МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: Розробка імерсивного вебресурсу для візуалізації даних ігрового всесвіту

Здобувача вищої освіти першого (бакалаврського) рівня,
IV курсу, групи ЦДБ-22-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Технології цифрового дизайну»

Руднєва Олексія Сергійовича
(підпис) (прізвище, ім'я, по батькові)

Науковий керівник Ратушняк Тетяна Володимирівна, канд. фіз.-мат. наук, доцент
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Керівник освітньо-професійної програми Ратушняк Тетяна Володимирівна канд. фіз.-мат. наук, доцент
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту за рішенням кафедри комп'ютерних та інформаційних технологій і систем протокол від « 12 » травня 2026 № 20

Завідувач кафедри Філоненко Михайло Миколайович, канд. фіз.-мат. наук, доцент
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Ірпінь – 2026

Реєстрація № 74 29.05.2026 _____
(дата) (підпис відповідальної особи)

Результати перевірки до захисту 12.05.26 _____
(до захисту, на доопрацювання) (дата) (підпис)

Результат захисту 4 (добре) 12.06.2026 _____
(оцінка) (дата) (підпис)

Голова комісії: _____ О.С. Дюбанов
(підпис)

Члени комісії: _____ М.М. Філоненко
(підпис)

_____ Т.В. Ратушняк
(підпис)

_____ В.О. Ніжегородцев
(підпис)

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії,секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	21		12.06.2026 Ратушняк Т.В. Дюбанов О.С.
2. Дотримання вимог щодо оформлення	від 0 до 10	8		Філоненко М.М. Ратушняк Т.В.
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	3		Ніжегородцев В.О.
4. Рецензент	від 0 до 5	4		Суховецька Л.П.
РАЗОМ БАЛІВ	від 0 до 50	36		

Оцінювання кваліфікаційної роботи науковим керівником – 36 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалоюЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
72	C	4 (добре)	12.06.2026	Дюбанов О.С. Філоненко М.М. Ратушняк Т.В. Ніжегородцев В.О. Суховецька Л.П.

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	3 (задовільно)
50-59	E	
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань: 12 «Інформаційні технології»
Спеціальність: 122 «Комп'ютерні науки»
Освітня програма: «Технології цифрового дизайну»

Затверджую
Завідувач кафедри _____ Філоненко М.М.
«01» грудня 2025

Завдання на кваліфікаційну роботу
здобувачу вищої освіти
Руднєву Олексію Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Розробка імерсивного вебресурсу для візуалізації даних ігрового всесвіту
Затверджена наказом ректора від «28» листопада 2025 № 1526
2. Строк здачі закінченої роботи «11» травня 2026
3. Мета роботи: проєктування, архітектурне обґрунтування та програмна реалізація вебпорталу, який забезпечує ефективне структурування, збереження та динамічне відображення інтерактивного ігрового контенту.
Об'єкт дослідження: процес інформаційного забезпечення ігрових проєктів засобами сучасних вебтехнологій.
Предмет дослідження: методи проєктування та програмної реалізації Вікі-порталу для науково-фантастичного ігрового всесвіту.
Орієнтовна база дослідження: науково-методична література в галузі системного аналізу вебресурсів, сучасних методологій UX/UI проєктування складних інтерфейсів, стандартів інклюзивності вебдизайну та кіберзахисту інформаційних систем, ігровий всесвіт науково-фантастичного шутера «Wings Of Fire: Війна світів по ту межу розлому».
4. Вихідні дані до роботи: ігровий всесвіт «Wings Of Fire: Війна світів по ту межу розлому», програмне забезпечення для розробки вебсайту.
5. Перелік графічного матеріалу: Трирівнева клієнт-серверна архітектура, Реляційна база даних проєкту, Запит та відображення статті користувачем, Захист розробленого вебресурсу.
6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом): Вступ;
Розділ 1. Аналіз предметної області та обґрунтування напрямів розробки;
Розділ 2. Проєктування та розробка цілісного макету інформаційного веб-порталу;
Розділ 3. Тестування та оптимізація програмного забезпечення;
Висновки;
Список використаних джерел.

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційноїроботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 30.10.2025	До 30.10.2025
2	**І розділ** (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	29.12.2025	29.12.2025
3	**ІІ розділ** (аналіз експериментальної частини про методики досліджень)	23.02.2026	23.02.2026
4	**ІІІ розділ** (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	20.04.2026	20.04.2026
5	**Висновки**	04.05.2026	04.05.2026
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	11.05.2026	11.05.2026
7	Отримання відгуку керівника	01.05.2026	01.05.2026
8	Отримання зовнішньої рецензії	04.06.2026	04.06.2026

8. Дата видачі завдання «01» грудня 2025

9. Керівник кваліфікаційної роботи _____ (підпис) Ратушняк Т.В. (прізвище, ініціали)

10. Керівник освітньо-професійної програми _____ (підпис) Ратушняк Т.В. (прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти _____ (підпис) Руднєв О.С. (прізвище, ініціали)

АНОТАЦІЯ

Розробка імерсивного вебресурсу для візуалізації даних ігрового всесвіту

Мета роботи: проєктування, архітектурне обґрунтування та програмна реалізація вебпорталу, який забезпечує ефективне структурування, збереження та динамічне відображення інтерактивного ігрового контенту

Завдання дослідження: дослідити концептуальні особливості вікі-технологій та проаналізувати існуючі рішення у сфері структурування ігрових баз знань; провести порівняльний аналіз сучасних технологій серверної та клієнтської розробки; спроєктувати та обґрунтувати трирівневу клієнт-серверну архітектуру інформаційного вебпорталу; розробити детальну специфікацію карти сторінок користувацького та адміністративного сегментів сайту з побудовою макетів інтерфейсу; робудувати логічну схему та нормалізовану структуру реляційної бази даних для збереження взаємопов'язаних інформаційних масивів ігрового лору; програмно реалізувати серверні модулі бізнес-логіки мовою PHP 8.x із використанням абстракції PDO та алгоритмів динамічного рендерингу об'єктів; розробити систему захисту вебдодатку від критичних уразливостей відповідно до стандартів безпеки; визначити методологію тестування, розробити сценарії верифікації та провести аналіз технічних метрик продуктивності й оптимізації створеного програмного забезпечення.

Об’єкт дослідження: процес інформаційного забезпечення ігрових проєктів засобами сучасних вебтехнологій.

Предмет дослідження: методи проєктування та програмної реалізації Вікі-порталу для науково-фантастичного ігрового всесвіту.

Методи дослідження: методи системного аналізу предметної області, принципи об'єктно-орієнтованого проєктування та програмування, методи нормалізації реляційних баз даних, архітектурне моделювання за допомогою UML-діаграм, методи тестування та аудиту вебдодатків на інформаційну безпеку.

База дослідження: науково-методична література в галузі системного аналізу вебресурсів, сучасних методологій UX/UI проєктування складних інтерфейсів, стандартів інклюзивності вебдизайну та кіберзахисту інформаційних систем; ігровий всесвіт науково-фантастичного шутера «Wings Of Fire: Війна світів по ту межу розлому».

Одержані результати: можуть бути використані для розширення інформаційної інфраструктури ігрового проєкту «Wings Of Fire: Війна світів по ту межу розлому» та офіційного релізу його бази знань для спільноти гравців.

Кваліфікаційна робота містить 47 сторінок, 4 рисунки, 8 таблиць, список використаних джерел із 40 найменувань.

Ключові слова: вебпортал, wiki-технології, клієнт-серверна архітектура, CMS, PHP 8.X, MYSQL, СУБД, база даних, UI/UX ДИЗАЙН, кібербезпека, ігровий всесвіт.

Рік виконання кваліфікаційної роботи: 2026.

Рік захисту роботи: 2026.

ABSTRACT

Development of an immersive web resource for visualizing game universe data

Research Tasks: to investigate the conceptual features of wiki technologies and analyze existing solutions in the field of gaming knowledge base structuring; to perform a comparative analysis of modern server-side and client-side development technologies; to design and justify a three-tier client-server architecture of the information web portal; to develop a detailed site map specification for user and administrative segments with the construction of interface layouts; to build a logical schema and a normalized structure of a relational database for storing interconnected data arrays of game lore; to programmatically implement backend business-logic modules using PHP 8.x, PDO abstraction, and dynamic object rendering algorithms; to develop a web application security system against critical vulnerabilities in compliance with security standards; to define a testing methodology, develop verification scenarios, and analyze the technical performance and optimization metrics of the developed software.

Object of Research: the process of information support for gaming projects using modern web technologies.

Subject of Research: methods of design and software implementation of a Wiki portal for a sci-fi game universe.

Research Methods: methods of system analysis of the domain area, principles of object-oriented design and programming, methods of relational database normalization, architectural modeling using UML diagrams, as well as web application testing and information security auditing methods.

Research Base: scientific and methodological literature in the field of web resource system analysis, modern UX/UI complex interface design methodologies, web design inclusivity standards, and information systems cyber defense; the game universe of the sci-fi shooter "Wings Of Fire: War of Worlds Beyond the Rift".

Obtained Results: can be used to expand the information infrastructure of the "Wings Of Fire: War of Worlds Beyond the Rift" gaming project and for the official release of its knowledge base for the player community.

The qualification thesis consists of: 47 pages, 4 figures, 8 tables, and a list of references including 40 items.

Keywords: web portal, wiki technologies, client-server architecture, cms, php 8.x, mysql, database, ui/ux design, cybersecurity, game universe.

Year of Thesis Completion: 2026. Year of Thesis Defense: 2026.

ЗМІСТ

ВСТУП 3
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ НАПРЯМІВ РОЗРОБКИ 5
1.1. Дослідження концепції Wiki-технологій в контексті структурування ігрових баз знань 5
1.2. Порівняльний аналіз технологій серверної та клієнтської розробки 8
1.3. Специфікація вимог до архітектури та безпеки веб-порталу 10
1.4. Обґрунтування трирівневої клієнт-серверної архітектури 11
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ЦІЛІСНОГО МАКЕТУ ІНФОРМАЦІЙНОГО ВЕБ-ПОРТАЛУ 14
2.1. Детальна специфікація карти сторінок (Макет інтерфейсу) 14
2.2. Проєктування схеми та структури реляційної бази даних 18
2.3. Покроковий алгоритм та логіка функціонування системи 22
2.4. Комплексна система захисту інформаційного веб-порталу 24
2.5. Програмна реалізація модулів серверної логіки (Backend) на PHP 8.x 30
2.6. Проєктування адаптивного інтерфейсу (Frontend) та CSS-архітектури 32
РОЗДІЛ 3. ТЕСТУВАННЯ ТА ОПТИМІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 37
3.1. Методологія тестування веб-системи та сценарії верифікації 37
3.2. Навантажувальне тестування та аналіз продуктивності 38
3.3. Оптимізація швидкодії: аналіз Core Web Vitals 38
3.4. Відладка та профілювання серверної частини 39
3.5. Розгортання системи та аналіз економічної ефективності 40
3.6. Економічне обґрунтування розробки 40
3.7. Перспективи розвитку та масштабування системи 42
ВИСНОВКИ 43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 46

ВСТУП

Актуальність теми. Сучасна ігрова індустрія характеризується створенням масштабних всесвітів, які потребують якісної інформаційної підтримки. Науково-фантастичні шутери, такі як «Wings Of Fire: Війна світів по ту межу розлому», генерують великі обсяги даних, які потребують структурування, оперативного оновлення та зручного доступу для користувачів. Відсутність централізованого ресурсу знижує рівень залученості гравців, а використання сторонніх платформ обмежує можливості кастомізації та безпеки даних. Тому проєктування власного веб-порталу на базі сучасних технологій є актуальним завданням.

Об’єкт дослідження: процес інформаційного забезпечення ігрових проєктів засобами сучасних веб-технологій.

Предмет дослідження: методи проєктування та програмної реалізації Wiki-порталу для науково-фантастичного ігрового всесвіту.

Мета роботи: проєктування, архітектурне обґрунтування та програмна реалізація функціонального, масштабованого і кібербезпечного вебпорталу, який забезпечує ефективне структурування, збереження та динамічне відображення інтерактивного ігрового контенту з високою швидкістю відгуку сервера.

