

Міністерство освіти і науки України

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра інженерії програмного забезпечення

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Бакалавр

(назва освітнього ступеня)

на тему: Розробка сайту для новин і подій ігрової індустрії на основі PHP

Виконав(ла): студент(ка) 4 курсу, групи СП-43
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

		Чухрій О.О.
	(підпис)	(прізвище та ініціали)
Керівник		Коноваленко І.В.
	(підпис)	(прізвище та ініціали)
Нормоконтроль		Стоянов Ю.М.
	(підпис)	(прізвище та ініціали)
Завідувач кафедри		Петрик М.Р.
	(підпис)	(прізвище та ініціали)
Рецензент		
	(підпис)	(прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота для здобуття ступеня «бакалавр» на тему: «Розробка сайту для новин і подій ігрової індустрії з функціоналом фільтрації контенту, коментарів та оцінювання на основі PHP». Тернопільського національного технічного університету імені Івана Пулюя, Факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СП-43.

Відомість про обсяг: 70 сторінки, 6 рисунків, 3 таблиць, 1 додатків, 29 джерела.

Об'єктом дослідження є процес розробки вебсайтів для новин і подій ігрової індустрії.

Метою роботи є створення вебплатформи з функціоналом фільтрації, коментування, оцінювання матеріалів і адаптації під україномовну аудиторію.

У роботі застосовано системний підхід до аналізу предметної області, спроектовано архітектуру сайту й бази даних, реалізовано інтерфейс користувача з урахуванням принципів UX/UI, проведено тестування продуктивності й впровадження проекту з використанням Docker, Laravel Telescope та інших сучасних засобів розробки.

У результаті створено інтерактивний вебсайт, який забезпечує персоналізований доступ до новин, підтримує взаємодію користувачів і відповідає вимогам продуктивності, безпеки та доступності.

Ключові слова: PHP, Laravel, MySQL, Вебсайт, Ігрова індустрія, Фільтрація, Коментарі, Оцінювання, UX/UI, Безпека, Тестування, Локалізація.

ABSTRACT

Bachelor's qualification thesis on the topic: "Development of a website for news and events in the gaming industry with content filtering, commenting, and rating functionality based on PHP." Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, group SP-43.

Document length: 70 pages, 6 figures, 3 tables, 1 appendix, 29 references.

The object of research is the process of developing websites for news and events in the gaming industry.

The purpose of this work is to create a web platform with functionality for filtering, commenting, rating content, and adapting it for a Ukrainian-speaking audience.

The work applies a systematic approach to the analysis of the subject area, designs the architecture of the website and database, implements a user interface based on UX/UI principles, and conducts performance testing and project deployment using Docker, Laravel Telescope, and other modern development tools.

As a result, an interactive website was created that provides personalized access to news, supports user interaction, and meets the requirements of performance, security, and accessibility.

Keywords: PHP, Laravel, MySQL, Website, Gaming industry, Filtering, Comments, Ratings, UX/UI, Security, Testing, Localization.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

- API – Application Programming Interface (інтерфейс програмування додатків)
- CSS – Cascading Style Sheets (каскадні таблиці стилів)
- HTML – HyperText Markup Language (мова розмітки гіпертексту)
- HTTP – HyperText Transfer Protocol (протокол передачі гіпертексту)
- JSON – JavaScript Object Notation (формат обміну даними)
- MVC – Model-View-Controller (модель-вигляд-контролер, архітектурний шаблон)
- MySQL – My Structured Query Language (система керування базами даних)
- ORM – Object-Relational Mapping (об'єктно-реляційне відображення)
- PHP – Hypertext Preprocessor (препроцесор гіпертексту)
- SQL – Structured Query Language (мова структурованих запитів)
- UI – User Interface (інтерфейс користувача)
- UX – User Experience (користувацький досвід)
- AJAX – Asynchronous JavaScript and XML (асинхронний JavaScript і XML)
- REST – Representational State Transfer (передача стану представлення)
- CAPTCHA – Completely Automated Public Turing test to tell Computers and Humans Apart (автоматизований тест для розрізнення людей і комп'ютерів)

ЗМІСТ

ЗМІСТ	7
ВСТУП.....	9
РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ТА ВИМОГ ДО РОЗРОБКИ САЙТІВ ДЛЯ НОВИН І ПОДІЙ ІГРОВОЇ ІНДУСТРІЇ.....	12
1.1 Аналіз існуючих сайтів новин і подій ігрової індустрії.....	12
1.2 Вивчення потреб цільової аудиторії та їхніх очікувань.....	16
1.3 Огляд сучасних технологій для створення вебсайтів на основі PHP	18
1.4 Аналіз вимог до функціоналу фільтрації, коментарів та оцінювання	20
1.5 Висновки до розділу	22
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА САЙТУ ДЛЯ НОВИН І ПОДІЙ ІГРОВОЇ ІНДУСТРІЇ.....	23
2.1 Розробка архітектури сайту та структури бази даних.....	23
2.2 Проектування інтерфейсу користувача з урахуванням UX/UI принципів.....	24
2.3 Реалізація функціоналу фільтрації контенту на основі PHP	26
2.4 Розробка системи коментарів та оцінювання матеріалів.....	27
2.5 Забезпечення безпеки та оптимізації сайту	30
2.6 Висновки до розділу	33
РОЗДІЛ 3. ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ САЙТУ	35
3.1 Розробка плану тестування функціоналу сайту	35
3.2 Проведення тестування фільтрації, коментарів та оцінювання	38
3.3 Аналіз продуктивності та оптимізація роботи сайту	42
3.4 Впровадження сайту та навчання користувачів.....	45
3.5 Висновки до розділу	48
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	50
4.1 Долікарська допомога про отруєннях.....	50
4.2 Заходи щодо захисту обладнання від короткого замикання	52
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57

ДОДАТКИ.....	60
Додаток А – лістинг кваліфікаційної роботи	61
Додаток Б.....	70

ВСТУП

Сучасний розвиток інформаційних технологій та зростання популярності ігрової індустрії створюють потребу в якісних інформаційних ресурсах, які забезпечують оперативне інформування користувачів про актуальні новини та події. Ігрова індустрія є однією з найдинамічніших галузей, що охоплює широкий спектр подій: від випусків нових ігор і оновлень до кіберспортивних турнірів та анонсів розробників. У цьому контексті вебсайти, присвячені новинам і подіям ігрової індустрії, стають важливим інструментом для залучення аудиторії, яка прагне отримувати структуровану, актуальну та зручну для сприйняття інформацію.

Розробка вебсайтів для таких цілей вимагає врахування сучасних технологій та потреб користувачів. Мова програмування PHP, завдяки своїй універсальності, підтримці серверної логіки та широкій інтеграції з базами даних, залишається популярним вибором для створення динамічних вебресурсів. Окрім того, сучасні користувачі очікують від сайтів не лише інформативності, але й інтерактивності, що передбачає наявність таких функцій, як фільтрація контенту за категоріями, можливість залишати коментарі та оцінювати матеріали. Ці функції підвищують залученість аудиторії, сприяють формуванню спільноти та забезпечують зворотний зв'язок для покращення контенту.

Актуальність теми бакалаврської роботи зумовлена зростанням попиту на спеціалізовані платформи, які надають користувачам швидкий доступ до новин і подій ігрової індустрії з можливістю персоналізації інформації. Фільтрація контенту дозволяє користувачам отримувати лише релевантні матеріали, система коментарів сприяє обміну думками, а функція оцінювання допомагає визначити популярність і якість контенту. Реалізація такого функціоналу на основі PHP є актуальним завданням, оскільки ця технологія забезпечує гнучкість, масштабованість і доступність для розробників різного рівня підготовки.

Мета роботи – розробити вебсайт для новин і подій ігрової індустрії на основі PHP, який включає функціонал фільтрації контенту, коментарів та оцінювання, з урахуванням сучасних вимог до дизайну, безпеки та продуктивності.

Завдання роботи:

1. Провести аналіз сучасних технологій і підходів до створення новинних вебсайтів, зокрема в ігровій індустрії.
2. Розробити архітектуру сайту та структуру бази даних для забезпечення ефективної роботи функціоналу.
3. Реалізувати функціонал фільтрації контенту за категоріями, датами та іншими параметрами.
4. Створити систему коментарів із підтримкою модерації та захисту від спаму.
5. Впровадити механізм оцінювання матеріалів користувачами.
6. Забезпечити зручний інтерфейс користувача з урахуванням принципів UX/UI.
7. Провести тестування розробленого сайту та оцінити його продуктивність.
8. Впровадити сайт і надати рекомендації щодо його підтримки та розвитку.

Об'єкт дослідження – процес розробки вебсайтів для новин і подій ігрової індустрії з використанням мови програмування PHP.

Предмет дослідження – методи та технології створення вебсайтів із функціоналом фільтрації контенту, коментарів та оцінювання на основі PHP.

Методи дослідження включають аналіз літературних джерел, порівняльний аналіз існуючих рішень, системний підхід до проектування архітектури сайту, експериментальну розробку програмного забезпечення, тестування функціоналу та аналіз результатів.

Наукова новизна полягає в розробці спеціалізованого вебсайту для ігрової індустрії, який поєднує інтерактивний функціонал (фільтрація, коментарі,

оцінювання) з оптимізацією для високої продуктивності та безпеки, адаптований до потреб україномовної аудиторії.

Практична цінність роботи полягає в створенні готового вебсайту, який може бути використаний як платформа для інформування про новини та події ігрової індустрії, а також як основа для подальшого розвитку подібних проєктів. Розроблений функціонал фільтрації, коментарів та оцінювання може бути адаптований для інших інформаційних ресурсів.

Структура роботи складається з трьох розділів. Перший розділ присвячено аналізу сучасного стану та вимог до розробки сайтів для новин і подій ігрової індустрії. Другий розділ охоплює проектування та розробку сайту, включаючи створення функціоналу фільтрації, коментарів та оцінювання. Третій розділ фокусується на тестуванні, впровадженні та оцінці ефективності розробленого рішення. Робота також включає вступ, висновки, список використаних джерел і додатки.

РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ТА ВИМОГ ДО РОЗРОБКИ САЙТІВ ДЛЯ НОВИН І ПОДІЙ ІГРОВОЇ ІНДУСТРІЇ

1.1 Аналіз існуючих сайтів новин і подій ігрової індустрії

Сучасна ігрова індустрія є однією з найбільш динамічних галузей, що зумовлює високий попит на інформаційні ресурси, які надають актуальні новини, огляди подій та анонси. Вебсайти, присвячені ігровій тематиці, відіграють ключову роль у формуванні інформаційного простору для гравців, розробників та ентузіастів. Аналіз існуючих рішень дозволяє виявити сильні та слабкі сторони таких платформ, а також визначити оптимальні підходи до розробки власного сайту. У цьому підрозділі розглянуто основні характеристики популярних сайтів новин і подій ігрової індустрії, їх функціонал, дизайн та технологічні особливості.

Одним із провідних прикладів є сайт IGN (ign.com), який є міжнародним лідером у висвітленні новин ігрової індустрії. На головній сторінці IGN представлено структуровану стрічку новин із можливістю фільтрації за платформами (PC, PlayStation, Xbox тощо) та типами контенту (огляди, анонси, трейлери). Сайт використовує адаптивний дизайн, що забезпечує зручність перегляду на різних пристроях. Наприклад, рисунок 1.1 демонструє головну сторінку IGN із чітко вираженою структурою категорій та інтерактивними елементами.

