase().includes(tnm))

                                allFound.push({ d: di, num: l.lesson_number,
g: target.n, name: l.lesson_name, lnk: l.link });

                        });

                    });

                }

            }

        }

    }

```

```

        });
    }

    } catch(e){}

    done++; var p = Math.round((done/urls.length)*100);
    bar.style.width = p + '%'; txt.innerText = p + '%';
}

prg.style.display = 'none'; btn.disabled = false;

if(allFound.length === 0) res.innerHTML = '<p>Пар не
знайдено.</p>'; else renderGlobalResults(allFound);

} catch(e) { prg.style.display = 'none'; btn.disabled = false;
err.innerText = 'Помилка'; err.style.display = 'block'; }
}

function renderTable(week) {

    var res = document.getElementById('uni-results'), days = ["Понеділок",
    "Вівторок", "Середа", "Четвер", "П'ятниця", "Субота", "Неділя"], h = '';

    week.forEach(function(ls, i){

        if(!ls) return;

        var real = ls.filter(function(x){ return x && x.lesson_name });

        if(real.length === 0) return;

        h += '<div style="margin-bottom:20px; border:1px solid #eee;
padding:10px; border-radius:5px;"><div style="font-weight:bold;
color:#0073aa; border-bottom:2px solid #0073aa; margin-bottom:10px;">' +
days[i] + '</div>';

        real.forEach(function(l){

            var content = makeLinks(l.lesson_name);

            h += '<div style="display:flex; padding:5px; border-bottom:1px
solid #f9f9f9;"><div style="width:30px; font-weight:bold;">' +
l.lesson_number + '</div><div style="flex:1;">' + content + (l.link ? ' <a
href="'+l.link+'" target="_blank" style="color:green; font-weight:bold;
text-decoration:none; margin-left:10px;">[Вхід]</a>' : '') +
'</div></div>';

        });

        h += '</div>';

    });

    res.innerHTML = h || 'Пар не знайдено.';

}

function renderGlobalResults(lessons) {

```

```

    var res = document.getElementById('uni-results'), days = ["Понеділок",
    "Вівторок", "Середа", "Четвер", "П'ятниця", "Субота", "Неділя"], h = '',
    curD = -1;

    lessons.sort(function(a,b){ return (a.d - b.d) || (a.num -
    b.num) }).forEach(function(l){

        if(l.d !== curD) {

            if(curD !== -1) h += '</div>';

            h += '<div style="margin-bottom:20px; border:1px solid
            #d63384; border-radius:5px; overflow:hidden;"><div
            style="background:#d63384; color:white; padding:10px; font-weight:bold;">'
            + days[l.d] + '</div>';

            curD = l.d;

        }

        var content = makeLinks(l.name);

        h += '<div style="padding:10px; border-bottom:1px solid #eee;
        background:#fff;"><div style="font-weight:bold; color:#0073aa;">' + l.num
        + ' napa | ' + l.g + '</div><div style="margin-top:4px;">' + content +
        '</div>' + (l.lnk ? '<a href="'+l.lnk+'" target="_blank"
        style="display:inline-block; margin-top:8px; color:white;
        background:#28a745; padding:4px 10px; border-radius:3px; text-
        decoration:none; font-size:12px;">Вхід<a>' : '')

        + '</div>';

    });

