

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

**Навчально-науковий інститут прикладного системного аналізу
Кафедра системного проектування**

«На правах рукопису»
УДК 004.75: 681.3

До захисту допущено:
Завідувач кафедри
_____ Вадим МУХІН
« ____ » _____ 2024 р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою
“Інтелектуальні сервіс-орієнтовані розподілені обчислювання”
зі спеціальності 122 "Комп'ютерні науки"
на тему: «Розробка WordPress плагіна для керування файловою системою»**

Виконав (-ла):
студент (-ка) II курсу, групи ДА-21мп
Таран Анастасія Ігорівна _____

Науковий керівник:
доцент, к.т.н.
Кисельов Геннадій Дмитрович _____

Консультант з нормоконтролю:
доцент, к.т.н.
Кирюша Богдан Анатолійович _____

Рецензент:
доцент кафедри АПЕПС, к.т.н.
Шаповалова Світлана Ігорівна _____

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.
Студент (-ка) _____

Київ – 2024

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра системного проектування

Рівень вищої освіти – другий (магістерський)

Спеціальність – 122 "Комп'ютерні науки"

Освітньо-професійна програма – "Інтелектуальні сервіс-орієнтовані розподілені обчислювання"

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Вадим МУХІН

«___» _____ 2024 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Таран Анастасії Ігорівні

1. Тема роботи «Розробка WordPress плагіна для керування файловою системою», науковий керівник дисертації Кисельов Геннадій Дмитрович, к.т.н., доцент,

затверджені наказом по університету від «___» _____ 20__ р. № _____

2. Термін подання студентом роботи – 25 грудня 2023 р.

3. Об'єкт дослідження - WordPress плагін, для керування файловою системою веб-сайту.

4. Предмет дослідження:

1. Поточний стан файлової системи WordPress: аналіз і огляд наявних можливостей керування файлами в попередньо розробленому плагіні WordPress.
2. Проблеми та вимоги: ідентифікація проблем, з якими користувачі можуть зіштовхнутися під час роботи з попередньо розробленим плагіном WordPress, а також встановлення вимог до нового функціоналу для покращення керування файлами.
3. Реалізація та тестування плагіна: вивчення технічних аспектів для реалізації нового функціоналу плагіна та проведення його тестування для перевірки функціональності та надійності.
5. Перелік завдань, які потрібно розробити:
 1. Аналіз поточного стану плагіна.
 2. Розробка користувацького інтерфейсу.
 3. Реалізація нових функцій плагіна.
 4. Тестування та налагодження плагіна.
6. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо):
 1. Презентація до захисту роботи.
7. Перелік публікацій:
 1. Збірник доповідей II науково-практичної конференції «Системні науки та інформатика», 4–8 грудня 2023 р. Київ. – К., НН ІПСА КПІ ім. ІгоряСікорського, 2023. – 416 с.
8. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Стартап проект	Кисельов Г.Д.		

9. Дата видачі завдання – 25 червня 2023 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Захист звіту з переддипломної практики	25.10.2023	
2	Виявлення проблем та вимог файлової системи WordPress	10.11.2023	
3	Проектування плагіна	15.11.2023	
4	Реалізація та тестування плагіна	25.11.2023	
5	Оцінка та документування результатів	10.12.2023	
6	Захист дисертації	25.12.2023	

Студент (-ка)

А.І. Таран

Науковий керівник

Г.Д. Кисельов

РЕФЕРАТ

магістерської дисертації Таран Анастасії Ігорівни на тему «Розробка WordPress плагіна для керування файловою системою»

Робота виконана на 105 сторінках, містить 50 ілюстрацій, 23 таблиці, 1 додаток. При підготовці використовувалась література з 13 джерел.

Актуальність дослідження. Зі зростанням обсягу контенту на веб-сайтах користувачі WordPress зіштовхуються з необхідністю ефективного та зручного управління своєю файловою системою. На сьогоднішній день у такому прогресивному інструменту веб-розробки, як WordPress, все ще відсутня можливість «розумного» керування файлами та структурою їх відображення [1]. Розробка плагіна, спрямованого на спрощення процесу управління файловою системою в даній системі керування вмістом, визнається як важлива та актуальна, оскільки відповідає потребам користувачів.

Мета та задачі дослідження. Метою магістерської дисертації є вдосконалення та розширення функціоналу існуючого WordPress плагіна для управління файловою системою. Вихідний плагін, який був розроблений під час виконання бакалаврської дипломної роботи, вже обладнаний базовим функціоналом, таким як завантаження файлів, введення інформації про автора та рік створення документів, вибір категорії до якої відносяться ресурси та виведення цих даних на веб-сайт. Його існуючий потенціал став основою для подальших досліджень та розвитку, спрямованих на розширення можливостей плагіна та адаптацію його функціоналу до різноманітних потреб користувачів.

Об'єкт досліджень - WordPress плагін, для керування файловою системою веб-сайту.

Предмет досліджень - процес створення вдосконаленого WordPress плагіна для ефективного управління файловою системою веб-сайтів, який включає в себе: аналіз і огляд існуючих рішень, визначення вимог до функціоналу WordPress плагіна для керування файловою системою, розробка сучасного дизайну для покращення користувацького досвіду, реалізація нового функціоналу та тестування плагіна.

Наукова новизна. Новизна магістерської дисертації полягає в представленні наукового і технічного внеску у сферу управління файловою системою в екосистемі WordPress, шляхом розробки та вдосконалення плагіна. Унікальність цієї роботи виявляється в інтеграції інноваційних підходів для оптимізації та покращення функціональності файлової системи, забезпечуючи високу ефективність та зручність використання для всіх користувачів.

Практична застосовність. Дана розробка матиме практичну застосовність в різноманітних освітніх установах, компаніях та організаціях, де потрібно взаємодіяти з великим обсягом файлів. Також цей плагін WordPress стане важливим інструментом для управління файловою системою на сайті кафедри системного проектування Навчально-наукового інституту прикладного системного аналізу НТУУ "КПІ ім. Ігоря Сікорського". Зокрема, цей інструмент можна ефективно використовувати для відображення інформації про дипломні роботи на сторінці «[Дипломні проекти](#)», де він дозволить зручно та організовано представляти дані про роботи студентів [1].

Ключові слова: CMS, плагін WordPress, управління файловою системою, PHP, Shortcode.

ABSTRACT

the master's thesis of Anastasiia Taran Ihorivna on "Development of a WordPress plugin for file system management."

The work is completed on 105 pages, contains 50 illustrations, 23 tables, 1 appendix. Literature from 13 sources was used in the preparation.

Relevance of the research. With the increasing volume of content on websites, WordPress users face the necessity of efficient and convenient management of their file system. Currently, in this progressive web development tool, WordPress, the capability for "smart" file management and the structure of their display is still lacking [1]. Developing a plugin aimed at simplifying the file system management process in this content management system is recognized as important and relevant, as it meets the users' needs.

Purpose and objectives of the research. The purpose of the master's thesis is to enhance and expand the functionality of the existing WordPress plugin for file system management. The initial plugin, developed during the completion of the bachelor's thesis, is already equipped with basic functionality such as file uploads, entering information about the author and the year of document creation, selecting categories to which resources belong, and displaying this data on the website. Its existing potential serves as the foundation for further research and development, aimed at expanding the plugin's capabilities and adapting its functionality to diverse user needs.

Object of research - WordPress plugin for managing the website's file system.

Subject of research - the process of creating an improved WordPress plugin for effective file system management on websites, including: analysis and review of existing solutions, defining requirements for the functionality of the WordPress plugin for file system management, designing a modern interface to enhance user experience, implementing and testing the plugin.

Scientific Novelty. The novelty of the master's thesis lies in presenting a scientific and technical contribution to the field of file system management within the WordPress ecosystem through the development and enhancement of a plugin. The uniqueness of this work is evident in the integration of innovative approaches to optimize and improve the functionality of the file system, ensuring high efficiency and user convenience for all users.

Practical applicability. This development will have practical applicability in various educational institutions, companies, and organizations where interaction with a large volume of files is necessary. Additionally, this WordPress plugin will become an essential tool for managing the file system on the website of the Department of System Design of the Educational and Scientific Institute of Applied System Analysis at the Igor Sikorsky Kyiv Polytechnic Institute. Specifically, this tool can be effectively used to display information about diploma projects on the "Diploma Projects" page, where it will allow for a convenient and organized presentation of data on students' works [1].

Keywords: CMS, WordPress plugin, file system management, PHP, Shortcode.

ЗМІСТ

ВСТУП.....	12
1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ У СФЕРІ УПРАВЛІННЯ ФАЙЛОВОЮ СИСТЕМОЮ WORDPRESS.....	14
1.1 Огляд існуючих рішень.....	14
1.1.1 File Manager.....	14
1.1.2 Real Media Library.....	17
1.1.3 FileBird.....	18
1.2 Порівняльний аналіз плагінів.....	20
1.3 Висновки до розділу.....	22
2 АНАЛІЗ ПРОБЛЕМ ІСНУЮЧОГО ПЛАГІНА.....	23
2.1 Виявлення слабких місць та обмежень.....	23
2.2 Висновки до розділу.....	27
3 ПРОЕКТУВАННЯ ПЛАГІНА.....	29
3.1 Визначення основних функціональних можливостей плагіна.....	29
3.2 Розробка дизайну плагіна.....	33
3.3 Висновки до розділу.....	39
4 РОЗРОБКА ТА ТЕСТУВАННЯ ПЛАГІНА.....	41
4.1 Опис процесу програмування.....	41
4.1.1 Активація плагіна.....	42
4.1.2 Оновлення інтерфейсу плагіна.....	45
4.1.3 Інтеграція плагіна з бібліотекою Font Awesome.....	47
4.1.4 Реалізація динамічних полів для додавання додаткової інформації про файли.....	48

4.1.5 Реалізація функціоналу для різного відображення файлів.....	50
4.1.6 Розширення функціоналу виведення файлів на веб-сайт.....	54
4.2 Проведення тестування.....	55
4.3 Висновки до розділу.....	65
5 РОЗРОБКА СТАРТАП-ПРОЕКТУ «РОЗРОБКА WORDPRESS ПЛАГІНА ДЛЯ КЕРУВАННЯ ФАЙЛОВОЮ СИСТЕМОЮ».....	67
5.1 Опис ідеї проекту.....	67
5.2 Технологічний аудит ідеї проекту.....	70
5.3 Аналіз ринкових можливостей запуску стартап-проекту.....	71
5.4 Розробка ринкової стратегії проекту.....	79
5.5 Розробка маркетингової програми.....	83
5.6 Висновки до розділу.....	86
ВИСНОВКИ.....	88
ПЕРЕЛІК ПОСИЛАНЬ.....	90
ДОДАТОК А.....	92

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

CMS - Content Management System - система керування вмістом веб-сайтів.

WP – WordPress - система керування вмістом з відкритим вихідним кодом, написана на PHP.

HTML - Hyper Text Markup Language - це стандартизована мова розмітки документів для перегляду веб-сторінок у браузера.

CSS - Cascading Style Sheets - це спеціальна мова стилю сторінок що використовується для опису їх зовнішнього вигляду.

PHP - Hypertext Preprocessor - це широко використовувана мова сценаріїв загального призначення із відкритим вихідним кодом.

JS - Java Script - це динамічна, об'єктно-орієнтована прототипна мова програмування.

AJAX - Asynchronus Javastripet and XML - технологія, що дозволяє здійснювати асинхронний обмін даними між клієнтом та сервером без потреби перезавантажувати сторінку.

ВСТУП

У сучасному інформаційному суспільстві, де віртуалізація та інтернет-технології набувають все більшого значення, ефективне управління вмістом сайтів стає ключовою складовою успіху для багатьох організацій, користувачів та розробників. В цьому контексті системи керування контентом (CMS) - програмне забезпечення, котре дозволяє редагувати вміст веб-сайтів за допомогою зручного інтерфейсу користувача та забезпечує публікацію контенту для відвідувачів разом із зручними засобами навігації [2] - відіграють важливу роль у полегшенні управління веб-проектами.

Однією з найпоширеніших та ефективних CMS є WordPress. Популярність WordPress в сучасному цифровому світі вражає: більше третини всіх веб-сайтів (починаючи від візиток і лендінгів і закінчуючи онлайн-магазинами та корпоративними ресурсами) базується на цій CMS. Серед них можна знайти як невеликі стартапи, так і проекти зі світовими іменами. Своє безперечне лідерство дана CMS утримує завдяки низці якостей - мультимовності, універсальності, доступності та багатофункціональності [3]. WordPress має програмне забезпечення з відкритим кодом, що дає змогу користувачам самостійно та безкоштовно вносити зміни, розширені модифікації [4]. Система дозволяє встановлювати додаткові плагіни (безкоштовні чи платні), цим самим розширюючи можливості WordPress [4].

Однак, разом з розширенням можливостей WordPress, виникають нові виклики, зокрема, проблема ефективного керування файловою системою. За замовчуванням, WordPress пропонує базовий функціонал для завантаження файлів, але відсутність зручного та гнучкого інтерфейсу для організації та взаємодії з файлами стає обмеженням для великих та складних веб-проектів.

Таким чином, розробка WordPress плагіна для ефективного управління файловою системою стає надзвичайно актуальною в сучасному веб-середовищі. З урахуванням того, що WordPress займає значну частку ринку і використовується для створення різноманітних типів веб-сайтів, оптимізація управління файловою системою стає ключовою задачею.

Така розробка спрямована на полегшення роботи як звичайних користувачів, так і веб-розробників. Шляхом створення функціонального та зручного плагіна мета полягає в поліпшенні продуктивності та оптимізації робочих процесів, що в свою чергу сприятиме покращенню якості веб-сайтів, розроблених на базі WordPress. Забезпечення зручного та ефективного управління файлами в рамках цієї CMS визначається як ключовий аспект для досягнення високого рівня функціональності та задоволення потреб користувачів.

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ У СФЕРІ УПРАВЛІННЯ ФАЙЛОВОЮ СИСТЕМОЮ WORDPRESS

З метою вдосконалення та розширення функціоналу файлової системи WordPress необхідно ретельно проаналізувати існуючі підходи та рішення в даній області.

Даний розділ присвячений огляду популярних плагінів та інших інструментів, призначених для управління файлами на платформі WordPress. Це дозволить нам виокремити сильні та слабкі сторони кожного розглянутого інструменту, щоб визначити ті аспекти, які варто врахувати при подальшій розробці та вдосконаленні плагіна для керування файловою системою WordPress.

1.1 Огляд існуючих рішень

Важливий крок у вивченні сучасних підходів до управління файловою системою WordPress — це детальний огляд популярних плагінів та інших інструментів, розроблених для вирішення цих завдань. У рамках вивчення сучасних підходів до управління файловою системою WordPress, було обрано три найпопулярніших плагіни, які буде ретельно проаналізовано. Проведений огляд дозволить визначити ключові характеристики, основні функції та оцінити ефективність цих плагінів у реальних умовах використання.

1.1.1 File Manager

File Manager (рис. 1.1) є одним з найбільш популярних та зручних плагінів для управління файлами у WordPress.

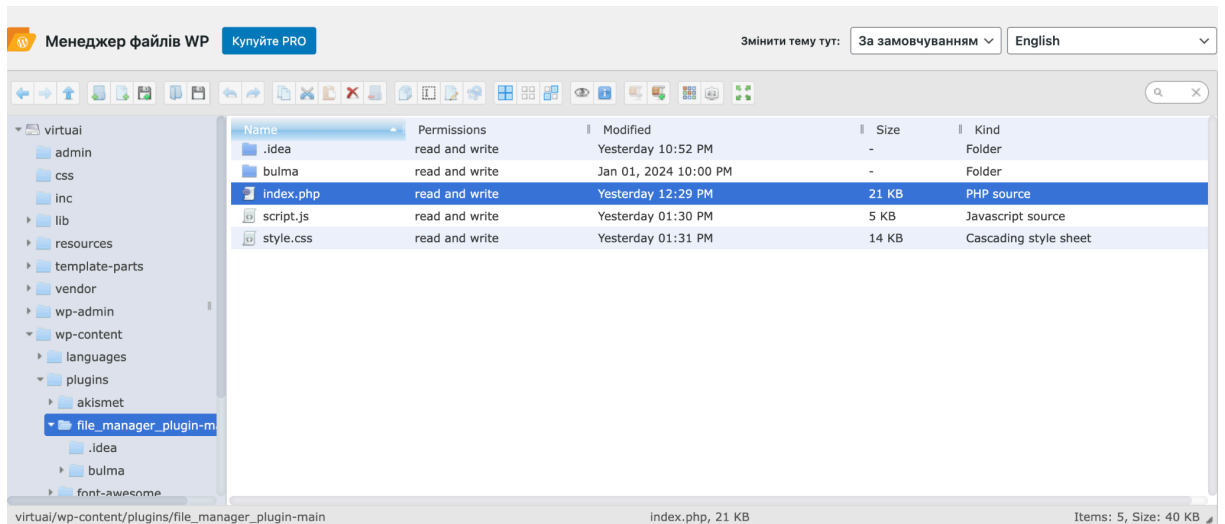


Рисунок 1.1 – Інтерфес плагіна File Manager

Плагін дозволяє налаштовувати та виконувати багато операцій з вашими файлами [5]:

1. Він надає можливість копіювати, вставляти, архівувати, завантажувати, видаляти та вивантажувати необхідні документи з вашого WordPress проекту.
2. Даний плагін підтримує попередній перегляд для аудіо, відео, mp3 файлів (рис. 1.2).
3. Плагін має можливість рецензування коду та має вбудований Синтаксичний перевірник, що дозволяє переглядати код перед збереженням файлів.
4. File Manager володіє функціоналом пошуку файлів, що може бути корисним для користувачів, які шукають ефективний спосіб доступу до конкретних документів серед розмаїття файлової структури.

аспектом для користувачів, які шукають зручний інструмент для організації та навігації через великі обсяги даних.

Отже, якщо мета користувача - створення структурованого файлового архіву на веб-сайті, можливо, варто розглянути інші рішення або розширення, які надають потрібний функціонал.

1.1.2 Real Media Library

Real Media Library (рис. 1.3) є плагіном для керування медіа-бібліотекою в системі керування вмістом WordPress. Плагін дозволяє користувачам більш ефективно керувати великим обсягом медіафайлів, роблячи робочий процес організації та пошуку файлів більш приємним і зручним.

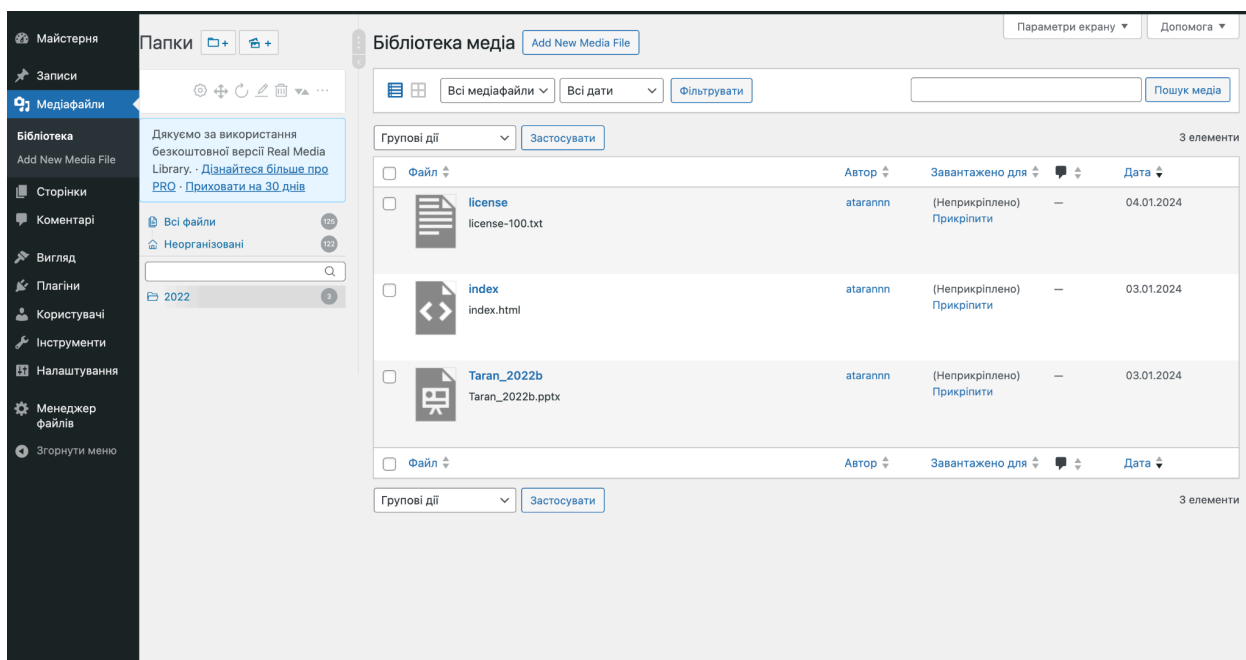


Рисунок 1.3 – Інтерфес плагіна Real Media Library.

Основні функції Real Media Library включають у себе [6]:

1. Створення папок та підпапок для організації файлів.
2. Перетягування та вибіркове переміщення файлів між папками.
3. Наявність фільтрації, пошуку та сортування файлів.

4. Можливість виконання масових операцій (видалення, переміщення, перейменування) для групи файлів.
5. Попередній перегляд зображень безпосередньо в медіа-бібліотеці.
6. Додаткові налаштування для керування виглядом та функціоналом.

Порівняно з попереднім плагіном, Real Media Library має більш сучасний інтерфейс та забезпечує інтуїтивно зрозумілий доступ до всіх необхідних функцій. Плагін дозволяє ефективно організовувати файли та створювати структури папок без необхідності занурюватися в детальний огляд документації для розуміння робочих процесів.

Проте, Real Media Library, хоч і є зручним інструментом для управління файлами в WordPress, але має досить обмежений функціонал для додавання додаткової інформації про файли. Плагін дозволяє додавати лише короткий опис і це обмеження впливає на можливості ефективної фільтрації та категоризації файлів. Зокрема, фільтрація можлива лише за назвою файлу та датою його завантаження в файлову систему WordPress, що обмежує можливості користувачів в зручному пошуку та організації файлів.

Також, аналогічно попередньому плагіну, Real Media Library не має функціоналу для додавання файлів на веб-сайт, так як його головна мета застосування — це ефективна організація та структурування медіафайлів.

1.1.3 FileBird

FileBird - це потужний інструмент для управління файлами в системі керування вмістом WordPress (рис. 1.4), який надає користувачам повний контроль над організацією та структуризацією файлів.

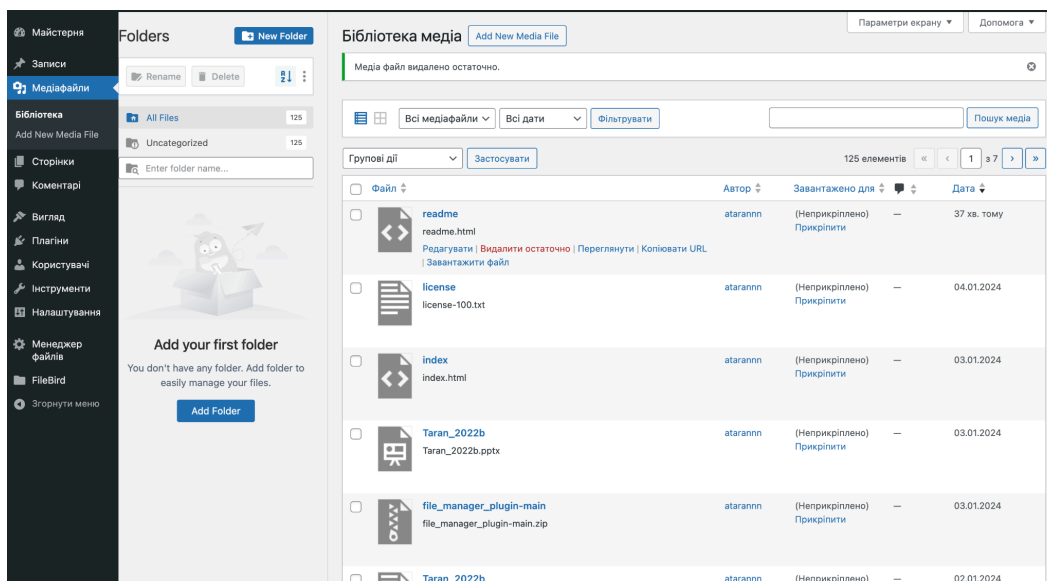


Рисунок 1.4 – Інтерфес плагіна FileBird.

