

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна графіка і Web-дизайн»

Група: 4КГ-06

Дипломний проект

здобувачки освіти денної форми навчання
КГ.06.01.000.ДП

БАБІЧ
ВІКТОРІЇ СЕРГІЇВНИ

м. Одеса
2023 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність 123 – «Комп'ютерна інженерія»

Освітня програма «Комп'ютерна графіка і web-дизайн»

Група 4КГ-06

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Створення репозитарію методичної роботи для ВСП «ОТФК ОНТУ»

Проектний матеріал складається з пояснювальної записки на 88 сторінках та графічного (презентаційного) матеріалу на 27 аркушах (слайдах).

Дипломник В. Сергєу (Бабіч В.С.)

Керівник проекту [підпис] (Суліма Ю.Є.)

Консультанти:

з економічної частини [підпис] (Копайгородська Т.Г.)

з охорони праці [підпис] (Чорновол Н.І.)

за дотриманням вимог ЄСКД [підпис] (Петрашова В.І.)

старший консультант [підпис] (Кривченко А.А.)

До захисту допущений

Голова циклової комісії [підпис] (Кривченко Ю.В.)

Завідувач відділення [підпис] (Скорнякова О.В.)

Захист «22» червня 2023 р. Протокол ДКК № 4

Оцінка ДКК 5/відмінно

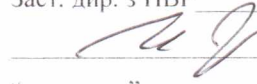
Секретар ДКК [підпис]

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна графіка і Web-дизайн»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР Беркань Іг.В.



“ ” 2023р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти Бабіч Вікторії Сергіївни
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Створення репозитарію методичної роботи для ВСП «ОТФК ОНТУ»

затверджена наказом по коледжу від “ 17 ” жовтня 202 2 р. № 235 – А2 – ОД

2. Термін здачі закінченого проекту (роботи) 12.06.2023 р.

3. Вихідні дані до проекту (роботи) Створити репозитарій методичної роботи для ВСП «ОТФК ОНТУ», згідно технічному завданню на дипломний проект.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

1. Вступ. 2. Аналіз предметної області. 3. Технології та засоби розробки. 4. Проектування дизайну веб-інтерфейсу та архітектури файлового менеджера. 5. Розробка Python додатку. 6. Розробка MVC-додатку на веб-хостинг. 7. Публікація додатку на хостинг. 8. Робота зі створеним репозитарієм. 9. Економічний розрахунок. 10. Розділ охорони праці. 11. Висновок. 12. Перелік використаних джерел.

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Презентація – 15 слайдів (Структура репозитарію; Інформаційна архітектура файлового менеджера; Взаємодія програмних модулів репозитарію; Модульна структура MVC-додатку; Файлова структура MVC-додатку; Структура файлового менеджера MVC-додатку; Розробка дизайну головної сторінки; Скріншоти роботи з репозитарієм)

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Вступ, технологічний розділ, висновок	Суліма Ю.Є.		
Економічний розрахунок	Копайгородська Т.Г.		
Охорона праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Кривченко А.А.		

7. Дата видачі завдання

01.05.2023 р.

Керівник

Завдання прийняв до виконання

(підпис)
В. Сергєу
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1.	Вступ	22.05.2023 р.	виконано
2.	Аналіз предметної області	23.05.2023 р.	виконано
3.	Технології та засоби розробки	24.05.2023 р.	виконано
4.	Проектування дизайну веб-інтерфейсу та архітектури файлового менеджера	25.05.2023 р.	виконано
5.	Розробка Python додатку	26.05.2023 р.	виконано
6.	Розробка MVC-додатку на веб-хостинг	29.05.2023 р.	виконано
7.	Публікація додатку на хостинг	31.05.2023 р.	виконано
8.	Робота зі створеним репозитарієм	02.06.2023 р.	виконано
9.	Економічний розрахунок	05.06.2023 р.	виконано
8.	Розділ охорони праці	06.06.2023 р.	виконано
9.	Висновок. Перелік використаних джерел	07.06.2023 р.	виконано
10.	Оформлення пояснювальної записки	08.06.2023 р.	виконано
11.	Оформлення графічної (презентаційної) частини	09.06.2023 р.	виконано
12.	Підготовка доповіді для захисту	09.06.2023 р.	виконано
13.	Малий захист дипломного проекту	12.06.2023 р.	виконано
14.	Захист дипломного проекту	19.06.2023 р.	виконано

Дипломник

(підпис)

Керівник

(підпис)

ЗМІСТ

ВСТУП	6
1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ	9
1 Аналіз предметної області	9
1.1 Огляд існуючих рішень	13
1.2 Технології та засоби розробки	19
2 Реалізація проекту	30
2.1 Технічне завдання на розробку	30
2.2 Проектування дизайну веб-інтерфейсу та архітектури файлового менеджера	32
2.3 Реалізація додатку	37
2.3.1 Розробка Python додатку	39
2.3.2 Розробка MVC-додатку на веб-хостинг	42
2.3.3 Розробка файлового менеджера	55
2.4 Публікація додатку на хостинг	58
2.5 Робота зі створеним репозитарієм	60
2 ЕКОНОМІЧНА ЧАСТИНА	68
3 ОХОРОНА ПРАЦІ	78
3.1 Аналіз та безпека умов праці працівника на робочому місці	78
3.1.1 Організація робочого місця	78
3.1.2 Коротка характеристика і основні вимоги безпеки до мікроклімату виробничих приміщень, освітлення, шуму, вібрації, ультразвуку, інфразвуку, виробничих випромінювань, небезпека ураження електричним струмом	80

					КГ 06.01.000.00 ДП ПЗ	Арк.
						4
Ізм.	Лист	№ докум.	Підпис	Дата		

3.2 Пожежна безпека	83
3.3 Висновки за розділом	86
ВИСНОВОК	87
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	88

					КГ 06.01.000.00 ДП ПЗ	Арк.
						5
Ізм.	Лист	№ докум.	Підпис	Дата		

ВСТУП

В дипломному проєкті розглядається створення репозитарію методичної роботи для ВСП «ОТФК ОНТУ».

Репозитарій або репозиторій – це цифрове сховище, де містяться дані певного типу. Найчастіше він представляє собою каталог інформації з файлами, локальний або розташований на зовнішньому ресурсі в мережі, що розрахований на віддалений авторизований доступ до цієї інформації.

Мережеві репозитарії – великі сховища, звідки користувачі можуть завантажити потрібні дані за спеціальними протоколами. Такий спосіб завантажувати інформацію часто застосовують користувачі Linux та Unix-подібних систем.

Часто слово «репозитарій» використовують у контексті систем контролю версій, таких як Git. У цьому значенні репозитарій – локальна папка, де зберігаються версії проєкту, або глобальне сховище на віддаленому сервісі, наприклад такому як GitHub.

Віддалений репозитарій може бути: публічним – це означає, що його можуть переглядати всі користувачі, і кожен може завантажити звідти дані; приватним – доступ мають лише власник та люди, яким він дав доступ.

Приватними репозитаріями користуються компанії та приватні розробники, які не бажають оприлюднювати свої роботи.

Публічні репозитарії застосовуються для програм, фреймворків та технологій з відкритим вихідним кодом, а також для іншої вільної для розповсюдження інформації. На публічних репозитаріях також може бути передбачена система авторизації та розмежування прав доступу до інформації, наприклад – певні категорії користувачів можуть лише переглядати інформацію без можливості її завантаження на власний

					КГ 06. 01 000. 00 ДП ПЗ	Арк.
						6
Ізм.	Лист	№ докум.	Підпис	Дата		

пристрій, інші можуть переглядати тільки певні розділи репозитарію, а обрані будуть мати повний доступ до даних.

Репозитарій, який буде створено під час виконання дипломного проектування, призначений для збереження навчально-методичних розробок, які забезпечують провадження освітнього процесу за навчальними дисциплінами, що викладаються на цикловій (предметній) комісії комп'ютерних технологій та програмної інженерії, а також для постійного віддаленого доступу до цих матеріалів здобувачів освіти.

Принцип роботи репозитарію наступний: користувач заходить за певною адресою на сайт репозитарію, де проходить процедуру авторизації; після успішної авторизації в автоматичному режимі запускається Python-скрипт, що забезпечує запуск мережевого сервісу тунелювання diode та формує активне посилання на сервер, де розміщені матеріали репозитарію; після з'єднання з сервером відбувається запуск файлового менеджера для перегляду матеріалів репозитарію (файлів у форматі PDF) та роботи з ним.

Проектований репозитарій складатиметься з трьох програмних додатків:

- Python-додаток для керування роботою мережевого сервісу тунелювання diode, який забезпечуватиме доступ до матеріалів репозитарію, який розміщено на сервері, через мережу Інтернет;

- RNP-додаток на хостингу для реалізації функцій файлового менеджера;

- RNP-файловий менеджер на локальній машині для випадків, коли доступ до даних через мережу неможливий.

Розробка створюватиметься за допомогою наступного інструментарію:

					КГ 06. 01 000. 00 ДП ПЗ	Арк.
						7
Ізм.	Лист	№ докум.	Підпис	Дата		

– для розробки графічного інтерфейсу та клієнтського додатку (Front-End) використовуватимуться бібліотеки фреймворків Bootstrap 5.2 і JQuery 3.6;

– для розробки серверного коду додатку (Back-End) використовуватиметься середовище PHP 8.1;

– для супутніх із розробкою операцій (DevOps), наприклад, налаштування серверу, використовуватиметься веб-сервер Apache 2.4, мережевий сервіс тунелювання diode 1.8.1, середовище розробки (IDE) Python 3.11.

Створення репозитарію методичних розробок включатиме ряд етапів:

- проектування дизайну веб-інтерфейсу;
- проектування архітектури файлового менеджера;
- реалізація окремих додатків та їх спільної роботи;
- узгодження роботи додатків і мережевого сервісу тунелювання;
- публікація додатку на хостинг та реалізація доступу до даних за вказаною адресою.

Пояснювальна записка до дипломного проекту складається з технологічного розділу, економічної частини, розділу охорони праці, висновків та переліку використаних літературних джерел. До пояснювальної записки додаються слайди мультимедійної презентації, які будуть використані під час захисту дипломного проекту, на них розміщені: схеми взаємодії модулів додатку; модульні структури додатків; скріншоти роботи репозитарію та інше.

					КГ 06. 01 000. 00 ДП ПЗ	Арк.
						8
Ізм.	Лист	№ докум.	Підпис	Дата		

1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

1 Аналіз предметної області

Розглянемо більш детально існуючі рішення на ринку програмного забезпечення для створення репозитарію, тобто архіву електронних ресурсів, а також які дані зберігаються найчастіше за змістом та за типом.

За статистичними даними OpenDOAR (каталогу сховищ із відкритим доступом, який містить записи про репозитарії відкритого доступу та дозволяє шукати і сортувати їх за певними параметрами) найбільш популярними у 2022 році програмними платформами були такі: DSpace – 39%, EPrints – 11%, WEKO – 9%, Digital Commons – 5% та інші, доля кожного з яких становила менше 4%, а разом вони склали 36% (рис. 1).

[1]

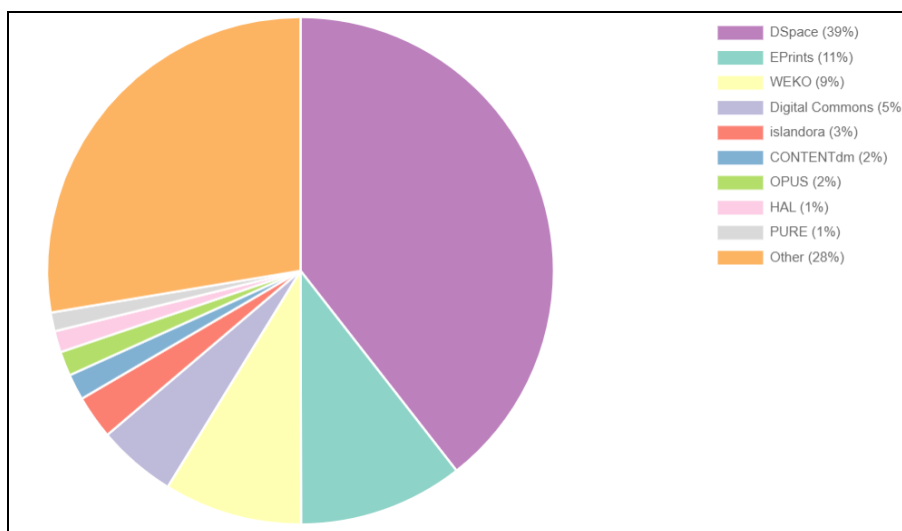


Рисунок 1 – Рейтинг репозитаріїв відкритого доступу

Щоб репозитарій був включений у список OpenDOAR, він має відповідати наступним критеріям включення:

1. На сайті повинен бути доступний контент у відкритому доступі:
 - безкоштовно – користувачу не потрібно платити за доступ до матеріалу;

					КГ 06.16.001.00 ДП ПЗ	Арк.
						9
Ізм.	Лист	№ докум.	Підпис	Дата		

– без бар'єрів доступу – користувачу не потрібно входити/реєструватися, навіть якщо це безкоштовно;

– повний обсяг інформації – користувачу надано доступ до повної, нескороченої версії вмісту, а не лише до витягу або попереднього перегляду реферату, бібліографічного запису, анотації, запису метаданих тощо.

Досить часто деякий вміст у репозитарії обмежується правами доступу/ембарго та представлений лише бібліографічним списком (метаданими). Це добре, якщо принаймні деякий вміст у сховищі не обмежено таким чином і фактично користувач має доступну повну текстову копію.

2. Сайт має бути актуальним і надійно доступним для будь-якого користувача Інтернету в усьому світі.

3. Сайт має містити елементи, які є або науковими результатами (наприклад, журнальні статті, тези, дисертації, звіти, робочі документи, матеріали конференцій, книги або розділи книги) та/або академічні ресурси з достатніми метаданими чи документацією, щоб зробити матеріал придатним для повторного використання (наприклад, архівні матеріали, набори даних, програмне забезпечення, зображення, відео, навчальний матеріал).

4. Сайт не повинен бути електронним журналом або порталом для портфоліо журналів організації.

5. Сайт не повинен бути агрегатором, який містить лише посилання на вміст відкритого доступу на зовнішніх сайтах.

6. Сайт не повинен бути бібліотечним каталогом або колекцією локально доступних електронних книг (тобто вміст відкритого доступу не доступний).

					КГ 06.16.001.00 ДП ПЗ	Арк.
						10
Ізм.	Лист	№ докум.	Підпис	Дата		

В Україні переліченим критеріям відповідають 110 репозитаріїв (рис. 2) і всі вони доступні до перегляду в каталозі OpenDOAR.

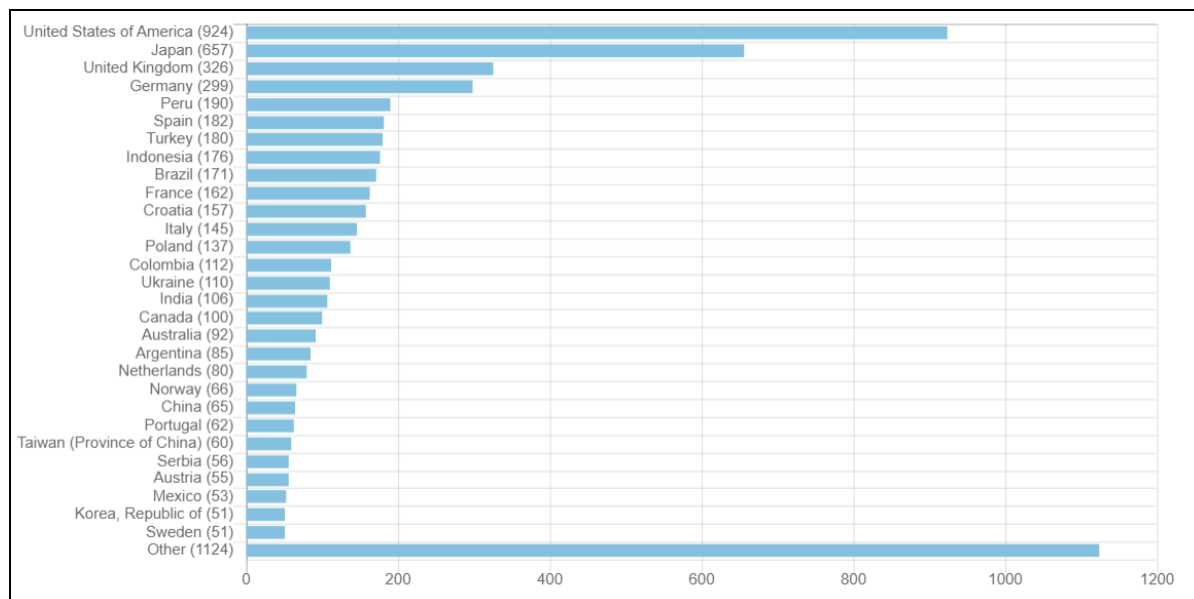


Рисунок 2 – Кількісний розподіл репозитаріїв за країнами

За типом вмісту найбільшу частку даних в репозитаріях складають наукові статті, тези та дисертації, книги, матеріали конференцій, звіти та робоча документація, навчальні матеріали, бібліографічні посилання, патенти, програмне забезпечення та інше (рис. 3).

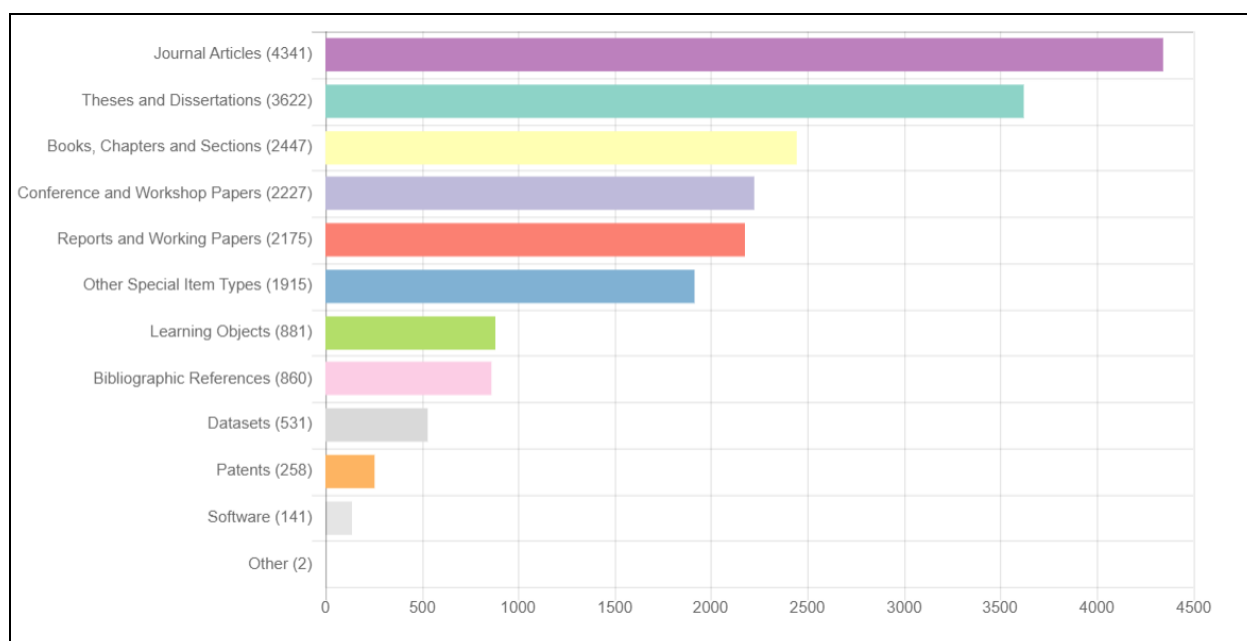


Рисунок 3 – Кількісний розподіл за типом вмісту репозитаріїв

Найбільш популярними серед вмісту репозитаріїв є галузі соціальних, гуманітарних та інших наук, технологій, здоров'я та медицини, мистецтва, інженерії, математики та ін. (рис. 4).

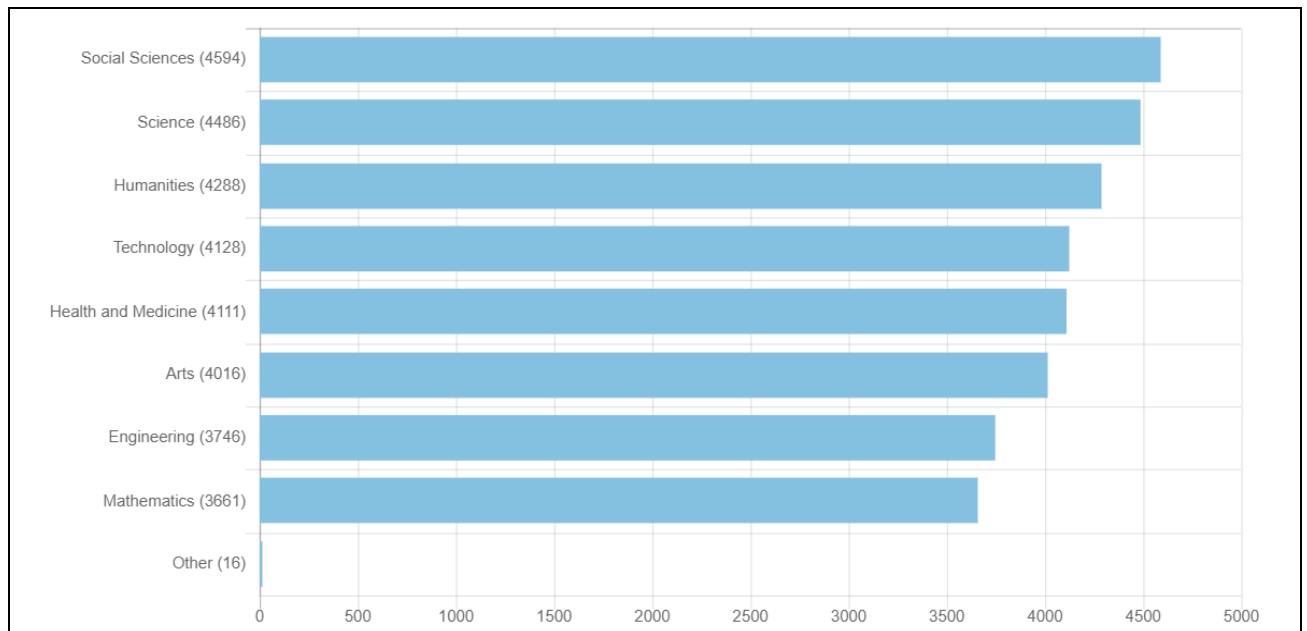


Рисунок 4 – Кількісний розподіл за галузями знань

Якщо проаналізувати всі українські репозитарії, доступні до перегляду в OpenDOAR, то можна отримати відомості стосовно найбільш популярних програмних платформ, за допомогою яких було розгорнуто цифрове сховище матеріалів саме в Україні (рис. 5).

Repository Name	Type	Program	Country
Електронний репозитарій ХНУІМЕ ім. А. Н. Бекетова	Тип інституційний	програмне забезпечення EPrints	Україна
Інституційний репозитарій Київського університету імені Бориса Грінченка	Тип інституційний	програмне забезпечення EPrints	Україна
Електронна бібліотека НАН України	Тип інституційний	програмне забезпечення EPrints	Україна
Цифровий репозитарій Острозької академії	Тип інституційний	програмне забезпечення EPrints	Україна
Цифровий репозитарій Національного університету відного господарства та інженерії природокористування	Тип інституційний	програмне забезпечення EPrints	Україна
Бібліотека МКС	Тип інституційний	програмне забезпечення EPrints	Україна
Інституційний репозитарій Мелітопольського державного педагогічного університету імені Богдана Хмельницького	Тип інституційний	програмне забезпечення EPrints	Україна
Репозитарій Дніпропетровської медичної академії	Тип інституційний	програмне забезпечення EPrints	Україна
Бібліотека Житомирського державного університету	Тип інституційний	програмне забезпечення EPrints	Україна

Рисунок 5 – Результат фільтрації українських репозитаріїв за обраним ПЗ

Виявляється, що з усіх репозитаріїв, які доступні в OpenDOAR в травні 2023 р, 98 розгорнуті за допомогою програмної платформи DSpace, 9 – за допомогою програмної платформи EPrints і лише 2 на основі самостійно створених програмних рішень (рис. 6).

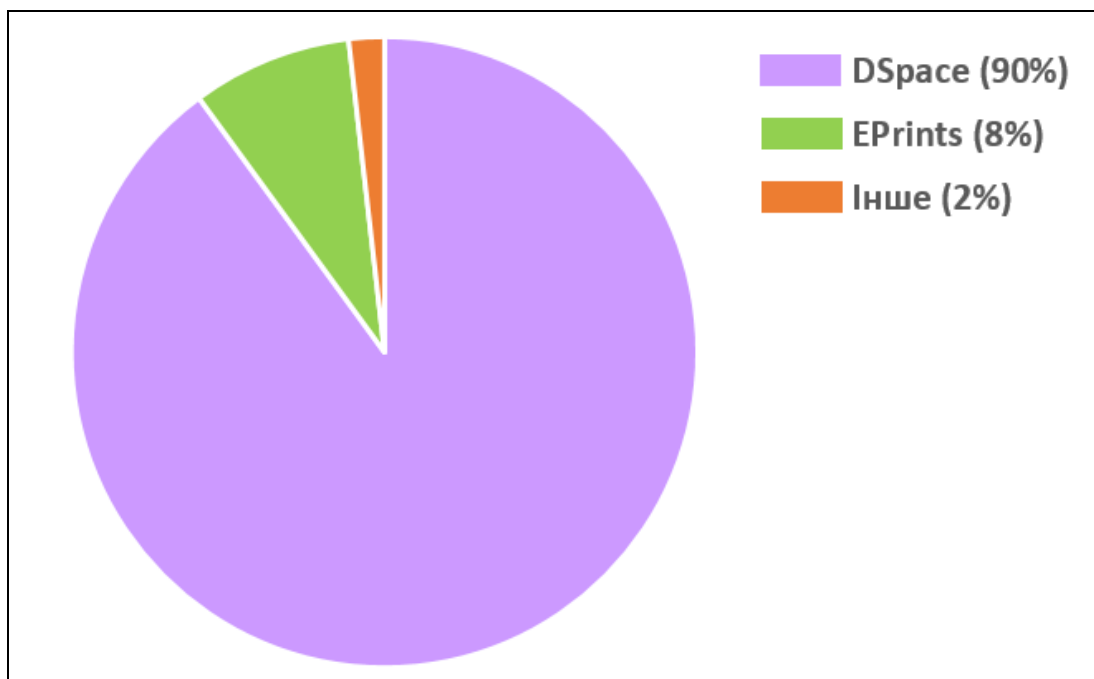


Рисунок 6 – Рейтинг репозитаріїв відкритого доступу в Україні

Тобто програмна платформа DSpace є беззаперечним лідером серед засобів для розгортання репозитарію. Розглянемо її більш детально.

1.1 Огляд існуючих рішень

DSpace – це веб-програма, яка дозволяє дослідникам і вченим публікувати документи та дані. Незважаючи на те, що DSpace має деякі функції, що збігаються з системами керування вмістом і системами керування документами, програмне забезпечення репозитарію DSpace відповідає певним потребам, як система цифрових архівів.

DSpace зосереджена на довгостроковому зберіганні, доступі та збереженні цифрового вмісту, що робить її затребуваною для академічних,

некомерційних і комерційних організацій, які створюють відкриті цифрові сховища.

DSpace безкоштовний і легко встановлюється «з коробки» і повністю налаштовується відповідно до потреб будь-якої організації.

DSpace зберігає та забезпечує легкий і відкритий доступ до всіх типів цифрового вмісту, включаючи текст, зображення, рухомі зображення, відеофайли і набори даних.

DSpace має підтримку постійно зростаючої спільноти розробників, які прагнуть вже протягом 20 років розширювати та вдосконалювати програмне забезпечення. Тому кожна наступна версія цього додатку отримує доопрацювання та має певні переваги над попередніми.

Перша загальнодоступна версія DSpace була випущена в листопаді 2002 року як спільна робота розробників з MIT (Массачусетський технологічний інститут) і HP Labs (дослідницька лабораторія компанії з інформаційних технологій Hewlett-Packard).

Після першої зустрічі групи користувачів у березні 2004 року група зацікавлених установ сформувала федерацію DSpace Federation, яка визначила управління майбутньою розробкою програмного забезпечення, прийнявши модель розвитку спільноти Apache Foundation, а також заснувавши групу DSpace Committer Group.

У липні 2007 року, коли спільнота користувачів DSpace зросла, HP і MIT спільно створили DSpace Foundation, некомерційну організацію, яка забезпечувала керівництво проектом та його технічну підтримку.

У травні 2009 року співпраця над відповідними проектами та зростаюча взаємодія між організаціями DSpace Foundation і Fedora Commons призвела до їх об'єднання в некомерційну організацію під назвою DuraSpace для виконання спільної місії.

					КГ 06.16.001.00 ДП ПЗ	Арк.
						14
Ізм.	Лист	№ докум.	Підпис	Дата		

DuraSpace і LYRASIS об'єдналися в липні 2019 року. Наразі спільнота програмного забезпечення та користувачів DSpace отримує керівництво від LYRASIS. Роботою спільноти та підтримкою програмного забезпечення керують робочі групи Governance та DSpace.

Проект DSpace розроблявся як програмне забезпечення для сховищ, що забезпечує засоби для того, щоб зробити інформацію відкритою, доступною та легкою для керування і структуризації.

Основною метою було створити програмне забезпечення з відкритим вихідним кодом, використовуючи навички активної спільноти розробників, енергію та розуміння зацікавлених і активних користувачів, а також фінансову підтримку учасників проекту та зареєстрованих постачальників послуг.

Понад 3000 організацій зараз використовують програмне забезпечення DSpace у виробничому чи проектному середовищі. Найпоширенішим використанням є дослідницькі бібліотеки, такі як інституційні репозиторії, однак є багато організацій, які використовують програмне забезпечення для розміщення та керування тематичними репозиторіями, репозиторіями набору даних або медіа-репозиторіями. DSpace розроблений таким чином, щоб бути гнучким і налаштовуватися для потреб будь-якої організації, незалежно від типу її діяльності.

Переваги DSpace включають:

- безкоштовне програмне забезпечення з відкритим кодом;
- повністю налаштовується відповідно до потреб користувача;
- дозволяє керувати та зберігати всі формати цифрового вмісту (файли PDF, Word, JPEG, MPEG, TIFF);
- пошук метаданих і повного текстового вмісту на основі Apache SOLR;
- підтримка UTF-8;

					КГ 06.16.001.00 ДП ПЗ	Арк.
						15
Ізм.	Лист	№ докум.	Підпис	Дата		

- інтерфейс доступний 22 мовами;
- деталізований груповий контроль доступу, що дозволяє встановлювати дозволи на рівні окремих файлів;
- оптимізовано для індексування Google Scholar;
- крім того, багато інших функцій, які з'являються з кожним випуском оновлень DSpace.

DSpace – це програмне забезпечення цифрового сховища з відкритим вихідним кодом, яке широко використовується установами для керування та збереження цифрового вмісту. Хоча в DSpace є беззаперечні переваги, ця платформа також має певні недоліки, які користувачі повинні враховувати:

1. Складність: DSpace може бути складним для встановлення та конфігурації, що вимагає певних технічних знань і знань базових технологій. Налаштування та розширення програмного забезпечення для відповідності конкретним вимогам також може потребувати тривалого навчання та тьюторів.

2. Витратність ресурсів: для ефективної роботи DSpace потрібні відповідні апаратні ресурси, включаючи дисковий простір, пам'ять і потужність процесора (потрібні: потужний, бажано серверний процесор, великий обсяг швидкої оперативної пам'яті, RAID-масив з твердотільних накопичувачів, системи резервування та безперервного електричного живлення тощо). Великі репозитарії або установи з обмеженими ресурсами можуть зіткнутися з проблемами під час розгортання та підтримки DSpace.