    res.innerHTML = h + (curD !== -1 ? '</div>' : '');
}
</script>

```

4.5 Результати

Після завершення роботи з кодом, система пройшла етап тестування на поточних даних.

Тестування включало:

Перевірку швидкості роботи:

Незважаючи на великий обсяг тексту в темах робіт, таблиця завантажується в браузері миттєво.

Перевірку адаптивності:

Таблиця коректно відображається на різних пристроях.

Перевірку безпеки:

Спроби відкрити роботи відрахованих студентів блокувалися на рівні генерації HTML-коду.

Тестування інтеграції:

Окремим етапом впровадження стала інтеграція розробленої сторінки в меню сайту кафедри.

Використовуючи блоковий редактор сайту WordPress, нову сторінку репозитарію було успішно додано як дочірній елемент у розділі головного меню.

Щодо інтерактивного розкладу:

Під час тестування було виявлено помилку, пов'язаних з завантаженням файлів. Скрипт намагався побудувати таблицю раніше завантаження

даних. Проблему було вирішено шляхом впровадження оператора `await fetch()`, що змушує систему дочекатися спочатку відповіді сервера.

Результат:

Розроблені програмні модулі повністю відповідають поставленим на початку роботи вимогам.

Кафедра отримала сучасний, швидкий та зручний інструмент для каталогізації студентських робіт, який не потребує витрат на обслуговування та підтримується зусиллями звичайних операторів без знання мов програмування.

ВИСНОВКИ

Під час виконання цієї кваліфікаційної роботи було розроблено та впроваджено інформаційну систему «Репозиторій кваліфікаційних робіт» для кафедри інформаційних технологій та моделювання Рівненського державного гуманітарного університету.

Головним результатом роботи став програмний модуль, який вирішує критичну проблему кафедри, а саме відсутність єдиного, зручного та публічного каталогу студентських робіт.

До моменту розробки всі дані зберігалися розрізнено, що ускладнювало пошук інформації та підготовку звітів. Тепер ця проблема вирішена.

У ході розробки було прийнято кілька важливих технічних рішень, які зробили систему ефективною та простою.

По-перше, ми відмовилися від використання великих і складних баз даних на користь легких CSV файлів. Це дозволяє персоналу кафедри керувати всім каталогом через звичайний Excel.

Такий підхід робить систему автономною: для її підтримки не потрібно залучати професійних програмістів, адже оновити таблицю може будь-який впевнений користувач комп'ютера.

По-друге, використання мови PHP разом із бібліотекою DataTables дозволило створити інтерфейс, який працює дуже швидко. Замість того, щоб чекати завантаження кожної нової сторінки при пошуку, користувач бачить результати миттєво. Весь процес фільтрації за роком, спеціальністю чи науковим керівником відбувається безпосередньо в браузері, що значно підвищує комфорт роботи з сайтом.

Важливим досягненням стала реалізація доступу в один клік. Було створено систему, де користувач бачить усі необхідні дані одразу в таблиці та може миттєво відкрити PDF файл в тій самій вкладці. Це робить репозиторій максимально практичним інструментом для студентів, які шукають матеріали для навчання, та для абітурієнтів, які хочуть ознайомитися з рівнем підготовки на кафедрі.

Окрему увагу було приділено питанням безпеки. У коді реалізована логіка, яка автоматично розпізнає статус студента. Якщо робота не підлягає публікації або студент був відрахований, система не отримує пряме посилання і замінює його на попередження про відсутність даних.

Завершальним етапом стала повна інтеграція репозитарію в офіційну структуру сайту кафедри. Сторінку було додано в головне меню, що зробило її легкодоступною для аудиторії. Проведене тестування показало, що система стабільно працює на різних пристроях, включаючи смартфони та планшети, що є критично важливим для сучасної молоді.

У підсумку можна стверджувати, що створений репозиторій є не просто навчальним проєктом, а реальним робочим інструментом. Він підвищує престиж кафедри, забезпечує прозорість освітнього процесу та сприяє дотриманню принципів академічної доброчесності. У майбутньому систему можна легко розширювати, додаючи нові поля або інтегруючи її з іншими сервісами університету, що підтверджує її гнучкість та перспективність.

Також було реалізовано інтерактивний веб інтерфейс для візуалізації розкладу занять, що дозволило вирішити проблему використання статичних файлів.