Основні функції FileBird включають [7]:

1. Створення папок та підпапок для організації файлів.
2. Перетягування та вибіркове переміщення файлів між папками.
3. Наявність фільтрації, пошуку та сортування файлів.
4. Можливість виконання масових операцій (видалення, переміщення, перейменування) для групи файлів.
5. Попередній перегляд зображень безпосередньо в медіа-бібліотеці.
6. Додаткові налаштування для керування виглядом та функціоналом.
7. FileBird підтримує вбудовані галереї та можливість вставляти галереї у вміст сторінок чи дописів.

Даний плагін має легкий у використанні інтерфейс, який суттєво спрощує управління файлами на WordPress сайті, який досить схожий на інтерфейс плагіна Real Media Library. Також обидва плагіни володіють схожими функціональностями.

Проте, FileBird також не надає додаткового функціоналу для деталізації інформації про файли. Як і Real Media Library, плагін FileBird дозволяє лише додавати короткий опис файлів, обмежуючи можливості ефективної фільтрації та категоризації. Отже, обидва плагіни мають обмеження у розширенні інформації про файли, що може впливати на організацію та пошук файлів.

Також, так як і його попередники, FileBird не надає можливості швидкого додавання файлів безпосередньо на сам веб-сайт.

1.2 Порівняльний аналіз плагінів

Вибір критеріїв для оцінки плагінів WordPress для управління файловою системою базується на необхідності забезпечити користувачам максимально повний та зручний функціонал.

Обґрунтування вибору критеріїв для оцінювання:

1. Зручність та інтуїтивно зрозумілий інтерфейс: важливо мати зручний інтерфейс для покращення користувацького досвіду.
2. Можливість додавання додаткової інформації: додаткова інформація про файли допомагає у категоризації та фільтрації, полегшуючи організацію файлів.
3. Підтримка пошуку: дозволяє швидко організовувати файли.
4. Підтримка пошуку: дозволяє швидко знаходити необхідні файли.
5. Підтримка фільтрації: забезпечує можливість точного визначення файлів за різними критеріями.
6. Можливість додавання файлів на веб-сайт: важливо для безпосереднього управління вмістом веб-сайту.

Аналіз можливостей плагінів WordPress для управління файловою системою продемонстровано в табл. 1.1.

Таблиця 1.1.

Критерії оцінювання	File Manager	Real Media Library	FileBird
Зручність та інтуїтивно зрозумілий інтерфейс	-	+	+
Можливість додавання докладної інформації про файли	-	-	-
Підтримка сортування файлів	+	+	+
Можливість пошуку файлів	+	+	+
Підтримка фільтрації за різними критеріями	-	-	-
Можливість додавання файлів на веб-сайт	-	-	-

Усі три плагіни можуть використовуватися для структурування файлів у WordPress, але їхні можливості додавання файлів та розширення описів обмежені. Вибір серед них слід робити в залежності від конкретних вимог та потреб користувача.

1.3 Висновки до розділу

У даному розділі був проведений детальний огляд трьох популярних плагінів WordPress для управління файловою системою: File Manager, Real Media Library і FileBird. Проведений аналіз цих плагінів дозволив визначити ключові функціональні та інтерфейсні особливості кожного плагіна.

Отримані дані стануть основою для подальшої розробки та вдосконалення плагіна WordPress для управління файловою системою, який був розроблений у рамках виконання бакалаврської дипломної роботи. Враховуючи ключові характеристики та функціональні особливості найпопулярніших плагінів, ми зможемо адаптувати та розширити функціонал нашого продукту, для більш ефективного та зручного використання.

Результати аналізу допоможуть у визначенні пріоритетних напрямків розвитку та вдосконалення плагіна, щоб відповідати найвищим стандартам управління файловою системою WordPress.

2 АНАЛІЗ ПРОБЛЕМ ІСНУЮЧОГО ПЛАГІНА

Аналіз проблем плагіна WordPress для управління файловою системою, який був розроблений у рамках виконання бакалаврської дипломної роботи є важливою складовою роботи над вдосконаленням та розвитком продукту.

У цьому розділі буде проведений детальний аналіз недоліків та викликів, з якими стикається існуючий плагін. Це дасть змогу точно визначити області для подальших вдосконалень та розробити стратегію виправлення виявлених недоліків.

2.1 Виявлення слабких місць та обмежень

Існуючий плагін, надає базовий, але важливий функціонал для управління файловою системою WordPress (рис 2.1).

Однією з ключових можливостей даного плагіна є функція завантаження файлів до файлової системи WordPress. Вона надає користувачам зручний та швидкий спосіб додавання різноманітних документів, забезпечуючи їхню легку доступність для подальшого використання. Ця функція є важливою, оскільки дозволяє зберігати та організовувати файли в одному централізованому місці, що сприяє систематизації та ефективному управлінню контентом. Крім того, такий інструмент спрощує рутинні завдання, пов'язані із завантаженням та роботою з різноманітними типами файлів.

Також даний плагін пропонує користувачам можливість додавати деталізовану інформацію про файли, таку як, рік створення документу та інформацію про автора. Це робить процес організації та класифікації файлів більш зручним та ефективним.

Додатково, плагін дозволяє призначати категорії для файлів, що робить його гнучким у використанні. Користувач може структурувати свої роботи з урахуванням їхнього призначення, вибираючи категорії, такі як бакалаврські, магістерські чи аспірантські роботи. Це значно полегшує навігацію та пошук необхідних файлів, особливо в разі наявності великого обсягу документів у медіа-бібліотеці.

Дана інформація є важливою для більш деталізованої організації файлів. Однак важливо відзначити, що поля для цієї інформації є статичними, і їхню кількість не можна змінювати. Для поліпшення гнучкості та індивідуалізації використання плагіна, було б корисно мати можливість динамічно змінювати кількість додаткової інформації, яку користувач бажає додати до своїх файлів.

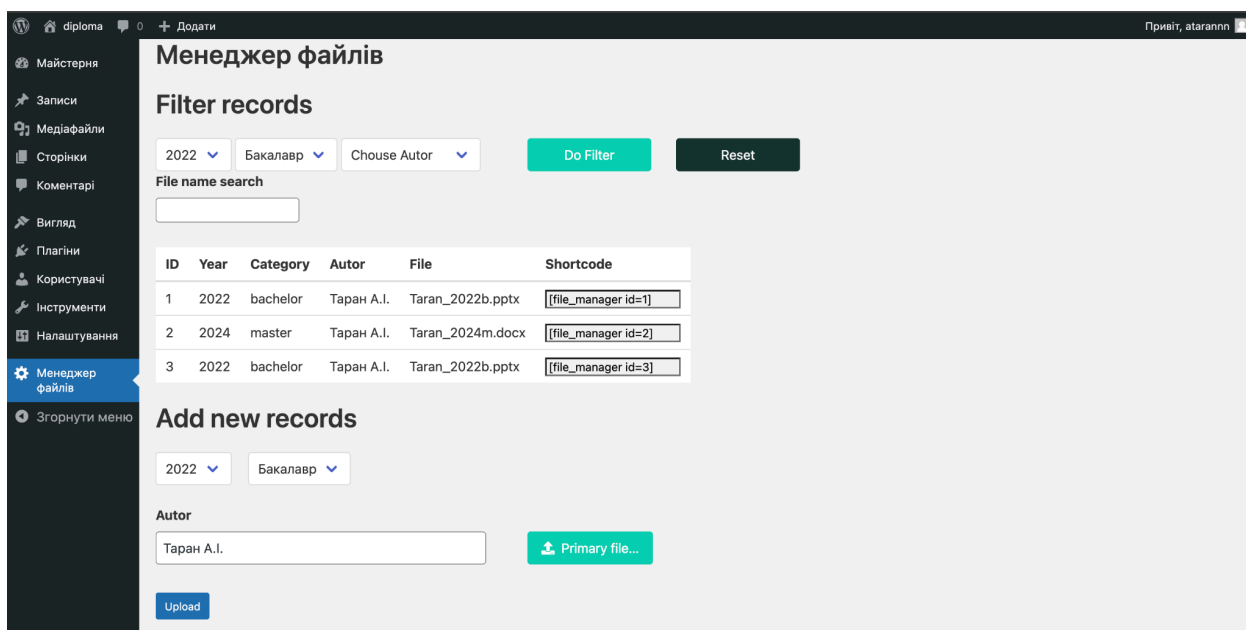


Рисунок 2.1 – Інтерфейс існуючого плагіна WordPress для керування файловою системою.

Функція фільтрації, котру має плагін, вже надає широкі можливості для користувачів в управлінні файловою системою WordPress. Вона дозволяє фільтрувати файли за роком, автором, категорією (рис. 2.2) та виконувати пошук за назвою файлу.

Слід відзначити, що наявність цього розширеного функціоналу фільтрації вже робить плагін потужним і зручним для користувачів. Це забезпечує ефективний та гнучкий спосіб організації та взаємодії з файловою системою, спрощуючи завдання пошуку необхідних файлів навіть у великих медіа-бібліотеках.

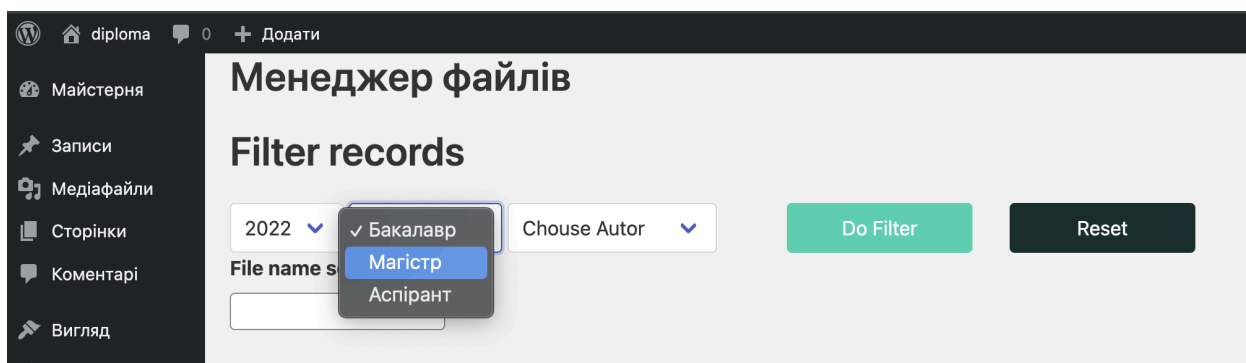


Рисунок 2.2 – Процес фільтрації файлів.

Інтерфейс даного плагіна використовує стиль CSS фреймворку Bulma - це CSS фреймворк, призначений для легкого створення адаптивних сайтів і веб-інтерфейсів, який надає готові зовнішні компоненти, котрі можна легко впроваджувати у свій код [8]. Застосування Bulma дозволяє забезпечити досить лаконічний та зрозумілий дизайн, який сприяє легкій взаємодії користувача з інтерфейсом.

Проте, у контексті покращення користувацького досвіду та забезпечення плагіна більш сучасним виглядом, рекомендується переглянути та оновити його інтерфейс. Використання актуальних технологій та елементів дизайну може позитивно вплинути на сприйняття плагіна користувачами.

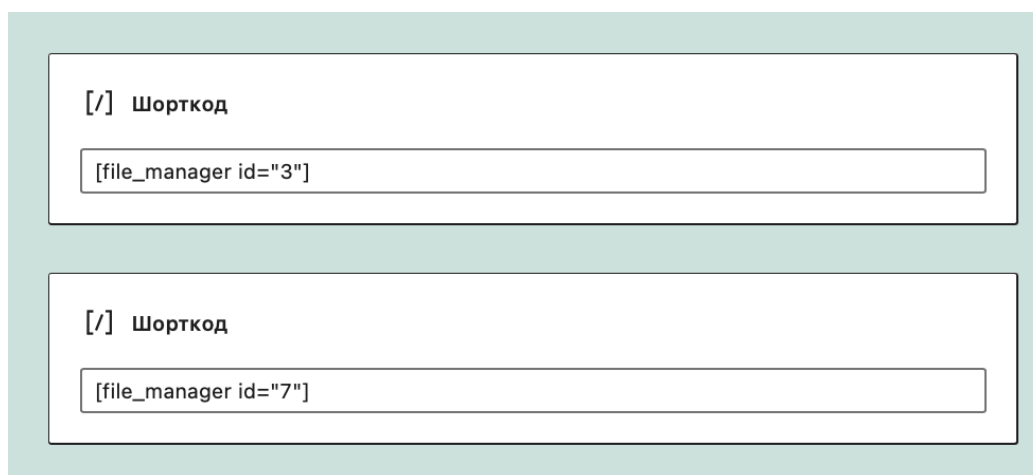
Пропонується впровадити інтеграцію з бібліотекою Font Awesome. Використання іконок з даної бібліотеки в інтерфейсі плагіна може зробити його більш привабливим та інтуїтивно зрозумілим.

Окрім цього, під час оновлення інтерфейсу плагіна та інтеграції з бібліотекою Font Awesome, важливо розглянути можливість динамічної зміни

іконок відповідно до формату файлів. Це дозволить створити більш інформативний та зручний інтерфейс для користувачів.

Можливість вибору формату відображення файлів – списком або плиткою, також може стати досить корисним функціоналом для користувачів. Плагін стане більш гнучким та пристосованим до індивідуальних потреб користувачів. Ця функціональність дозволяє кожному користувачеві налаштовувати інтерфейс плагіна згідно з власними уподобаннями та робочими потребами. Формат списку може бути зручним для швидкого перегляду великої кількості файлів, тоді як формат плитки може бути більш естетично привабливим [9].

Функціонал виведення інформації на сайт за допомогою shortcode є зручним інструментом для представлення файлового контенту користувачам. Він дозволяє зручно впорядковувати та відображати файли на веб-сайті. Зараз існуючий функціонал дозволяє виводити файли відповідно до їхнього id (рис. 2.3 - 2.4). Це є не зовсім зручно у випадках, коли потрібно виводити файли за конкретний рік чи категорію.



The image shows two identical shortcode boxes stacked vertically. Each box has a title bar that says "[/] Шорткод" and a text input field containing the shortcode "[file_manager id="3"]". The top box is for id="3" and the bottom box is for id="7".

Шорткод
[file_manager id="3"]
[file_manager id="7"]

Рисунок 2.3 – Процес додавання документів на веб-сайт за допомогою shortcode.

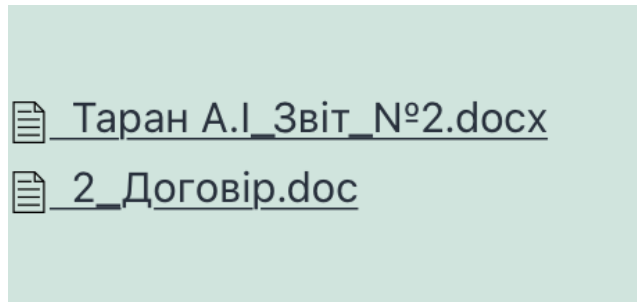


Рисунок 2.4 – Результат додавання документів за допомогою shortcode.

Для подальшого поліпшення цього функціоналу рекомендується розширити можливості виведення файлів через shortcode, додавши параметри за роками та категоріями. Це надасть користувачам більше гнучкості в управлінні та відображенні файлового контенту на сторінках їхнього веб-сайту.

2.2 Висновки до розділу

У даному розділі, був проведений аналіз існуючого плагіна WordPress для керування файловою системою, в процесі якого ми визначили кілька аспектів, які потребують подальшого поліпшення для забезпечення більшої функціональності та зручності в користуванні плагіна. Основні напрямки для подальшого розвитку та вдосконалення продукту включають:

1. Реалізація динамічних полів для додавання додаткової інформації про файли.
2. Оновлення інтерфейсу плагіна.
3. Інтеграція плагіна з бібліотекою Font Awesome.
4. Реалізація функціоналу для різного відображення файлів.
5. Розширення функціоналу виведення файлів на веб-сайт.

Ці покращення дозволять зробити плагін більш адаптивним та універсальним для різних потреб користувачів, а також полегшать роботу з великою кількістю файлів та їхньою організацією.

3 ПРОЕКТУВАННЯ ПЛАГІНА

3.1 Визначення основних функціональних можливостей плагіна

Виходячи з результатів аналізу, проведеного у попередньому розділі, сформовано конкретні вимоги до функціоналу плагіна для управління файловою системою в WordPress:

1. Реалізація динамічних полів для додавання додаткової інформації про файли.

Динамічні поля додаткової інформації є ключовим елементом для розширення функціоналу плагіна, оскільки вони забезпечують користувачам можливість гнучкого та індивідуалізованого налаштування представлення інформації про файли. Важливість цієї функціональності полягає у здатності користувачів адаптувати плагін до своїх конкретних потреб і вимог.

Вимоги до даного функціоналу включають:

- Динамічне додавання та видалення полів: реалізації можливості користувачів додавати та видаляти поля для додаткової інформації. Це забезпечить гнучкість та індивідуальний підхід до організації інформації відповідно до особистих потреб.
- Інтуїтивний інтерфейс налаштувань: забезпечення простого та зрозумілого інтерфейсу для керування полями додаткової інформації, щоб користувачі легко розуміли, як налаштовувати та використовувати ці дані.
- Підтримка великої кількості полів: забезпечення стабільної та продуктивної роботи з великою кількістю полів, щоб користувачі могли деталізувати та організовувати інформацію для кращого управління файлами.

Реалізація динамічних полів додаткової інформації з урахуванням цих аспектів, гарантує користувачам найвищий рівень гнучкості та можливостей в адмініструванні файлів.

2. Оновлення інтерфейсу плагіна.

Велику увагу слід приділити модернізації інтерфейсу плагіна для управління файловою системою в WordPress. Основна мета - створити сучасний та зручний дизайн, який сприятиме ефективному використанню плагіна.

Нижче визначено ключові аспекти цього оновлення:

- **Мінімалістичний дизайн:** забезпечення простоти та чистоти дизайну для покращення сприйняття користувачами інтерфейсу. Мінімалістичний дизайн сприятиме зосередженості на основних функціях плагіна.
- **Інтуїтивний інтерфейс:** розробка інтуїтивного інтерфейсу, який дозволить користувачам використовувати основні функції без зайвих зусиль. Чіткі та логічні розділи та елементи управління зроблять роботу з плагіном максимально зручною.
- **Адаптивний дизайн:** врахування можливостей адаптації до різних розмірів екранів. Адаптивний дизайн дозволить зручно використовувати плагін як на комп'ютерах, так і на мобільних пристроях.
- **Кольорова палітра:** вибір кольорової палітри, яка відповідатиме стильові WordPress та буде приємна для очей. Коректно вибрані кольори підкреслять професіоналізм та сприятимуть позитивному враженню від використання плагіна.

Оновлення інтерфейсу плагіна є важливим етапом у забезпеченні зручності та ефективності використання продукту користувачами, який сприятиме отриманню приємних вражень від роботи з плагіном.

3. Інтеграція плагіна з бібліотекою Font Awesome.

Використання візуально привабливих та сучасних іконок з бібліотеки Font Awesome може значно покращити естетику та розуміння елементів інтерфейсу плагіна WordPress. Зазначена бібліотека пропонує широкий спектр стилізованих та адаптивних іконок, які можуть використовуватися для різноманітних функціональних елементів плагіна.

Також, розробка механізму, який визначає формат файлу та автоматично адаптує іконки з бібліотеки Font Awesome відповідно до конкретного типу документу, може створити більш інформативний та зручний інтерфейс для користувачів. Динамічна зміна іконок значно полегшить розпізнавання та візуальне сприйняття різних типів файлів, що зробить плагін більш інтуїтивно зрозумілим для користувачів. Такий підхід додасть не лише сучасності, але й підвищить зручність використання плагіна.

4. Реалізація функціоналу для різного відображення файлів.

Функціонал перемикання між режимами перегляду файлів стане важливим функціоналом плагіна, оскільки він дозволить враховувати індивідуальні потреби та уподобання кожного користувача. Реалізація цієї функціональності дасть можливість вибирати між двома режимами відображення: списком або плиткою.

Можливість перегляду файлів у вигляді списку дозволить користувачам швидко переглядати та організовувати велику кількість файлів. Цей режим особливо зручний, коли потрібно оперативно знаходити певний файл серед багатьох інших. З іншого боку, відображення плиткою буде надавати більш естетичний та візуально привабливий вигляд файлів. Окрім цього, інформація про файл буде сприйматись краще в такому форматі відображення.

Ця можливість перемикання між режимами відображення спростить навігацію та дозволить кожному користувачеві вибрати той режим, який відповідає його індивідуальним вподобанням та завданням. Такий підхід допоможе плагіну забезпечити кращий користувацький досвід, зробивши його більш зручним у використанні.

5. Розширення функціоналу виведення файлів на веб-сайт.

В контексті подальшого вдосконалення плагіна для управління файловою системою в WordPress, увагу слід спрямувати і на розширення функціоналу виведення файлів на веб-сайт. Розглянемо ключові аспекти, які потрібно враховувати при покращенні цієї функціональності:

- Параметризація `shortcode`: можливість параметризації `shortcode` для точного визначення, які файли слід виводити на сайті. Слід забезпечити можливість користувачів визначати та керувати тим, яку інформацію виводити на їх веб-сайті. Розуміючи, що потреби користувачів можуть значно відрізнятися, важливо створити гнучкий та інтуїтивний механізм параметризації.
- Відображення файлів на веб-сайті: забезпечення сумісності з веб-сайтом для відображення файлів на сторінці.

Це розширення функціоналу дозволить плагіну не лише динамічно взаємодіяти з веб-сайтом, але і підвищити його ефективність та зручність у використанні.

3.2 Розробка дизайну плагіна

Процес розробки дизайну для вдосконаленого WordPress плагіна для управління файловою системою є важливим етапом, що включає в себе креативний підхід до створення вигляду та взаємодії користувача з плагіном.

Дизайн відіграє критичну роль у створенні успішного продукту. Важливість дизайну виявляється на кількох рівнях [10]:

1. Користувацький досвід: чіткий і зручний дизайн інтерфейсу допомагають користувачам швидко зорієнтуватись та використовувати всі функціональні можливості продукту.
2. Інновації та відповідність трендам: використання сучасних дизайнерських технік та елементів дизайну дозволяє плагіну виглядати актуально та високотехнологічно. Це важливо для забезпечення конкурентоспроможності та привертання уваги користувачів.
3. Брендова ідентичність: індивідуальний та запам'ятовуваний дизайн допомагає плагіну вирізнитися серед інших та будувати власний стиль.
4. Покращення функціональностей: ефективний дизайн дозволяє легко інтегрувати нові функціональності та розширювати можливості плагіна. Він забезпечує гнучкість та швидкість реагування на зміни в вимогах користувачів чи ринку.

Усі ці аспекти доводять, що вірно підібраний та грамотно реалізований дизайн є ключовим елементом успіху WordPress плагіна для управління файловою системою.

Під час розробки WordPress плагіна для управління файловою системою, ми звернулася до ряду стратегій та дизайн-рішень для створення ефективного та зручного інтерфейсу, які включають:

1. Використання темних кольорів.

У процесі розробки дизайну плагіна, особливу увагу було приділено вибору кольорової палітри, при цьому віддаючи перевагу темним відтінкам (рис. 3.1). Це стратегічне рішення має ряд вагомих переваг і є важливою складовою дизайну, оскільки відображається в єдності з темною адміністративною панеллю WordPress.

Також темні відтінки сприяють зниженню втоми очей користувача, забезпечуючи комфортне використання плагіна протягом тривалого часу [11]. Це особливо важливо, оскільки користувачі можуть проводити значний час, взаємодіючи з файловою системою.