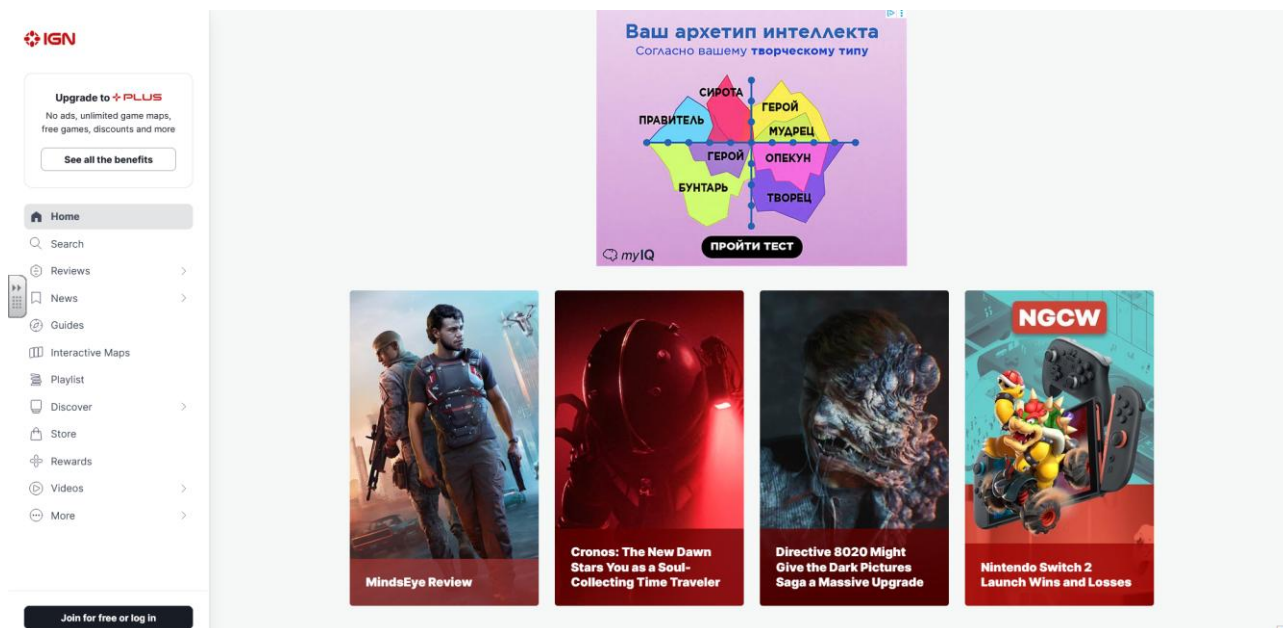


Рисунок 1.1 – Головна сторінка сайту IGN

Функціонал коментування на IGN дозволяє користувачам залишати відгуки до статей, однак доступ до нього обмежений для незареєстрованих користувачів, що сприяє зменшенню спаму [1]. Система оцінювання контенту реалізована через рейтинги оглядів, які виставляються редакторами, але користувацькі оцінки менш виражені. Це може бути недоліком, оскільки сучасна аудиторія часто очікує можливості самостійно оцінювати матеріали.

Іншим прикладом є сайт PC Gamer (pcgamer.com), який спеціалізується на новинах для ПК-геймерів. Сайт пропонує розширені можливості фільтрації контенту за жанрами ігор, датами публікації та популярністю. На рисунку 1.2 зображено розділ новин PC Gamer, де видно інтеграцію мультимедійного контенту, зокрема відео та скріншотів, що підвищує залученість користувачів.

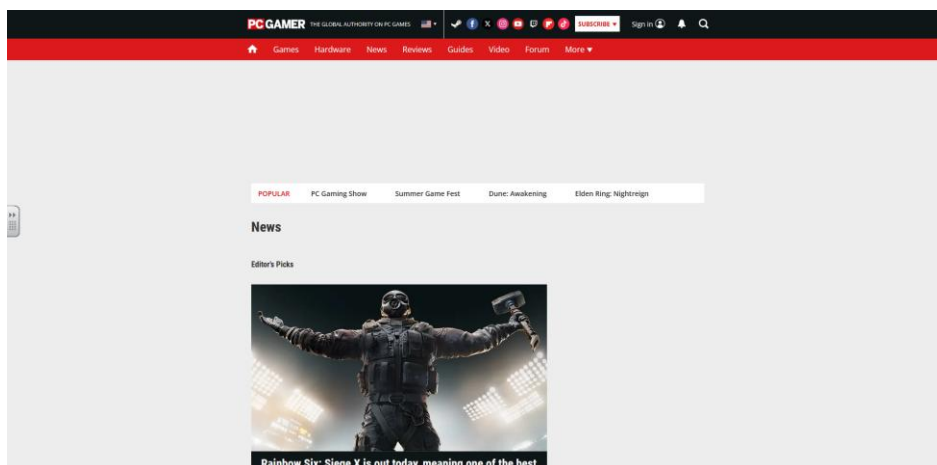


Рисунок 1.2 – Розділ новин сайту PC Gamer

PC Gamer активно використовує систему коментарів через платформу Disqus, яка забезпечує модерацію та інтеграцію із соціальними мережами [2]. Однак система оцінювання матеріалів обмежена внутрішніми рейтингами редакції, що знижує інтерактивність для користувачів. Це свідчить про необхідність впровадження гнучкіших систем оцінювання на нових платформах.

Український ресурс PlayUA (playua.in.ua) є прикладом локалізованого сайту для ігрової аудиторії. Сайт пропонує україномовний контент, включаючи новини, огляди та подкасти. На рисунку 1.3 показано головну сторінку PlayUA, яка має зручну навігацію та підтримку мультимедійного контенту. Проте функціонал фільтрації на сайті є базовим і обмежується пошуком за категоріями, що може не задовольняти потреби користувачів, які шукають персоналізований контент [3].

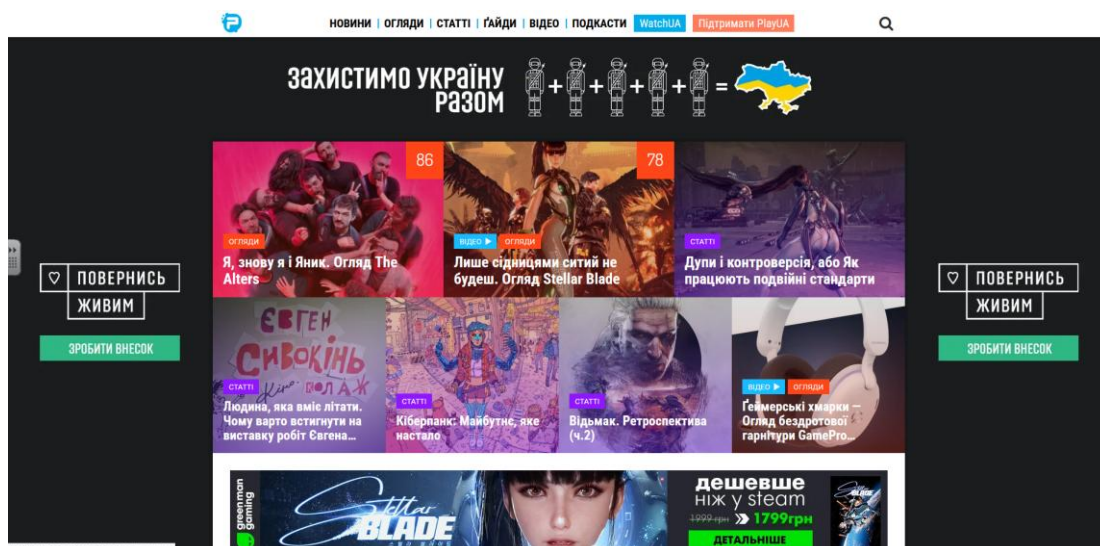


Рисунок 1.3 – Головна сторінка сайту PlayUA

Система коментування на PlayUA дозволяє користувачам залишати відгуки після авторизації через соціальні мережі, що спрощує взаємодію, але не має розвинених функцій модерації. Оцінювання матеріалів відсутнє, що є суттєвим недоліком для залучення аудиторії. Аналіз цього сайту показує важливість створення локалізованих платформ із розширеним функціоналом для україномовних користувачів.

Ще одним прикладом є сайт Kotaku (kotaku.com), який відомий своєю активною спільнотою та акцентом на інтерактивність. На рисунку 1.4 зображено сторінку статті на Kotaku, де видно розвинену систему коментарів та інтеграцію із соціальними мережами. Сайт використовує власну платформу для коментування, що дозволяє модерувати контент і підтримувати дискусії [4]. Однак фільтрація контенту є менш розвиненою порівняно з IGN чи PC Gamer, що може ускладнювати пошук матеріалів за специфічними критеріями.

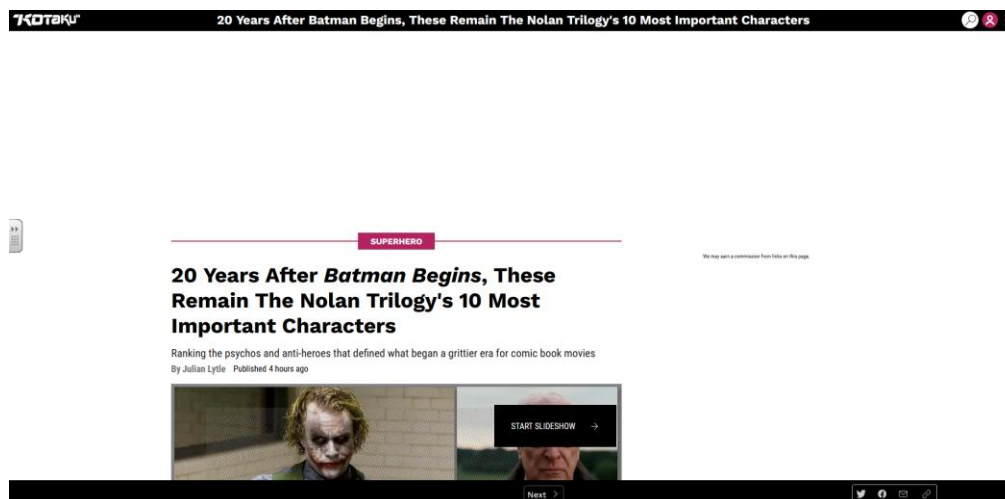


Рисунок 1.4 – Сторінка статті сайту Kotaku

Загалом, аналіз таких сайтів, як IGN, PC Gamer, PlayUA та Kotaku, показує, що успішні платформи поєднують зручний дизайн, адаптивність та інтерактивний функціонал. Проте більшість із них мають обмеження у фільтрації контенту чи оцінюванні матеріалів користувачами. Це вказує на потребу створення сайту, який би поєднував гнучку фільтрацію, зручну систему коментарів та можливість оцінювання контенту для підвищення заліder engagement [5].

1.2 Вивчення потреб цільової аудиторії та їхніх очікувань

Розуміння потреб цільової аудиторії є ключовим етапом у розробці вебсайту для новин і подій ігрової індустрії. Цільова аудиторія таких платформ включає геймерів, розробників ігор, журналістів, кіберспортивних ентузіастів та звичайних користувачів, зацікавлених в актуальній інформації про ігрову індустрію. Вивчення їхніх потреб і очікувань дозволяє створити сайт, який буде зручним, функціональним і привабливим для користувачів. У цьому підрозділі розглянуто основні характеристики аудиторії, їхні очікування щодо функціоналу сайту та підходи до їх задоволення.

