

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Створення вебплатформи на мові PHP для перегляду і
завантаження модифікацій в комп'ютерній грі Minecraft»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти бакалавр
групи 126ІСТ_бд_2022
Галюк Максим Дмитрович
Керівник: Протас Надія Михайлівна
Рецензент: Яхін Сергій Валерійович

Полтава – 2026 року

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

Освітня програма Інформаційні управляючі системи
Спеціальність 126 Інформаційні системи та технології
Рівень вищої освіти перший (бакалаврський)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Юрій УТКІН

«10» червня 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Галука Максиміо Дмитровича

1. Тема кваліфікаційної роботи:
«Створення вебплатформи на мові PHP для перегляду і завантаження модифікацій в компютерній грі Minecraft»,
Керівник роботи: к. с.-г. н., доцент, доцент кафедри інформаційних систем та технологій Протас Надія Михайлівна
Затверджено наказом закладу вищої освіти від «10» квітня 2026 року № 383-ст
2. Строк подання здобувачем вищої освіти роботи «08» червня 2026 року.
3. Вихідні дані до роботи: статистичні, науково-популярні джерела мережі інтернет, специфікації з веброзробки, статистичні дані аналітичних компаній <https://respawn.outlookindia.com/gaming/>, Anystat <https://www.pcgamer.com/>, <https://anystat.ai/> дані та програмне забезпечення офіційних сайтів <https://code.visualstudio.com/>, <https://www.minecraft.net/>, <https://www.work.ua/>
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Розділ 1 Теоретичні основи розробки вебплатформ для поширення ігрового контенту
Розділ 2. Проектування вебплатформи для перегляду та завантаження модифікацій Minecraft
Розділ 3. Програмно-технічна реалізація та тестування вебплатформи
5. Перелік графічного матеріалу: схеми, рисунки, діаграми за темою та об'єктом дослідження

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання отримав
Оцінювання економічної ефективності результатів дослідження	Дивнич О. Д., к. е. н., доцент, доцент кафедри економіки та публічного управління	01.04.2026 р.	29.05.2026 р.

7. Дата видачі завдання «10» червня 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Вибір і затвердження теми роботи	02.06.2025 р. – 04.06.2025	
2	Складання і затвердження розгорнутого плану та завдання на кваліфікаційну роботу	05.06.2025 р. – 10.06.2025 р.	
3	Опрацювання джерел інформації	11.06.2025 р. – 15.06.2025 р.	
4	Збір, вивчення і обробка інформації, необхідної для виконання роботи	16.06.2025 р. – 20.06.2025 р.	
5	Виконання теоретико-методологічного розділу роботи	08.09.2025 р. – 01.11.2025 р.	
6	Виконання дослідницько-аналітичного розділу роботи	19.01.2026 р. – 14.02.2026 р.	
7	Виконання проєктно-рекомендаційного розділу роботи	16.02.2026 р. – 31.03.2026 р.	
8	Оцінювання економічної ефективності результатів дослідження	01.04.2026 р. – 29.05.2026 р.	
9	Оформлення тексту роботи	01.06.2026 р. – 06.06.2026р.	
10	Попередній захист роботи на кафедрі	08.06.2026 р.	
11	Доопрацювання роботи з урахуванням зауважень і пропозицій	09.06.2026 р. – 10.06.2026 р.	
12	Нормоконтроль	11.06.2026 р. – 15.06.2026 р.	
13	Захист кваліфікаційної роботи	16.06.2026 р. - 18.06.2026 р.	

Здобувач вищої освіти

Максімо ГАЛЮК

Керівник роботи

Надія ПРОТАС

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБПЛАТФОРМ ДЛЯ ПОШИРЕННЯ ІГРОВОГО КОНТЕНТУ	9
1.1 Особливості комп'ютерної гри Minecraft та роль модифікацій у розширенні її функціональних можливостей	9
1.2 Аналіз сучасних вебресурсів для перегляду та завантаження модифікацій	13
1.3 Огляд технологій і засобів розробки вебплатформ на основі PHP	17
РОЗДІЛ 2 ПРОЄКТУВАННЯ ВЕБПЛАТФОРМИ ДЛЯ ПЕРЕГЛЯДУ ТА ЗАВАНТАЖЕННЯ МОДИФІКАЦІЙ MINECRAFT	21
2.1 Формування функціональних вимог та проєктування структури вебплатформи	21
2.2 Проєктування структури та навігаційної моделі вебсайту	26
2.3 Проєктування користувацького інтерфейсу та навігації вебплатформи	30
РОЗДІЛ 3 ПРОГРАМНО-ТЕХНІЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБПЛАТФОРМИ	37
3.1 Розробка клієнтської частини засобами HTML, CSS і PHP та основних функціональних модулів	37
3.2 Реалізація серверної частини вебплатформи засобами PHP	43
3.3 Економічне оцінювання варіантів розробки вебсайту	46
ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТКИ	52

ВСТУП

Актуальність теми кваліфікаційної роботи розкривається через сприйняття того факту, що комп'ютерні ігри стали важливою складовою сучасної цифрової культури та однією з найбільш динамічних галузей індустрії інформаційних технологій. Розробка, підтримка та супровід ігрових продуктів потребують використання широкого спектра сучасних технологій, зокрема веброзробки, програмування, роботи з базами даних, мережесервісів і систем керування контентом. Тому розуміння принципів функціонування комп'ютерних ігор та набуття практичних навичок створення програмних рішень для ігрової індустрії є важливим напрямом підготовки фахівців у сфері інформаційних технологій. Такі проєкти дозволяють поєднати теоретичні знання з практичними навичками проєктування та розробки сучасних інформаційних систем.

Однією з найпопулярніших комп'ютерних ігор у світі є Minecraft, яка завдяки відкритості, творчій свободі та широким можливостям модифікації об'єднує багатомільйонну спільноту користувачів різного віку. Важливою особливістю гри є підтримка модифікацій (модів), які дозволяють розширювати ігровий процес, додавати нові механіки, об'єкти та функціональні можливості.

Існуючі ресурси для пошуку та завантаження модифікацій часто мають складний інтерфейс, містять надлишкову рекламу або не забезпечують зручного механізму керування контентом. Крім того, значна кількість користувачів потребує швидкого доступу до актуальних версій модифікацій, їх описів та інструкцій зі встановлення. У зв'язку з цим виникає потреба у створенні спеціалізованих вебплатформ, які забезпечують зручний перегляд, пошук, категоризацію та завантаження модифікацій, зокрема, для гри Minecraft. Розробка такої вебплатформи із використанням мови програмування PHP є актуальним завданням, оскільки дозволяє поєднати сучасні вебтехнології з потребами ігрової спільноти, забезпечити ефективне зберігання

та обробку даних, а також створити доступний інструмент для поширення користувацького контенту. Реалізація подібного проєкту сприяє набуттю практичного досвіду розробки веборієнтованих інформаційних систем, що є важливим для майбутньої професійної діяльності фахівців ІТ-галузі.

Метою кваліфікаційної роботи є проєктування та розробка вебплатформи на мові програмування PHP для перегляду, пошуку та завантаження модифікацій комп'ютерної гри Minecraft із забезпеченням зручного інтерфейсу користувача та ефективного керування контентом.

Завдання кваліфікаційної роботи:

- провести аналіз предметної області та особливостей поширення модифікацій для гри Minecraft;
- обґрунтувати вибір мови програмування PHP та супутніх технологій;
- розробити структуру та архітектуру вебплатформи;
- спроектувати базу даних для зберігання інформації про модифікації та користувачів;
- реалізувати функціонал перегляду, пошуку та завантаження модифікацій;
- розробити адміністративні засоби керування контентом;
- створити адаптивний та зручний користувацький інтерфейс;
- провести тестування вебплатформи та оцінити результати її функціонування;
- оцінити практичну ефективність розробленого рішення.

Об'єктом дослідження кваліфікаційної роботи процес розробки та функціонування веборієнтованих інформаційних систем для поширення цифрового контенту.

Предметом дослідження є методи, технології та програмні засоби створення вебплатформи на мові PHP для перегляду, пошуку та завантаження модифікацій гри Minecraft.

Методи дослідження:

- аналіз та узагальнення науково-технічної літератури і вебресурсів;

- системний аналіз предметної області;
- методи проєктування баз даних;
- методи об'єктно-орієнтованого та структурного програмування;
- методи вебпроєктування та UX/UI-дизайну;
- методи тестування програмного забезпечення для перевірки працездатності вебплатформи.

Інформаційну базу кваліфікаційної роботи становлять наукові праці вітчизняних і зарубіжних авторів з вебпрограмування та проєктування інформаційних систем, навчальна і довідкова література з мов програмування PHP, HTML, CSS та JavaScript, матеріали офіційної технічної документації використовуваних програмних засобів і технологій, а також інформаційні ресурси мережі інтернет, присвячені розробці вебзастосунків та функціонуванню модифікацій для комп'ютерної гри Minecraft, матеріали тематичних спільнот розробників.

Практична значущість роботи полягає у створенні працездатної вебплатформи для перегляду та завантаження модифікацій гри Minecraft із використанням мови програмування PHP та сучасних вебтехнологій, що забезпечують зручну взаємодію користувачів із контентом. Розроблений програмний продукт може бути використаний як основа для створення тематичних ігрових порталів, каталогів модифікацій та інших вебресурсів, орієнтованих на спільноти користувачів комп'ютерних ігор. Отримані результати можуть бути застосовані у подальших проєктах з розробки вебінформаційних систем та вдосконалення сервісів керування цифровим контентом.

Структура і обсяг кваліфікаційної роботи: пояснювальна записка має 50 сторінок і складається із вступу, трьох розділів; висновків, списку використаних джерел, додатків. Робота містить 22 рисунки, 5 таблиць, посилання на 32 інформаційні джерела.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБПЛАТФОРМ ДЛЯ ПОШИРЕННЯ ІГРОВОГО КОНТЕНТУ

1.1 Особливості комп'ютерної гри Minecraft та роль модифікацій у розширенні її функціональних можливостей

Сучасна індустрія комп'ютерних ігор є однією з найдинамічніших сфер розвитку інформаційних технологій (ІТ). За оцінками аналітичних компаній, обсяг світового ринку відеоігор у 2024 р. перевищив 180 млрд доларів США, а у 2025 р. наблизився до 190–200 млрд доларів, демонструючи стабільне зростання навіть в умовах загальної нестабільності світової економіки [1]. Значна частина доходів галузі формується за рахунок цифрової дистрибуції контенту, онлайн-сервісів та внутрішньоігрових покупок, що свідчить про високий рівень цифровізації ігрової індустрії [2].



Рисунок 1.1 – Динаміка обсягу світового ринку відеоігор за 2017–2026 рр. (на основі аналітичних даних [1])

Ринок програмного забезпечення (ПЗ) для комп'ютерних ігор об'єднує провідні технологічні корпорації світу. Серед найбільших компаній-розробників і видавців за обсягами ігрових доходів у 2024–2025 рр. лідируючі позиції займають Tencent, Sony Interactive Entertainment та Microsoft Gaming (рис. 1.2) за версією статистичних даних ANYSTAT [3]. Річні доходи їхніх ігрових підрозділів оцінюються у десятки мільярдів доларів США, що підтверджує значний економічний потенціал галузі та її вагоме місце у світовій цифровій економіці.

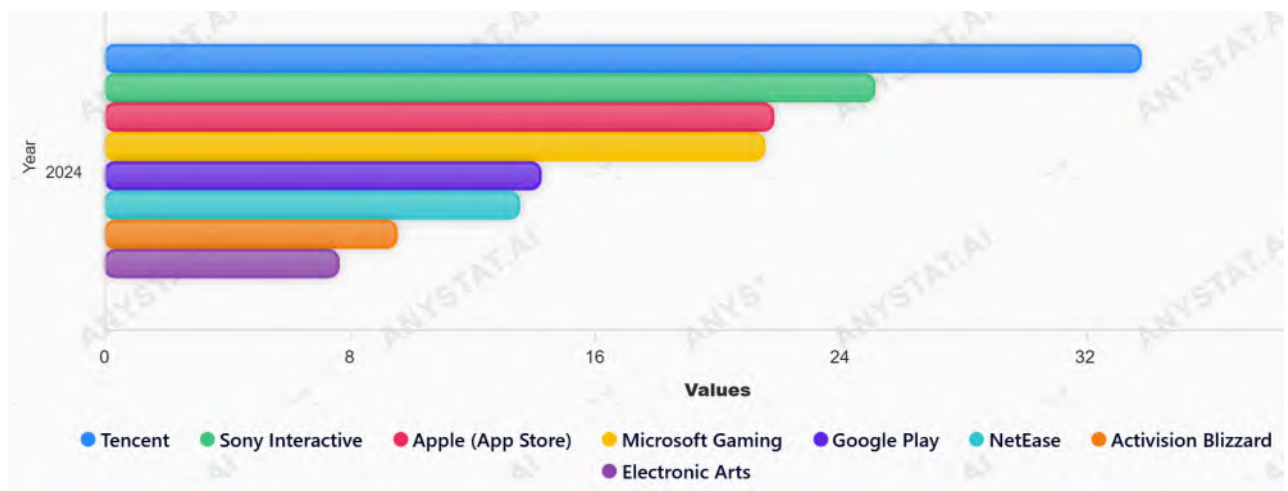


Рисунок 1.2 - **Top Video Game Companies by Revenue 2024 [3]**

Індустрія комп'ютерних ігор поєднує програмування, комп'ютерну графіку, мережеві технології, проектування користувацьких інтерфейсів, технології баз даних, штучний інтелект та управління цифровим контентом. Саме тому розробка ПЗ для ігрових платформ розглядається як одна з перспективних сфер професійної діяльності фахівців галузі ІТ. Вивчення принципів функціонування ігрових систем дозволяє майбутнім розробникам отримати практичний досвід створення складних програмних продуктів, що поєднують різноманітні програмні компоненти та сучасні підходи до проектування інформаційних систем (ІС).