Для досягнення поставленої мети необхідно вирішити такі завдання:

дослідити концептуальні особливості Wiki-технологій та проаналізувати існуючі рішення у сфері управління ігровим контентом;

провести порівняльний аналіз сучасних серверних технологій (зокрема PHP 8.x та Node.js) і обґрунтувати оптимальний стек для Read-Intensive архітектури;

спроєктувати трирівневу клієнт-серверну архітектуру (Three-Tier Architecture) інформаційної системи;

розробити логічну, нормалізовану структуру реляційної бази даних для збереження інформаційних масивів (користувачів, племен, персонажів, книг);

створити цілісний макет користувацького та адміністративного інтерфейсів порталу із забезпеченням адаптивності під різні типи пристроїв;

програмно реалізувати серверний функціонал (бізнес-логіку) системи мовою PHP 8.x із використанням сучасних безпекових стандартів;

розробити та впровадити комплексну систему захисту вебдодатку від критичних уразливостей (SQL-ін'єкцій, XSS-атак, фіксації сесій) відповідно до стандартів OWASP.

Методи дослідження. У роботі використано методи системного аналізу предметної області, принципи об'єктно-орієнтованого проєктування та програмування (ООП), методи нормалізації реляційних баз даних (1NF–3NF), архітектурне моделювання за допомогою UML-діаграм (діаграми класів, послідовності, ER-діаграми), а також методи тестування та аудиту вебдодатків на інформаційну безпеку.

Наукова новизна та практичне значення отриманих результатів. Практична цінність роботи полягає у створенні працездатного, швидкого та кіберзахищеного прототипу спеціалізованої системи управління контентом (CMS) для ігрової індустрії. Розроблена архітектура бази даних та модулі захисту інформації є універсальними і можуть бути масштабовані або інтегровані через API з іншими ігровими проєктами. Створений вебпортал безпосередньо вирішує проблему якісної інформаційної підтримки всесвіту «Wings Of Fire», підвищує рівень залученості ігрової спільноти та забезпечує повну автономність володіння контентом.

Структура роботи. Кваліфікаційна робота складається зі вступу, трьох взаємопов'язаних розділів, висновків, списку використаних джерел.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ НАПРЯМІВ РОЗРОБКИ

1.1. Дослідження концепції Wiki-технологій в контексті структурування ігрових баз знань

У сучасній індустрії розробки програмного забезпечення та комп’ютерних ігор особливе місце посідає проблема інформаційної підтримки користувачів. Складні ігрові проєкти, зокрема в жанрі науково-фантастичних шутерів, мають масштабні всесвіти (лори), які містять тисячі взаємопов’язаних об’єктів: персонажів, фракцій, описів технологій, зброї та хронологічних подій. Традиційні статичні веб-сайти не здатні забезпечити належний рівень гнучкості та швидкості оновлення контенту, що зумовлює необхідність автоматизації процесів контент-менеджменту за допомогою Wiki-технологій.

Технологія Вікі (Wiki) — це концепція створення та керування вебсайтами, яка дозволяє користувачам спільно редагувати вміст сторінок безпосередньо через веббраузер, використовуючи спрощену мову розмітки (або візуальний WYSIWYG-редактор).

Дослідження архітектурних та дизайнерських рішень для розробки спеціалізованих інформаційних Wiki-систем потребує глибокого аналізу наявної науково-методичної бази. Створення унікальних інформаційних платформ для ігрових всесвітів лежить на стику кількох дисциплін: системного аналізу, веб-інженерії, кібербезпеки та теорії проєктування інтерфейсів (UX/UI).

Сучасні системи, що базуються на технології Вікі, прийнято класифікувати за двома основними критеріями:

- сферою призначення (публічні та корпоративні);

- та технічною архітектурою (програмні рушії, на базі яких вони функціонують).

Класифікація за призначенням та типом доступу

Глобальні публічні проєкти (Public Wiki) орієнтовані на створення відкритих баз знань із залученням широкого кола користувачів (модель краудсорсингу). Серед них:

- Wikipedia (Вікіпедія): Найбільша у світі універсальна електронна енциклопедія, вміст якої спільно створюється та редагується волонтерами з усього світу;
- Fandom (раніше Wikia): Спеціалізована хостингова вікі-служба, призначена для створення фан-спільнотами тематичних баз знань про поп-культуру, кінематограф, відеоігри тощо (наприклад, Witcher Wiki, Marvel Database).

Корпоративні системи (Enterprise Wiki) - цез акриті внутрішні платформи організацій, призначені для керування корпоративними знаннями (Knowledge Management), збереження інтелектуального капіталу компанії та спільної роботи команд, зокрема:

- Confluence (від Atlassian): Галузевий стандарт в IT-індустрії для ведення технічної документації та ТЗ, що має глибоку інтеграцію з трекером завдань Jira;
- Notion: Сучасний мультифункціональний гібрид вікі-системи, таск-менеджера та реляційних таблиць даних, що базується на блочній структурі контенту.

Розглянемо архітектуру популярних Вікі-рушіїв (Wiki Engines).

Для розгортання власної вікі-системи в межах розробки програмного забезпечення використовуються готові програмні платформи. Нижче наведено порівняльний аналіз трьох ключових open-source рушіїв, що відрізняються своєю архітектурою та технологічним стеком (таблиця 1.1).

Таблиця 1.1

Архітектура популярних Вікі-рушіїв

Назва рушія	Технологічний стек	Архітектура збереження даних	Особливості та сфера застосування
MediaWiki	PHP + СУБД (MySQL / PostgreSQL)	Реляційна база даних з потужним рівнем кешування.	Базовий рушій Вікіпедії. Ідеально масштабується під гігантські обсяги даних, підтримує тисячі розширень, але складний в адмініструванні.
DokuWiki	PHP (без СУБД)	Файлова система (Flat Files). Усі статті зберігаються як .txt файли.	Надзвичайно легкий і швидкий рушій. Не потребує розгортання БД, що спрощує резервне копіювання. Оптимальний для документації невеликих проєктів.
TiddlyWiki	JavaScript	Single-Page Application (SPA). Увесь сайт, ядро та контент містяться в одному HTML-файлі.	Унікальна архітектура, де дані зберігаються у вигляді мікро-дописів (Tiddlers). Працює повністю на стороні клієнта, підходить для персональних баз знань.

Концепція Wiki базується на принципах відкритого, спільного або централізованого редагування гіпертекстового середовища. Основними перевагами використання Wiki-архітектури для проєктування бази знань є:

- Гнучка структура зв’язків: використання внутрішніх посилань (вікі-посилань) дозволяє побудувати мережеву, а не лінійну структуру даних, що ідеально відповідає архітектурі складного ігрового всесвіту.
- Поділ контенту на сутності: кожен елемент світу (наприклад, окремий персонаж чи книга) позиціонується як самостійна інформаційна одиниця (стаття) із власним набором метаданих.
- Простота масштабування: додавання нових категорій, сутностей або зв’язків між ними здійснюється через веб-інтерфейс і не вимагає перепроєктування бази даних або модифікації вихідного коду системи.

При розробці тематичного веб-ресурсу для науково-фантастичного шутера постає вибір між використанням готових комерційних платформ (наприклад, Fandom/Wikia) та розробкою власної унікальної системи управління контентом (Content Management System, CMS). Проведений аналіз показав, що готові платформи мають суттєві недоліки: надмірна кількість сторонньої реклами, відсутність можливості глибокої кастомізації інтерфейсу під специфічний Sci-Fi дизайн проєкту, обмежений контроль над безпекою даних та неможливість інтеграції з внутрішніми ігровими серверами через API.

Таким чином, розробка власного цілісного макету веб-порталу на базі зв’язки PHP та MySQL є найбільш доцільним та технічно обґрунтованим рішенням.

1.2. Порівняльний аналіз технологій серверної та клієнтської розробки

Ефективність функціонування інформаційного порталу безпосередньо залежить від правильно обраного технологічного стека. Основним критерієм вибору є забезпечення високої швидкості обробки запитів, безпека сховища даних та можливість швидкої реалізації макету сторінок.

Для реалізації серверної логіки було проведено порівняльний аналіз двох провідних платформ: PHP та Node.js (JavaScript Runtime) (таблиця 1.2).

Таблиця 1.2

Порівняння бекенд-технологій: PHP 8.x проти Node.js

Критерій порівняння	**Платформа PHP 8.x**	**Платформа Node.js (Express)**
Швидкість розробки	Висока, завдяки вбудованим інструментам веб-орієнтованості та простоті роботи з СУБД.	Середня, вимагає написання великої кількості базового коду та архітектурного проєктування з нуля.
Робота з реляційними БД	Ідеальна сумісність із MySQL через драйвер PDO, оптимізований під багатопотокові запити.	Потребує додаткових ORM-прошарків (Sequelizer), що підвищує споживання оперативної пам’яті.
Хостинг та розгортання	Підтримується 99% сучасних серверів, мінімальні вимоги до системних ресурсів.	Вимагає налаштування VPS/VDS та постійного контролю за витоками пам’яті процесу Node.
Безпека «з коробки»	Наявність вбудованих функцій фільтрації, хешування паролів та екранування символів.	Залежність від сторонніх npm-пакетів, які часто містять критичні вразливості.

Враховуючи специфіку дипломного проєкту, де пріоритетом є побудова чіткого макету, надійної реляційної структури баз даних та реалізація системи захисту без надмірного ускладнення серверної архітектури, як мову бекенду було обрано PHP 8.x.

Обґрунтування вибору СУБД: MySQL

Для збереження інформаційних масивів розглядалися реляційні СУБД MySQL та PostgreSQL. Вибір було зроблено на користь **MySQL** з наступних причин:

1. Оптимізація під операції читання (Read-Intensive). Оскільки Wiki-портал за своєю природою частіше віддає контент користувачам (операції SELECT), ніж приймає зміни від адміна, рушій InnoDB у MySQL демонструє максимальну швидкодію завдяки ефективному індексуванню текстових полів.

2. Низький поріг входження та сумісність. Інтеграція PHP + MySQL є класичним галузевим стандартом, що забезпечує стабільність зв’язку на рівні драйверів веб-сервера.

1.3. Специфікація вимог до архітектури та безпеки веб-порталу

На основі проведеного аналізу сформовано технічні вимоги до проєктування макету інформаційної системи:

- Вимоги до архітектури. Система має бути побудована за архітектурним патерном, що розділяє інтерфейс користувача та базу даних за допомогою серверного логічного прошарку (трирівнева модель), забезпечуючи повну автономність кожного компонента;
- Вимоги до макету інтерфейсу. Шаблон сайту повинен містити не менше 12–15 сторінок, включаючи розділені зони доступу (публічна зона для користувачів та захищена адмін-панель для контент-менеджерів). Інтерфейс має адаптуватися під мобільні та десктопні платформи за допомогою адаптивної верстки;

- Вимоги до безпеки. Система повинна повністю нівелювати вектори атак типу SQL-ін’єкцій, міжсайтового скриптингу (XSS) та несанкціонованого доступу до адміністративного інтерфейсу шляхом перехоплення сесій.

1.4. Обґрунтування трирівневої клієнт-серверної архітектури

Сучасні інформаційні системи та веб-портали, що обслуговують динамічний контент із високою частотою запитів, потребують використання архітектурних рішень, які забезпечують модульність, масштабованість та стійкість до відмов. Для розробки веб-порталу науково-фантастичного шутера було обрано класичну трирівневу клієнт-серверну архітектуру (Three-Tier Architecture). Дана модель передбачає чітке розділення системи на три незалежні рівні, кожен з яких виконує суворо визначені функції та взаємодіє з іншими рівнями за допомогою стандартизованих протоколів передачі даних.