3. Обмеження налаштувань: хоч DSpace пропонує низку параметрів конфігурації, налаштування, окрім цих параметрів, можуть бути обмеженими. Реалізація широких налаштувань або модифікація основних

функцій може вимагати значних зусиль в розробці від того, хто буде розгорнути платформу, та неабиякого досвіду в подібних справах.

4. Інтерфейс користувача: стандартний інтерфейс користувача DSpace є функціональним, але може не запропонувати найсучасніший або візуально привабливий досвід. Установам, які прагнуть отримати досконалий і зручний інтерфейс, може знадобитися інвестувати в додаткові розробки або інтегрувати DSpace з іншими програмними рішеннями.

5. Підтримка спільноти: хоча DSpace має активну спільноту користувачів і розробників, рівень підтримки та доступність ресурсів може відрізнятися. Користувачі можуть зіткнутися з труднощами під час пошуку конкретної документації або отримання своєчасних відповідей на свої запити, особливо в менш поширених випадках використання.

6. Інтеграція із зовнішніми системами: інтеграція DSpace з іншими системами або програмами сторонніх розробників іноді може бути складною. Під час спроби підключити DSpace до різних платформ можуть виникнути проблеми сумісності або потреба в індивідуальній розробці.

7. Продуктивність для великих сховищ: DSpace може мати проблеми з продуктивністю під час керування дуже великими сховищами з великим обсягом вмісту та одночасними користувачами. Масштабованість і оптимізація продуктивності можуть вимагати додаткових інвестицій або оптимізації інфраструктури.

Важливо зазначити, що хоча DSpace має ці недоліки, багато з них можна пом'якшити за допомогою належного планування, досвіду та наявності ресурсів.

EPrints є другою за популярністю відкритою системою управління цифровими репозитаріями, яка використовується для збереження та поширення наукових матеріалів як в Україні, так і в світі.

					КГ 06.16.001.00 ДП ПЗ	Арк.
						17
Ізм.	Лист	№ докум.	Підпис	Дата		

Наведемо відмінності між DSpace та EPrints:

1. Архітектура: EPrints та DSpace мають різні архітектурні підходи. EPrints базується на модульній архітектурі, де репозитарій будується з використанням різних модулів, які можуть бути легко налаштовані та розширені. З іншого боку, DSpace використовує платформу Java та базується на структурованій базі даних, що забезпечує стабільність та надійність.

2. Конфігурація та налаштування: EPrints зазвичай вважається більш простим у налаштуванні та конфігурації, оскільки він пропонує графічний інтерфейс користувача для настройки багатьох параметрів. DSpace, натомість, вимагає більше технічних знань та володіння командним рядком для зміни налаштувань.

3. Функціональні можливості: EPrints та DSpace надають схожий набір функціональних можливостей, таких як керування метаданими, розподіл ресурсів, доступ до вмісту, пошук тощо. Однак, деякі функції можуть мати різні імплементації та можливості в залежності від системи.

4. Інтерфейс користувача: в обох системах є стандартні інтерфейси користувача, але їх зовнішній вигляд і можливості можуть відрізнятися. EPrints має більш гнучкі можливості для налаштування зовнішнього вигляду та стилізації, тоді як DSpace має менш гнучкий, але функціональний інтерфейс.

5. Спільнота та підтримка: як EPrints, так і DSpace мають активні спільноти користувачів та розробників. Обидві системи надають документацію, форуми для обговорення та підтримки, але обсяг та доступність ресурсів можуть відрізнятися.

6. Шаблони та розширення: EPrints надає ширший набір готових шаблонів та додаткових розширень, які можна використовувати для здійснення швидких налаштувань та розширення функціональності.

					КГ 06.16.001.00 ДП ПЗ	Арк.
						18
Ізм.	Лист	№ докум.	Підпис	Дата		

DSpace, з іншого боку, зазвичай потребує більшої «ручної» роботи для виконання певних налаштувань та для розширення.

Обидві системи мають свої переваги та недоліки, і вибір між ними залежить від конкретних потреб, вмінь та ресурсів організації.

Перед використанням DSpace або EPrints слід ознайомитися з документацією, провести порівняння функціональності. Їх придатність як цифрового репозиторію залежить від конкретних вимог, можливостей і обмежень кожної установи чи організації. Хоча програмні платформи і є безкоштовними, для успішного їх розгортання в закладі освіти виникає потреба у додаткових витратах на підготовку персоналу та реалізацію проекту.

При реалізації інституційного репозитарію на фізико-технічному факультеті Дніпровського національного університету ім. Олеся Гончара та репозитарію Вінницького національного аграрного університету не використовувалися DSpace та EPrints. [1]

Тобто є заклади освіти, яким вдалося розгорнути репозитарій за допомогою власних напрацювань.

Розглянемо технології та засоби розробки, за допомогою яких пропонується створити репозитарій методичної роботи для ВСП «ОТФК ОНТУ».

1.2 Технології та засоби розробки

Для розробки графічного інтерфейсу та клієнтського додатку (Front-End) використовуватимуться JS-бібліотеки – фреймворки Bootstrap 5.2 та JQuery 3.6. Для розробки серверного коду додатку (Back-End) використовуватиметься мова PHP 8.1. Для супутніх із розробкою операцій (DevOps), наприклад, налаштування серверу, використовуватиметься веб-

					КГ 06.16.001.00 ДП ПЗ	Арк.
						19
Ізм.	Лист	№ докум.	Підпис	Дата		

сервер Apache 2.4, мережевий сервіс тунелювання diode 1.8.1 та середовище розробки (IDE) Python 3.11.

Bootstrap 5.2 – популярний інтерфейсний фреймворк, який використовується для розробки адаптивних і орієнтованих на мобільні пристрої веб-сайтів і веб-додатків. Він надає колекцію компонентів CSS і JavaScript, а також попередньо оформлені шаблони та утиліти, щоб оптимізувати процес розробки та забезпечити узгодженість дизайну та функціональності. Перелічимо нижче його можливості.

Адаптивний веб-дизайн: Bootstrap спрощує створення адаптивних макетів, які адаптуються до різних розмірів екрана та пристроїв. Його сітка дозволяє розробникам легко організувати та вирівнювати вміст, що робить його придатним для створення веб-сайтів, які добре виглядають і функціонують як на настільних ПК та ноутбуках, так і на планшетах і мобільних пристроях.

Компоненти інтерфейсу користувача: Bootstrap пропонує широкий спектр готових до використання компонентів інтерфейсу користувача, таких як навігаційні меню, кнопки, форми, моди, каруселі тощо. Ці компоненти можна легко налаштувати та інтегрувати у веб-проекти, заощаджуючи час і зусилля на розробку.

Послідовний стиль: Bootstrap надає згуртований і візуально привабливий набір стилів і тем, забезпечуючи узгоджений вигляд і досвід користувача на різних сторінках і розділах веб-сайту. Використовуючи попередньо визначені класи CSS Bootstrap, розробники можуть досягти професійного та відшліфованого дизайну без створення великого власного стилю.

Підхід, орієнтований на мобільні пристрої: із збільшенням використання мобільних пристроїв Bootstrap дотримується підходу, орієнтованого на мобільні пристрої, віддаючи пріоритет дизайну та

					КГ 06.16.001.00 ДП ПЗ	Арк.
						20
Ізм.	Лист	№ докум.	Підпис	Дата		

функціональності для менших екранів. Це гарантує, що веб-сайти, створені за допомогою Bootstrap, оптимізовані для мобільних користувачів і пропонують безперебійну роботу на різних пристроях з різною площиною екрану (в будь-якому випадку користувач може бути впевнений, що площа дисплею використовується якнайповніше).

Сумісність з веб-переглядачами (браузерами): фреймворк Bootstrap розроблено для роботи в різних веб-браузерах, у тому числі в таких популярних, як Chrome, Firefox, Safari та Microsoft Edge. Він піклується про обробку невідповідностей веб-переглядача та забезпечує стабільну роботу для користувачів незалежно від браузера, який вони використовують.

Налаштування та оформлення тем: Bootstrap дозволяє розробникам налаштовувати стилі та компоненти за замовчуванням відповідно до їхніх конкретних вимог до дизайну. Використовуючи змінні та Sass (надбудова над CSS, скриптова мова, що інтерпретується в каскадні таблиці стилів, англ. Syntactically Awesome Style Sheets), розробники можуть змінювати кольори, типографіку, інтервали та інші аспекти фреймворку або навіть створювати за його допомогою власні теми.

Загалом Bootstrap 5.2 широко використовується для пришвидшення процесу розробки, створення візуально привабливих веб-сайтів з адаптивним дизайном, а також для забезпечення узгодженої взаємодії з користувачем на різних пристроях з різними екранами і браузерах.

jQuery 3.6 – популярна бібліотека JavaScript, яка спрощує обхід дерева тегів HTML-документів, обробку подій, анімацію та взаємодію з об'єктною моделлю документа (DOM). Ця бібліотека надає широкий спектр функцій і утиліт, які полегшують розробку інтерактивних і динамічних веб-сторінок. Приведемо кілька типових випадків використання jQuery 3.6.

					КГ 06.16.001.00 ДП ПЗ	Арк.
						21
Ізм.	Лист	№ докум.	Підпис	Дата		

Маніпулювання DOM: jQuery спрощує роботу з DOM, надаючи стислий та інтуїтивно більш зрозумілий синтаксис. Це дозволяє розробникам легко вибирати та керувати елементами HTML, змінювати вміст, змінювати стилі та обробляти події. Селектори та методи jQuery роблять ефективним проходження та маніпулювання деревом DOM.

Обробка подій: jQuery спрощує обробку подій у JavaScript. Фреймворк надає методи для прикріплення прослуховувачів подій до елементів HTML і обробки (отримання реакції на події) різних вчинків користувача, таких як клацання, рухи миші, введення з клавіатури тощо. Можливості обробки подій jQuery дозволяють розробникам створювати інтерактивні веб-сторінки з привабливим зовнішнім виглядом.

AJAX і HTTP-запити: jQuery містить потужний набір функцій для створення асинхронних HTTP-запитів за допомогою технології AJAX (асинхронний JavaScript і XML). Це спрощує надсилання запитів на сервер, отримання даних у фоновому режимі та оновлення частин веб-сторінки без необхідності її повного перезавантаження. Функції AJAX, реалізовані у фреймворку jQuery, дозволяють веб-програмістам розробляти динамічні та інтерактивні веб-додатки.

Анімація та ефекти: jQuery надає ряд функцій анімації та ефектів для створення візуально привабливих елементів веб-сторінок. Це дозволяє розробникам анімувати HTML-елементи, застосовувати переходи, ефекти появи / зникнення, ковзання та ін. Наявні можливості анімації jQuery підвищують інтерактивність і естетику веб-сторінок.

Сумісність із різними браузерами: jQuery допомагає впоратися з невідповідностями веб-переглядача та забезпечує послідовний API, який працює в різних веб-браузерах. Він абстрагується від складнощів специфічного для браузера JavaScript і забезпечує єдиний інтерфейс, що полегшує написання коду, який правильно функціонує в різних браузерах.

					КГ 06.16.001.00 ДП ПЗ	Арк.
						22
Ізм.	Лист	№ докум.	Підпис	Дата		

Екосистема плагінів: jQuery має величезну екосистему плагінів, розроблених веб-спільнотою. Ці плагіни розширюють основні функції jQuery, надаючи додаткові функції, віджети та інтеграцію з іншими бібліотеками та фреймворками. Екосистема плагінів дозволяє розробникам використовувати існуючі рішення та додавати розширені функції до своїх проектів з мінімальними зусиллями.

Загалом jQuery 3.6 використовується для спрощення та оптимізації розробки JavaScript, полегшуючи маніпулювання об'єктами DOM, обробку подій, виконання запитів AJAX, створення анімації та досягнення крос-браузерної сумісності. Його стислий синтаксис і багатий набір функцій роблять його потужним інструментом для інтерфейсної веб-розробки. Однак варто зазначити, що завдяки прогресу сучасного JavaScript і доступності власних API браузера, використання jQuery останніми роками зменшилося, і розробники часто віддають перевагу використанню «чистого» JavaScript або більш сучасних фреймворків/бібліотек, таких як React або Vue.js.

PHP (Hypertext Preprocessor) є популярною мовою програмування, яка широко використовується для розробки веб-додатків. PHP надає розширені можливості для взаємодії з сервером та створення динамічних веб-сторінок. Основні можливості PHP для розробки веб-додатків перелічимо нижче.

Робота з базами даних: PHP має різноманітні розширення для підключення до різних типів баз даних, таких як MySQL, PostgreSQL, SQLite та багато інших. Є можливість створювати, зчитувати, оновлювати та видаляти дані з баз даних, виконувати складні запити та керувати транзакціями.

Робота з HTML-формами та обробка даних з них: PHP надає зручні функції для обробки форм, отримання даних від користувачів та їх

					КГ 06.16.001.00 ДП ПЗ	Арк.
						23
Ізм.	Лист	№ докум.	Підпис	Дата		

перевірки. Дозволяє зчитувати, перевіряти та зберігати дані з HTML-форм, завантажувати файли, верифікувати користувачів та їх права, а також реалізовувати різні заходи безпеки.

Робота з зображеннями: PHP надає функціонал для маніпулювання зображеннями, такий як зміна розміру, обрізка, маніпуляції пікселями, створення зображень з нуля та інші операції. Можна створювати графіки, обробляти фотографії та генерувати зображення на основі даних.

Робота з файлами: PHP дозволяє працювати з файлами на сервері. Є можливість створювати, читати, записувати та видаляти файли, керувати папками (каталогами), зчитувати та записувати дані в файли, а також здійснювати різні операції з файловими архівами.

Робота із зовнішніми сервісами: PHP може взаємодіяти з іншими веб-сервісами та API, використовуючи різні протоколи, такі як HTTP, SOAP, REST тощо. Є можливість відправляти та отримувати дані зовнішніх сервісів, таких як соціальні мережі, платіжні шлюзи, картографічні сервіси тощо.

Створення сесій та керування станом: PHP дозволяє зберігати стан між різними запитами користувача за допомогою сесій. Є можливість створювати сесії, зберігати дані в таких сесіях та керувати правами доступу до них. Це дозволяє реалізувати автентифікацію користувачів, зберігати персоналізовані дані та інші функції, пов'язані зі збереженням стану веб-системи.

Шаблонізація: PHP підтримує використання шаблонів для розділення логіки програми від представлення даних. Є можливість використовувати шаблонні движки, наприклад такі як Smarty або Twig, для зручного та організованого відображення даних на сторінках.

Робота з електронною поштою: PHP надає функції для відправки та отримання електронної пошти. Можливо створювати, відправляти та

					КГ 06.16.001.00 ДП ПЗ	Арк.
						24
Ізм.	Лист	№ докум.	Підпис	Дата		

обробляти електронні листи, працювати із вкладеними файлами, керувати поштовими скриньками та виконувати інші дії, пов'язані з електронною поштою.

Мова PHP також підтримує розширення, бібліотеки та фреймворки, які значно розширюють його функціональні можливості та спрощують розробку веб-додатків.

PHP 8.1 – це основний випуск мови програмування PHP, який містить кілька нових функцій, покращень і оптимізацій. PHP – це широко поширена серверна мова сценаріїв, яка в основному використовується для веб-розробки.

PHP широко використовується для створення динамічних та інтерактивних веб-сайтів. З версії PHP 8.1 веб-розробники можуть скористатися новими функціями та вдосконаленнями для написання більш ефективного та безпечного коду. PHP 8.1 надає надійний набір інструментів, бібліотек і фреймворків (таких як Laravel та Symfony), які полегшують задачі веб-розробника, включаючи доступ до бази даних, обробку HTML-форм, керування сесіями тощо.

PHP 8.1 має в своєму арсеналі різні оптимізації продуктивності, включаючи компіляцію на льоту (за технологією JIT – англ. just-in-time), що значно підвищує швидкість виконання PHP-коду. Компілятор JIT динамічно оптимізує та компілює сегменти коду, що часто виконуються, що призводить до швидшого часу виконання та покращення загальної продуктивності.

Безпека та строгість типів: PHP 8.1 покращує безпеку типів і вводить суворіші функції введення. Він вводить новий тип повернення never, який вказує, що функція ніколи не повертається. Крім того, введено суворіші перевірки типів для арифметичних операцій і операцій з

					КГ 06.16.001.00 ДП ПЗ	Арк.
						25
Ізм.	Лист	№ докум.	Підпис	Дата		

окремими бітами, що допомагає розробникам виявляти потенційні помилки та підвищувати надійність коду загалом.

Покращення нульових значень: PHP 8.1 пропонує точнішу обробку нульових значень. Функція об'єднання типів дозволяє розробникам вказати, що параметр або значення, що повертається, може мати декілька типів, у тому числі і NULL. Це спрощує вирази та застосування вимог до нульових значень у сигнатурах функцій.

Волокна (fibers, файберз) та асинхронне програмування: PHP 8.1 представляє концепцію волокон (функції, що можуть переривати своє виконання), які є легкими кооперативними багатозадачними одиницями. Волокна дозволяють розробникам створювати асинхронний код, який можна призупиняти та відновлювати, надаючи альтернативу традиційним моделям асинхронного програмування на основі зворотного виклику. Це полегшує написання ефективних і масштабованих асинхронних програм за допомогою PHP.

Покращена обробка помилок та удосконалення безпеки: PHP 8.1 представляє новий FiberExit-виняток, який дозволяє плавно завершувати волокна. Він також покращує звіти про помилки, надаючи більш детальні повідомлення про помилки, що полегшує виявлення та усунення проблем у програмному коді. В PHP 8.1 є кілька покращень безпеки, включаючи кращу обробку даних, які надає користувач, і покращені параметри безпеки за замовчуванням.

Веб-сервер Apache 2.4 є одним з найпопулярніших веб-серверів у світі. Він широко використовується для розгортання веб-сайтів та веб-додатків і надає ряд важливих функцій і можливостей.

Серверна обробка HTTP-запитів: Apache 2.4 може обробляти HTTP-запити, що надходять від клієнтів (веб-браузерів) і надсилати їм відповіді. Веб-сервер взаємодіє з клієнтами за допомогою протоколу HTTP і може

					КГ 06.16.001.00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		26

обслуговувати різні типи веб-запитів, наприклад такі як GET, POST, PUT, DELETE та ін.

Розгортання веб-сайтів: Apache 2.4 дозволяє розгортати веб-сайти, надаючи доступ до веб-сторінок, статичних файлів, зображень та інших ресурсів. Він може обслуговувати різні типи контенту, включаючи HTML, CSS, JavaScript, зображення, аудіо, відео та інші файлові формати.

Обробка динамічного контенту: Apache 2.4 може взаємодіяти з мовами програмування, такими як PHP, Python, Perl та інші, щоб обробляти динамічний контент на веб-сервері. Він може бути налаштований для виконання скриптів і передачі керування до відповідних інтерпретаторів, які динамічно генерують веб-сторінки.

Керування віртуальними хостами: Apache 2.4 дозволяє налаштовувати та керувати віртуальними хостами. Це означає, що є можливість розгортання декількох веб-сайтів на одному сервері, кожен зі своїм власним доменним ім'ям та IP-адресою.

Керування безпекою: Apache 2.4 надає різні можливості для забезпечення безпеки веб-додатків. Він підтримує криптографічні протоколи, такі як SSL / TLS, для шифрування комунікації між сервером і клієнтами, може встановлювати правила контролю доступу на основі IP-адрес, паролів або інших факторів, та включає захист від певних типів атак, таких як крос-сайтовий скриптинг (XSS) та SQL-ін'єкції.

Розширення та модульність: Apache 2.4 підтримує розширення та модулі, що дозволяють розширити його функціональність. Можливо додавати сторонні модулі, які надають додаткові можливості, наприклад, компресію контенту, кешування, логування, автентифікацію, реверс-проксі тощо.

Моніторинг та логування: Apache 2.4 забезпечує механізми для моніторингу роботи сервера та логування подій. Є можливість

					КГ 06.16.001.00 ДП ПЗ	Арк.
						27
Ізм.	Лист	№ докум.	Підпис	Дата		

відстежувати активність сервера, помилки, запити клієнтів тощо. Це допомагає відстежувати проблеми та забезпечувати належне функціонування сервера.

Зрештою, можна зробити висновок, що веб-сервер Apache 2.4 має багатий набір функцій і можливостей, які дозволяють розгортати та керувати веб-сайтами та веб-додатками з високою надійністю, безпекою та розширюваністю.

Сервіс тунелювання Diode є розподіленою мережею, що є альтернативою платному комерційному сервісу ngrok, та надає можливість забезпечити безпеку та конфіденційність при обміні даними між різними комп'ютерами або пристроями через мережу Інтернет. Diode пропонує захищений канал зв'язку, який гарантує односторонню передачу даних, що означає, що дані можуть лише відправлятись з одного кінця до іншого, але не можуть бути повернуті або змінені на зворотний шлях.

У системі Diode використовується принцип криптографічних функцій та блокування доступу для забезпечення цілісності та безпеки даних. Кожна сторона, що взаємодіє з мережею Diode, володіє ключами шифрування, що забезпечують захист від перехоплення та зміни даних.

Сервіс Diode може бути використаний для безпечного обміну даними в різних сценаріях, таких як захищена передача даних між серверами, захист IoT-пристроїв від несанкціонованого доступу, підтримка безпечних транзакцій та інше. Він забезпечує захист від різноманітних загроз, включаючи атаки MITM (Man-in-the-Middle) та перехоплення даних.

Diode може бути використаний як окремий сервіс або вбудований у різні додатки та протоколи для забезпечення безпеки та конфіденційності обміну даними.

					КГ 06.16.001.00 ДП ПЗ	Арк.
						28
Ізм.	Лист	№ докум.	Підпис	Дата		

Python – це потужна та універсальна мова програмування, яка використовується в різних сферах і має широкий спектр застосувань. Нижче зробимо їх перелік.

Веб-розробка: Python використовується для розробки веб-додатків і веб-сайтів. Його фреймворки, такі як Django і Flask, дозволяють швидко та ефективно створювати веб-додатки з розширеними можливостями, включаючи обробку HTTP-запитів, роботу з базами даних, шаблонізацію, аутентифікацію та ін.

Наукові обчислення та аналіз даних: Python є популярним вибором для наукових обчислень і аналізу даних. Бібліотеки, такі як NumPy, Pandas, SciPy та Matplotlib, надають потужні інструменти для роботи з числовими даними, виконання математичних обчислень, статистичного аналізу, машинного навчання та візуалізації даних.

Штучний інтелект та машинне навчання: Python є популярним рішенням для розробки алгоритмів штучного інтелекту та систем машинного навчання. Бібліотеки, такі як TensorFlow, PyTorch та scikit-learn, надають потужні інструменти для реалізації нейронних мереж, глибинного навчання, класифікації, кластеризації, обробки природної мови та інших завдань машинного навчання.

Сценарії та автоматизація: Python може бути використаний для написання сценаріїв та автоматизації завдань. Його простий синтаксис та багатий набір стандартних бібліотек дозволяють легко писати скрипти для автоматизації завдань на різних платформах, роботи з файлами. Саме цю функцію середовища розробки Python було використано при рішенні завдання, що виносилося на дипломне проектування.

Python може бути використаний для налаштування та керування серверами на різних рівнях.

					КГ 06.16.001.00 ДП ПЗ	Арк.
						29
Ізм.	Лист	№ докум.	Підпис	Дата		

Python дозволяє створювати скрипти для автоматичного виконання задач серверного адміністрування. Є можливість використовувати бібліотеки, такі як Fabric або Paramiko, для підключення до сервера по протоколу SSH для реалізації функції віддаленого керування та виконання команд на ньому. Це дозволяє автоматизувати процеси налаштування сервера, установки програмного забезпечення, копіювання файлів, налаштування служб та ін.

Python може використовуватися для зчитування та запису конфігураційних файлів сервера. Наприклад, можливо створити скрипт, який зчитує конфігураційний файл веб-сервера Apache і вносить необхідні зміни, такі як додавання віртуальних хостів, налаштування параметрів безпеки тощо.

Python може бути використаний для розробки серверних програм, які збирають статистику, здійснюють моніторинг стану сервера та реагують на події.

В підсумку зазначимо, що Python є потужною мовою програмування з багатою екосистемою бібліотек, які крім іншого дозволяють створювати різноманітні серверні рішення та автоматизувати задачі серверного адміністрування.

2 Реалізація проекту

2.1 Технічне завдання на розробку

Репозитарій методичної роботи для ВСП «ОТФК ОНТУ» має мати привабливий та яскравий інтерфейс першого вікна.

При створенні репозитарію слід:

1. Передбачити доступ до репозитарію через мережу Інтернет за вказаною адресою (з використанням технології тунелювання чи ін.)

					КГ 06.16.001.00 ДП ПЗ	Арк.
						30
Ізм.	Лист	№ докум.	Підпис	Дата		

2. Розробити інтерфейс для адміністрування репозитарію, що включає можливість додавання/видалення папок та файлів, їх перейменування, копіювання та переміщення.

3. Створити структуру репозитарію методичної роботи по цикловим комісіям, по викладачам та по предметам.

[ЦИКЛОВА КОМІСІЯ «НАЗВА_ЦК»]

- [Викладач_1]

- [Викладач_2]

- [Викладач_3]

- [Дисципліна_1]

- [Дисципліна_2]

- [Дисципліна_3]

 - Навчальний посібник_1.pdf

 - Навчальний посібник_2.pdf

 - Навчальний посібник_3.pdf

4. Передбачити можливість пошуку за прізвищем викладача, за роком видання методичної розробки (закінчення назви файлу – останні чотири цифри), за назвою предмету та за типом методичної розробки (конспект лекцій, методичні вказівки до виконання лабораторних робіт, методичні вказівки до виконання практичних робіт, методичні вказівки до проведення семінарських занять, методичні вказівки до виконання курсової роботи (проекту), методичні вказівки до виконання самостійної роботи тощо).

5. Передбачити на головній сторінці репозитарію (після відкриття додатку) видачі сумарної інформації – скільки знаходиться всього робіт в репозитарії, скільки з них – конспектів лекцій, методичних вказівок до виконання практичних робіт, методичних вказівок до проведення

					КГ 06.16.001.00 ДП ПЗ	Арк.
						31
Ізм.	Лист	№ докум.	Підпис	Дата		

семінарських занять, методичних вказівок до виконання курсової роботи (проекту), методичних вказівок до виконання самостійної роботи ін. та який обсяг дискового простору вони займають.

6. Передбачити захист від несанкціонованого доступу (вхід до репозитарію має здійснюватися за попередньою авторизацією). Звичайний користувач повинен мати доступ лише на перегляд методичних розробок, він має обмеження на завантаження навчальної документації, адміністратор – може все переглядати і все завантажувати (і до репозитарію, і з нього).

Створення репозитарію включатиме ряд етапів: проектування дизайну веб-інтерфейсу та архітектури файлового менеджера; реалізація окремих додатків проекту та їх спільної роботи; узгодження роботи додатків і мережевого сервісу тунелювання; публікація додатку на хостинг та реалізація доступу до даних за вказаною адресою.

Розглянемо детально кожний із етапів створення репозитарію методичної роботи для ВСП «ОТФК ОНТУ».

2.2 Проектування дизайну веб-інтерфейсу та архітектури файлового менеджера

Розуміння вимог і потреб користувачів є першим і важливим кроком у розробці дизайну веб-інтерфейсу. Слід визначити, які функції та можливості повинен мати веб-додаток, на яку аудиторію він розрахований, які основні цілі мають бути досягнені. Це допоможе визначити напрямок дизайну та встановити основні вимоги до нього.

Другим кроком є розробка структури та інформаційної архітектури файлового менеджера. Слід чітко визначити, як користувачі будуть навігувати по додатку та звертатися до різних його функцій.

					КГ 06.16.001.00 ДП ПЗ	Арк.
						32
Ізм.	Лист	№ докум.	Підпис	Дата		

Інформаційна архітектура файлового менеджера (рис. 7) визначає яка інформація буде відображатися на різних сторінках, та як саме вона буде організована.

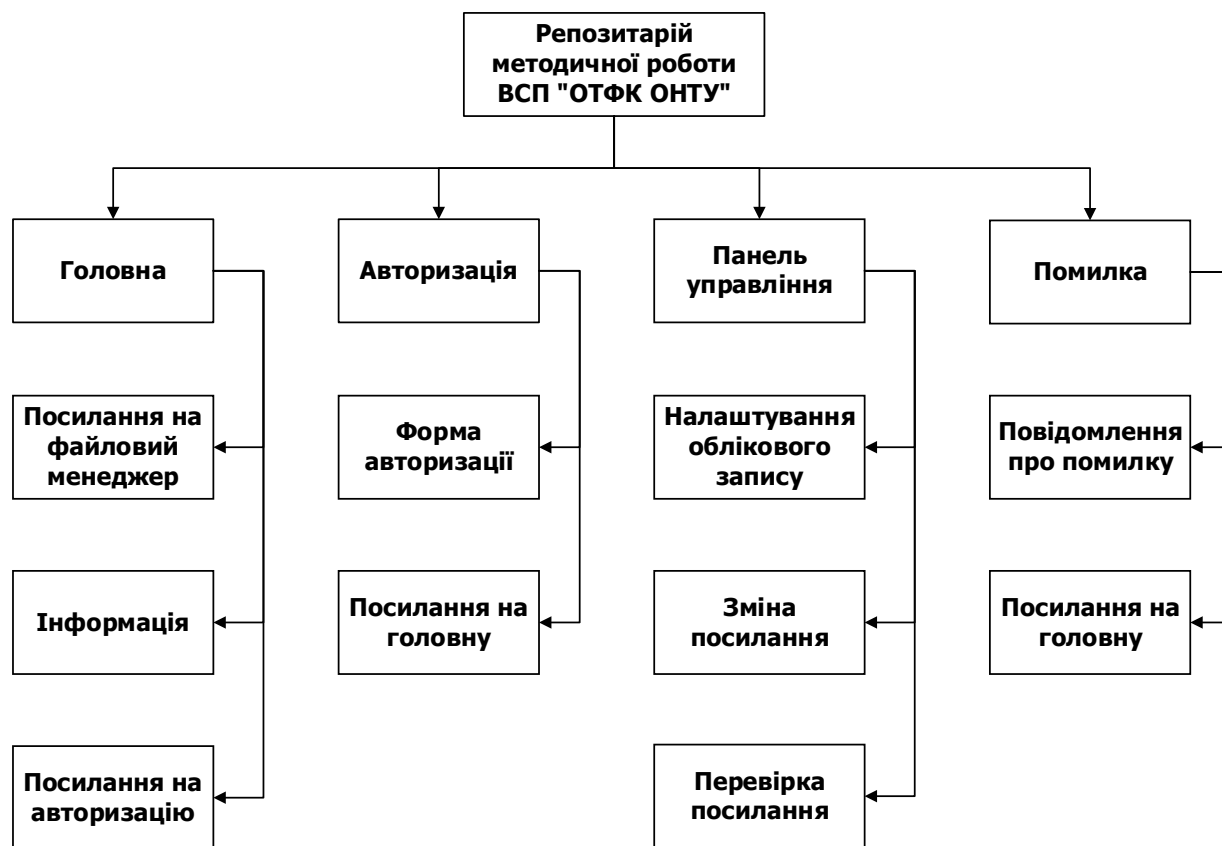


Рисунок 7 – Інформаційна архітектура файлового менеджера

Третім кроком у проектуванні дизайну веб-інтерфейсу є створення скетчів. Прості скетчі відображають основну структуру та компоненти майбутнього веб-додатку, допомагають візуалізувати розташування елементів, розміщення кнопок, полів вводу, меню та інших компонентів. Для створення скетчів можна використовувати як звичайний папір, так і спеціальні інструменти для створення візуальних скетчів (наприклад такі як Sketch або Figma).