Згідно проведених досліджень, розробникам ігор стає дедалі складніше керувати витратами на розробку гри, водночас задовольняючи високі

очікування геймерів. Один зі способів збалансувати зростаючі очікування геймерів та стрес від розробки – це створити активну спільноту моддерів навколо гри. Особливе місце серед сучасних комп'ютерних ігор займають так звані «пісочниці» (sandbox games), які надають користувачам значну свободу дій та можливість самостійно формувати ігровий процес. Існує кілька прикладів ігор з надзвичайно активною та успішною спільнотою моддерів, однією з найпомітніших з яких є гра Minecraft.

Станом на 2025 р. гра продовжує залишатися найпродаванішою відеогрою в історії, а кількість проданих копій перевищила 350 млн, що є безпрецедентним показником для індустрії ігор [5].

Minecraft була створена шведським програмістом Маркусом Перссоном (Notch) у 2009 р. У подальшому розробкою гри займалася компанія Mojang Studios, яка у 2014 р. стала частиною корпорації Microsoft [4]. Основною особливістю Minecraft є відкритий віртуальний світ, що складається з блоків різних типів. Гравці можуть досліджувати навколишнє середовище, добувати ресурси, створювати предмети, будувати споруди та взаємодіяти з іншими користувачами.

Популярність Minecraft пояснюється поєднанням простоти ігрової механіки та практично необмежених можливостей для творчості. На відміну від багатьох інших комп'ютерних ігор, де користувачі діють за заздалегідь визначеним сценарієм, Minecraft дозволяє самостійно визначати цілі та способи їх досягнення. Завдяки цьому гра використовується не лише для розваг, але й у навчальних цілях, зокрема для розвитку просторового мислення, логіки та навичок командної роботи [5-6].

Важливою перевагою Minecraft є її висока адаптивність до потреб користувачів. Такий підхід сприяв формуванню великої міжнародної спільноти розробників і користувачів, які створюють власні доповнення та програмні модифікації [7].

Модифікацією (англ. modification, mod) називають програмне доповнення, яке змінює або розширює функціональність комп'ютерної гри без

внесення змін до її базового вихідного коду. У Minecraft модифікації можуть додавати нові блоки, предмети, механізми, мобів, біоми, технологічні системи та навіть повністю нові сценарії гри.

Однією з причин популярності Minecraft серед розробників є використання мови програмування Java у версії Minecraft Java Edition. Це дозволяє створювати модифікації за допомогою відносно доступних програмних інструментів і забезпечує широкі можливості інтеграції додаткового функціоналу. Для спрощення процесу розробки модів використовуються спеціальні програмні платформи, серед яких найбільш поширеними є Forge, Fabric та NeoForge [8]. Модифікації можна класифікувати за їх функціональним призначенням (табл. 1.1).

Таблиця 1.1 – Класифікація модифікацій Minecraft [9]

Клас модифікацій	Призначення модифікації
Перша група	модифікації, що додають новий контент: блоки, інструменти, ресурси та персонажів
Друга група	технічні модифікації, які змінюють механіку гри або додають складні виробничі процеси.
Третя група	графічні модифікації, спрямовані на покращення візуального оформлення та продуктивності гри.
Додаткові	модифікації, що забезпечують підтримку багатокористувацьких серверів та автоматизацію ігрових процесів

Завдяки модифікаціям користувачі отримують можливість налаштовувати ігровий процес відповідно до власних потреб та інтересів. Наприклад, популярні модифікації можуть додавати елементи промислового виробництва, енергетичні системи, нові виміри, магічні механіки або транспортні засоби. У результаті Minecraft перетворюється з базової гри на універсальну платформу для реалізації різноманітних ігрових концепцій [9].

Значне поширення модифікацій зумовило появу спеціалізованих вебресурсів для їх зберігання та поширення. Такі платформи забезпечують каталогізацію контенту, пошук модифікацій за категоріями, підтримку різних версій гри, систему оцінювання та коментування. Вони виступають важливим

елементом взаємодії між розробниками модифікацій та кінцевими користувачами [10].

Автори роботи [8] проаналізували взаємозв'язок між 33 характеристиками за 5 вимірами характеристик модів та популярністю модів (тобто категорія мода, документація мода, екологічний контекст мода, винагорода за мод та внесок спільноти за мод), щоб зрозуміти характеристики популярних модів Minecraft. Спочатку перевірили, чи досліджувані виміри мають значну пояснювальну силу у визначенні популярності досліджуваних модів. Потім оцінили внесок кожної з 33 характеристик за 5 вимірами. У результаті прийшли до висновку, що популярні моди, як правило, мають високоякісний опис та заохочують внесок спільноти.

Сучасні вебплатформи для поширення модифікацій повинні забезпечувати зручний доступ до інформації, швидкий пошук необхідного контенту, безпечне завантаження файлів та ефективне керування даними. Саме тому розробка спеціалізованих вебресурсів для перегляду та завантаження модифікацій Minecraft є актуальним напрямом застосування сучасних вебтехнологій та засобів програмування. Реалізація подібних систем дозволяє поєднати можливості веброзробки з потребами великої ігрової спільноти, забезпечуючи ефективне поширення та використання користувацького контенту. Таким чином, обґрунтовано значення ігрової індустрії для програмної інженерії; роль Minecraft як технологічної платформи; значення модифікацій та спільнот моддерів; освітню та практичну цінність розробки програмних рішень для ігрового середовища.

1.2 Аналіз сучасних вебресурсів для перегляду та завантаження модифікацій

Відеоігри – це передові програмні продукти, і водночас вони є складними витворами творчості та мистецтва. Це злиття дисциплін, потреба в тому, щоб

продукт був цікавим, і невинне зростання індустрії відеоігор створили значні відмінності між традиційною програмною інженерією (SE) та ігровою програмною інженерією (GSE) [11]. Ці умови вимагають постійної та швидкої комунікації між розробниками, гнучкого дизайну та зручної реалізації. Більше того, відеоігри, як правило, є продуктами з набагато обмеженішим життєвим циклом, ніж інші програмні продукти. Ігрова індустрія відома своїми проблемами. Вони варіюються від технічних, наприклад, 80% ігор у Steam потребують критичних оновлень, до проблем управління, наприклад, стислі терміни та нереалістичні масштаби [12]. Ігри з відкритим кодом трапляються рідко, а основні інструменти, що використовуються розробниками ігор, тобто ігрові двигуни, є власницькими (наприклад, Unity та Unreal). Наприклад, GitHub має мало популярних проєктів ігрових рушіїв, тоді як його аналоги, програмні фреймворки, численні.

Стрімкий розвиток спільноти Minecraft та популярність користувацьких модифікацій сприяли появі спеціалізованих вебресурсів для їх розміщення, поширення та підтримки. Такі платформи забезпечують взаємодію між авторами модифікацій та користувачами, надаючи інструменти для публікації, пошуку, оцінювання та завантаження контенту. Сучасні ресурси також виконують функції каталогізації, контролю сумісності та підтримки різних версій гри.

Одним із найбільших ресурсів для поширення модифікацій є CurseForge. Платформа функціонує з 2006 року та містить найбільшу колекцію модів, модпаків, ресурс-паків та інших доповнень для Minecraft [13]. Користувачам доступний багаторівневий пошук за категоріями, версіями гри та типами завантажуваного контенту. Серед переваг CurseForge можна виділити велику базу проєктів, підтримку авторів контенту та інтеграцію з популярними лаунчерами гри (рис. 1.3). Разом із тим користувачі нерідко відзначають перевантаженість інтерфейсу рекламою та складність навігації для новачків. Крім того, значна кількість функцій орієнтована на досвідчених користувачів, що може ускладнювати процес пошуку необхідних модифікацій.

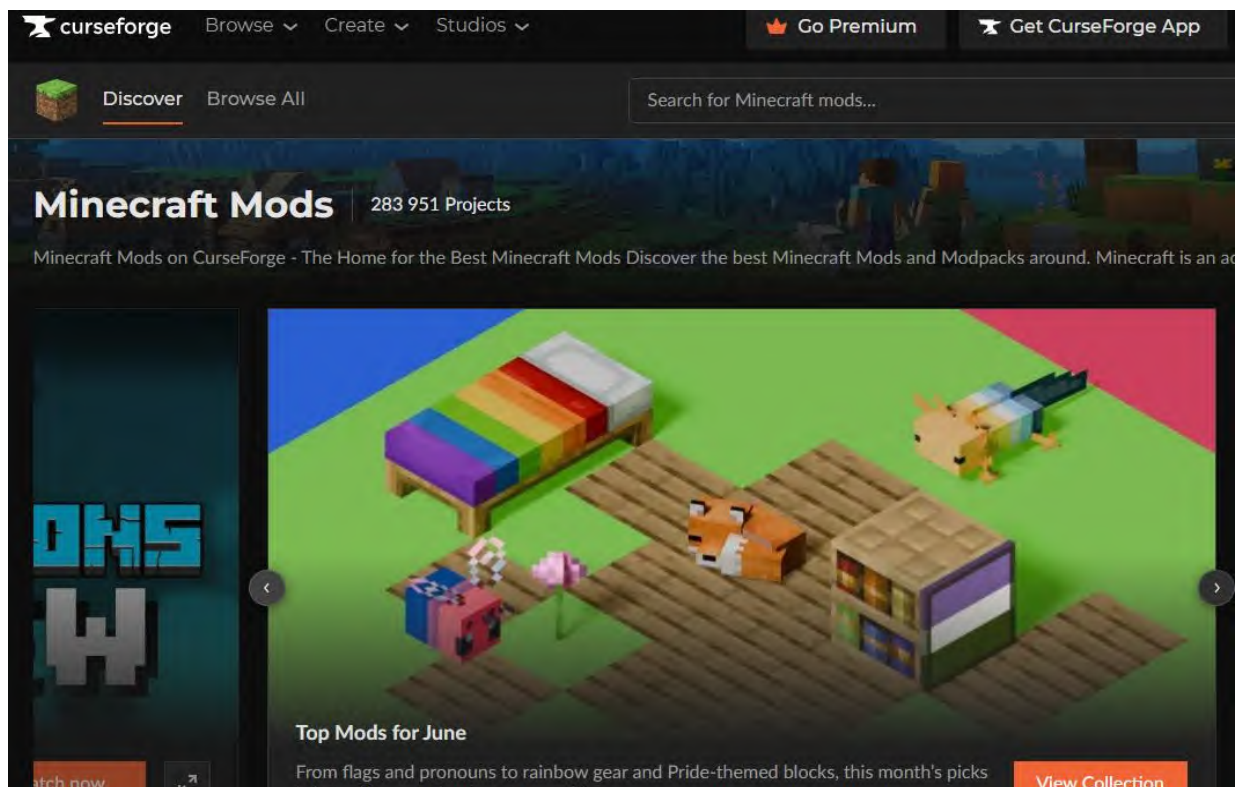


Рисунок 1.3 – Огляд проєктів на платформі CurseForge

Іншим популярним ресурсом є Modrinth [14], який з'явився значно пізніше, проте швидко здобув популярність серед авторів та користувачів модифікацій. Основною особливістю платформи є сучасний інтерфейс, відкритість програмного коду та зручна система пошуку (рис. 1.4).

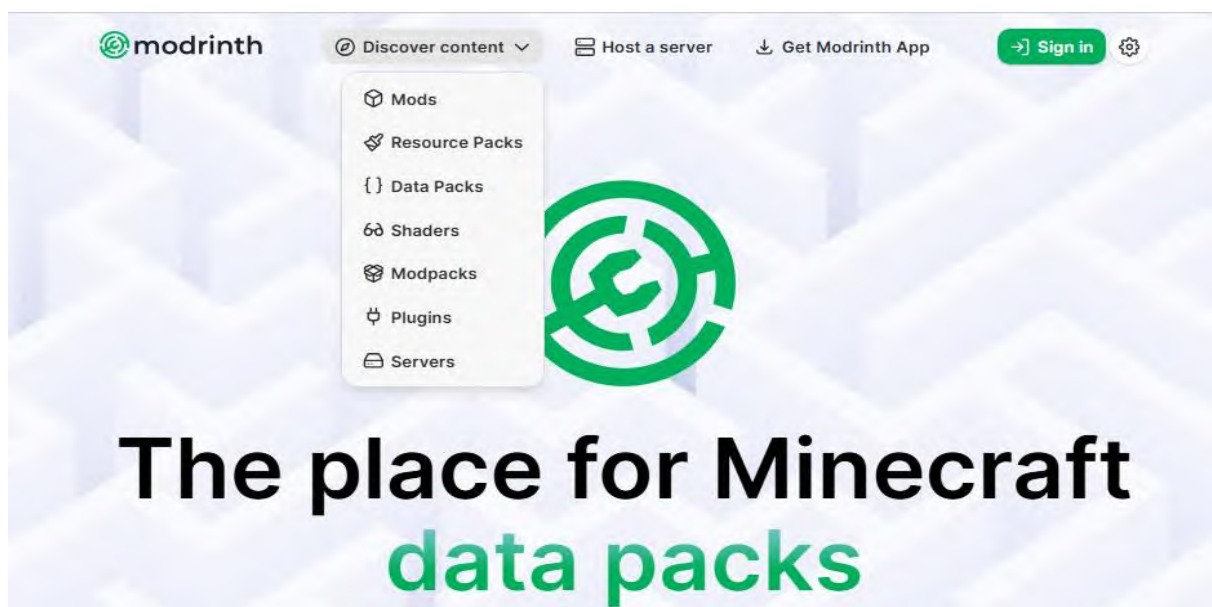


Рисунок 1.4 – Інтерфейс платформи Modrinth

Ресурс підтримує не лише модифікації, а й модпаки, шейдери, плагіни та пакети ресурсів. Перевагами Modrinth є висока швидкодія, детальна система фільтрації та орієнтація на потреби спільноти розробників. Значна увага приділяється сумісності модифікацій із різними версіями Minecraft та середовищами розробки.