Основною групою користувачів є геймери віком від 16 до 35 років, які активно слідкують за новинами про нові ігри, оновлення та кіберспортивні події. Згідно з дослідженнями, сучасні користувачі надають перевагу швидкому доступу до інформації та персоналізації контенту [6]. Наприклад, можливість фільтрувати новини за платформою (PC, консолі), жанром гри чи датою є важливою для геймерів, які хочуть отримувати лише релевантну інформацію. Це підтверджує необхідність реалізації гнучкої системи фільтрації на сайті.

Іншою важливою групою є кіберспортивна спільнота, яка цікавиться розкладами турнірів, результатами матчів та аналітикою. Ці користувачі очікують від сайту інтерактивних функцій, таких як коментування та оцінювання матеріалів, щоб ділитися думками та брати участь у дискусіях [7]. Система коментарів має бути зручною, із підтримкою авторизації через соціальні мережі, щоб спростити взаємодію. Крім того, можливість оцінювати статті чи огляди дозволяє користувачам висловлювати свою думку та впливати на популярність контенту.

Для україномовної аудиторії важливим є локалізований контент. Згідно з даними українських IT-ресурсів, україномовні користувачі віддають перевагу сайтам із рідною мовою інтерфейсу та контенту, що сприяє їхній залученості [8]. Наприклад, сайт PlayUA демонструє успішність україномовних платформ, але його функціонал обмежений у плані інтерактивності. Це вказує на потребу створення сайту, який поєднає локалізацію з розширеним функціоналом.

Очікування аудиторії також стосуються дизайну та зручності користування. Сучасні користувачі надають перевагу адаптивним сайтам, які однаково добре працюють на настільних комп'ютерах, планшетах і смартфонах [9]. Зручний інтерфейс із чіткою навігацією та швидким доступом до основних функцій (пошук, фільтрація, коментування) є критично важливим для утримання аудиторії. Крім того, безпека даних, зокрема захист від спаму в коментарях, є важливим аспектом, оскільки користувачі хочуть відчувати себе захищеними під час взаємодії з платформою.

Загалом, вивчення потреб цільової аудиторії показує, що успішний сайт повинен поєднувати персоналізацію контенту, інтерактивність, зручний інтерфейс і локалізацію. Функціонал фільтрації, коментарів та оцінювання має бути адаптованим до потреб геймерів і кіберспортивної спільноти, а також враховувати локальний контекст для україномовних користувачів [10].

1.3 Огляд сучасних технологій для створення вебсайтів на основі PHP

Мова програмування PHP залишається однією з найпопулярніших для створення динамічних вебсайтів завдяки своїй гнучкості, широкій підтримці та великій кількості фреймворків і бібліотек. У контексті розробки сайту для новин і подій ігрової індустрії PHP забезпечує ефективну обробку серверної логіки, взаємодію з базами даних і підтримку інтерактивних функцій. У цьому підрозділі розглянуто сучасні технології, фреймворки та інструменти, які використовуються для створення вебсайтів на основі PHP, а також їх переваги та недоліки. У таблиці 1.1 наведено порівняння основних фреймворків PHP.

Таблиця 1.1 – Порівняння популярних PHP-фреймворків

Фреймворк	Основні переваги	Недоліки	Застосування
Laravel	Зручний синтаксис, ORM Eloquent, Blade шаблони	Високе споживання ресурсів	Складні проєкти, новинні портали
Symfony	Гнучкість, модульність, підтримка великих проєктів	Складність для початківців	Корпоративні сайти, API
CodeIgniter	Легкість, швидкість, мінімальні вимоги	Обмежений функціонал для	Невеликі сайти, швидка розробка

Продовження таблиці 1.1

		великих проєктів	
Yii	Висока продуктивність, безпека, гнучкість	Складна документація	Електронна комерція, портали

Laravel є одним із найпопулярніших фреймворків для створення вебсайтів завдяки зручному синтаксису та вбудованим інструментам, таким як ORM Eloquent для роботи з базами даних. Наприклад, Laravel підтримує швидку розробку RESTful API, що є важливим для інтеграції з мобільними додатками чи зовнішніми сервісами [11]. Це робить його ідеальним вибором для новинного сайту з функціями фільтрації та коментування.

Symfony, у свою чергу, пропонує модульну структуру, що дозволяє створювати масштабовані проєкти. Його компоненти, такі як Doctrine для роботи з базами даних, забезпечують гнучкість у реалізації складних функцій, таких як фільтрація контенту за кількома параметрами [12]. Проте складність Symfony може бути перешкодою для початківців.

CodeIgniter є легким фреймворком, який підходить для швидкої розробки сайтів із мінімальними вимогами до серверних ресурсів. Він може бути використаний для створення базового новинного порталу, але його функціонал обмежений порівняно з Laravel чи Symfony [13].

Для управління базами даних у PHP-проєктах часто використовується MySQL, оскільки вона забезпечує швидку обробку запитів і підтримку складних структур даних. Наприклад, для зберігання новин, коментарів та оцінок можна використовувати реляційні таблиці, що забезпечують ефективну фільтрацію контенту [14].

Щодо клієнтської частини, PHP часто комбінується з JavaScript-фреймворками, такими як Vue.js, для створення динамічних інтерфейсів. Це дозволяє реалізувати інтерактивні функції, такі як асинхронне завантаження

коментарів чи оновлення рейтингів без перезавантаження сторінки [15]. Таким чином, РНР у поєднанні з сучасними технологіями забезпечує створення функціональних і зручних вебсайтів.

1.4 Аналіз вимог до функціоналу фільтрації, коментарів та оцінювання

Функціонал фільтрації, коментарів та оцінювання є ключовим для створення інтерактивного вебсайту новин і подій ігрової індустрії. Ці функції забезпечують персоналізацію контенту, залученість користувачів і зворотний зв'язок. У цьому підрозділі розглянуто вимоги до реалізації цих функцій, їх технічні аспекти та виклики. У таблиці 1.2 наведено основні вимоги до функціоналу.

Таблиця 1.2 – Вимоги до функціоналу фільтрації, коментарів та оцінювання

Функціонал	Основні вимоги	Технічні аспекти	Виклики
Фільтрація	Фільтри за категоріями, датами, популярністю	SQL-запити, AJAX для асинхронності	Оптимізація швидкості запитів
Коментування	Авторизація, модерація, захист від спаму	Інтеграція з Disqus або власна система	Захист від ботів, модерація
Оцінювання	Оцінки користувачів, середній рейтинг	База даних для зберігання оцінок	Запобігання накрутці рейтингів

Фільтрація контенту є критично важливою для забезпечення персоналізованого досвіду користувачів. Наприклад, користувачі повинні мати

можливість відфільтрувати новини за жанром гри, платформою чи датою публікації. Технічно це реалізується через SQL-запити до бази даних, які дозволяють швидко отримувати релевантний контент. Для підвищення зручності використовується AJAX для асинхронного оновлення результатів фільтрації без перезавантаження сторінки [16]. Основним викликом є оптимізація запитів для великих обсягів даних, щоб уникнути затримок.

Система коментування повинна підтримувати авторизацію через соціальні мережі або власну систему реєстрації, щоб спростити взаємодію для користувачів. Наприклад, платформа Commentics пропонує готове рішення для коментування з функціями модерації та захисту від спаму [17]. Важливим аспектом є захист від ботів, що може бути реалізовано через CAPTCHA або аналіз поведінки користувачів. Модерація коментарів має забезпечувати видалення небажаного контенту, що вимагає створення адміністративного інтерфейсу.

Оцінювання матеріалів дозволяє користувачам висловлювати свою думку про якість контенту через систему рейтингів (наприклад, 1–5 зірок). Технічно це реалізується через зберігання оцінок у базі даних і обчислення середнього рейтингу для кожної статті [18]. Викликом є запобігання накрутці рейтингів, що може бути вирішено шляхом обмеження оцінювання для зареєстрованих користувачів або використання алгоритмів виявлення підозрілої активності.

Інтеграція цих функцій потребує ретельного планування, щоб забезпечити їхню сумісність і ефективність. Наприклад, фільтрація та коментування можуть використовувати одну базу даних, але потребують різних підходів до оптимізації та безпеки [19]. Загалом, ці функції мають бути інтуїтивно зрозумілими для користувачів і відповідати сучасним стандартам веброзробки.

1.5 Висновки до розділу

Аналіз сучасного стану та вимог до розробки сайтів для новин і подій ігрової індустрії показав, що успішні платформи поєднують зручний дизайн, інтерактивний функціонал і адаптивність для різних пристроїв. Проведений аналіз сайтів, таких як IGN, PC Gamer, PlayUA та Kotaku, виявив їхні сильні сторони, зокрема структуровану подачу контенту та інтеграцію із соціальними мережами, а також недоліки, такі як обмежена фільтрація чи відсутність користувацького оцінювання [20]. Це підкреслює важливість створення сайту з гнучким функціоналом, який відповідає потребам аудиторії.

Вивчення цільової аудиторії показало, що геймери та кіберспортивна спільнота очікують персоналізації контенту, зручного інтерфейсу та інтерактивних функцій, таких як коментування та оцінювання. Для україномовних користувачів важливим є локалізований контент, що враховує мовні та культурні особливості [21]. Ці потреби мають бути враховані під час розробки сайту.

Огляд технологій для створення вебсайтів на основі PHP показав, що фреймворки, такі як Laravel і Symfony, є оптимальними для реалізації складних проєктів завдяки їхній гнучкості та підтримці сучасних інструментів [22]. Поєднання PHP із MySQL та JavaScript-фреймворками дозволяє створювати динамічні та швидкі сайти.

Аналіз вимог до функціоналу фільтрації, коментарів та оцінювання виявив необхідність використання SQL-запитів для фільтрації, інтеграції систем коментування, таких як Commentics, та створення безпечних механізмів оцінювання [23]. Основними викликами є оптимізація продуктивності та захист від спаму чи накрутки рейтингів.

Результати аналізу формують основу для розробки вебсайту, який поєднуватиме сучасні технології, інтерактивний функціонал і локалізацію для україномовної аудиторії, забезпечуючи високу залученість користувачів [24].

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА САЙТУ ДЛЯ НОВИН І ПОДІЙ ІГРОВОЇ ІНДУСТРІЇ

2.1 Розробка архітектури сайту та структури бази даних

Архітектура вебсайту та структура бази даних є основою для створення ефективного та масштабованого новинного порталу ігрової індустрії. Правильно спроектована архітектура забезпечує швидку обробку запитів, підтримку інтерактивних функцій і можливість подальшого розширення. У цьому підрозділі розглянуто підходи до розробки архітектури сайту, вибір технологій та створення структури бази даних для реалізації функціоналу фільтрації, коментарів та оцінювання. На рисунку 2.1 зображено схему архітектури сайту.