Четвертим кроком є розробка дизайну веб-інтерфейсу. Дизайн веб-інтерфейсу повинен бути налаштований на потреби і очікування цільової аудиторії. Щоб розробити детальний дизайн веб-інтерфейсу необхідно

визначити кольорову палітру, типографіку, стилі кнопок, форм, заголовків, списків тощо. Слід враховувати принципи дизайну інтерфейсів, такі як простота, зручність використання.

Для розробки детального дизайну веб-інтерфейсу зазвичай використовують графічні редактори, такі як Adobe Photoshop або Sketch, або веб-інструменти для розробки дизайну, такі як Figma або Adobe XD.

Починаємо створення дизайну веб-інтерфейсу із створення нового проекту Adobe Photoshop (рис. 8). Тип для документа вказуємо – Web, розмір монтажної області – Веб (великий розмір, 1920, 1080) та натискаємо на кнопку ОК.

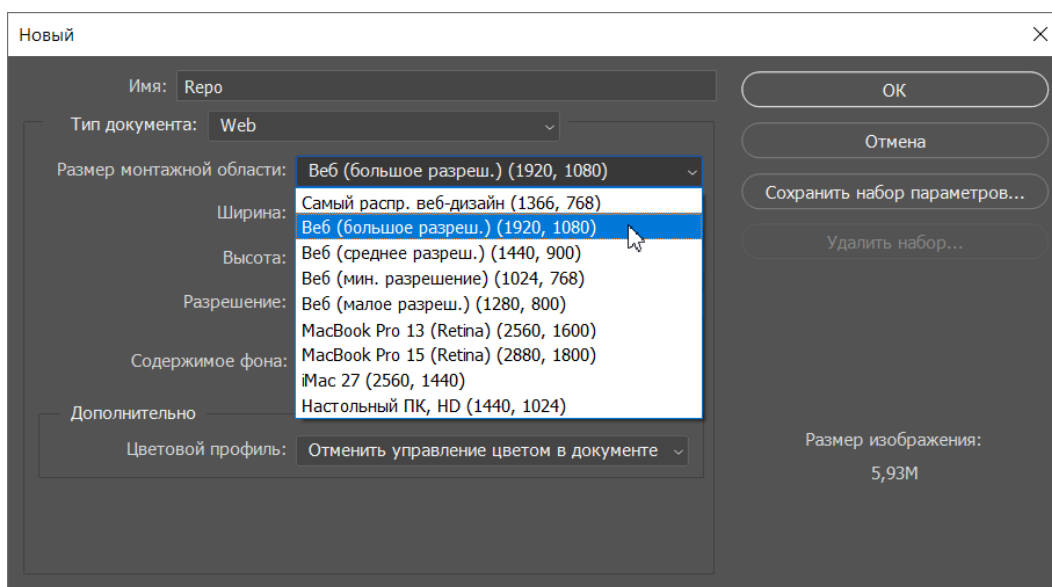


Рисунок 8 – Налаштування нового проекту Adobe Photophor

Без вдало підбраного графічного матеріалу зробити дизайн веб-інтерфейсу гармонійним, приємним для відвідувачів і при цьому зручним для користування практично неможливо.

Всі матеріали завантажуюмо з безкоштовних ресурсів (фотосток – Unsplash, бібліотека іконок – icon-icon, бібліотека шрифтів – fonts-online), де за ключовими словами можна здійснити фільтрацію та завантажити найбільш відповідні фотографії, іконки та шрифти.

В якості тла в проєкті буде використано растрове зображення (фото), яке відповідає тематиці проєкту. Зображення знаходимо в «стоці» за ключовим словом «library». Серед запропонованих результатів пошуку обираємо доречний та завантажуємо його у великому розмірі (Original size, роздільна здатність фото – 5013×4634 р).

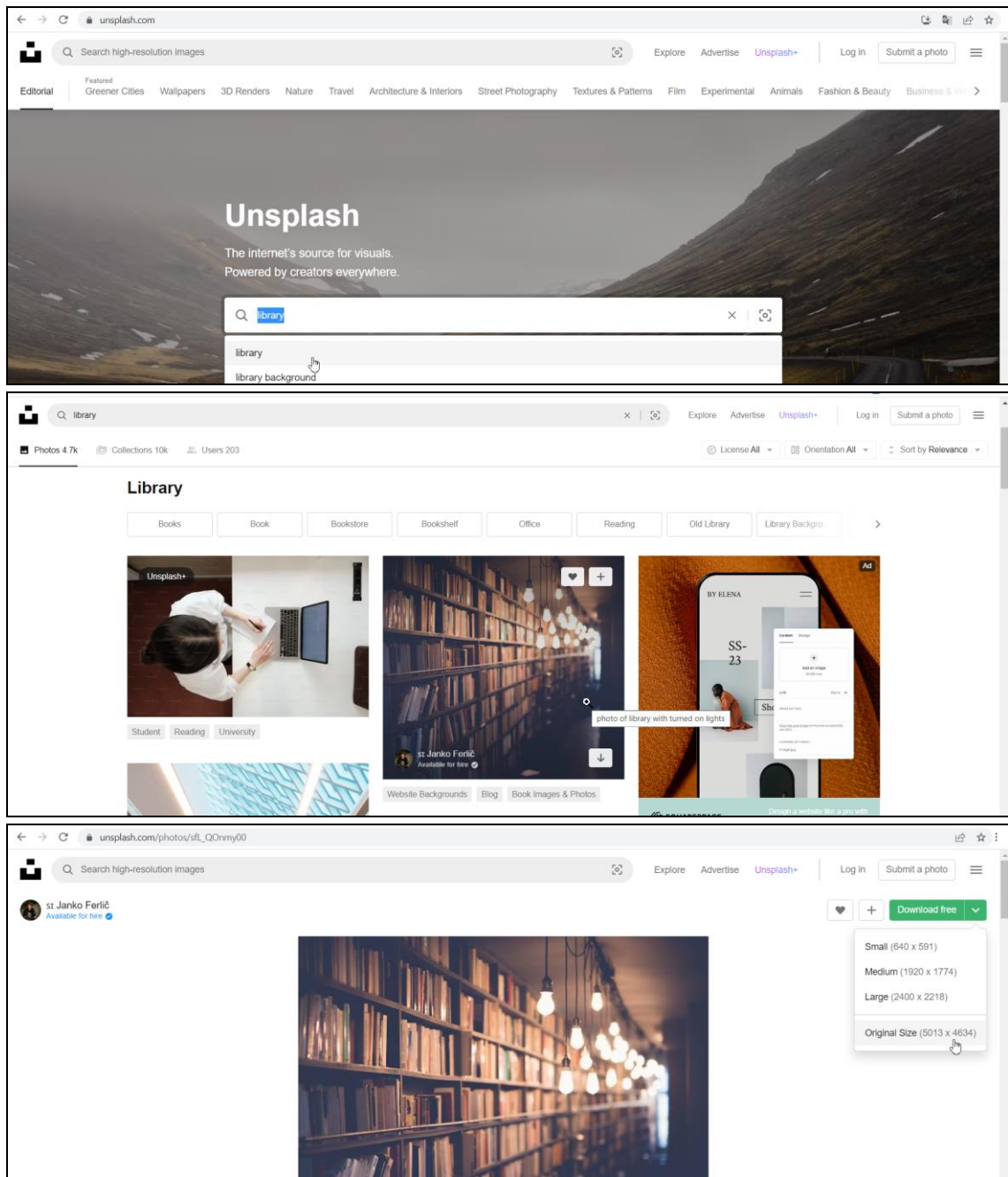


Рисунок 9 – Завантаження фонового зображення для проєкту

Ізм.	Лист	№ док.м.	Підпис	Дата

Розташуємо растрове зображення по всій монтажній області. Для цього додаємо в робочу область растрове зображення, яке слід перетворити на смарт-об'єкт для того, щоб воно не втратило якості після трансформації (зменшення розмірів).

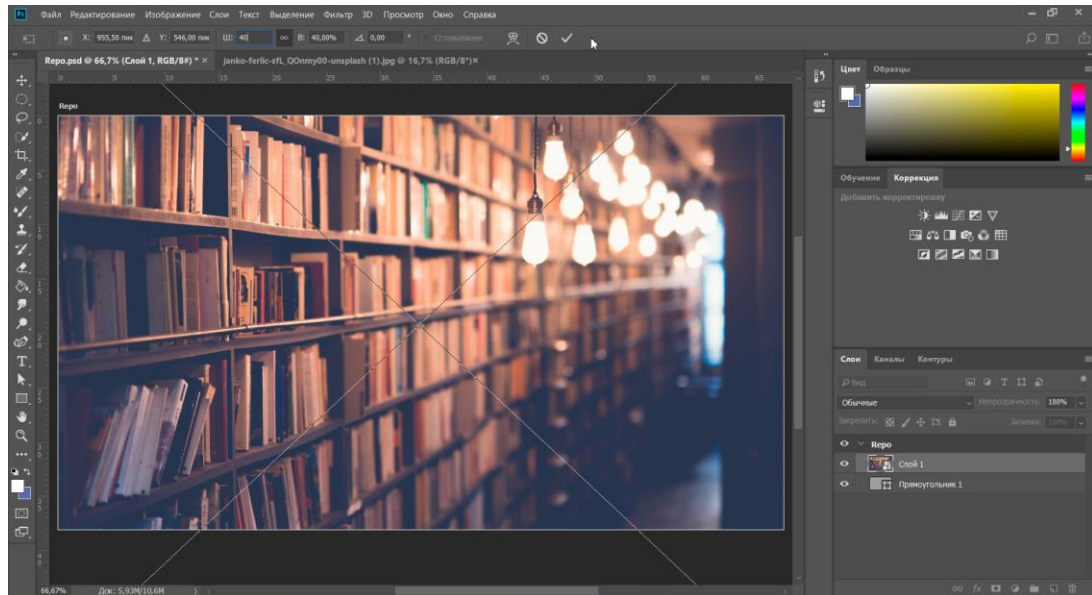


Рисунок 10 – Додавання фонового зображення

По центру монтажної області розташовуємо заголовок проекту «РЕПОЗИТАРІЙ», підзаголовок – «методичної роботи ВСП «ОТФК ОНТУ», а також елементи керування додатком – кнопка «Відкрити репозитарій», іконки «налаштування» та «інформація». Для акцентування уваги на елементах керування виділимо їх за допомогою прямокутника без заливки, що має обведення білого кольору.

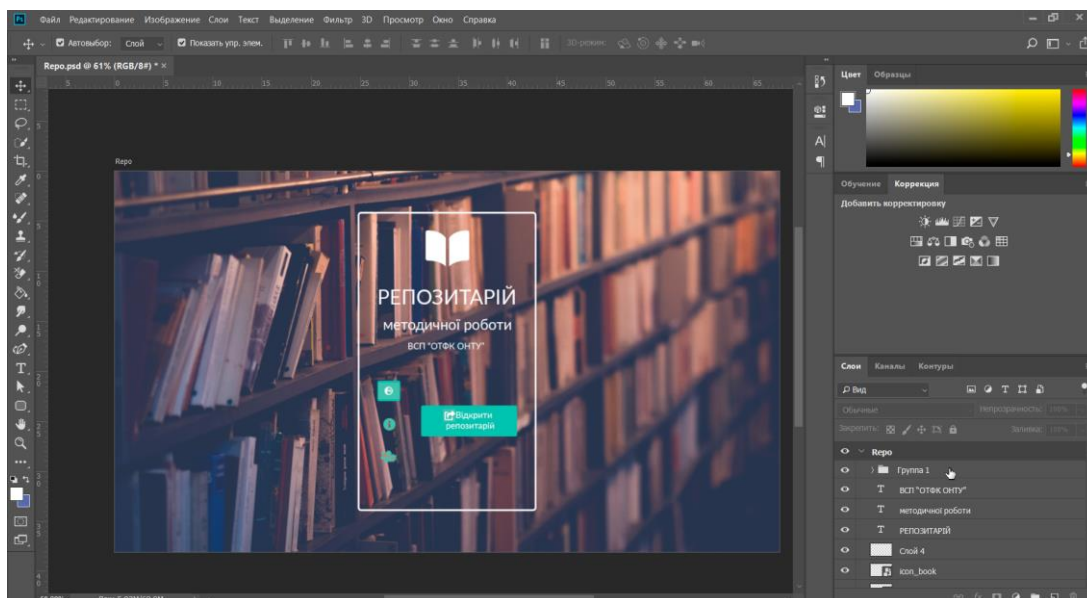


Рисунок 11 – Дизайн веб-інтерфейсу, розроблений в Adobe Photoshop

Після створення дизайну веб-інтерфейсу переходимо безпосередньо до його реалізації.

2.3 Реалізація додатку

Проектований репозитарій складатиметься з трьох програмних додатків (рис. 12):

- Python-додаток для керування роботою мережевого сервісу тунелювання diode, який забезпечуватиме доступ до матеріалів репозитарію, який розміщено на сервері, через мережу Інтернет;
- PHP-додаток на хостингу для реалізації функцій файлового менеджера;
- PHP-файловий менеджер на локальній машині для випадків, коли доступ до даних через мережу Internet неможливий.

PHP-додаток на хостингу являє собою додаток з монолітною архітектурою. Монолітна архітектура програмного додатку є традиційним підходом, в якому весь додаток розгортається як єдиний, незалежний від внутрішньої структури модуль.

										Арк.
										37
Ізм.	Лист	№ докум.	Підпис	Дата						

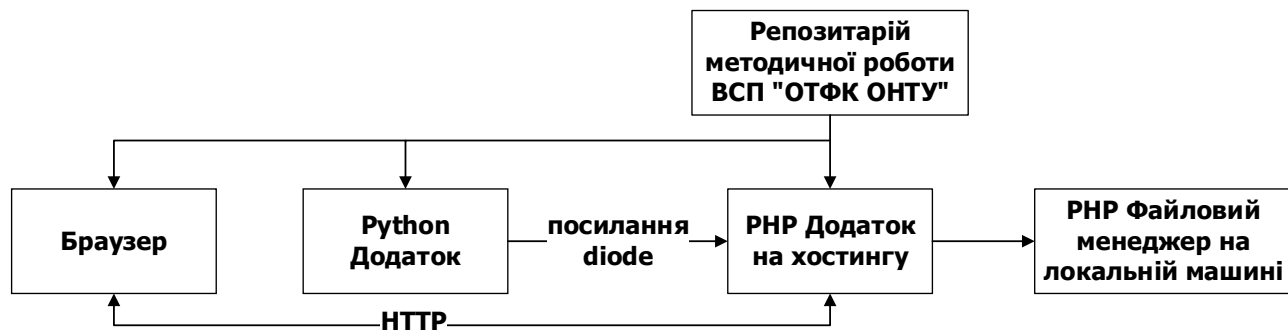


Рисунок 12 – Взаємодія програмних модулів

Розглянемо основні особливості монолітної архітектури. Монолітний додаток – це єдине, самодостатнє ціле. Всі компоненти додатку (наприклад, користувацький інтерфейс, бізнес-логіка, база даних тощо) розташовані в одному великому кодовому базисі. Це спрощує розгортання і моніторинг додатку.

Всі компоненти додатку спілкуються між собою безпосередньо. Наприклад, функції можуть викликати інші функції в межах додатку, а доступ до бази даних може бути здійснений безпосередньо з будь-якого компонента. Це забезпечує простоту взаємодії між компонентами, але може призводити і до збільшення зв'язності.

Монолітні додатки зазвичай розгортаються на одному сервері або в одному контейнері, що дозволяє економити ресурси та значно спрощує управління додатком.

Розробка і тестування монолітного додатку можуть бути більш простими, оскільки весь код знаходиться в одному місці, що полегшує виявлення і виправлення помилок.

Існує обмеження масштабованості монолітних додатків. Оскільки весь додаток працює як єдиний процес, масштабування може бути складним завданням. Збільшення навантаження на один компонент може вимагати масштабування всього додатку.

Внесення змін в монолітний додаток може бути складним і часом затратним, оскільки будь-яка зміна в коді може мати неочікувані наслідки на роботу інших компонентів.

Ізоляція помилок обмежена в монолітних додатках. Помилки можуть впливати на всі компоненти системи, оскільки вони спільно використовують ресурси. Одна некоректна операція може призвести до зупинки всього додатку.

Хоча монолітна архітектура має свої переваги, такі як простота в розробці та тестуванні, вона також може стати обмеженням у випадку потреби в масштабованості, гнучкості та легкості внесення змін до проекту. Тому в останні роки популярнішими стали інші архітектурні підходи, такі як мікросервісна архітектура, що дозволяє розбити додаток на більш маленькі та незалежні компоненти, забезпечуючи більшу масштабованість та гнучкість.

Так як репозитарій реалізовано з використанням принципів монолітної архітектури, то додаток являється мультисторінковим, кожен розділ в репозитарії є окремою сторінкою, які перезавантажуються.

2.3.1 Розробка Python-додатку

Python-додаток призначений для керування роботою мережевого сервісу тунелювання diode, який забезпечуватиме доступ до матеріалів репозитарію, який розміщено на сервері, через мережу Інтернет.

Diode – це сервіс у вигляді додатку, DevOps інструмент, котрий дозволяє зробити видимим локальний сайт без виділеної IP-адреси в глобальній мережі інтернет (WAN). Для цього використовується технологія тунелювання (VPN).

Тунелювання в сучасних комп'ютерних мережах – це процес передачі одного протоколу даних через інший протокол, який не надає

					КГ 06.16.001.00 ДП ПЗ	Арк.
						39
Ізм.	Лист	№ докум.	Підпис	Дата		

прямої підтримки для передачі цих даних. Тунелювання використовується для забезпечення безпеки, конфіденційності та надійності передачі даних.

Під час тунелювання дані, які передаються одним протоколом (називається внутрішнім протоколом), упаковуються в пакети і передаються через інший протокол (називається зовнішнім протоколом). Зовнішній протокол використовується для передачі упакованих пакетів через мережу, а потім вони розпаковуються на приймальному кінці, щоб отримати початкові дані.

Один з популярних прикладів тунелювання – це використання віртуальної приватної мережі (Virtual Private Network, VPN). VPN створює тунель між комп'ютером користувача і сервером VPN, який може бути розташований в іншому місці. Внутрішні дані, які користувач передає через VPN, упаковуються в пакети і передаються через загальнодоступну мережу, таку як Інтернет, за допомогою зовнішнього протоколу (наприклад, IP-протокол). На сервері VPN пакети розпаковуються, і дані користувача доставляються до кінцевого пункту призначення.

Тунелювання в сучасних комп'ютерних мережах є потужним інструментом, який дозволяє розширити можливості мережі і забезпечити безпеку та приватність передачі даних. Воно широко використовується в різних сферах, таких як корпоративні мережі, хмарні обчислення та забезпечення доступу до ресурсів з віддаленої локації.

Одним з найбільш популярних сервісів для побудови таких тунелів є ngrok. Однак через значні обмеження функціоналу у безкоштовній версії, в проекті буде використовуватися сервіс diode, який є безкоштовним аналогом сервісу ngrok, а також підтримує необхідну функціональність. Це безкоштовне, компромісне та тимчасове рішення у зв'язку з відсутністю виділеної IP адреси для репозитарію методичної роботи.

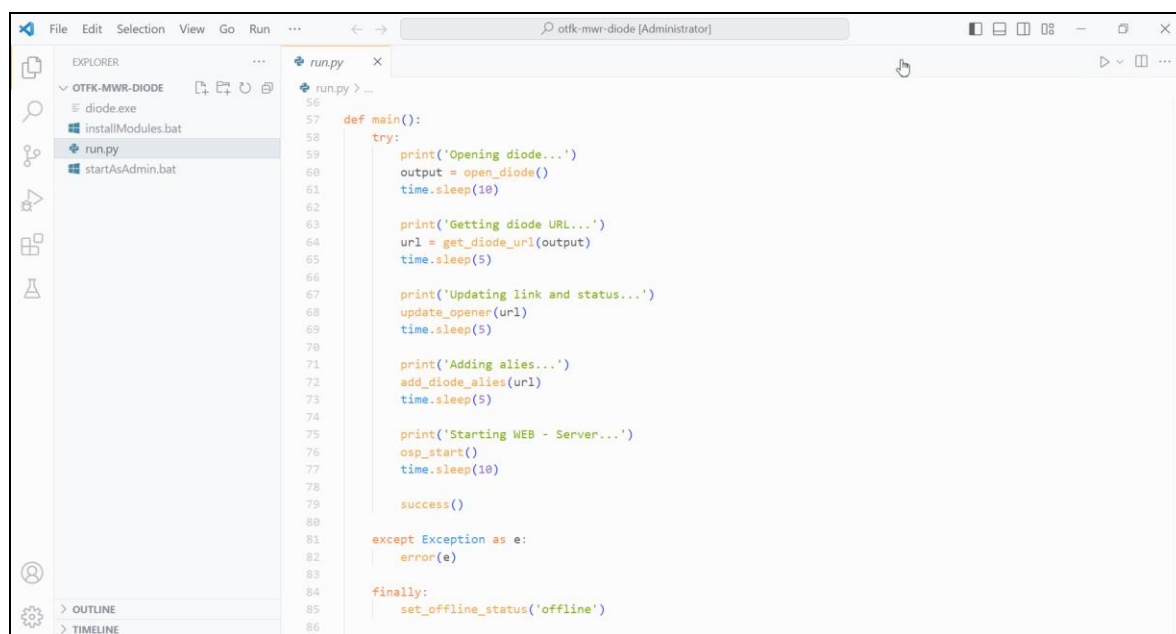
					КГ 06.16.001.00 ДП ПЗ	Арк.
						40
Ізм.	Лист	№ докум.	Підпис	Дата		

В разі наявності у комп'ютера, на якому планується розмістити репозитарій, постійної виділеної («білої») IP-адреси, необхідність у використанні сервісів тунелювання, таких як ngrok чи Diode, відпадає з тієї причини, що постійну IP-адресу можливо було б використовувати у програмному коді замість створення для кожної сесії використання динамічного посилання.

Сервіс Diode використовує NAT-тунелювання, яке створює таблицю перенаправлень портів.

Процес перенаправлення мережного трафіку з одного порту на інший порт називається перенаправленням портів або тунелюванням. Переадресація портів дає багато переваг. Тобто її можна використовувати для безпечного обміну даними між двома системами. Його також можна використовувати для зв'язку, який неможливий у випадках, коли порт заблокований, але до нього потрібен доступ локально або віддалено.

Для роботи сервісу тунелювання Diode напишемо Python-скрипт у процедурному стилі. В функції main викликаються процедури, кожна з них підписана.



```
def main():
    try:
        print('Opening diode...')
        output = open_diode()
        time.sleep(10)

        print('Getting diode URL...')
        url = get_diode_url(output)
        time.sleep(5)

        print('Updating link and status...')
        update_opener(url)
        time.sleep(5)

        print('Adding alies...')
        add_diode_alies(url)
        time.sleep(5)

        print('Starting WEB - Server...')
        osp_start()
        time.sleep(10)

        success()

    except Exception as e:
        error(e)

    finally:
        set_offline_status('offline')
```

Рисунок 13 – Python-скрипт для роботи diode

Функція `open_diode` забезпечує запуск сервісу тунелювання `diode`, функція `get_diode_url` забезпечує формування посилання для доступу к комп'ютеру, на якому буде розміщений репозитарій; функція `update_opener` формує підсумкове посилання для доступу к інформації, що міститься у репозитарії, засобами веб-браузера шляхом додавання сформованого сервісом `diode` посилання до основного URL репозитарію, а саме: <https://otfk-mwr.tk>. Інші функції додають шляхи доступу до файлового менеджера репозитарію, запускають сервер `OpenServer`, та видають повідомлення користувачеві про успішний старт сервісу (функція `success`) або ж про наявність помилок (функція `error`).

Тобто – скрипт запускає сервіс тунелювання `diode`, отримує посилання на ПК, де розміщено репозитарій методичної роботи, додає це посилання до URL-адреси доступу до репозитарію через веб-браузер, визначає шлях доступу до PHP-додатку файлового менеджера (виконаного за технологією MVC – «модель – представлення – контролер»), запускає PHP-скрипт на виконання та видає повідомлення – чи був цей запуск успішним (викликається функція `success`) або ж ні (викликається функція `error`).

Далі розглянемо процес створення MVC-додатку, який буде розміщено на хостингу.

2.3.2 Розробка MVC-додатку для веб-хостингу

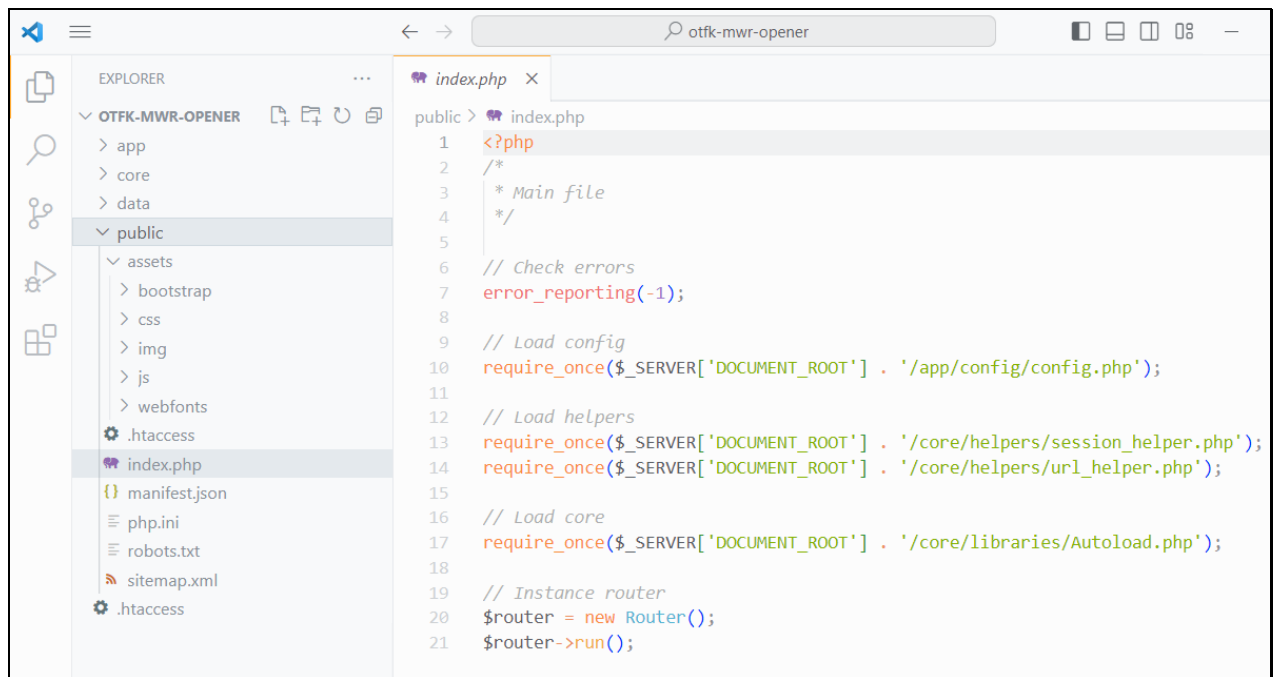
Для створення MVC-додатку за допомогою `Visual Studio Code` створюємо папку «`app`», яка містить папки «`config`» – зберігає файли конфігурації зі змінними середовища; «`controlles`» – зберігає власне файли зі змінними середовища; «`models`» – зберігає класи моделей, які містять SQL-запити на отримання, видалення, редагування інформації для кожної

					КГ 06.16.001.00 ДП ПЗ	Арк.
						42
Ізм.	Лист	№ докум.	Підпис	Дата		

таблиці БД; «views» – зберігає файли з HTML розміткою графічного інтерфейсу.

Створюємо папку «core» з самописним фреймворком додатком, яка містить папки «helpers» – зберігає файли з помічниками, наприклад, для генерації повідомлень або редіректів; «libraries» – зберігає класи з MVC-логікою для роботи файлового менеджера; «utils» – пуста папка, яку планується використовувати далі.

Створюємо папку «data», яка містить файл data.dat для зберігання файлів та папку «public», яка містить файл index.php – який є точкою входу у додаток (рис. 14).



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders 'app', 'core', 'data', and 'public'. The 'public' folder is expanded, showing subfolders 'assets' (with 'bootstrap', 'css', 'img', 'js', 'webfonts') and files '.htaccess', 'index.php', 'manifest.json', 'php.ini', 'robots.txt', and 'sitemap.xml'. The code editor shows the content of 'index.php' in the 'public' directory. The code is as follows:

```
1 <?php
2 /*
3  * Main file
4  */
5
6 // Check errors
7 error_reporting(-1);
8
9 // Load config
10 require_once($_SERVER['DOCUMENT_ROOT'] . '/app/config/config.php');
11
12 // Load helpers
13 require_once($_SERVER['DOCUMENT_ROOT'] . '/core/helpers/session_helper.php');
14 require_once($_SERVER['DOCUMENT_ROOT'] . '/core/helpers/url_helper.php');
15
16 // Load core
17 require_once($_SERVER['DOCUMENT_ROOT'] . '/core/libraries/autoload.php');
18
19 // Instance router
20 $router = new Router();
21 $router->run();
```

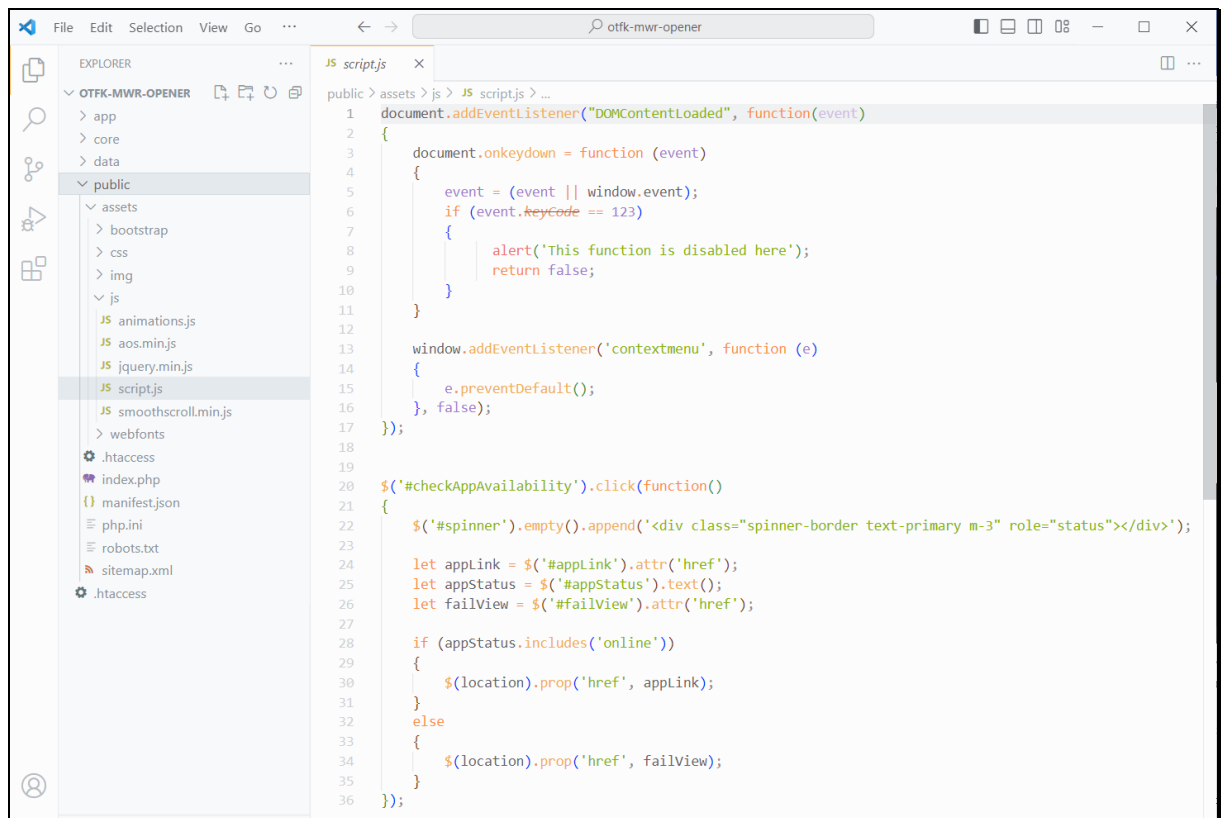
Рисунок 14 – Файл index.php

У сучасних веб-додатках прийнято використовувати концепцію єдиної точки входу. Ця концепція зводиться до того, що всі запити до серверних додатків переадресовуються на один файл параметрів, який, виходячи із запиту, координує подальшу поведінку скрипту.