Суттєве місце серед ресурсів спільноти займає Planet Minecraft. На відміну від CurseForge та Modrinth, цей ресурс орієнтований не лише на модифікації, а й на широкий спектр користувацького контенту [15]. Тут розміщуються карти, текстури, скіни, готові будівлі та інші творчі проєкти. Основною перевагою платформи є розвинена соціальна складова: користувачі можуть створювати власні сторінки, публікувати новини, обмінюватися досвідом та оцінювати роботи інших учасників спільноти. Водночас функціонал пошуку модифікацій є менш спеціалізованим порівняно з профільними ресурсами.

Дослідження показують, що найбільшу популярність серед платформ поширення модифікацій мають саме CurseForge та Planet Minecraft, які містять тисячі активних проєктів і виступають основними центрами розповсюдження користувацького контенту для Minecraft. У наукових дослідженнях CurseForge часто розглядається як найбільша платформа для аналізу модифікацій завдяки значному обсягу доступних проєктів та відкритих статистичних даних.

Важливим аспектом сучасних платформ є забезпечення безпеки завантажуваного контенту. Оскільки модифікації фактично є виконуваними програмними компонентами, користувачі повинні мати можливість завантажувати файли лише з перевірених джерел. Останні дослідження кібербезпеки демонструють, що поширення шкідливого програмного забезпечення через неофіційні сайти з модифікаціями залишається актуальною проблемою. Саме тому великі платформи впроваджують механізми перевірки файлів та модерації опублікованого контенту.

Порівняння функціональних можливостей популярних ресурсів дозволяє виділити основні критерії ефективної вебплатформи для поширення

модифікацій: наявність структурованого каталогу, зручної системи пошуку та фільтрації, підтримки різних версій гри, детальних сторінок опису модифікацій, можливості оцінювання контенту та безпечного завантаження файлів. Крім того, важливу роль відіграють простота інтерфейсу та швидкість доступу до інформації.

Аналіз існуючих рішень показав, що жодна з розглянутих платформ не є універсальною. Одні ресурси пропонують широкий вибір контенту, але мають складний інтерфейс, інші забезпечують зручність використання, проте поступаються за кількістю доступних модифікацій. Це створює передумови для розробки власної вебплатформи, яка поєднуватиме переваги існуючих рішень та забезпечуватиме зручний механізм перегляду, пошуку і завантаження модифікацій для гри Minecraft.

1.3 Огляд технологій і засобів розробки вебплатформ на основі РНР

Розвиток ІТ сприяв широкому поширенню веборієнтованих ІС, які використовуються для надання доступу до різноманітних цифрових ресурсів та сервісів. Сучасні вебплатформи забезпечують взаємодію користувачів через мережу інтернет, підтримують обробку великих обсягів даних та дозволяють організовувати централізоване керування інформаційним контентом. Для реалізації подібних систем використовуються різноманітні програмні засоби, серед яких важливе місце займають серверні мови програмування та системи управління базами даних.

Однією з найпоширеніших мов програмування для створення вебзастосунків є РНР (Hypertext Preprocessor). Мова була розроблена у 1995 р.і Расмусом Лердорфом і протягом багатьох років залишається одним із найбільш популярних інструментів веброботи [16]. За даними сервісу W3Techs, РНР використовується більш ніж на 71 % вебсайтів, серверна частина яких реалізована за допомогою відомих технологій програмування.

Така популярність пояснюється відкритістю мови, широкими можливостями інтеграції та значною кількістю програмних бібліотек і фреймворків [17]. На рис. 1.5 представлена діаграма, що показує відсотки вебсайтів, які використовують різні мови серверного програмування.

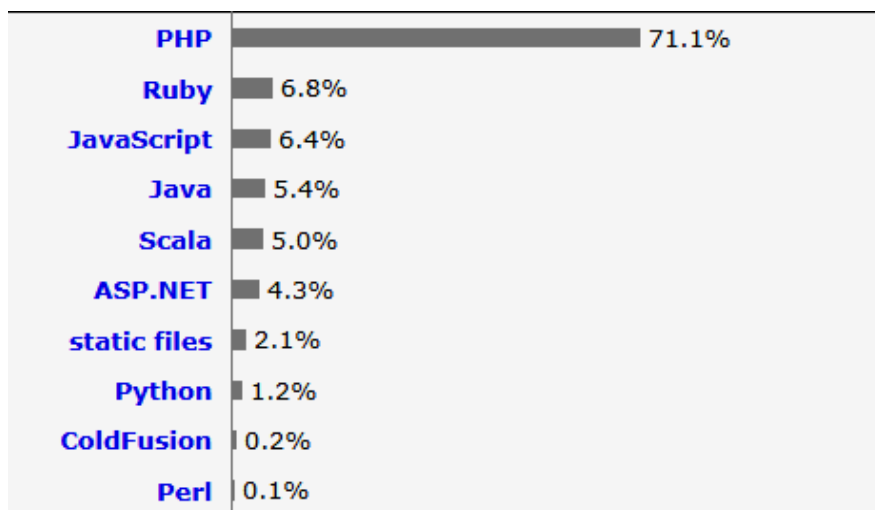


Рисунок 1.5 – Розподіл вебсайтів за частотою використання мов серверного програмування

Я бачимо, PHP використовується 71,1% усіх вебсайтів, чий мови серверного програмування нам відомі.

PHP є серверною мовою програмування, яка виконується безпосередньо на вебсервері. На відміну від клієнтських технологій, що працюють у браузері користувача, PHP забезпечує обробку запитів, взаємодію з базами даних, автентифікацію користувачів, керування файлами та формування вебсторінок. Після виконання програмного коду користувачу передається готовий HTML-документ, який відображається у браузері [18].

До основних переваг PHP належать простота вивчення, підтримка об'єктно-орієнтованого програмування, кросплатформеність та сумісність із більшістю сучасних вебсерверів. Мова підтримується операційними системами Windows, Linux та macOS, що спрощує процес розробки та розгортання вебзастосунків. Важливою перевагою є також наявність великої спільноти розробників та значного обсягу технічної документації [17].

Для створення повноцінних вебплатформ PHP використовується разом з іншими вебтехнологіями. Основою будь-якого вебінтерфейсу є мова розмітки HTML (HyperText Markup Language), яка визначає структуру вебсторінки та її інформаційне наповнення. HTML дозволяє створювати текстові блоки, таблиці, форми введення даних, меню навігації та інші елементи інтерфейсу [19].

Для оформлення зовнішнього вигляду вебсторінок застосовується каскадна таблиця стилів CSS (Cascading Style Sheets). Використання CSS забезпечує розділення структури документа та його візуального представлення. За допомогою стилів реалізуються кольорові схеми, адаптивний дизайн, анімація елементів та коректне відображення інтерфейсу на різних типах пристроїв [20].

Динамічна взаємодія користувача з вебсторінками реалізується за допомогою мови програмування JavaScript. Вона дозволяє виконувати перевірку даних без перезавантаження сторінки, створювати інтерактивні елементи інтерфейсу, працювати з асинхронними запитами та покращувати загальний користувацький досвід. У сучасній веброзробці JavaScript є фактичним стандартом для реалізації клієнтської логіки вебзастосунків [21].

Важливим компонентом будь-якої вебплатформи є система управління базами даних. Для зберігання інформації про користувачів, модифікації, категорії та завантаження найчастіше використовується СУБД MySQL [22]. Вона належить до реляційних систем управління базами даних та забезпечує надійне зберігання, пошук і обробку великих обсягів інформації. Перевагами MySQL є висока продуктивність, безкоштовне використання та повна сумісність із PHP.

Для локальної розробки та тестування вебзастосунків широко використовуються інтегровані програмні середовища, що містять вебсервер Apache, інтерпретатор PHP та систему керування базами даних MySQL. Найпоширенішими рішеннями є XAMPP, Open Server та WAMP. Такі програмні комплекси дозволяють створити локальне середовище розробки без необхідності використання віддаленого сервера [23].

З метою підвищення продуктивності розробки часто застосовуються спеціалізовані середовища програмування. Серед найпопулярніших редакторів коду можна виділити Visual Studio Code, PhpStorm та Sublime Text. Вони забезпечують підсвічування синтаксису, автоматичне доповнення коду, інструменти налагодження та інтеграцію із системами контролю версій [24].

Для керування змінами програмного коду широко використовується система контролю версій Git. Вона дозволяє відстежувати історію змін, організовувати командну роботу над проєктом та забезпечувати надійне зберігання вихідного коду. Хмарні сервіси GitHub і GitLab надають додаткові засоби спільної розробки та резервного копіювання проєктів [25].

Під час створення вебплатформи для перегляду та завантаження модифікацій Minecraft доцільним є використання комплексу технологій PHP, HTML, CSS, JavaScript та MySQL. Таке поєднання забезпечує реалізацію серверної та клієнтської частин системи, підтримку взаємодії з базою даних, зручний користувацький інтерфейс та можливість подальшого розширення функціоналу. Крім того, використання відкритих і широко поширених технологій дозволяє мінімізувати витрати на розробку та спрощує подальший супровід програмного продукту.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ВЕБПЛАТФОРМИ ДЛЯ ПЕРЕГЛЯДУ ТА ЗАВАНТАЖЕННЯ МОДИФІКАЦІЙ MINECRAFT

2.1 Формування функціональних вимог та проєктування структури вебплатформи

Основною метою кваліфікаційної роботи є створення вебресурсу, який забезпечує зручний доступ до каталогу модифікацій Minecraft, дозволяє користувачам переглядати доступний контент, завантажувати файли модифікацій та розмішувати власні розробки. При цьому важливо продемонструвати вміння обрати та використовувати низку програмних продуктів та сервісів, поєднувати окремі частини проєкту в одне цілі і забезпечувати заданий функціонал. Вебресурс отримав назву «WorldMods».

Процес інженерного проєктування складних програмних систем та веб-орієнтованих ресурсів вимагає чіткої формалізації концептуальних вимог, побудови раціональної архітектури та обґрунтування вибору технологічного стеку [26]. Створення «WorldModse комплексною розробкою, що інтегрує клієнт-серверні технології, динамічні бази даних та модулі безпеки даних.

На етапі концептуального планування було сформовано цільове призначення системи та визначено коло її кінцевих користувачів. Головна місія вебресурсу полягає у забезпеченні централізованої платформи, де взаємодіють дві основні ролі користувачів:

1. Споживачі контенту (гравці): користувачі, які потребують швидкого доступу до каталогу модифікацій, можливості миттєвого пошуку контенту, ознайомлення з його характеристиками (версія, автор, опис) та безпосереднього завантаження бінарних файлів на свій локальний пристрій.

2. Творці контенту (розробники модів): автентифіковані користувачі, які мають можливість інтегрувати власні програмні продукти (модифікації) у загальний каталог за допомогою спеціальних веб-форм, завантажуючи архіви

файлів на сервер. Для досягнення поставлених завдань було визначено набір функціональних вимог, які повинна реалізовувати система. До основних функціональних вимог вебплатформи належать:

- відображення каталогу доступних модифікацій;
- перегляд інформації про модифікації;
- завантаження файлів модифікацій користувачами;
- реєстрація нових користувачів;
- авторизація користувачів;
- керування користувацькими сесіями;
- відображення персональної інформації в особистому кабінеті;
- додавання нових модифікацій до каталогу;
- зберігання інформації про модифікації та користувачів у базі даних.

До нефункціональних вимог належать забезпечення зручності інтрефейсу, швидкодії при обробці запитів, надійності зберігання даних, сумісності з більшістю веббраузерів та можливості подальшого розширення функціоналу [27].

Відповідно до визначених вимог було обрано класичну клієнт-серверну архітектуру вебзастосунку. У даній моделі користувач взаємодіє із системою через веббраузер, який виступає клієнтом. Обробка запитів виконується на сервері за допомогою мови програмування PHP. Зберігання даних здійснюється в реляційній базі даних MySQL, а файли модифікацій розміщуються у файловому сховищі вебсервера [28].

Функціональна структура розробленої вебплатформи включає декілька взаємопов'язаних модулів. Структурна схема показана на рис. 2.1.