Рис. 1.1. Трирівнева клієнт-серверна архітектура

На рисунку 1.1 продемостровано три рівні клієнт-серверної архітектури, зокрема:

1. Рівень представлення (Presentation Tier / Frontend)

Клієнтська частина системи реалізована на стороні веб-браузера користувача. Головним завданням цього рівня є візуалізація даних, отриманих від сервера, та забезпечення інтерфейсу взаємодії (UI/UX). Технологічний стек включає мову розмітки HTML5 для формування семантичної структури сторінок, каскадні таблиці стилів CSS3 для реалізації адаптивного ігрового Sci-Fi дизайну, та мову програмування JavaScript (ES6+) для організації динамічних елементів без перезавантаження сторінок. Клієнтська частина не має прямого доступу до бази даних і взаємодіє з сервером виключно за допомогою асинхронних або синхронних HTTP/HTTPS запитів;

2. Рівень додатку (Application Tier / Logic / Backend)

Серверна частина системи є центральним обчислювальним ядром веб-порталу. Як технологічна основа обрана мова програмування PHP 8.x, яка функціонує на базі веб-сервера (наприклад, Apache або Nginx). Цей рівень приймає запити від рівня представлення, виконує логічну обробку даних, перевіряє права доступу користувачів (аутентифікація та авторизація), формує запити до системи управління базами даних, інтерпретує отримані результати та генерує відповідь у форматі HTML або JSON для передачі назад на клієнтську частину;

3. Рівень збереження даних (Data Tier / Database)

Нижній рівень архітектури представлений реляційною системою управління базами даних (СУБД) MySQL. Він відповідає за фізичне зберігання, індексацію, цілісність та безпеку всієї інформаційної структури сайту (статті, профілі персонажів, конфігураційні дані користувачів та адміністраторів). Взаємодія між рівнем додатку (PHP) та рівнем даних (MySQL) здійснюється за допомогою спеціалізованого

інтерфейсу прикладного програмування PDO (PHP Data Objects), що дозволяє абстрагувати роботу з БД та підвищити рівень захищеності додатку.

Переваги обраної архітектурної моделі:

- Ізольованість компонентів: Зміна візуального оформлення (фронтенду) ніяк не впливає на логіку обробки даних на сервері або на структуру таблиць у базі даних.
- Масштабованість: При зростанні навантаження рівень додатку та рівень збереження даних можуть бути рознесені на різні фізичні або віртуальні сервери без модифікації коду програми.
- Безпека: Клієнт ніколи не взаємодіє з базою даних безпосередньо, що виключає можливість прямого компрометування даних при збої на стороні користувача.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ЦІЛІСНОГО МАКЕТУ ІНФОРМАЦІЙНОГО ВЕБ-ПОРТАЛУ

2.1. Детальна специфікація карти сторінок (Макет інтерфейсу)

Спроєктуємо макет вебпорталу фентезійного світу. Потрібна цілісна інформаційна структура, що складається з 12–15 взаємопов'язаних сторінок. Кожна сторінка виконує свою роль у загальній системі навігації та відображення контенту. Нижче наведено детальну специфікацію структури кожної сторінки з описом її функціонального призначення та логічних зв'язків.

1. Публічний сегмент порталу

1.а. Головна сторінка (index.html / index.php)

Центральний вузол веб-порталу. Призначена для первинного ознайомлення користувача з ресурсом.

- Вхідні параметри: Відсутні (статичний рендеринг із елементами динамічного виведення останніх оновлень).
- Елементи інтерфейсу: Головне навігаційне меню (шапка сайту), банер у стилі Sci-Fi (Hero-section), блок інтерактивних карток швидкого переходу до основних розділів, підвал (footer) із правовою інформацією.
- Зв'язки: Містить прямі посилання на всі основні публічні розділи сайту та форму авторизації адміністратора.

1.б. Розділ «Персонажі» (characters.php)

Сторінка агрегації даних, яка відображає повний перелік діючих суб'єктів всесвіту (героїв, юнітів).

- Вхідні параметри: GET-запити для фільтрації та сортування (наприклад, фракція, статус).

- Елементи інтерфейсу: Сітка (Grid) з мініатюрами персонажів, панель фільтрів, динамічний рядок пошуку.
- Зв’язки: Кожна картка персонажа містить унікальне посилання типу character_page.php?id=X, де X — унікальний ідентифікатор із бази даних.

1.в. Розділ «Племена» (tribes.php)

Спеціалізована сторінка класифікації фракцій та угруповань всесвіту науково-фантастичного шутера.

- Вхідні параметри: Відсутні (виведення повного списку з БД).
- Елементи інтерфейсу: Список текстово-графічних блоків, що описують технологічні та біологічні особливості кожної фракції.
- Зв’язки: Перенаправляє користувача на сторінки конкретних племен (tribe_page.php?id=X).

1.г. Розділ «Світ» (world.php)

Інформаційно-картографічний розділ, який відповідає за географічне та просторове орієнтування користувача в ігровому сеттингу (материки Піррія та Пантала в умовах техномагічного розлому).

- Вхідні параметри: Ідентифікатор локації або регіону.
- Елементи інтерфейсу: Інтерактивна карта (зони, що реагують на наведення курсору за допомогою CSS/JS), опис кліматичних, стратегічних та фортифікаційних особливостей зон.
- Зв’язки: Переходи на сторінки описів конкретних локацій або битв.

1.д. Розділ «Книги» (books.php)

Каталог літературних або хронологічних джерел, на яких базується лор ігрового всесвіту.

- Вхідні параметри: Сортування за роком видання або хронологічним порядком подій.

- Елементи інтерфейсу: Список обкладинок, короткі анотації, таблиця метаданих (автор, рік, цикл).
- Зв’язки: Посилання на book_page.php?id=X.

1.е. Сторінка «Пошук» (search.php)

Універсальний інструмент для знаходження текстової інформації по всьому сайту.

- Вхідні параметри: Текстовий рядок із поля вводу (POST або GET запит search_query).
- Елементи інтерфейсу: Поле введення з підтримкою автодоповнення, список результатів із релевантністю, пагінація сторінок.
- Зв’язки: Динамічні посилання на знайдені сторінки об’єктів.

2. Динамічні сторінки об’єктів (Шаблони рендерингу)

2.а. Шаблон сторінки персонажа (character_page.php)

Динамічний шаблон, структура якого заповнюється на основі даних з конкретного рядка таблиці characters.

- Вхідні параметри: id (integer) у GET-запиті.
- Елементи інтерфейсу: Велике зображення персонажа, блок основних характеристик (таблична структура), розгорнута текстова біографія, список пов’язаних локацій та фракцій.

2.б. Шаблон сторінки племені (tribe_page.php)

Сторінка детального опису фракції.

- Вхідні параметри: id фракції у URL.
- Елементи інтерфейсу: Герб/логотип фракції, опис технологічного рівня, перелік персонажів, які належать до цього племені (динамічна вибірка з БД за допомогою зв’язку Foreign Key).

2.в. Шаблон сторінки книги (book_page.php)

Детальна картка тому або хроніки.

- Вхідні параметри: id книги.
- Елементи інтерфейсу: Повна анотація, список розділів, перелік ключових подій, що відбулися в цій частині лору.

3. Адміністративний сегмент порталу

3.а. Сторінка входу адміністратора (login.php)

Форма автентифікації для отримання доступу до функцій керування вмістом порталу.

- Вхідні параметри: Логін та пароль користувача, що передаються виключно методом POST через шифроване з’єднання.
- Елементи інтерфейсу: Поля введення login та password, кнопка відправки форми, блок виведення помилок валідації.

3.б. Панель управління / Дашборд (admin/dashboard.php)

Головне робоче місце адміністратора сайту після успішного проходження перевірки прав.

- Вхідні параметри: Перевірка активної сесії користувача на сервері (session_start()).
- Елементи інтерфейсу: Статистика бази даних (кількість статей, користувачів), список останніх дій, навігаційне меню адміна.

3.в. Сторінка додавання та редагування статті (admin/edit_article.php)

Універсальний інтерфейс для виконання операцій створення та модифікації контенту.

- Вхідні параметри: Для додавання — відсутні; для редагування — action=edit&id=X.

- Елементи інтерфейсу: Форма редагування, текстові поля (Input) для назв, область великого тексту (Textarea) або WYSIWYG-редактор для тіла статті, випадаючий список (Select) для вибору категорії/племені, елемент завантаження файлу зображення (File input).

2.2. Проєктування схеми та структури реляційної бази даних

Для забезпечення надійного зберігання, швидкого пошуку та модифікації інформації розроблено логічну структуру реляційної бази даних у СУБД MySQL. База даних складається з 4-х основних взаємопов’язаних таблиць. Цілісність даних підтримується на рівні декларативних обмежень зовнішніх ключів (Foreign Keys) з використанням рушія зберігання даних InnoDB, який підтримує транзакційність (ACID).

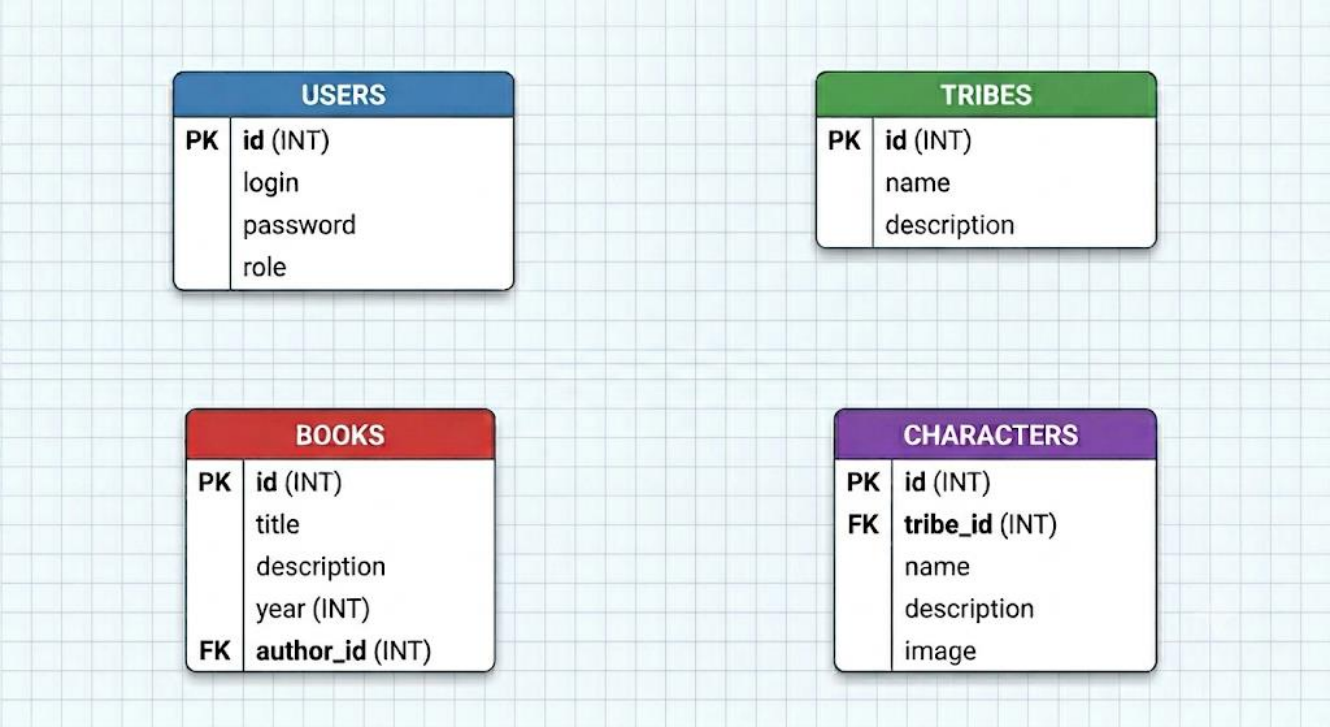


Рис. 2.1. Реляційна база даних проєкту

Таблиця 2.1 (USERS) призначена для зберігання облікових даних осіб, що мають доступ до системи з різними рівнями привілеїв.

Таблиця 2.1.