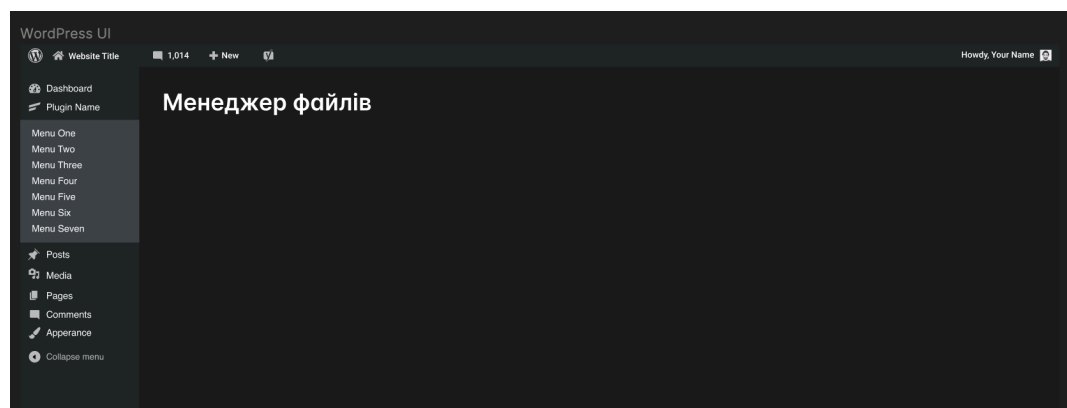


Рисунок 3.1 – Створення дизайну WordPress плагіна.

2. Виділення блоків додавання файлів та фільтрації.

Під час процесу розробки дизайну плагіна, увага була приділена розташуванню ключових елементів інтерфейсу. З урахуванням зручності використання та швидкого доступу до основних функцій, було вирішено виокремити блок для завантаження файлів та блок фільтрації (рис. 3.2).

Блок для додавання файлів був розміщений на видному місці, щоб бути постійно в полі зору користувача. Це забезпечить легкий та швидкий доступ до функції завантаження файлів, що є однією з основних задач плагіна. Всі поля, які використовуються для введення додаткової інформації про файл, були

розташовані таким чином, щоб забезпечити їх зручний доступ та використання. Кожен елемент займає визначене місце, сприяючи чіткості та організованості інтерфейсу. Кнопка для додавання файлу та відповідних даних про нього виокремлена яскравим кольором. Це сприяє простоті та впізнаваності функції, роблячи її легкою для знаходження. Всі іконки, розташовані в даному блоці, були додані за допомогою бібліотеки Font Awesome. Це не лише додає сучасний вигляд елементам, але й забезпечує їхню високу розпізнаваність.

Блок для фільтрації файлів був виділений та розташований пріоритетно, відразу під блоком для додавання файлів. Це дозволяє користувачам швидко використовувати фільтр та здійснювати точний вибір файлів, що відповідають їхнім потребам. Усі поля, що використовуються для вибору параметрів фільтрації, мають аналогічний вигляд до полів, котрі використовуються для введення інформації про файл. Це забезпечує консистентність дизайну. Кнопки для застосування та скидання фільтру також акцентовані яскравими кольорами, створюючи виділені області в інтерфейсі. Це робить функції фільтрації більш помітними для користувача.

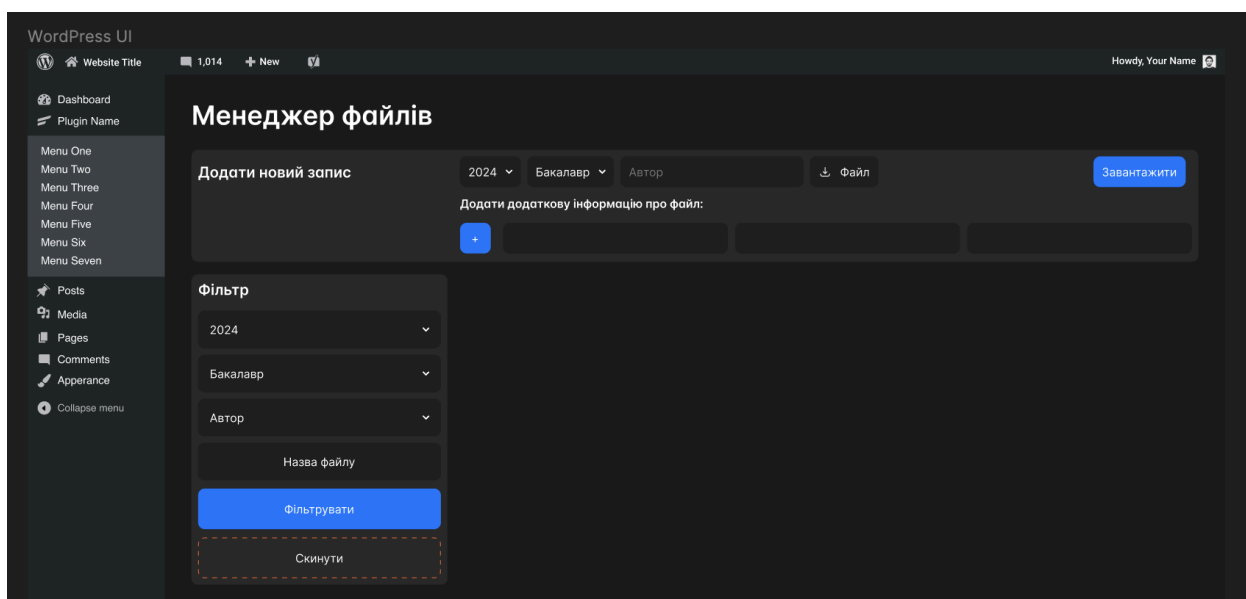


Рисунок 3.2 – Створення дизайну WordPress плагіна.

Розташування елементів інтерфейсу враховує принципи зручності та інтуїтивності. Плагін створений таким чином, щоб користувачі могли легко орієнтуватися та взаємодіяти з основними функціями, забезпечуючи ефективність в роботі з файловою системою WordPress.

3. Логічне розташування файлів.

Для створення оптимального користувацького досвіду враховано важливість візуальної організації файлів поруч із блоком фільтрації. Це рішення спрямоване на забезпечення комфортної роботи користувачів та максимальної ефективності використання плагіна. Всі файли були розташовані поруч із блоком фільтрації, створивши єдиний та компактний інтерфейс (рис. 3.3). Це сприятиме зручному спостереженню за результатами фільтрації та надасть користувачеві чітку структуру для орієнтації серед файлів.

Для поліпшення візуального сприйняття та забезпечення зручності користування плагіном внесено кілька важливих оптимізацій в таблицю файлів:

- Шапка таблиці з файлами виокремлена для кращого візуального розпізнавання. Це надасть користувачам можливість легко визначати, які дані представлені в кожному стовпці таблиці.
- Shortcode виділений для полегшення його копіювання. Це забезпечить користувачам зручний доступ до коду та сприятиме подальшому використанню його за межами плагіна.
- Всі поля таблиці мають аналогічний вигляд та відповідають загальному стилю плагіна. Це надасть єдності дизайну та зробить взаємодію з таблицею однорідною та приємною.

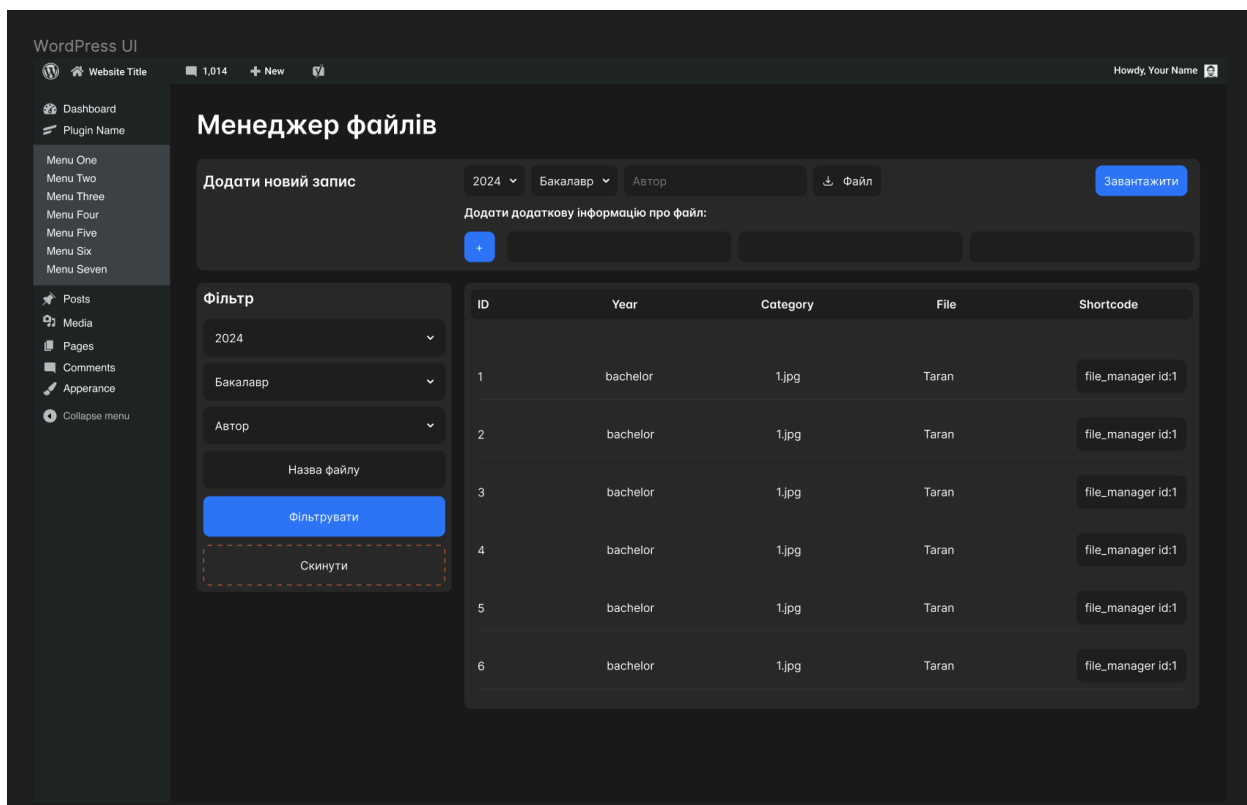
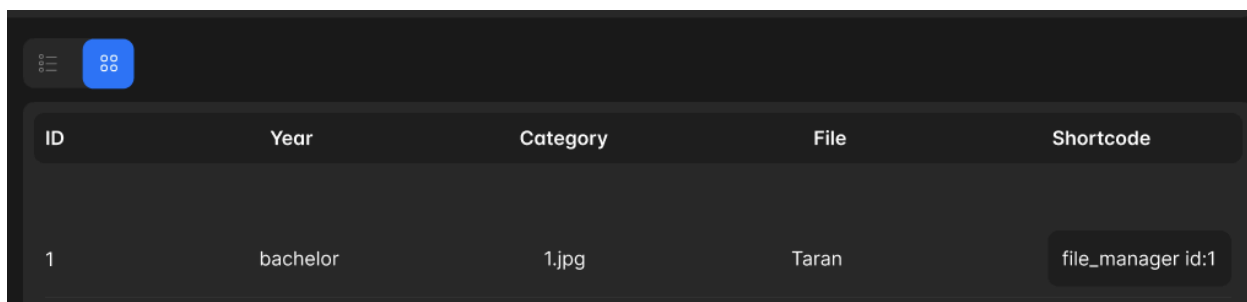


Рисунок 3.3 – Створення дизайну WordPress плагіна.

Візуально організоване розташування файлів сприятиме легкій взаємодії користувача з ними. Кожен елемент розташований таким чином, щоб його можна було легко виявити та використовувати, що зробить роботу з файлами більш ефективною.

4. Кнопка перемикавання відображення.

Однією з ключових функцій, спрямованих на поліпшення зручності користування плагіном, є кнопки перемикавання режиму відображення файлів. Було прийнято рішення розмістити даний перемикач поруч із блоком файлів (рис. 3.4), щоб забезпечити швидкий та зручний доступ користувачів до функції зміни режиму відображення з максимальним збереженням комфорту використання.



ID	Year	Category	File	Shortcode
1	bachelor	1.jpg	Taran	file_manager id:1

Рисунок 3.4 – Створення дизайну WordPress плагіна.

Рішення щодо розташування перемикача відображення має на меті зберегти зручність використання плагіну. Користувачі зможуть безперешкодно переключати режими відображення, не віддаляючись від основної робочої області.

5. Відображення файлів у вигляді плитки.

Додавання режиму відображення файлів у вигляді плитки - це нова можливість, яка спрямована на полегшення взаємодії з контентом та робить процес користування плагіном ще більш гнучким та зручним. Режим відображення у вигляді плитки дозволить переглядати файли в більш організованому та інтуїтивно зрозумілому форматі.

Кожен файл буде представлений окремою карткою, яка надаватиме короткий та зрозумілий огляд інформації (рис. 3.5). Також, для покращення візуального вигляду та легшого розпізнавання формату файлів, будуть використовуватись іконки з бібліотеки Font Awesome. Кожен тип файлу буде асоціюватись з відповідною іконкою, яка надаватиме елементу графічного представлення до назви файлу. Це робиться для того, щоб користувачі могли швидше і легше визначати формат файлу та забезпечити їм більш інтуїтивний інтерфейс для роботи з різноманітними типами документів.

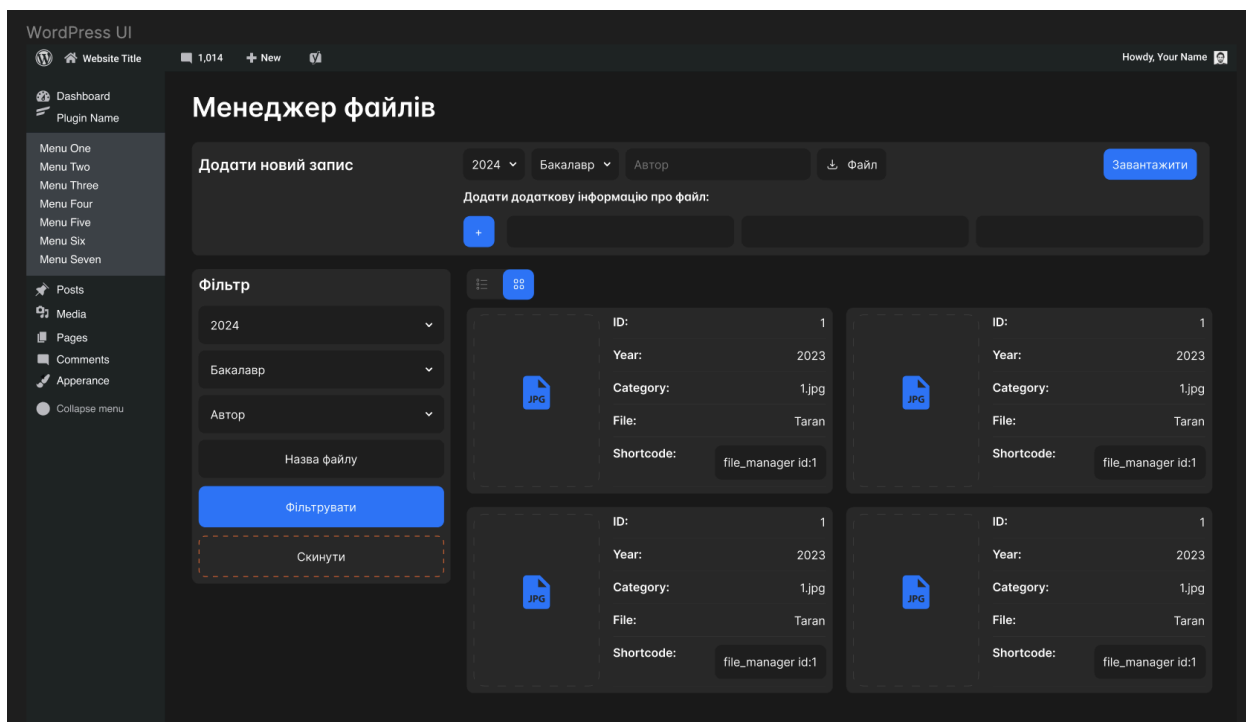


Рисунок 3.5 – Створення дизайну WordPress плагіна.

Дизайн режиму відображення файлів у вигляді плитки буде гармонійно вписуватись в загальний стиль плагіна та підтримуватиме єдність інтерфейсу. Введення даного режиму буде корисним удосконаленням для тих, хто шукає більше інтерактивності та гнучкості у відображенні своїх файлів.

3.3 Висновки до розділу

У даному розділі було сформовано конкретні вимоги до функціоналу плагіна WordPress для управління файловою системою відповідно до результатів дослідження трьох популярних аналогічних рішень, які згадувалися у попередньому розділі. Також у даному розділі було розроблено дизайн плагіна, для забезпечення зручного та інтуїтивно зрозумілого інтерфейсу для користувачів, з урахуванням всіх функціональних вимог.

Загалом, цей розділ демонструє, що розробка плагіна з урахуванням визначених функціональних та дизайнерських аспектів сприятиме створенню ефективного та сучасного інструменту для користувачів WordPress.

4 РОЗРОБКА ТА ТЕСТУВАННЯ ПЛАГІНА

Розробка та тестування плагіна для управління файловою системою у WordPress - це ключові етапи, які визначають високі стандарти функціональності, надійності та зручності використання даного продукту. WordPress, як відома платформа для створення та управління веб-сайтами, надає безліч можливостей для розширення функціоналу через плагіни, тому наше завдання полягає в створенні продукту, який відповідає високим стандартам якості даної CMS.

4.1 Опис процесу програмування

Даний підрозділ відображає основний технічний аспект даного продукту, розкриваючи етапи та рішення, які допоможуть створити функціональний та високоякісний плагін для системи керування вмістом WordPress, який буде повністю задовольнити потреби користувачів та відповідати вимогам, які були сформовані у попередньому розділі.

Для розробки WordPress плагіна ми будемо використовувати наступний стек технологій [12]:

1. PHP. PHP є основною мовою програмування для розробки плагінів у середовищі WordPress. Використовуючи PHP, ми зможемо легко взаємодіяти з функціоналом WordPress, забезпечуючи швидку та ефективну обробку даних та подій. Реалізація логіки обробки даних, робота з базою даних, створення користувацьких функцій будуть реалізовуватись цією мовою програмування.
2. HTML. HTML використовується для створення структури та розмітки веб-сторінок. У випадку розробки плагіна, HTML буде використовуватись

для створення користувацького інтерфейсу, а саме: форм, списків, карток, кнопок та інших елементів інтерфейсу.

3. CSS. CSS використовується для стилізації та оформлення веб-сторінок, щоб зробити їх привабливими для користувача. Забезпечення зовнішнього вигляду елементів інтерфейсу, встановлення кольорів, шрифтів та розміщення елементів буде відбуватись з використанням даної мови стилів.
4. JavaScript. JavaScript використовується для динамічного оновлення веб-сторінок та взаємодії з користувачем без перезавантаження сторінки. Реалізація відмальовки файлів у режимі списку та плитки, переключення між режимами відображення, анімаційні ефекти та взаємодія з користувачем будуть реалізовані завдяки даній мові програмування.
5. MySQL. MySQL є стандартною системою управління базами даних, яка використовується в WordPress для зберігання та управління даними.

4.1.1 Активація плагіна

Щоб продовжити розробку плагіна WordPress для керування файловою системою та впроваджувати нові функції, необхідно активувати плагін. Цей етап відкриває можливості для подальшого розвитку та налаштування плагіна. Після завантаження архіву з кодом плагіна та його активації, ми можемо безпосередньо спостерігати вплив даного програмного продукту на функціоналітет WordPress.

Перейдемо до адміністративної панелі нашого WordPress проекту, де ми можемо спостерігати навігаційну панель, яка містить всі необхідні вкладки (рис. 4.1).

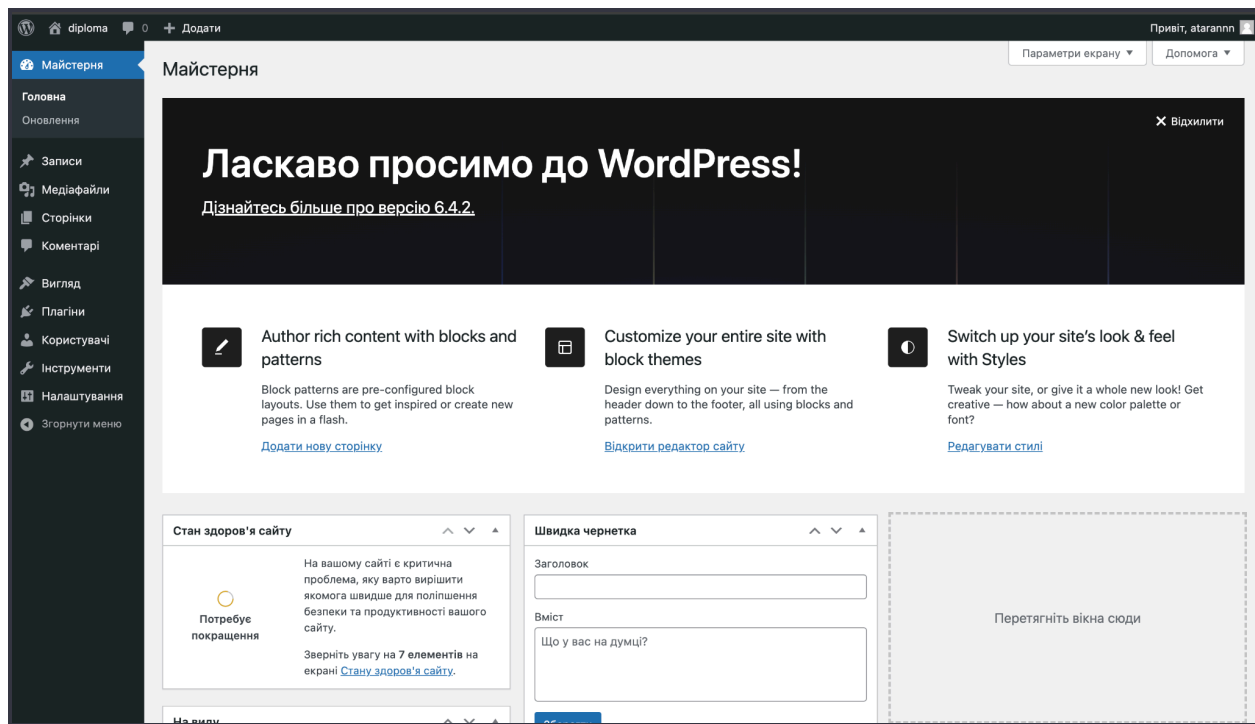


Рисунок 4.1 – Адміністративний інтерфейс CMS WordPress.

Нас цікавить вкладки Плагіни. Після переходу до даної вкладки, ми зможемо переглянути всі активні та неактивні плагіни, котрі має даний WordPress проект. Але наша мета встановити та активувати власний плагін "File Manager Plugin", тому для цього натискаємо кнопку "Додати новий".

Перед нами з'явиться перелік доступних плагінів для WordPress, які є найбільш популярними та лідерами за кількістю завантажень та активацій. Для завантаження та активації власного плагіна "File Manager Plugin" ми маємо натиснути кнопку "Завантажити плагін". Далі слід обрати архів .zip в якому знаходиться код нашого плагіна та завантажити його (рис. 4.2).

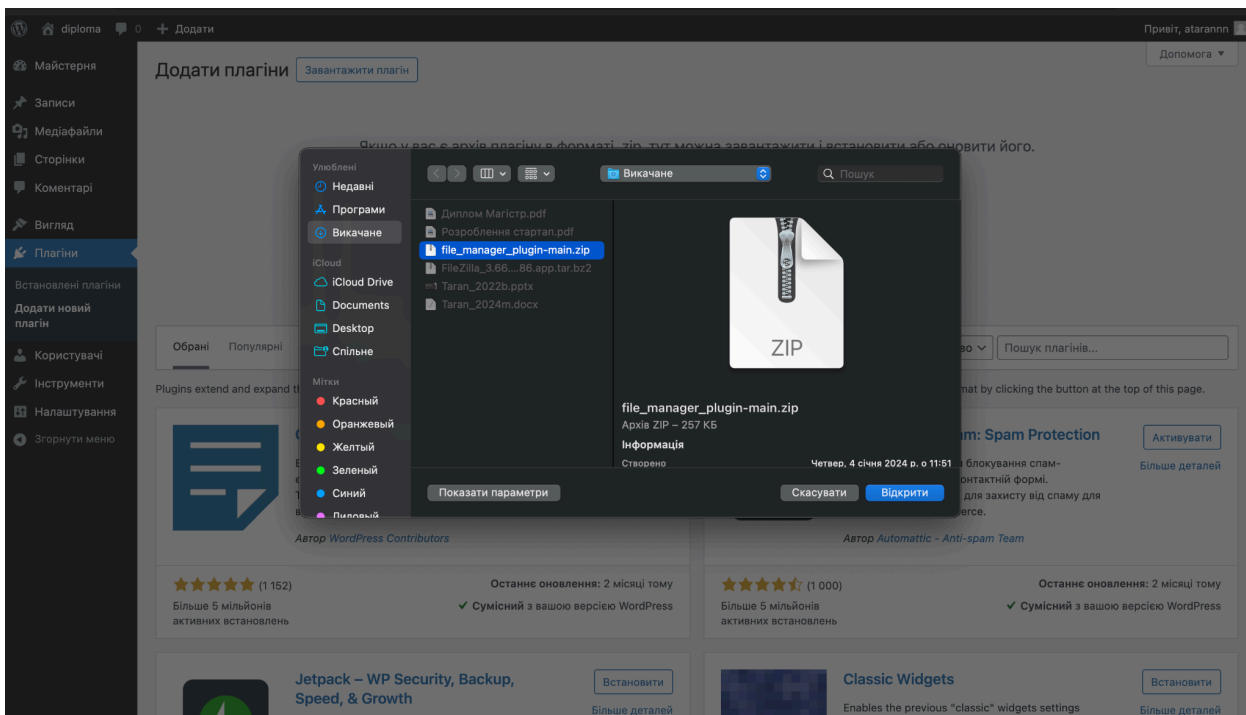


Рисунок 4.2 – Завантаженні плагіна "File Manager Plugin".