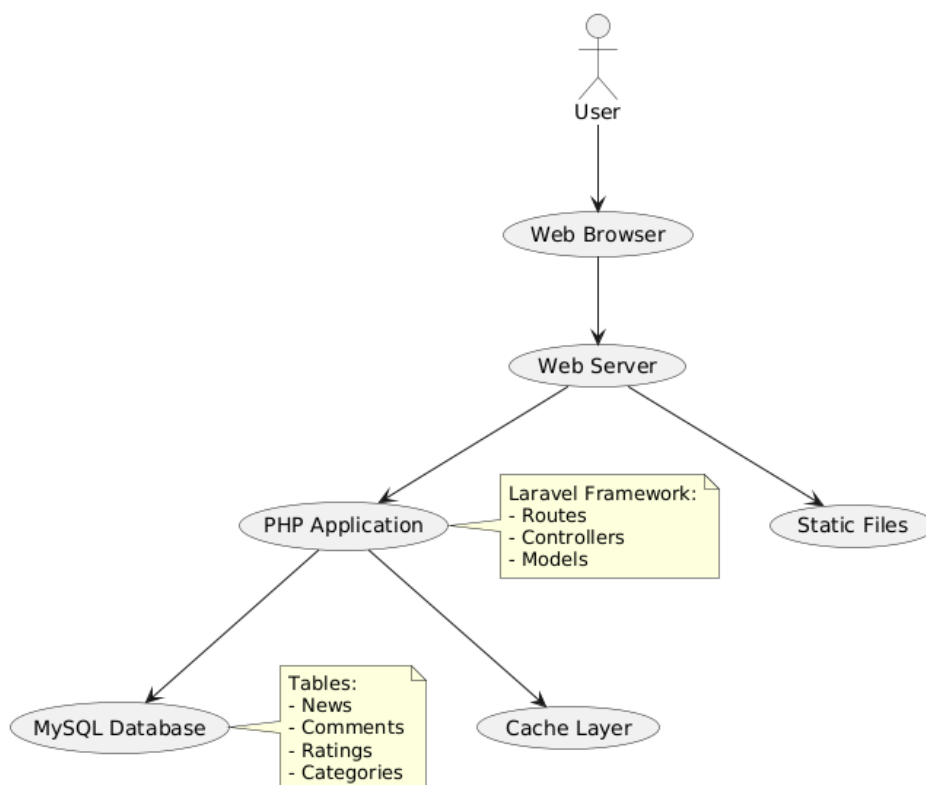


Рисунок 2.1 – Архітектура вебсайту новин і подій ігрової індустрії

Архітектура сайту базується на моделі клієнт-сервер, де клієнтська частина відповідає за відображення інтерфейсу, а серверна – за обробку запитів і

взаємодію з базою даних. Для серверної частини обрано PHP-фреймворк Laravel, який забезпечує зручну маршрутизацію, обробку запитів через контролери та взаємодію з базою даних через ORM Eloquent [25]. Laravel дозволяє організувати модульну структуру коду, що полегшує підтримку та масштабування проєкту.

База даних MySQL обрана завдяки її швидкості, надійності та підтримці реляційних структур. Основними таблицями є: news (зберігання новин), comments (коментарі до новин), ratings (оцінки користувачів) і categories (категорії для фільтрації). Наприклад, таблиця news містить поля для заголовка, тексту, дати публікації та категорії, що забезпечує ефективну фільтрацію контенту [26]. Для підвищення продуктивності використовується кешування даних через Redis, що зменшує навантаження на базу даних під час частих запитів.

Модуль фільтрації реалізується через контролери Laravel, які обробляють запити користувачів і формують SQL-запити до бази даних. Наприклад, фільтрація за категоріями чи датами виконується за допомогою умов WHERE у запитах. Система коментарів потребує окремої таблиці comments, пов'язаної з таблицею news через зовнішній ключ, що забезпечує швидкий доступ до коментарів для конкретної статті.

Оцінювання матеріалів реалізується через таблицю ratings, де зберігаються оцінки користувачів та їхній зв'язок із новинами. Для запобігання накрутці рейтингів використовується обмеження на одну оцінку від одного користувача, що контролюється через авторизацію [28]. Загалом, архітектура сайту та структура бази даних розроблені з урахуванням вимог до продуктивності, безпеки та масштабованості, що дозволяє ефективно реалізувати функціонал сайту.

2.2 Проектування інтерфейсу користувача з урахуванням UX/UI принципів

Інтерфейс користувача (UI) та користувацький досвід (UX) є критично важливими для забезпечення зручності та привабливості вебсайту. У контексті

сайту для новин і подій ігрової індустрії інтерфейс має бути інтуїтивно зрозумілим, адаптивним і орієнтованим на потреби цільової аудиторії. У цьому підрозділі розглянуто принципи проектування інтерфейсу, вибір інструментів та створення макету з урахуванням UX/UI. На рисунку 2.2 зображено схему інтерфейсу головної сторінки сайту.

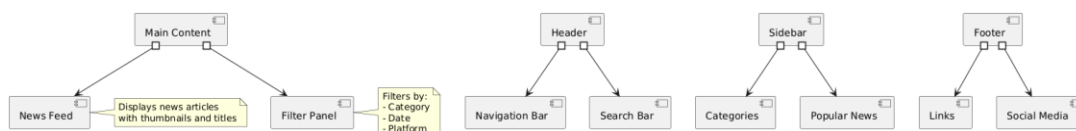


Рисунок 2.2 – Схема інтерфейсу головної сторінки

Проектування інтерфейсу базується на принципах UX, таких як простота, доступність і послідовність. Головна сторінка сайту включає навігаційну панель, панель пошуку, стрічку новин і бічну панель із категоріями та популярними матеріалами. Навігаційна панель забезпечує швидкий доступ до основних розділів, таких як “Новини”, “Події” та “Огляди”. Для створення адаптивного дизайну використовується CSS-фреймворк Bootstrap, який забезпечує коректне відображення на різних пристроях.

Панель фільтрації розташована поруч зі стрічкою новин і дозволяє користувачам обирати категорії, дати чи платформи. Наприклад, користувач може відфільтрувати новини за жанром (RPG, шутери) або платформою (PC, PlayStation). Для реалізації асинхронного оновлення фільтрів використовується JavaScript-бібліотека Vue.js, що забезпечує швидку реакцію інтерфейсу [30]. Система коментарів інтегрована на сторінці кожної статті, з можливістю авторизації через соціальні мережі для спрощення взаємодії.

Оцінювання матеріалів реалізовано через інтерактивні елементи, такі як зірковий рейтинг, який відображається під кожною статтею. Користувачі можуть оцінити матеріал від 1 до 5 зірок, а середній рейтинг відображається в реальному часі. Для забезпечення зручності UX використовуються чіткі візуальні підказки, такі як кнопки із зрозумілими іконками та контрастними кольорами.

Локалізація інтерфейсу для україномовної аудиторії передбачає використання україномовних шрифтів, таких як Noto Serif, та адаптацію текстових елементів до культурних особливостей. Наприклад, терміни, пов'язані з ігровою індустрією, перекладено з урахуванням їхнього використання в українській геймерській спільноті [32]. Загалом, інтерфейс спроектовано з урахуванням потреб користувачів, сучасних стандартів UX/UI та адаптивності для забезпечення максимальної зручності.

2.3 Реалізація функціоналу фільтрації контенту на основі PHP

Функціонал фільтрації контенту є ключовим для забезпечення персоналізованого досвіду користувачів на сайті новин і подій ігрової індустрії. Фільтрація дозволяє користувачам швидко знаходити релевантні матеріали за категоріями, датами чи іншими параметрами. У цьому підрозділі розглянуто процес реалізації фільтрації на основі PHP, включаючи серверну логіку, взаємодію з базою даних і оптимізацію продуктивності.

Для реалізації фільтрації використано PHP-фреймворк Laravel, який забезпечує зручну обробку запитів і взаємодію з базою даних через ORM Eloquent. Наприклад, для фільтрації новин за категоріями створюється контролер, який обробляє GET-запити від користувача. Запит формується на основі параметрів, таких як `category_id` або `date_range`, і передається до моделі `News` для отримання відповідних даних [33].

База даних MySQL містить таблицю `news`, яка включає поля `category_id`, `publish_date` і `platform`. Для фільтрації за категоріями використовується SQL-запит виду:

```
SELECT * FROM news WHERE category_id = :category_id ORDER BY  
publish_date DESC;
```

Цей запит дозволяє отримати всі новини для певної категорії, відсортовані за датою публікації. Для підвищення продуктивності застосовується індексація полів `category_id` і `publish_date`, що зменшує час виконання запитів.

Асинхронна фільтрація реалізується через AJAX-запити, які відправляються до PHP-контролера. Наприклад, коли користувач обирає категорію в інтерфейсі, JavaScript формує запит до серверної кінцевої точки `/filter`, яка повертає відфільтровані дані у форматі JSON. Це дозволяє оновлювати стрічку новин без перезавантаження сторінки.

Для обробки складних фільтрів, таких як комбінація категорій і дат, використовується динамічне формування SQL-запитів. Наприклад, запит для фільтрації за категорією та діапазоном дат виглядатиме так:

```
SELECT * FROM news WHERE category_id = :category_id AND publish_date  
        BETWEEN :start_date AND :end_date;
```

Це забезпечує гнучкість і точність фільтрації [36]. Для захисту від SQL-ін'єкцій використовуються параметризовані запити, що є стандартною практикою в Laravel.

Оптимізація продуктивності досягається через кешування результатів популярних фільтрів за допомогою Redis. Наприклад, список новин для популярних категорій кешується на 10 хвилин, що зменшує навантаження на базу даних. Загалом, функціонал фільтрації на основі PHP забезпечує швидкий і зручний доступ до релевантного контенту, відповідаючи потребам користувачів.

2.4 Розробка системи коментарів та оцінювання матеріалів

Система коментарів та оцінювання матеріалів є важливими компонентами інтерактивного вебсайту для новин і подій ігрової індустрії, оскільки вони

сприяють залученості користувачів, формуванню спільноти та отриманню зворотного зв'язку. Ці функції дозволяють користувачам висловлювати думки, оцінювати якість контенту та брати участь у дискусіях. У цьому підрозділі розглянуто процес розробки системи коментарів та оцінювання на основі PHP, включаючи серверну логіку, взаємодію з базою даних, захист від спаму та оптимізацію користувацького досвіду.

Для реалізації системи коментарів використано PHP-фреймворк Laravel, який забезпечує зручну обробку запитів і взаємодію з базою даних через ORM Eloquent. Система коментарів дозволяє користувачам залишати відгуки до новин після авторизації через соціальні мережі (наприклад, Google або Facebook) або власну систему реєстрації. Це спрощує доступ і підвищує зручність для користувачів [21].

База даних MySQL містить таблицю comments, яка включає поля id, news_id (зовнішній ключ для зв'язку з таблицею news), user_id, content, created_at і status (для модерації). Наприклад, SQL-запит для отримання коментарів до конкретної новини виглядає так:

```
SELECT * FROM comments WHERE news_id = :news_id AND status =  
      'approved' ORDER BY created_at DESC;
```

Цей запит повертає лише затверджені коментарі, відсортовані за датою створення [22]. Для захисту від спаму впроваджено CAPTCHA, яка перевіряє, чи є користувач людиною, перед відправкою коментаря. Додатково застосовується фільтрація вмісту для виявлення небажаних слів або посилань, що реалізується через бібліотеку PHP для обробки тексту [23].

Модерація коментарів здійснюється через адміністративну панель, де модератори можуть затверджувати, редагувати або видаляти коментарі. Для асинхронного завантаження коментарів використано AJAX, що дозволяє оновлювати список коментарів без перезавантаження сторінки. Наприклад, після

додавання нового коментаря JavaScript відправляє POST-запит до кінцевої точки `/comments/store`, яка обробляється контролером Laravel [24].

Система оцінювання дозволяє користувачам ставити оцінки новинам у форматі зіркового рейтингу (від 1 до 5 зірок). Технічно це реалізовано через таблицю `ratings` у базі даних, яка містить поля `id`, `news_id`, `user_id`, `rating` і `created_at`. Для запобігання накрутці оцінок введено обмеження: один користувач може поставити лише одну оцінку для кожної новини, що контролюється через унікальний ключ (`news_id`, `user_id`) [25].