Такий підхід дає величезні переваги, як на етапі створення, так і на етапі підтримки проекту, так як кардинально зменшується надлишковість коду, а для додатків, маніпулюючих динамічним контентом, єдина точка входу – це єдине правильне рішення.

Концепція єдиної точки входу в реалізацію зводиться до того, що необхідно упорядкувати веб-сервер, та перенаправляти всі звернені до нього запити до файлу `index.php`, який буде єдиною точкою входу.

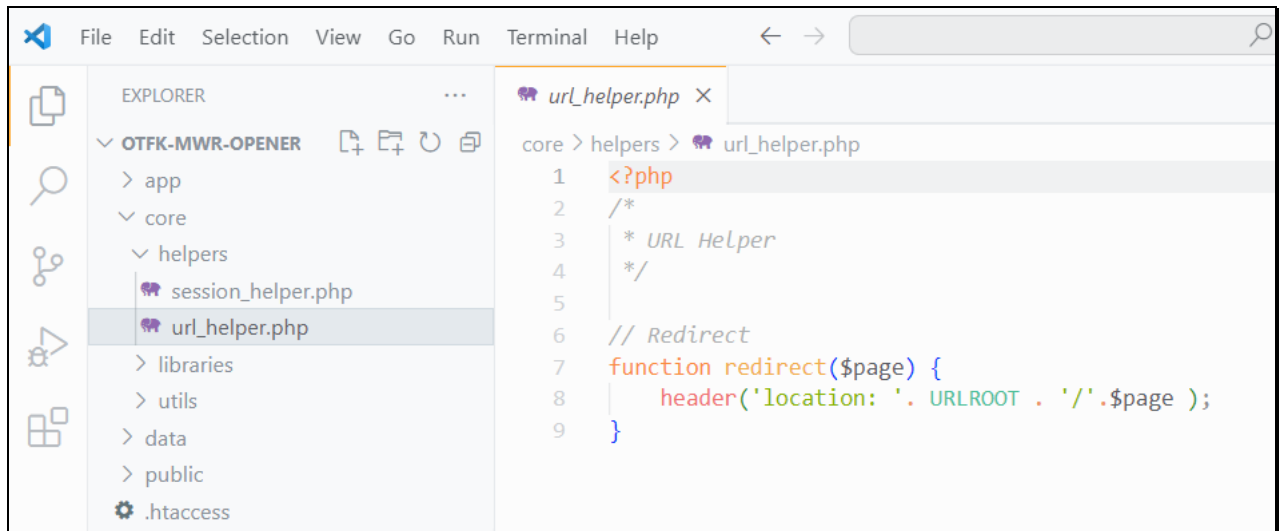
Папка «public» містить папку «assets», в якій зберігаються CSS-файли стилів та JS-файли скриптів (рис. 15).



```
1 document.addEventListener("DOMContentLoaded", function(event)
2
3     document.onkeydown = function (event)
4     {
5         event = (event || window.event);
6         if (event.keyCode == 123)
7         {
8             alert('This function is disabled here');
9             return false;
10        }
11    }
12
13    window.addEventListener('contextmenu', function (e)
14    {
15        e.preventDefault();
16    }, false);
17
18
19
20    $('#checkAppAvailability').click(function()
21    {
22        $('#spinner').empty().append('<div class="spinner-border text-primary m-3" role="status"></div>');
23
24        let appLink = $('#appLink').attr('href');
25        let appStatus = $('#appStatus').text();
26        let failView = $('#failView').attr('href');
27
28        if (appStatus.includes('online'))
29        {
30            $(location).prop('href', appLink);
31        }
32        else
33        {
34            $(location).prop('href', failView);
35        }
36    });
```

Рисунок 15 – Файл `script.js`

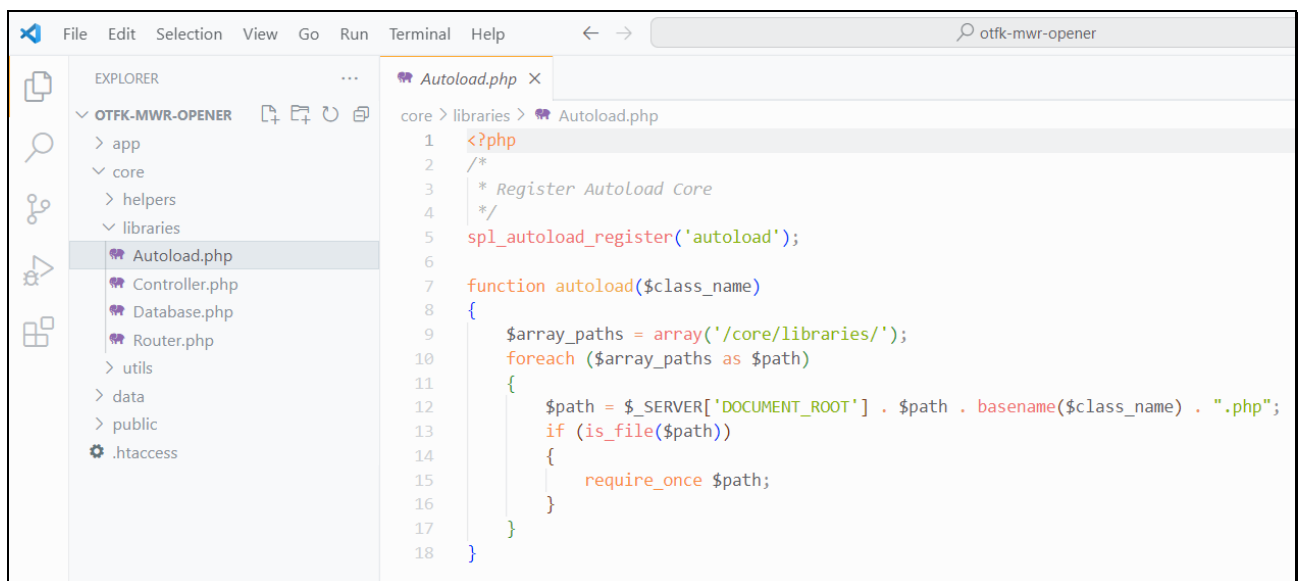
Папка «data» зберігає DAT-файл з посиланням на сервіс diode (рис. 16). Папка «core» зберігає файли «`session_helper.php`» та «`url_helper.php`» (рис. 17 та рис. 18). Це так звані «файли-помічники» або хелпери.



```
1 <?php
2 /*
3  * URL Helper
4  */
5
6 // Redirect
7 function redirect($page) {
8     header('location: '. URLROOT . '/' . $page );
9 }
```

Рисунок 18 – Файл url_helper.php

Папка «libraries» містить класи з MVC-логікою. Клас «Autoload.php» відповідає за реєстрацію класів – тобто всього простору імен класів.



```
1 <?php
2 /*
3  * Register Autoload Core
4  */
5 spl_autoload_register('autoload');
6
7 function autoload($class_name)
8 {
9     $array_paths = array('/core/libraries/');
10    foreach ($array_paths as $path)
11    {
12        $path = $_SERVER['DOCUMENT_ROOT'] . $path . basename($class_name) . ".php";
13        if (is_file($path))
14        {
15            require_once $path;
16        }
17    }
18 }
```

Рисунок 19 – Файл Autoload.php

Клас «Controller.php» відповідає за логіку зв'язування моделі та її представлення (рис. 20). MVC (Model-View-Controller) – архітектурний шаблон проектування програмного забезпечення, який розділяє додаток на три компоненти: модель (Model), представлення (View) і контролер (Controller) (рис. 21).

```

1 <?php
2 /*
3  * Base Controller Class
4  * Loads the models and views
5  */
6 class Controller {
7     // Load model
8     public function model($model) {
9         // Require model file
10        require_once $_SERVER['DOCUMENT_ROOT'] . '/app/models/' . $model . '.php';
11
12        // Instantiate model
13        return new $model();
14    }
15
16    // Load view
17    public function view($view, $data = []) {
18        // Check for view file
19        if(file_exists($_SERVER['DOCUMENT_ROOT'] . '/app/views/' . $view . 'View.php')) {
20            require_once $_SERVER['DOCUMENT_ROOT'] . '/app/views/' . $view . 'View.php';
21        } else {
22            // View does not exist
23            die('View does not exist');
24        }
25    }
26 }

```

Рисунок 20 – Файл Controller.php

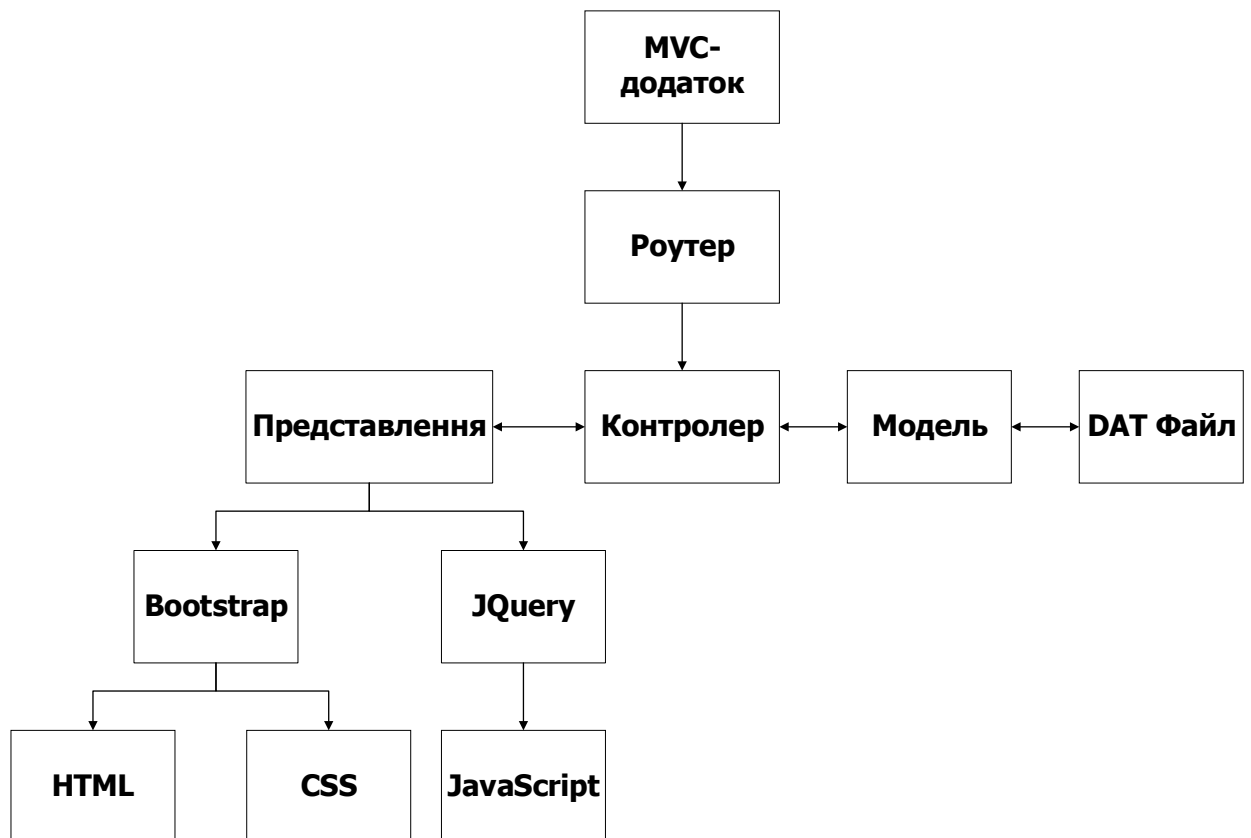


Рисунок 21 – Модульна структура MVC-додатку

Основна мета використання технології MVC – це розділити логіку додатку від його представлення, що полегшує розробку, тестування та підтримку коду в подальшому.

Основний принцип зв'язування моделі та представлення в MVC-додатку полягає в тому, що модель не повинна напряму залежати від представлення, а контролер відповідає за координацію комунікації між представленням та моделлю.

Нижче наведено загальну логіку зв'язування моделі та її представлення в MVC-додатку:

1. Модель (Model) представляє бізнес-логіку та дані додатку. Вона повинна бути незалежною від будь-якого конкретного представлення. Модель може надавати методи для отримання та оновлення даних, а також виконання керуючої логіки додатку.

2. Представлення (View) відображає дані моделі користувачу і обробляє користувацький ввід. Воно не повинно містити керуючої логіки або ж безпосередньо взаємодіяти з моделлю.

Представлення може сприймати події користувача (наприклад, натискання кнопки) і передавати їх контролеру для подальшої обробки.

3. Контролер (Controller) обробляє події від представлення і взаємодіє з моделлю відповідно до логіки керування додатком. Він отримує вхідні дані від представлення, виконує необхідні дії і оновлює модель.

Контролер також може оновлювати представлення, передаючи йому відповідні дані з моделей після їх змін.

Загальний порядок взаємодії між цими компонентами може виглядати таким чином:

1. Подія відбувається в представленні, наприклад, користувач натискає кнопку.

					КГ 06.16.001.00 ДП ПЗ	Арк.
						48
Ізм.	Лист	№ докум.	Підпис	Дата		

2. Представлення повідомляє про цю подію контролеру.
3. Контролер отримує вхідні дані та виконує необхідні дії, можливо, оновлює модель.
4. Якщо модель була змінена, контролер оновлює представлення, передаючи йому відповідні дані з моделі.
5. Представлення оновлює свій інтерфейс залежно від нових даних, які отримано з моделі.

Ця логіка дозволяє забезпечити розділення сфери відповідальності між моделлю, представленням і контролером, що полегшує розробку, тестування та підтримку додатку. Кожен компонент може бути реалізований окремо і замінений без впливу на решту системи, що сприяє модульності та гнучкості програмного забезпечення.

Клас «Router.php» відповідає за відстеження URL-адрес (рис. 22). Клас «Database.php» являє собою адаптер бази даних – це компонент програмного забезпечення, який забезпечує з'єднання і взаємодію між додатком і конкретною базою даних. Він служить посередником між додатком і базою даних, забезпечуючи прозорий доступ до даних і виконання операцій з базою даних (рис. 23).

Адаптер бази даних надає набір методів і функцій, які дозволяють використовувати такі операції, як зчитування, запис, оновлення та видалення даних у базі даних. Він забезпечує зручний інтерфейс для роботи з базою даних, що прикриває від кінцевого користувача складність взаємодії з конкретною системою управління базами даних (СУБД).

Адаптер бази даних може бути специфічним для конкретного СУБД або узагальненим для роботи з різними типами баз даних. Зазвичай він надає функції для встановлення та розриву з'єднань із базою даних, виконання запитів, обробки результатів запитів, транзакцій та інших операцій.

					КГ 06.16.001.00 ДП ПЗ	Арк.
						49
Ізм.	Лист	№ докум.	Підпис	Дата		

Папка «Views» зберігається в папці «App» та містить представлення для кожної зі сторінок, а також папку «template», яка містить в собі шаблонізатор. Це означає, що тут зберігаються елементи макету, які використовуються у всіх веб-сторінках, наприклад, футер, навігація та підключення бібліотек та скриптів.

```
Router.php Database.php
core > libraries > Router.php
1 <?php
2 /*
3  * Base Router class
4  * Loads the controllers
5  */
6 class Router
7 {
8     private $routes;
9
10    public function __construct()
11    {
12        // Set Routes
13        $this->routes = include_once('../app/config/routes.php');
14    }
15
16    // Get URI
17    private function getURI()
18    {
19        if (empty($_SERVER['REQUEST_URI']))
20        {
21            return trim($_SERVER['REQUEST_URI'], '/');
22        }
23    }
24
25    // Run Router
26    public function run()
27    {
28        $uri = $this->getURI();
29
30        $success = false;
31
32        foreach ($this->routes as $uriPattern => $path)
33        {
34            if (preg_match("~^$uriPattern$~", $uri))
35            {
36                $internalRoute = preg_replace("~^$uriPattern$~", $path, $uri);
37                $segments = explode('/', $internalRoute);
38                $controllerName = array_shift($segments). 'Controller';
39                $controllerName = ucfirst($controllerName);
40                $actionName = ucfirst(array_shift($segments));
41                $parameters = $segments;
42                $controllerFile = $_SERVER['DOCUMENT_ROOT'] . '/app/controllers/' . basename($controllerName) . '.php';
43                if (file_exists($controllerFile))
44                {
45                    require_once $controllerFile;
46                }
47                else
48                {
49                    break;
50                }
51                $controllerObject = new $controllerName;
52                $result = call_user_func_array(array($controllerObject, $actionName), $parameters);
53                if ($result != null)
54                {
55                    break;
56                }
57                else
58                {
59                    $success = true;
60                }
61            }
62        }
63
64        if ($success == false)
65        {
66            require_once $_SERVER['DOCUMENT_ROOT'] . '/app/views/ErrorView.php';
67        }
68    }
69 }
70 }
```

Рисунок 22 – Файл Router.php

```

Database.php X
core > libraries > Database.php
1 <?php
2 /*
3  * PDO Database Class
4  * Connect to database
5  * Create prepared statements
6  * Bind values
7  * Return rows and results
8  */
9 class Database {
10     private $host = DB_HOST;
11     private $user = DB_USER;
12     private $pass = DB_PASS;
13     private $dbname = DB_NAME;
14
15     private $dbh;
16     private $stmt;
17     private $error;
18
19     public function __construct() {
20         $temp = "mysql:host=". $this->host . ";";
21         $dsn = "mysql:host=". $this->host . ";dbname=". $this->dbname;
22         $options = array(
23             PDO::ATTR_PERSISTENT => true,
24             PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION
25         );
26
27         try {
28             $this->dbh = new PDO($dsn, $this->user, $this->pass, $options);
29         }
30         catch(Exception $e) {
31             $this->error = $e->getMessage();
32             echo $this->error;
33
34             $this->dbh = new PDO($temp, $this->user, $this->pass, $options);
35
36             $sql = "CREATE DATABASE IF NOT EXISTS " . $this->dbname . " ";
37             $this->stmt = $this->dbh->prepare($sql);
38             $this->stmt->execute();
39
40             $this->dbh = new PDO($dsn, $this->user, $this->pass, $options);
41             $dump = file_get_contents("db/hc.sql");
42             $this->dbh->exec($dump);
43         }
44     }
45
46     // Prepare statement with query
47     public function query($sql){
48         $this->stmt = $this->dbh->prepare($sql);
49     }
50
51     // Bind values
52     public function bind($param, $value, $type = null){
53         if(is_null($type)){
54             switch(true){
55                 case is_int($value):
56                     $type = PDO::PARAM_INT;
57                     break;
58                 case is_bool($value):
59                     $type = PDO::PARAM_BOOL;
60                     break;
61                 case is_null($value):
62                     $type = PDO::PARAM_NULL;
63                     break;
64                 default:
65                     $type = PDO::PARAM_STR;
66             }
67         }
68         $this->stmt->bindValue($param, $value, $type);
69     }
70
71     // Execute the prepared statement
72     public function execute() {
73         return $this->stmt->execute();
74     }
75
76     // Get result set as array of objects
77     public function resultSet() {
78         $this->execute();
79         return $this->stmt->fetchAll(PDO::FETCH_OBJ);
80     }
81
82     // Get single record as object
83     public function single() {
84         $this->execute();
85         return $this->stmt->fetch(PDO::FETCH_OBJ);
86     }
87
88     // Get row count
89     public function rowCount() {
90         return $this->stmt->rowCount();
91     }
92
93 }

```

Рисунок 23 – Файл Database.php

```

scripts.php
1 <script src="c:\php echo ASSETSROOT; ?/assets/js/jquery.min.js"></script>
2 <script src="c:\php echo ASSETSROOT; ?/assets/js/bootstrap.min.js"></script>
3 <script src="c:\php echo ASSETSROOT; ?/assets/js/aos.min.js"></script>
4 <script src="c:\php echo ASSETSROOT; ?/assets/js/animations.js"></script>
5 <script src="c:\php echo ASSETSROOT; ?/assets/js/script.js"></script>
6 <script src="c:\php echo ASSETSROOT; ?/assets/js/smoothscroll.min.js"></script>
7
8 </scripts>

stylesheets.php
1 <link rel="stylesheet" href="c:\php echo ASSETSROOT; ?/assets/css/bootstrap.min.css">
2 <link rel="stylesheet" href="c:\php echo ASSETSROOT; ?/assets/css/Fontawesome.min.css">
3 <link rel="stylesheet" href="c:\php echo ASSETSROOT; ?/assets/css/Fontawesome.min.css">
4 <link rel="stylesheet" href="c:\php echo ASSETSROOT; ?/assets/css/aos.min.css">
5 <link rel="stylesheet" href="c:\php echo ASSETSROOT; ?/assets/css/animate.min.css">
6 <link rel="stylesheet" href="c:\php echo ASSETSROOT; ?/assets/css/holder.css">
7 <link rel="stylesheet" href="c:\php echo ASSETSROOT; ?/assets/css/styles.css">
8

```

Рисунок 24 – Файли Scripts.php та Stylesheets.php

Розглянемо типове представлення для стартової веб-сторінки репозитарію. У файлі HomeView.php прописано підключення шаблонів макету, міститься шапка веб-сторінки (тег «header») та розділи (тег «sections») (рис. 25). Розглянемо те, як працює адаптивна верстка за допомогою адаптивної сітки Bootstrap. Кожна секція (тег «section») містить в собі клас контейнеру (div container) для обмеження області макету. Контейнер містить в собі класи «row» та «column». Вони відповідають за ґрид-систему та адаптивність, так як «row» – це рядок, а «column» – стовпець. Крім того, використовується клас «tabs» – для створення вкладок. Вони роблять додаток більш адаптивним.

```
1 <DOCTYPE html>
2 <html>
3
4 <head>
5 <?php require_once APPROOT . '/views/template/meta.php'; ?>
6 <?php require_once APPROOT . '/views/template/stylesheets.php'; ?>
7 </head>
8
9 <body style="background: url(<?php echo ASSETSROOT; ?>/assets/img/bg.jpg) center / auto;">
10
11 <div class="border rounded border-1 shadow position-absolute top-50 start-50 translate-middle" data-aos="zoom-in">
12 <?php require_once APPROOT . '/views/template/header.php'; ?>
13
14 <main>
15
16 <div class="container">
17 <div class="text-center d-flex flex-column m-3">
18 <ul class="nav nav-pills nav-fill d-flex flex-column m-3" role="tablist">
19 <li class="nav-item" role="presentation" data-bs-hover-animate="pulse"><a class="nav-link active" role="tab" data-bs-toggle="pill" href="#tab-1"><i class="fas fa-globe-europe"></i></a></li>
20 <li class="nav-item" role="presentation" data-bs-hover-animate="pulse"><a class="nav-link" role="tab" data-bs-toggle="pill" href="#tab-2"><i class="fas fa-info-circle"></i></a></li>
21 <li class="nav-item" role="presentation" data-bs-hover-animate="pulse"><a class="nav-link" role="tab" data-bs-toggle="pill" href="#tab-3" target="_parent"><i class="fas fa-users-cog"></i></a></li>
22 </ul>
23 <div class="tab-content d-flex justify-content-center align-items-center align-self-center align-content-center align-self-center m-auto m-3" style="height: 200px; min-width: 170px; max-width: 250px;">
24 <div class="tab-pane fade show active" role="tabpanel" id="tab-1">
25 <a id="checkAppAvailability" class="btn btn-primary m-3" role="button" data-bs-hover-animate="pulse"><i class="fas fa-share-square me-1"></i></a></div>
26 <div class="text-center m-3" id="spinner"></div>
27 </div>
28 <div class="tab-pane fade" role="tabpanel" id="tab-2">
29 <p class="text-break text-white-50 m-1">Цей репозитарій містить в собі методичну роботу циклових комісії ВСП "ОТЭК ДНУ".</p>
30 </div>
31 <div class="tab-pane fade" role="tabpanel" id="tab-3">
32 <a class="btn btn-primary m-3" data-bs-hover-animate="pulse" type="button" href="#panel"><i class="fas fa-share-square me-1"></i></a></div>
33 </div>
34 </div>
35 <div id="appStatus" hidden=<?php echo $data['status']></div>
36 <a class="invisible" id="appLink" href="#<?php echo 'http://'. $data['link']>"></a>
37 <a class="invisible" id="fallView" href="#fall"></a>
38 </div>
39 </main>
40
41 <?php require_once APPROOT . '/views/template/footer.php'; ?>
42 </div>
43
44 <?php require_once APPROOT . '/views/template/scripts.php'; ?>
45 </body>
46 </html>
```

Рисунок 25 – Файл HomeView.php

В папці «Models» зберігаються класи для маніпулювання DAT-файлом. Папка «Controllers» зв'язує між собою модель та представлення, а саме породжує данні з БД та відображає їх у представленні або навпаки відправляє данні з представлення у модель. На кожне представлення (вид) існує свій контролер (рис. 27). На рисунках 28 та 29 представлено типові класи контролеру. В конструкторі класу створюється об'єкт моделі.

Метод index – це метод, який спрацьовує при старті сторінки. Також в ньому викликаються методи моделі. Потім отримані дані додаються до масиву і цей масив завантажується разом із веб-сторінкою.

Файл «config.php», що розташовується в папці «config», зберігає глобальні змінні для веб-додатку, а файл «router.php» – зберігає шляхи URL.

Наприклад, конструкція 'home' => 'home/index' означає, що якщо в url буде http://:otfk-mwr-opener/home, то завантажиться контролер HomeController.php та метод index(), який підвантажує дані з БД та відображає їх у своєму представленні.

```

1  <?php
2  class LinkModel
3  {
4      public function getLink()
5      {
6          $dataFile = '../data/link.dat';
7          $openedFile = file_get_contents($dataFile);
8          return $openedFile;
9      }
10
11     public static function setLink($data)
12     {
13         $dataFile = '../data/link.dat';
14         $openedFile = file_put_contents($dataFile, $data);
15         return true;
16     }
17
18     public function getStatus()
19     {
20         $dataFile = '../data/status.dat';
21         $openedFile = file_get_contents($dataFile);
22         return $openedFile;
23     }
24
25     public static function setStatus($data)
26     {
27         $dataFile = '../data/status.dat';
28         $openedFile = file_put_contents($dataFile, $data);
29         return true;
30     }
31 }

```

Рисунок 26 – Файл LinkModel.php

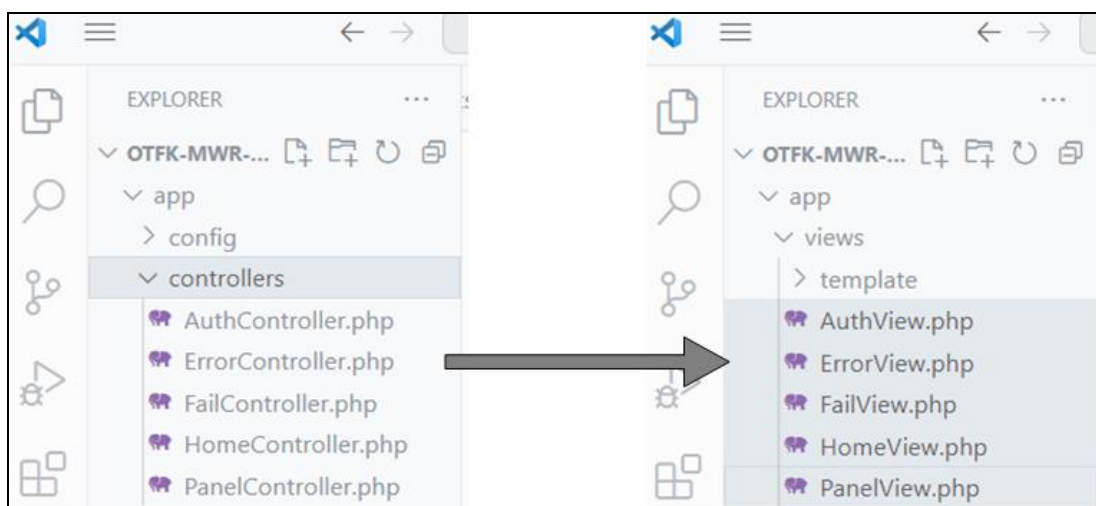


Рисунок 27 – Зв'язок між контролером та представленням

```

1 <?php
2 class AuthController extends Controller
3 {
4     public function __construct()
5     {
6         if(!isLoggedIn())
7         {
8             redirect('panel');
9         }
10
11         $this->userModel = $this->model('UserModel');
12     }
13
14     public function index()
15     {
16         $this->view('auth');
17         flash('message');
18     }
19
20     public function login()
21     {
22         if ($SERVER['REQUEST_METHOD'] == 'POST')
23         {
24             $POST = filter_input_array(INPUT_POST, FILTER_SANITIZE_FULL);
25
26             $data = [
27                 'login' => trim($POST['login']),
28                 'password' => trim($POST['password']),
29             ];
30
31             $loggedInUser = $this->userModel->getUser($data['login'], $data['password']);
32
33             if($loggedInUser)
34             {
35                 $this->createUserSession($loggedInUser);
36                 redirect('panel');
37             }
38             else
39             {
40                 flash('message', 'Неправильні дані');
41                 redirect('auth');
42             }
43         }
44         else
45         {
46             redirect('auth');
47         }
48     }
49 }

```

```

1 <?php
2 class UserModel
3 {
4     public function getUser($login, $password)
5     {
6         $dataLogin = 'admin';
7         $dataPassword = password_hash("54650008", PASSWORD_DEFAULT);
8
9         if(password_verify($password, $dataPassword))
10        {
11            return $dataLogin;
12        }
13        else
14        {
15            return false;
16        }
17    }
18 }

```

Рисунок 28 – Типовий клас контролеру (авторизація)

```

1 <?php
2 // Routes array
3 return array(
4     '' => 'home/index',
5     'home' => 'home/index',
6     'fail' => 'fail/index',
7     'error' => 'error/index',
8     'auth' => 'auth/index',
9     'auth/login' => 'auth/login',
10    'auth/logout' => 'auth/logout',
11    'panel' => 'panel/index',
12    'panel/editLink' => 'panel/editLink',
13    'panel/editStatus' => 'panel/editStatus',
14    'panel/editLinkByUri/(.*)' => 'panel/editLinkByUri/$1',
15    'panel/editStatusByUri/(.*)' => 'panel/editStatusByUri/$1',
16 );

```

```

1 <?php
2 class HomeController extends Controller
3 {
4     public function __construct()
5     {
6         $this->linkModel = $this->model('LinkModel');
7     }
8
9     public function index()
10    {
11        $link = $this->linkModel->getLink();
12        $status = $this->linkModel->getStatus();
13
14        $data = [
15            'link' => $link,
16            'status' => $status,
17        ];
18
19        $this->view('home', $data);
20        flash('message');
21    }
22 }

```

Рисунок 29 – Типовий клас контролеру (використання методу index())

Повна файлова структура створеного MVC-додатку представлена на рисунку 30. Далі розглянемо процес створення файлового менеджера, який буде здійснювати навігацію, редагування та реалізовувати пошук до бази даних репозитарію методичної роботи ВСП «ОТФК ОНТУ».