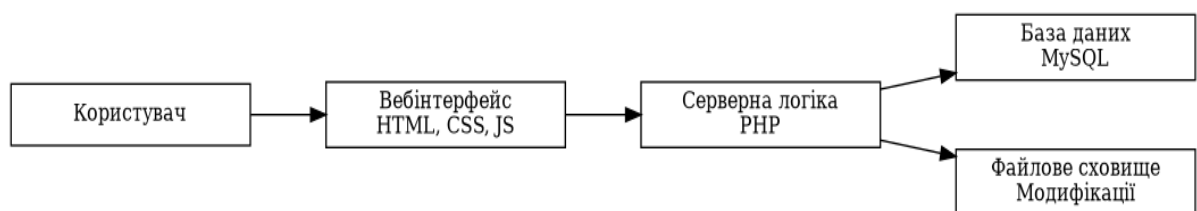


Рисунок 2.1 – Структурна схема вебплатформи

Для побудови інформаційної системи «WorldMods» було обрано клієнт-серверний архітектурний шаблон з тривірневою структурою: рівень представлення (клієнт), рівень бізнес-логіки (аплікаційний сервер) та рівень збереження даних (база даних та файлове сховище). У результаті розроблена сукупність файлів, які виконують певні функції в системі (табл. 2.1).

Таблиця 2.1 – Зміст і призначення модулів у структурі проекту

Модуль проєкту	Призначення
index.php	Головна сторінка каталогу модифікацій
register.php	Реєстрація користувачів
login.php	Авторизація
logout.php	Завершення сеансу
profile.php	Особистий кабінет
uploadmods.php	Завантаження нових модифікацій
db.php	Підключення до бази даних
style.css	Оформлення головної сторінки
profile.css	Стилі профілю
register.css	Стилі авторизації та реєстрації
mods.js	Пошук і взаємодія з каталогом

Взаємозв'язок архітектурних рівнів системи представлено у вигляді концептуальної схеми на рис. 2.2.

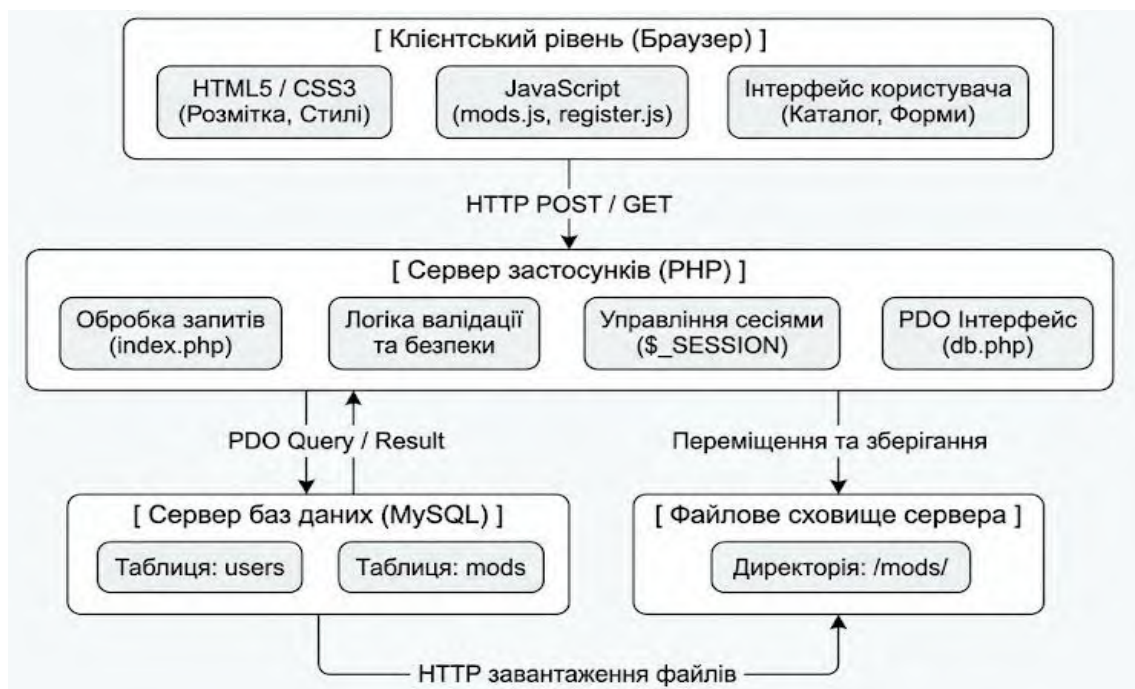


Рисунок 2.2 – Схема архітектури вебзастосунку

Згідно запланованої архітектури, взаємозв'язок архітектурних рівнів системи має такий вигляд:

1. Рівень представлення (Presentation Layer / Front-end): реалізований за допомогою мов розмітки HTML5, каскадних таблиць стилів CSS3 (style.css, register.css) та мови програмування JavaScript (index.js, mods.js, register.js). Завданням цього рівня є візуалізація карток модифікацій, обробка подій введення тексту у пошуковому полі в реальному часі за допомогою слухача подій `addEventListener("input", ...)` та первинна перевірка коректності введення даних користувачем (наприклад, валідація довжини пароля та збігу полів паролів при реєстрації на клієнтській стороні).

2. Рівень бізнес-логіки (Application Layer / Back-end): Побудований на базі інтерпретованої мови програмування PHP. Серверна частина відповідає за обробку HTTP-запитів (методи GET та POST), керування станом сесій користувачів, обробку масивів даних модифікацій та безпосередній контроль логіки завантаження файлів через глобальний масив `$_FILES`. На цьому рівні реалізовано критично важливі функції безпеки:

- валідація розширень завантажуваних файлів на стороні сервера (дозвіл лише для безпечних форматів архівів zip, rar, 7z та блокування потенційно небезпечних скриптів);

- генерація унікальних імен файлів шляхом додавання часових міток (`time() . '_' . basename(...)`) для запобігання конфліктів та перезапису даних на сервері;

- хешування паролів за допомогою криптографічно стійкого алгоритму `password_hash(..., PASSWORD_DEFAULT)` перед передачею даних на нижчий рівень.

3. Рівень збереження даних (Data Layer): Представлений системою керування реляційними базами даних MySQL, доступ до якої реалізовано через уніфікований інтерфейс PDO (PHP Data Objects). Цей рівень складається з двох компонентів:

- реляційна база даних, яка зберігає структуровану інформацію про користувачів (таблиця `users` із полями логіну, пошти та хешу пароля) та метадані модифікацій (таблиця `mods` із полями назви, опису, імені файлу та ідентифікатора автора). Взаємодія із БД захищена від SQL-ін'єкцій завдяки використанню підготовлених запитів (`$pdo->prepare()`) та параметризованого виконання (`$stmt->execute()`).

- локальне файлове сховище: виділена директорія сервера (`mods/`), де безпосередньо розміщуються завантажені бінарні файли модифікацій для їхньої подальшої дистрибуції користувачам за прямими HTTP-посиланнями.

Таким чином, запропонована та реалізована архітектурна концепція вебресурсу «WorldMods» забезпечує надійне зчеплення між ізольованими компонентами системи, мінімізує вразливості, пов'язані з безпекою даних, та гарантує виконання поставлених у кваліфікаційній роботі функціональних вимог щодо зручного перегляду, пошуку та завантаження модифікацій.

Центральним елементом системи є головна сторінка, яка забезпечує доступ до каталогу модифікацій. На цій сторінці користувач може переглядати доступний контент та отримувати інформацію про розміщені модифікації.

Для роботи із персональними даними реалізовано модуль реєстрації та авторизації користувачів. Реєстрація передбачає створення нового облікового запису шляхом введення імені користувача, електронної пошти та пароля. Після успішної авторизації користувач отримує доступ до додаткових функцій системи, зокрема до особистого кабінету та можливості завантаження нових модифікацій.

Особистий кабінет призначений для відображення інформації про зареєстрованого користувача та забезпечення доступу до індивідуальних можливостей системи. Наявність такого модуля підвищує рівень персоналізації вебплатформи та створює передумови для подальшого розвитку сервісу.

Окремим компонентом системи є модуль завантаження модифікацій. Його призначення полягає у додаванні нових файлів до каталогу вебплатформи. Під час завантаження користувач вводить основні відомості про

модифікацію та обирає файл для розміщення на сервері. Після перевірки дані зберігаються в базі даних, а сам файл розміщується у відповідній директорії файлового сховища.

Для забезпечення взаємодії між усіма компонентами використовується модуль доступу до бази даних. Він відповідає за встановлення з'єднання із сервером MySQL та виконання операцій читання і запису інформації. Завдяки централізованому доступу до даних забезпечується цілісність інформації та спрощується супровід програмного коду.

Структура вебплатформи була сформована з урахуванням принципів простоти використання та логічної організації навігації. Користувач може швидко перейти до необхідного розділу за допомогою головного меню, а всі основні функції доступні без виконання великої кількості додаткових дій. Такий підхід сприяє підвищенню зручності роботи із системою та покращує загальне сприйняття вебресурсу.

У результаті проектування було сформовано структуру вебплатформи, яка забезпечує реалізацію всіх необхідних функцій для перегляду, завантаження та керування модифікаціями гри Minecraft. Запропонована архітектура є достатньо простою для підтримки та водночас дозволяє розширювати функціональні можливості системи шляхом додавання нових модулів і сервісів у майбутньому.

2.2 Проектування структури та навігаційної моделі вебсайту

Ефективне функціонування вебресурсу «WorldMods» безпосередньо залежить від раціональної організації даних на рівні збереження інформації (Data Layer). На етапі проектування підсистеми даних вирішувалися завдання створення логічної структури для збереження облікових записів користувачів та метаданих файлів модифікацій, забезпечення цілісності інформації, оптимізації швидкості виконання запитів та захисту конфіденційних даних.

База даних (БД) забезпечує централізоване зберігання інформації про користувачів, модифікації та результати взаємодії між ними. Використання реляційної моделі даних дозволяє організувати інформацію у вигляді взаємопов'язаних таблиць, що забезпечує цілісність даних, зручність їх обробки та можливість подальшого масштабування системи.

Для реалізації вебплатформи було обрано систему керування базами даних MySQL. Дане рішення обумовлене її високою продуктивністю, сумісністю з мовою програмування PHP, простотою адміністрування та широким використанням у сучасній веброзробці. Крім того, MySQL підтримує механізми забезпечення цілісності даних та дозволяє ефективно працювати з великими обсягами інформації [29].

На основі проведеного аналізу функціональних вимог було визначено дві основні сутності предметної області: користувачі та модифікації. Для їх зберігання у базі даних створено відповідні таблиці, між якими встановлено логічні зв'язки.

Основною таблицею системи є таблиця користувачів `users`, яка містить інформацію про зареєстрованих відвідувачів вебплатформи. До її структури входять унікальний ідентифікатор користувача, ім'я користувача, адреса електронної пошти, хешований пароль та дата створення облікового запису. Використання унікальних ідентифікаторів дозволяє однозначно ідентифікувати кожного користувача незалежно від можливих змін інших атрибутів.

Таблиця `users` використовується під час виконання таких операцій:

- реєстрація нового користувача;
- авторизація в системі;
- формування користувацької сесії;
- відображення інформації в особистому кабінеті;
- прив'язка завантажених модифікацій до конкретного автора.

Для забезпечення безпеки персональних даних паролі користувачів не зберігаються у відкритому вигляді. Перед записом у базу даних вони проходять процедуру хешування засобами PHP, що унеможливорює отримання

початкового значення пароля навіть у випадку несанкціонованого доступу до бази даних. Зміст полів показано в таблиці 2.2.

Таблиця 2.2 – Специфікація структури таблиці користувачів `users`

Назва поля	Тип даних	Ключі / Індекси	NOT NULL	Значення за замовчуванням	Опис призначення поля
<code>id</code>	<code>INT</code>	Primary Key	Так	<i>Auto Increment</i>	Унікальний ідентифікатор (суррогатний ключ) облікового запису
<code>username</code>	<code>VARCHAR(50)</code>	Unique Index	Так	—	Унікальний логін користувача для авторизації в системі
<code>email</code>	<code>VARCHAR(100)</code>	Unique Index	Так	—	Адреса електронної пошти для зв'язку та верифікації
<code>password_hash</code>	<code>VARCHAR(255)</code>	—	Так	—	Криптографічний безпечний хеш пароля (PASSWORD_DEFAULT)
<code>created_at</code>	<code>TIMESTAMP</code>	—	Так	<code>CURRENT_TIMESTAMP</code>	Дата та час автоматичної реєстрації аккаунта в системі

Другою основною таблицею є таблиця `mods`, призначена для зберігання інформації про модифікації Minecraft. У таблиці містяться відомості про назву модифікації, її опис, ім'я завантаженого файлу та автора, який додав модифікацію до системи.

Кожен запис таблиці `mods` містить такі основні атрибути:

- унікальний ідентифікатор модифікації;
- назву модифікації;
- опис модифікації;
- ім'я файлу;
- ідентифікатор або ім'я користувача, який завантажив модифікацію.

Опис структури та властивостей полів таблиці `mods` бази даних показано в табл. 2.3.

Таблиця 2.3 – Специфікація структури таблиці модифікацій mods

Назва поля	Тип даних	Ключі / Індекси	NOT NULL	Значення за замовчуванням	Опис призначення поля
id	INT	Primary Key	Так	<i>Auto Increment</i>	Унікальний порядковий номер запису модифікації
name	VARCHAR(150)	—	Так	—	Назва додатка/модифікації, що відображається в каталозі
description	TEXT	—	Ні	NULL	Детальний текстовий опис функціоналу та особливостей мода
filename	VARCHAR(255)	—	Так	—	Унікальне санітизоване ім'я файлу (з timestamp) на сервері
uploaded_by	INT	Foreign Key	Так	—	Ідентифікатор автора (зв'язок 1:М із полем id таблиці users)
downloads	INT	—	Так	0	Лічильник кількості успішних скачувань файлу користувачами
filename	VARCHAR(255)	—	Так	—	Унікальне санітизоване ім'я файлу (з timestamp) на сервері

Наявність окремої таблиці для модифікацій дозволяє формувати каталог доступного контенту, здійснювати пошук та відображення інформації про кожен мод незалежно від інших елементів системи.