Після успішного завантаження плагіна слід його активувати. Для цього слід натиснути кнопку "Увімкнути плагін" (рис. 4.3).

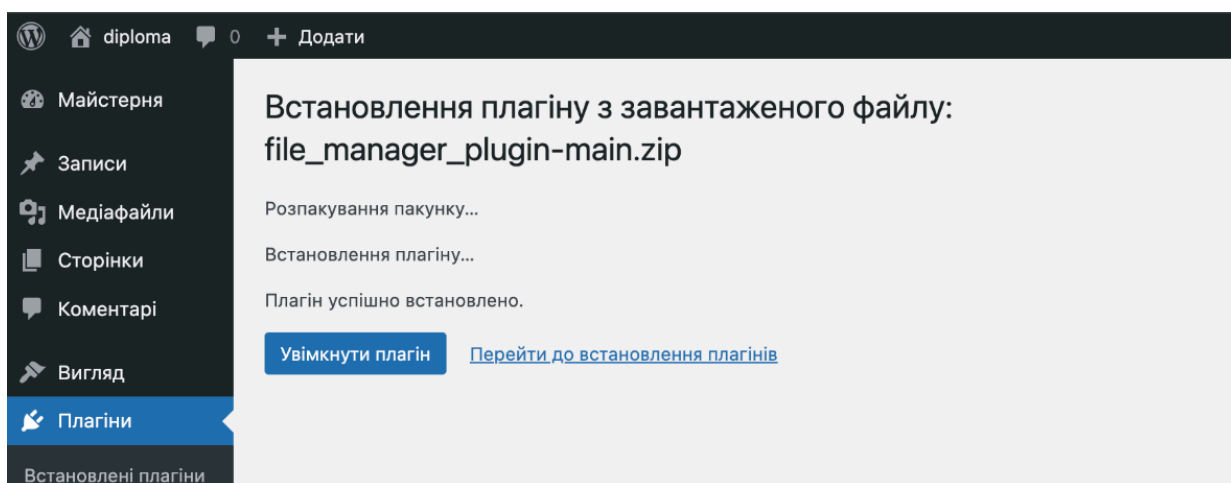


Рисунок 4.3 – Активація плагіна "File Manager Plugin".

Після успішної активації ми спостерігаємо, що "File Manager Plugin" додано до списку активних плагінів (рис. 4.4).

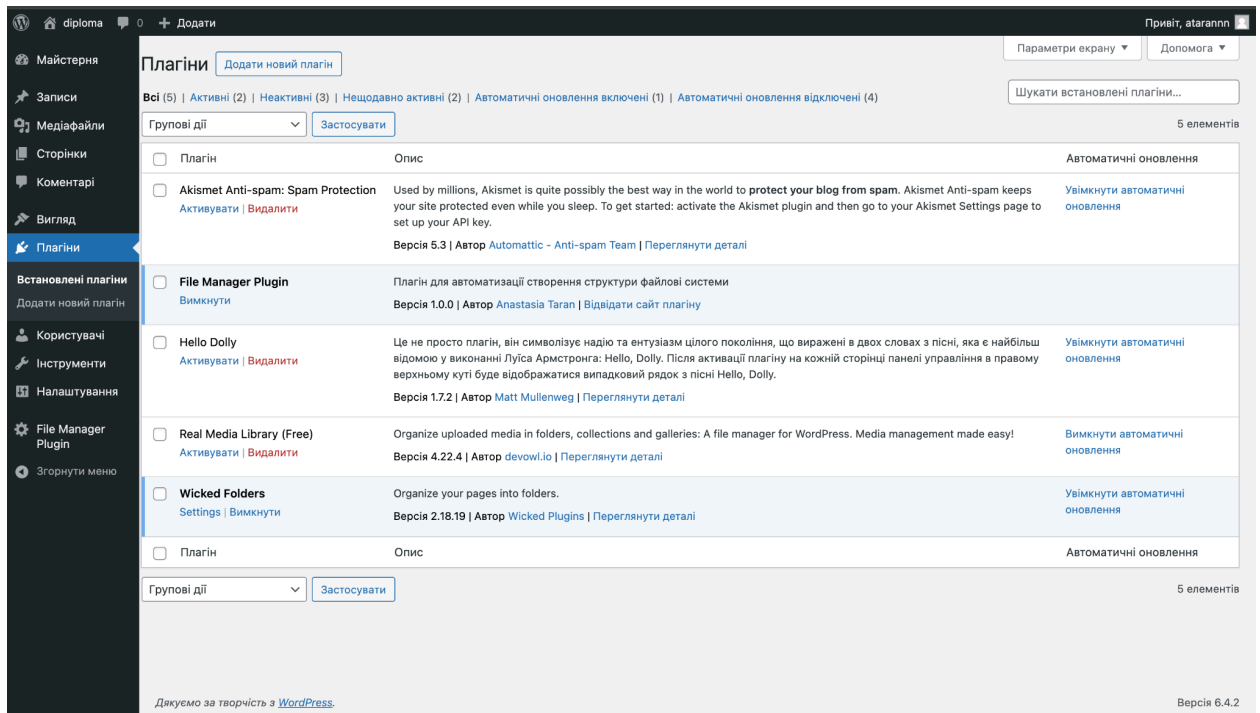


Рисунок 4.4 – Активований плагін "File Manager Plugin".

Тепер можна приступити до розробки плагіна, код якого розташовано за адресою `wp-content/plugins/file_manager_plugin-main` у нашому WordPress проєкті.

4.1.2 Оновлення інтерфейсу плагіна

Розробка плагіна для управління файловою системою в середовищі WordPress є комплексним процесом, і одним із ключових етапів є оновлення інтерфейсу відповідно до розробленого дизайну. Цей етап має на меті покращення користувацького досвіду і забезпечення ефективної навігації користувача в системі. Ми почали саме з цього етапу, оскільки впровадження нових функцій стає більш простим після створення відповідного інтерфейсу.

Початковим етапом процесу оновлення інтерфейсу було створення адаптивної HTML-розмітки, яка відповідає вимогам розробленого дизайну. Цей крок визначає основу нашого інтерфейсу та його структуру, яка повинна бути гнучкою та відмінно пристосованою до різних пристроїв та екранів.

У процесі створення адаптивної HTML-розмітки важливо врахувати основні елементи, які визначають функціональність плагіна. Це включає в себе контейнери для вміщення різноманітних елементів, кнопки для взаємодії з користувачем, форми для зручного введення даних, а також елементи списків для відображення файлів чи іншого контенту.

Далі ми перейшли до створення CSS-стилів для нашого інтерфейсу, щоб надати йому не лише естетичний вигляд, а й забезпечити коректну роботу та зручність взаємодії. Використовуючи сучасні практики стилізації, ми прагнули досягти резинового дизайну, що означає його гнучкість та адаптивність до різних розмірів екранів. Застосовуючи медіа-запити, ми забезпечили адаптацію нашого інтерфейсу до різних пристроїв, починаючи від великих настільних моніторів та закінчуючи малими мобільними екранами. Це дозволило забезпечити зручність користувача та оптимальний вигляд незалежно від того, на якому пристрої відбувається перегляд (рис. 4.5).

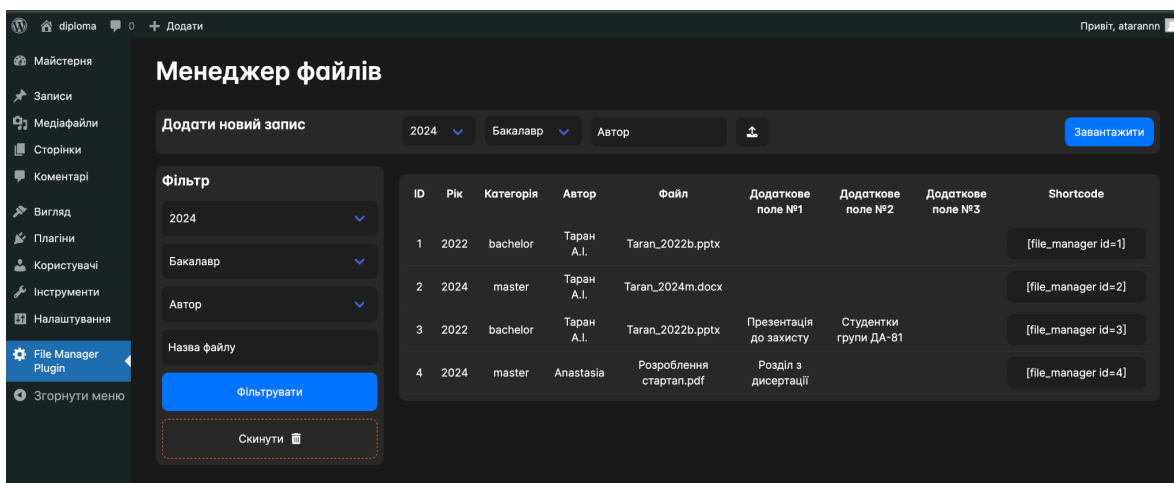


Рисунок 4.5 – Результат оновлення інтерфейсу плагіна.

Ми подбали про ефективне використання каскадних таблиць стилів (CSS) для деталізації різноманітних елементів інтерфейсу, забезпечуючи їхню відповідність до визначеного дизайну та функціональності. Кожен відмінний від стандартного стиль враховує естетичні та практичні аспекти, створюючи

вигляд, який не лише приваблює, але й оптимально пристосовується до потреб користувача на будь-якому пристрої.

4.1.3 Інтеграція плагіна з бібліотекою Font Awesome

Бібліотека Font Awesome – це зручний інструмент для виведення веб-іконок на сторінках різних веб-проектів. Ця бібліотека надає доступ до вражаючого арсеналу стилізованих та адаптивних іконок, які використовуються для поліпшення дизайну та візуальної привабливості різних продуктів. Іконки Font Awesome відзначаються не лише своєю красою та різноманіттям, але і практичністю, адже їх легко інтегрувати та налаштовувати. Використання цієї бібліотеки є ефективним засобом для підвищення візуального враження та забезпечення інтерактивності для користувачів [13].

Для коректної роботи плагіна із використанням іконок Font Awesome, нам необхідно встановити та активувати плагін Font Awesome (рис. 4.6) у своєму WordPress проекті, інакше іконки можуть відображатись некоректно або не відображатись взагалі.

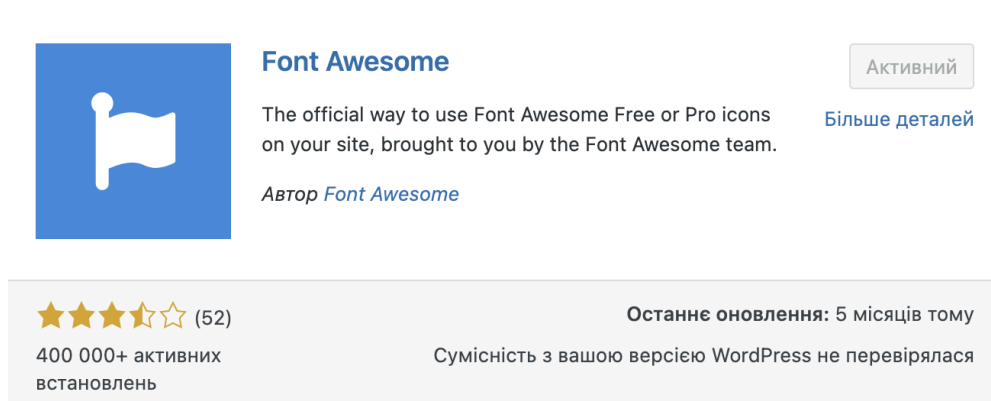


Рисунок 4.6 – Активація плагіна Font Awesome.

Далі нам слід додати код для підключення Font Awesome до нашого плагіна (рис. 4.7).

```

no usages
function file_manager_plugin(){
    add_menu_page( 'File Manager Plugin', 'File Manager Plugin', 'manage_options', 'file_manager_plugin', 'file_manager_init' );

    wp_enqueue_style('custom-styles', plugins_url('/style.css', __FILE__));
    wp_enqueue_style('font-awesome', 'https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css', array(), '5.15.3', 'all');

    wp_enqueue_script('file-manager', 'https://code.jquery.com/jquery-3.6.0.min.js');
    wp_localize_script('file-manager', 'myAjax', array( 'ajaxurl' => admin_url( 'admin-ajax.php' ) ));
    wp_enqueue_script('file-manager-script', plugins_url('/script.js', __FILE__));
}

```

Рисунок 4.7 – Підключення Font Awesome до нашого плагіна.

Після даних операцій ми можемо інтегрувати іконки Font Awesome в наш плагін (рис. 4.8 - 4.9).

```

<button id="reset-btn" onclick="btnReset(event)">
    Скинути
    <i class="fa-solid fa-trash-can" style="..."></i>
</button>

```

Рисунок 4.8 – Приклад використання іконок Font Awesome.

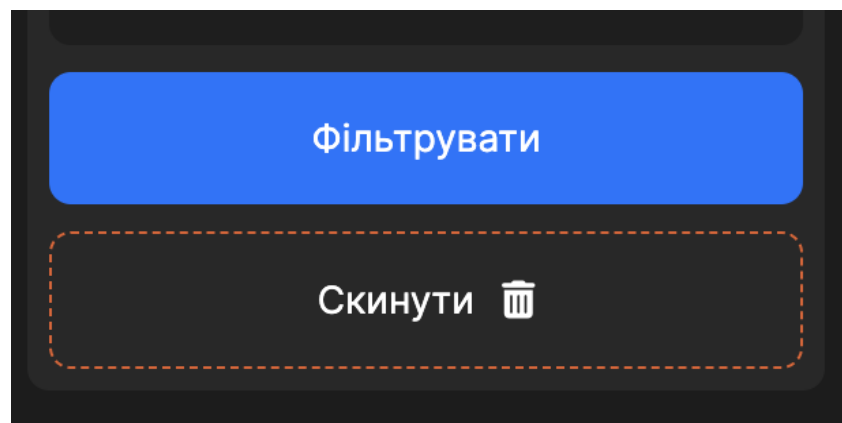


Рисунок 4.9 – Результат використання іконок Font Awesome.

4.1.4 Реалізація динамічних полів для додавання додаткової інформації про файли

Однією з ключових та інноваційних особливостей нашого плагіна є можливість динамічно додавати додаткові поля під час заповнення інформації про файл.

Можливість динамічного додавання полів створює надзвичайну гнучкість для користувачів у налаштуванні обсягу додаткової інформації, яку вони

бажають додати. Користувачі можуть самостійно визначати об'єм даних, які їм необхідно вказати, відповідно до їхніх конкретних потреб та вимог.

Процес реалізації динамічного списку полів даних про файл починається з підготовки розмітки та визначення необхідних елементів у HTML (рис. 4.10).

```
<div class="container new-fields">
  <p class="title">Додати додаткову інформацію про файл:</p>
  <div class="row-new-fields">
    <button class="btn btn-primary add_form_field"><i class="fa-solid fa-plus" style="..."></i></button>
    <div class="row-new-fields_inner">
      </div>
    </div>
  </div>
</div>
```

Рисунок 4.10 – Реалізація динамічного списку полів даних про файл.

JavaScript та JQuery код відповідає за динамічне додавання полів та їх видалення. Користувач матиме змогу як додавати додаткові поля, так і видаляти їх (рис. 4.11 - 4.12).

При кліці на кнопку "+", перевіряється, чи не перевищено максимальну кількість полів. Якщо ні, то додається нове поле. Користувач також може видалити будь-яке поле, натиснувши на відповідну кнопку.

Також ми зазначили можливість користувача змінювати максимальну кількість полів. Кожен користувач зможе швидко корегувати кількість додаткових полів, змінивши значення поля `max_fields`.

Рисунок 4.11 – Реалізація динамічного додавання полів.

ID	Рік	Категорія	Автор	Файл	Додаткове поле №1	Додаткове поле №2	Додаткове поле №3	Shortcode
1	2022	bachelor	Таран А.І.	Taran_2022b.pptx				[file_manager id=1]
2	2024	master	Таран А.І.	Taran_2024m.docx				[file_manager id=2]
3	2022	bachelor	Таран А.І.	Taran_2022b.pptx	Презентація до захисту	Студентки групи ДА-81		[file_manager id=3]

Рисунок 4.12 – Результат додавання файлу з додатковими полями.

4.1.5 Реалізація функціоналу для різного відображення файлів

Розробка перемикача для вибору різних режимів відображень файлів (список чи плитка), в інтерфейсі нашого плагіна WordPress для керування файловою системою, включає декілька ключових етапів.

Початково, ми впроваджуємо перемикач в верстці плагіна. Це означає додавання відповідних HTML-елементів, які будуть представляти сам перемикач у веб-інтерфейсі(рис. 4.13) .

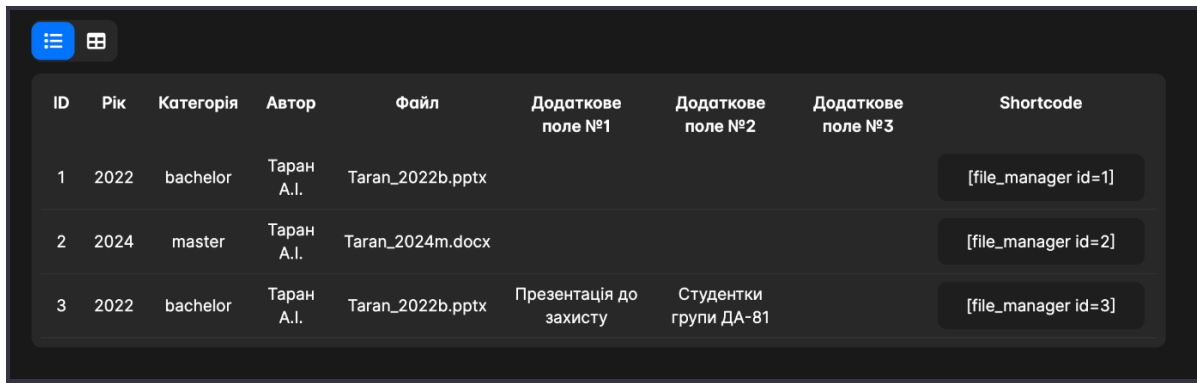
```

<label class="switch">
  <input type="checkbox" id="viewSwitch">
  <span class="slider round">
    <i class="fa-solid fa-list" style="..."></i>
    <i class="fa-solid fa-table" style="..."></i>
  </span>
</label>

```

Рисунок 4.13 – Верстка перемикача відображення файлів.

Також важливо враховувати дизайн та стилізацію даного елемента, щоб він гармонійно виглядав в інтерфейсі плагіна та надавав зрозумілість його функціоналу (рис. 4.14).



ID	Рік	Категорія	Автор	Файл	Додаткове поле №1	Додаткове поле №2	Додаткове поле №3	Shortcode
1	2022	bachelor	Таран А.І.	Taran_2022b.pptx				[file_manager id=1]
2	2024	master	Таран А.І.	Taran_2024m.docx				[file_manager id=2]
3	2022	bachelor	Таран А.І.	Taran_2022b.pptx	Презентація до захисту	Студентки групи ДА-81		[file_manager id=3]

Рисунок 4.14 – Стилізація перемикача відображення файлів.

Функціонал, який при кліці буде викликати перемикання виду відображення, реалізований за допомогою JavaScript та бібліотеки jQuery, в той час як самі режими відображення реалізовані завдяки мові програмування PHP.

За допомогою JavaScript з використанням бібліотеки jQuery ми реалізуємо функціонал перемикача, що включає обробку подій для реагування на вибір користувача та виклик відповідних дій. Тобто, JavaScript використовуються для створення інтерактивності та динамічної зміни елементів веб-сторінки в реальному часі.

З іншого боку, PHP відповідає за генерацію та обробку розмітки на серверному боці, забезпечуючи необхідні дані та параметри для відображення відповідних режимів.

Такий розподіл функціоналу між PHP та JavaScript дозволяє оптимально використовувати можливості обох мов програмування для створення багатофункціонального та ефективного інтерфейсу плагіна WordPress (рис. 4.15).

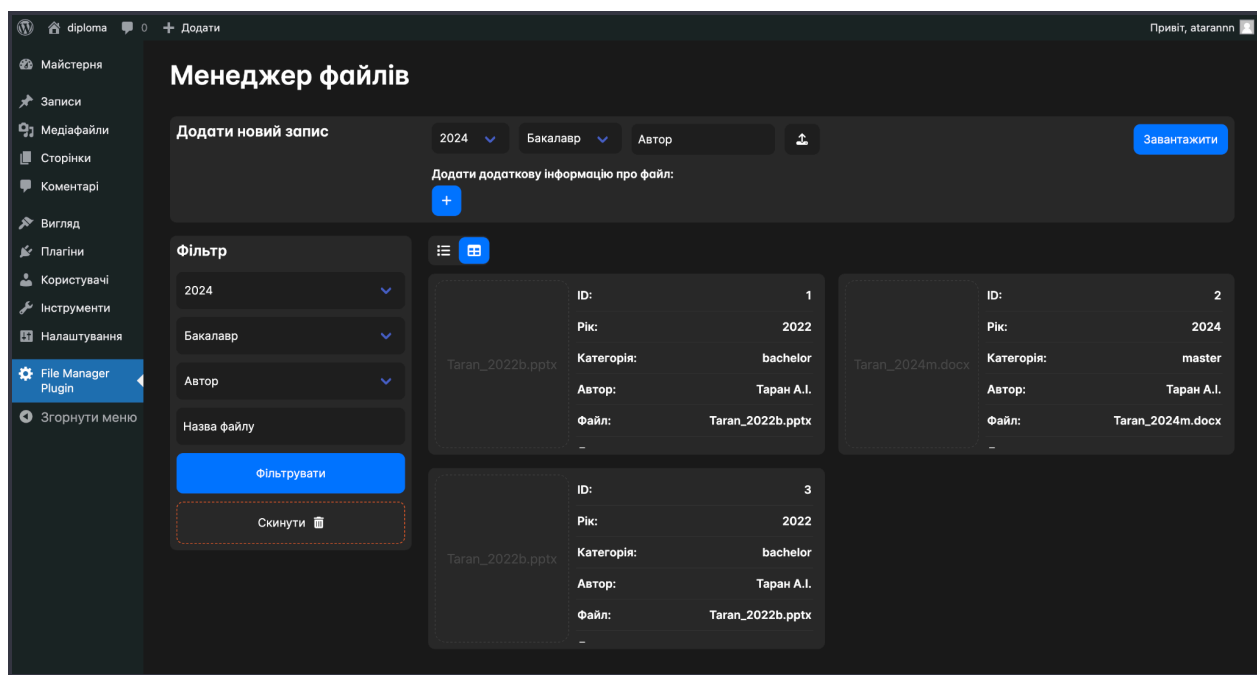


Рисунок 4.15 – Реалізація відображення файлів у вигляді плитки.