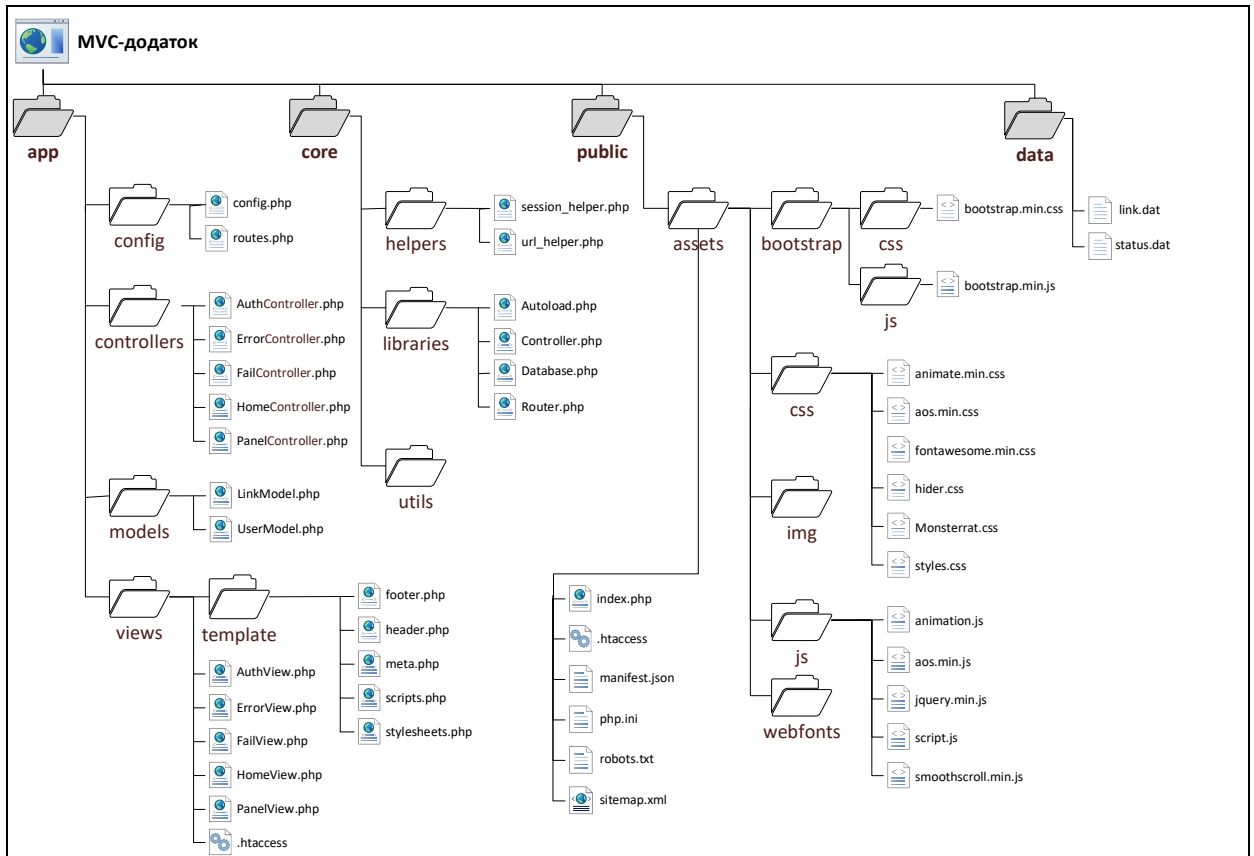


Рисунок 30 – Файлова структура MVC-додатку

2.3.3 Розробка файлового менеджера

За основу файлового менеджера, структура якого представлена на рисунку 31, при розробці репозитарію методичної роботи для ВСП «ОТФК ОНТУ» було взято безкоштовний менеджер «TinyFileManager». TinyFileManager – це простий файловий менеджер, який можна використовувати для керування файлами і каталогами на веб-сервері.

Основні причини використання TinyFileManager в проекті:

1. Керування файлами. TinyFileManager надає інтерфейс для завантаження, перегляду, редагування, переміщення та видалення файлів на веб-сервері. Тобто можна легко організувати файли та папки, створювати нові папки та переміщувати файли між ними.

2. Перегляд файлів. Можна використовувати TinyFileManager для перегляду вмісту файлів прямо в браузері. Це особливо зручно для текстових файлів, таких як HTML, CSS, JavaScript тощо, оскільки їх можна переглядати та редагувати безпосередньо через веб-інтерфейс.

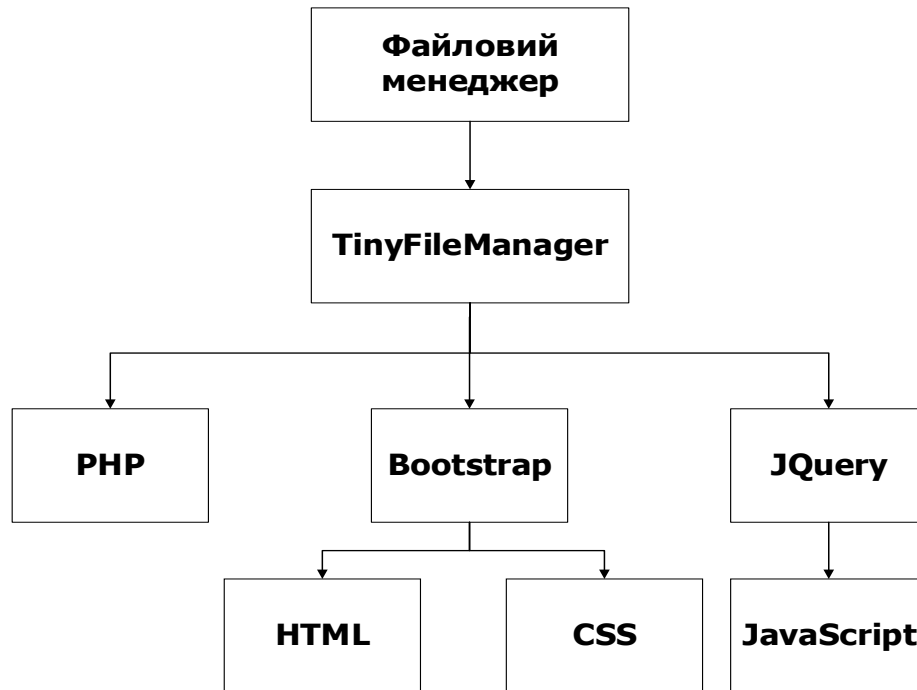


Рисунок 31 – Структура файлового менеджера MVC-додатку

3. Забезпечення безпеки. TinyFileManager можна налаштувати для обмеження доступу до файлів та каталогів. Тобто можна заборонити виконання скриптів у вибраних папках або застосовувати обмеження на завантаження файлів з певними розширеннями. Це допомагає забезпечити безпеку сервера, уникнути несанкціонованого доступу та відправку шкідливих файлів.

4. Просте налаштування та використання. TinyFileManager має простий і зрозумілий інтерфейс, а його налаштування відбувається шляхом зміни конфігураційних параметрів у вихідному коді (рис. 32). Це дозволяє швидко впроваджувати його на веб-сервері та приступати до роботи з файлами.

2.4 Публікація додатку на хостинг

Для того, щоб здійснити публікацію додатку на хостингу, необхідно спочатку пройти процедуру реєстрації домену. Зареєструємо безкоштовний домен Freenom на один рік та вкажемо простір імен CloudFlare для управління DNS (рис. 33).

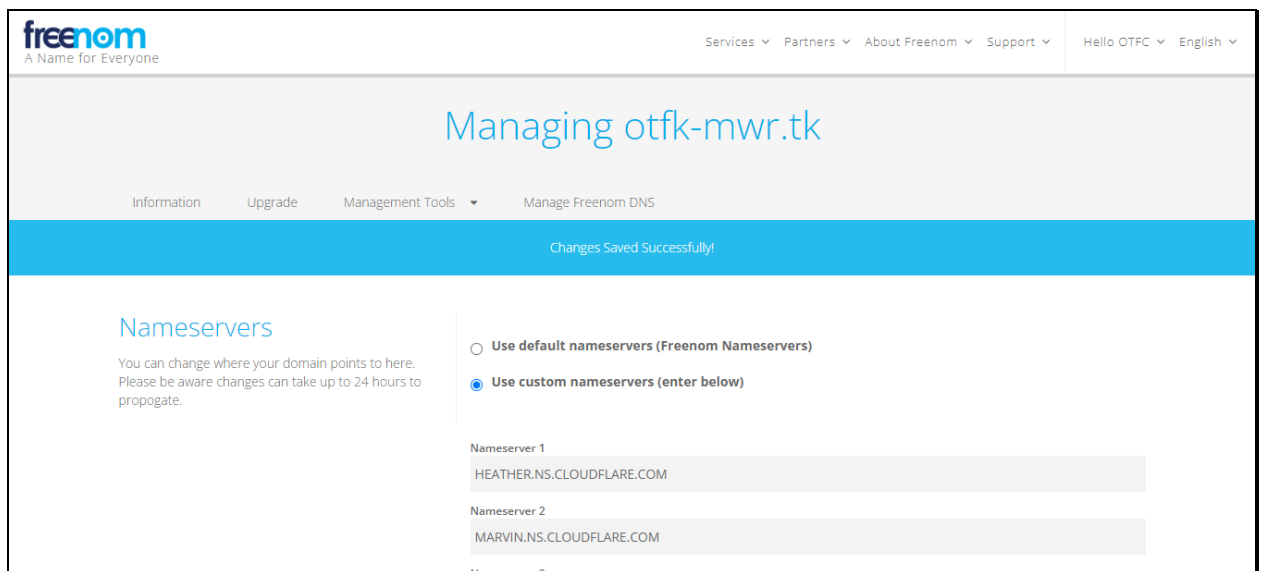


Рисунок 33 – Реєстрація домену для репозитарію

Після цього зареєструємо безкоштовний хостинг на сервісі 000webhost. Завантажимо сайт за допомогою вбудованого файлового менеджера (рис. 34).

Використовуємо сервіс Cloudflare, який представляє собою мережу серверів по всьому світі, щоб підключити веб-додаток і збільшити швидкість його завантаження, а також захистити його від DDoS-атак. Також за допомогою сервісу Cloudflare можна здійснювати менеджмент DNS та сертифікатів SSL. Перебросимо домен третього рівня, який дає 000webhost, на домен першого рівня .tk (рис. 35).

					КГ 06.16.001.00 ДП ПЗ	Арк.
						58
Ізм.	Лист	№ докум.	Підпис	Дата		

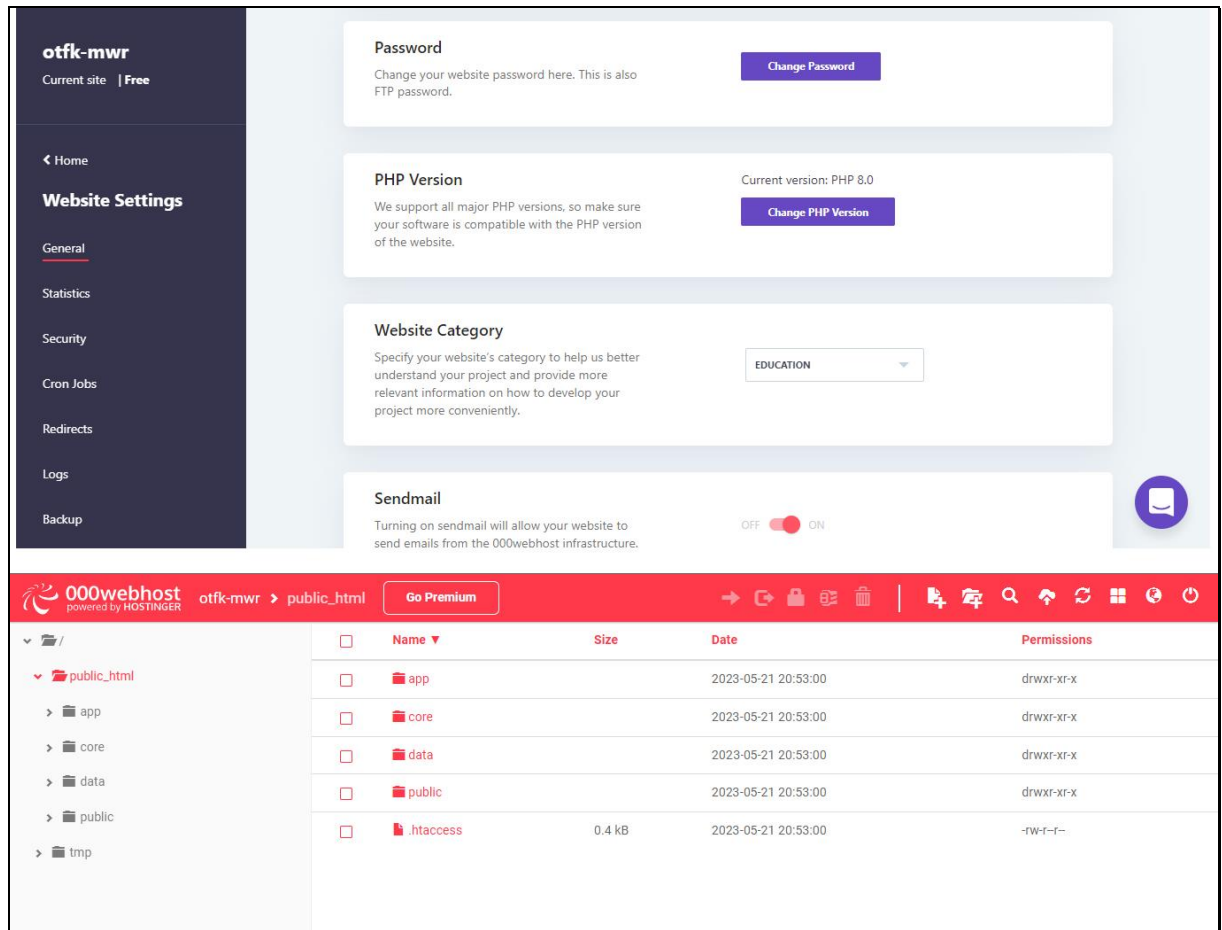


Рисунок 34 – Реєстрація хостингу для репозитарію

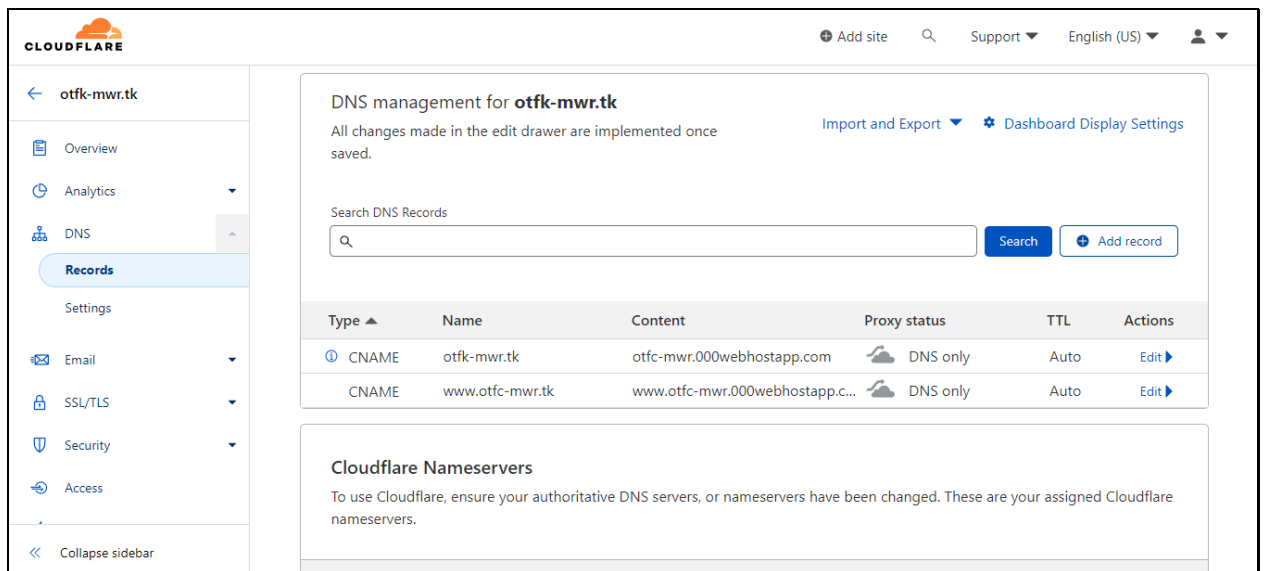


Рисунок 35 – Зміна рівня домену для репозитарію

Для використання функціоналу сервісу diode не потрібно проходити процедуру реєстрації, достатньо завантажити двійковий файл та інсталиувати його (рис. 36).

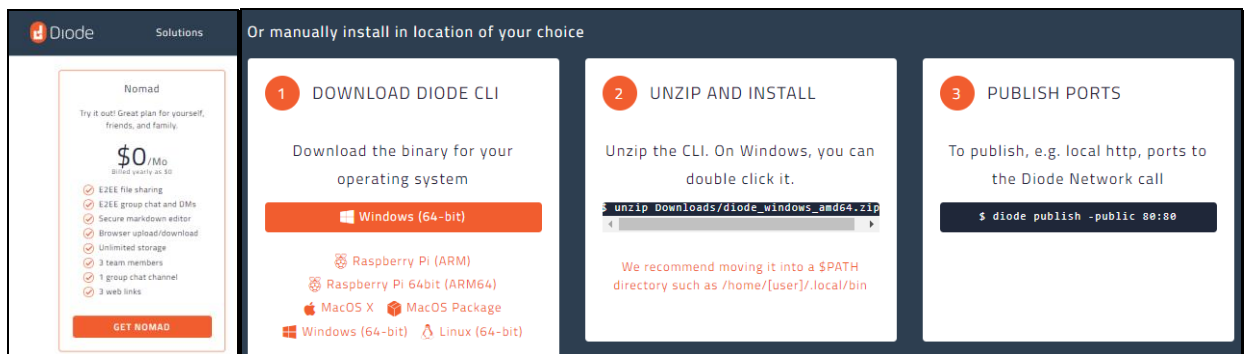


Рисунок 36 – Завантаження diode

2.5 Робота зі створеним репозитарієм

Для початку роботи з репозитарієм в режимі користувача (user) слід на головному вікні веб-додатка натиснути на кнопку «Відкрити репозитарій» (рис. 37).

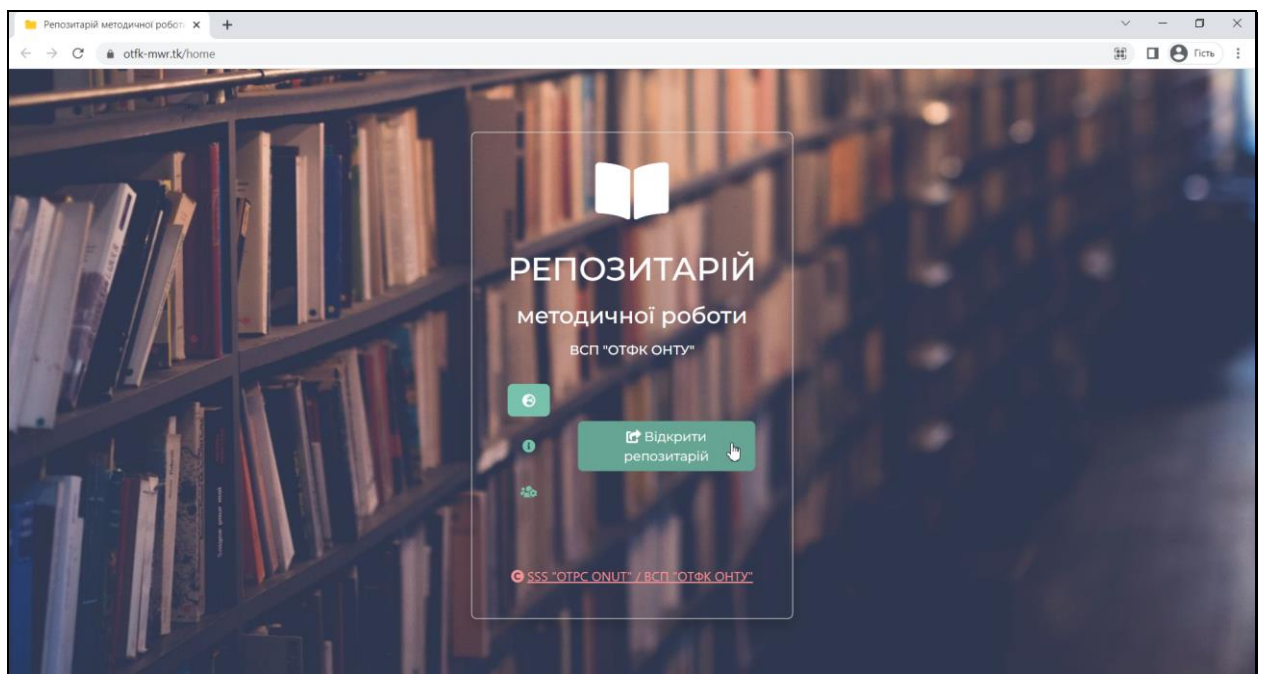


Рисунок 37 – Початок роботи з репозитарієм

					КГ 06.16.001.00 ДП ПЗ	Арк.
						60
Ізм.	Лист	№ докум.	Підпис	Дата		

Далі слід ввести в поле «Користувач» логін користувача user або admin, а також пароль у відповідне поле. Після чого слід натиснути на кнопку «Увійти» (рис. 38).

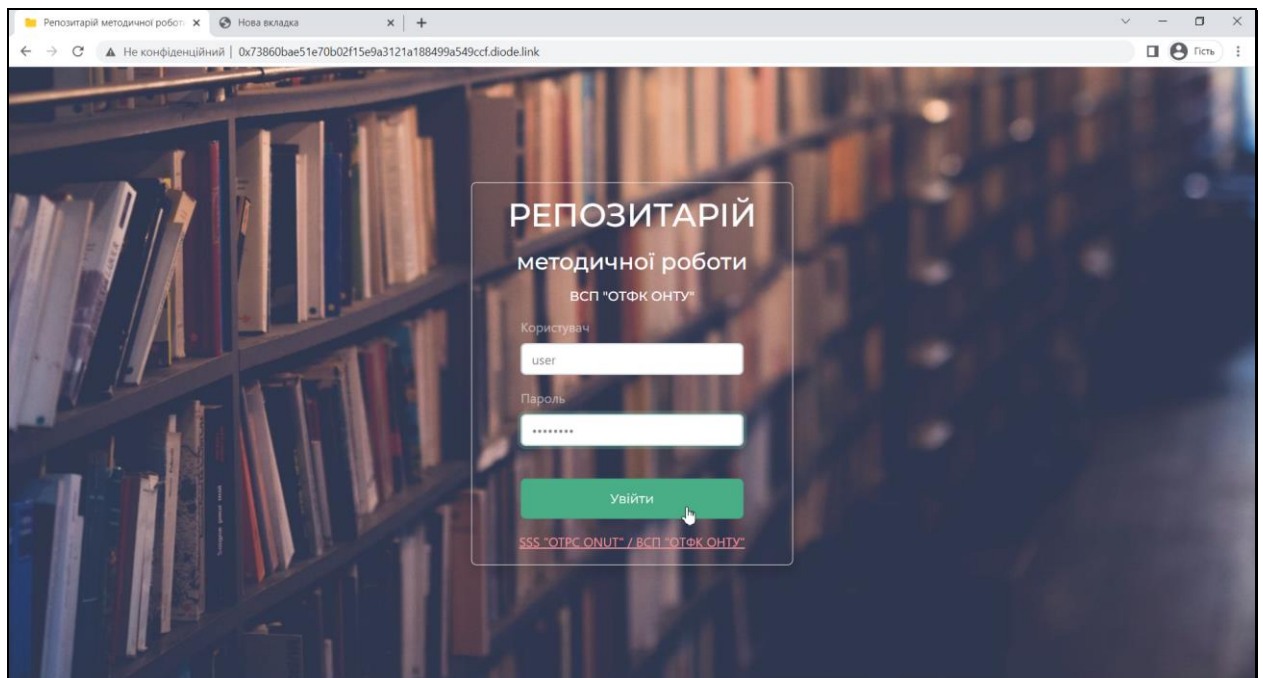


Рисунок 38 – Вхід в репозитарій в режимі користувача (user)

Користувач потрапляє в бібліотеку електронних матеріалів, де всі навчально-методичні розробки згруповані по цикловим комісіям (рис. 39), викладачам (рис. 40) та дисциплінам, які вони вивчають (рис. 41).

Для переходу всередину папки достатньо клікнути по її назві.

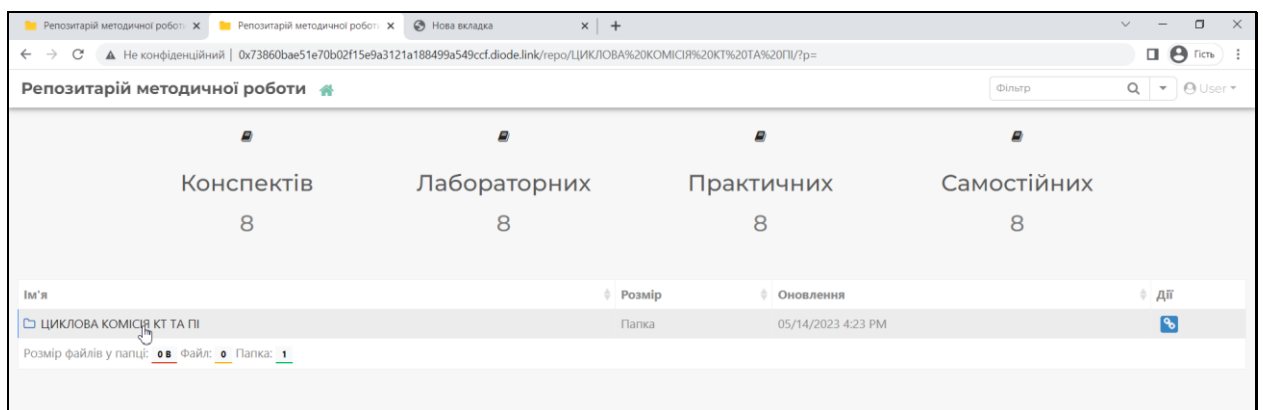


Рисунок 39 – Бібліотека електронних матеріалів

Після цього користувач потрапляє всередину відповідної папки і може відкрити вкладену папку з матеріалами, або ж повернутися на рівень назад, натиснувши на синю стрілочку «назад» (рис. 40).

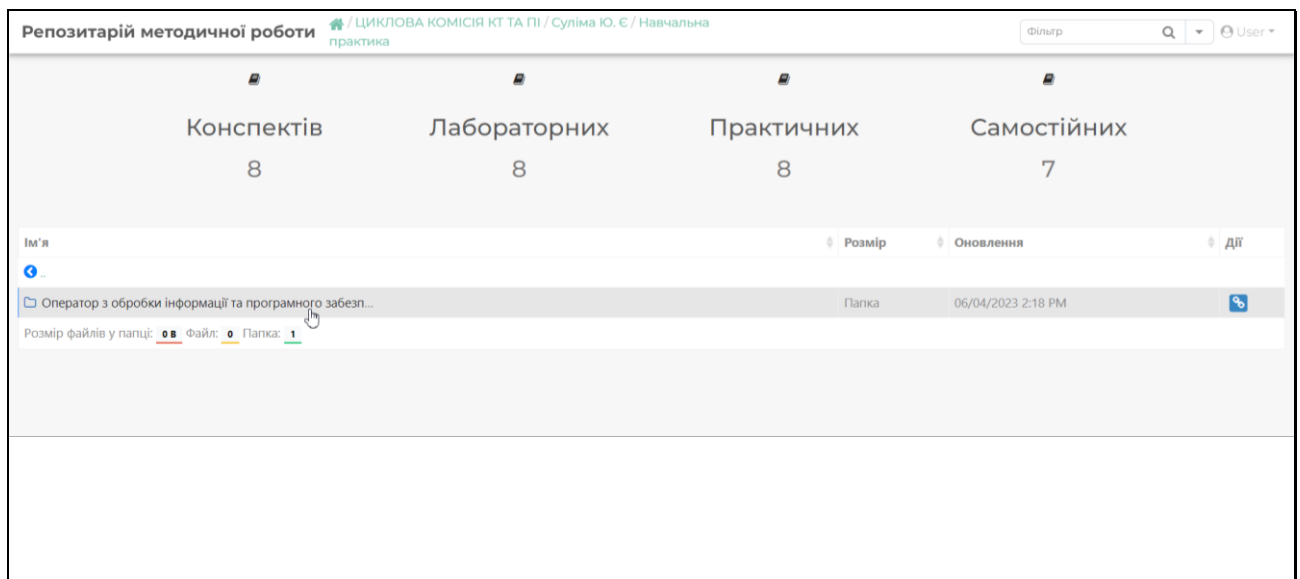
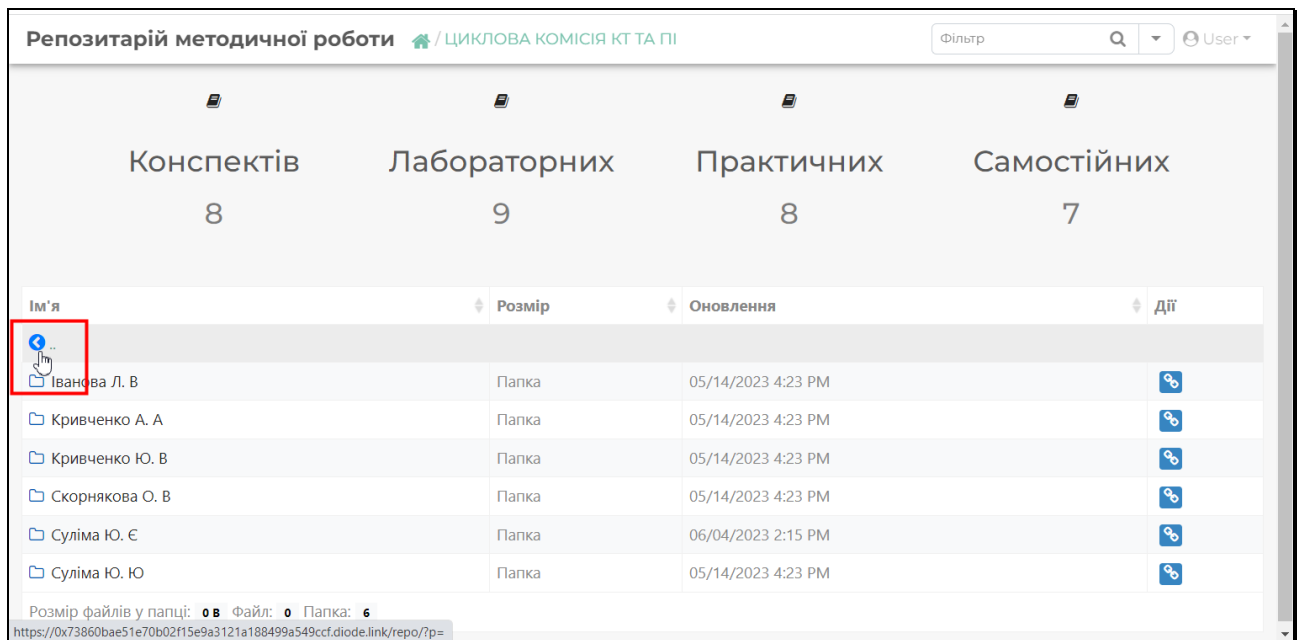


Рисунок 40 – Навігація по матеріалам всередині репозитарію

В папках відповідних дисциплін зберігається їх навчально-методичне забезпечення: конспекти лекцій, інструкції до виконання лабораторних, практичних та самостійних робіт. Користувач може отримати посилання на файл або завантажити його (якщо виконаний вхід

під логіном адміністратора), натиснувши на відповідну іконку та підтвердивши свій вибір у вікні, що з'явиться (рис. 41).