Таблиці пов'язані реляційним зв'язком за допомогою зовнішнього ключа (Foreign Key) `mods.uploaded_by → users.id`. Це забезпечує посилальну цілісність даних (якщо видалити користувача, система за допомогою каскадних правил дізнається, які модифікації йому належали).

Поля `username` та `email` у таблиці користувачів мають статус `Unique Index`. Це на рівні самої бази даних блокує реєстрацію дублікатів (не дозволить створити два акаунти з однаковою поштою чи логіном). Для поля `password_hash` виділено `VARCHAR(255)`. Це важливий архітектурний нюанс: хоча користувач вводить короткий пароль, функція

`password_hash()` у PHP завжди генерує довгий 60-символьний рядок, обсяг якого в майбутньому може зрости при зміні алгоритмів шифрування.

Особливістю інженерного проектування ресурсів дистрибуції є необхідність розділення збереження метаданих та безпосередньо самих файлів програмних продуктів. Фізичне збереження бінарних файлів великого обсягу у полях типу BLOB всередині СУБД призводить до критичного зниження продуктивності бази даних, розростання її розміру та збільшення часу відгуку сервера застосунків [29]. Запропонована архітектура бази даних та підсистема файлового збереження утворюють надійний, масштабований фундамент вебресурсу, що мінімізує час генерації сторінок каталогу і гарантує цілісність даних.

2.3 Проектування користувацького інтерфейсу та навігації вебплатформи

Важливою складовою розробки вебресурсу є проектування користувацького інтерфейсу (UI) та логіки взаємодії (UX) [30]. При створенні платформи «WorldMods» ключовими критеріями проектування фронтенд-частини були обрані мінімалізм, інтуїтивно зрозуміла навігація, адаптивність під різні типи пристроїв та забезпечення миттєвого відгуку системи на дії користувача. Для перегляду та завантаження модифікацій Minecraft особлива увага приділялася забезпеченню простоти навігації, доступності основних функцій та логічній структурі сторінок.

Проектування інтерфейсу здійснювалося відповідно до принципів сучасного вебдизайну, серед яких важливе місце займають мінімалізм, адаптивність, зрозумілість елементів керування та швидкий доступ до необхідної інформації. Враховуючи, що основною цільовою аудиторією системи є користувачі комп'ютерної гри Minecraft, інтерфейс було орієнтовано на швидкий пошук модифікацій та зручну роботу з каталогом контенту.

Проектування здійснювалося в інтегрованому середовищі Visual Studio Code, який підтримує роботу з усіма ораними мовами вебпрограмування [24]. Вигляд проєкту показано на рис. 2.3.

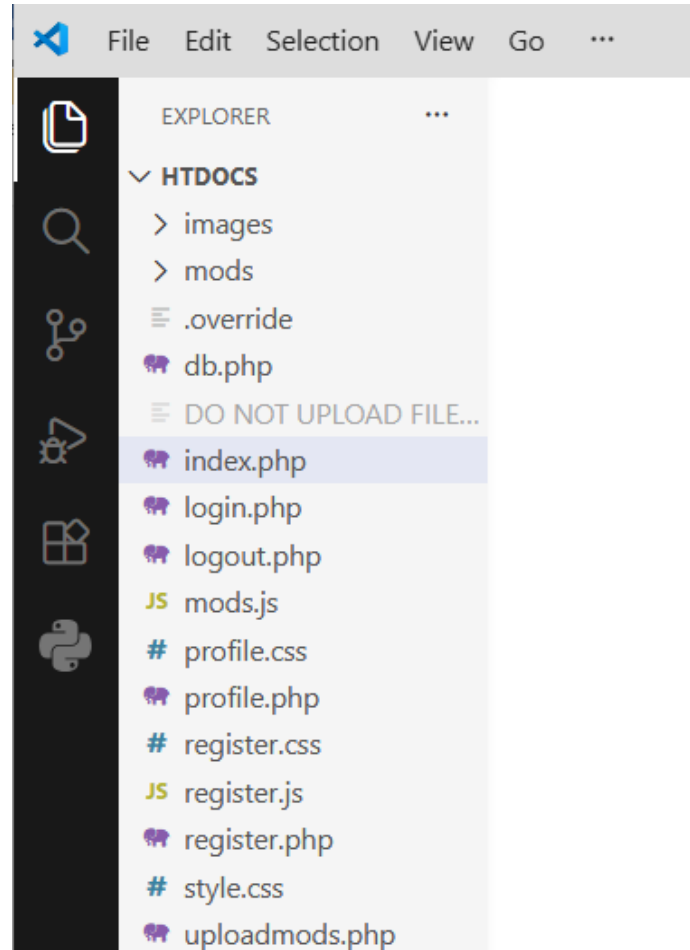


Рисунок 2.3 – Структура проєкту на панелі навігації Visual Studio Code

Структура вебплатформи побудована за принципом ієрархічної навігації, при якій користувач отримує доступ до всіх основних функцій через головне меню. Такий підхід забезпечує швидке переміщення між сторінками та спрощує освоєння системи новими відвідувачами. Центральним елементом вебплатформи є головна сторінка, яка виконує функцію стартового інформаційного вузла системи.

Інтерфейс платформи спроектовано за принципом SPA-елементів (Single Page Application) в межах головної сторінки, що дозволяє динамічно взаємодіяти з каталогом без постійного перезавантаження всього HTML-

документа. Підбір елементів керування слідує законам юзабіліті, має фіксовану модульну сітку. Вигляд макету головної сторінки представлено на рис. 2.4.

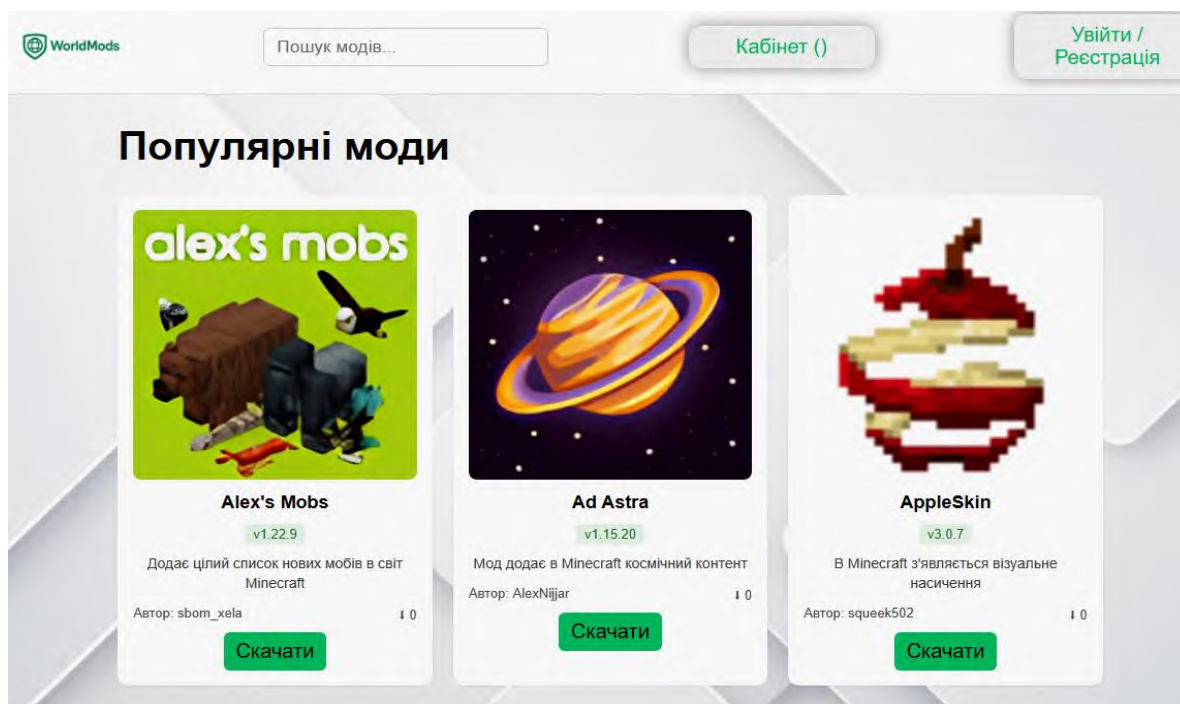


Рисунок 2.4 – Фрагмент макету інтерфейсу користувача на головній сторінці

На сторінці відображається перелік доступних модифікацій, що зберігаються у базі даних. Кожна модифікація представлена окремим інформаційним блоком із назвою, коротким описом та можливістю завантаження відповідного файлу (кнопка «Скачати»). Такий спосіб подання інформації забезпечує наочність каталогу та дозволяє користувачеві швидко ознайомитися зі змістом доступного контенту.

Для забезпечення зручності роботи із системою реалізовано навігаційне меню, яке містить посилання на основні функціональні модулі вебплатформи. Залежно від стану авторизації користувача набір доступних пунктів меню може змінюватися. Незареєстровані користувачі отримують доступ до перегляду каталогу та сторінок реєстрації й входу до системи, тоді як авторизовані користувачі можуть переходити до особистого кабінету та сторінки завантаження модифікацій.

Користувацький інтерфейс розділено на три базові функціональні зони:

1. Верхня панель навігації (Header): наскрізний елемент, що закріплений у верхній частині екрана за допомогою CSS-властивості `position: sticky; top: 0;`. Це забезпечує постійний доступ до навігації незалежно від глибини прокручування сторінки. У шапці розміщено фірмовий логотип бренду, інтерактивне поле пошуку та контекстну кнопку автентифікації/доступу до персонального кабінету.

2. Головна робоча область (Main): центральний контейнер, що займає оптимальну для сприйняття ширину (80 % від площі екрана). Тут транслюється динамічна сітка карток модифікацій, форми реєстрації, авторизації та завантаження файлів.

3. Каталог контенту (Grid Layout): реалізований на базі сучасної технології CSS Grid. Декларація `display: grid; grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));` гарантує автоматичну перебудову інтерфейсу: кількість колонок у каталозі адаптивно змінюється залежно від роздільної здатності монітора чи екрана мобільного телефона.

Основним елементом представлення інформації є картка модифікації (Mod Card). Вона спроектована як цілісний інформаційний блок, що містить такі компоненти:

- візуальний маркер (прев'ю-зображення модифікації);
- текстові атрибути (назва, версія сумісності, короткий опис функціоналу);
- метадані (автор розробки та лічильник завантажень);
- цільова кнопка дії («Скачати»).

Для покращення інтерактивності (UX Feedbacks) застосовано плавні анімаційні ефекти. При наведенні курсору миші на картку мода (`.mod-card:hover`) спрацьовує трансформація підйому `transform: translateY(-3px);` та посилення тіні об'єкта протягом 0.2 секунди, що створює ефект тривимірного об'єму і фокусує увагу.

Окрему увагу приділено інтерфейсам реєстрації та авторизації. Форма зорієнтована по центру екрана, має фіксовану ширину (350 px) і контрастне забарвлення відносно фону. Поля введення забезпечені псевдокласом `:focus`, який при активації інпуту підсвічує його контур яскраво-зеленим кольором (`outline: 2px solid #00c16a;`), сигналізуючи про готовність до введення.

Навігаційні маршрути вебплатформи базуються на гібридному підході. Перехід між глобальними станами (каталог, реєстрація, завантаження) реалізовано через прямі гіперпосилання, оформлені у вигляді ергономічних кнопок-контейнерів із заокругленими кутами (`border-radius: 12px;`). Схему навігації представлено на рис. 2.5, яка дає повне уявлення про призначення кожного елемента і порядок доступу.