Розглядаючи важливість механізму динамічної зміни іконок у залежності від типу файлів нам слід було визначити різні типів файлів і призначити їм відповідні іконки. Розглянемо, як це виглядає в контексті нашого плагіна:

1. Відокремимо різні типи файлів, які підтримує наш плагін: .docx, .pdf, .txt, .pptx, .zip.
2. Для кожного типу файлів визначимо відповідну іконку з бібліотеки Font Awesome (рис. 4.16).

```

case 'docx':
    $iconClass = 'fa-file-word';
    break;
case 'pdf':
    $iconClass = 'fa-file-pdf';
    break;
case 'txt':
    $iconClass = 'fa-file-lines';
    break;
case 'pptx':
    $iconClass = 'fa-file-powerpoint';
    break;
case 'zip':
    $iconClass = 'fa-file-zipper';
    break;
default:
    $iconClass = 'fa-file';
    break;

```

Рисунок 4.16 – Призначення іконок для різних форматів файлів.

Використовуючи PHP, ми реалізували функціонал, який динамічно змінює іконки відповідно до типу файлів (рис. 4.17).

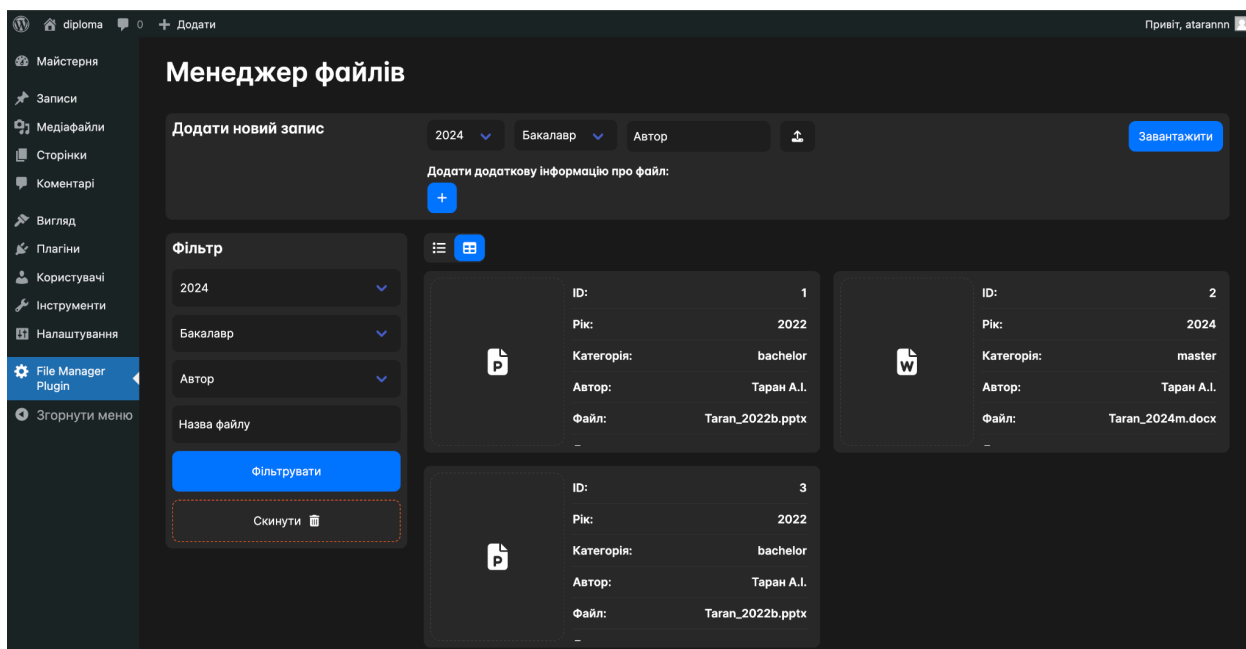


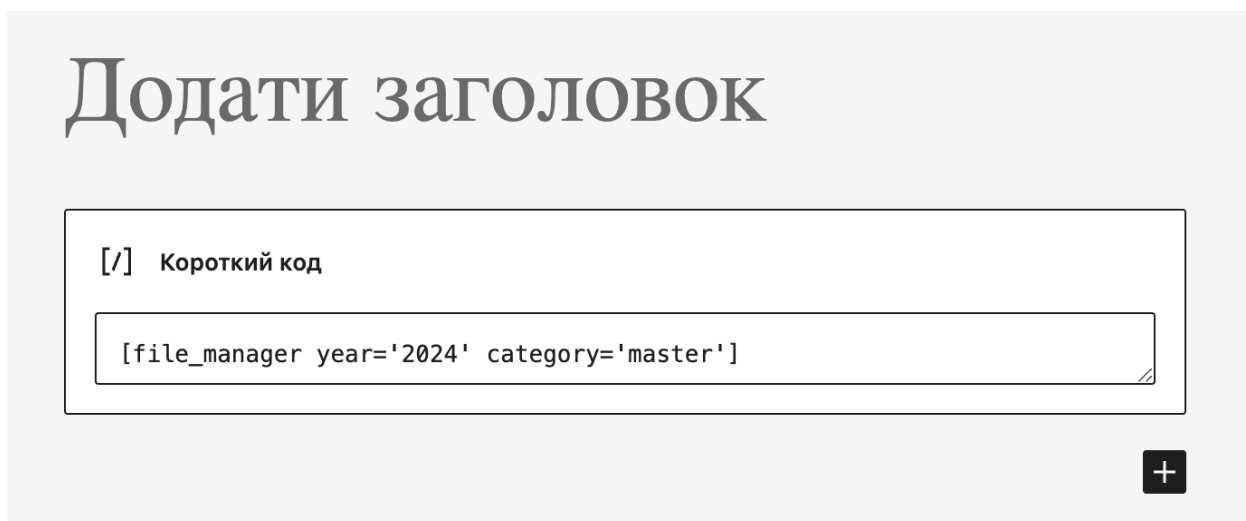
Рисунок 4.17 – Реалізація динамічної зміни іконок відповідно до форматів файлів.

Такий підхід не лише полегшує розпізнавання типів файлів, але і створює більш збалансований та зручний інтерфейс для користувачів, підвищуючи їхню взаємодію з плагіном.

4.1.6 Розширення функціоналу виведення файлів на веб-сайт

Функціонал виведення файлів на веб-сайт є критично важливим для користувачів, які взаємодіють із файловою системою та бажають легко та швидко виводити необхідні файли на їх веб-сайт. У зв'язку з цим, розширення можливостей наявного функціоналу для виведення файлів за допомогою shortcode стає важливим елементом оптимізації роботи з файлами.

Однією з основних можливостей, яку було реалізовано - це розширення параметрів, які користувач може використовувати при виведенні файлів на свій веб-сайт. Користувачам будуть доступні параметри, такі як year, autor, та category. Завдяки цьому, користувачі зможуть легко вибирати файли за цими параметрами та виводити їх на сторінки веб-сайту (рис. 4.18 - 4.19).



Додати заголовок

[/] Короткий код

```
[file_manager year='2024' category='master']
```

+

Рисунок 4.18 – Приклад shortcode для виведення файлів на веб-сайт.

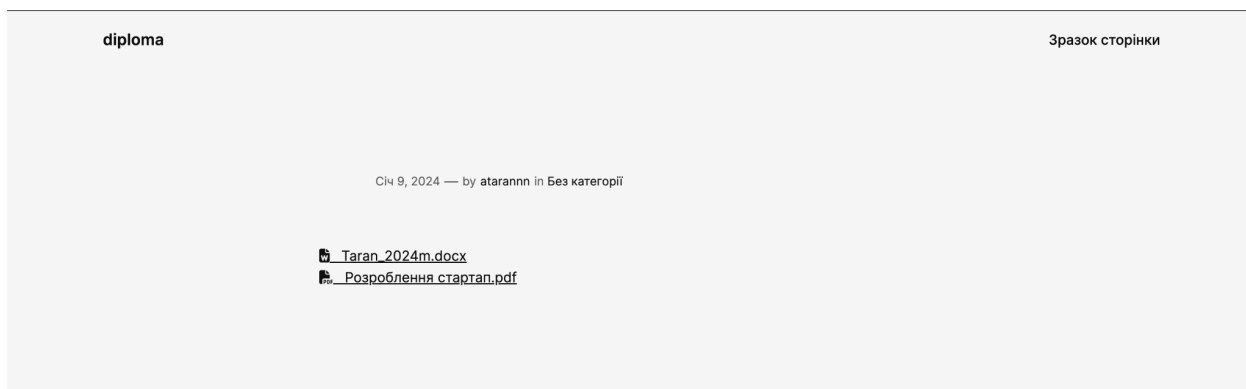


Рисунок 4.19 – Результат виведення shortcode на веб-сайт.

4.2 Проведення тестування

В даному розділі надається детальний огляд процесу функціонального тестування плагіна WordPress для управління файловою системою, який гарантує високу якість нашого продукту.

Підхід до функціонального тестування базується на глибокому розумінні потреб користувачів та активному використанні вищих стандартів ефективної роботи на платформі WordPress. Основною метою - є надати продукт, який не лише відповідає очікуванням, але й перевершує їх, надаючи надзвичайний досвід використання для кожного користувача.

Розпочнемо тестування плагіна, заповнивши обов'язкові поля для завантаження файлів. Для цього перейдемо на сторінку плагіна та оберемо рік створення файлу. Потім виберемо категорію, до якої відноситься завантажуваний файл, вкажемо ім'я автора та завантажимо сам файл. Цей процес дозволить нам перевірити правильність роботи ключових функцій плагіна та його взаємодію з введеними даними.

Процес тестування включає в себе аналіз здатності до перемикання параметрів у полях "Рік" (рис. 4.20) та "Категорія"(рис. 4.21).

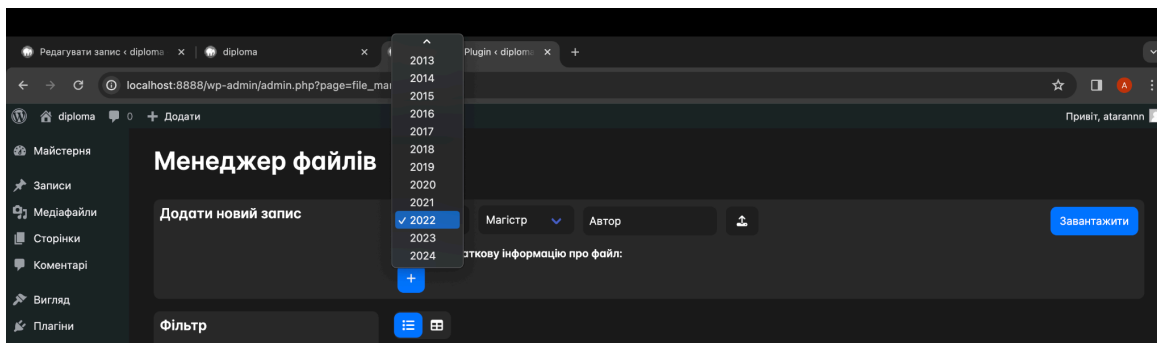


Рисунок 4.20 – Вибір року.

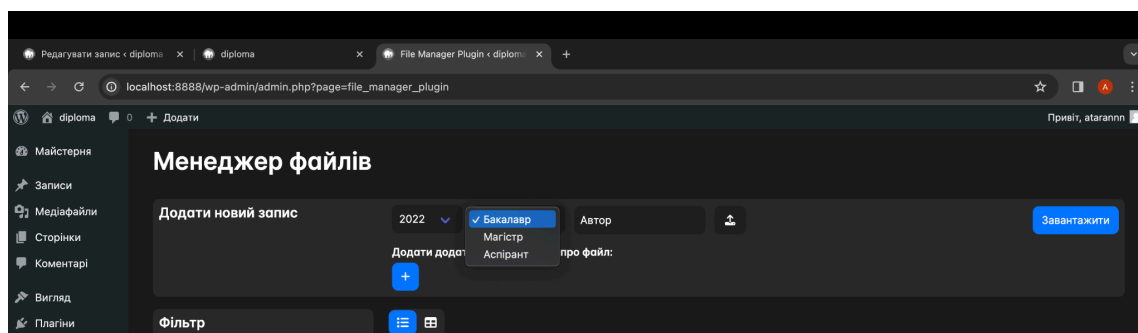


Рисунок 4.21 – Вибір категорії.

Наступним етапом нашого тестування буде уважне оцінювання функціоналу поля для введення інформації про автора завантажуваного файлу (рис. 4.22).

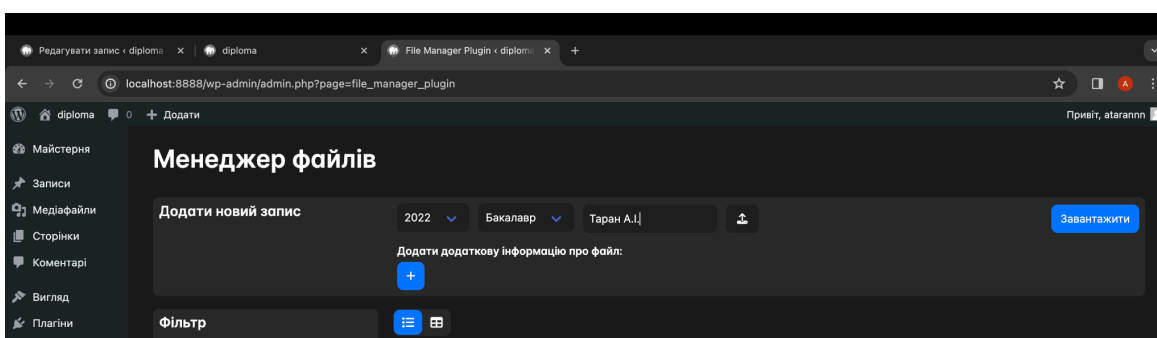


Рисунок 4.22 – Заповнення поля про автора.

Для тестування функціоналу додавання файлу було обрано презентацію до бакалаврської дипломної роботи в форматі .pptx (рис. 4.23). Цей формат є досить поширеним для презентацій, і важливо впевнитися, що плагін працює оптимально з файлами цього типу.

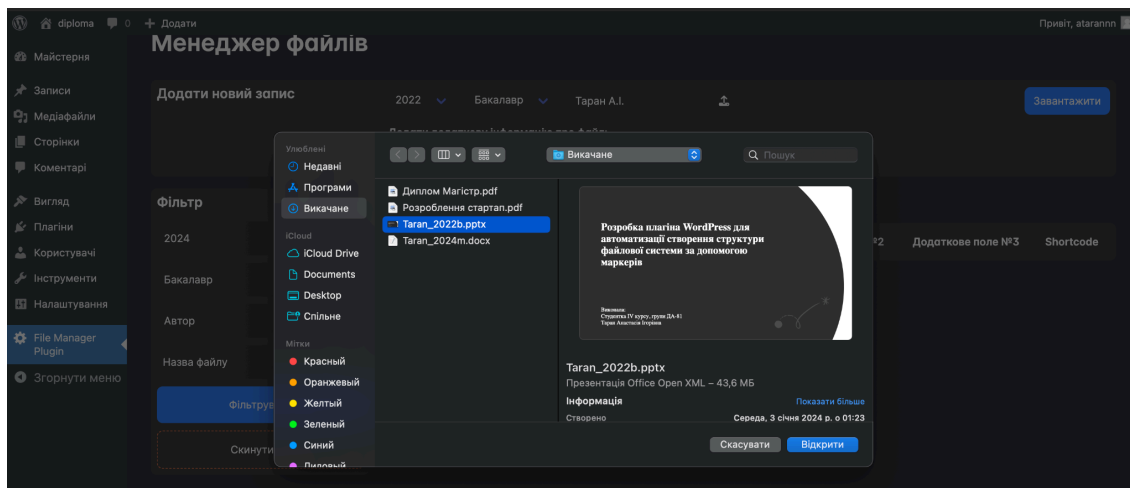


Рисунок 4.23 – Процес завантаження файлу.

В ході проведення тестування було використано два динамічних поля, призначених для додаткової інформації (рис. 4.24). Це важливий етап, оскільки користувачі можуть бажати додати конкретні деталі чи коментарі до своїх файлів.

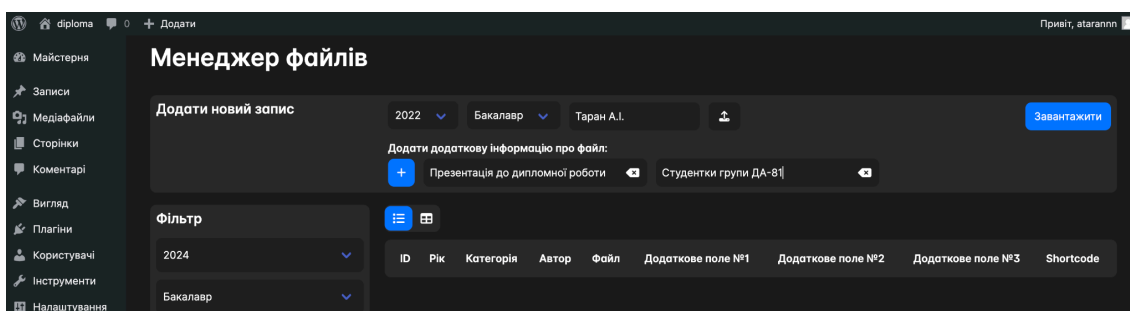


Рисунок 4.24 – Результат заповнення полів для завантаження файлу.

Після цього, натискаємо кнопку “Завантажити” і в результаті бачимо, що інформація про файл додається до списку (рис. 4.25). Можна зробити висновок, що тестування функціоналу завантаження файлів пройшло успішно:

- Тестування вибору параметрів, таких як рік та категорія, та заповнення поля з інформацією про автора пройшло без проблем, так як користувач має можливість легко визначити необхідні параметри для файлу і вводити необхідну інформацію про автора.

- Файл презентації .pptx було успішно завантажено, підтверджуючи гнучкість та зручність роботи плагіна для користувачів, які працюють із різноманітними форматами файлів.
- Додаткові поля ефективно інтегруються в роботу плагіна, а їх заповнення відбувається без зайвих труднощів.

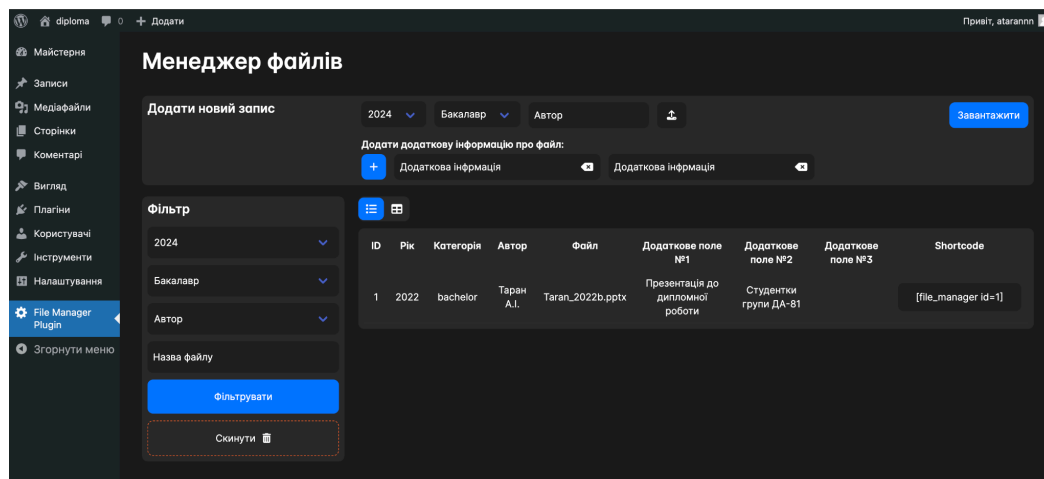


Рисунок 4.25 – Результат завантаження файлу.

В доповнення до тестування плагіна додамо декілька файлів із різноманітною інформацією (рис. 4.26). Це дозволить створити реалістичне середовище для тестування плагіна, оскільки воно повинно ефективно опрацьовувати різноманітні дані.

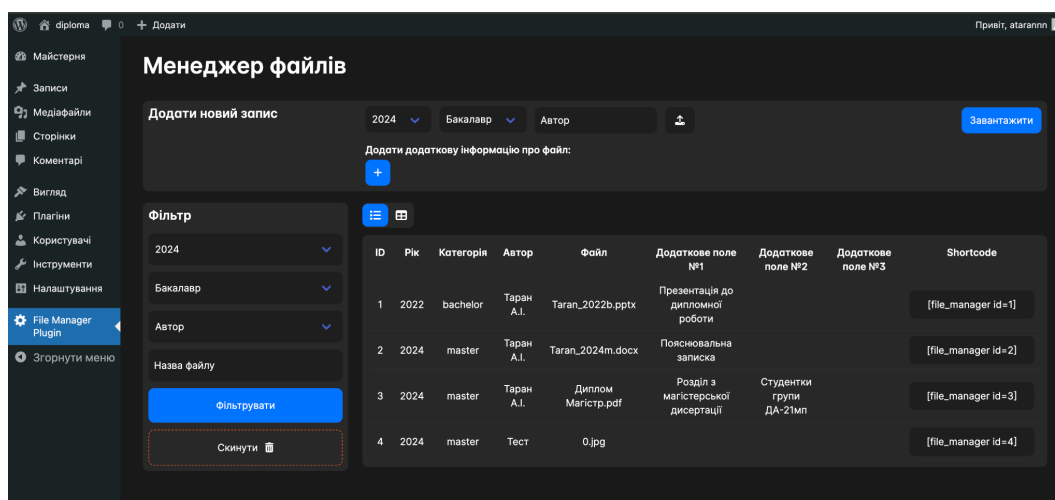


Рисунок 4.26 – Результат завантаження файлів.

Наступним етапом нашого тестування буде оцінка ефективності фільтра, розташованого в лівій частині інтерфейсу. Для цього необхідно вибрати рік та категорію, які відповідають завантаженим файлам, які ми плануємо відобразити в таблиці. Крім того, фільтр можна додатково звужити шляхом обрання конкретного автора та введення назви файлу, який ми шукаємо. Цей етап дозволить нам переконатися, що фільтрація працює належним чином і надає зручний інструмент для швидкого пошуку конкретних файлів серед великої кількості документів.

Планується провести тестування з різними комбінаціями параметрів (рис. 4.27 - 4.29), щоб впевнитися, що фільтрація надійно працює у всіх можливих сценаріях використання.

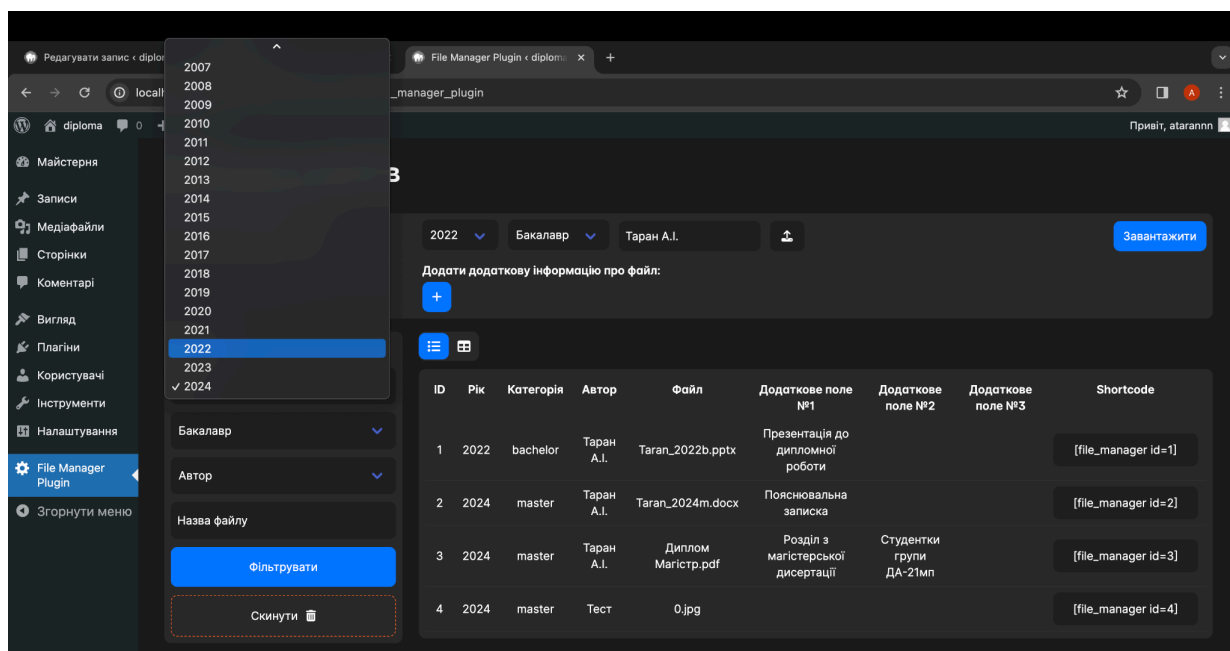


Рисунок 4.27 – Вибір року для фільтрації файлів.