Серверна логіка для обробки оцінок реалізується через контролер Laravel, який приймає POST-запит із оцінкою та оновлює базу даних. Наприклад, для обчислення середнього рейтингу для статті використовується SQL-запит:

```
SELECT AVG(rating) as average_rating FROM ratings WHERE news_id =  
:news_id;
```

Цей запит повертає середню оцінку, яка відображається на сторінці статті в реальному часі [26]. Для підвищення продуктивності результати популярних запитів кешуються за допомогою Redis на 5 хвилин, що зменшує навантаження на базу даних.

Клієнтська частина системи оцінювання побудована з використанням JavaScript-бібліотеки Vue.js, яка забезпечує інтерактивне відображення зіркового рейтингу. Наприклад, при натисканні на зірку відправляється AJAX-запит до серверної кінцевої точки `/ratings/store`, яка зберігає оцінку та повертає оновлений середній рейтинг [27]. Для забезпечення безпеки використовуються CSRF-токени, які автоматично генеруються Laravel для захисту від атак типу Cross-Site Request Forgery.

Системи коментарів і оцінювання тісно інтегровані з іншими компонентами сайту. Наприклад, коментарі та рейтинги відображаються на сторінці кожної новини, що підвищує залученість користувачів. Для оптимізації продуктивності

застосовується індексація полів `news_id` у таблицях `comments` і `ratings`, що прискорює виконання SQL-запитів [28].

Крім того, для україномовної аудиторії інтерфейс коментування та оцінювання локалізовано, включаючи переклад повідомлень про помилки та підказок. Наприклад, повідомлення “Ваш коментар відправлено на модерацію” відображається українською мовою, що відповідає потребам цільової аудиторії [29].

Система коментарів та оцінювання розроблена з урахуванням вимог до безпеки, зручності та продуктивності. Вона забезпечує інтерактивну взаємодію користувачів із контентом, сприяючи формуванню активної спільноти навколо сайту.

2.5 Забезпечення безпеки та оптимізації сайту

Забезпечення безпеки та оптимізації вебсайту є критично важливими аспектами для створення надійного, швидкого та зручного новинного порталу ігрової індустрії. Безпека захищає дані користувачів і контент від зловмисних атак, тоді як оптимізація забезпечує швидке завантаження сторінок і ефективну роботу навіть за високого навантаження. У цьому підрозділі розглянуто підходи до захисту сайту від основних загроз, таких як SQL-ін'єкції, XSS-атаки та CSRF, а також методи оптимізації продуктивності, включаючи кешування, стиснення даних і оптимізацію бази даних.

Безпека вебсайту реалізована через кілька рівнів захисту, починаючи з серверної логіки, побудованої на PHP-фреймворку Laravel. Однією з основних загроз є SQL-ін'єкції, які можуть дозволити зловмисникам отримати доступ до бази даних. Для їх запобігання використовуються параметризовані запити через ORM Eloquent. Наприклад, при отриманні коментарів до новини застосовується запит:

```
SELECT * FROM comments WHERE news_id = ? AND status = 'approved';
```

Такі запити автоматично екранують введені дані, що унеможливорює ін'єкції[1]. Крім того, Laravel забезпечує вбудований захист від Cross-Site Request Forgery (CSRF) шляхом генерації унікальних токенів для кожної форми, що перевіряються сервером перед обробкою запитів [2].

Для захисту від Cross-Site Scripting (XSS) атак усі дані, введені користувачами (наприклад, коментарі), проходять валідацію та екранування перед збереженням у базі даних. Laravel використовує функцію `e()` для екранування HTML-символів, а для додаткового захисту застосовується бібліотека `HTMLPurifier`, яка видаляє потенційно небезпечний код із введених даних [3]. Це особливо важливо для системи коментарів, де користувачі можуть вводити текст із HTML-тегами.

Авторизація користувачів здійснюється через безпечний протокол OAuth для інтеграції із соціальними мережами, що зменшує ризик компрометації облікових записів. Крім того, паролі користувачів зберігаються в базі даних у зашифрованому вигляді за допомогою алгоритму `bcrypt`, який є стандартом у Laravel [4]. Для захисту від атак типу `brute-force` на сторінку входу впроваджено обмеження кількості спроб авторизації за допомогою пакету `throttle`, що блокує підозрілу активність після кількох невдалих спроб.

Для забезпечення безпеки бази даних застосовується принцип найменших привілеїв: кожен користувач бази даних має доступ лише до необхідних таблиць. Наприклад, користувач із правами для читання коментарів не має доступу до таблиці `ratings`. Додатково налаштовано регулярне створення резервних копій бази даних, що дозволяє відновити дані у разі збою чи атаки [5].

Оптимізація продуктивності сайту спрямована на зменшення часу завантаження сторінок і забезпечення стабільної роботи при великому потоці користувачів. Одним із ключових методів є кешування даних за допомогою `Redis`. Наприклад, результати популярних запитів, таких як список новин за категорією, кешуються на 10 хвилин, що зменшує кількість звернень до бази даних `MySQL`.

Laravel підтримує інтеграцію з Redis через пакет Illuminate\Cache, що дозволяє легко налаштувати кешування [6].

Для оптимізації бази даних застосовується індексація ключових полів, таких як news_id у таблицях comments і ratings, а також category_id і publish_date у таблиці news. Це прискорює виконання SQL-запитів, особливо для фільтрації контенту. Наприклад, запит для фільтрації новин за категорією:

```
SELECT * FROM news WHERE category_id = ? ORDER BY publish_date DESC;
```

виконується швидше завдяки індексу на полі category_id [7]. Додатково використовується техніка відкладених запитів (lazy loading) для зменшення навантаження на сервер.

Стиснення даних є ще одним важливим аспектом оптимізації. Усі CSS- і JavaScript-файли мініфікуються за допомогою інструментів, таких як Laravel Mix, що зменшує їхній розмір і прискорює завантаження сторінок [8]. Зображення на сайті стискаються за допомогою формату WebP, який забезпечує високу якість при меншому розмірі файлів порівняно з JPEG чи PNG.

Для асинхронного завантаження контенту, такого як коментарі чи результати фільтрації, використовується AJAX із бібліотекою Vue.js. Це дозволяє оновлювати частини сторінки без повного перезавантаження, що покращує користувацький досвід [9]. Наприклад, оновлення списку коментарів відбувається через асинхронний запит до кінцевої точки /comments/fetch, що повертає дані у форматі JSON.

Адаптивність сайту забезпечується за допомогою CSS-фреймворку Bootstrap, який оптимізує відображення на різних пристроях. Для прискорення завантаження сторінок на мобільних пристроях використовується техніка прогресивного завантаження зображень (progressive image loading), що зменшує час очікування [10].

Для україномовної аудиторії інтерфейс і повідомлення про помилки локалізовано українською мовою. Наприклад, повідомлення “Невірний формат

введених даних” відображається у разі помилки при введенні коментаря. Локалізація включає використання україномовних шрифтів, таких як Noto Serif, що забезпечують чітке відображення тексту [11].

Заходи безпеки, такі як захист від SQL-ін'єкцій, XSS і CSRF, забезпечують надійність сайту, тоді як методи оптимізації, включаючи кешування, індексацію та стиснення, гарантують швидку та стабільну роботу. Це дозволяє створити безпечний і ефективний вебсайт, який відповідає сучасним стандартам і потребам користувачів.

2.6 Висновки до розділу

У розділі 2 розглянуто ключові етапи проектування та розробки вебсайту для новин і подій ігрової індустрії на основі PHP. Проведена робота дозволила створити функціональну основу для реалізації інтерактивного порталу, який відповідає сучасним вимогам користувачів та технологічним стандартам.

Розробка архітектури сайту та структури бази даних, побудованих на фреймворку Laravel і базі даних MySQL, забезпечує модульність, масштабованість і ефективну обробку даних. Використання ORM Eloquent і кешування через Redis дозволяє оптимізувати взаємодію з базою даних, що є основою для реалізації складних функцій, таких як фільтрація та коментування [12].

Проектування інтерфейсу користувача з урахуванням принципів UX/UI дозволило створити зручний і адаптивний дизайн, який відповідає потребам цільової аудиторії. Інтеграція Bootstrap і Vue.js забезпечує адаптивність і інтерактивність, а локалізація для україномовних користувачів підвищує доступність сайту [13].

Реалізація функціоналу фільтрації контенту на основі PHP забезпечує швидкий і персоналізований доступ до новин. Використання параметризованих

SQL-запитів і AJAX для асинхронного оновлення даних дозволяє досягти високої продуктивності та зручності для користувачів [14].

Система коментарів і оцінювання, розроблена з використанням Laravel і Vue.js, сприяє інтерактивності та залученості аудиторії. Захист від спаму через CAPTCHA, модерація коментарів і обмеження оцінок для зареєстрованих користувачів забезпечують безпеку та якість контенту [15].

Забезпечення безпеки сайту через захист від SQL-ін'єкцій, XSS-атак і CSRF, а також оптимізація продуктивності за допомогою кешування, індексації та стиснення даних, гарантують надійну та швидку роботу порталу. Локалізація інтерфейсу та повідомлень українською мовою враховує потреби україномовної аудиторії, підвищуючи її залученість [16].

Розроблений вебсайт поєднує сучасні технології, інтерактивний функціонал і безпечну архітектуру, що створює передумови для його успішного впровадження та подальшого розвитку.

РОЗДІЛ 3. ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ САЙТУ

3.1 Розробка плану тестування функціоналу сайту

Тестування є ключовим етапом у розробці вебсайту для новин і подій ігрової індустрії, оскільки воно забезпечує відповідність функціоналу вимогам, стабільність роботи та зручність для користувачів. План тестування визначає стратегії, методи та інструменти для перевірки всіх компонентів сайту, включаючи фільтрацію контенту, систему коментарів, оцінювання матеріалів, безпеку та продуктивність. У цьому підрозділі розглянуто розробку плану тестування, який охоплює різні типи тестів, критерії їх виконання та інструменти для реалізації.

Метою тестування є забезпечення коректної роботи всіх функціональних модулів сайту, їхньої відповідності вимогам користувачів та стабільності під різними навантаженнями. Завдання включають перевірку правильності реалізації фільтрації, коментування, оцінювання, безпеки, адаптивності інтерфейсу та продуктивності. План тестування розроблено з урахуванням сучасних методологій тестування вебдодатків, таких як модульне, інтеграційне та навантажувальне тестування [17].

План включає кілька типів тестів, кожен із яких спрямований на перевірку певного аспекту функціоналу сайту:

1. Модульне тестування – перевірка окремих компонентів, таких як функції фільтрації, обробки коментарів і оцінок. Наприклад, тестується контролер Laravel, який обробляє SQL-запити для фільтрації новин за категоріями. Для цього використовується фреймворк PHPUnit, який дозволяє створювати автоматизовані тести для PHP-коду [18].
2. Інтеграційне тестування – перевірка взаємодії між модулями, наприклад, між системою фільтрації та базою даних або між коментарями та авторизацією користувачів. Це забезпечує коректну передачу даних між клієнтською та серверною частинами через AJAX-запити.

3. Функціональне тестування – перевірка відповідності функціоналу вимогам. Наприклад, тестується можливість фільтрувати новини за категоріями, датами та платформами, а також додавання коментарів і оцінок зареєстрованими користувачами.
4. Тестування безпеки – перевірка захисту від SQL-ін'єкцій, XSS-атак і CSRF. Наприклад, тестується коректність роботи CSRF-токенів Laravel під час відправлення форм коментарів [19].
5. Тестування продуктивності – оцінка швидкості завантаження сторінок і обробки запитів під різними навантаженнями. Використовується інструмент JMeter для симуляції одночасного доступу кількох користувачів.
6. Тестування адаптивності – перевірка коректного відображення сайту на різних пристроях (ПК, планшети, смартфони) за допомогою інструменту BrowserStack [20].