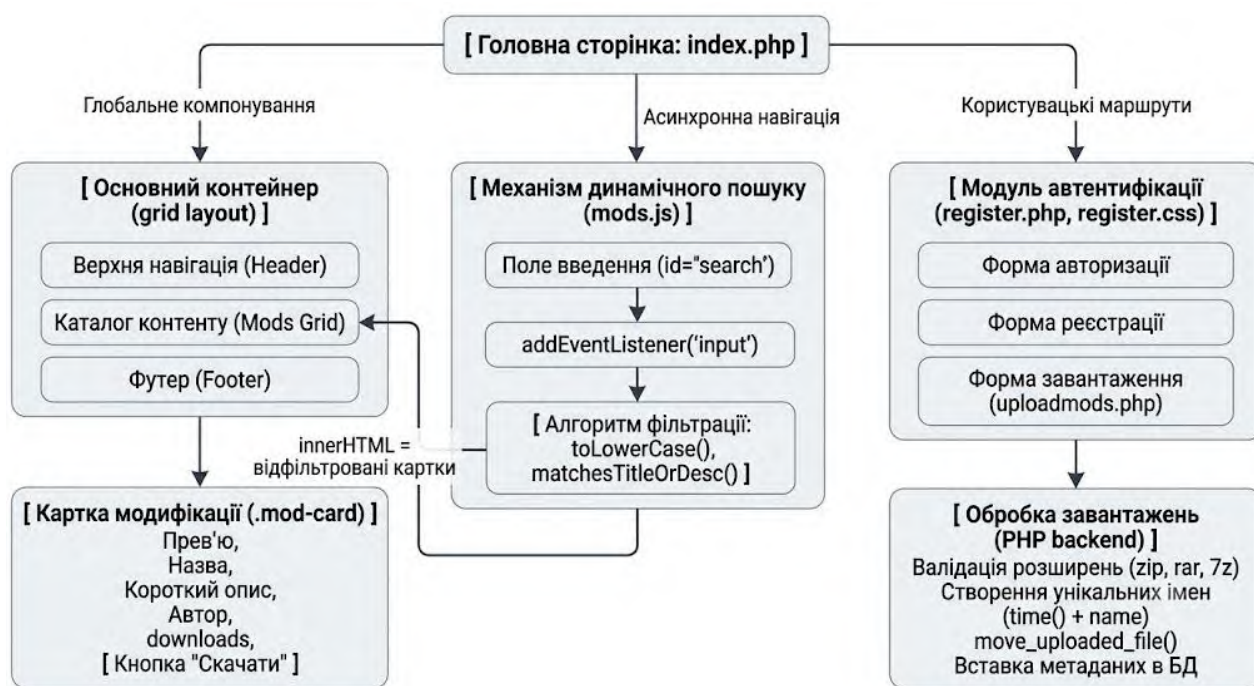


Рисунок 2.5 – Навігаційні маршрути вебплатформи

Як видно з рис. 2.6, архітектура інтерфейсу побудована на взаємодії клієнтських та серверних сценаріїв. Представлена схема наочно демонструє розподіл функціональних обов'язків вебресурсу на три автономні вектори, що беруть свій початок від кореневого компонента `index.php`.

1. Вектор глобального компонування (UI-структура): описує статичну та динамічну декомпозицію інтерфейсу. Через систему фіксованих CSS-правил та модульної сітки (Grid Layout) здійснюється структурування головного екрана (шапка, адаптивна сітка карт контенту, футер). Кінцева сутність `.mod-card` спроектована як самодостатній інформаційний UI-блок, який інкапсулює у собі всі необхідні метадані для гравця та цільову кнопку скачування файлу.

2. Вектор асинхронної навігації (логіка клієнтського пошуку): показує алгоритм роботи підсистеми миттєвої фільтрації контенту без перезавантаження сторінки. Скрипт `mods.js` через слухач подій перехоплює введення символів у поле `#search`, валідує рядок, порівнює його із заголовками або описами модифікацій і здійснює пряму маніпуляцію з DOM-деревом (через очищення та перезапис контейнера `innerHTML`), забезпечуючи безшовний UX.

3. Вектор користувацьких маршрутів (серверні сценарії): відображає логіку переходу на ізольовані форми системи (реєстрація, авторизація, завантаження нових розробок). Схема демонструє, як дані з форми `uploadmods.php` проходять двоетапну обробку на бекенді: фізичну санітизацію та перевірку безпеки файлу (дозволені розширення `zip`, `rar`, `7z` та конкатенація імені з `time()`), з одночасним розчепленням потоків даних - завантаженням бінарного архіву в директорію `/mods/` та реєстрацією метаданих у реляційній базі даних.

Важливим компонентом інтерфейсу є форма реєстрації користувачів. Під час її проєктування було забезпечено мінімальну кількість обов'язкових полів для введення інформації: необхідно вказати ім'я облікового запису, адресу електронної пошти та пароль. Спрощена структура форми сприяє швидкому створенню нового облікового запису та зменшує ймовірність виникнення помилок під час введення даних.

Аналогічний підхід використано під час проєктування форми авторизації. Вона містить лише необхідні поля для введення логіна та пароля, що забезпечує швидкий доступ до функціональних можливостей системи. Після

успішного входу користувач перенаправляється до персоналізованого середовища роботи із вебплатформою.

Особистий кабінет користувача виконує функцію центру керування обліковим записом. На відповідній сторінці відображається інформація про зареєстрованого користувача, а також доступні дії щодо роботи із системою. Реалізація особистого кабінету підвищує рівень персоналізації сервісу та забезпечує більш зручну взаємодію користувача з платформою.

Для підвищення інтерактивності системи використовується JavaScript, який забезпечує обробку окремих дій користувача без необхідності перезавантаження сторінки. Використання клієнтських сценаріїв дозволяє покращити швидкодію інтерфейсу та підвищити комфорт роботи із вебплатформою.

Особлива увага під час проєктування приділялася адаптації інтерфейсу до різних розмірів екранів. Вебплатформа коректно відображається на персональних комп'ютерах, ноутбуках та інших пристроях із сучасними веббраузерами. Це забезпечує доступність сервісу для широкого кола користувачів незалежно від використовуваного обладнання.

Таким чином, розроблений користувацький інтерфейс забезпечує швидкий доступ до основних функцій вебплатформи, зручну навігацію між її компонентами та комфортну взаємодію користувачів із системою. Запропоновані рішення відповідають сучасним вимогам до вебдодатків та створюють основу для подальшого розвитку функціональних можливостей платформи.

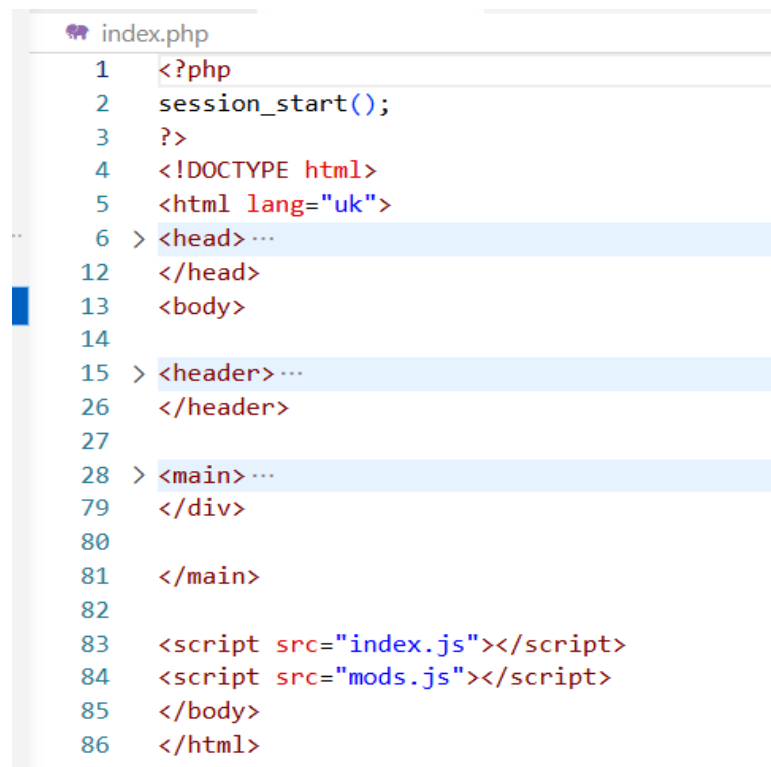
РОЗДІЛ 3

ПРОГРАМНО-ТЕХНІЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

ВЕБПЛАТФОРМИ

3.1 Розробка клієнтської частини засобами HTML, CSS І PHP та основних функціональних модулів

Після завершення етапу проєктування було виконано розробку вебплатформи для перегляду та завантаження модифікацій комп'ютерної гри Minecraft. Сайт складається з головної сторінки `index.php`, файлу стилів `style.css` для цієї сторінки, сторінки реєстрації `register.php` та файлу зі скриптами `register.js` для неї, авторизації `login.php`, файлу стилів `register.css` для цих двох сторінок, сторінки профілю користувача `profile.php` і файлу стилів `profile.css` цієї сторінки, і файлу `db.php`, що відповідає за підключення баз даних. Окремо для файлів модів виділена папка під назвою `mods`, а також для зображень під назвою `images` (рис. 3.1).



```
index.php
1  <?php
2  session_start();
3  ?>
4  <!DOCTYPE html>
5  <html lang="uk">
6  > <head> ...
12 </head>
13 <body>
14
15 > <header> ...
26 </header>
27
28 > <main> ...
79 </div>
80
81 </main>
82
83 <script src="index.js"></script>
84 <script src="mods.js"></script>
85 </body>
86 </html>
--
```

Рисунок 3.1 – Структура головної сторінки `index.php`

Повний лістинг коду `index.php` представлено в додатку А. Сторінка забезпечує доступ до каталогу модифікацій. Після відкриття сторінки користувач отримує перелік доступних модів, інформація про які автоматично завантажується з бази даних. Для кожної модифікації створюється окремий HTML-блок, який містить назву, версію та зображення моду. Виведення здійснюється у циклі, що дозволяє автоматично відображати будь-яку кількість модів без зміни структури сторінки з відповідним стилем показу. Кожен блок має структуру, представлену на рис. 3.2.

```
foreach ($mods as $mod): ?>
    <div class="mod-card">
        "
        style="width:100%; border-radius:8px; margin-bottom:10px;" />
        <div class="mod-title"><?php echo htmlspecialchars($mod['title']); ?></div>
        <div class="mod-version">Версія <?php echo htmlspecialchars($mod['version']); ?></div>
        <div class="mod-desc"><?php echo htmlspecialchars($mod['desc']); ?></div>
        <div class="mod-footer">
            <span>Автор: <?php echo htmlspecialchars($mod['author']); ?></span>
            <span>Скачано: <?php echo htmlspecialchars($mod['downloads']); ?></span>
        </div>
        <a href="<?php echo htmlspecialchars($mod['file']); ?>" class="download-btn">Скачати</a>
    </div>
<?php endforeach; ?>
```

Рисунок 3.2 – HTML-блок для кожної модифікації

Фрагмент бере записану інформацію про моди за тегами і виводить її у відповідних чарунках. За функціонал блоків, що містять відповідні моди на сайті, відповідає частина коду з файлу `index.php`, рядки 35-63 (див. додаток А).

Окрему роль у роботі клієнтської частини відіграє сценарій `mods.js`. Даний JavaScript-модуль використовується для реалізації інтерактивних елементів інтерфейсу та обробки дій користувача без необхідності повного перезавантаження сторінки. Це підвищує швидкодію вебресурсу та забезпечує більш комфортну взаємодію із системою.

Для оформлення зовнішнього вигляду вебплатформи було створено набір каскадних таблиць стилів. Файл `style.css` використовується для оформлення головної сторінки та каталогу модифікацій. Файл `register.css` забезпечує єдине оформлення сторінок реєстрації та авторизації, а `profile.css` використовується для візуального оформлення особистого кабінету користувача. Такий підхід

дозволяє підтримувати модульність проєкту та спрощує внесення змін до окремих компонентів інтерфейсу. На рис. 3.3 показано момент трансформації зображення (зсув картки по вертикалі при наведенні).

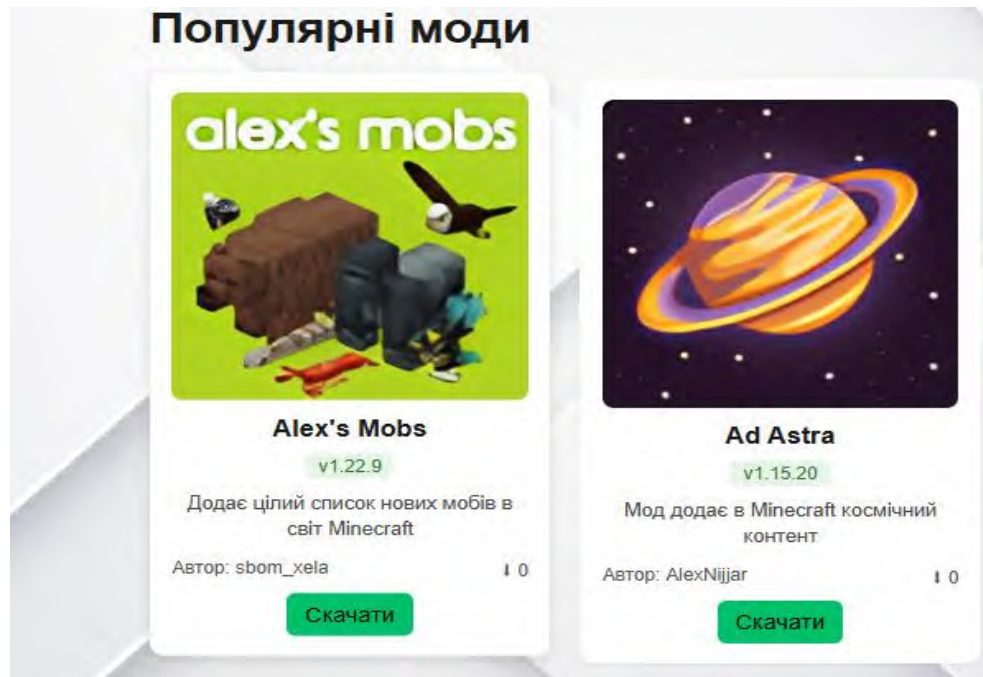


Рисунок 3.3 – Приклад трансформації (динамічний ефект) при наведенні

Ефект, показаний на рис. 3.3, описаний у відповідній частині файлу стилів style.css, де описані й інші властивості карток (рис. 3.4).

```

53 .mod-card {
54     background: white;
55     padding: 15px;
56     border-radius: 10px;
57     box-shadow: 0 1px 4px rgba(0,0,0,0.06);
58     <element class="mod-card" :hover>
59     Selector Specificity: (0, 2, 0)
60
61     .mod-card:hover {
62         transform: translateY(-20px);
63         box-shadow: 0 3px 10px rgba(0,0,0,0.12);
64     }

```

Рисунок 3.4 – Приклад застосування функції трансформації для класу .mod-card

Важливою складовою системи є модуль реєстрації користувачів, реалізований у файлі register.php. Його призначення полягає у створенні нових

облікових записів та занесенні інформації про користувачів до бази даних. На основній сторінці сайту є кнопка особистого кабінету в правій верхній частині сторінки. Вона дає можливість користувачеві увійти в особистий кабінет. Але якщо користувач не зареєструвався, або не авторизувався на сервері, то замість кнопки «Кабінет», він буде бачити іншу кнопку з написом «Увійти / Реєстрація» (рис. 3.5). Під час реєстрації користувач вводить ім'я облікового запису, адресу електронної пошти та пароль. Після перевірки коректності введених даних інформація зберігається в таблиці користувачів, а пароль проходить процедуру хешування для підвищення рівня безпеки.