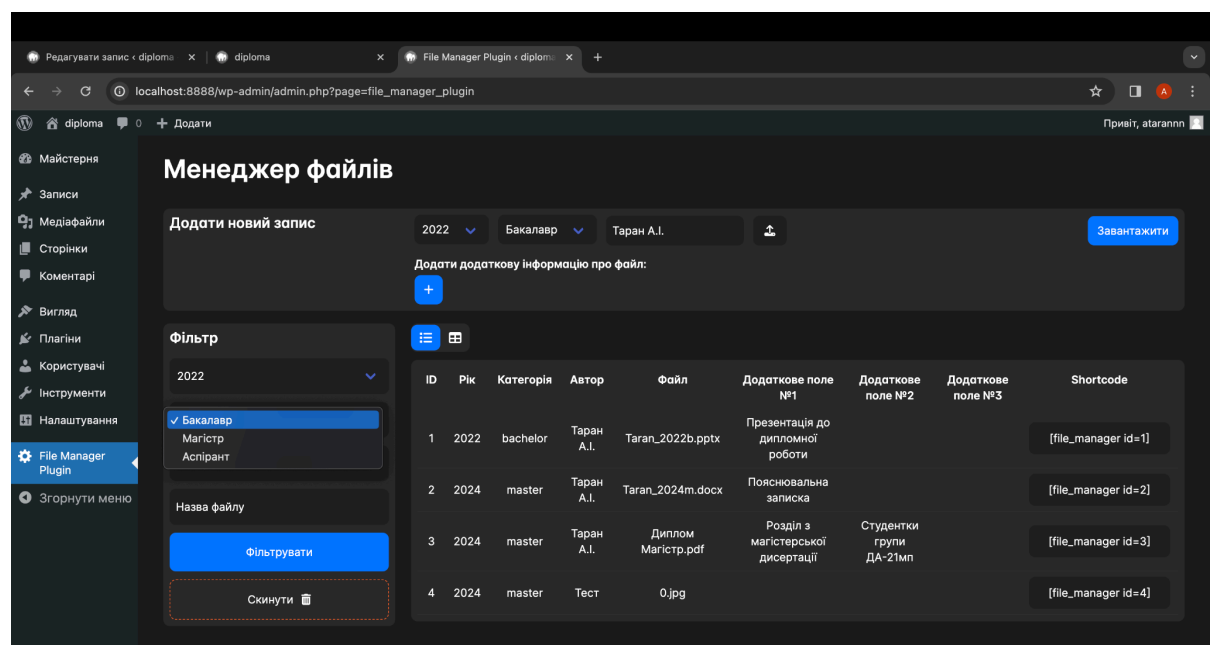


Рисунок 4.28 – Вибір категорії для фільтрації файлів.

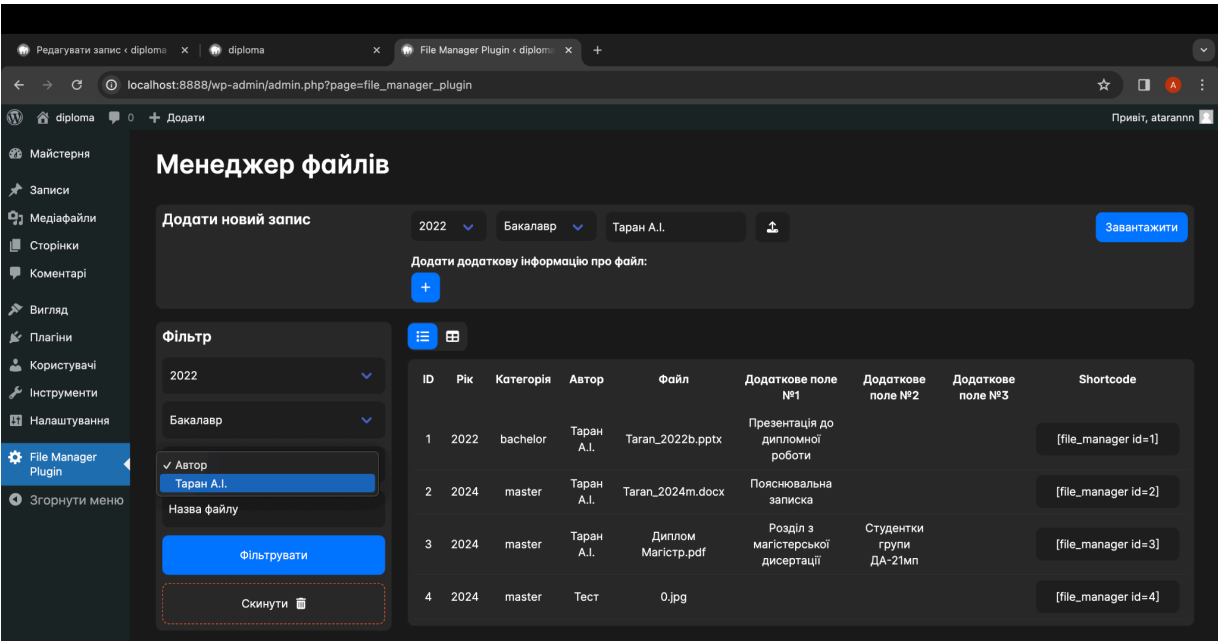


Рисунок 4.28 – Вибір категорії для фільтрації файлів.

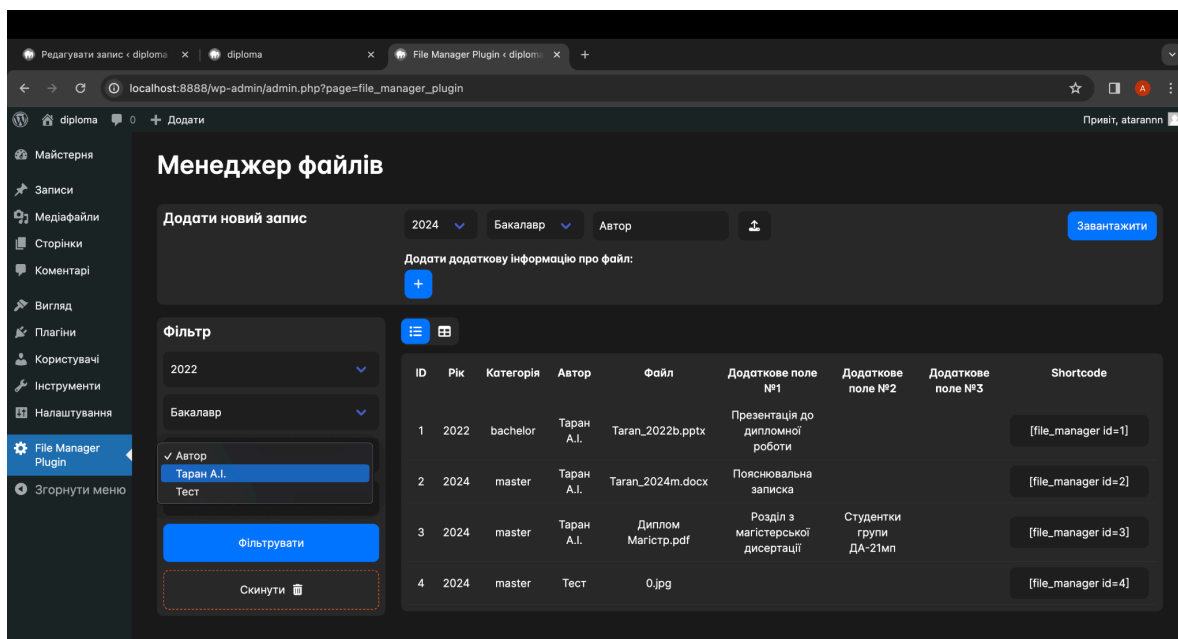


Рисунок 4.29 – Вибір автора для фільтрації файлів.

Щоб здійснити фільтрацію та відобразити в списку лише ті файли, які відповідають обраним параметрам, необхідно натиснути кнопку "Фільтрувати".

На основі застосованого фільтру тепер в списку представлені лише ті файли, які відповідають обраним параметрам, а саме: рік - 2022 рік, категорія - бакалавр, автор - Таран А.І. (рис. 4.30).

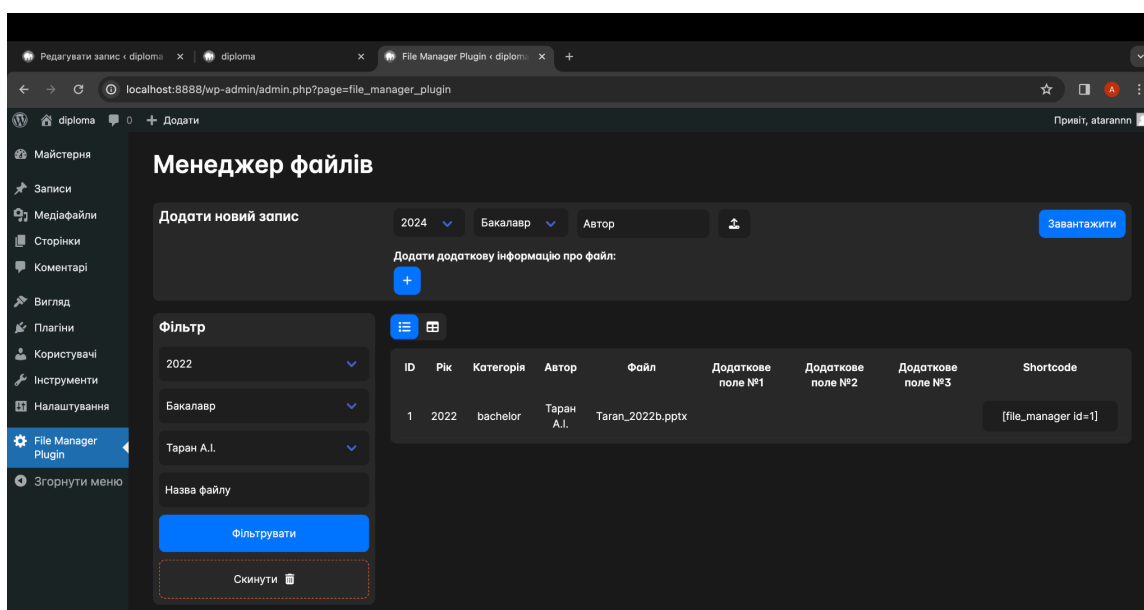


Рисунок 4.30 – Результат роботи фільтру.

Також, є можливість відновити таблицю до початкового вигляду, відобразивши повний список всіх файлів. Для цього достатньо натиснути на кнопку “Скинути” (рис. 4.31).

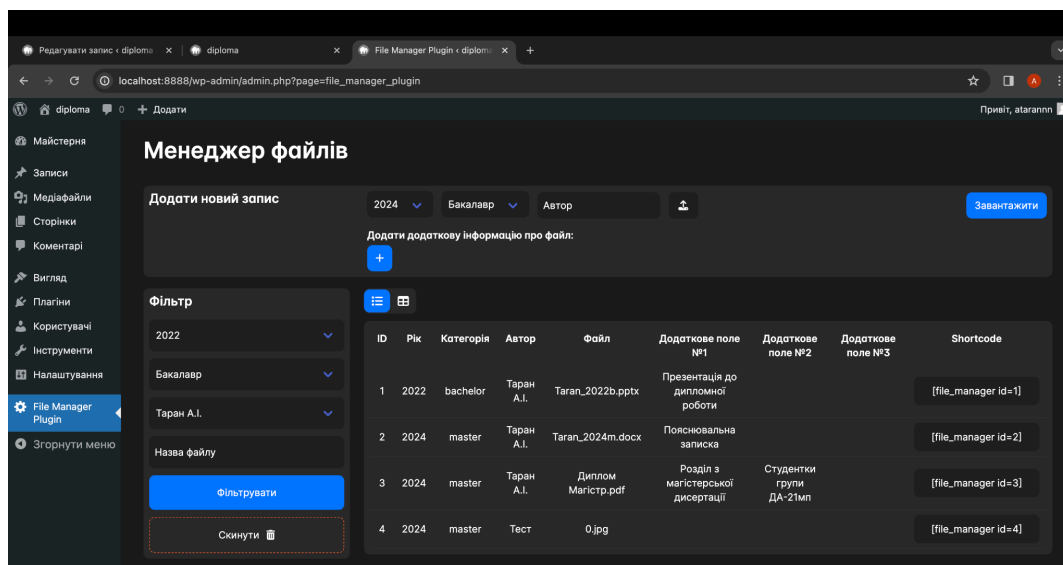


Рисунок 4.31 – Результат скидання фільтру та початковий вид списку.

Далі переходимо до тестування функціоналу перемикача відображення файлів. Для цього необхідно натиснути на перемикач, розташований над списком файлів.

Під час проведення тестування функціоналу перемикача режиму відображення файлів, ми взяли до уваги деталі його роботи. Під час кожного перемикання відображення відповідало обраному режиму (рис. 4.32), що дало нам впевнитися в його правильному функціонуванні.

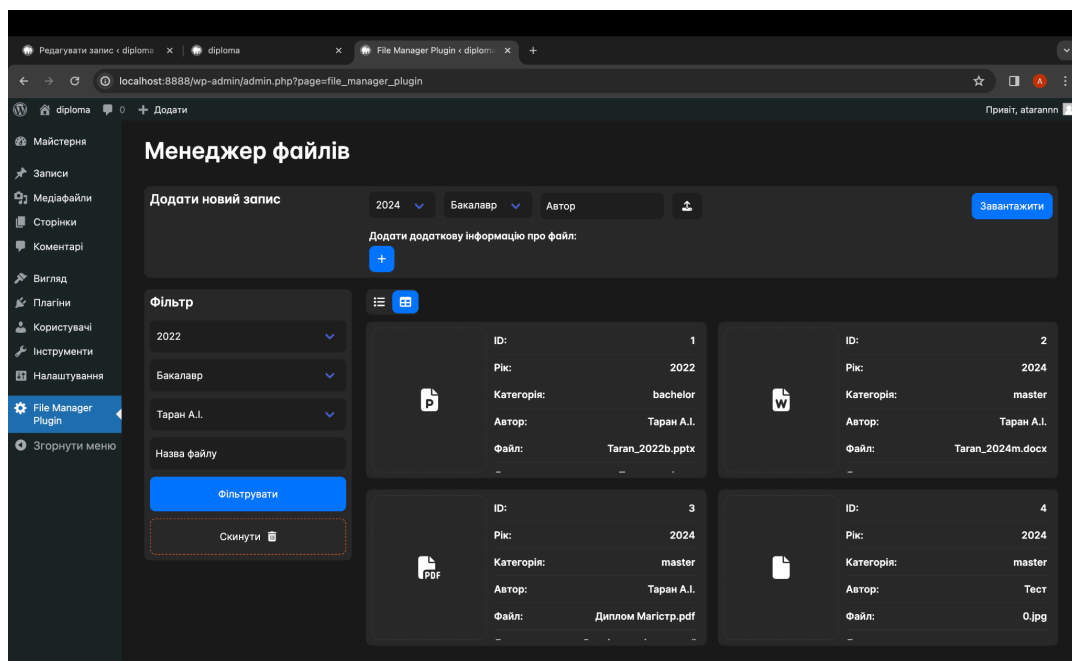


Рисунок 4.32 – Відображення файлів плиткою.

Також ми провели фільтрацію файлів з використанням попередніх параметрів, таких як рік (2022), категорія (бакалавр) та автор (Таран А.І.) (рис. 4.33), щоб впевнитись, що режим відображення зберігається.

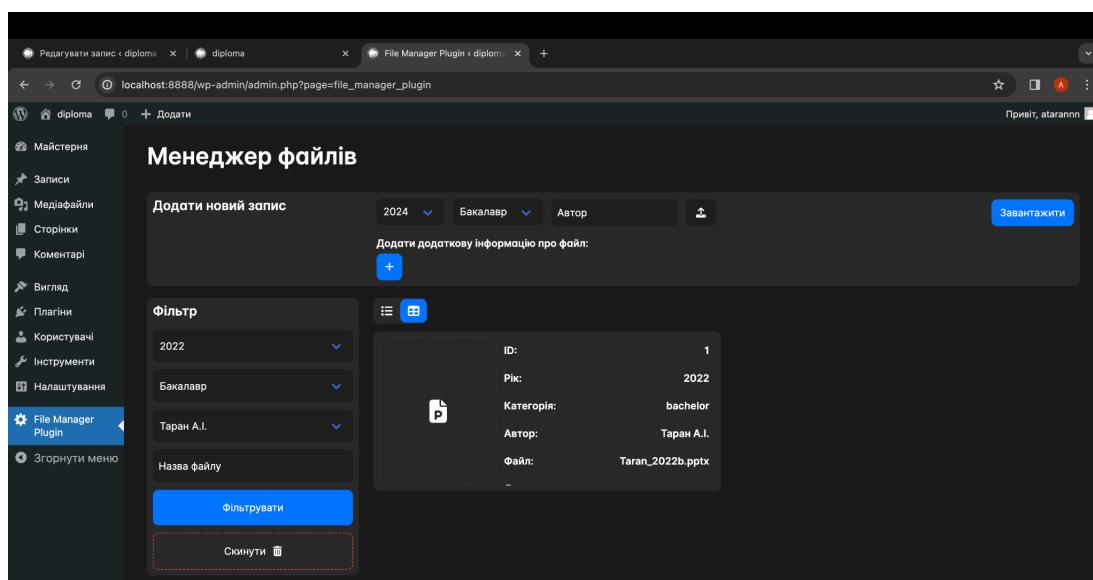


Рисунок 4.33 – Результат роботи фільтру при відображенні файлів плиткою.

Після виконання фільтрації ми визначили, що режим відображення плиткою працює коректно. Усі файли, які відповідають заданим параметрам, відображаються відповідно до встановленого режиму.

Тепер перейдемо до етапу тестування відображення файлів на веб-сайті за допомогою shortcode. Для цього перейдемо до вкладки "Записи", яка розташована в панелі навігації WordPress, і оберемо опцію "Додати новий запис". Після вибору цієї опції відкриється веб-інтерфейс, який дозволяє додати інформацію на веб-сайт (рис. 4.34).

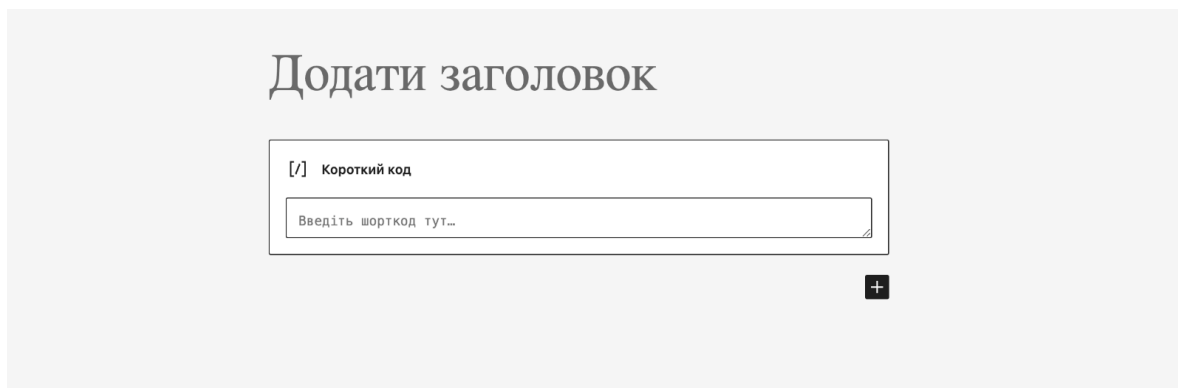


Рисунок 4.34 – Додавання інформації на веб-сайт.

Щоб вивести на веб-сайт інформацію про завантажені файли з файлової системи WordPress, натискаємо на кнопку "+" та обираємо опцію "Короткий код" (рис. 4.35).

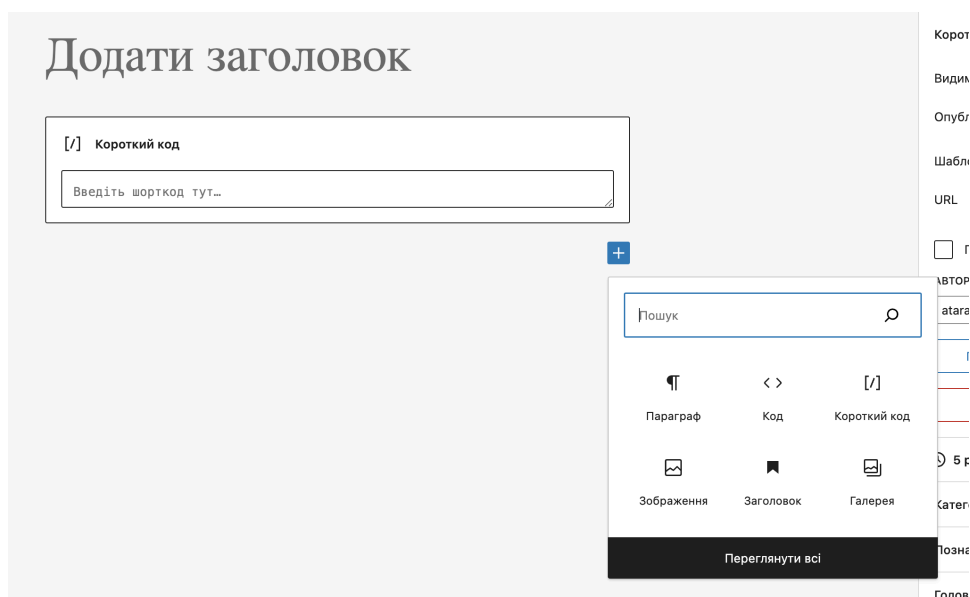


Рисунок 4.35 – Додавання інформації на веб-сайт: вибір виду інформації, котра буде відображатись на веб-сайті.

Після цього вводимо назву shortcode (рис. 4.36).

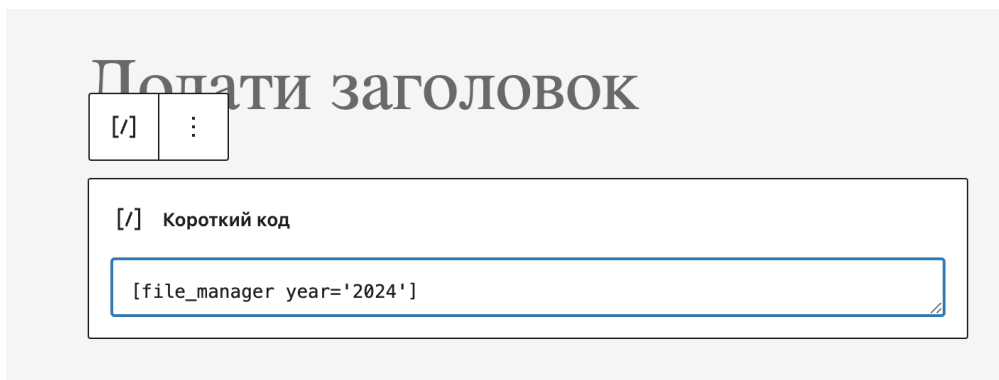


Рисунок 4.36 – Введення shortcode.

Після введення назви шорткоду, потрібно зберегти зміни, натиснувши кнопку "Опублікувати". Далі переходимо за посиланням "Переглянути веб-сайт", де можна побачити, що дані були успішно відображені на веб-сайті за допомогою введеного shortcode (рис. 4.37), а саме ми вивели всі файли які були створені в 2024 році.

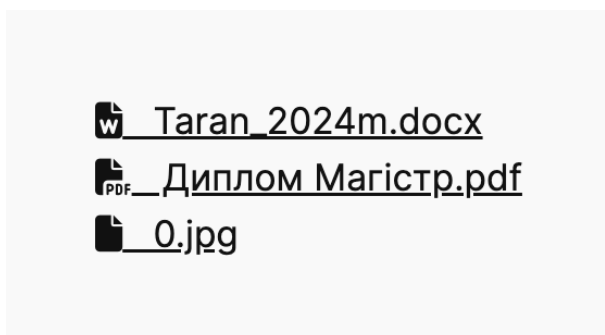


Рисунок 4.37 – Виведення файлів на веб-сайт.