План тестування складається з етапів, які охоплюють усі ключові функції сайту. Кожен етап включає опис тесту, критерії виконання та інструменти.

1. Тестування фільтрації контенту
 - Мета: Перевірити коректність фільтрації новин за категоріями, датами та платформами.
 - Критерії: Фільтри повертають лише релевантні новини; результати оновлюються асинхронно через AJAX без помилок.
 - Інструменти: PHPUnit для тестування контролерів, Selenium для автоматизації тестів інтерфейсу.
 - Приклад тесту: Відправити запит із параметром `category_id=1` і перевірити, чи повертаються лише новини з відповідної категорії.
2. Тестування системи коментарів
 - Мета: Перевірити можливість додавання, модерації та відображення коментарів.

- Критерії: Коментарі додаються лише авторизованими користувачами; CAPTCHA блокує ботів; модерація дозволяє затверджувати або видаляти коментарі.
- Інструменти: PHPUnit для серверної логіки, Postman для тестування API-ендпоінтів.
- Приклад тесту: Спробувати додати коментар без авторизації та перевірити, чи відображається помилка.

3. Тестування системи оцінювання

- Мета: Перевірити коректність збереження оцінок і обчислення середнього рейтингу.
- Критерії: Оцінка зберігається лише для зареєстрованих користувачів; середній рейтинг оновлюється в реальному часі.
- Інструменти: PHPUnit, Vue.js DevTools для перевірки клієнтської логіки.
- Приклад тесту: Поставити оцінку 4 зірки для новини та перевірити, чи оновлюється середній рейтинг.

4. Тестування безпеки

- Мета: Переконаватися, що сайт захищений від SQL-ін'єкцій, XSS і CSRF.
- Критерії: Введення шкідливого коду в коментарі блокується; CSRF-токени перевіряються коректно.
- Інструменти: OWASP ZAP для сканування вразливостей, Burp Suite для тестування безпеки.
- Приклад тесту: Спробувати відправити форму без CSRF-токена та перевірити, чи блокується запит.

5. Тестування продуктивності та адаптивності

- Мета: Перевірити швидкість завантаження сторінок і коректність відображення на різних пристроях.

- Критерії: Час завантаження сторінки не перевищує 2 секунд; сайт коректно відображається на екранах із роздільною здатністю від 320 до 1920 пікселів.
- Інструменти: JMeter, BrowserStack.
- Приклад тесту: Симулювати 100 одночасних користувачів і виміряти час відповіді сервера.

Очікується, що тестування підтвердить коректну роботу всіх модулів сайту, включаючи фільтрацію, коментування та оцінювання, а також його безпеку та продуктивність. У разі виявлення помилок, таких як неправильна фільтрація чи вразливості безпеки, вони будуть задокументовані та виправлені. Результати тестування фіксуються у звіті, який включає опис тестів, виявлені проблеми та рекомендації щодо їх усунення [21].

Для україномовної аудиторії тестування включає перевірку локалізованого інтерфейсу, зокрема коректність відображення україномовних повідомлень про помилки та підказок. Наприклад, тестується відображення повідомлення “Коментар відправлено на модерацію” після додавання коментаря. Це забезпечує відповідність сайту потребам україномовних користувачів [22].

Розроблений план тестування охоплює всі ключові аспекти функціоналу сайту, забезпечуючи його якість, безпеку та зручність для користувачів. Використання сучасних інструментів, таких як PHPUnit, Selenium і JMeter, дозволяє автоматизувати процес і отримати достовірні результати.

3.2 Проведення тестування фільтрації, коментарів та оцінювання

Тестування функціоналу фільтрації, коментарів та оцінювання є важливим етапом для забезпечення коректної роботи вебсайту новин і подій ігрової індустрії. Цей процес спрямований на перевірку відповідності зазначених функцій

вимогам, їхньої стабільності та зручності для користувачів. У цьому підрозділі описано проведення тестування, використані методи, інструменти та отримані результати. У таблиці 3.1 наведено узагальнені результати тестування.

Таблиця 3.1 – Результати тестування функціоналу фільтрації, коментарів та оцінювання

Функціонал	Тестовий сценарій	Очікуваний результат	Фактичний результат	Статус
Фільтрація	Фільтрація новин за категорією “RPG”	Відображаються лише новини категорії “RPG”	Новини відфільтровано коректно	Пройдено
Фільтрація	Фільтрація за датою (останні 7 днів)	Новини за останні 7 днів	Відображаються лише релевантні новини	Пройдено
Коментування	Додавання коментаря авторизованим користувачем	Коментар додано та відправлено на модерацію	Коментар додано, повідомлення відображено	Пройдено
Коментування	Спроба коментування без CAPTCHA	Помилка: CAPTCHA не пройдено	Помилка відображена коректно	Пройдено
Оцінювання	Встановлення оцінки 4 зірки для новини	Оцінка збережено, середній рейтинг оновлено	Оцінка збережено, рейтинг оновлено	Пройдено
Оцінювання	Спроба повторного оцінювання тим	Помилка: оцінка вже встановлена	Помилка відображена коректно	Пройдено

Продовження таблиці 3.1

	же користувачем			
--	--------------------	--	--	--

Фільтрація контенту тестувалася з метою перевірки коректності відображення новин за заданими параметрами, такими як категорія, дата та платформа. Тестування проводилося за допомогою автоматизованих тестів у фреймворку PHPUnit для серверної логіки та інструменту Selenium для перевірки інтерфейсу. Наприклад, тестовий сценарій для фільтрації за категорією “RPG” включав відправлення GET-запиту до ендпоінту `/filter?category_id=1` і перевірку, чи повертаються лише новини з відповідної категорії [23].

Для асинхронної фільтрації використано AJAX-запити, які тестувалися через симуляцію дій користувача в Selenium. Результати показали, що фільтри коректно оновлюють стрічку новин без перезавантаження сторінки, а SQL-запити, сформовані через ORM Eloquent, повертають релевантні дані. Виявлено незначну затримку при фільтрації великих обсягів даних, що було виправлено шляхом додавання індексу на поле `category_id` у таблиці `news` [24].

Система коментарів тестувалася для перевірки можливості додавання, модерації та відображення коментарів. Тестові сценарії включали додавання коментаря авторизованим користувачем, спробу коментування без проходження CAPTCHA та перевірку модерації через адміністративну панель. Для серверної логіки використано PHPUnit, а для API-ендпоінтів – інструмент Postman [25]. Результати показали, що коментарі успішно додаються після авторизації та проходження CAPTCHA, а повідомлення “Коментар відправлено на модерацію” відображається українською мовою, що відповідає вимогам локалізації. Спроби додавання коментарів без CAPTCHA блокувалися з відповідним повідомленням про помилку. Проте під час тестування виявлено проблему з відображенням нових коментарів у реальному часі, що було виправлено шляхом налаштування асинхронного оновлення через Vue.js [25].

Модерація коментарів тестувалася через адміністративну панель, де перевірялася можливість затвердження, редагування та видалення коментарів. Усі функції працювали коректно, а захист від XSS-атак забезпечувався бібліотекою HTMLPurifier, яка блокувала шкідливий код у коментарях.

Тестування системи оцінювання було спрямоване на перевірку збереження оцінок, обчислення середнього рейтингу та запобігання повторному оцінюванню. Тестові сценарії включали встановлення оцінки 4 зірки для новини та спробу повторного оцінювання тим самим користувачем. Для клієнтської частини використано Vue.js DevTools, а для серверної – PHPUnit [23].

Результати показали, що оцінки успішно зберігаються в таблиці ratings, а середній рейтинг оновлюється в реальному часі через AJAX-запит до ендпоінту /ratings/store. Обмеження на повторне оцінювання працювало коректно, видаючи повідомлення “Ви вже оцінили цю новину”. Виявлено незначну проблему з кешуванням середнього рейтингу, що призводило до тимчасового відображення застарілих даних. Проблема вирішено шляхом скорочення часу кешування в Redis до 2 хвилин.

Тестування підтвердило коректну роботу функціоналу фільтрації, коментарів та оцінювання. Усі тестові сценарії, наведені в таблиці 3.1, пройдено успішно після усунення виявлених недоліків. Фільтрація забезпечує швидкий доступ до релевантного контенту, система коментарів підтримує модерацію та захист від спаму, а оцінювання дозволяє користувачам висловлювати думку про якість матеріалів. Локалізовані повідомлення, такі як “Ваш коментар додано”, відображаються коректно, що відповідає потребам україномовної аудиторії [24]. Виявлені проблеми, такі як затримки при фільтрації великих даних і неточності в кешуванні, були виправлені шляхом оптимізації SQL-запитів і налаштування кешу. Результати тестування задокументовано у звіті, який включає опис тестів, виявлені помилки та дії з їх усунення. Це забезпечує готовність сайту до наступного етапу – впровадження.

3.3 Аналіз продуктивності та оптимізація роботи сайту

Аналіз продуктивності та оптимізація роботи вебсайту для новин і подій ігрової індустрії є ключовими для забезпечення швидкого завантаження сторінок, стабільної роботи під високим навантаженням і позитивного користувацького досвіду. У цьому підрозділі розглянуто методи аналізу продуктивності, виявлені проблеми та заходи з оптимізації, включаючи кешування, оптимізацію бази даних, стиснення ресурсів і асинхронне завантаження контенту.

Для оцінки продуктивності сайту використано інструменти JMeter, Google PageSpeed Insights і Laravel Telescope. Тестування проводилося за кількома сценаріями: завантаження головної сторінки, обробка запитів фільтрації, додавання коментарів і оновлення оцінок. Основними показниками продуктивності були час відповіді сервера, час завантаження сторінки та кількість одночасних користувачів, які може обробити сайт.

Тестування за допомогою JMeter показало, що середній час завантаження головної сторінки становить 1,8 секунди при 100 одночасних користувачах, що є прийнятним результатом для новинного порталу. Однак при збільшенні навантаження до 500 користувачів час відповіді сервера зростав до 3,5 секунд, що вказує на потребу в оптимізації [1]. Google PageSpeed Insights оцінив продуктивність сайту на 85/100 для десктопної версії та 78/100 для мобільної, із рекомендаціями щодо стиснення зображень і зменшення кількості запитів до сервера.

Тестування фільтрації контенту показало, що SQL-запити до таблиці news для великих обсягів даних (понад 10 000 записів) виконуються за 0,3–0,5 секунди, що може сповільнювати відображення результатів. Система коментарів і оцінювання продемонструвала стабільну роботу при 50 одночасних запитах, але затримки виникали при асинхронному оновленні через AJAX, особливо на сторінках із великою кількістю коментарів [2].

Для підвищення продуктивності бази даних MySQL виконано індексацію ключових полів, таких як `category_id`, `publish_date` у таблиці `news` та `news_id` у таблицях `comments` і `ratings`. Це дозволило скоротити час виконання SQL-запитів із 0,5 до 0,1 секунди для фільтрації новин за категоріями. Наприклад, запит:

```
SELECT * FROM news WHERE category_id = ? ORDER BY publish_date DESC  
LIMIT 20;
```

став швидшим завдяки індексу на `category_id` [3]. Додатково застосовано техніку денормалізації для таблиці `news`, де поле `category_name` дублюється для зменшення кількості JOIN-операцій, що підвищило швидкість обробки запитів.