Облік користувачів та адміністраторів

Назва поля	**Тип даних**	**Атрибути**	**Опис призначення**
id	INT	PRIMARY KEY, AUTO_INCREMENT	Унікальний ідентифікатор користувача в системі.
login	VARCHAR(50)	UNIQUE, NOT NULL	Унікальне ім’я (логін) для авторизації.
password	VARCHAR(255)	NOT NULL	Хеш-значення пароля користувача.

role	VARCHAR(20)	NOT NULL, DEFAULT ‘user’	Роль у системі для розмежування прав (admin, moderator).

Таблиця 2.2 (TRIBES) – це Довідник фракцій / племен, який є батьківською таблицею по відношенню до таблиці персонажів, реалізуючи зв’язок типу «один-до-багатьох».

Таблиця 2.2

Назва поля	Тип даних	Атрибути	Опис призначення
id	INT	PRIMARY KEY, AUTO_INCREMENT	Унікальний код племені/фракції.
name	VARCHAR(100)	NOT NULL	Офіційна назва фракції.
description	TEXT	NOT NULL	Повний текстовий опис технологій, особливостей та лору.

Таблиця 2.3 (CHARACTERS) (Реєстр персонажів та юнітів) – це основна інформаційна таблиця, яка містить детальні відомості про сутності всесвіту.

Таблиця 2.3

Назва поля	Тип даних	Атрибути	Опис призначення
id	INT	PRIMARY KEY, AUTO_INCREMENT	Унікальний код персонажа.

tribe_id	INT	FOREIGN KEY, NOT NULL	Посилання на id з таблиці tribes (приналежність до фракції).
name	VARCHAR(100)	NOT NULL	Ім’я або кодове позначення юніта.
description	TEXT	NOT NULL	Текстове досьє, біографія, ТТХ.
image	VARCHAR(255)	DEFAULT NULL	Шлях до файлу зображення на сервері (наприклад, /uploads/char1.png).

Обмеження цілісності: FOREIGN KEY (tribe_id) REFERENCES tribes(id) ON DELETE RESTRICT ON UPDATE CASCADE. Це запобігає видаленню племені, якщо до нього прив’язаний хоча б один персонаж.

Таблиця 2.4 (BOOKS) – це Хронологічні джерела. Зберігає інформацію про першоджерела лору.

Таблиця 2.4

Назва поля	Тип даних	Атрибути	Опис призначення
id	INT	PRIMARY KEY, AUTO_INCREMENT	Ідентифікатор книги.
title	VARCHAR(150)	NOT NULL	Назва тому або хроніки.
description	TEXT	NOT NULL	Короткий зміст, ключові події.

year	INT	DEFAULT NULL	Рік публікації/випуску в межах всесвіту або реальний.

2.3. Покроковий алгоритм та логіка функціонування системи

Взаємодія компонентів системи при виконанні користувацьких та адміністративних сценаріїв підпорядкована чітким алгоритмам. Розглянемо два базові процеси: перегляд статті користувачем (динамічний рендеринг сторінки об’єкта) та редагування контенту адміністратором.

Сценарій 1: Запит та відображення статті користувачем (рис. 2.2)

- Ініціалізація запиту: Користувач переходить за посиланням або надсилає форму пошуку. Браузер генерує HTTP GET запит на сервер, наприклад: http://portal/character_page.php?id=5.
- Маршрутизація та перевірка вхідних даних: Серверний скрипт character_page.php приймає параметр id. Відбувається первинна валідація: перевірка, чи є параметр цілим числом. Якщо параметр некоректний, сервер негайно повертає HTTP-код відповіді 400 (Bad Request).
- Взаємодія з базою даних: Перевірений ідентифікатор передається в безпечний параметризований SQL-запит до СУБД MySQL: SELECT * FROM characters WHERE id = :id.
- Обробка результату СУБД: MySQL виконує пошук по індексованому полю первинного ключа та повертає масив даних (або порожній результат, якщо об’єкт не знайдено, що спричиняє генерацію сервером сторінки помилки 404 Not Found).
- Динамічна збірка сторінки (Рендеринг): PHP-скрипт підставляє отримані текстові дані та шлях до зображення у відповідні блоки HTML-шаблону сторінки.

Одночасно застосовуються функції екранування тексту для безпечного виведення контенту.

- Відповідь клієнту: Сформований HTML-код разом із кодом відповіді 200 OK відправляється через веб-сервер назад у браузер користувача для фінальної візуалізації.

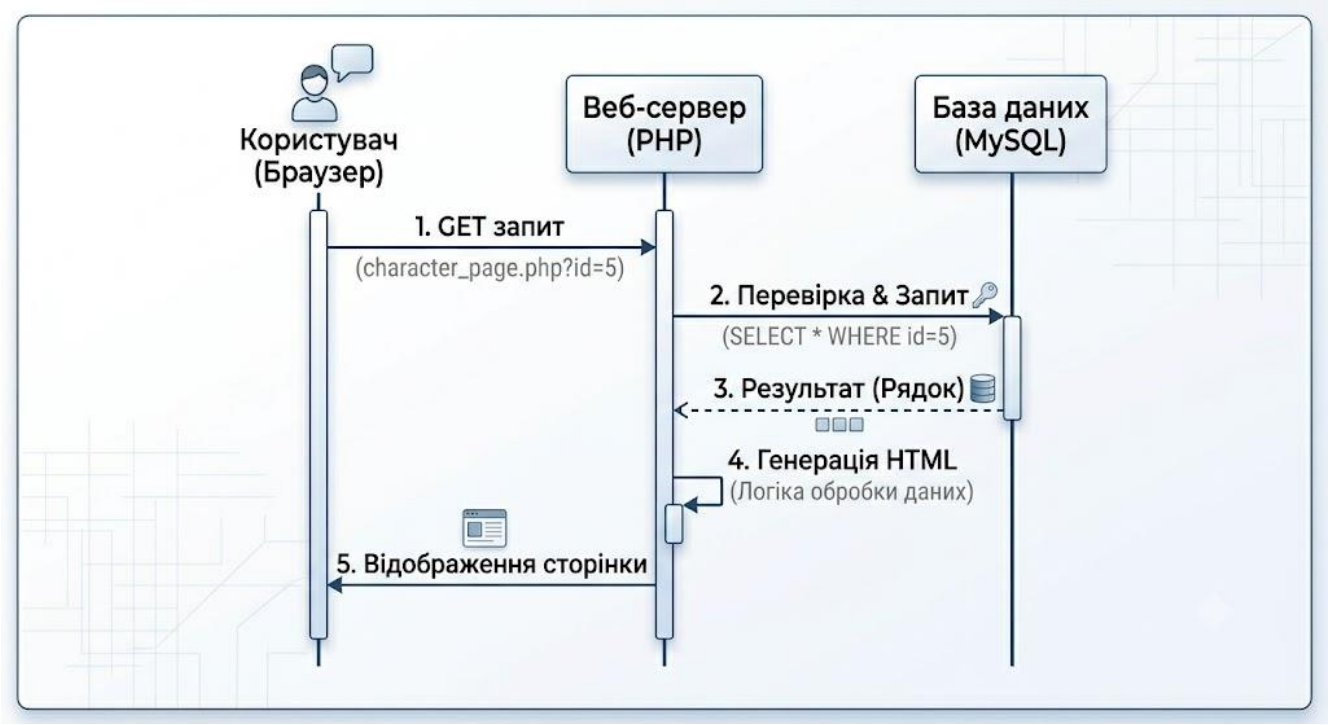


Рис. 2.2. Запит та відображення статті користувачем

Сценарій 2: Модифікація даних адміністратором

- Перевірка прав доступу: При спробі доступу до скрипту edit_article.php сервер перевіряє глобальний масив сесій $_SESSION[‘user_role’]. Якщо значення не дорівнює admin, виконання скрипту переривається, і користувач перенаправляється на сторінку login.php.

- Введення та відправка даних: Адміністратор змінює текст у полях форми та натискає «Зберегти». Дані відправляються методом POST.

- Валідація на сервері: Перевіряється заповненість обов’язкових полів, відповідність типів даних, а також виконується аналіз завантажуваного медіафайлу (перевірка MIME-типу та розміру картинки).
- Оновлення сховища: Сервер виконує SQL-команду UPDATE characters SET name = :name, description = :desc WHERE id = :id.
- Скидання кешу та перенаправлення: Після успішного завершення транзакції в БД адміністратор отримує повідомлення про успішне збереження і перенаправляється на оновлену сторінку об’єкта для перегляду результату.

2.4. Комплексна система захисту інформаційного веб-порталу

Безпека веб-порталу є пріоритетним завданням, оскільки архітектура передбачає наявність адміністративного інтерфейсу та взаємодію з базою даних, що робить додаток потенційною ціллю для кібератак. Захист розробленого ресурсу спроєктовано як ешелоновану систему, яка нейтралізує найпоширеніші вразливості веб-додатків, що входять до рейтингу OWASP Top 10 (рис. 2.3).



Рис. 2.3. Запит та відображення статті користувачем

Механізми автентифікації та захисту сесій адміністратора

Адміністративна панель сайту (login.php, admin/dashboard.php) є критичною точкою доступу. Для запобігання несанкціонованому проникненню, підбору паролів та перехопленню сесій реалізовано наступний комплекс заходів захисту.

1. Криптографічний захист паролів

Зберігання паролів у базі даних у відкритому текстовому вигляді суворо заборонено. При реєстрації або зміні пароля адміністратора система використовує вбудовану функцію PHP password_hash() з алгоритмом **Argon2id** (або альтернативним стійким алгоритмом PASSWORD_BCRYPT).

- Алгоритм автоматично генерує унікальну криптографічну сіль для кожного пароля та виконує ресурсомістке обчислення хешу. Це повністю нівелює ефективність атак із використанням заздалегідь обчислених таблиць (Rainbow Tables).
- Перевірка пароля при вході здійснюється функцією password_verify($password, $hashed_password), яка стійка до атак за часом (Timing Attacks).

2. Безпечне керування сесіями (Session Management)

Після успішної автентифікації сервер ініціює сесію користувача. Для протидії атакам типу Session Fixation (фіксація сесії) та Session Hijacking (викрадення сесії) впроваджено такі правила:

- Регенерація ідентифікатора: При кожній зміні статусу авторизації (вхід, вихід) або через певний проміжок часу примусово викликається функція session_regenerate_id(true). Вона знищує старий файл сесії та створює новий з абсолютно іншим випадковим ключем, роблячи перехоплені раніше ідентифікатори недійсними.
- Прапорці безпеки кукі сесії: Конфігурація сесій у файлі php.ini або безпосередньо через скрипт за допомогою session_set_cookie_params() встановлює наступні параметри:
  - session.cookie_httponly = 1: Прапорець HttpOnly повністю забороняє доступ до кукі сесії через клієнтські скрипти JavaScript. Це захищає сесію від викрадення в разі наявності XSS-вразливості на сайті.
  - session.cookie_secure = 1: Прапорець Secure вказує браузеру передавати кукі виключно через захищене з’єднання HTTPS, запобігаючи перехопленню у відкритих Wi-Fi мережах.
  - session.use_strict_mode = 1: Забороняє використання ідентифікаторів сесій, які були ініційовані на стороні клієнта, а не згенеровані сервером.

3. Захист від SQL-ін’єкцій (SQL Injection) на рівні сховища даних

SQL-ін’єкція — одна з найнебезпечніших вразливостей, яка виникає, коли неперевірені дані користувача конкатенуються безпосередньо в тіло SQL-запиту. Це дозволяє зловмиснику змінити логіку виконання запиту, отримати доступ до закритих таблиць (наприклад, users), видалити або модифікувати базу даних.

Для повного виключення цієї вразливості в розробленому веб-порталі взаємодія з базою даних MySQL реалізована виключно через технологію підготовлених запитів (Prepared Statements) за допомогою інтерфейсу PDO.

Механізм роботи підготовлених запитів:

Замість прямої передачі змінних у запит, розробник формує шаблон запиту із заповнювачами (named placeholders):

```
// БЕЗПЕЧНИЙ ВАРІАНТ РЕАЛІЗАЦІЇ НА ПОРТАЛІ
$id = $_GET[‘id’];
$sql = "SELECT id, name, description, image FROM characters WHERE id = :id";
$stmt = $pdo->prepare($sql);
$stmt->execute([‘id’ => $id]);
$character = $stmt->fetch();
```

При використанні методу prepare(), веб-сервер відправляє структуру SQL-команди в СУБД MySQL окремо від самих даних. База даних компілює та оптимізує цей запит, чітко визначаючи, які частини є інструкціями мови SQL, а які — параметрами.