Ключовою особливістю модуля стала організація взаємодії з віддаленим сховищем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. CSS-TRICKS. CSS Flexbox Layout Guide. URL: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/> (дата звернення: 10.05.2026).
2. w3schools. CSS Flexbox. URL: https://www.w3schools.com/css/css3_flexbox.asp (дата звернення: 10.05.2026).
3. DOU.ua. CSS Flexbox layout. URL: <https://dou.ua/lenta/articles/css-flexbox-layout/> (дата звернення: 10.05.2026).
4. developer.mozilla.org. Flexbox. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/CSS_layout/Flexbox (дата звернення: 10.05.2026).
5. High Bar Journal. Повне керівництво із Flexbox у CSS . URL: <https://journal.gen.tech/post/css-flexbox-korysne-kerivnytstvo> (дата звернення: 10.05.2026).
6. jQuery. jQuery. URL: <https://jquery.com/> (дата звернення: 10.05.2026).
7. w3schools.com. jQuery Tutorial. URL: <https://www.w3schools.com/jquery/default.asp> (дата звернення: 10.05.2026).

8. w3schoolsua. jQuery Підручник. URL:
<https://w3schoolsua.github.io/jquery/index.html#gsc.tab=0> (дата звернення: 10.05.2026).
9. w3schools.com. jQuery Introduction. URL:
https://www.w3schools.com/jquery/jquery_intro.asp (дата звернення: 10.05.2026).
10. NPM. jquery. URL: <https://www.npmjs.com/package/jquery>
(дата звернення: 10.05.2026).
11. DataTables. DataTables | Javascript table library. URL:
<https://datatables.net/> (дата звернення: 10.05.2026).
12. University of St Andrews. Using jQuery DataTables. URL:
<https://www.st-andrews.ac.uk/dpl/0.5.5/datatables.html> (дата звернення: 10.05.2026).
13. Google Charts. DataTables and DataViews. URL:
https://developers.google.com/chart/interactive/docs/datatables_dataviews (дата звернення: 10.05.2026).
14. Drupal. DataTables. URL:
<https://www.drupal.org/project/datatables> (дата звернення: 10.05.2026).
15. CX.UA. Як працювати з datatables. URL:
<https://ncarlau.ferge.cx.ua/articles/jak-pracjuvati-z-datatables.html>
(дата звернення: 10.05.2026).

16. CSS.in.ua. Довідник по HTML тегам. URL:
<https://css.in.ua/html/tags> (дата звернення: 10.05.2026).

17. w3schoolsua. HTML Основні приклади. URL:
https://w3schoolsua.github.io/html/html_basic.html#gsc.tab=0
(дата звернення: 10.05.2026).

18. MDN Web Docs. HTML. URL:
<https://developer.mozilla.org/ru/docs/Web/HTML> (дата
звернення: 10.05.2026).

19. html-css.co.ua. HTML і CSS довідник українською. URL:
<https://html-css.co.ua/> (дата звернення: 10.05.2026).

20. w3schoolsua. HTML Підручник. URL:
<https://w3schoolsua.github.io/html/index.html#gsc.tab=0> (дата
звернення: 10.05.2026).

21. uk.javascript.info. Сучасний підручник з JavaScript. URL:
<https://uk.javascript.info/> (дата звернення: 10.05.2026).

22. w3schoolsua. JavaScript Підручник. Основи веб
програмування. URL:
<https://w3schoolsua.github.io/js/index.html#gsc.tab=0> (дата
звернення: 10.05.2026).

23. MDN Web Docs. JavaScript. URL:

<https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата звернення: 10.05.2026).

24. EPAM Ukraine. JavaScript: матеріали для самопідготовки. URL:

<https://campus.epam.ua/ua/blog/255> (дата звернення: 10.05.2026).

25. WebDoky. Посібник з JavaScript. URL:

<https://webdoky.org/uk/docs/Web/JavaScript/Guide/> (дата звернення: 10.05.2026).

26. CSS.in.ua. Український веб-довідник. URL: <https://css.in.ua/>

(дата звернення: 10.05.2026).

27. w3schoolsua. CSS підручник. URL:

<https://w3schoolsua.github.io/css/index.html#gsc.tab=0> (дата звернення: 10.05.2026).

28. bagri.whitl.cx.ua. Як створити таблиці в рнр. URL:

<https://bagri.whitl.cx.ua/articles/jak-stvoriti-tablici-v-rnr.html> (дата звернення: 10.05.2026).

29. kovelpost. Взаємодія PHP з HTML. URL:

<https://kovelpost.com/blogs/9752> (дата звернення: 10.05.2026).

30. php.net. Порівняльна таблиця типів PHP. URL: <https://www.php.net/manual/uk/types.comparisons.php> (дата звернення: 10.05.2026).

31. brj.cz. Тернарні оператори в PHP (?:) – умова в одному рядку. URL: <https://uk.php.brj.cz/ternarni-operatori-v-php-umova-v-odnomu-ryadku> (дата звернення: 10.05.2026).

32. Київські учнівські олімпіади з інформаційних технологій. Javascript: структури даних. URL: https://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/65_Javascript/index.html (дата звернення: 10.05.2026).

33. PHP. PHP: Documentation. URL: <https://www.php.net/docs.php> (дата звернення: 10.05.2026).