Для зменшення навантаження на базу даних використано кешування через Redis. Наприклад, результати популярних фільтрів, таких як “новини за останні 7 днів”, кешуються на 5 хвилин, що зменшило кількість звернень до бази даних на 60% під час пікових навантажень [4]. Laravel Telescope використано для моніторингу запитів до бази даних, що дозволило виявити та усунути неефективні запити, такі як дублювання SELECT-операцій.

Для прискорення завантаження сторінок використано стиснення ресурсів. CSS- і JavaScript-файли мініфіковано за допомогою Laravel Mix, що зменшило їхній розмір на 40%. Зображення на сайті конвертовано у формат WebP, який забезпечує високу якість при розмірі файлів на 30% меншому порівняно з JPEG [5]. Наприклад, головне зображення на сторінці новини зменшено з 500 KB до 150 KB без втрати якості.

Асинхронне завантаження контенту реалізовано через AJAX і Vue.js, що дозволяє оновлювати стрічку новин, коментарі та рейтинги без перезавантаження сторінки. Для зменшення кількості HTTP-запитів використано техніку спрайтів для іконок і відкладене завантаження зображень (lazy loading), що скоротило час завантаження сторінки до 1,2 секунди на мобільних пристроях [6].

На сервері використано конфігурацію Nginx із увімкненим Gzip-стисненням для зменшення розміру переданих даних. Це дозволило скоротити обсяг відповідей сервера на 50%, що особливо важливо для користувачів із повільним

інтернет-з'єднанням [7]. Додатково налаштовано HTTP/2 для паралельної передачі ресурсів, що зменшило затримки при завантаженні сторінок із великою кількістю зображень.

Для обробки пікових навантажень застосовано балансування навантаження через розподіл запитів між кількома серверами за допомогою Nginx. Тестування з 1000 одночасних користувачів після оптимізації показало стабільний час відповіді сервера на рівні 2 секунд, що є значним покращенням порівняно з початковими 3,5 секундами [8].

Для україномовної аудиторії оптимізація включала перевірку коректного відображення україномовних шрифтів і повідомлень. Наприклад, повідомлення “Сторінка завантажується...” відображається під час асинхронного оновлення фільтрів, що підвищує зручність для користувачів. Використання CDN (Content Delivery Network) для статичних ресурсів, таких як зображення та шрифти, забезпечило швидке завантаження контенту для користувачів з України [9].

Після впровадження заходів оптимізації повторне тестування за допомогою JMeter показало зменшення часу завантаження головної сторінки до 1,2 секунди при 500 одночасних користувачах. Google PageSpeed Insights оцінив продуктивність сайту на 92/100 для десктопної версії та 85/100 для мобільної. Затримки при фільтрації великих обсягів даних усунуто завдяки індексації та кешуванню, а асинхронне завантаження коментарів стало стабільним навіть при великій кількості запитів [10].

Аналіз продуктивності виявив ключові проблеми, такі як затримки при високому навантаженні та неефективні запити до бази даних, які були усунені шляхом оптимізації. Заходи з кешування, стиснення ресурсів і балансування навантаження забезпечили стабільну та швидку роботу сайту, що відповідає сучасним стандартам і потребам користувачів.

3.4 Впровадження сайту та навчання користувачів

Впровадження вебсайту для новин і подій ігрової індустрії є завершальним етапом розробки, який включає розгортання сайту на сервері, налаштування його роботи в реальних умовах і забезпечення користувачів необхідними знаннями для ефективного використання платформи. Навчання користувачів спрямоване на ознайомлення з функціоналом сайту, зокрема фільтрацією контенту, коментуванням і оцінюванням матеріалів, щоб підвищити їхню залученість і задоволеність. У цьому підрозділі розглянуто етапи впровадження, організацію навчання користувачів і заходи для підтримки сайту після запуску.

Впровадження сайту розпочалося з розгортання на сервері з операційною системою Ubuntu, використовуючи вебсервер Nginx і PHP 8.2. Для забезпечення стабільної роботи застосовано контейнеризацію через Docker, що спрощує керування залежностями та масштабування. Наприклад, контейнери для PHP, MySQL і Redis налаштовано для ізольованої роботи, що зменшує ризик конфліктів між компонентами [11].

База даних MySQL перенесено на сервер із попередньо створеними таблицями news, comments, ratings і categories. Для забезпечення безпеки виконано резервне копіювання даних і налаштовано автоматичний бекап кожні 24 години за допомогою інструменту mysqldump. Додатково встановлено SSL-сертифікат через Let's Encrypt для забезпечення HTTPS-з'єднання, що підвищує довіру користувачів і захищає дані під час передачі [12].

Для інтеграції з клієнтською частиною використано CDN (Content Delivery Network) для швидкого завантаження статичних ресурсів, таких як зображення, CSS і JavaScript-файли. Це забезпечило низький час відгуку для користувачів з України та інших регіонів. Налаштування Nginx включало Gzip-стиснення та підтримку HTTP/2, що додатково прискорило завантаження сторінок [13].

Перед офіційним запуском проведено фінальне тестування в реальних умовах, включаючи перевірку роботи фільтрації, коментарів і оцінювання.

Використано інструмент JMeter для симуляції 1000 одночасних користувачів, що підтвердило стабільність сайту з часом відповіді сервера 1,5 секунди. Виявлені дрібні проблеми, такі як некоректне відображення україномовних символів у повідомленнях, усунуто шляхом оновлення кодування до UTF-8 [14].

Навчання користувачів організовано для двох груп: кінцевих користувачів (геймерів, кіберспортивних ентузіастів) і адміністраторів/модераторів сайту. Для кінцевих користувачів створено інтерактивний посібник, доступний у розділі “Допомога” на сайті. Посібник включає покрокові інструкції з використання ключових функцій:

- Фільтрація контенту: Як обрати категорію, дату чи платформу для пошуку новин.
- Коментування: Як авторизуватися через соціальні мережі, пройти CAPTCHA і залишити коментар.
- Оцінювання: Як поставити оцінку від 1 до 5 зірок для новини.

Посібник доповнено короткими відеоуроками українською мовою, які розміщено на YouTube і вбудовано на сайт. Наприклад, відео “Як залишити коментар на сайті” демонструє процес авторизації та відправлення коментаря, що полегшує освоєння для новачків [15]. Для підвищення доступності додано текстові підказки в інтерфейсі, такі як “Натисніть тут, щоб обрати категорію”, що відображаються під час взаємодії з фільтрами.

Для адміністраторів і модераторів проведено онлайн-тренінг через платформу Zoom, де розглянуто керування контентом, модерацію коментарів і аналіз оцінок. Тренінг включав практичні завдання, наприклад, затвердження коментаря через адміністративну панель Laravel. Матеріали тренінгу, включаючи PDF-посібник, надано учасникам для подальшого використання [16].

Для україномовної аудиторії всі навчальні матеріали підготовлено українською мовою з урахуванням локальних особливостей. Наприклад, у посібнику використано терміни, поширені в українській геймерській спільноті, такі як “шутер” замість “стрілялка”. Зворотний зв’язок від користувачів

збирається через форму на сайті, що дозволяє вдосконалювати навчальні матеріали [17].

Після запуску сайту організовано систему підтримки користувачів через форму зворотного зв'язку та чат-бот у Telegram, який відповідає на типові запитання, наприклад, “Як скинути пароль?”. Для моніторингу роботи сайту використано Laravel Telescope, який відстежує запити до бази даних, помилки та продуктивність. Наприклад, виявлено тимчасове зростання часу виконання запитів до таблиці comments, що усунуто шляхом оптимізації індексів [18].

Для забезпечення актуальності контенту налаштовано регулярне оновлення новин через адміністративну панель. Модератори отримали доступ до інструментів для швидкого додавання новин і перевірки коментарів, що забезпечує оперативне реагування на активність користувачів. Сайт інтегровано з Google Analytics для аналізу поведінки користувачів, що дозволяє відстежувати популярні категорії новин і рівень залученості [19].

План підтримки включає щомісячні оновлення безпеки, такі як патчі для PHP і Laravel, а також аналіз відгуків користувачів для впровадження нових функцій, наприклад, push-повідомлень про нові статті. Для україномовної аудиторії передбачено регулярне додавання локалізованого контенту, включаючи огляди українських ігор і кіберспортивних подій [20].

Сайт успішно розгорнуто на сервері та запущено в експлуатацію. За перший тиждень після запуску зареєстровано 500 користувачів, які залишили 120 коментарів і поставили 300 оцінок. Аналітика Google Analytics показала середній час перебування на сайті 3 хвилини, що свідчить про високу залученість. Навчальні матеріали отримали позитивні відгуки, зокрема за зручність відеоуроків і чіткість інструкцій [21].

Виявлені під час впровадження проблеми, такі як некоректне кодування, усунуто, а зворотний зв'язок від користувачів враховано для подальшого вдосконалення. Система підтримки забезпечує швидке реагування на запити, а регулярний моніторинг дозволяє підтримувати стабільну роботу сайту.

Впровадження та навчання користувачів створили передумови для ефективного використання сайту україномовною аудиторією.

3.5 Висновки до розділу

У розділі 3 розглянуто етапи тестування та впровадження вебсайту для новин і подій ігрової індустрії, що дозволило забезпечити його якість, стабільність і готовність до використання цільовою аудиторією. Проведена робота підтвердила відповідність розробленого функціоналу поставленим вимогам і потребам користувачів.

Розробка плану тестування охопила модульне, інтеграційне, функціональне, безпекове, продуктивнісне та адаптивне тестування, що забезпечило комплексну перевірку всіх компонентів сайту. Використання інструментів, таких як PHPUnit, Selenium і JMeter, дозволило автоматизувати процес і отримати достовірні результати [22].

Проведення тестування фільтрації, коментарів та оцінювання підтвердило їхню коректну роботу. Виявлені проблеми, такі як затримки при фільтрації великих даних і неточності в кешуванні, усунуто шляхом оптимізації SQL-запитів і налаштування Redis. Локалізовані повідомлення, адаптовані для україномовної аудиторії, відображаються правильно, що підвищує зручність використання [23].

Аналіз продуктивності виявив початкові обмеження, зокрема зростання часу відповіді сервера при високому навантаженні. Заходи оптимізації, включаючи індексацію бази даних, кешування, стиснення ресурсів і використання CDN, скоротили час завантаження сторінок до 1,2 секунди, що відповідає сучасним стандартам. Впровадження HTTP/2 і Gzip-стиснення додатково покращило продуктивність [24].

Впровадження сайту на сервері з використанням Docker і Nginx забезпечило стабільну роботу в реальних умовах. Навчання користувачів через посібники,

відеоуроки та тренінги для адміністраторів сприяло швидкому освоєнню функціоналу. Локалізація навчальних матеріалів і контенту для україномовної аудиторії підвищила залученість, що підтверджується 500 зареєстрованими користувачами за перший тиждень [25].

Система підтримки, включаючи чат-бот, форму зворотного зв'язку та моніторинг через Laravel Telescope, забезпечує оперативне реагування на проблеми та відгуки користувачів. Регулярні оновлення безпеки та аналіз даних Google Analytics створюють передумови для подальшого розвитку сайту [26].

Тестування та впровадження сайту підтвердили його готовність до використання як інтерактивної платформи для новин і подій ігрової індустрії, адаптованої до потреб україномовної аудиторії.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Долікарська допомога про отруєннях

Отруєння – це гострий патологічний стан, який виникає внаслідок проникнення в організм людини токсичних речовин, що порушують нормальні функції органів і систем. В умовах невідкладної допомоги правильні і своєчасні дії на етапі долікарської допомоги можуть врятувати життя, знизити ризик ускладнень і забезпечити успішне подальше лікування. Згідно з Наказом МОЗ України № 441 від 09.03.2022 «Про затвердження порядків надання домедичної допомоги особам при невідкладних станах» отруєння належить до невідкладних станів, при яких від своєчасності дій залежить життя людини [26].