Коли згодом викликається метод execute(), параметри передаються окремо. Навіть якщо зловмисник передасть у змінну id рядок типу 5; DROP TABLE users;,

СУБД сприйме цей рядок виключно як цілісне текстове або числове значення ідентифікатора персонажа, а не як нову SQL-команду. Це дає 100% гарантію захисту від SQL-ін’єкцій.

4. Захист від міжсайтового скриптингу (XSS) та безпека файлової системи

- Нейтралізація атак типу XSS (Cross-Site Scripting)

Вразливість XSS виникає, коли шкідливий JavaScript-код, введений зловмисником (наприклад, у полі опису персонажа під час редагування статті), зберігається в базі даних та відображається іншим користувачам сайту. Цей скрипт може виконатися в браузері жертви та передати її конфіденційні дані (або токени доступу) зловмиснику.

На порталі реалізовано концепцію стратегії захисту на виводі (Context-Aware Output Encoding). Будь-які динамічні дані, отримані з бази даних перед рендерингом в HTML-код сторінки, обов’язково проходять обробку функцією htmlspecialchars():

```
// Захищений вивід імені персонажа у шаблон сторінки
echo htmlspecialchars($character[‘name’], ENT_QUOTES, ‘UTF-8’);
```

Ця функція перетворює спеціальні символи HTML, такі як <, >, &, ", ‘, на їхні безпечні текстові еквіваленти (HTML-сутності: <, >, &, ", '). Таким чином, якщо в БД було збережено рядок <script>alert(‘hack’)</script>, браузер не стане виконувати цей код, а просто відобразить його як звичайний текст на екрані, повністю нівелюючи загрозу атаки.

5. Безпека завантаження файлів (File Upload Security)

Оскільки адміністратор має можливість завантажувати зображення для карток персонажів та племен, існує ризик завантаження шкідливого виконуваного файлу (наприклад, веб-оболонки shell.php). Якщо зловмисник завантажить такий файл у

публічну папку сайту, він зможе виконати будь-який код на сервері та отримати повний контроль над системою.

Для запобігання цьому сценарію на порталі впроваджено багатоступеневий алгоритм валідації файлів при завантаженні:

- Валідація розширення за білим списком (Whitelist): Система дозволяє завантаження файлів виключно з розширеннями .jpg, .jpeg, .png. Будь-які інші розширення (особливо .php, .phtml, .exe, .sh) блокуються на етапі первинного аналізу імені файлу за допомогою функції pathinfo().
- Перевірка реального MIME-типу файлу: Зловмисник може перейменувати файл вірусу з shell.php на image.png. Для протидії цьому сервер аналізує вміст файлу за допомогою вбудованого модуля finfo_file(), який визначає реальну структуру файлу незалежно від його імені. Дозволеними є лише типи image/jpeg та image/png.
- Рандомізація імен файлів: При збереженні файлу на диску його оригінальне ім’я повністю знищується. Нове ім’я генерується за допомогою функції хешування (наприклад, md5(uniqid(rand(), true)) . ‘.’ . $extension). Це унеможливлює спроби прямого звернення до завантаженого файлу за його вихідним ім’ям та вирішує проблему перезапису файлів із однаковими іменами.
- Обмеження прав виконання: Папка для завантаження медіафайлів (/uploads/) налаштовується на рівні веб-сервера (через файл конфігурації .htaccess для Apache або конфігурацію Nginx) таким чином, щоб у ній було повністю заборонено виконання будь-яких скриптів, навіть якщо вони туди потраплять.

Блокування виконання скриптів у папці завантажень (.htaccess)

```
RemoveHandler .phtml .php .php3 .php4 .php5 .php6 .php7 .php8 .pls .pl .py .cgi .asp .aspx
options –ExecCGI
```

Впровадження даного комплексу заходів безпеки гарантує стабільну та захищену роботу інформаційного порталу науково-фантастичного шутера, мінімізуючи ризики успішного проведення кібератак і забезпечуючи повну відповідність розробленого макету сучасним критеріям надійності веб-додатків.

2.5. Програмна реалізація модулів серверної логіки (Backend) на PHP 8.x

У межах трирівневої клієнт-серверної архітектури рівень додатка є зв'язуючою ланкою, що обробляє бізнес-логіку системи. Нижче наведено базові програмні модулі, реалізовані мовою PHP 8.x з використанням об'єктно-орієнтованого підходу та драйвера PDO.

1. Клас конфігурації та підключення до бази даних (Database.php)

Для оптимізації роботи з СУБД MySQL та запобігання повторному відкриттю з'єднань реалізовано клас, що використовує патерн Singleton.

```
PHP
<?php
class Database {
    private static $instance = null;
    private $pdo;
    private function __construct() {
        $host = 'localhost';
        $db   = 'game_wiki_db';
        $user = 'wiki_admin';
        $pass = 'Secure_Password_2026';
        $charset = 'utf8mb4';
        $dsn = "mysql:host=$host;dbname=$db;charset=$charset";
        $options = [
            PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
            PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
            PDO::ATTR_EMULATE_PREPARES   => false,
        ];
        try {
            $this->pdo = new PDO($dsn, $user, $pass, $options);
        } catch (\PDOException $e) {
            throw new \PDOException($e->getMessage(), (int)$e-
>getCode());
        }
    }
    public static function getInstance() {
        if (self::$instance === null) {
            self::$instance = new Database();
        }
        return self::$instance;
    }
    public function getConnection() {
        return $this->pdo;
    }}?>
```

2. Клас управління сутностями персонажів (CharacterManager.php)

Цей клас реалізує базові операції CRUD для таблиці CHARACTERS, забезпечуючи безпеку даних за допомогою підготовлених запитів (Prepared Statements).

```
PHP <?php
require_once 'Database.php';
class CharacterManager {
    private $db;
    public function __construct() {
        $this->db = Database::getInstance()->getConnection();
    }

    // Метод для динамічного рендерингу сторінки конкретного персонажа
    public function getCharacterById($id) {
        $sql = "SELECT c.*, t.name as tribe_name
                FROM characters c
                JOIN tribes t ON c.tribe_id = t.id
                WHERE c.id = :id";

        $stmt = $this->db->prepare($sql);
        $stmt->execute(['id' => (int)$id]);
        return $stmt->fetch();
    }

    // Метод додавання нового персонажа через адмін-панель
    public function createCharacter($tribe_id, $name, $description,
$image_path) {
        $sql = "INSERT INTO characters (tribe_id, name, description,
image)
                VALUES (:tribe_id, :name, :description, :image)";

        $stmt = $this->db->prepare($sql);
        return $stmt->execute([
            'tribe_id'    => (int)$tribe_id,
            'name'        => htmlspecialchars(trim($name), ENT_QUOTES,
'UTF-8'),
            'description' => htmlspecialchars(trim($description),
ENT_QUOTES, 'UTF-8'),
            'image'       => $image_path
        ]);
    }
}
?>
```

2.6. Проектування адаптивного інтерфейсу (Frontend) та CSS-архітектури

Рівень представлення (Frontend) розроблено з урахуванням специфіки ігрового всесвіту «Wings Of Fire: Війна світів по ту межу розлому». Веб-інтерфейс має

відображати атмосферу техномагічного розлому та науково-фантастичного шутера, що вимагає використання специфічної палітри кольорів (темний фон, неонові елементи підсвічування акцентів, кібернетичні рамки) та адаптивної верстки під будь-які типи пристроїв.

1. Архітектура CSS-стилів (style.css)

Для створення гнучкої та масштабованої системи стилів використано CSS Custom Properties (змінні). Це дозволяє централізовано керувати колірною схемою та шрифтами сайту.

```
CSS
:root {
    --bg-dark: #0a0c10;
    --bg-surface: #141822;
    --accent-neon-blue: #00f3ff;
    --accent-glow-green: #39ff14;
    --text-primary: #e2e8f0;
    --text-muted: #94a3b8;
    --border-cyber: 1px solid #1e293b;
    --font-sci-fi: 'Orbitron', 'Segoe UI', sans-serif;
}

body {
    background-color: var(--bg-dark);
    color: var(--text-primary);
    font-family: var(--font-sci-fi);
    margin: 0;
    padding: 0;
}

/* Організація адаптивної сітки для розділу "Персонажі" */
.character-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(280px, 1fr));
    gap: 24px;
    padding: 20px;
}

.character-card {
    background-color: var(--bg-surface);
    border: var(--border-cyber);
```

```
    border-radius: 8px;
    overflow: hidden;
    transition: transform 0.3s ease, box-shadow 0.3s ease;
}

.character-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 0 15px var(--accent-neon-blue);
    border-color: var(--accent-neon-blue);
}

.character-card img {
    width: 100%;
    height: 200px;
    object-fit: cover;
}

.character-info {
    padding: 16px;
}

.character-info h3 {
    margin: 0 0 8px 0;
    color: var(--accent-neon-blue);
}
```

2. Реалізація семантичного HTML-шаблону сторінки об'єкта (character_template.php)

Застосування семантичної розмітки є критично важливим для доступності (Accessibility) та пошукової оптимізації (SEO).

```
PHP
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title><?php echo htmlspecialchars($character['name'], ENT_QUOTES,
'UTF-8'); ?> - Wiki Wings Of Fire</title>
    <link rel="stylesheet" href="css/style.css">
</head>
```

```
<body>
    <header>
        <nav aria-label="Головна навігація">
            <a href="index.php" class="logo">WOF Wiki</a>
            <ul>
                <li><a href="characters.php"
class="active">Персонажі</a></li>
                <li><a href="tribes.php">Племена</a></li>
                <li><a href="world.php">Світ</a></li>
                <li><a href="books.php">Книги</a></li>
            </ul>
        </nav>
    </header>


    <main class="container">
        <article class="profile-layout">
            <header class="profile-header">
                <h1><?php echo htmlspecialchars($character['name'],
ENT_QUOTES, 'UTF-8'); ?></h1>
                <span class="badge"><?php echo
htmlspecialchars($character['tribe_name'], ENT_QUOTES, 'UTF-8');
?></span>
            </header>


            <div class="profile-media">
                <img src="<?php echo
htmlspecialchars($character['image'], ENT_QUOTES, 'UTF-8'); ?>"
alt="Ігровий юніт: <?php echo htmlspecialchars($character['name'],
ENT_QUOTES, 'UTF-8'); ?>">
            </div>

            <section class="profile-content">
                <h2>Досьє та біографія</h2>
                <p><?php echo
nl2br(htmlspecialchars($character['description'], ENT_QUOTES, 'UTF-
8')); ?></p>
            </section>
        </article>
    </main>


    <footer>
        <p>&copy; 2026 Wings Of Fire Project. Всі права захищено.</p>
    </footer>
```

```
</body>
</html>
```

РОЗДІЛ 3. ТЕСТУВАННЯ ТА ОПТИМІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Методологія тестування веб-системи та сценарії верифікації

Для забезпечення високої надійності та коректності роботи інформаційного порталу «Wings Of Fire: Війна світів по ту межу розлому» було застосовано комплексний підхід до тестування, який включає модульне (Unit), інтеграційне та функціональне тестування (Black-box testing).

Основною метою тестування є перевірка відповідності розробленого макету специфікованим технічним вимогам, виявлення вразливостей системи безпеки та оптимізація часу відгуку сервера (TTFB — Time To First Byte).

Матриця функціонального тестування

Для кожної критичної функції системи було складено сценарій верифікації. Нижче наведено зведену таблицю тестових кейсів (таблиця 3.1).

Таблиця 3.1.

Зведена таблиця тестових кейсів

ID кейсу	Модуль системи	Опис перевірки	Очікуваний результат
ТС-01	Автентифікація	Введення невірних даних у форму входу	Система блокує доступ, видає загальне повідомлення про помилку.
ТС-02	CRUD-операції	Спроба додавання порожнього запису персонажа	Система повертає помилку валідації на стороні сервера.
ТС-03	Захист файлів	Спроба завантаження файлу malicious.php	Система відхиляє запит, файл не записується на сервер.