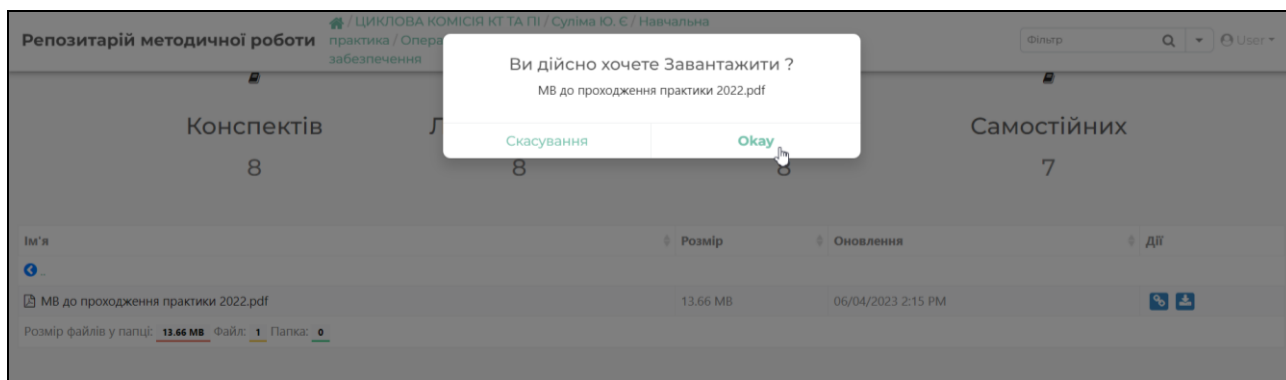
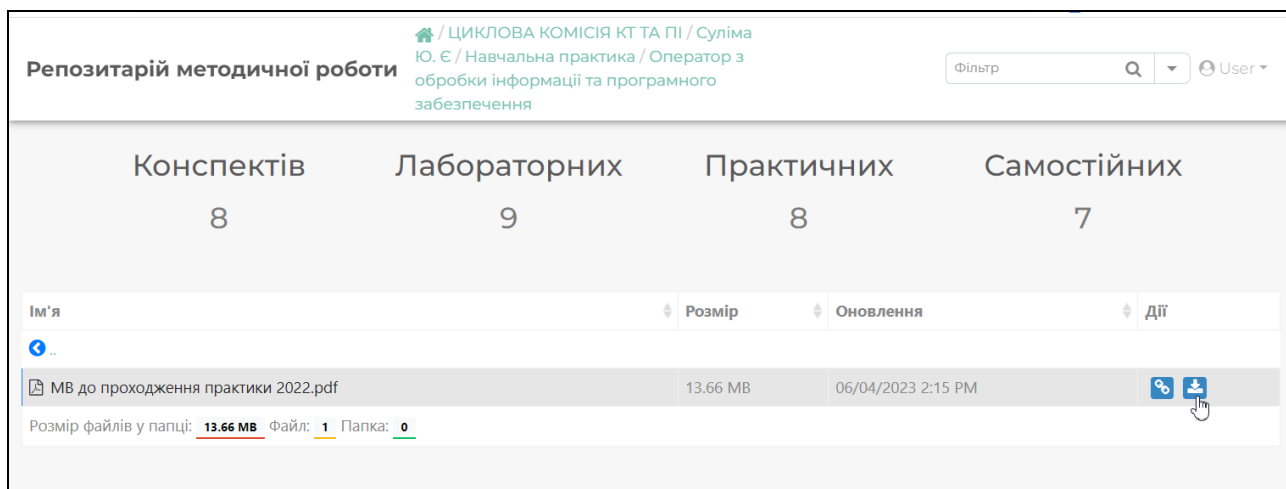
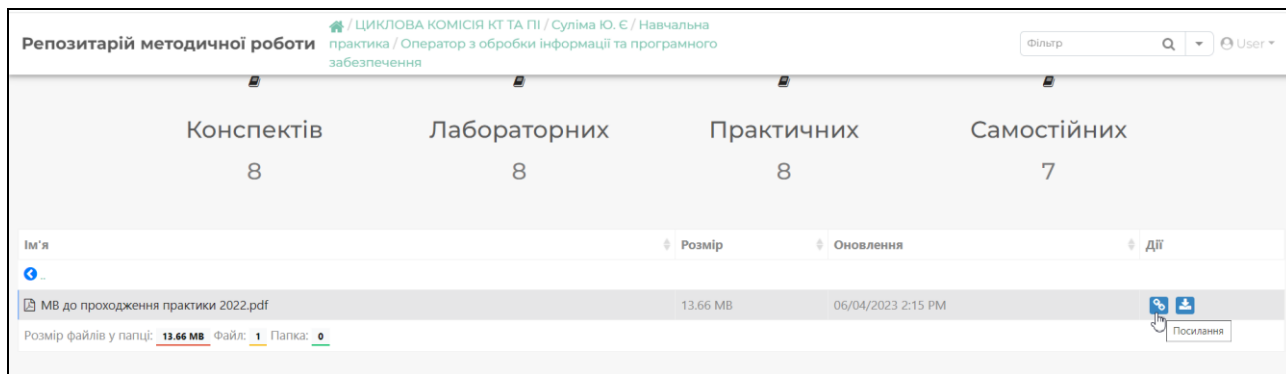


Рисунок 41 – Завантаження файлу користувачем

Натиснувши на назву файлу, користувач матиме змогу переглянути його безпосередньо у вікні репозитарію або відкрити для перегляду в новому вікні браузера (рис. 42).

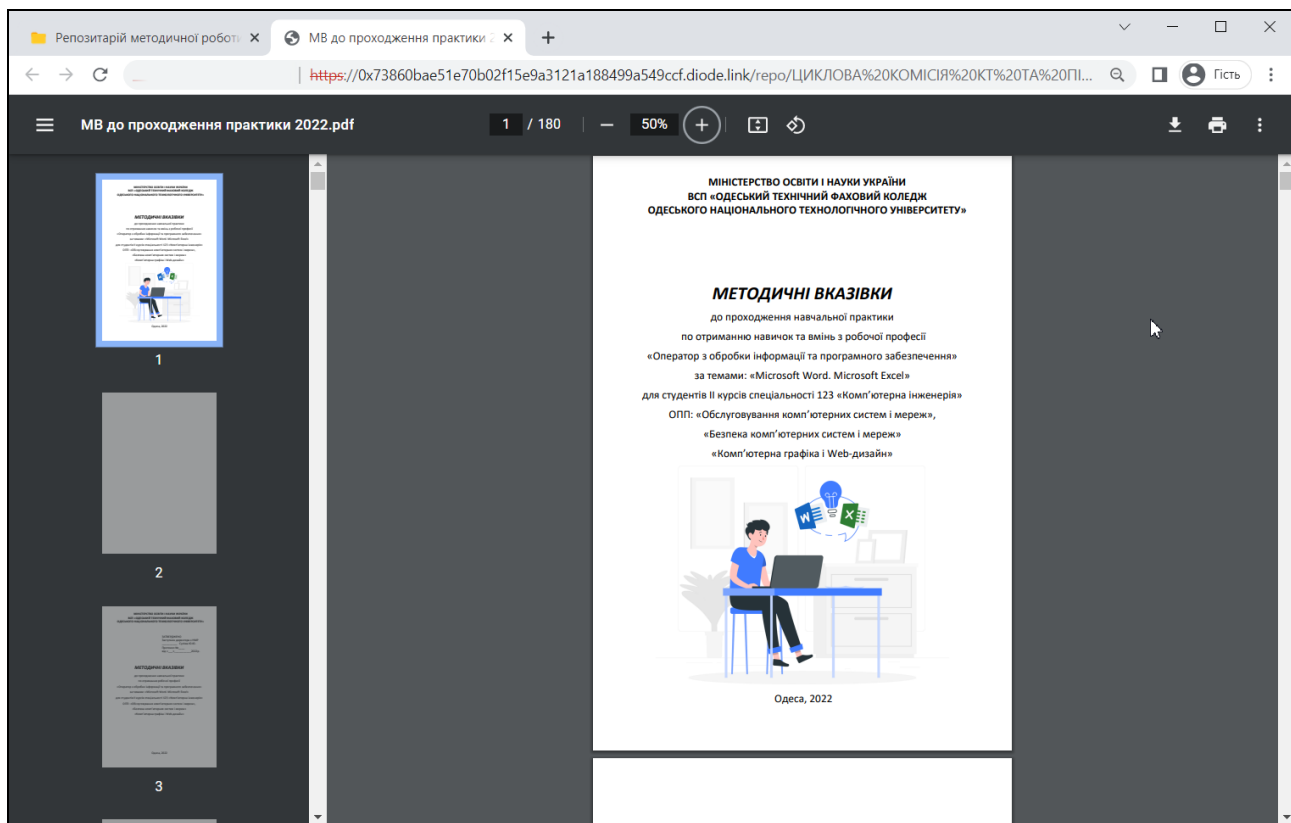
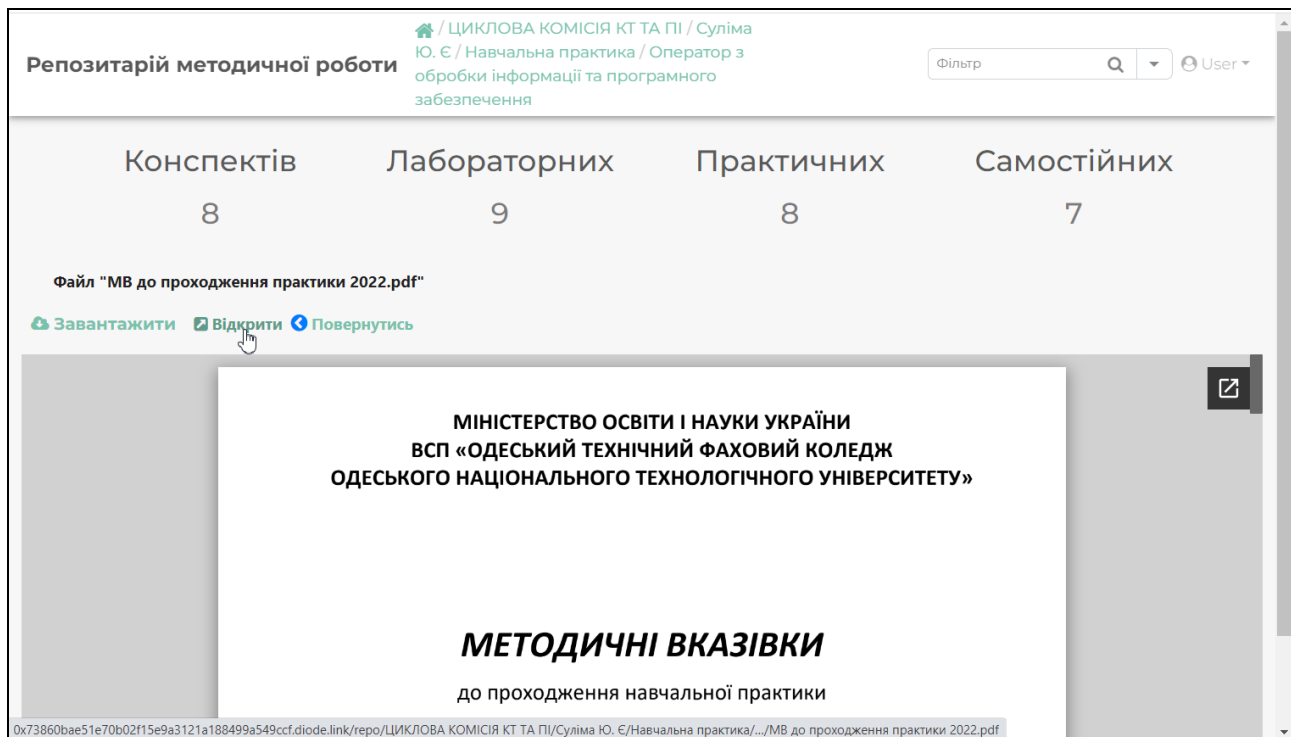


Рисунок 42 – Перегляд файлу користувачем

Додавати, перейменовувати або видаляти папки та файли в репозитарії може лише адміністратор. Для цього треба пройти процедуру авторизації (рис. 43).

					КГ 06.16.001.00 ДП ПЗ	Арк.
						64
Ізм.	Лист	№ докум.	Підпис	Дата		

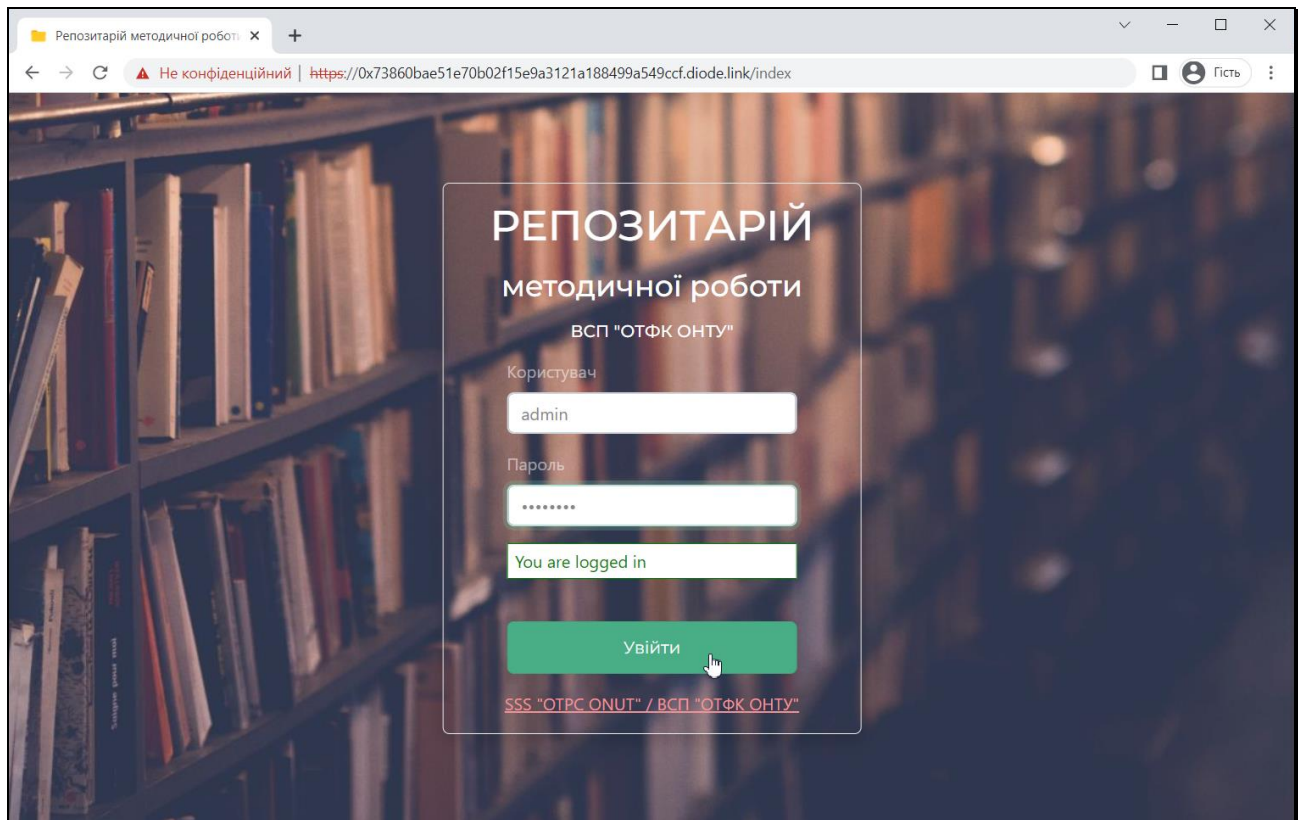
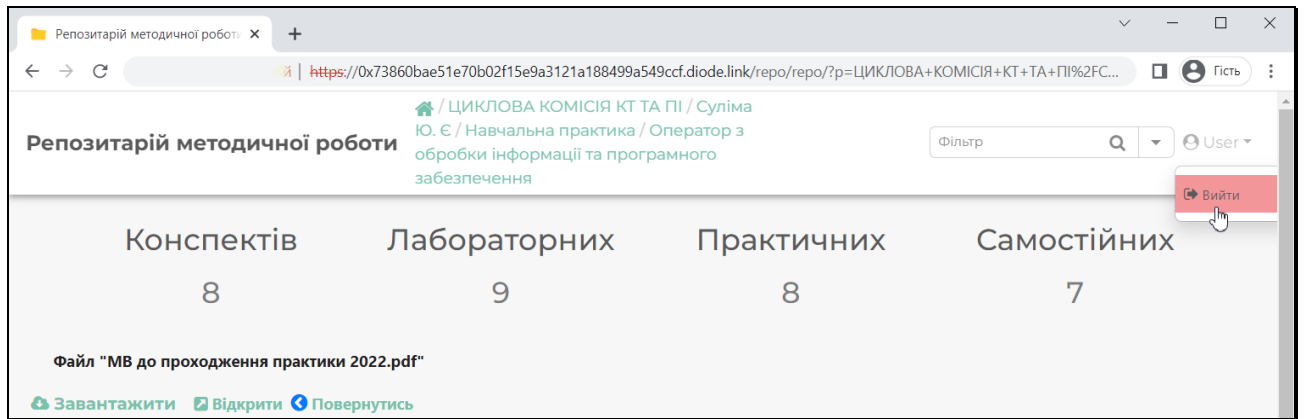


Рисунок 43 – Вхід в систему в режимі адміністратора

В режимі адміністратора в інтерфейсі репозитарію присутні кнопки «Створити» та «Завантажити», за допомогою яких адміністратор має змогу створити папки для всіх циклових комісій ВСП «ОТФК ОНТУ» (рис. 44) та наповнити їх (рис. 45). На рисунку 46 пронумеровано кнопки, за допомогою яких адміністратор може керувати файлами: «1» – «Видалити»; «2» – «Перейменувати»; «3» – «Копіювати»; «4» – «Посилання»; «5» – «Завантажити».

					КГ 06.16.001.00 ДП ПЗ	Арк.
						65
Ізм.	Лист	№ докум.	Підпис	Дата		

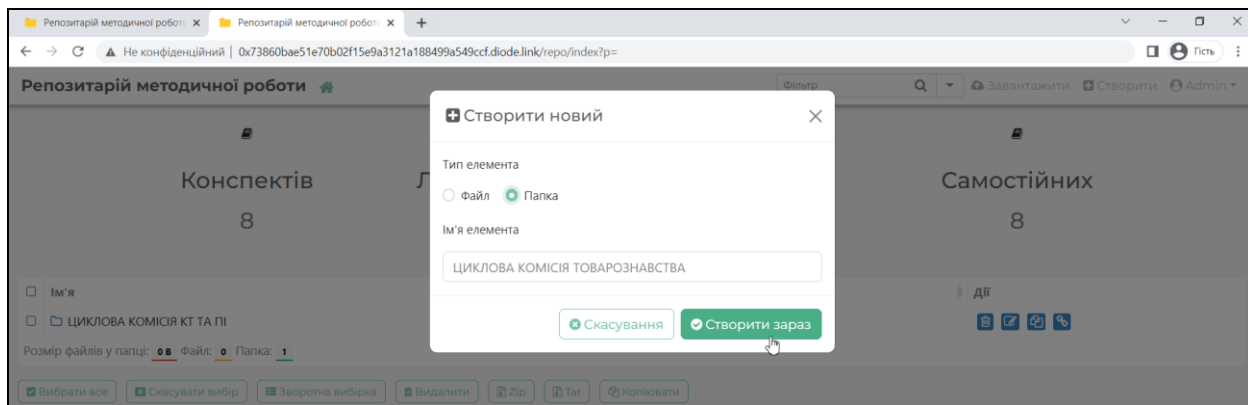


Рисунок 44 – Створення папки в режимі адміністратора

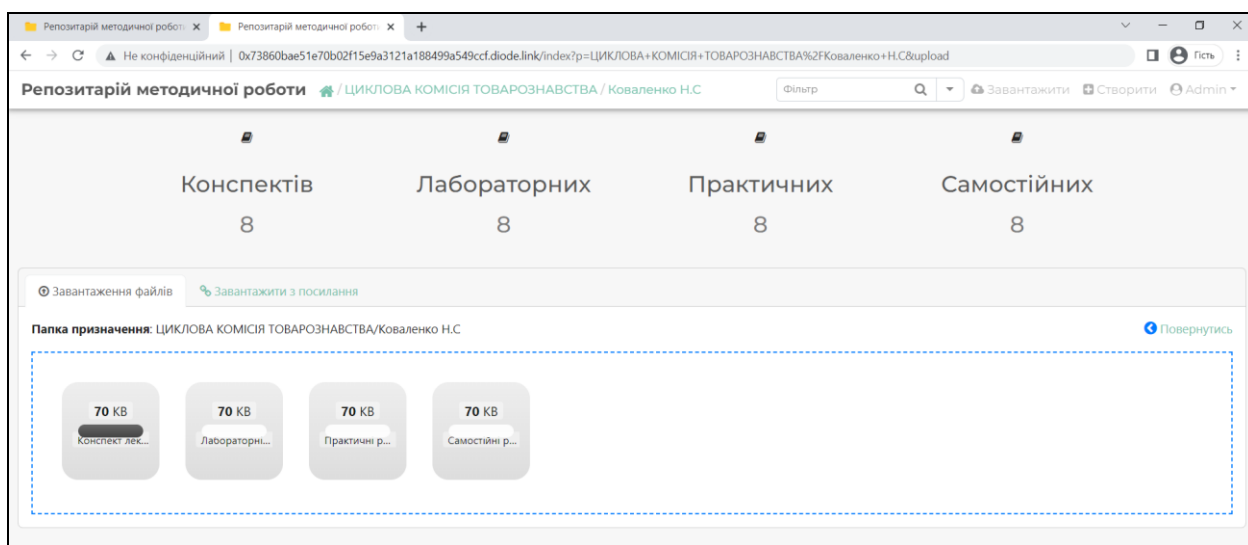


Рисунок 45 – Завантаження файлів в режимі адміністратора

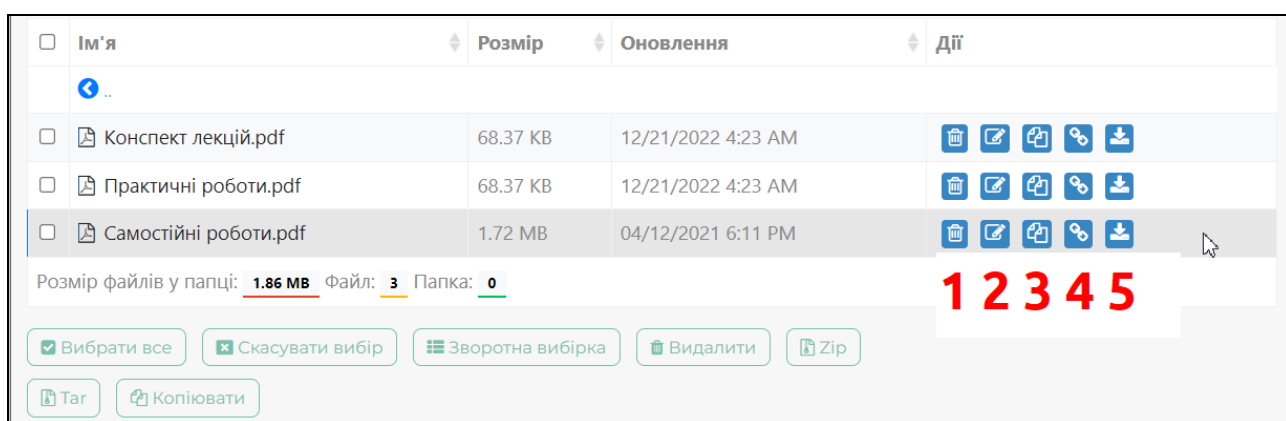


Рисунок 46 – Засоби керування файлами в режимі адміністратора

В подальшому планується використовувати створений репозитарій методичної роботи для інших циклових комісій та кафедр коледжу, передбачається його поступове наповнення та модифікація за потребою. Також планується розміщення матеріалів репозитарію на сервері зі статичною IP-адресою, що дозволить у подальшому відмовитися від використання сервісів тунелювання, таких як diode або ngrok.

В наступному розділі пояснювальної записки проведемо економічні розрахунки.

					КГ 06.16.001.00 ДП ПЗ	Арк.
						67
Ізм.	Лист	№ докум.	Підпис	Дата		

2 ЕКОНОМІЧНА ЧАСТИНА

В дипломному проекті створений репозитарій методичної роботи для ВСП «ОТФК ОНТУ».

Репозитарій призначений для збереження навчально-методичних розробок, які забезпечують провадження освітнього процесу за навчальними дисциплінами, що викладаються на цикловій (предметній) комісії комп'ютерних технологій та програмної інженерії, а також для постійного віддаленого доступу до цих матеріалів здобувачів освіти.

Репозитарій складається з трьох програмних додатків: Python-додаток для керування роботою мережевого сервісу тунелювання diode, який забезпечуватиме доступ до матеріалів репозитарію, який розміщено на сервері, через мережу Інтернет; PHP-додаток на хостингу для реалізації функцій файлового менеджера; PHP-файловий менеджер на локальній машині для випадків, коли доступ до даних через мережу неможливий.

Розробка реалізована за допомогою наступного інструментарію: для розробки графічного інтерфейсу та клієнтського додатку (Front-End) були задіяні бібліотеки фреймворків Bootstrap 5.2 і JQuery 3.6; для розробки серверного коду додатку (Back-End) – середовище PHP 8.1; для супутніх із розробкою операцій (DevOps), наприклад, налаштування серверу, використовувався веб-сервер Apache 2.4, мережевий сервіс тунелювання diode 1.8.1, середовище розробки (IDE) Python 3.11.

Репозитарій – є централізованою цифровою платформою, яка використовується для зберігання, організації, управління та поширення навчально-методичних розробок, наукових матеріалів, презентацій та інших ресурсів, що стосуються діяльності конкретного закладу освіти.

					КГ 06. 01 002. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		68

Головне призначення репозитарію полягає у полегшенні доступу до інформації, збільшенні її видимості та прискоренні обміну знаннями. Він дозволяє викладачам зберігати свої конспекти лекцій, методичні вказівки до виконання практичних робіт, методичні вказівки до проведення семінарських занять, методичні вказівки до виконання курсової роботи (проекту), методичні вказівки до виконання самостійної роботи, наукові публікації та інші матеріали в одному місці, що робить їх доступними для інших колег, а також для здобувачів освіти.

Репозитарій сприяє збереженню, організації та довгостроковому зберіганню цифрових ресурсів, забезпечуючи їх сталу доступність.

При оцінці ефективності створюваного веб-додатку слід виходити з того, що залежно від характеру ефекту, що досягається, можуть бути визначені наступні види ефективності веб-додатку: економічна, функціональна та соціальна ефективність. Ці види ефективності в свою чергу залежать від технологічних, технічних, маркетингових та економічних показників веб-сайту. Між доходом від веб-додатку і витратами на його створення існує певна залежність. Слід також враховувати, що на величину доходу від веб-додатку, окрім інших чинників ефективності, можуть вплинути витрати на його оптимізацію та просування.

В цьому розділі пояснювальної записки до дипломного проекту обчислимо економічну ефективність створення репозитарію методичної роботи для ВСП «ОТФК ОНТУ».

Загальні витрати (V_3) на створення веб-додатку складаються з декількох параметрів:

$$V_3 = V_p + V_v + V_e ,$$

де V_p – витрати на розробку веб-додатку;

					КГ 06. 01 002. 00 ДП ПЗ	Арк.
						69
Ізм.	Лист	№ докум.	Підпис	Дата		

V_b – витрати на впровадження веб-додатку;

V_e – витрати на експлуатацію сайту;

Витрати на розробку веб-додатку (V_p) є одноразовими та складаються з вартості наступних видів робіт зі створення сайту: розробка макетів дизайну для веб-інтерфейсу додатку; реалізація на сайті навігації; підготовка сторінок-шаблонів; наповнення веб-додатку інформацією; обробка зображень для їх публікації; програмна реалізація модулів веб-додатку та ін.

Витрати на впровадження веб-додатку (V_b) складаються з витрат на реєстрацію доменного імені терміном мінімум на один рік та з витрат на реєстрацію в пошукових системах.

Витрати на експлуатацію веб-додатку (V_e) складаються з витрат на підтримку репозитарію в робочому стані та витрат на хостинг, заробітну платню обслуговуючому персоналу – адміністратору, який буде підтримувати репозитарій у робочому стані (як правило не щомісячну, а в разі необхідності – згідно акту виконаних робіт) та оплати послуг провайдера доступу до мережі Інтернет.

В зв'язку з тим, що орендна плата за канали зв'язку, приміщення, електрику та ін. входить до оплати послуг хостингу (так само, як і амортизаційні відрахування), при розрахунку економічної ефективності веб-додатку, який не розміщений на технічному обладнанні підприємства та у його приміщеннях (орендованих площах), ці витрати окремою статтею враховувати не потрібно.

Роботи по підтримці репозитарію в робочому стані включають в себе: оновлення даних; створення нових розділів; видалення застарілої інформації; додавання потрібної інформації; налаштування параметрів сервера хостингу; моніторинг роботи сервера хостингу та ін. Для певних

					КГ 06. 01 002. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		70

робіт з цього переліку може використовуватися обслуговуючий персонал (адміністратор репозитарію).

Для визначення витрат на розробку репозитарію (B_p) необхідно розрахувати оплату праці виконавців, безпосередньо задіяних для її виконання. Для реалізації проекту використовуються такі професійні ресурси: дизайнер, програміст, веб-технолог і тестувальник.

Для визначення трудомісткості розробки репозитарію (B_p) складено план-графік по розробці веб-додатку і тривалості виконання робіт. Розподіл робіт по етапах і видах виконавців наведено в таблиці 2.1.

Таблиця 2.1 – План-графік по розробці репозитарію

№	Назва етапу	Час виконання (днів)	Посада виконавця
1	Аналіз завдання на розробку	4	Веб-технолог
2	Проектування дизайну веб-інтерфейсу	4	Дизайнер
3	Проектування архітектури файлового менеджера	4	Веб-технолог
4	Реалізація окремих додатків та їх спільної роботи	6	Програміст
5	Узгодження роботи додатків і мережевого сервісу тунелювання	6	Програміст
6	Тестування роботи веб-інтерфейсу та його оптимізація	2	Тестувальник Дизайнер Програміст Веб-технолог
7	Публікація додатку на хостинг та реалізація доступу до даних за вказаною адресою	2	Веб-технолог Тестувальник
ВСЬОГО:		28	

Розрахунок трудомісткості здійснений у такій послідовності:

1. Складений перелік всіх етапів і видів робіт, які необхідно виконати в ході розробки. Після узгодження з керівником проекту є можливим виключення, доповнення, об'єднання окремих етапів і видів робіт.

2. По кожному виду робіт визначена кваліфікація рівень виконавців. В разі виконання однієї роботи виконавцями різної кваліфікації, робота розподілена на ряд паралельних конкретних робіт для кожної категорії виконавця.

В умовах відсутності нормативної бази тривалість виконання окремих робіт розраховується на основі вірогідних оцінок робіт, що задаються виконавцями.

Розмір заробітної плати розраховується виходячи з чисельності різних категорій виконавців, трудомісткості, що витрачається ними на виконання різних видів робіт, а також їх середньої заробітної плати (ставки) за один робочий день. Витрати на заробітну плату приведені в таблиці 3.2.