Рисунок 3.5 – Набір кнопок для входу в кабінет або реєстрації

Лістинг скрипта для обробки логіну, пароллю користувача показано на рис. 3.6.

```
JS register.js > ...
1  const form = document.getElementById('registerForm');
2  const password = document.getElementById('password');
3  const confirm_password = document.getElementById('confirm_password');
4  const error = document.getElementById('error');
5
6  form.addEventListener('submit', function(e) {
7      error.textContent = "";
8
9      if (password.value.length < 8) {
10         e.preventDefault();
11         error.textContent = "Пароль має бути не менше 8 символів!";
12     } else if (password.value !== confirm_password.value) {
13         e.preventDefault();
14         error.textContent = "Паролі не співпадають!";
15     }
16 });
```

Рисунок 3.6 – Лістинг коду на JavaScript для обробки даних форми

Коли користувач натискає кнопку «Увійти / Реєстрація», посилання відправляє його на сторінку авторизації у спеціальній формі (рис. 3.7).

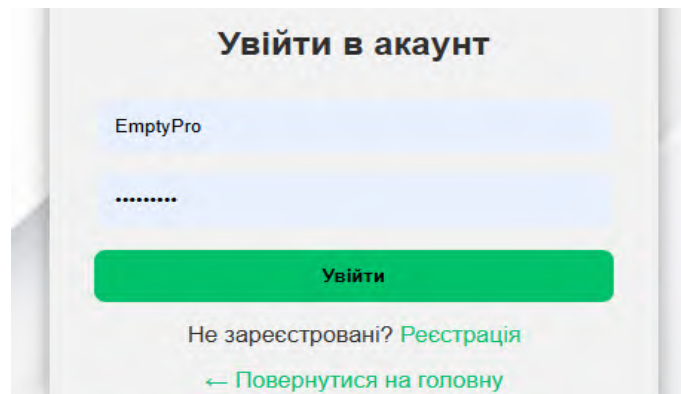


Рисунок 3.7. Зовнішній вигляд сторінки авторизації на сайті

Поля вводу (див. рис. 3.7) призначені для того, щоб користувач використав свої дані для авторизації на сайті, якщо він був зареєстрований до цього. У верхнє поле він має ввести свій логін, а в нижнє – пароль.

За структуру цієї сторінки відповідає файл `login.php`, а за стиль цієї сторінки – `register.css`. Ось так зі сторони коду виглядає структура форми, яку заповнює користувач для авторизації, а також кнопок, що відповідають за перехід на інші сторінки сайту, зокрема головну сторінку і реєстрації (рис. 3.8).

```

62 <form action="" method="POST">
63     <input type="text" name="username" placeholder="Логін" required>
64     <input type="password" name="password" placeholder="Пароль" required>
65     <button type="submit">Увійти</button>
66     <p class="message" style="color: <?php echo $messageColor; ?>;">
67     <?php echo htmlspecialchars($message); ?>
68     </p>
69     <p>Не зареєстровані? <a class="login" href="register.php">Реєстрація</a></p>
70     <p id="error" style="color:red;"></p>
71     <p id="reg-message" style="color:green;"></p>
72 </form>
73 <a class="back" href="index.php">← Повернутися на головну</a>
74 </div>
75
76 <script src="register.js"></script>

```

Рисунок 3.8 – Опис структури форми реєстрації і перевірки даних у `login.php`

Коли користувач заповнив форму, тобто ввів свої дані: логін і пароль, форма відправляє запит до частини коду, який показано в додатку Б. Ця частинка коду, отримує запит, що містить вхідні дані, і тільки тоді починає

обробляти його. Спочатку скрипт перевіряє, чи існує користувач з введеним логіном. Якщо так, то система пропускає далі, якщо ж ні, то відправляє команду на виведення червоного надпису під кнопкою «Увійти» про те, що користувача з таким логіном не існує. В ситуації, якщо логін такий існує, скрипт перевіряє чи співпадає введений у форму пароль з паролем із таблиці в базі даних. Якщо ні, то видає відповідну помилку, що і з логіном. Інакше підтверджує вхід користувача. Якщо користувач заповнив не всі поля, то скрипт не запуститься, видасть помилку, потрібно буде заповнити всі поля.

Скрипт, який відповідає за обробку запиту при заповненні цієї форми, має лістинг, показаний на рис. 3.9.

```

15  if ($_SERVER['REQUEST_METHOD'] === 'POST') {
16
17      $username = trim($_POST['username']);
18      $email = trim($_POST['email']);
19      $password = $_POST['password'];
20
21      if (!$username || !$email || !$password) {
22          $message = "Будь ласка, заповніть усі поля!";
23      } else {
24          $passwordHash = password_hash($password, PASSWORD_DEFAULT);
25
26          $stmt = $pdo->prepare("SELECT id FROM users WHERE username = ? OR email = ?");
27          $stmt->execute([$username, $email]);
28
29          if ($stmt->fetch()) {
30              $message = "Користувач з таким логіном або email вже існує!";
31          } else {
32              $stmt = $pdo->prepare("INSERT INTO users (username, email, password_hash) VALUES (?, ?, ?)");
33              if ($stmt->execute([$username, $email, $passwordHash])) {
34                  $message = "Реєстрація пройшла успішно!";
35                  $messageColor = 'green';
36              } else {
37                  $message = "Сталася помилка при збереженні бази.";
38              }
39          }
20      }

```

Рисунок 3.9 – Частина коду в register.php, яка відповідає за валідацію форми

Скрипт, отримавши дані з форми, перевіряє, чи зайнятий введений логін. Якщо так, то скрипт видає помилку, а якщо ні, то переходить до наступного етапу. Він перевіряє, чи зайнята введена пошта: якщо так – помилка, інакше переходить до обробки паролю. Він перевіряє, чи співпадають між собою паролі паролі, введені в поля «Пароль» і «Повторіть пароль», а також звертається до файлу **register.js**, скрипт в якому перевіряє, чи введений пароль

містить 8, або більше символів. Якщо ні – помилка, інакше скрипт відправляє повідомлення про успішну реєстрацію.

CSS3 (Cascading Style Sheets) забезпечує візуалізацію інтерфейсу на базі сучасних модулів макетування (Flexbox, CSS Grid). Для оформлення зовнішнього вигляду вебплатформи було створено набір каскадних таблиць стилів. Файл `style.css` використовується для оформлення головної сторінки та каталогу модифікацій. Файл `register.css` забезпечує єдине оформлення сторінок реєстрації та авторизації, а `profile.css` використовується для візуального оформлення особистого кабінету користувача. Такий підхід дозволяє підтримувати модульність проєкту та спрощує внесення змін до окремих компонентів інтерфейсу.

У результаті реалізації клієнтської частини було створено комплекс взаємопов'язаних функціональних модулів, які забезпечують виконання всіх основних завдань вебплатформи. Розроблена система надає користувачам можливість переглядати каталог модифікацій, проходити реєстрацію та авторизацію, завантажувати власні модифікації та працювати з персоналізованими сервісами вебресурсу. Отриманий програмний продукт характеризується простою структурою, зручним інтерфейсом та можливістю подальшого розширення функціональних можливостей.

3.2 Реалізація серверної частини вебплатформи засобами PHP

Серверна частина вебплатформи реалізована за допомогою мови програмування PHP, яка забезпечує обробку запитів користувачів, взаємодію з базою даних, керування сесіями та виконання операцій із файлами. Використання PHP дозволило створити динамічний вебзастосунок, що формує вебсторінки відповідно до поточного стану системи та даних, які зберігаються у базі даних. Архітектура серверної частини побудована на взаємодії окремих PHP-модулів, кожен з яких відповідає за виконання певної функції. Основними

компонентами є модулі підключення до бази даних, реєстрації та авторизації користувачів, керування користувацькими сесіями та завантаження модифікацій.

Для роботи із базою даних використовується окремий модуль `db.php`, який забезпечує централізоване підключення до сервера MySQL [22]. Такий підхід дозволяє уникнути дублювання коду та спрощує адміністрування системи. Підключення до бази даних здійснюється шляхом створення об'єкта класу `mysql` із зазначенням параметрів сервера, імені користувача, пароля та назви бази даних. Вигляд вікна для роботи з базою даних показано на рис. 3.10.

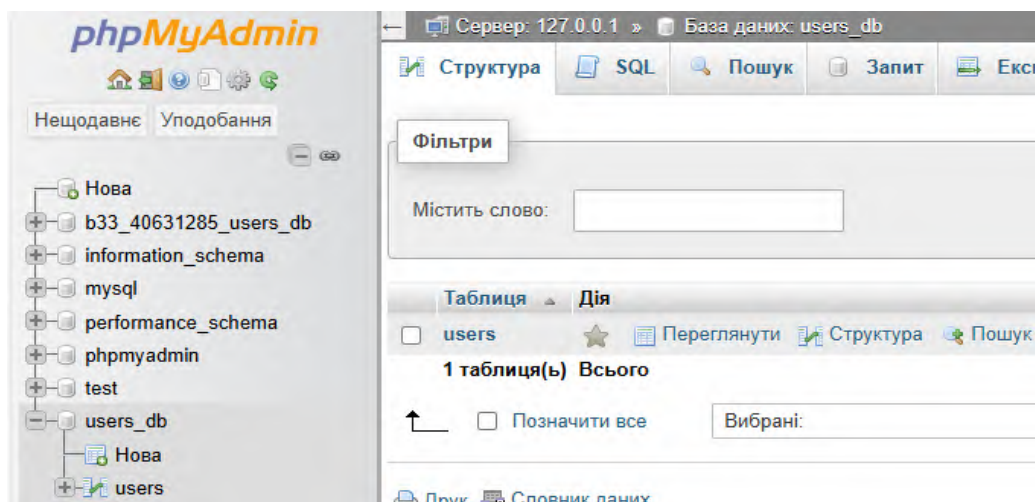


Рисунок 3.10 – Зовнішній вигляд бази даних SQL під іменем `users_db`

Таблиця `users` зберігає дані користувачів, що пройшли реєстрацію. Інформацію про кожного користувача можна редагувати прямо в таблиці, що є корисно для розробника сайту, адже при зміні чи додаванні нових змінних в таблицю можна, наприклад, змінити роль користувача на значення «admin». За роботу бази даних відповідає файл під назвою `db.php`, який має вигляд (рис. 3.11). Цей файл, підключає базу даних SQL до сайту, дозволяючи файлам, що мають посилання на цей файл, отримувати, дані з бази даних, і відправляти дані в базу даних.

Одна з функцій, яку виконує цей файл, – запускає скрипт, який на базі отриманих даних з форми реєстрації, намагається створити новий профіль в таблиці бази даних, якщо активне підключення до потрібної БД.

```

1  <?php
2  $host = "sql100.byethost33.com";
3  $dbname = "b33_40631285_users_db";
4  $user = "b33_40631285";
5  $password = "_umabor_619";
6  $charset = 'utf8mb4';
7
8  $dsn = "mysql:host=$host;dbname=$db;charset=$charset";
9  $options = [
10     PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
11     PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
12 ];
13
14 try {
15     $pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $user, $password);
16     $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
17 } catch (PDOException $e) {
18     die("Помилка підключення до бази: " . $e->getMessage());
19 }
20 ?>

```

Рисунок 3.11 – Лістинг коду файлу db.php, відповідального за базу даних

Після успішної авторизації створюється користувацька сесія. Використання механізму сесій PHP дозволяє зберігати інформацію про авторизованого користувача між різними сторінками вебресурсу. Дані сесії використовуються для контролю доступу до особистого кабінету та сторінки завантаження модифікацій. Для завершення роботи із системою реалізовано механізм знищення активної сесії:

```

<?php
session_start();
session_destroy();
header("Location: login.php");
exit();

```

Особливе значення для вебплатформи має модуль завантаження модифікацій. Під час виконання цієї операції сервер приймає файл від користувача, перевіряє його параметри та зберігає у файловому сховищі (рис. 3.11). Одночасно до бази даних записується інформація про нову модифікацію. Приклад завантаження файлу на сервер:

```

$fileName = $_FILES['modfile']['name']; $tmpName =
$_FILES['modfile']['tmp_name'];

move_uploaded_file( $tmpName, "mods/" . $fileName );

```

```

11     $file = $_FILES['mod_file'];
12     $upload_dir = 'mods/';
13
14     // Перевірка файлу
15     $allowed_ext = ['zip', 'rar', '7z'];
16     $ext = pathinfo($file['name'], PATHINFO_EXTENSION);
17
18     if (!in_array(strtolower($ext), $allowed_ext)) {
19         $message = "Непідтримуваний формат файлу!";
20     } elseif ($file['error'] !== 0) {
21         $message = "Сталася помилка при завантаженні файлу!";
22     } else {
23         $new_filename = time() . '_' . basename($file['name']);
24         if (move_uploaded_file($file['tmp_name'], $upload_dir . $new_filename)) {
25             // Зберігаємо в базі
26             $stmt = $pdo->prepare("INSERT INTO mods (name, description, filename, uploaded_by) VALUES (?, ?, ?, ?)");
27             if ($stmt->execute([$mod_name, $description, $new_filename, $user_id])) {
28                 $message = "Мод успішно завантажено! Чекає модерації.";
29             } else {
30                 $message = "Помилка при збереженні інформації в базу.";

```

Рисунок 3.12 – Код завантаження мод в uploadmods.php

Використання PHP дозволило реалізувати повний цикл обробки користувацьких запитів: від отримання даних через вебформи до їх збереження у базі даних та подальшого відображення на сторінках вебплатформи. Розроблена серверна частина забезпечує стабільну роботу системи, підтримує авторизацію користувачів, керування модифікаціями та взаємодію із файловим сховищем.