4.3 Висновки до розділу

Розробка та тестування плагіна WordPress для управління файловою системою, були важливими кроками у вдосконаленні його функціональності та забезпеченні високого рівня користувацької зручності.

Додавання динамічних полів, оновлення інтерфейсу, інтеграція з бібліотекою Font Awesome та реалізація перемикача для відображення файлів дозволили створити потужний та привабливий інструмент для роботи з файлами на платформі WordPress. Функціонал виведення файлів на веб-сайт за допомогою shortcode виявився надзвичайно гнучким, надаючи користувачам різноманітні можливості відображення контенту.

В результаті проведення тестування було виявлено, що плагін працює стабільно та ефективно на різних конфігураціях, враховуючи останні версії PHP та WordPress.

5 РОЗРОБКА СТАРТАП-ПРОЕКТУ «РОЗРОБКА WORDPRESS ПЛАГІНА ДЛЯ КЕРУВАННЯ ФАЙЛОВОЮ СИСТЕМОЮ»

5.1 Опис ідеї проекту

Стартап-проект передбачає створення WordPress плагіна, спрямованого на оптимізацію та поліпшення керування файловою системою веб-сайтів. Основна ідея полягає в розширенні функціональності інструментів управління файлами, надавши користувачам більше можливостей для організації та редагування їхнього контенту.

Для здобуття повноцінного розуміння суті ідеї та ідентифікації основних потенційних ринків важливо створити детальну таблицю, яка охоплює різноманітні аспекти концепції. В табл. 2 буде надано опис ідеї, розглянуті можливі напрямки застосування та визначені основні вигоди для користувачів.

Таблиця 5.1 – Опис ідеї, напрямків застосування та вигоди для користувачів даного стартап-проекту.

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Ідеї стартап-проекту полягає в розробці WordPress плагіна, який спрямований на оптимізацію та поліпшення управління файловою системою веб-сайтів.	1. Забезпечення зручного та ефективного завантаження та організації файлів в одному централізованому місці.	Дозволяє користувачам ефективно витратити свій час на інші важливі завдання.
	2. Надання зручних інструментів для додавання інформації про файли.	Дозволяє користувачам керувати та налаштовувати інформацію відповідно до власних вимог.
	3. Надання можливості фільтрації та пошуку	Користувачі отримають можливість швидко

	файлів за різними критеріями.	знаходити необхідні файли за допомогою фільтрації та пошуку, що економить їхній час та збільшує продуктивність.
	4. Надання можливості користувачам швидко додавати завантажені файли на веб-сайт.	Простий та швидкий процес завантаження дозволяє користувачам ефективно виконувати цю рутинну задачу, зекономивши час та зусилля.

Отже, ідея стартапу полягає в створенні інструменту для користувачів, які мають потребу в ефективному керуванні великим обсягом файлів у системі управління вмістом (CMS) WordPress. Цей продукт забезпечить їм зручний і швидкий спосіб завантаження файлів, внесення додаткової інформації про них та зручну їх фільтрацію. Також, за допомогою даного рішення, користувачі зможуть значно зекономити час та зусилля, використовуючи його для швидкого відображення своїх файлів на веб-сайті WordPress.

Аналіз техніко-економічних переваг даної ідеї в порівнянні з існуючими аналогами є ключовим етапом формування стратегії конкурентоспроможності. У табл. 5.2 представлено огляд технічних та економічних параметрів, які визначають унікальність пропонованого стартап-проекту на ринку.

Цей аналіз служить основою для визначення стратегічних кроків у розвитку та позиціонуванні продукту на ринку. Перегляд цих параметрів допомагає зрозуміти, як саме дана ідея відрізняється від інших пропозицій і як можна максимізувати її переваги.

Таблиця 5.2 – Слабкі, нейтральні та сильні сторони ідеї проекту.

№ п/ п	Техніко- економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтра льна сторон а)	S (сильна сторона)
		Мій проект	Конкуре нт 1	Конкуре нт 2	Конкур ент 3			
1.	Форма виконання	WP плагін	WP плагін	WP плагін	WP плагін			+
2.	Інтерфейс користувача	інтуїтивн ий та відповідн ий до стандарті в UX/UI:	стандартн	інтуїтив ний	інтуїтив ний			+
3.	Маштабованість	висока	висока	висока	висока			+
4.	Гнучкість	висока	низька	середня	середня			+
5.	Керування та організація файлів	так	ні	так	так			+
6.	Потреба в технічних навичках	так	ні	так	так		+	
7.	Економічна вигода для користувача	так	ні	ні	ні			+

Підсумовуючи проведений аналіз, виокремлено кілька суттєвих сильних сторін, які визначають конкурентоспроможність даного продукту у контексті ринку управління файловою системою веб-сайтів:

1. Інтерфейс користувача: виокремлено високий рівень зручності та доступності інтерфейсу плагіна.
2. Гнучкість: наявність різноманітних налаштувань для відображення інформації про файли, що відповідає унікальним потребам кожного користувача.

3. Керування та організація файлів: визначено ефективні інструменти для організації та управління файлами, які сприяють легкості використання та підвищують продуктивність користувачів.
4. Економічна вигода для користувача: продукт забезпечує економічну вигоду для користувачів. Це включає збереження часу та ресурсів, які раніше витрачалися на складне управління файлами.

Враховуючи ці сильні сторони, даний продукт готовий конкурувати на ринку та задовольняти високі стандарти якості та ефективності управління файловою системою WordPress.

5.2 Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару).

Для реалізації проекту з розробки WordPress плагіна для керування файловою системою веб-сайтів рекомендується використовувати сучасні технології та інструменти, що забезпечать ефективний розвиток, високу продуктивність та зручність в користуванні.

В табл. 5.3 наведені деякі ключові технології, які необхідно використовувати для цього проекту:

Таблиця 5.3 – Перелік технологій для реалізації проекту.

№ п/п	Ідея проекту	Технології	Наявність технологій	Доступність технологій
	Ідеї стартап-проекту полягає в розробці WordPress плагіна,	JavaScript та AJAX	Наявна	Доступна
		PHP	Наявна	Доступна

	який спрямований на оптимізацію та поліпшення управління файловою системою веб-сайтів.	HTML5 та CSS3	Наявна	Доступна
--	--	---------------	--------	----------

WordPress плагіни пишуться в більшості випадків мовою програмування PHP. Так як, PHP - це серверна мова, яка використовується для розширення функціональності WordPress та взаємодії з базою даних. Окрім цього, для розробки фронтенду плагінів, таких як інтерфейси користувача або асинхронні запити, використовується JavaScript, включаючи бібліотеки, такі як jQuery. Також, під час розробки WordPress плагіна, необхідно використання мови HTML для розмітки та CSS для стилізації елементів плагіна.

Отже, для реалізації плагіна WordPress буде використовуватись комбінація PHP, JavaScript, HTML і CSS. Дані технології допоможуть розширити можливості цієї платформи.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку проводимо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку, результат якого продемонстровано в табл. 5.4.

Таблиця 5.4 – Попередня характеристика потенційного ринку для проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	10
2	Загальний обсяг продаж, грн/ум.од	1000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу	Необхідні розробники та початковий капітал
5	Специфічні вимоги до стандартизації та сертифікації	Наявність ліцензії GPLv2 для публікації плагіна
6	Середня норма рентабельності в галузі (або по ринку), %	25%

Отже, середня норма рентабельності в даній галузі менша, ніж банківський відсоток на вкладення. З цього можна зробити висновок, що вкладення коштів у цей проект є обґрунтованим, оскільки вимоги до стандартизації та сертифікації не є досить специфічними і досить просто виконуються.

Надалі визначимо потенційні групи клієнтів, їх характеристики, та сформуємо орієнтовний перелік вимог до товару для кожної групи у табл. 5.5.

Таблиця 5.5 - Характеристика потенційних клієнтів стартап-проекту.

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Оптимізація роботи файлової системи WordPress	Веб-розробники: особи, які відповідають за створення та підтримку веб-сайтів.	Веб-розробники можуть шукати плагіни з можливостями автоматичного виведення даних на веб-сайт.	WP плагін: Функціональний, адаптивний, гнучкий, має сучасний дизайн

2	Оптимізація роботи файлової системи WordPress	Адміністратори веб-сайтів: особи, які відповідають за адміністрування та оновлення вмісту веб-сайту.	Адміністратори веб-сайтів можуть бути зацікавлені в швидкому додаванні нових файлів з інтерфейсу адміністрування.	Компанія-постачальник: Підтримка та обслуговування, безпека
3	Оптимізація роботи файлової системи WordPress	Контент-менеджери: особи, які відповідають за завантаження та організацію	Контент-менеджери можуть оцінювати плагіни, які надають зручний інтерфейс для організації великої кількості медіафайлів.	

У результаті аналізу ми ідентифікували три ключові цільові аудиторії, які мають спільні потреби щодо оптимізації роботи файлової системи WordPress. Однак важливо враховувати відмінності в їхніх технічних навичках та конкретних потребах.

Після визначення потенційних груп клієнтів слід провести аналіз ринкового середовища шляхом формування таблиці факторів, що сприяють ринковому впровадженню проекту (табл. 5.7), та факторів, що йому перешкоджають (табл. 5.6).

Таблиця 5.6 – Фактори загроз

№ п/п	Загроза	Зміст загрози	Можлива реакція компанії
1	Конкуренція на ринку	Поява нових, більш привабливих для користувачів плагінів	Розробка та впровадження нових функцій, які вирішують актуальні проблеми користувачів та роблять продукт плагін більш привабливим.

2	Технічні труднощі	Зміни, внесені іншими розробниками, можуть впливати на роботу плагіна або навпаки.	Встановлення зв'язку із спільнотою розробників може допомогти заздалегідь отримувати інформацію про плановані зміни та адаптувати плагін відповідно.
3	Залежність від WordPress	Якщо WordPress внесе зміни, які не підтримують або навіть протистоять деяким функціям плагіна, це може викликати проблеми.	Компанія може використовувати тести сумісності для визначення, як плагін взаємодіє з CMS WordPress та забезпечити його найкращу працездатність.
5	Проблеми легітимності	Порушення авторських прав або ліцензій може викликати проблеми.	Посилення контролю за додержанням авторських прав та ліцензій,

Було наведено основні фактори загроз для стартап-проекту, з яких можна зробити висновки, що для успішного подолання загроз необхідно прагнути до гнучкості, активної взаємодії зі спільнотою розробників та оперативного реагування на зміни. Ретельне вивчення ринкових умов і вчасна адаптація дозволять зберегти конкурентоспроможність стартапу.

Таблиця 5.7 – Фактори можливостей

№ п/п	Можливість	Зміст можливості	Можлива реакція компанії
1	Оптимізація роботи з файлами	Зручні та швидкі інструменти для роботи з файловою системою.	Постійний аналіз та вдосконалення інтерфейсу та функціоналу.
2	Гнучкість плагіна	Можливість адаптації до різних потреб користувачів.	Продовження вдосконалення

			функціоналу, реагування на зміни.
3	Розширення функціоналу	Додавання нових корисних функцій та можливостей.	Активний розвиток та оновлення, залучення додаткових ресурсів.
4	Ефективний маркетинг	Спроможність ефективно рекламувати та популяризувати продукт.	Запуск маркетингових кампаній, участь у заходах та конференціях.

Ця таблиця підкреслює ключові можливості, які сприяють ринковому впровадженню проекту. Спроможність реагувати на зміни, активна співпраця та оптимізовані інструменти роботи з файлами формують основу для успіху у ринковому середовищі.

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку (табл. 5.8).

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Вказати тип конкуренції: олігополія	У сфері розробки WordPress плагінів існує кілька ключових учасників, що мають значний вплив.	Постійне оновлення продуктів та послуг для збереження та збільшення своєї частки національного ринку
За рівнем конкурентної боротьби: інтернаціональна	Більшість конкурентів є закордонними.	Розвиток маркетингових стратегій, які акцентують увагу на закордонному споживачеві

За галузевою ознакою: внутрішньогалузева	Конкуренція відбувається всередині галузі розробки WordPress плагінів для управління файловою системою.	Активна участь у вирішенні галузевих проблем та розвитку нових технологій для збільшення конкурентоспроможності
Конкуренція за видами товарів: товарно-видова	Виробництво товарів специфічних видів та врахування потреб конкретної групи клієнтів	Виробництво товару, який відповідає потребам різних сегментів ринку
За характером конкурентних переваг: нецінова	Зосередженість на якісних та функціональних особливостях продукту	Інновації у функціональності та якості продукту, щоб збільшити задоволення клієнтів та залучити нових
За інтенсивністю: марочна	Кожен конкурент прагне визначити себе унікальними рішеннями.	Зміцнення та розвиток бренду для підсилення ідентичності та відзначення від конкурентів

Ці характеристики визначають особливості конкурентного середовища і можуть служити основою для розробки стратегій підприємства для підтримання та розвитку його конкурентоспроможності.

Далі проведемо більш детальний аналіз умов конкуренції в галузі (за моделлю 5 сил М. Портера) (табл. 5.9).

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Real Media Library	Бар`єр - отримання ліцензії	WordPress Сильні сторони:	Відгуки та оцінки користувачів	-

			стабільний, популярний		
Висновки:	Популярні плагіни, які активно оновлюють свій функціонал, хоч і досить обмежений	Потенційно може опублікуватись будь-який плагін з необхідним функціоналом для користувачів	Так, диктують певні вимоги, які впливають на публікацію плагінів	Диктують умови	-

Отже, дана таблиця дає можливість впевнено стверджувати, що проект має великий потенціал на ринку. Серед представлених конкурентів відсутні ті, що здатні повністю конкурувати з стартап-проектом, оскільки дане рішення значно полегшує та прискорює роботу фахівця.

На основі аналізу результатів, отриманих під час попередніх аналізів, визначається та обґрунтовується перелік факторів конкурентоспроможності (табл. 5.10).

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Ринкова потреба	Існує велика ринкова потреба в даному продукті, так як поки що немає готового рішення з необхідним функціоналом

2	Якість продукту	Висока якість продукту виділяє його серед конкурентів і напряду впливає на задоволення користувачів.
---	-----------------	--

За визначеними факторами конкурентоспроможності з табл. 5.10 проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 5.11).

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін «Розробка плагіна WordPress для керування файловою системою»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)						
			-3	-2	-1	0	+1	+2	+3
1	Ринкова потреба	18			+				
2	Якість продукту	18			+				

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (табл. 5.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 5.11).

Таблиця 5.12 – SWOT- аналіз стартап-проекту

Сильні сторони: сучасний дизайн, зручний інтерфейс та широкий функціонал	Слабкі сторони: відсутність відгуків, мала кількість користувачів
Можливості: кількість сайтів на платформі WordPress постійно зростає і це створює потребу в розробці плагінів, для управління файловою системою даної CMS.	Загрози: поява нових, більш привабливих для користувачів плагінів та наявність аналогічних продуктів, які довго знаходяться на ринку і мають більше

	ресурсів та своїх постійних користувачів
--	--

Далі, на основі SWOT-аналізу розробляються альтернативи ринкової поведінки (табл. 5.13) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Розширення функціоналу плагіна, що як результат витісняє конкурентів	50%	12 місяців
2	Вихід на ринок з обмеженим функціоналом і поступове розширення плагіна	40%	6 місяців

Отже, було визначено альтернативу ринкового впровадження стартап-проекту - вихід на ринок з обмеженим функціоналом і поступове розширення функціоналу плагіна. Так як час на реалізацію додаткових функцій є значно більшим та не пропонує значно більшу ймовірність отримання ресурсів.

5.4 Розробка ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 5.14).

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Установи, яким потрібно оптимізувати роботу з великими обсягом файлів на їх веб-сайтах WordPress	Готові якнайшвидше	Високий	Інтенсивна	Середня
2	Компанії, які містять великий обсяг документів і хочуть їх демонструвати відвідувачам на їх WordPress сайтах	Готові якнайшвидше	Середній	Інтенсивна	Середня
Які цілові групи обрано: обрано обидві цілові групи					

Отже, було обрано наступні цілові групи: установи, яким потрібно оптимізувати роботу з великими обсягом файлів на їх веб-сайтах WordPress та компанії, які містять великий обсяг документів і хочуть їх демонструвати відвідувачам на їх WordPress сайтах, так як обидві групи готові відразу сприйняти продукт і мають схожу мету - оптимізація роботи в файловій системі WordPress.

Далі визначається базова стратегія розвитку (табл. 5.15).

Таблиця 5.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Створення продукту, який буде відрізнятись від конкурентів та матиме широкий спектр властивостей, необхідних для користувачів	Створення потужного маркетингу	Надання продукту важливих з точки зору споживача властивостей, які роблять його відмінним від конкурентів.	Стратегія диференціації

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 5.16).

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідце м» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Ні	Так	Ні, буде проводитись моніторинг оновлень в продукті конкурента і в разі виявлення корисного функціоналу він буде реалізовуватись в нашому продукті	Наступальна

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розробляється стратегія позиціонування (табл. 5.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 5.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентноспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Коректна робота, зручний інтерфейс, великий вибір функцій	Стратегія диференціації	Гнучкість є основною позицією даного стартап-проекту	Сучасний інтерфейс, гнучкість, персоналізація

Отже, було визначено стратегію позиціонування, а саме визначено основні вимоги до товару цільової аудиторії: Коректна робота, зручний інтерфейс, великий вибір функцій. Також було визначено базову стратегію розвитку: диференціація. Ключові конкурентноспроможні позиції стартап-проекту: гнучкість, адже є ключовою перевагою даного стартап-проекту. Також сформовано комплексну позицію проекту: сучасний інтерфес, гнучкість, персоналізація.

5.5 Розробка маркетингової програми

Першим кроком є формування маркетингової концепції товару, який повинен отримати споживач. Для цього у табл. 5.18 підсумуємо результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Оптимізація роботи файлової системи WordPress	Продукт є простим та зручним у використанні, так як пропонує інтуїтивний інтерфейс та можливість гнучких налаштувань для файлів користувачів.	Персоналізований підхід

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту, його фізичні складові, особливості процесу його надання (табл. 5.19).

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Оптимізація роботи файлової системи WordPress. Внаслідок використання даного плагіна, користувач, матиме можливість ефективно завантажувати та організовувати файли в одному централізованому місці, матиме зручні інструменти для додавання інформації про файли, матиме можливість фільтрації та пошуку

	файлів за різними критеріями та зможе швидко додавати завантажені файли на веб-сайт, без особливих технічних навичок.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Сучасний інтерфейс.	М	Е
	2. Гнучкість.	М	Тл
	3. Персоналізація	М	Тл
	Оптимізує роботу установ, які працюють з великим обсягом файлів		
	Плагін WordPress, який користувачі встановлюють в свій WordPress проект		
	File Manager Plugin		
III. Товар із підкріпленням	-		
	Підтримка, тьюторіал до використання		
За рахунок чого потенційний товар буде захищено від копіювання: даний продукт буде завантажено та ретельно перевірено через платформу WordPress.org. Після успішної публікації, авторство плагіна буде закріплене.			

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (табл. 5.20). Аналіз проводиться експертним методом.

Таблиця 5.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінни ки	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	1200 грн. / місяць	2000 грн. / місяць	>100000 грн. / місяць	1500-2000 грн. / місяць

Визначено межі встановлення ціни на користування WordPress плагіном, а саме рівень цін на товари-замінники – 1200 грн. за 1 місяць користування, рівень цін на товари-аналоги – 2000 грн. за 1 місяць користування, рівень доходів цільової групи споживачів - більше 100000 грн на місяць, а верхня та нижня межі встановлення ціни на товар – 1500-2000 грн за 1 місяць користування.

Наступним кроком є визначення оптимальної системи збуту (табл. 5.21).

Таблиця 5.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Клієнти зможуть оцінити можливості платформи через безкоштовну версію з обмеженим функціоналом	Фінансування збутових операцій, обслуговування проданих товарів	Канал нульового рівня	Проводити збут власними силами

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій (табл. 5.22).

Таблиця 5.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
-------	---------------------------------------	--	--	----------------------------------	--------------------------------

1	Установи, які прагнуть мати весь необхідний функціонал в одному місці	Електронна пошта, соціальні мережі	Гнучкість	Привернення уваги до продукту, заохочення потенційних клієнтів спробувати продукт	Ви заслуговуєте на краще – спробуйте наш плагін вже сьогодні та переконайтеся, наскільки легко може бути управління файловою системою!
---	---	------------------------------------	-----------	---	--

В результаті, було розроблено концепцію маркетингових комунікацій, яка включає в себе: специфіку поведінки цільових клієнтів - прагнення клієнта мати весь необхідний функціонал в одному місці: канали комунікацій, якими користуються цільові клієнти - електронна пошта та соціальні мережі; ключові позиції, обрані для позиціонування – гнучкість; завдання рекламного повідомлення - привернути увагу до продукту, заохотити потенційних клієнтів спробувати продукт; концепцію рекламного звернення – ви заслуговуєте на краще, тому спробуйте наш плагін вже сьогодні та переконайтеся, наскільки легко може бути.

5.6 Висновки до розділу

Даний розділ був присвячений дослідженню основних аспектів виходу на ринок плагіна WordPress для керування файловою системою.

Аналіз ринку підтвердив наявність попиту на рішення для оптимізації роботи з файловою системою WordPress. Широкий спектр потенційних

користувачів, від розробників до адміністраторів веб-сайтів, свідчить про високий інтерес до такого продукту.

Оптимізація роботи з файловою системою набуває все більшої важливості, що створює додаткові можливості для успішного впровадження продукту на ринок. Гнучкість налаштувань, для відображення інформації про файли, що відповідають унікальним потребам користувачів, та сучасний і зручний інтерфейс надають даному продукту значні переваги перед конкурентами.

Розглянувши потенційні групи клієнтів, проаналізувавши бар'єри входження та оцінивши конкурентоспроможність проекту, були визначені чіткі перспективи впровадження даного продукту.

Загалом, наявність попиту, позитивної динаміки ринку та переваг проекту створюють сприятливі умови для його успішної ринкової комерціалізації та подальшого розвитку.

ВИСНОВКИ

В результаті виконання магістерської дисертації було проведено аналіз існуючих рішень у сфері управління файловою системою WordPress, а також детальний аналіз проблем плагіна, розробленого в рамках виконання бакалаврської дипломної роботи. Результати аналізу дали можливість визначити ключові вади та обмеження існуючого плагіна, які слугували основою для формулювання вимог до нового рішення.

Завдяки чітким вимогам, продукт успішно пройшов етап розробки нового функціоналу, який ефективно вирішив виявлені проблеми у попередній версії плагіна. У рамках проведеного дослідження та розробки було впроваджено ряд значущих покращень у функціонал плагіна для управління файловою системою в екосистемі WordPress:

1. Було успішно реалізовано можливість додавання додаткової інформації про файли через використання динамічних полів. Це дає можливість користувачам ефективно та гнучко керувати та адаптувати інтерфейс плагіна до своїх потреб.
2. Інтерфейс плагіна був значно поліпшений та оновлений для забезпечення зручного та інтуїтивно зрозумілого використання. Це дозволяє користувачам швидше орієнтуватися та використовувати всі функції плагіна без зайвих зусиль.
3. Здійснено успішну інтеграцію плагіна з бібліотекою Font Awesome, що дозволяє використовувати різноманітні іконки в інтерфейсі плагіна. Це не лише поліпшує візуальний аспект, а й надає додаткові можливості для виразного відображення різних типів файлів та їх властивостей.
4. Реалізовано функціонал для вибору режиму відображення файлів, завдяки якому було досягнуто високого рівня гнучкості плагіна. Це дозволяє

кожному користувачеві працювати з файлами так, як йому більш зручно та ефективно.

5. Функціонал виведення файлів на веб-сайт було розширено, щоб забезпечити більше можливостей та варіативності в представленні інформації про файли. Забезпечення різноманітних варіантів виведення дозволяє користувачам ефективно презентувати свою інформацію.

Новий функціонал не лише усунув існуючі недоліки, але й позитивно підтвердив свою працездатність під час тестування. Результати тестів свідчать не лише про успішну роботу плагіна в цілому, але й про його повну відповідність поставленим вимогам, що відкриває перспективи для його подальшого розвитку та вдосконалення, однією з яких може стати впровадження інструментів для редагування та перегляду файлів прямо через плагін WordPress.