Відповідно до статті 8 «Закону про Державний бюджет України на 2023» встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2023 року – 6700 гривень; мінімальну погодинну тарифну ставку – 40,43 грн. Середня заробітна плата за один робочий день для кожного виконавця визначається по формулі: $Z_{\text{ден}} = \text{п.т.с.} \times 8$, де

п.т.с. – погодинна тарифна ставка, грн.; 8 – тривалість робочого дня, год.

$$Z_{\text{ден}} \text{ дизайнера} = 45,5 \times 8 = 364,0 \text{ грн.}$$

$$Z_{\text{ден}} \text{ програміста} = 55,4 \times 8 = 442,2 \text{ грн.}$$

$$Z_{\text{ден}} \text{ веб-технолога} = 45,5 \times 8 = 364,0 \text{ грн.}$$

$$Z_{\text{ден}} \text{ тестувальника} = 42,3 \times 8 = 338,4 \text{ грн.}$$

					КГ 06. 01 002. 00 ДП ПЗ	Арк.
						72
Ізм.	Лист	№ докум.	Підпис	Дата		

Таблиця 2.2 – Витрати на заробітну плату

№	Персонал	Етапи розробки	Кількість днів	Денна ставка, грн.	Витрати, грн.
1	Дизайнер	2, 6	6	364,0	2184,0
2	Програміст	4-6	14	443,2	6204,8
3	Веб-технолог	1, 3, 6, 7	12	364,0	4368,0
4	Тестувальник	6, 7	4	338,4	1353,6
ВСЬОГО:					14110,4

До складу витрат на оплату праці також включаються податки, збори та інші обов'язкові платежі, встановлені діючою системою оподаткування. Розмір єдиного соціального внеску наразі складає 22% від заробітної плати. Отже обчислюємо його за наступною формулою:

$$V_{\text{есв}} = V_{\text{зп}} \times 0,22$$

$$V_{\text{есв}} = 14110,4 \times 0,22 = 3104,29 \text{ грн.}$$

Визначимо загальні витрати (V_p) на розробку репозитарію методичної роботи для ВСП «ОТФК ОНТУ», як суму витрат на заробітну плату праці персоналу ($V_{\text{зп}}$) та єдиного соціального внеску ($V_{\text{есв}}$):

$$V_p = V_{\text{зп}} + V_{\text{есв}}$$

$$V_p = 14110,4 + 3104,29 = 17214,69 \text{ грн.}$$

Витрати на впровадження та експлуатацію репозитарію варто розглядати як постійні витрати ($V_{\text{пост}}$), які будуть враховуватися протягом життя проекту:

$$V_{\text{пост}} = V_{\text{в}} + V_{\text{е.}}$$

У таблиці 3.3 визначимо постійні витрати як суму витрат на впровадження та експлуатацію репозитарію протягом року (щомісячні

					КГ 06. 01 002. 00 ДП ПЗ	Арк.
						73
Ізм.	Лист	№ докум.	Підпис	Дата		

витрати помножимо на 12 для отримання суми витрат протягом року). Підтримкою веб-додатку займатиметься один із співробітників організації, якому встановлять доплату розміром 10% від мінімальної заробітної платні за виконання функцій адміністратора репозитарію.

Включимо до постійних витрат реєстрацію доменного імені, послуги хостингу (щомісячна витрата), послуги провайдера, що надає широкосмуговий доступ (ШСД) до мережі Інтернет (щомісячна витрата), доплата адміністратору репозитарію.

Таблиця 2.3 – Постійні витрати

№	Стаття витрат	Вартість за рік, грн.
1	Доменне ім'я	300,00
2	Хостинг	12*100,00 = 1200,00
3	Доступ до мережі Інтернет	12*200,00 = 2400,00
4	Доплата адміністратору репозитарію	12*670,00 = 8040,00
ВСЬОГО:		11940,00

Враховуючи отримані показники, розрахуємо загальні витрати (V_3) на створення, впровадження та експлуатацію репозитарію:

$$V_3 = V_p + (V_b + V_e) = V_p + V_{\text{пост}}$$

$$V_3 = 17214,69 + 11940,00 = 29154,69 \text{ грн.}$$

Економічна ефективність за рік (E_p) визначається як сукупність коштів, вивільнених за рахунок впровадження та експлуатації репозитарію:

Вивільнення коштів може бути досягнуто за рахунок скорочення витрат на друк навчально-методичних матеріалів:

– папір А4 500 арк. – 230,00 грн.,

- брошурування – 30 грн. за 1 примірник;
- інші витратні матеріали (закупівля та/або заправка картриджів, заміна друкуючих барабанів тощо) – 600 грн. картридж з ресурсом на 1500 аркушів.

Розрахуємо орієнтовну вартість друку одного примірника навчального-методичного забезпечення, наприклад, конспекту лекцій зі 100 аркушів.

$$V_d = (V_{\text{п}} + V_{\text{б}} + V_{\text{вм}}),$$

де V_d – вартість друку, $V_{\text{п}}$ – вартість паперу, $V_{\text{б}}$ – вартість брошурування, $V_{\text{вм}}$ – вартість витратних матеріалів.

$$V_{\text{п}} = (230,0/500) * 100 = 46,0 \text{ грн.}$$

$$V_{\text{б}} = 30,0 \text{ грн.}$$

$$V_{\text{вм}} = (600/1500) * 100 = 40,0 \text{ грн.}$$

$$V_d = 46,0 + 30,0 + 40,0 = 116,0 \text{ грн.}$$

Тобто вартість друку одного примірника навчального-методичного забезпечення обсягом 100 аркушів становить 116,0 грн.

Як правило, з навчально-методичного комплексу навчальної дисципліни здобувачу освіти потрібні конспект лекцій, методичні вказівки до виконання лабораторних/практичних робіт та методичні вказівки до самостійних робіт.

$$V_{\text{нмк}} = V_d * 3 = 116,0 * 3 = 348,0 \text{ грн.}$$

де $V_{\text{нмк}}$ – вартість друку обов'язково потрібної частини навчально-методичного комплексу для одного здобувача освіти.

Вартість друку для навчальної групи з 25 осіб складе:

$$V_{\text{г}} = V_{\text{нмк}} * 25 = 348,0 * 25 = 8700,0 \text{ грн.}$$

					КГ 06. 01 002. 00 ДП ПЗ	Арк.
						75
Ізм.	Лист	№ докум.	Підпис	Дата		

В коледжі за денною формою навчання щорічно навчається приблизно 50 груп із середньою наповненістю 25 осіб. Для 1-го і 2-го курсу більшість предметів забезпечуються підручниками з фонду бібліотеки коледжу (в т.ч. електронного).

Тому вважатимемо, що отримання навчально-методичного комплексу з дисципліни буде потрібним групам 3-4 курсу та бакалаврату, що складатиме не менш ніж 50% від загальної кількості груп, які навчаються за денною формою навчання.

Тобто, приблизно 25 груп в середньому по 25 осіб у групі, що дорівнюватиме 625 осіб, які будуть потребувати отримання навчально-методичного комплексу з дисципліни в друкованому вигляді.

В середньому група вивчає 15 навчальних дисциплін на навчальний рік. Враховуючи зазначене вище, визначимо витрати на забезпечення навчальними посібниками здобувачів освіти коледжу по всім предметам на навчальний рік.

$$V_{зк} = V_{г} * 25 * 15 = 8700 * 25 * 15 = 3\ 262\ 500 \text{ грн.}$$

В реальності з всієї групи лише 1-2 особам потрібна друкована версія навчально-методичного комплексу. До того ж є дисципліни з яких є навчальні посібники, підручники, тобто друкувати навчально-методичний комплекс не потрібно. Приймемо, що на навчальний рік в кожній групі 3-4 курсу та бакалаврату 1 особа надрукує повний навчально-методичний комплекс з 10 предметів, які вивчає.

Враховуючи цю поправку обчислюємо реально вивільнені кошти на рік:

$$V_{р} = V_{нмк} * 1 * 25 * 10 = 87000,0 \text{ грн.}$$

					КГ 06. 01 002. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		76

Економічна ефективність за рік (E_p) визначається як сукупність коштів, вивільнених за рахунок впровадження та експлуатації репозитарію:

$$E_p = B_p = 87000,0 \text{ грн.}$$

Коефіцієнт економічної ефективності за перший рік вираховується таким чином:

$$K_e = E_p / B_3 = 87000,0 / 29154,69 = 2,98$$

Запровадження репозитарію методичної роботи для ВСП «ОТФК ОНТУ» дозволить вивільнити 87000,0 грн. на рік за рахунок скорочення витрат на друк навчально-методичних комплексів. Коефіцієнт економічної ефективності розробки складає ≈ 3 , що є достатнім показником.

					КГ 06. 01 002. 00 ДП ПЗ	Арк.
						77
Ізм.	Лист	№ докум.	Підпис	Дата		

3 ОХОРОНА ПРАЦІ

У цьому розділі дипломного проекту будуть розглянуті питання охорони праці, а також техніки безпеки на підприємствах, що стосуються роботи з комп'ютерною технікою, оскільки при розробці будь-якого програмного забезпечення, веб-додатків, більшу частину часу веб-розробник (програміст) проводить за комп'ютером, це ж стосується і користувачів створеного репозитарію методичної роботи для ВСП «ОТФК ОНТУ».

Повністю безпечних та нешкідливих виробництв не існує. Завдання охорони праці в Україні – звести до мінімуму вірогідність нещасного випадку або захворювання того, хто працює, забезпечивши при цьому максимальну продуктивність праці.

Охорона праці забезпечується проведенням технічних, санітарно-гігієнічних і правових заходів. Нагляд і контроль за дотриманням законодавства про працю та правил по охороні праці здійснюють державні органи інспекції та профспілки.

3.1 Аналіз та безпека умов праці працівника на робочому місці

3.1.1 Організація робочого місця

Для аналізу умов праці за основне робоче місце працівника ми приймаємо робоче місце користувача ПЕОМ.

Дані вимоги регламентуються Державними санітарними нормами і правилами роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98.

На робочому місці веб-розробника (програміста) можуть існувати різні фактори небезпеки, які можуть впливати на їхнє здоров'я та безпеку:

- Неправильна ергономіка робочого місця;

					КГ 06.01.003.00 ДП ПЗ	Арк.
						78
Ізм.	Лист	№ докум.	Підпис	Дата		

- Перенапруга очей;
- Одноманітна робота може спричиняти втому, перевтому та зниження концентрації, що може призвести до помилок та нещасних випадків;
- Безперервне набирання на клавіатурі або клацання мишею, може призводити до розвитку стресових травм, таких як синдром карпального каналу або тунельний синдром;
- Відсутність фізичної активності. Довгі періоди сидячої роботи можуть спричиняти седентарний спосіб життя, що може призвести до проблем зі здоров'ям, такими як ожиріння, серцево-судинні захворювання та м'язово-скелетні проблеми.

Ці фактори варто враховувати та приділяти їм увагу, щоб забезпечити безпечні умови праці веб-розробників на робочому місці.

Вимоги до робочого місця користувача ПЕОМ регламентуються Державними санітарними нормами і правилами роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98:

Розміщення робочих місць з ВДТ, ЕОМ і ПЕОМ у підвальних приміщеннях, на цокольних поверхах заборонено.

Площа на одне робоче місце має становити не менше ніж $6,0 \text{ м}^2$, а об'єм не менше ніж $20,0 \text{ м}^3$.

Віконні прорізи приміщень для роботи з ВДТ мають бути обладнані регульованими пристроями (жалюзі, завіски, зовнішні козирки).

Для внутрішнього оздоблення приміщень з ВДТ слід використовувати дифузно-відбивні матеріали з коефіцієнтами відбиття для стелі 0,7 - 0,8, для стін 0,5 - 0,6. Покриття підлоги повинне бути матовим з коефіцієнтом відбиття 0,3 - 0,5. Поверхня підлоги має бути рівною, неслизькою, з антистатичними властивостями.

					КГ 06.01.003.00 ДП ПЗ	Арк.
						79
Ізм.	Лист	№ докум.	Підпис	Дата		

Забороняється для оздоблення інтер'єру приміщень ВДТ застосовувати полімерні матеріали (деревинно-стружкові плити, шпалери, що миються, рулонні синтетичні матеріали, шаруватий паперовий пластик тощо), що виділяють у повітря шкідливі хімічні речовини.

Виробничі приміщення можуть обладнуватись шафами для зберігання документів, магнітних дисків, полицями, стелажми, тумбами тощо з урахуванням вимог до площі приміщень.

У приміщеннях з ВДТ слід щоденно робити вологе прибирання. Приміщення з ВДТ мають бути оснащені аптечками першої медичної допомоги.

При приміщеннях з ВДТ мають бути обладнані побутові приміщення для відпочинку під час роботи, кімната психологічного розвантаження. В кімнаті психологічного розвантаження слід передбачити встановлення пристроїв для приготування й роздачі тонізуючих напоїв, а також місця для занять фізичною культурою (СНиП 2.09.04.-87).

Приміщення для роботи з ВДТ мають бути обладнані системами опалення, кондиціонування повітря, або припливно-витяжною вентиляцією відповідно до СНиП 2.04.05-91.

3.1.2 Коротка характеристика і основні вимоги безпеки до мікроклімату виробничих приміщень, освітлення, шуму, вібрації, ультразвуку, інфразвуку, виробничих випромінювань, небезпека ураження електричним струмом

Значення мікроклімату для здоров'я працівника, який працює за комп'ютером, дуже велике. Мікрокліматом називають сукупність кліматичних чинників усередині приміщення, які впливають на працездатність людини. Нормальний мікроклімат: температура 21-25 °С, вологість повітря – 40-60%, швидкість руху повітря – 0,1 м/с. Істотний

					КГ 06.01.003.00 ДП ПЗ	Арк.
						80
Ізм.	Лист	№ докум.	Підпис	Дата		

вплив на мікроклімат роблять джерела тепла, що знаходиться в приміщеннях. Також сюди відносять теплоту, що надходить через вікна від сонячної радіації та притока теплоти через непрозорі захищаючі конструкції.

Основним обладнанням, яке виділяє тепло в приміщенні для розробки програмних продуктів є персональні комп'ютери – в середньому до 85% сумарного тепловиділення. Тепловиділення від приладів освітлення складає в середньому до 10%, від людей – до 2%, від сонячної радіації – 6%, інші види – до 1%.

Значний вплив на організм людини, а також на роботу обладнання в приміщенні надає відносна вологість повітря. При вологості повітря до 40% стає крихкою основа магнітного диска, виходить з ладу ізоляція дротів, а також виникає статична електрика. При відносній вологості повітря більше 75-80% знижується опір ізоляції, істотно змінюються робочі характеристики елементів, зростає інтенсивність відмов.

Необхідно регулярно провітрювати виробниче приміщення. Для досягнення якісного мікроклімату слід використовувати кондиціонер, бажано із вбудованим іонізатором, регулярно проводити вологе прибирання із використанням якісного пилососу зі спеціальним фільтром.

В холодні періоди року температура повітря повинна складати 22-24 °С; швидкість руху повітря – 0,1 м/с; відносна вологість повітря – 40-60 %. В теплі періоди року температура повітря повинна складати 23-25 °С; швидкість руху повітря – 0,1-0,2 м/с; вологість – 40-60%. Температура повітря може коливатися від 22 до 26°С при збереженні інших параметрів мікроклімату. Необхідно надати можливість індивідуального регулювання роздачі повітря в приміщеннях шляхом встановлення кондиціонерів.

					КГ 06.01.003.00 ДП ПЗ	Арк.
						81
Ізм.	Лист	№ докум.	Підпис	Дата		

Нормовані параметри мікроклімату, іонного складу повітря, вмісту шкідливих речовин мають відповідати вимогам СН 4088-86, СН 2152-80, ГОСТ 12.1.005-88, ГОСТ 12.1.007-76 (дод. 2, 3).

Правильне проектування і виконання освітлення забезпечує високий рівень працездатності, надає позитивну психологічну дію на працівників, сприяє підвищенню продуктивності праці.

Приміщення для роботи з ВДТ повинні мати природне та штучне освітлення відповідно до ДБН В.2.5-28:2018 « Природне і штучне освітлення».

Природне освітлення має здійснюватися через світлові прорізи, орієнтовані переважно на північ чи північний схід. Віконні прорізи приміщень для роботи з ВДТ мають бути обладнані регульованими пристроями (жалюзі, завіски, зовнішні козирки

Загальне освітлення слід виконувати у вигляді безперервних або ж переривистих ліній світильників, розташованих збоку від робочих місць. Мінімальна норма освітлення робочого місця – 200 лк. Для освітлення треба використовувати або звичайні лампи накаливання із матованою поверхнею, або галогенові економичні лампи чи сучасні світлодіодні лампи, що дають у приміщенні теплий відтінок світла, по статистиці саме цей відтінок нормалізує самопочуття у тих, хто знаходиться в приміщенні (а також значно заощаджує електроенергію – близько 8 разів для галогенових, та близько 20 для світлодіодних).

Практично все обладнання в комп'ютерному залі відноситься до електричних установок, що представляють для людини велику небезпеку, оскільки в процесі експлуатації або проведення профілактичних робіт людина може торкнутися частин, що знаходяться під напругою.

Під час роботи (розробки програмного забезпечення) при дотику до будь-якого з елементів обладнання можуть виникнути розрядні струми

					КГ 06.01.003.00 ДП ПЗ	Арк.
						82
Ізм.	Лист	№ докум.	Підпис	Дата		

статичної електрики. Такі розряди небезпеки для людини не представляють, проте окрім неприємних відчуттів можуть привести до виходу з ладу техніки, що знаходиться в комп'ютерному залі.

В сучасних ПК дуже висока щільність розміщення елементів схем. При протіканні по ним електричного струму виділяється значна кількість теплоти, що може привести до підвищення температури окремих вузлів до 120 °С. При цьому можливе оплавлення ізоляції сполучних дротів, їх оголення і, як наслідок, коротке замикання, яке супроводжується іскрінням, що веде до недопустимих перевантажень елементів електронних схем.

Коли людина знаходиться у сфері дії інтенсивного електромагнітного поля або безпосередньо стикається з провідниками електричного струму, що знаходяться під напругою, по її тілу проходить електричний струм. В результаті дії струму на організм може виникнути електротравма, тобто значне порушення функцій життєдіяльності людини. Характер та інтенсивність порушень в організмі, що викликані електричним струмом, в основному визначаються типом і величиною струму, тривалістю його дії тощо.

Поразка організму людини більшою мірою залежить від величини струму, що проходить через життєво важливі органи людини – мозок, центральну нервову систему, органи управління дихання. Сила струму, що проходить через організм, визначається величиною прикладеної напруги та опором тіла людини.

3.2 Пожежна безпека

Пожежна небезпека – можливість виникнення та (або) розвитку пожежі в будь-якій речовині, процесі, стані. Слід зазначити, що пожеж безпечних не буває, якщо вони і не створюють прямої загрози життю та

					КГ 06.01.003.00 ДП ПЗ	Арк.
						83
Ізм.	Лист	№ докум.	Підпис	Дата		

здоров'ю людини, то завдають чи призводять до значних матеріальних втрат. Всі працівники, які перебувають у службових приміщеннях, мають бути ознайомлені та мають виконувати правила пожежної безпеки:

– меблі та обладнання необхідно розміщувати таким чином, щоб забезпечувався вільний евакуаційний прохід до дверей виходу з приміщення (завширшки не менше 1 м). Евакуаційні шляхи та виходи необхідно постійно утримувати вільними, нічим не захаращувати;

– електромережі, електроприлади і апаратуру експлуатувати тільки у справному стані з урахуванням вказівок та рекомендацій підприємств-виготовлювачів. У разі виявлення пошкоджень електромереж, вимикачів, розеток та інших електровиробів слід негайно вимкнути їх та вжити необхідних заходів щодо приведення в пожежобезпечний стан;

– документи, папір та інші горючі матеріали слід зберігати на відстані не менше 1 м від електрощитів; 0,5 м від електросвітильників; 0,6 м від сповіщувачів автоматичної пожежної сигналізації та 0,15 м від приладів центрального водяного опалення.

– засоби протипожежного захисту слід утримувати у справному стані.

Усі працівники повинні вміти користуватись наявними вогнегасниками, іншими первинними засобами пожежогасіння, знати місце їх знаходження.

Відстань від найбільш віддаленого місця приміщення до місця розташування вогнегасника не повинна перевищувати 20 м.

У службових приміщеннях не допускати:

- влаштування тимчасових електромереж;
- прокладання електричних проводів безпосередньо по горючій основі;
- експлуатація електроприладів, які мають механічні пошкодження;

					КГ 06.01.003.00 ДП ПЗ	Арк.
						84
Ізм.	Лист	№ докум.	Підпис	Дата		

- захаращування підступів до засобів пожежогасіння;
- куріння, використання легкозаймистих рідин;
- проведення вогневих, зварювальних та інших робіт без спеціального дозволу;
- вмикання електронагрівальних приладів (чайників, кип'ятильників тощо) без негорючих підставок та в тих місцях, де їх використання не передбачено (або заборонено);
- захаращування шляхів евакуації та евакуаційних виходів.

У разі, якщо пожежі не вдалось уникнути, необхідно:

- терміново повідомити пожежну охорону по телефону 101, вказати при цьому адресу, кількість поверхів, місце виникнення пожежі, наявність людей, своє прізвище;
- організувати евакуацію людей та матеріальних цінностей;
- повідомити про виникнення пожежі адміністрацію та чергового (за його наявності);
- вимкнути, у разі необхідності, струмоприймачі та вентиляцію;
- розпочати гасіння пожежі наявними первинними засобами пожежогасіння;
- організувати зустріч підрозділів пожежної охорони й надати їм консультаційну та іншу допомогу в процесі гасіння пожежі.

У всіх офісних приміщеннях на видних місцях біля телефонів мають бути розміщені таблички із зазначенням номера телефону для виклику пожежно-рятувальної служби. Якщо на поверсі працюють більше 25 осіб одночасно, то варто розробити та вивісити на видних місцях плани евакуації людей на випадок пожежі. Якщо ж на одному поверсі – більше 50 осіб, то крім схематичного плану евакуації має бути розроблена інструкція, що визначає дії персоналу задля безпечної та швидкої евакуації.

					КГ 06.01.003.00 ДП ПЗ	Арк.
						85
Ізм.	Лист	№ докум.	Підпис	Дата		

3.3 Висновки за розділом

При організації робочого місця фахівця, яке передбачає роботу з персональним комп'ютером (ПК) слід дотримуватися таких норм:

– комфортне та безпечне робоче місце. Стіл, стілець і монітор мають бути належним чином налаштовані для запобігання напруги та надмірного навантаження на тіло. Фахівець має регулярно робити перерви, щоб уникнути напруги і перевтоми.

– перед початком роботи слід перевірити чи всі електричні розетки та кабелі в хорошому стані, без пошкоджень. Не можна перевантажувати розетки подовжувачами або множниками. Слід уникати підключення занадто великої кількості пристроїв до однієї розетки, оскільки це може призвести до перегріву та загоряння.

– завжди слід пам'ятати про пожежні ризики. В робочому приміщенні треба встановити пожежні сповіщувачі та вогнегасники поруч з робочим місцем. Також слід проводити регулярні навчання, щоб пересвідчитися, що фахівці знають, як правильно використовувати вогнегасник і де знаходиться найближчий вихід у разі виникнення пожежі.

					КГ 06.01.003.00 ДП ПЗ	Арк.
						86
Ізм.	Лист	№ докум.	Підпис	Дата		

ВИСНОВОК

В дипломному проекті створено репозитарій методичної роботи для ВСП «ОТФК ОНТУ» на прикладі циклової комісії комп'ютерних технологій і програмної інженерії.

Для роботи з репозитарієм користувач (викладач або здобувач освіти) має перейти за адресою: <https://www.otfk-mwr.tk>. Далі потрібно пройти процедуру авторизації, після якої в автоматичному режимі запуститься Python-скрипт, що забезпечить старт мережевого сервісу тунелювання diode та згенерує активне посилання на сервер, на якому розміщені файли репозитарію та файловий менеджер для роботи з матеріалами репозитарію. Файли в репозитарії зберігаються у форматі PDF.

При створенні репозитарію методичної роботи для ВСП «ОТФК ОНТУ» було задіяно такий інструментарій: для розробки графічного інтерфейсу та клієнтського додатку – бібліотеки фреймворків Bootstrap 5.2 і JQuery 3.6; для розробки серверного коду додатку – середовище PHP 8.1; для супутніх із розробкою операцій – веб-сервер Apache 2.4, мережевий сервіс тунелювання diode 1.8.1, середовище розробки (IDE) Python 3.11; дизайн веб-інтерфейсу розроблявся засобами графічного редактору Adobe Photoshop.

В пояснювальній записці розглянуті всі питання передбачені технічним завданням на дипломне проектування, проведено аналіз предметної області; детально описано технології та засоби, які використовувалися при створенні репозитарію; проведено розрахунок економічної ефективності створення та запровадження репозитарію методичної роботи за рахунок скорочення витрат на друк навчально-методичних комплексів; розглянуті питання охорони праці та наведений перелік використаних джерел.

					КГ 06. 01 000. 00 ДП ПЗ	Арк.
						87
Ізм.	Лист	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. OpenDOAR Statistics [Електронний ресурс] Режим доступу до ресурсу: https://v2.sherpa.ac.uk/view/repository_visualisations/1.html (дата звернення 22.05.2023)
2. Спарлок Дж. Bootstrap / Дж. Спарлок. – 2013. [Електронний ресурс] – Режим доступу до ресурсу: <https://goo.gl/ExSvsK> (дата звернення 29.05.2023)
3. Manage assets with Diode CLI [Електронний ресурс] Режим доступу до ресурсу: <https://diode.io/solutions/cli/> (дата звернення 18.05.2023)
4. About DSpace [Електронний ресурс] Режим доступу до ресурсу: <https://dspace.lyrasis.org/about/> (дата звернення 20.05.2023)
5. Eprints [Електронний ресурс] Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Eprints> (дата звернення 17.05.2023)
6. Охорона праці на підприємстві: основні вимоги [Електронний ресурс] Режим доступу до ресурсу: https://bilgorod-d.gov.ua/page/ohorona_prac_na_pdprimstv_osnovn_vimogi (дата звернення 25.05.2023)
7. ДСанПіН 3.3.6.042-99 «Державні санітарні норми мікроклімату виробничих приміщень»
8. ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ЕОМ.
9. НАПБ А.0.001-95. «Правила пожежної безпеки в Україні»

					КГ 06. 01 000. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		88

Додаток А.

Слайди мультимедійної презентації

ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 – «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна графіка і Web-дизайн»

ДИПЛОМНИЙ ПРОЕКТ

КГ 06.01.000.00 ДП

БАБІЧ ВІКТОРІЇ СЕРГІЇВНИ

НА ТЕМУ:

Створення репозитарію методичної роботи для ВСП «ОТФК ОНТУ»

Загальні відомості про репозитарії

Репозитарій або **репозиторій** – це цифрове сховище, де містяться дані певного типу. Найчастіше він представляє собою каталог інформації з файлами, локальний або розташований на зовнішньому ресурсі в мережі, що розрахований на віддалений авторизований доступ до цієї інформації.

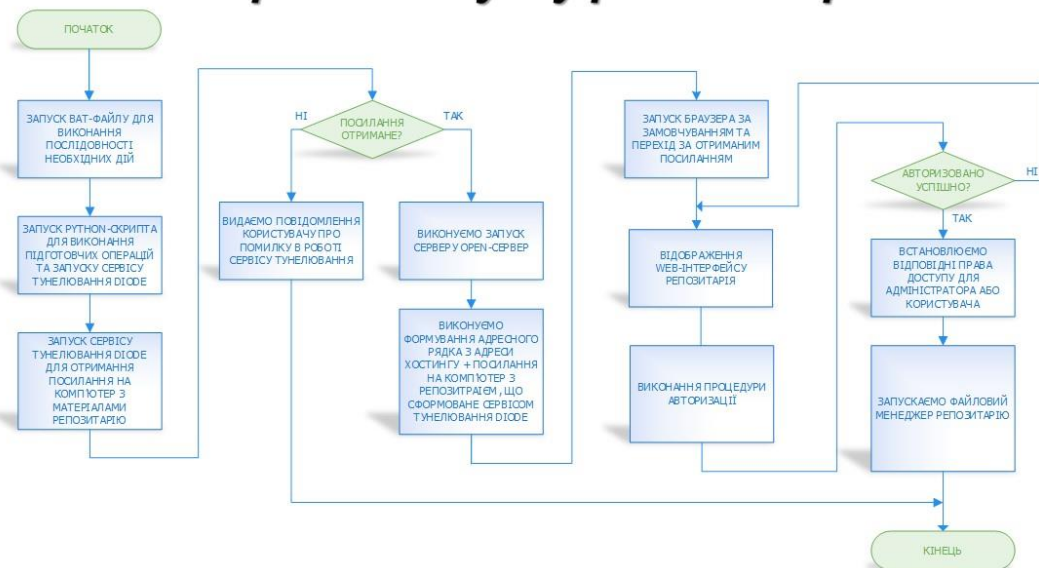
Репозитарій, який створено під час виконання дипломного проектування, призначений для збереження навчально-методичних розробок, які забезпечують провадження освітнього процесу за навчальними дисциплінами, що викладаються на цикловій (предметній) комісії комп'ютерних технологій та програмної інженерії, а також для постійного віддаленого доступу до цих матеріалів здобувачів освіти.