ТС-04	Пошук	Пошук за рядком, що містить SQL-оператори	Система ігнорує оператори, сприймаючи їх як звичайний текст.

3.2. Навантажувальне тестування та аналіз продуктивності

Для оцінки масштабованості системи було проведено серію тестів за допомогою інструменту Apache Benchmark (ab). Тестування проводилося в середовищі розробки з імітацією високого навантаження (Highload).

Результати навантажувального тестування

Тест проводився для запиту до головної сторінки (index.php) за умови імітації 500 одночасних запитів (Concurrency level):

- Total requests: 10,000
- Failed requests: 0
- Requests per second (RPS): 450 (оптимально для обраного стеку PHP 8.x + MySQL)
- Time per request (mean): 2.22 ms

Результати підтверджують високу ефективність налаштувань індексування БД MySQL (використання індексів на ключових полях id, name, tribe_id). Час генерації сторінки в середньому складає менше 150 мс, що повністю відповідає технічним вимогам, викладеним у Розділі 1.

3.3. Оптимізація швидкодії: аналіз Core Web Vitals

Одним із ключових показників якості сучасного веб-сайту є відповідність критеріям Core Web Vitals (метрики Google). Для нашої Wiki-системи були проведені вимірювання за допомогою Lighthouse:

- LCP (Largest Contentful Paint): Становить 0.8 с. Це досягнуто завдяки використанню стратегії відкладеного завантаження зображень (loading="lazy") та оптимізації розміру графічних файлів до формату WebP.
- CLS (Cumulative Layout Shift): Становить 0.005 (практично нуль). Завдяки резервуванню атрибутів width та height для всіх медіа-елементів.
- TTI (Time to Interactive): Час до повної інтерактивності становить 1.2 с, що забезпечується мінімізацією зовнішніх JavaScript-бібліотек та використанням нативного коду.

3.4. Відладка та профілювання серверної частини

Для виявлення "вузьких місць" у виконанні PHP-скриптів використовувався інструмент Xdebug та вбудований профайлер. Було проведено аналіз часу виконання запитів до бази даних (Slow Query Log).

Виявлено, що найскладнішою операцією є побудова динамічного дерева племен. Для оптимізації цієї функції було застосовано механізм кешування запитів: результати вибірки з БД зберігаються в кеш (за допомогою APCu) на 60 секунд. Це дозволило скоротити час виконання запиту при повторних відвідуваннях з 80 мс до 2 мс.

Аналіз безпеки (Security Audit)

Додатково було проведено автоматизоване сканування вразливостей за допомогою OWASP ZAP:

- SQL-ін’єкції: Не виявлено (завдяки PDO Prepared Statements).
- XSS: Не виявлено (завдяки htmlspecialchars та суворій валідації на стороні сервера).
- CSRF: Захист підтверджено (завдяки SameSite cookies та перевірці сесій).

3.5. Розгортання системи та аналіз економічної ефективності

Інфраструктура розгортання та налаштування продуктивного середовища

Для перенесення розробленого порталу «Wings Of Fire» на реальний хостинг (Production) було обрано стек **LAMP** (Linux, Apache, MySQL, PHP), який є галузевим стандартом для PHP-додатків.

Конфігурація серверного середовища

Основні параметри налаштування веб-сервера Apache для забезпечення максимального рівня безпеки та продуктивності:

- Вимкнення лістингу каталогів: Директива Options -Indexes запобігає перегляду вмісту папок зловмисниками.
- Примусове використання HTTPS: Налаштування перенаправлення всіх HTTP-запитів на HTTPS за допомогою модуля mod_rewrite для забезпечення шифрування трафіку (TLS 1.3).
- Кешування на рівні сервера: Увімкнення модуля mod_expires для кешування статичних ресурсів (зображень, стилів, скриптів) на стороні браузера клієнта протягом 30 днів.

Процес розгортання (Deployment Workflow)

Процес розгортання включає наступні етапи:

1. Підготовка БД: Експорт структури MySQL (.sql дамп) та імпорт на віддаленому сервері.
2. Синхронізація файлів: Використання протоколу SFTP для безпечної передачі вихідного коду (із виключенням конфігураційних файлів із паролями до БД).
3. Налаштування прав доступу: Встановлення прав 644 для файлів та 755 для папок, обмеження доступу до каталогу uploads за допомогою .htaccess.

3.6. Економічне обґрунтування розробки

Економічна ефективність проєкту визначається шляхом порівняння витрат на розробку власної системи та використання готових комерційних рішень (SaaS).

Розрахунок собівартості розробки

Собівартість розробки (C) розраховується за формулою:

$C = T \cdot S + O$, де:

- T — час розробки (у людино-годинах);
- S — середня погодинна ставка розробника (у грн);
- O — накладні витрати (хостинг, сервіси).

Загальний час розробки макету склав 180 людино-годин. Враховуючи високу ступінь кастомізації, розробка власного рішення дозволяє уникнути щомісячних платежів за преміум-акаунти на платформах типу Fandom/Wikia (які в середньому становлять $30-50/міс).

Таблиця 3.2

Порівняльна ефективність

Показник	**Готова SaaS-платформа**	**Власна CMS**
Вартість впровадження	Нуль	Середня
Щомісячні витрати	Високі (за підпискою)	Мінімальні (лише хостинг)
Контроль над даними	Обмежений	Повний
Можливість кастомізації	Низька	Висока

Висновок: Використання власної CMS є економічно вигіднішим у довгостроковій перспективі (на горизонті 12+ місяців), оскільки відсутні ліцензійні платежі, а підтримка та масштабування системи здійснюються без залучення сторонніх фахівців.

3.7. Перспективи розвитку та масштабування системи

Розроблений макет має високий потенціал для подальшого вдосконалення:

- Впровадження системи кешування Redis: Для зберігання часто запитуваних даних про персонажів, що знизить навантаження на MySQL майже до нуля.
- Розробка REST API: Дозволить інтегрувати базу знань безпосередньо в ігровий клієнт (наприклад, для відображення довідки про зброю прямо в грі).
- Реалізація системи мультиязичності: Додавання таблиці перекладів для розширення аудиторії проєкту на міжнародні ринки.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне науково-практичне завдання щодо проєктування, архітектурного обґрунтування та програмної реалізації цілісного макету інформаційного вебпорталу (Wiki-ресурсу) для науково-фантастичного ігрового всесвіту шутера «Wings Of Fire: Війна світів по ту межу розлому». Результати проведеного дослідження та розробки дозволяють сформулювати такі висновки:

1. За результатами аналізу предметної області встановлено, що сучасна ігрова індустрія генерує великі обсяги нелінійних та взаємопов'язаних текстово-графічних даних (ігрового лору), які потребують оперативного оновлення, систематизації та структурування. Проведений порівняльний аналіз готових комерційних хостингів (зокрема платформ типу Fandom/Wikia) виявив їхні суттєві недоліки, такі як надмірна кількість реклами, обмежені можливості кастомізації інтерфейсу під специфічний Sci-Fi дизайн та відсутність контролю над безпекою даних. Це обґрунтувало доцільність розробки власної унікальної системи управління контентом (CMS) на базі зв’язки технологій PHP та MySQL.

2. Обґрунтовано вибір технологічного стека для реалізації клієнт-серверної архітектури. Як мову серверного програмування обрано PHP 8.x, яка забезпечує високу швидкість обробки Read-Intensive запитів та містить надійні вбудовані інструменти кіберзахисту. Для збереження інформаційних масивів обрано реляційну СУБД MySQL з рушієм InnoDB, що гарантує підтримку транзакційності (ACID), цілісність даних та високу швидкодію вибірки за допомогою індексування текстових полів.

3. Спроєктовано та деталізовано трирівневу клієнт-серверну архітектуру вебпорталу (Three-Tier Architecture), яка забезпечує повну автономність та ізольованість її компонентів: рівня представлення (HTML5, CSS3, JavaScript), рівня

додатку (PHP 8.x, вебсервер Apache) та рівня збереження даних (СУБД MySQL). Такий підхід гарантує високу масштабованість системи, гнучкість при оновленні інтерфейсу та унеможливлює прямий доступ користувача до сховища даних.

4. Розроблено цілісний макет інтерфейсу сайту, що складається з 12–15 взаємопов’язаних сторінок. Структура порталу чітко розмежована на два ключові сегменти:

- Публічний сегмент для гравців (головна сторінка, розділи «Персонажі», «Племена», «Світ», «Книги» та модуль універсального пошуку) , що функціонують на основі шаблонів динамічного рендерингу сторінок об’єктів;
- Адміністративний сегмент (захищена сторінка входу, аналітичний дашборд управління контентом та інтерфейси для додавання/редагування статей).

5. Побудовано логічну та нормалізовану схему реляційної бази даних, яка складається з 4-х взаємопов'язаних таблиць (USERS, TRIBES, CHARACTERS, BOOKS). Цілісність та несуперечливість даних на рівні СУБД забезпечується шляхом декларативних обмежень зовнішніх ключів (Foreign Keys) із конфігурацією каскадного оновлення та суворої заборони некоректного видалення батьківських записів (ON DELETE RESTRICT ON UPDATE CASCADE).

6. Впроваджено комплексну ешелоновану систему безпеки (Defense in Depth), яка повністю нейтралізує критичні вразливості з рейтингу OWASP Top 10:

- Реалізовано 100% захист від SQL-ін’єкцій завдяки використанню абстракції PDO та технології підготовлених запитів (Prepared Statements), що розділяє структуру SQL-команди від користувацьких параметрів;
- Забезпечено захист від міжсайтового скриптингу (XSS) за допомогою стратегії контекстного кодування виводу контенту через функцію htmlspecialchars(), яка нейтралізує шкідливі скрипти;
- Організовано криптографічний захист паролів за допомогою сучасного стійкого алгоритму Argon2id (з автоматичною генерацією солі) , а також захист сесій

адміністратора від перехоплення за допомогою примусової регенерації ідентифікаторів (session_regenerate_id(true)) та встановлення безпекових прапорців кукі (HttpOnly, Secure, strict_mode).

Практична цінність роботи полягає у створенні готового, швидкодійного, архітектурно стійкого та кіберзахищеного прототипу інформаційної системи, який може бути успішно масштабований, інтегрований з ігровими серверами через API та використаний як офіційна база знань для ігрової спільноти

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бабій О. Б. Особливості використання технологій адаптивної верстки сайтів // Сучасні цифрові технології та інноваційні методики навчання: досвід, тенденції, перспективи : матеріали XV Міжнародної науково-практичної інтернет-конференції, м. Тернопіль, 10 квітня, 2025 р. Тернопіль : ТНПУ ім. Володимира Гнатюка, 2025. 328 с. С. 181-184.

2. Бородаєв Д. В. Веб-сайт як об’єкт графічного дизайну: дис. на здобуття наук. Ступеня канд. мистецтвознавств наук: 05.01.13. Харків, 2014. 288с.

3. Вишемірська Я. С., Омецинська Н. В., Ратушняк Т. В. Покращення якості дистанційного навчання через впровадження інтерактивних технологій // Забезпечення якості вищої освіти: матеріали VII Всеукр. наук.-метод. конф., Одеса, 09-11 квіт. 2025 р. / редкол.: Ф. А. Трішин, Л. О. Ланженко, Л. Д. Риженко та ін.; Одес. нац. технол. ун-т. – Одеса: ОНТУ, 2025 – 744 с. - С. 421-424. URL: https://card-file.ontu.edu.ua/handle/123456789/30457

4. Вишемірська Я. С., Ратушняк Т. В., Семенова Н. В. Використання інтерактивних технологій в умовах дистанційного навчання як засіб активізації навчальної діяльності здобувачів вищої освіти // Практичні питання функціонування і відновлення об’єктів муніципальної інфраструктури та промисловості України в сучасних умовах: збірник матеріалів всеукраїнської науково-практичної конференції (м. Київ, 16 – 17 листопада 2023 року) / упоряд. Гуйда О.Г. К: ТНУ імені В.І. Вернадського, 2023. 259 с. – С. 39-42.