Проведений аналіз ринку є переконливим доказом високого рівня зацікавленості та попиту на розроблений продукт. Це свідчить не лише про актуальність плагіна WordPress для керування файловою системою, але й підкреслює його важливість для широкого спектру користувачів. З таким визнанням і підтримкою на ринку важливо продовжувати розширювати функціонал даного продукту. Розробка нових можливостей та інтеграція додаткового функціоналу може допомогти розширити його застосування та забезпечити задоволення потреб різних категорій користувачів, сприяючи довготривалому успіху та конкурентоспроможності плагіна на ринку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Таран, А. І. Розробка плагіна WordPress для автоматизації створення структури файлової системи за допомогою маркерів : дипломна робота ... бакалавра : 122 Комп'ютерні науки / Таран Анастасія Ігорівна. - Київ, 2022. - 96 с.

URI: <https://ela.kpi.ua/handle/123456789/53858>

2. Портал знань. Системи керування контентом

URL: [1. Системи керування контентом | Портал знань, портал знаний, дистанційне навчання](#). (дата звернення: 25.12.2023).

3. Червоний хамелеон. Створення та розробка сайтів на WordPress

URL: [Створення сайту на CMS WordPress | Заповіти розробку сайту на Вордпрес з нуля](#) (дата звернення: 25.12.2023).

4. Ольга Сомова, Оля Сомова.Webpromo. Чому 43% всіх сайтів створено на WordPress?

URL: [Чому 43% всіх сайтів створено на Wordpress? – Webpromo](#) (дата звернення: 25.12.2023).

5. [Aigars Silkalns](#).Colorlib. 7 Best WordPress File Manager Plugins 2023

URL: [7 Best WordPress File Manager Plugins 2023 - Colorlib](#) (дата звернення: 25.12.2023).

6. Real Media Library: Media Library Folder & File Manager

URL: [Media Library Folder & File Manager – WordPress плагін](#) (дата звернення: 25.12.2023).

7. FileBird – WordPress Media Library Folders & File Manager

URL: FileBird – WordPress Media Library Folders & File Manager (дата звернення: 25.12.2023).

8. Romul.name. Bulma сучасний CSS фреймворк

URL: Bulma сучасний CSS фреймворк - Фріланс Веб дизайнер та розробник (дата звернення: 25.12.2023).

9. Системні науки та інформатика: збірник доповідей II науково-практичної конференції «Системні науки та інформатика», 4–8 грудня 2023 року, Київ. – К., НН ІПСА КПІ ім. Ігоря Сікорського, 2023. – 416 с.

10. Darina. Komarov Design. 10 принципів гарного дизайну від Дітера Рамса

URL: 10 принципів гарного дизайну від Дітера Рамса (дата звернення: 25.12.2023).

11. Богдана Данилейко. EffectM. Секрети чорних тем в дизайні інтерфейсів та сайтів

URL: Секрети чорних тем в дизайні інтерфейсів та сайтів (дата звернення: 25.12.2023).

12. Yura Forostyuk. DOU. Актуальність WordPress: чому з ним треба працювати у 2023 році

URL: Актуальність WordPress: чому з ним треба працювати у 2023 році | DOU (дата звернення: 25.12.2023).

13. URL: Font Awesome (дата звернення: 25.12.2023).

ДОДАТОК А

Фрагменти лістингу програми

- index.php

```
<?php
/*
 * Plugin Name: File Manager Plugin
 * Description: Плагін для керування файловою системою
 * Version: 2.0.0
 * Author: Anastasia Taran
 */

add_action('admin_menu', 'file_manager_plugin');
add_shortcode('file_manager', 'file_manager_shrt');
session_start();

function file_manager_plugin(){
    add_menu_page( 'File Manager Plugin', 'File Manager Plugin', 'manage_options',
    'file_manager_plugin', 'file_manager_init' );

    wp_enqueue_style('custom-styles', plugins_url('/style.css', __FILE__));
    wp_enqueue_style('font-awesome',
    'https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css',
    array(), '5.15.3', 'all');

    wp_enqueue_script('file-manager',
    'https://code.jquery.com/jquery-3.6.0.min.js');
    wp_localize_script('file-manager', 'myAjax', array( 'ajaxurl' => admin_url(
    'admin-ajax.php' )));
    wp_enqueue_script('file-manager-script', plugins_url('/script.js', __FILE__));
}

function file_manager_shrt($atts) {
    global $wpdb;
    $table_name = $wpdb->prefix . "diploms_archive";
    $default = array(
        'year' => '',
        'autor' => '',
        'category' => ''
    );
    $a = shortcode_atts($default, $atts);
    $filters = '';

    $conditions = array();
    if (!empty($a['year'])) {
        $conditions[] = "`year` = '{$a['year']}'";
    }
    if (!empty($a['autor'])) {
        $conditions[] = "`autor` = '{$a['autor']}'";
    }
}
```

```

    }
    if (!empty($a['category'])) {
        $conditions[] = "`category` = '{$a['category']}'";
    }

    if (!empty($conditions)) {
        $filters = ' WHERE ' . implode(' AND ', $conditions);
    }

    $myrows = $wpdb->get_results("SELECT `id`, `year`, `category`, `autor`, `file`
FROM
$table_name$filters;");

    $res = '';

    foreach ($myrows as $details) {
        $format = pathinfo($details->file, PATHINFO_EXTENSION); // Отримання
        формату файлу

        $iconClass = 'fa-file';
        switch ($format) {
            case 'docx':
                $iconClass = 'fa-file-word';
                break;
            case 'pdf':
                $iconClass = 'fa-file-pdf';
                break;
            case 'txt':
                $iconClass = 'fa-file-lines';
                break;
            case 'pptx':
                $iconClass = 'fa-file-powerpoint';
                break;
            case 'zip':
                $iconClass = 'fa-file-zipper';
                break;
        }

        $res .= "<a class='shortcode' href='" . wp_get_upload_dir()['baseurl'] .
"/diploms/" . $details->file . "' target='_blank'>
        <i class='fas $iconClass'></i>
        &nbsp;&nbsp;&nbsp;" . $details->file . "
        </a><br>";
    }
    return $res;
}

function file_manager_init(){
    global $wpdb;
    $table_name = $wpdb->prefix . "diploms_archive";

    if($wpdb->get_var("show tables like '$table_name'") != $table_name) {

```

```

        $sql = "CREATE TABLE " . $table_name . " (
            id mediumint(9) NOT NULL AUTO_INCREMENT,
            year year(4) NOT NULL,
            category varchar(255) NOT NULL,
            autor varchar(255) NOT NULL,
            file varchar(255) NOT NULL,
            posts_id mediumint(9) NOT NULL,
            custom_data JSON DEFAULT NULL,
            UNIQUE KEY id (id)
        );";

        require_once(ABSPATH . 'wp-admin/includes/upgrade.php');
        dbDelta($sql);
    }
    ?>

<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bulma@0.9.3/css/bulma.min.css">

<div class="file-manager">
    <div class="file-manager__head">
        <h1 class="title">Менеджер файлів</h1>
    </div>
    <div class="file-manager__inner">
        <div class="file-loader">
            <h2 class="title">Додати новий запис</h2>
            <form method="post" enctype="multipart/form-data" id="add">
                <div class="row-main-field">
                    <div class="control">
                        <div class="select">
                            <select name="year">
                                <option value="2000">2000</option>
                                <option value="2001">2001</option>
                                <option value="2002">2002</option>
                                <option value="2003">2003</option>
                                <option value="2004">2004</option>
                                <option value="2005">2005</option>
                                <option value="2006">2006</option>
                                <option value="2007">2007</option>
                                <option value="2008">2008</option>
                                <option value="2009">2009</option>
                                <option value="2010">2010</option>
                                <option value="2011">2011</option>
                                <option value="2012">2012</option>
                                <option value="2013">2013</option>
                                <option value="2014">2014</option>
                                <option value="2015">2015</option>
                                <option value="2016">2016</option>
                                <option value="2017">2017</option>
                                <option value="2018">2018</option>
                                <option value="2019">2019</option>
                                <option value="2020">2020</option>

```

```

                <option value="2021">2021</option>
                <option value="2022">2022</option>
                <option value="2023">2023</option>
                <option value="2024" selected>2024</option>
            </select>
        </div>
    </div>
    <div class="control">
        <div class="select">
            <select name="category">
                <option value="bachelor"
selected>Бакалавр</option>
                <option value="master">Магістр</option>
                <option value="postgraduate">Аспірант</option>
            </select>
        </div>
    </div>
    <div class="control autor-field">
        <input class="input" type="text" placeholder="Автор"
name="autor">
    </div>
    <div class="file is-primary">
        <label class="file-label">
            <input class="file-input" type="file"
id='text_file' name='text_file'>
            <span class="file-cta">
                <i class="fa-solid fa-upload" style="color:
#ffffff;"></i>
            </span>
        </label>
    </div>
    <input class="submit" type="submit" value="Завантажити"
onclick="add(event)">
</div>
<div class="container new-fields">
    <p class="title">Додати додаткову інформацію про файл:</p>
    <div class="row-new-fields">
        <button class="btn btn-primary add_form_field"><i
class="fa-solid fa-plus" style="color: #ffffff;"></i></button>
        <div class="row-new-fields__inner">

            </div>
        </div>
    </div>
</form>
</div>
<div class="navigation">
    <div class="navigation__filter">
        <div class="navigation__filter-head">
            <h2 class="title">Фільм</h2>
        </div>
        <form method="post" id="filter-wrapper" class="form-filter">

```

```

<div class="select">
    <select id="filterByYear" name="filterByYear">
        <option value="2000">2000</option>
        <option value="2001">2001</option>
        <option value="2002">2002</option>
        <option value="2003">2003</option>
        <option value="2004">2004</option>
        <option value="2005">2005</option>
        <option value="2006">2006</option>
        <option value="2007">2007</option>
        <option value="2008">2008</option>
        <option value="2009">2009</option>
        <option value="2010">2010</option>
        <option value="2011">2011</option>
        <option value="2012">2012</option>
        <option value="2013">2013</option>
        <option value="2014">2014</option>
        <option value="2015">2015</option>
        <option value="2016">2016</option>
        <option value="2017">2017</option>
        <option value="2018">2018</option>
        <option value="2019">2019</option>
        <option value="2020">2020</option>
        <option value="2021">2021</option>
        <option value="2022">2022</option>
        <option value="2023">2023</option>
        <option value="2024" selected>2024</option>
    </select>
</div>
<div class="select">
    <select name="category" id="filterByCategory">
        <option value="bachelor" selected>Бакалавр</option>
        <option value="master">Магистр</option>
        <option value="postgraduate">Аспирант</option>
    </select>
</div>
<div class="select">
    <select id="filterByAutor" name="filterByAutor">
        <option selected value="">Автор</option>
        <?php
            global $wpdb;
            $table_name = "wp_diploms_archive";
            $myrows = $wpdb->get_results( "SELECT DISTINCT
`autor` FROM ".$table_name."");
            foreach ($myrows as $autor){
                if(!empty($autor->autor)) echo '<option
value="'. $autor->autor.'">'. $autor->autor.'">';
            }?>
    </select>
</div>
<div class="control autor-field">

```



```

        <input type="text" placeholder="Назва файлу"
id="filterByName" name="filterByName">
    </div>
    <button id="filter-btn"
onclick="btnClick(event)">Фільтрувати</button>
    <button id="reset-btn" onclick="btnReset(event)">
        Скинути
    <i class="fa-solid fa-trash-can" style="color:
#ffffff;"></i>
    </button>
</form>
</div>
<div class="navigation__inner_wrapper">
    <label class="switch">
        <input type="checkbox" id="viewSwitch">
        <span class="slider round">
            <i class="fa-solid fa-list" style="color:
#ffffff;"></i>
            <i class="fa-solid fa-table" style="color:
#ffffff;"></i>
        </span>
    </label>
    <div class="navigation__inner"></div>
</div>
</div>
</div>
<?php
}

function addDiplopm() {
    if(isset($_FILES['text_file'])){
        $pdf = $_FILES['text_file'];

        function wpse_custom_upload_dir( $dir_data ) {
            $custom_dir = 'diploms';
            return [
                'path' => $dir_data[ 'basedir' ] . '/' . $custom_dir,
                'url' => $dir_data[ 'url' ] . '/' . $custom_dir,
                'subdir' => '/' . $custom_dir,
                'basedir' => $dir_data[ 'error' ],
                'error' => $dir_data[ 'error' ],
            ];
        }

        add_filter( 'upload_dir', 'wpse_custom_upload_dir' );

        $uploaded=media_handle_upload('text_file', 0);

        if(is_wp_error($uploaded)){
            $message = $uploaded->get_error_message();
            if($message == 'Sorry, you are not allowed to upload this file type.'){

```

```

        $message = 'Вибачте, вам не дозволено завантажувати цей тип
файлів.';
    }
    echo json_encode(['result'=>0,'message'=>$message]);
}
else{
    global $wpdb;
    $lastrowId=$wpdb->get_col( "SELECT ID FROM wp_posts where
post_type='attachment' ORDER BY post_date DESC LIMIT 1" );

    $year = $_POST['year'];
    $category = $_POST['category'];
    $autor = $_POST['autor'];
    $file = $_FILES['text_file']['name'];
    $custom_data = json_encode($_POST['row'],JSON_UNESCAPED_UNICODE);

    $sql = "INSERT INTO `wp_diploms_archive`
(`year`,`category`,`autor`,`file`,`posts_id`,`custom_data`)
values ('$year', '$category', '$autor', '$file',
'$lastrowId[0]', '$custom_data')";
    $success = $wpdb->query($sql);
    echo json_encode(['result'=>1,'message' => $success]);
}
}
die();
}

function getFilteredRows($year, $autor, $name, $category) {
    global $wpdb;

    $table_name = "wp_diploms_archive";
    $query = "SELECT `id`,`year`,`category`,`autor`,`file` FROM " . $table_name . "
WHERE `year`='$year'";
    if (!empty($autor)) $query .= " AND `autor`='$autor'";
    if (!empty($name)) $query .= " AND `file` LIKE '%$name%'";
    if (!empty($category)) $query .= " AND `category`='$category'";
    return $wpdb->get_results($query);
}

function filterDiploms($data = NULL){
    $filterFields = array('year', 'autor', 'fileName', 'category');

    foreach ($filterFields as $field) {
        $value = !empty($_POST['data'][$field]) ? $_POST['data'][$field] :
$_SESSION[$field];
        $_SESSION[$field] = $value;
        $$field = $value;
    }

    $myrows = ($data != NULL) ? $data : getFilteredRows($year, $autor, $name,
$category);

    $output['data'] = filesView($myrows, $_SESSION['type']);
}

```

```

$output['switch'] = getSwitch();
echo json_encode($output);
die();
}

function getFilesInfo(){
    global $wpdb;
    $table_name = "wp_diploms_archive";
    $myrows = $wpdb->get_results( "SELECT `id`,`year`,`category`,`autor`,`file`,`
custom_data` FROM ".$table_name);
    return $myrows;
}

function filesView($myrows, $viewType) {
    $output = '';
    if ($viewType == 'card') {
        $_SESSION['type'] = 'card';
        $output .= '<div class="navigation_card">';

        foreach ($myrows as $details) {
            // Декодуємо JSON-рядок у масив об'єктів
            $customData = json_decode($details->custom_data, true);

            // Перевіряємо наявність та отримуємо значення за ключами
            $value1 = isset($customData['1']) ? $customData['1'] : '';
            $value2 = isset($customData['2']) ? $customData['2'] : '';
            $value3 = isset($customData['3']) ? $customData['3'] : '';
            $format = explode('.', $details->file);

            $iconClass = '';
            switch ($format[1]) {
                case 'docx':
                    $iconClass = 'fa-file-word';
                    break;
                case 'pdf':
                    $iconClass = 'fa-file-pdf';
                    break;
                case 'txt':
                    $iconClass = 'fa-file-lines';
                    break;
                case 'pptx':
                    $iconClass = 'fa-file-powerpoint';
                    break;
                case 'zip':
                    $iconClass = 'fa-file-zipper';
                    break;
                default:
                    $iconClass = 'fa-file';
                    break;
            }

            $output .= '<div class="card">

```

```

        <div class="card_file"><i class="fa-solid ' . $iconClass . '"
style="color: #ffffff;"></i></div>
        <div class="card_info">
            <div
class="row"><span>ID:</span><div>' . $details->id.' </div></div>
            <div
class="row"><span>Пік:</span><div>' . $details->year.' </div></div>
            <div
class="row"><span>Категорія:</span><div>' . $details->category.' </div></div>
            <div
class="row"><span>Автор:</span><div>' . $details->autor.' </div></div>
            <div
class="row"><span>Файл:</span><div>' . $details->file.' </div></div>
            <div class="row"><span>Додаткове поле
№1</span><div>' . $value1.' </div></div>
            <div class="row"><span>Додаткове поле
№2</span><div>' . $value2.' </div></div>
            <div class="row"><span>Додаткове поле
№3</span><div>' . $value3.' </div></div>
            <div class="row"><span>Shortcode:</span><div><input readonly
value="[file_manager id=' . $details->id.' ]" /></div></div>
        </div>
    </div>';
}

$output .= '</div>';
} else {
    $_SESSION['type'] = 'table';
    $output .= '<div class="navigation_table">';
    $output .= '<table class="table">';
    $output .= '<thead>
        <tr>
            <th>ID</th>
            <th>Пік</th>
            <th>Категорія</th>
            <th>Автор</th>
            <th>Файл</th>
            <th>Додаткове поле №1</th>
            <th>Додаткове поле №2</th>
            <th>Додаткове поле №3</th>
            <th>Shortcode</th>
        </tr>
    </thead>
    <tbody>';
    foreach ($myrows as $details) {
        // Декодуємо JSON-рядок у масив об'єктів
        $customData = json_decode($details->custom_data, true);

        // Перевіряємо наявність та отримуємо значення за ключами
        $value1 = isset($customData['1']) ? $customData['1'] : '';
        $value2 = isset($customData['2']) ? $customData['2'] : '';
        $value3 = isset($customData['3']) ? $customData['3'] : '';
    }
}

```

```

        $output .= '<tr class="item_row">
                    <td>'.$details->id.'</td>
                    <td>'.$details->year.'</td>
                    <td>'.$details->category.'</td>
                    <td>'.$details->autor.'</td>
                    <td>'.$details->file.'</td>
                    <td>'.$value1.'</td>
                    <td>'.$value2.'</td>
                    <td>'.$value3.'</td>
                    <td><input readonly value="[file_manager id='.$details->id.'])"
                /></td>
                </tr>';
    }
    $output .= '</tbody></table></div>';
}

return $output;
}

function paintFiles() {
    $myrows = getFilesInfo();
    $viewType = isset($_POST['action']) ? $_POST['action'] : '';
    if($_POST['action'] == 'card' || $_POST['action'] == 'table'){
        $_SESSION['type'] = $viewType;
        if(empty($_SESSION['year'])) {
            filterDiploms($myrows);
        }else{
            filterDiploms();
        }
    }
    if($_POST['action'] == 'resetFilter'){
        unset($_SESSION['year']);
        unset($_SESSION['autor']);
        unset($_SESSION['fileName']);
        unset($_SESSION['category']);
    }

    $output['data'] = filesView($myrows, $_SESSION['type']);
    $output['switch'] = getSwitch();
    echo json_encode($output);
    die();
}

function shortcode($form_fields, $post) {
    global $wpdb;
    $post_id = $post->ID;
    $archiv_ids=$wpdb->get_col( "SELECT id FROM wp_diploms_archive where
posts_id=".$post_id);

    $form_fields["rt-short-code"] = array(
        "label" => __("Shortcode"),

```

```

        "input" => "html",
        "value" => "[file_manager id=".$sarchiv_ids[0]."]"
    );
    return $form_fields;
}

function getSwitch()
{
    if(!empty($_SESSION['type'])) {
        return $_SESSION['type'];
    }else{
        return 'table';
    }
}

add_action('wp_ajax_nopriv_getSwitch','getSwitch');
add_action('wp_ajax_getSwitch','getSwitch');

add_action('wp_ajax_nopriv_add','addDiplopm');
add_action('wp_ajax_add','addDiplopm');

add_action('wp_ajax_nopriv_filter','filterDiploms');
add_action('wp_ajax_filter','filterDiploms');

add_action('wp_ajax_nopriv_resetFilter','paintFiles');
add_action('wp_ajax_resetFilter','paintFiles');

add_action('wp_ajax_nopriv_table','paintFiles');
add_action('wp_ajax_table','paintFiles');

add_action('wp_ajax_nopriv_card','paintFiles');
add_action('wp_ajax_card','paintFiles');

add_filter("attachment_fields_to_edit", "shortcode", null, 2);
?>

```

- script.js

```

function filterById(id) {
    var e = document.getElementById(id);
    return e.value;
}

function btnClick(e) {
    e.preventDefault();
    var frm = jQuery('#filter-wrapper');
    var year = filterById("filterByYear");
    var autor = filterById("filterByAutor");
    var fileName = filterById("filterByName");
    var category = filterById("filterByCategory");
    jQuery.ajax({

```

```

        method: 'POST',
        type: 'POST',
        data: {
            action: 'filter',
            data: {year:year, autor:autor, fileName:fileName, category:category},
        },
        url: myAjax.ajaxurl,
        success: function (data) {
            var resultObject = JSON.parse(data);

            let elem = document.querySelector('.navigation__inner');
            elem.innerHTML = resultObject.data;
        }
    });
}

function btnReset(e) {
    e.preventDefault();
    jQuery.ajax({
        method: 'POST',
        type: 'POST',
        data: {
            action: 'resetFilter',
        },
        url: myAjax.ajaxurl,
        dataType: "json",
        success: function (response) {
            if (response && response.data) {
                let elem = document.querySelector('.navigation__inner');
                elem.innerHTML = response.data;
            }
        }
    });
}

function view(e, type) {
    if (event) {
        event.preventDefault();
    }
    jQuery.ajax({
        method: 'POST',
        type: 'POST',
        data: {
            action: type,
        },
        url: myAjax.ajaxurl,
        dataType: "json",
        success: function (response) {
            if (response && response.data) {
                let elem = document.querySelector('.navigation__inner');
                elem.innerHTML = response.data;
            }
        }
    });
}

```

```

    }
  });
}

document.addEventListener('DOMContentLoaded', function () {
  const viewSwitch = document.getElementById('viewSwitch');

  function getSwitchState() {
    return localStorage.getItem('viewSwitchState') === 'true';
  }

  viewSwitch.checked = getSwitchState();

  viewSwitch.addEventListener('change', function () {
    const viewType = viewSwitch.checked ? 'card' : 'table';
    viewSwitch.checked ? viewSwitch.classList.toggle('card') :
viewSwitch.classList.remove('card');
    view(null, viewType);
    localStorage.setItem('viewSwitchState', viewSwitch.checked);
  });
});

jQuery(document).ready(function(){
  jQuery.ajax({
    method: 'POST',
    type: 'POST',
    data: {
      action: 'resetFilter',
    },
    url: myAjax.ajaxurl,
    dataType: "json",
    success: function (response) {
      if (response && response.data) {
        let elem = document.querySelector('.navigation__inner');
        elem.innerHTML = response.data;
      }
    }
  });
});

$(document).ready(function() {
  var max_fields = 3;
  var wrapper = $(".row-new-fields__inner");
  var add_button = $(".add_form_field");
  var x = 0;

  $(add_button).click(function(e) {
    e.preventDefault();
    if (x < max_fields) {
      x++;
      $(wrapper).append('<div class="form-group row"><i class="fa-solid fa-delete-left del" style="color: #ffffff;"></i><input placeholder="Додаткова

```



```

інформація" id="row['+x+']" type="text" class="form-control" name="row['+x+']"
value="" autocomplete="row['+x+']" autofocus required></div>');
    } else {
        alert('Максимальна кількість додаткових полів - 3')
    }
});

$(wrapper).on("click", ".del", function(e) {
    e.preventDefault();
    $(this).parent('div').remove();
    x--;
});

function add(event) {
    event.preventDefault();
    var formData = new FormData($('#add')[0]);
    formData.append('action', 'add');

    $.ajax({
        method: 'POST',
        url: '/wp-admin/admin-ajax.php',
        data: formData,
        processData: false,
        contentType: false,
        success: function(response) {
            var responseObject = JSON.parse(response);

            if (responseObject.result === 1) {
                btnReset(event);
                $('#add')[0].reset();
                alert('Файл успішно завантажено')
            } else {
                alert('Не вдалося виконати запит: ' + responseObject.message);
            }
        },
        error: function(error) {
        }
    });
}

```