Структура репозитарію

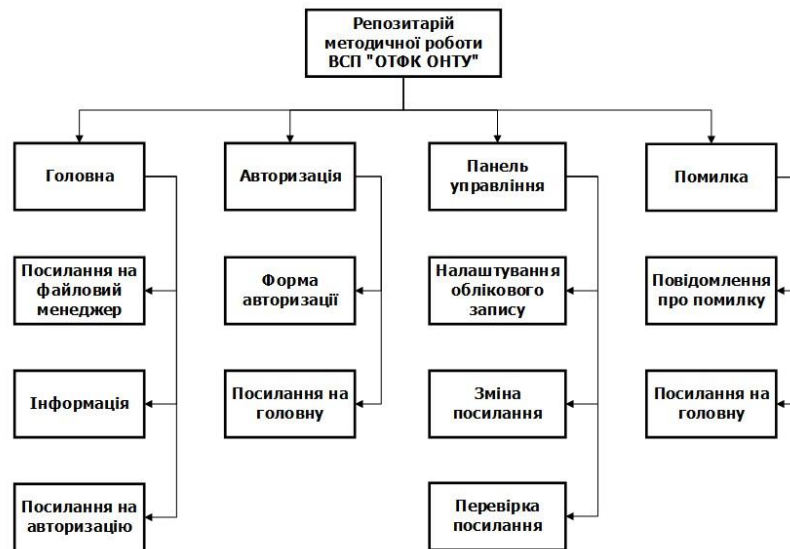
[ЦИКЛОВА КОМІСІЯ «НАЗВА_ЦК»]

- [Викладач_1]
- [Викладач_2]
- [Викладач_3]
 - [Дисципліна_1]
 - [Дисципліна_2]
 - [Дисципліна_3]
 - Навчальний посібник_1.pdf
 - Навчальний посібник_2.pdf
 - Навчальний посібник_3.pdf

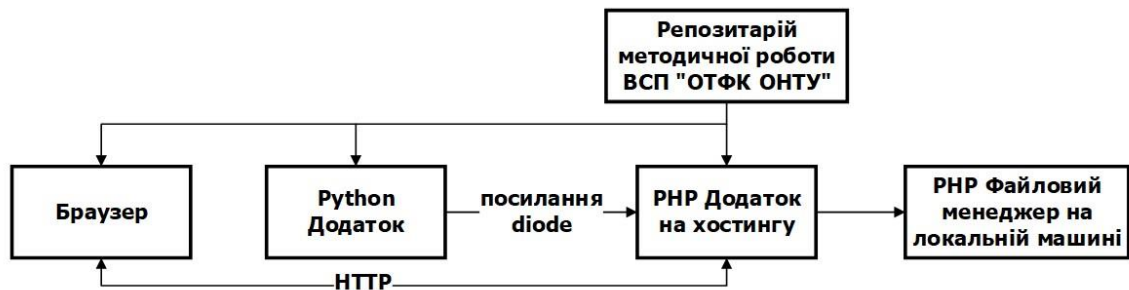
Алгоритм запуску репозитарію



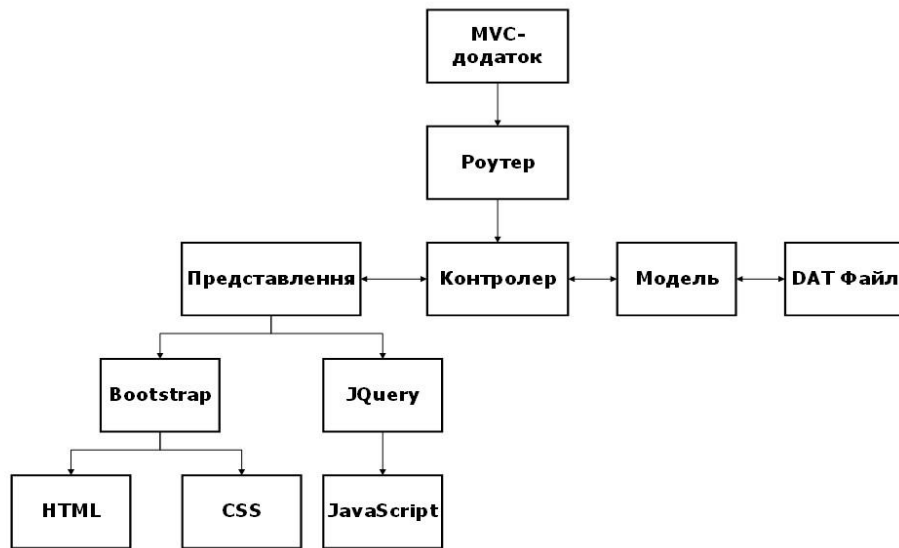
Інформаційна архітектура файлового менеджера



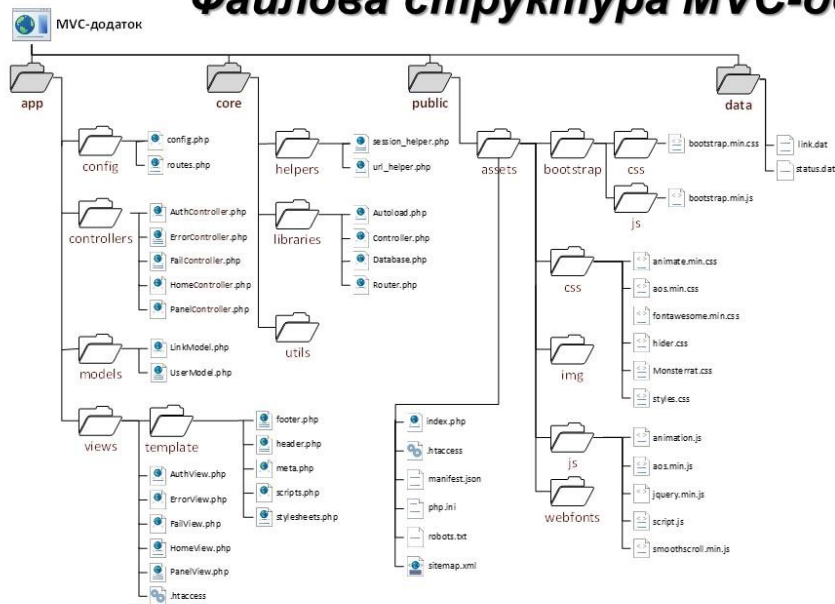
Взаємодія програмних модулів репозитарію



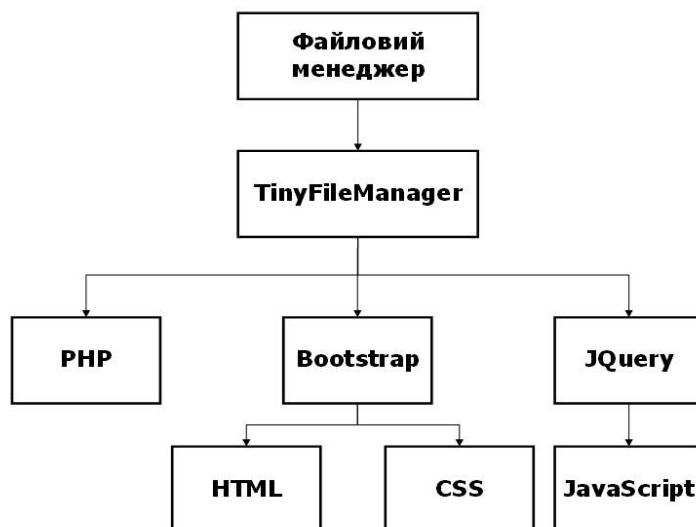
Модульна структура MVC-додатку



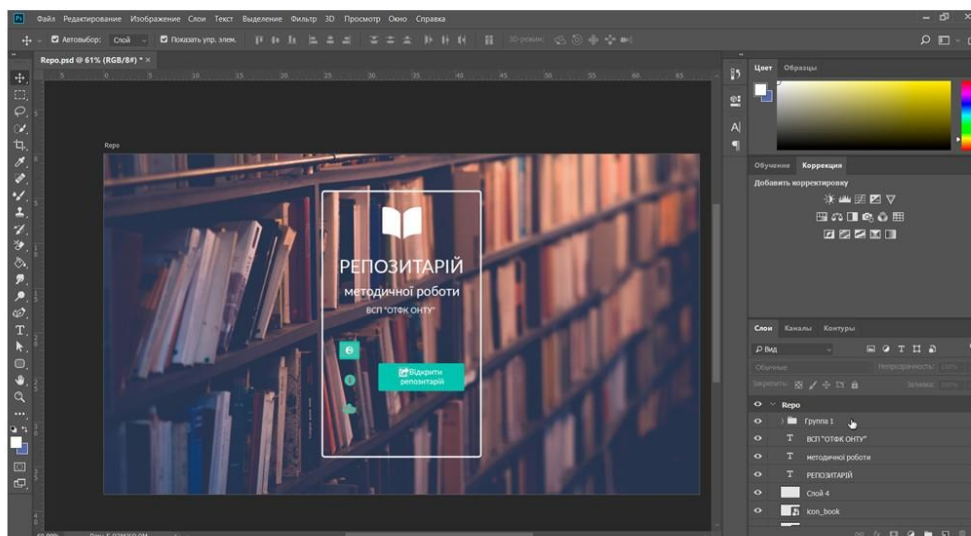
Файлова структура MVC-додатку



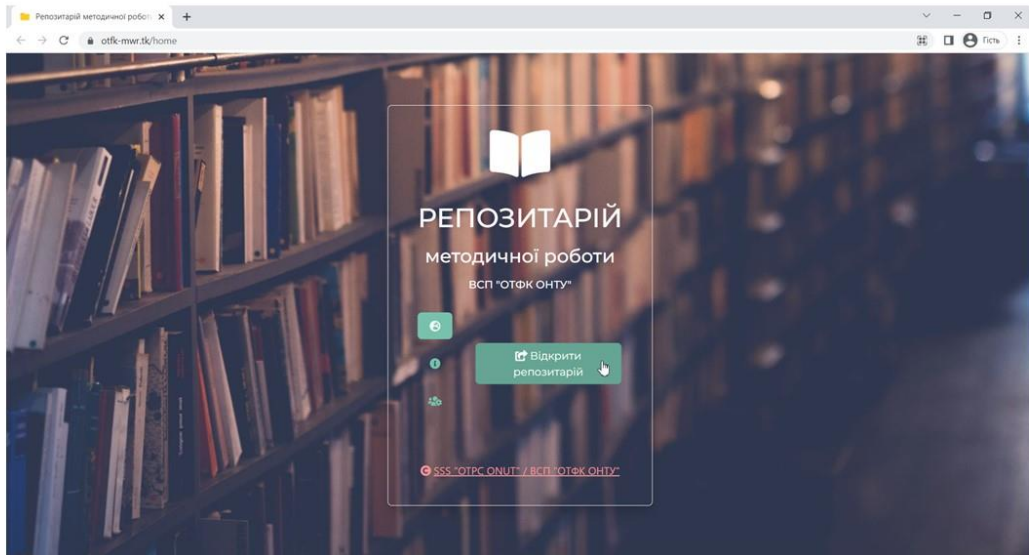
Структура файлового менеджера MVC-додатку



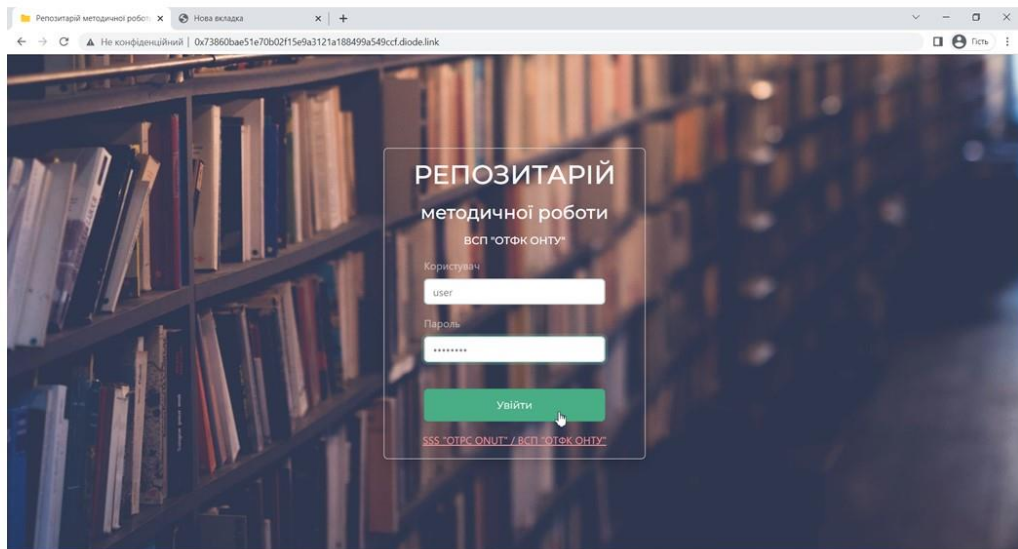
Розробка дизайну головної сторінки



Скріншоти роботи з репозитарієм



Скріншоти роботи з репозитарієм



Скріншоти роботи з репозитарієм

Репозитарій методичної роботи

Конспектів	Лабораторних	Практичних	Самостійних
8	8	8	8

Ім'я	Розмір	Оновлення	Дії
ЦИКЛОВА КОМІСІЯ КТ ТА ПІ	Папка	05/14/2023 4:23 PM	

Розмір файлів у папці: 0 Б Файл: 0 Папка: 1

Скріншоти роботи з репозитарієм

Репозитарій методичної роботи / ЦИКЛОВА КОМІСІЯ КТ ТА ПІ

Конспектів	Лабораторних	Практичних	Самостійних
8	9	8	7

Ім'я	Розмір	Оновлення	Дії
Іванова Л. В.	Папка	05/14/2023 4:23 PM	
Кривченко А. А.	Папка	05/14/2023 4:23 PM	
Кривченко Ю. В.	Папка	05/14/2023 4:23 PM	
Скорнякова О. В.	Папка	05/14/2023 4:23 PM	
Суліма Ю. Є.	Папка	06/04/2023 2:15 PM	
Суліма Ю. Ю.	Папка	05/14/2023 4:23 PM	

Розмір файлів у папці: 0 Б Файл: 0 Папка: 6


<https://0x73860bae51e70b02f15e9a3121a188499a549cfdiode.link/repo/>

Скріншоти роботи з репозитарієм

Репозитарій методичної роботи / ЦИКЛОВА КОМІСІЯ КТ ТА ПІ / Суліма Ю. Є / Навчальна практика

Фільтр 🔍 User ▾

Конспектів	Лабораторних	Практичних	Самостійних
8	8	8	7

Ім'я	Розмір	Оновлення	Дії
Оператор з обробки інформації та програмного забезп...	Папка	06/04/2023 2:18 PM	



Розмір файлів у папці: 0в Файл: 0 Папка: 1

Скріншоти роботи з репозитарієм

Репозитарій методичної роботи / ЦИКЛОВА КОМІСІЯ КТ ТА ПІ / Суліма Ю. Є / Навчальна практика / Оператор з обробки інформації та програмного забезпечення

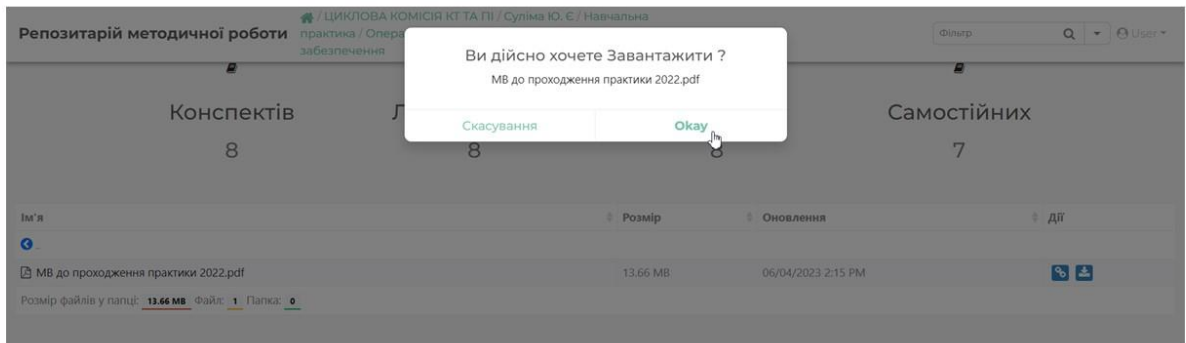
Фільтр 🔍 User ▾

Конспектів	Лабораторних	Практичних	Самостійних
8	9	8	7

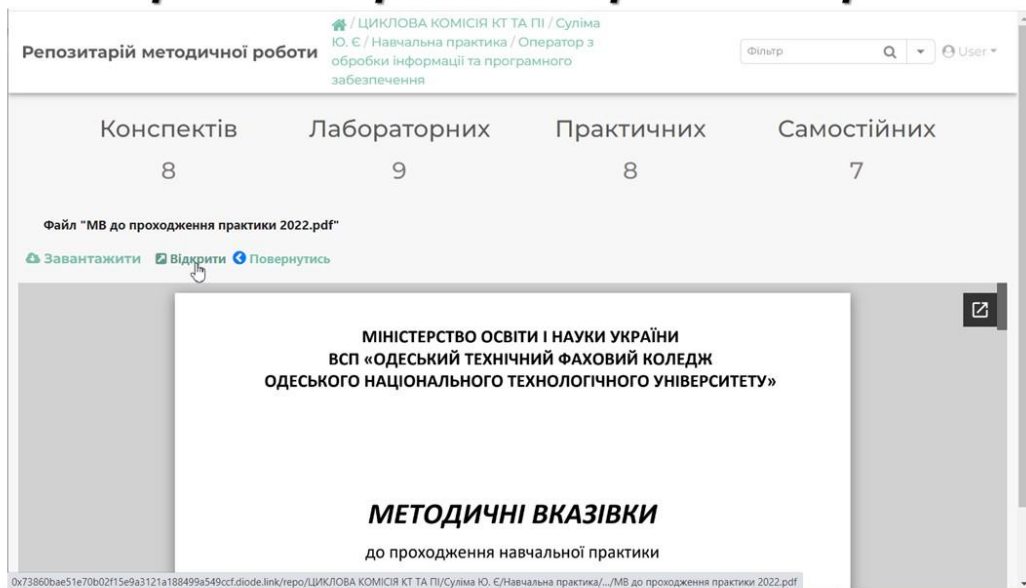
Ім'я	Розмір	Оновлення	Дії
МВ до проходження практики 2022.pdf	13.66 MB	06/04/2023 2:15 PM	 

Розмір файлів у папці: 13.66 MB Файл: 1 Папка: 0

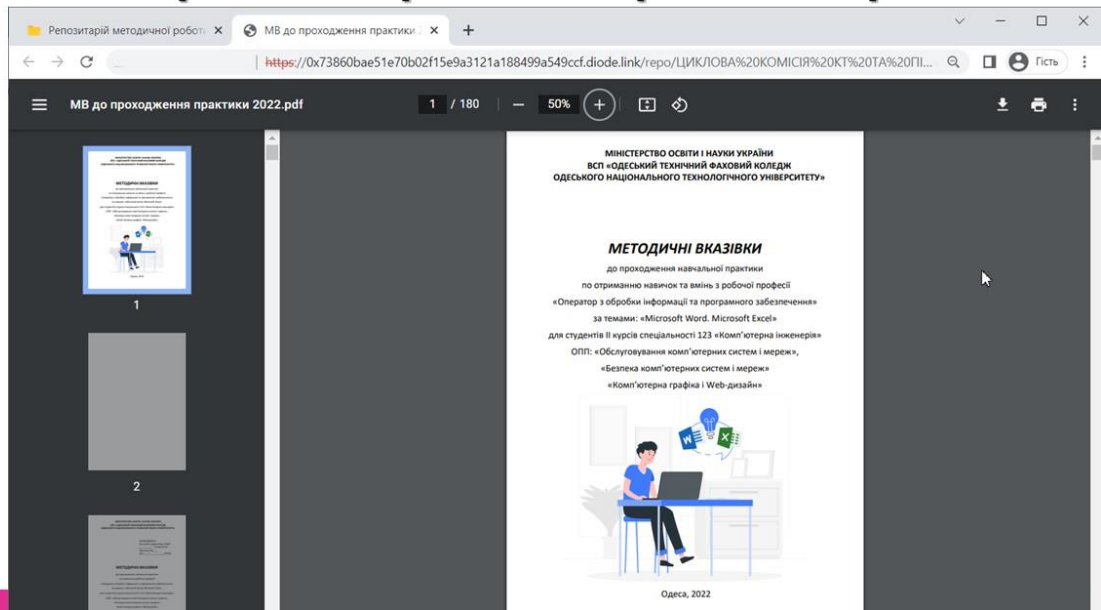
Скріншоти роботи з репозитарієм



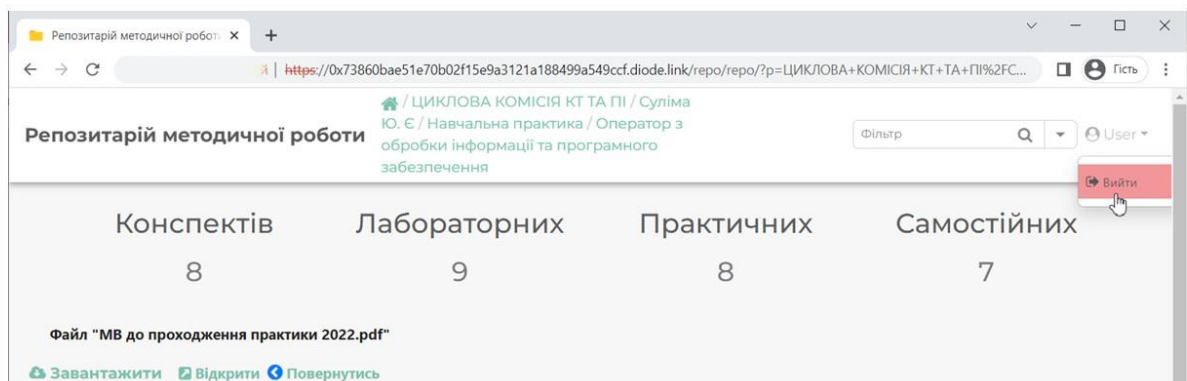
Скріншоти роботи з репозитарієм



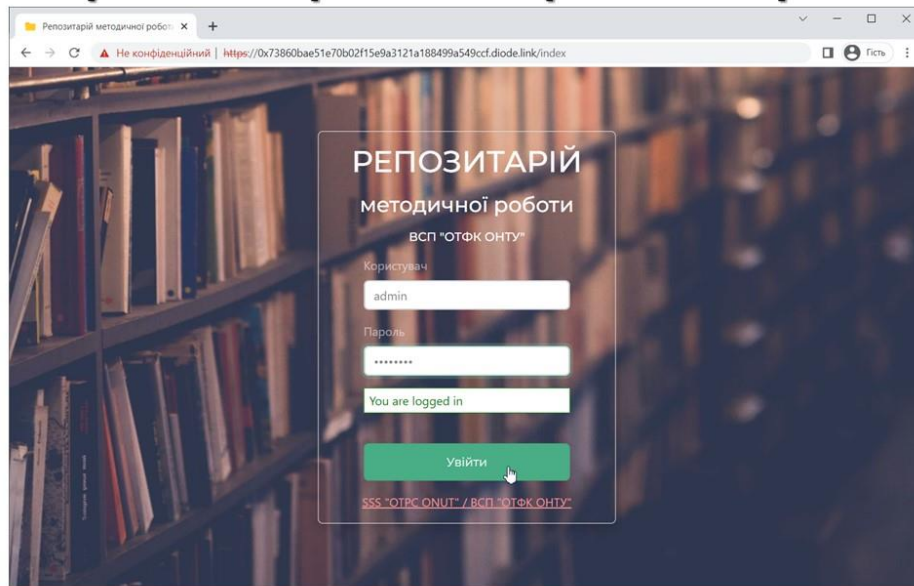
Скріншоти роботи з репозитарієм



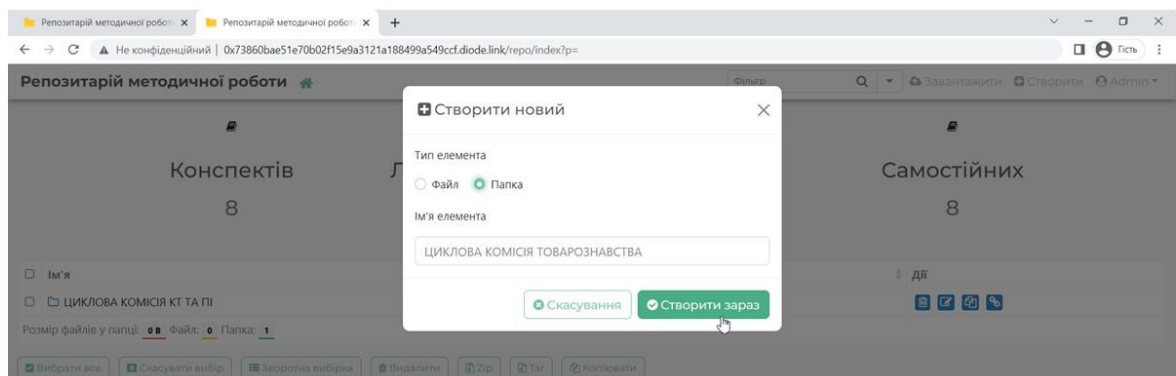
Скріншоти роботи з репозитарієм



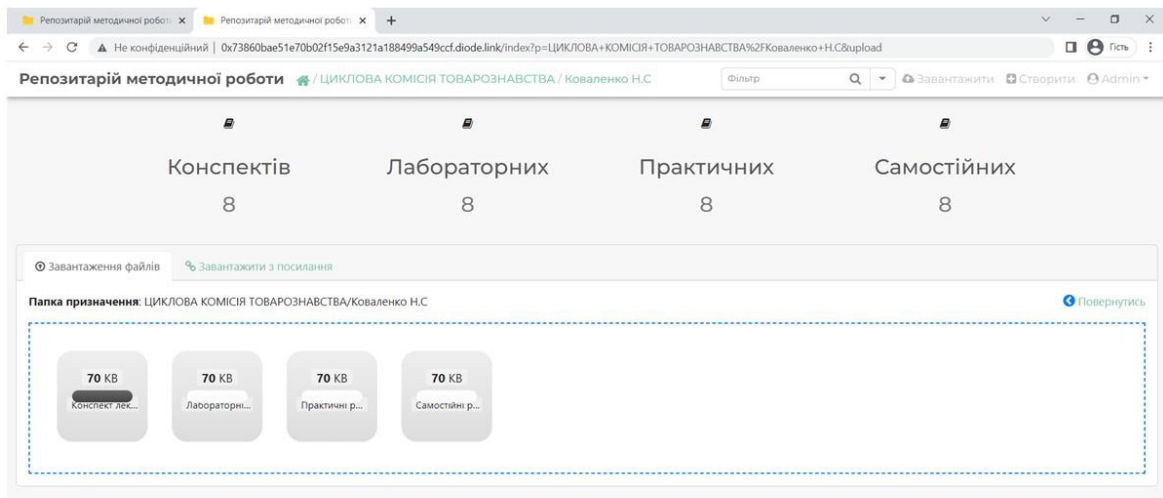
Скріншоти роботи з репозитарієм



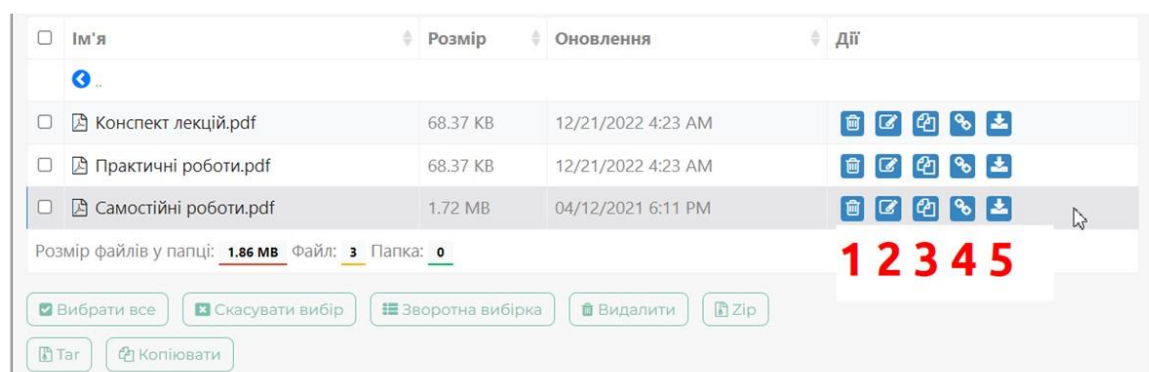
Скріншоти роботи з репозитарієм



Скріншоти роботи з репозитарієм



Скріншоти роботи з репозитарієм



«1» – «Видалити»; «2» – «Перейменувати»; «3» – «Копіювати»;
«4» – «Посилання»; «5» – «Завантажити»

В роботі було проведено економічний розрахунок:

- ✓ загальні витрати ($B_3 = B_p + B_n$) на створення репозитарію складають:
- ✓ де B_p – витрати на розробку репозитарію
- ✓ B_n – постійні витрати (витрати на впровадження та експлуатацію репозитарію)

$$B_3 = 17214,69 + 11940,00 = 29154,69 \text{ грн.}$$

- ✓ Запровадження репозитарію методичної роботи для ВСП «ОТФК ОНТУ» дозволить вивільнити 87000,0 грн. на рік за рахунок скорочення витрат на друк навчально-методичних комплексів. Коефіцієнт економічної ефективності розробки складає ≈ 3 , що є достатнім показником.

В розділі з охорони праці висвітлено питання:

- ✓ Організації безпечних умов праці працівника на робочому місці;
- ✓ Основних вимог безпеки до мікроклімату виробничих приміщень, освітлення, шуму, вібрації, ультразвуку, інфразвуку, виробничих випромінювань, небезпека ураження електричним струмом;
- ✓ Пожежної безпеки,

Оскільки саме ці питання мають першочергове значення при організації робочого місця фахівця, яке передбачає роботу з персональним комп'ютером (ПК).

РЕЦЕНЗІЯ

на дипломний проект (роботу) здобувача (здобувачки) освіти
відділення комп'ютерних систем

Бабіч Вікторії Сергіївни

(прізвище, ім'я та по батькові)

Спеціальність 123 «Комп'ютерна інженерія»

Освітня програма «Комп'ютерна графіка і Web-дизайн»

Керівник дипломного проекту (роботи) _____

Суліма Юлія Євгеніївна

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) _____

«Створення репозитарію методичної роботи для ВСП «ОТФК ОНТУ»

Обсяг розрахунково-пояснювальної записки 88 сторінок

Обсяг графічної (презентаційної) частини 27 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню
Створений дипломний проект повністю відповідає технічному завданню на проектування. Розроблений репозитарій методичної роботи може використовуватися за призначенням та має належну функціональність.

б) характеристика виконання кожного розділу дипломного проекту (роботи) _____

Всі розділи пояснювальної записки виконані на належному рівні та в достатньому обсязі, доповнені докладними коментарями та ілюстраціями, із дотриманням вимог ЄСКД.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту (роботи) _____

Загалом пояснювальна записка виконана грамотно, цілісно, справляє гарне враження. Всі питання, які мали бути розглянуті в процесі проектування, докладно висвітлені в пояснювальній записці. Пояснювальна записка та графічна частина оформлені якісно та справляють позитивне враження.

ВІДГУК

керівника на дипломний проект здобувачки освіти
відділення комп'ютерних систем

Бабіч Вікторії Сергіївни

(прізвище, ім'я та по батькові)

Спеціальність 123 «Комп'ютерна інженерія»

Освітня програма «Комп'ютерна графіка і web-дизайн»

Тема дипломного проекту (роботи) _____

«Створення репозитарію методичної роботи для ВСП «ОТФК ОНТУ»

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання роботи (графічного матеріалу і розрахунково-пояснювальної записки) Пояснювальна записка виконана на 88 сторінках формату А4 та доповнена графічною презентацією. Дипломний проект виконано на належному рівні та у достатньому обсязі, із дотриманням вимог ЄСКД. Всі питання, які необхідно було розглянути, докладно описані в дипломному проекті. Створений веб-проект репозитарію відповідає технічному завданню на проєктування та може використовуватися за призначенням.

б) самостійність роботи над проектом: _____

Бабіч В.С. працювала над виконанням ДП самостійно та отримувала консультації від керівника проекту по складним технічним питанням. Під час виконання дипломного проекту дипломниця активно користувалася технічною літературою та здійснювала результативний пошук необхідної інформації в мережі Internet.

в) теоретична підготовка випускниці Теоретична підготовка випускниці Бабіч В.С. знаходиться на належному рівні, що відповідає вимогам, які ставляться до молодшого спеціаліста з комп'ютерної графіки та веб-дизайну.

г) вміння розв'язувати виробничі та конструкторські питання _____

Випускниця Бабіч В.С. вмiє розв'язувати такі питання, що підтвердила під час роботи над дипломним проектом, під час виконання та оформлення якого вона користувалася засобами веб-розробки високого рівня, інтернет-сервісами і технологіями, сучасними web-редакторами, текстовими і графічними редакторами.

Оцінка розрахункової частини _____	5 (відмінно)
Оцінка графічної частини _____	5 (відмінно)
Загальна оцінка _____	5 (відмінно)

Прізвище, ім'я, по батькові керівника дипломного проекту _____

Суліма Юлія Євгеніївна

Місто роботи і посада керівника дипломного проекту _____
викладач вищої категорії циклової комісії комп'ютерних технологій та програмної інженерії ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету»

Підпис _____
«08» червня 2023 р.

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Бабіч Вікторія Сергіївна,
здобувачка освіти гр. 4КГ-06, та

Суліма Юлія Євгенівна,
керівник дипломного проекту,

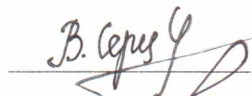
не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи молодшого спеціаліста на тему:

**«Створення репозитарію методичної роботи для ВСП «ОТФК ОНТУ»
(автор роботи – Бабіч В.С., керівник роботи – Суліма Ю.Є.)**

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2023 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

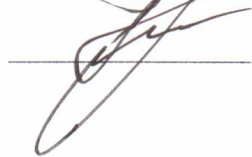
Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи, і даємо згоду на обробку персональних даних.

Виконавець



/ Бабіч В.С. /

Керівник



/ Суліма Ю.Є. /

« 09 » червня 20 23 р.

Ім'я користувача:
Наталія Вікторівна Копусь

ID перевірки:
1015474221

Дата перевірки:
07.06.2023 09:17:29 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
07.06.2023 09:22:45 EEST

ID користувача:
100011688

Назва документа: 4КГ-06_Вікторія_Бабіч

Кількість сторінок: 84 Кількість слів: 12816 Кількість символів: 97547 Розмір файлу: 15.63 MB ID файлу: 1015132373

11.7% Схожість

Найбільша схожість: 4.08% з Інтернет-джерелом (<http://vpr.sm.gov.ua/index.php/uk/2-uncategorised/505-13-dprch-upra>).

11.7% Джерела з Інтернету

914

Сторінка 86

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

4