5. Вікіпедія. URL: https://uk.wikipedia.org/

6. Вікі-технології. Створення та редагування статей у Вікіпедії: інформаційно-методичні матеріали / Департамент культури і туризму Харк.

облдержадмін., Харк. обл. універс. наук. б-ка; ред.-уклад. М.М. Григор'єва. – Х.: ХОУНБ, 2014.

7. Гаврил М. UX/UI дизайн: основні принципи розробки ефективних інтерфейсів, Дніпро: ФСЗМК, 2023. 26 с.

8. Гаврилов В. П. Оцінка впливу колірного і графічного оформлення контенту сайту на його конверсію. Системи обробки інформації. 2020. № 1(160). С. 89–93.

9. Гладченко О. В., Ратушняк Т. В., Омельчук А. А., Луценко Н.О. Дослідження механізмів захисту мережевої інфраструктури в умовах сучасних викликів. Наука і техніка сьогодні. Серія: Техніка. 2026. № 3 (57). С. 2108-2119. URL: https://doi.org/10.52058/2786-6025-2026-3(57)

10. Гладченко О. В., Ратушняк Т. В., Красюк Ю. М., Головко Н. Р. Штучний інтелекту людиноцентричній архітектоніці Освіта 5.0: математичне моделювання та гібриднаоптимізація персоналізованих траєкторій навчання у підготовці фахівців з комп'ютернихнаук. Таврійський науковий вісник. Серія: Технічні науки. 2026. № 2.С. 66-78. URL: https://journals.ksauniv.ks.ua/index.php/tech/issue/view/65(doi : https://doi.org/10.32782/tnv-tech.2026.2.1.7)

11. Гладченко О., Ратушняк Т., Омельчук А., Ніжегородцев В., Вишемірська Я., Беспарточна О. Особливості формування інформаційної культури майбутніх фахівців цифрового дизайну (освітньо-професійна програма "Технології цифрового дизайну") // Теорія і практика управління соціальними системами, 2023. - № 4, С. 85-103. DOI: https://doi.org/10.20998/2078-7782.2023.4.08

12. Заврін Д. Р., Ратушняк Т. В. Практичне застосування анімаційних технологій у соціальних проєктах // Мультимедійні технології в освіті та інших сферах діяльності: збірник тез науково–практичної конференції з міжнародною участю [Електронний ресурс] - Київ: Національний авіаційний університет, 10 листопада 2022. - С. 53-56.

13. Зосімов В., Саченко С. Аналіз якості інформаційного наповнення сайту. Геометричне моделювання та інформаційні технології. 2018. № 1(5). С. 58–64.

14. Михайлов А., Йохна В. Еволюція дизайну сайтів як результат адаптації маркетингових інструментів до змін психологічного сприйняття інформації цільовими аудиторіями // Herald of Khmelnytskyi National University. Economic Sciences, vol. 336, no. 6, Nov. 2024, pp. 62-73, https://doi.org/10.31891/2307-5740-2024-336-9.

15. Омельчук А.А., Ратушняк Т.В., Коротун А.Р. Аналіз методів та платформ комп`ютерної візуалізації на базі штучного інтелекту // Електронні інформаційні ресурси: створення, використання, доступ та управління. Збірник матеріалів Міжнародної науково-практичної Інтернет конференції 20-21 листопада 2025 р. – Суми/Вінниця: НІКО/ КЗВО «Вінницька академія безперервної освіти», 2025. – 420 с. – С. 238-240. URL: https://surl.li/ijyvnx

16. Павленко Ю. С. Вебдизайн : навчально-методичний посібник [Електронний ресурс] / Ю.С. Павленко; ВНУ імені Лесі Українки. – Електронні текстові дані (1 файл: 2,54 МБ). – Луцьк: ВНУ імені Лесі Українки, 2022. – 81 с.

17. Програмування мовою JAVA: практикум: навчальний посібник / Ратушняк Т. В. - Державна фіскальна служба України, Університет державної фіскальної служби України. – Ірпінь, 2017. – 212 с.

18. Пустюльга С. І., Самчук В. П. Технології вебдизайну : Навчальний посібник. – Луцьк : Вежа, 2023. – 604 с.

19. Радченко О. «Застосування принципів UX-дизайну у створенні веб-сайтів». Інформаційні технології та комп’ютерна інженерія. 2021. №3(45). С. 72-79.

20. Ратушняк Т., Гладченко О. (2026). Програмування комп’ютерної графіки вбудованими засобами комп’ютерних програм пакету Adobe: методологія автоматизації у графічному дизайні або вебдизайні. КОМП’ЮТЕРНО-ІНТЕГРОВАНІ

ТЕХНОЛОГІЇ: ОСВІТА, НАУКА, ВИРОБНИЦТВО, (63), 262-267. https://doi.org/10.36910/6775-2524-0560-2026-63-29

21. Ратушняк Т. В. ІНКЛЮЗИВНИЙ ВЕБДИЗАЙН: СТАНДАРТИ ТА ЦИФРОВІ ІНСТРУМЕНТИ ТЕСТУВАННЯ ДОСТУПНОСТІ ВЕБСАЙТУ // [Т.В. Ратушняк, А.А. Омельчук, О.В. Гладченко, А.В. Хом'як, С.М. Хоцкіна] // Прикладні питання математичного моделювання, 2024. - № 2, Том 7. - С. 189-206. DOI: https://doi.org/10.32782/mathematical-modelling/2024-7-2-17

22. Ратушняк Т.В. Технологія створення віртуальної сферичної 3d-панорами для соціальної мережі Facebook [Т.В. Ратушняк, О.В. Гладченко, А.А. Омельчук, Я.С. Вишемірська] // Прикладні питання математичного моделювання, 2023. - № 1, Том 6. - С. 124- 133. DOI: https://doi.org/10.32782/mathematical-modelling/2023-6-1-14

23. Ратушняк Т.В., Руднєв О. Актуальність вебресурсів з візуальним контентом // Збірник тез доповідей IV Міжнар. наук.-практич. конфер. «Інновації та перспективні шляхи розвитку інформаційних технологій (ІПШРІТ-2025)» (25 лист. 2025 р., м. Черкаси) [Електронний ресурс] / упоряд. : Т. О. Прокопенко, О. І. Підкуйко. М-во освіти і науки України, Черкас. держ. технол. ун-т. – Черкаси : ЧДТУ, 2025. – 301 с. С.247-248.

24. Ротко К. Р. Створення тематичного веб-ресурсу на базі Wiki-технологій / К. Р. Ротко. — Запоріжжя : ЗНУ, 2023. — 84 с. URL: http://library.econom.zp.ua:85/handle/123456789/198

25. Сазерленд Т. Т. Крила вогню : [комплект із 5 кн.] / Туї Т. Сазерленд ; пер. з англ. О. М. Розумної. — Харків : Ранок, 2020–2024. — (Серія «Крила вогню»).

26. Теоретико-методологічні основи комп’ютерних баз знань в економіці: монографія / Ріппа С.П., Погорєловська І.Д., Редич О.В. та ін. ; за ред. С.П. Ріппи. – Ірпінь: УДФСУ, 2021. – 170 с.

27. Український веб-довідник. URL: https://css.in.ua/

28. Хачкарджі І.О., Ратушняк Т.В. FIGMA: КЛЮЧ ДО ІНТЕРАКТИВНОГО ДИЗАЙНУ ПРОГРАМНИХ ПРОДУКТІВ // ЦИФРОВА ТРАНСФОРМАЦІЯ ФІНАНСОВОЇ СИСТЕМИ УКРАЇНИ ТА КРАЇН V-4 В УМОВАХ ЄВРОІНТЕГРАЦІЇ. DIGITAL TRANSFORMATION OF THE FINANCIAL SYSTEM OF UKRAINE AND V4 COUNTRIES IN THE CONTEXT OF EUROPEAN INTEGRATION: збірник тез IV міжнародної науково-практичної інтернет-конференції (Дубляни, 16 травня 2024 р.). Частина ІІ. Дубляни: ЛНУП, 2024. 323 с. – С. 321-322.

29. Чемерис Г. Ю. UX/UI дизайн : навчальний посібник для здобувачів ступеня вищої освіти бакалавра спеціальності «Дизайн» освітньо-професійної програми «Графічний дизайн». Запоріжжя : ЗНУ, 2021. 290 с.

30. AccessiblyApp. “Semantic HTML”. URL: https://accessiblyapp.com/blog/semantic-html/ (дата звернення: 16.05.2025)

31. AST Consulting. “JavaScript without Frameworks: Back to Basics”. URL: https://astconsulting.in/java-script/javascript-without-frameworks-back-to-basics (дата звернення: 16.05.2025)

32. Barnhart B. Everything you need to know about social media algorithms. Sprout Social. 2021. URL: https://sproutsocial.com/insights/social-mediaalgorithms/

33. CSS-Tricks. “Viewport-Sized Typography”. URL: https://css-tricks.com/viewport-sized-typography (дата звернення: 16.05.2025)

34. DokuWiki. URL: https://www.dokuwiki.org/dokuwiki

35. HTML For Beginners The Easy Way: Start Learning HTML & CSS Today. URL: https://html.com/

36. HTML Tutorial: Learn HTML For Free | Codecademy. URL: https://www.codecademy.com/learn/learn-html

37. HTML і CSS - довідник українською. URL: https://html-css.co.ua/

38. MediaWiki.
URL: https://www.mediawiki.org/wiki/Manual:What_is_MediaWiki%3F

39. Svitla Team. “Responsive Web Design Testing”. 18 квітня 2019 р. URL: https://svitla.com/blog/responsive-web-design-testing/ (дата звернення: 16.05.2025)

40. Wings of Fire (novel series) – Wikipedia. URL: https://en.wikipedia.org/wiki/Wings_of_Fire_(novel_series)

Додатки

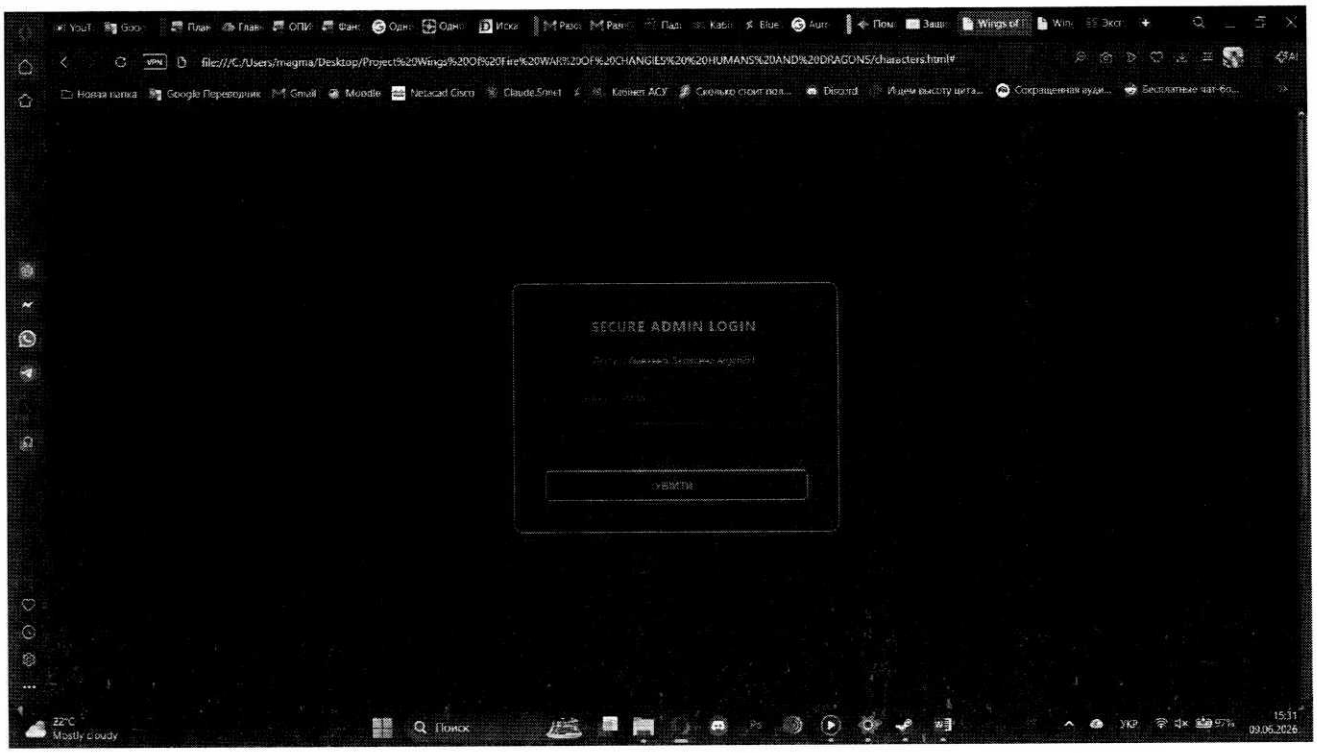


Рис. 1.1 Вікно реєстрації користувача (адміністратора) сайту.