Загальна перевірка й тестування платформи проводилося за основними напрямками. Приклад адаптивного перегляду на зменшених екранах показано у додатку В.

3.3 Економічне оцінювання варіантів розробки вебсайту

Економічна оцінка програмного продукту є важливим етапом аналізу результатів розробки інформаційної системи. Вона дозволяє визначити орієнтовну вартість створення вебплатформи та оцінити доцільність її впровадження. Для розрахунку вартості вебплатформи «WorldMods» враховано витрати на проектування, програмування, тестування та підготовку програмного продукту до експлуатації.

Необхідно взяти до уваги, що розробка вебплатформи здійснювалася із використанням безкоштовного програмного забезпечення та технологій з відкритим вихідним кодом, серед яких PHP, MySQL, HTML, CSS та JavaScript.

Основною статтею витрат виступає оплата праці розробника. Для оцінювання трудомісткості робіт проєкт було умовно поділено на окремі етапи, наведені в таблиці 3.1.

Таблиця 3.1 – Оцінка трудомісткості розробки вебплатформи

№ з/п	Етап виконання робіт	Тривалість, год
1.	Аналіз предметної області	12
2.	Проектування структури системи	16
3.	Проектування бази даних	10
4.	Розробка клієнтської частини	28
5.	Розробка серверної частини	36
6.	Реалізація авторизації та реєстрації	12
7.	Реалізація модуля завантаження модифікацій	18
8.	Тестування та налагодження	18
9.	Підготовка документації	10
10.	Разом	160

Для розрахунку вартості розробки приймемо середню погодинну ставку веброзробника на рівні 350 грн за годину [31]. Найбільш важливим пунктом при розробці продукту є розрахунок його собівартості й визначення кінцевої ціни, оскільки саме вона має основну вагу при визначенні переваг над конкурентами та доцільності розробки системи.

Станом на кінець 2025 р. – початок 2026 р. середня погодинна ставка Junior/Middle Front-End розробника в Україні становить приблизно 350–500 грн/год. Для наближеного оцінювання проєкту доцільно позначити погодинну ставку $C_{год}$ і взяти середнє значення погодинної ставки: $C_{год} = 350$ грн/год.

Основна вартість розробки в частині заробітної плати буде розрахована за формулою:

$$C_{розр} = T \times C_{год}, \quad (3.1)$$

де $C_{розр}$ – вартість розробки, грн;

T – загальна трудомісткість робіт, год;

$C_{год}$ – погодинна ставка розробника, грн/год.

Підставляючи числові значення, отримаємо:

$$C_{розр} = 160 \times 350 = 56000 \text{ (грн)}.$$

Таким чином, вартість робіт з розробки вебплатформи орієнтовно становить 56000 грн. Додатково можуть враховуватися витрати на розміщення вебресурсу в мережі Інтернет. Для невеликого проєкту достатньо використання віртуального хостингу вартістю близько 2500 грн/рік та доменного імені вартістю близько 500 грн/рік.

У такому випадку повна вартість робіт $C_{повн}$ буде становити:

$$C_{повн} = 56000 + 25000 + 500 ,$$

$$C_{повн} = 59000 \text{ грн}.$$

Дані, які використані для підрахунку витрат на розроблення вебсайту, було взято з відкритих джерел. Зокрема, відомості про розцінки роботи вебпрограміста публікує work.ua [31], хостинг – офіційний сайт [32].

Таким чином, загальна вартість створення та впровадження вебплатформи «WorldMods» становить приблизно 59 тис. грн. Отриманий результат свідчить про відносно невисоку собівартість програмного продукту, що пояснюється використанням відкритих технологій та відсутністю витрат на придбання ліцензійного програмного забезпечення.

Проведена оцінка демонструє економічну доцільність розробки вебплатформи для перегляду та завантаження модифікацій Minecraft. У разі комерційного використання подібного ресурсу витрати на його створення можуть бути компенсовані за рахунок реклами, платних сервісів або партнерських програм, пов'язаних із поширенням ігрового контенту.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було проведено аналіз сучасного стану індустрії комп'ютерних ігор та особливостей функціонування гри Minecraft як однієї з найпопулярніших платформ для створення користувацького контенту. Встановлено, що модифікації є важливим інструментом розширення функціональних можливостей гри та формування активних спільнот користувачів, тому, враховуючи актуальність відповідного програмного забезпечення, була створена спеціальна вебплатформа для завантаження модифікацій. Підсумовуючи досягнуті результати, можна зробити наступні висновки.

1. У теоретичній частині роботи досліджено існуючі вебресурси для перегляду та завантаження модифікацій, визначено їхні переваги та недоліки, що дозволило сформулювати перелік функціональних вимог до розроблюваної вебплатформи. Проаналізовані сучасні технології веброзробки та обґрунтована доцільність використання для серверної частини мови програмування PHP, системи керування базами даних MySQL, а також технологій HTML, CSS і JavaScript для реалізації поставленого завдання..

2. Спроектовано структуру вебплатформи, визначено її трирівневу архітектуру, функціональні модулі та механізми взаємодії між користувачем, серверною частиною, базою даних і файловим сховищем.

3. Розроблено структуру бази даних для зберігання інформації про користувачів та модифікації, визначено логічні зв'язки між основними сутностями та забезпечено ефективну організацію зберігання інформаційних ресурсів системи.

4. Реалізовано клієнтську частину вебресурсу, що забезпечує перегляд каталогу модифікацій, роботу навігаційних елементів та взаємодію користувача із системою, а також серверну частину вебплатформи засобами PHP, яка забезпечує обробку користувацьких запитів, взаємодію з базою даних, керування сесіями та виконання операцій із файлами.

5. Розроблено та впроваджено функціональні модулі реєстрації користувачів, авторизації, особистого кабінету та завантаження модифікацій, що забезпечують повний цикл роботи із контентом вебплатформи.

6. Реалізовано механізм безпечного зберігання облікових даних користувачів із використанням хешування паролів та засобів керування користувацькими сесіями.

7. Запропоновано вдале технічне рішення щодо роздільного зберігання метаданих модифікацій у базі даних та самих файлів у файловому сховищі сервера, що сприяє підвищенню продуктивності системи та спрощує її подальше адміністрування.

8. Виконано оцінювання вартості розробки вебплатформи та підтверджено економічну доцільність створення подібного програмного продукту на основі відкритих технологій і програмного забезпечення.

Практичним результатом роботи стала створена авторська вебплатформа «WorldMods», яка забезпечує перегляд, завантаження та розміщення модифікацій для гри Minecraft. Розроблена система є повністю працездатною, відповідає поставленим вимогам та може використовуватися як основа для подальшого розвитку спеціалізованих вебресурсів, орієнтованих на поширення користувацького ігрового контенту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Biggest Game Companies in the World and Their Metrics in 2025. *Outlook Respawn*. URL: <https://respawn.outlookindia.com/gaming/gaming-guides/biggest-game-companies-in-the-world-and-their-metrics-in-2025> (дата звернення: 04.05.2026).
2. Bonifield S. Report shows 90% of European gaming revenue in 2024 was digital purchases, but only 15% was on PC. *PC Gamer*. URL: <https://www.pcgamer.com/gaming-industry/report-shows-90-percent-of-european-gaming-revenue-in-2024-was-digital-purchases-but-only-15-percent-was-on-pc/> (дата звернення: 04.05.2026).
3. Top Video Game Companies by Revenue 2024. *ANYSTAT - Interactive Statistics & Charts*. ANYSTAT. URL: <https://anystat.ai/stats/148/top-video-game-companies-revenue-2024> (дата звернення: 04.05.2026).
4. Welcome to the official site of Minecraft. *Minecraft.net*. URL: <https://www.minecraft.net/en-us> (дата звернення: 04.05.2026).
5. Aleem, S., Capretz, L.F. & Ahmed, F. Game development software engineering process life cycle: a systematic review. *Journal of Software Engineering Research and Development*. 2016. 4 (6). <https://doi.org/10.1186/s40411-016-0032-7>
6. Engström H., Marklund B., Backlund P., Toftedahl M. Game development from a software and creative product perspective: A quantitative literature review approach. *Entertainment Computing*. 2018. Vol. 27. Pp. 10-22. <https://doi.org/10.1016/j.entcom.2018.02.008>.
7. Bile A. Development of intellectual and scientific abilities through game-programming in Minecraft. *Educ Inf Technol*. 2022. 27. Pp.7241–7256 <https://doi.org/10.1007/s10639-022-10894-z>
8. Lee D., Rajbahadur G. K., Lin D., Sayagh M., Bezemer C.-P., Hassan A. E. An Empirical Study of the Characteristics of Popular Minecraft Mods. *Empirical*

Software Engineering. 2020. 25 Pp. 3396-3429. <https://doi.org/10.1007/s10664-020-09840-9>

9. Fabric. *Fabric*. URL: <https://fabricmc.net/> (date of access: 08.06.2026).

10. Modrinth. *Modrinth*. URL: <https://modrinth.com/> (дата звернення: 04.05.2026).

11. Politowski C., Petrillo F., C. Ullmann G., Guéhéneuc Y. Game industry problems: An extensive analysis of the gray literature. *Information and Software Technology*. 2021. Volume 134. 106538. <https://doi.org/10.1016/j.infsof.2021.106538>

12. Politowski C., Petrillo F., Montandon J. E., Valente M. T., Guéhéneuc Y. Are game engines software frameworks? A three-perspective study. *Journal of Systems and Software*. 2021. Volume 171. 110846. ISSN 0164-1212. <https://doi.org/10.1016/j.jss.2020.110846>.

13. CurseForge. URL: <https://www.curseforge.com/minecraft>

14. Modrinth. URL: <https://modrinth.com/> (дата звернення: 04.05.2026).

15. PlanetMinecraft. URL: <https://www.planetminecraft.com/> (дата звернення: 04.05.2026).

16. PHP: Documentation. *PHP*. URL: <https://www.php.net/docs.php> (дата звернення: 04.05.2026).

17. Usage Statistics and Market Share of Server-side Programming Languages for Websites, June 2026. *W3Techs - extensive and reliable web technology surveys*. URL: https://w3techs.com/technologies/overview/programming_language (дата звернення: 04.05.2026).

18. PHP: Preface - Manual. *PHP*. URL: <https://www.php.net/manual/> (дата звернення: 04.05.2026).

19. HTML: HyperText Markup Language..MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (дата звернення: 04.05.2026).

20. CSS: Cascading Style Sheets. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення: 04.05.2026).
21. JavaScript. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення: 04.05.2026).
22. MySQL: MySQL Documentation. MySQL : Developer Zone. URL: <https://dev.mysql.com/doc/> (дата звернення: 04.05.2026).
23. Welcome! - The Apache HTTP Server Project. *Welcome! - The Apache HTTP Server Project*. URL: <https://httpd.apache.org/> (дата звернення: 04.05.2026).
24. Visual Studio Code. URL: <https://code.visualstudio.com/> (дата звернення: 04.05.2026).
25. Git - Reference. *Git*. URL: <https://git-scm.com/docs> (дата звернення: 04.05.2026).
26. Береза А. М., Краєва О. А., Краєв О. М. Інформаційні системи і технології. Київ : КНЕУ, 2017.
27. Морзе Н. В., Барна О. В. Основи вебтехнологій. Київ : Видавнича група BHV, 2018.
28. Кудрявцев Є. М. Бази даних. Проектування та використання. Київ : Каравела, 2020.
29. Avi Silberschatz, Henry F. Korth, S. Sudarshan. Database System Concepts. 7th Edition. McGraw-Hill, New York. 2020. 1344 p. <https://www.mpgcamb.com/wp-content/uploads/2024/12/Abraham-Silberschatz-Henry-F.-Korth-S.-Sudarshan-Database-System-Concepts-McGraw-Hill-Education-2019.pdf>
30. Що таке UI/UX дизайн. URL: <https://te.itstep.org/blog/ui-and-ux-design> (дата звернення 03.05.2026).
31. Веб-дизайнер – середня зарплата по Україні. <https://www.work.ua/salary> (дата звернення: 04.05.2026).

32. Тарифні плани. TheHost: Хостинг провайдер. URL:
<https://thehost.ua/ua/hosting/plans> (дата звернення: 04.05.2026).