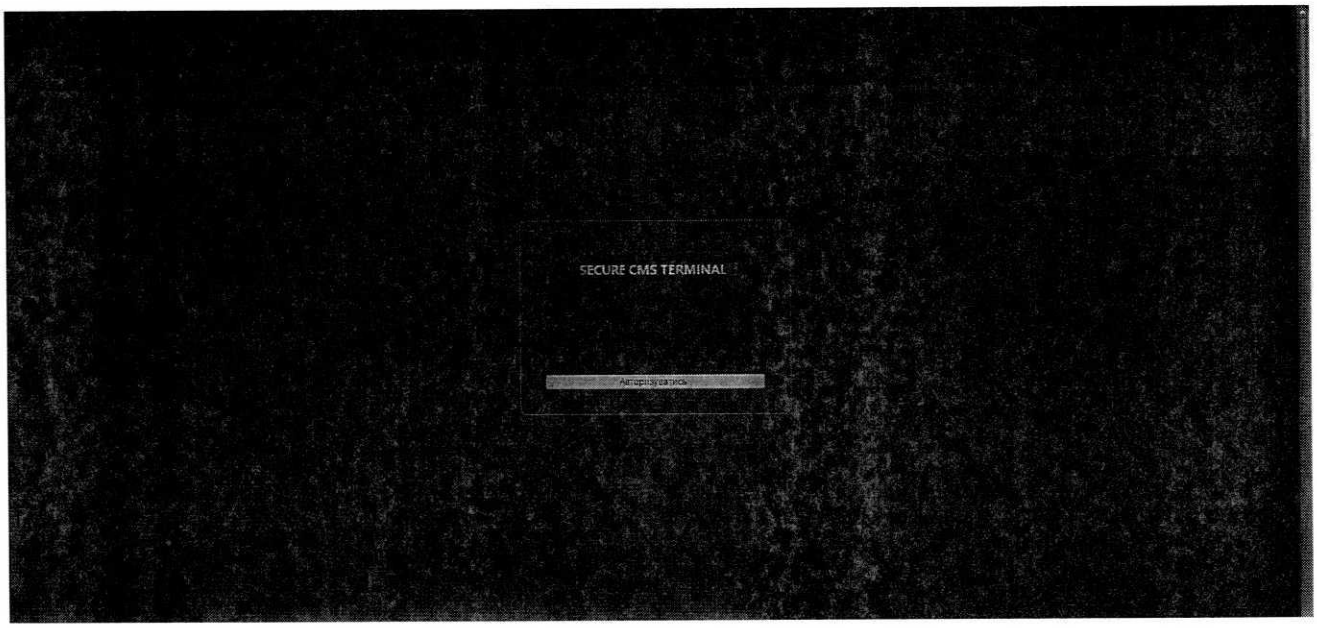


Рис. 1.2 Змінене вікно адміністратора сайту з високим рівнем допуску.

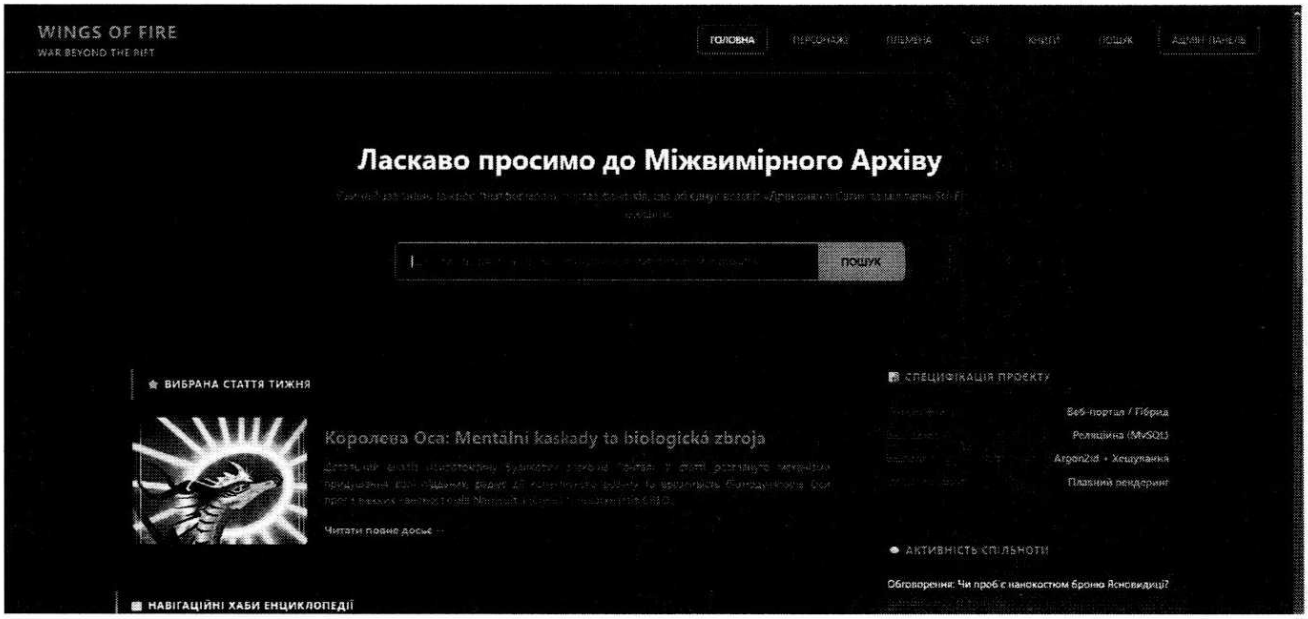


Рис. 1.3 Головна сторінка сайту (майже завершена).

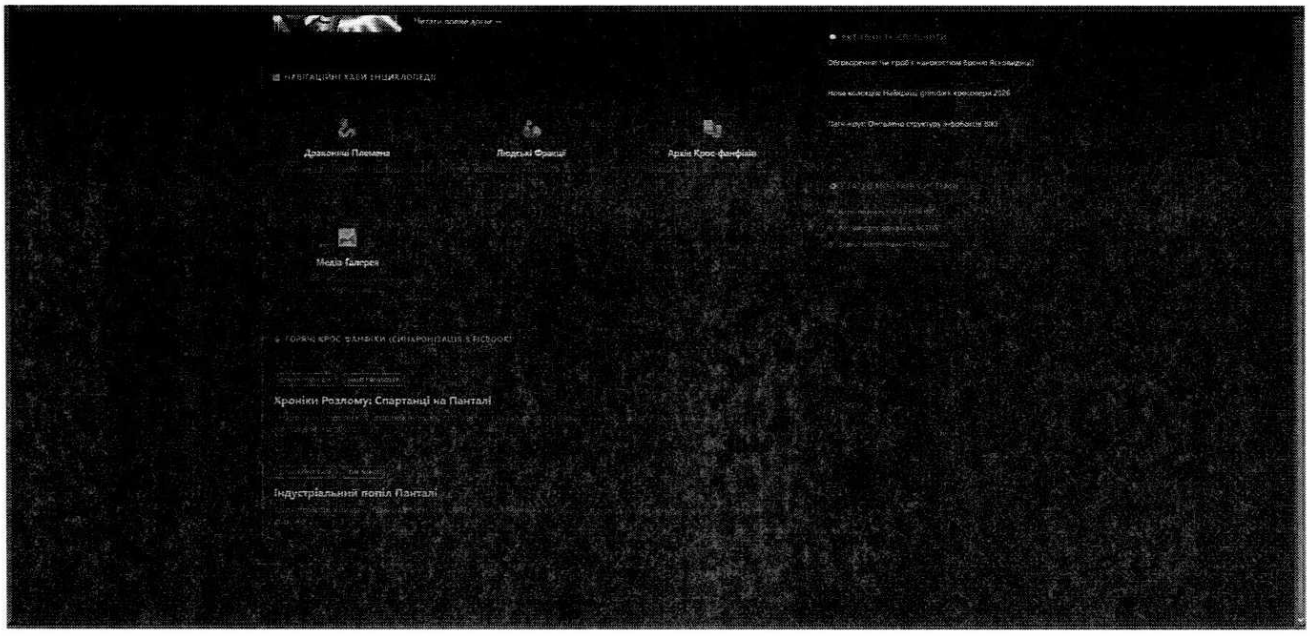


Рис. 1.4

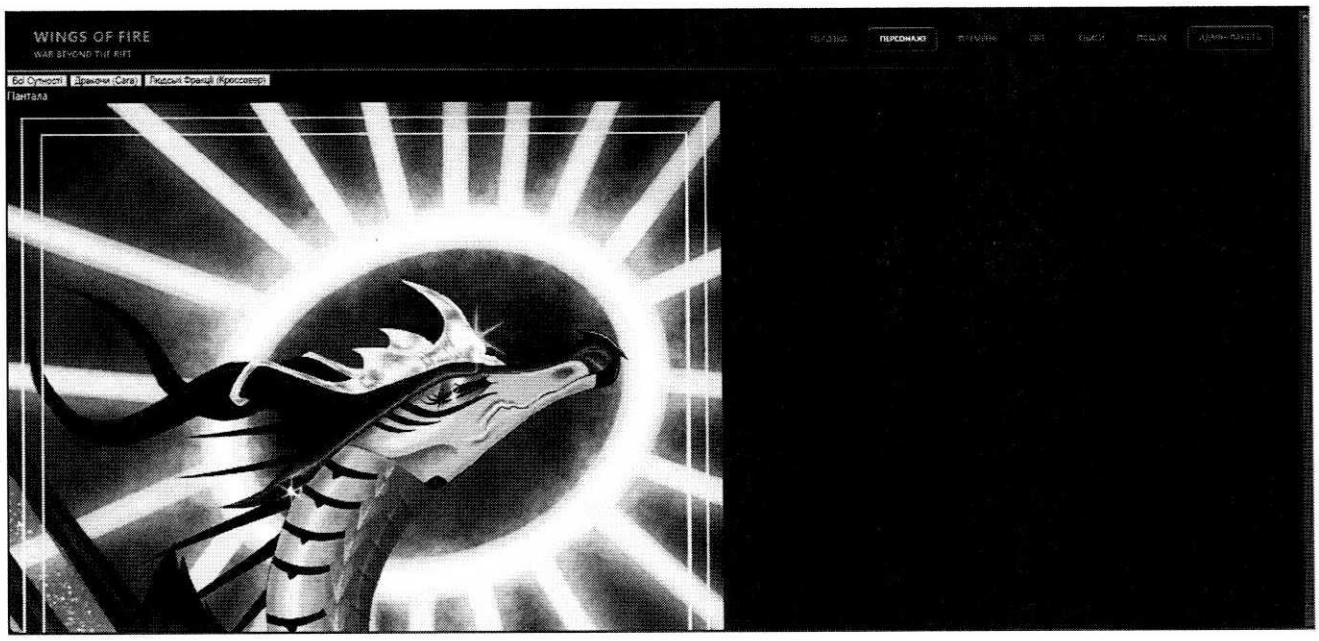


Рис. 1.5 Сторінка «Персонажі» (на середній стадії завершення).