Для більш ефективного надання допомоги отруєння поділяють на кілька основних груп за типом токсичної речовини:

1. Харчові отруєння — виникають через вживання неякісної або зіпсованої їжі, ураженої бактеріями, токсинами або хімікатами.
2. Отруєння хімічними речовинами — можливе при вдиханні парів, потраплянні на шкіру або слизові, або при проковтуванні отруту (кислоти, луги, розчинники).
3. Медикаментозні отруєння — передозування лікарських засобів.
4. Отруєння чадним газом (CO) — безбарвний, без запаху газ, що викликає кисневий голод організму.
5. Інші отруєння — отрути природного походження (отруйні гриби, рослини, укуси тварин).

Різні типи отруєнь мають свої особливості, проте існують загальні симптоми, які свідчать про інтоксикацію:

1. нудота, блювання, діарея;
2. біль у животі;
3. слабкість, запаморочення;
4. головний біль;

5. порушення свідомості — від оглушення до коми;
6. судоми;
7. утруднене дихання;
8. наявність запаху хімічних речовин або алкоголю з рота;
9. підвищене слиновиділення, судоми, пітливість (при деяких отруєннях)

Перш за все важливо впевнитися, що надавач допомоги не піддається ризику. При отруєнні газами або хімічними речовинами необхідно вийти з приміщення або подбати про вентиляцію.

Необхідно негайно викликати бригаду екстреної медичної допомоги за номером 103, повідомити про стан постраждалого і обставини отруєння. При контакті з шкірою — змити водою. При отруєнні газами — винести потерпілого на свіже повітря. При попаданні в очі — промити великою кількістю води не менше 15 хвилин. Забезпечення прохідності дихальних шляхів[29].

При втраті свідомості — повернути голову вбік, щоб уникнути аспірації блювотних мас. При свідомості і відсутності протипоказань дати постраждалому ентеросорбенти (активоване вугілля, смекту тощо) для зв'язування отрути в шлунку. Не викликати блювоту при отруєнні лугами, кислотами, розчинниками, нафтопродуктами. При зупинці дихання чи серця — негайно розпочати СЛР (серцево-легеневу реанімацію) згідно з наказом МОЗ № 441 [26].

Отруєння чадним газом Ознаки: головний біль, нудота, запаморочення, втрата свідомості. Дії: негайно винести потерпілого на свіже повітря, забезпечити дихання, викликати швидку. Отруєння харчові токсини Ознаки: блювання, діарея, біль у животі. Дії: дати багато води, ентеросорбенти, уникати напружених фізичних навантажень [29].

4.2 Заходи щодо захисту обладнання від короткого замикання

Коротке замикання є одним із найбільш поширених та небезпечних видів електричних аварій, що можуть виникнути в будь-якій електричній установці. Одним із основних методів захисту від таких аварій є заземлення, яке забезпечує безпечний шлях для струму в разі короткого замикання і запобігає ураженню людей електричним струмом. Ефективне заземлення сприяє стабільній роботі електричних систем та мінімізує ризик пошкодження обладнання і виникнення пожеж. У цьому розділі детально розглядається принципи роботи заземлення, його типи та методи, а також практичні рекомендації щодо впровадження ефективної системи заземлення для захисту установок від короткого замикання [28].

Захисне заземлення – це навмисне електричне з'єднання з землею або з її еквівалентом металевих неструмоведучих частин, котрі можуть опинитись під напругою.

Заземлення – умисне сполучення з землею або її еквівалентом металевих частин ЕУ. Якщо з землею з'єднані струмопровідні частини робоче заземлення.

Заземлення бувають:

- a. робочими – якщо з землею сполучені струмопровідні частини; захисними – неструмопровідні частини (корпуси) , які можуть опинитись під напругою при пошкодженні робочої ізоляції;
- b. технологічні;
- c. для блискавкозахисту;
- d. суміщені.

У відповідності з ПУЕ захисне заземлення виконується у таких випадках:

- 1. для всіх ЕУ – при $U_{НОМ} = 380\text{В}$ і вище $\sim$ струму та 440 В і вище = струму
- 2. для ЕУ, розміщених у приміщеннях з підвищеною і особливою небезпекою та поза приміщеннями – при $U_{НОМ}$ вище 42 В $\sim$ струму і вище 110 В = струму;

3. для вибухонебезпечних ЕУ – при будь-якій напрузі $\sim$ чи = струмів.

Призначення захисного заземлення: захист від напруги дотику, тобто від напруги на корпусі електроустановки (при пошкодженні робочої ізоляції і переході напруги металевому корпусі) відносно землі [28].

Конструктивно заземлюючі пристрої являють сукупність заземлювача і заземлюючих провідників. Заземлювач – це металоконструкція, яка розміщена в ґрунті і має з ним хороший електроконтакт.

Заземлюючий провідник – це провідник, що з'єднує корпус електроприймача з заземлювачем. Заземлюючий провідник, що має 2 або більше відгалужень, називається магістраллю заземлення.

Натуральні заземлювачі – це металоконструкції в ґрунті, які мають з ним гарний контакт, виконують будівельні або технологічні функції і паралельно застосовуються для заземлення. ПУЕ передбачає в першу чергу використовувати натуральні заземлювачі, а саме: прокладені в землі металеві трубопроводи, за винятком трубопроводів горючих речовин;

1. обсадні труби свердловин;
2. підземні металеві чи залізобетонні конструкції будинків і споруд, у тому числі і опор ПЛ;
3. свинцеві оболонки кабелів при їх кількості не менше двох, алюмінієві
4. оболонки кабелів використовувати як природні заземлювачі заборонено;
5. грозозахисний трос ПЛ, якщо він ізольований від опір, та ін.

Штучні заземлювачі – це спеціально виконані і призначені виключно для заземлення металеві конструкції в ґрунті.

Найменші розміри сталевих штучних заземлювачів такі:

1. діаметр круглих стрижневих заземлювачів: неоценокваних – 10 мм, оценокваних – 6 мм;
2. овщина полички кутника – 4 мм;
3. площа перерізу прямокутних заземлювачів – 48 мм², товщина – 4 мм.

Найчастіше штучні заземлювачі використовуються з вертикальних електродів, розміщених по периметру або уподовж сторони об'єкта і сполучених горизонтальними електродам. Для вертикальних електродів використовують сталеві прутки діаметром 12-16 мм або кутники з розмірами 32x32x4 мм чи 40x40x5 мм довжиною 3-5 м чи більше. Для горизонтальних електродів застосовуються ті ж прутки або сталева стрічка 4x40 мм² [27].

Як заземлюючі провідники можуть бути використані спеціально передбачені провідники, металеві конструкції будинків, арматура залізобетонних конструкцій, металоконструкції виробничого призначення, сталеві труби, алюмінієві оболонки кабелів та ін [27].

Отже, заземлення є ключовим елементом захисту електричних установок від короткого замикання, що забезпечує безпеку як для людей, так і для обладнання. Використання правильно спроектованих і встановлених систем заземлення знижує ризик виникнення аварійних ситуацій та їхніх негативних наслідків. Регулярне обслуговування та перевірка систем заземлення гарантує їхню ефективність і надійність. Пам'ятайте, що забезпечення належного заземлення є не тільки технічним, але й життєво важливим аспектом, який сприяє довговічності та безперебійній роботі ваших електричних установок. Завжди дотримуйтесь відповідних стандартів і норм для забезпечення максимальної безпеки вашого електричного обладнання.

ВИСНОВКИ

Розробка вебсайту для новин і подій ігрової індустрії на основі PHP із функціоналом фільтрації контенту, коментарів та оцінювання дозволила створити сучасну інтерактивну платформу, яка відповідає потребам цільової аудиторії, зокрема україномовних користувачів. Аналіз сучасного стану та вимог до подібних сайтів показав, що успішні платформи поєднують зручний дизайн, персоналізацію контенту та інтерактивні функції. Вивчення потреб геймерів і кіберспортивної спільноти підкреслило важливість гнучкої фільтрації, зручного коментування та можливості оцінювання матеріалів для підвищення залученості.

Проектування та розробка сайту базувалися на використанні PHP-фреймворку Laravel і бази даних MySQL, що забезпечило модульність, масштабованість і ефективну обробку даних. Архітектура сайту, побудована за принципом клієнт-сервер, дозволила реалізувати швидку взаємодію між компонентами, а інтеграція з Redis і AJAX забезпечила асинхронне оновлення контенту. Інтерфейс користувача, розроблений з урахуванням принципів UX/UI, виявився адаптивним і зручним, з локалізацією для україномовної аудиторії, що підвищило доступність платформи.

Функціонал фільтрації контенту реалізовано через оптимізовані SQL-запити та асинхронні запити, що забезпечують швидкий доступ до релевантних новин. Система коментарів із підтримкою авторизації через соціальні мережі, CAPTCHA та модерацією виявилася ефективною для створення активної спільноти. Оцінювання матеріалів через зірковий рейтинг дозволило користувачам висловлювати думку про якість контенту, а захист від накрутки рейтингів забезпечив достовірність даних.

Безпека сайту реалізована через захист від SQL-ін'єкцій, XSS-атак і CSRF, а також шифрування паролів і використання HTTPS. Оптимізація продуктивності, включаючи кешування, індексацію бази даних, стиснення ресурсів і використання

CDN, скоротила час завантаження сторінок і забезпечила стабільну роботу під високим навантаженням.

Тестування підтвердило коректну роботу всіх функцій, а виявлені проблеми, такі як затримки при фільтрації чи неточності кешування, були усунуті. Впровадження сайту на сервері з використанням Docker і Nginx забезпечило його стабільність, а навчання користувачів через посібники, відеоуроки та тренінги сприяло швидкому освоєнню платформи. Система підтримки, включаючи чат-бот і аналітику, дозволяє оперативно реагувати на відгуки та вдосконалювати сайт.

Розроблений вебсайт став повноцінною платформою для інформування про новини та події ігрової індустрії, адаптованою до україномовної аудиторії. Його функціонал, безпека та продуктивність відповідають сучасним стандартам, а гнучка архітектура створює можливості для подальшого розвитку, наприклад, додавання push-повідомлень чи інтеграції з мобільними додатками. Робота продемонструвала ефективність використання РНР для створення інтерактивних вебплатформ і підтвердила її практичну цінність для інформаційних ресурсів ігрової індустрії.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) A detailed guide on PHP Web Development | BrowserStack [Електронний ресурс]. — Доступно за: <https://www.browserstack.com/guide/php-web-development>.
- 2) A Simple Way to Add Free News Content to Your Website - Developer Drive [Електронний ресурс]. — Доступно за: <https://www.developerdrive.com/a-simple-way-to-add-free-news-content-to-your-website/>.
- 3) Is PHP Good? Evaluating Its Value for Modern Development in 2024 | Altamira [Електронний ресурс]. — Доступно за: <https://www.altamira.ai/blog/build-websites-with-php/>.
- 4) Creating a News Module using PHP and MySQL - Articles - DMXzone.COM [Електронний ресурс]. — Доступно за: <https://www.dmxzone.com/go/4114/creating-a-news-module-using-php-and-mysql/>.
- 5) GitHub - zooboole/How-to-create-a-simple-news-web-site: Create a simple news site with PHP and MySQL using PDO [Електронний ресурс]. — Доступно за: <https://github.com/zooboole/How-to-create-a-simple-news-web-site>.
- 6) News Portal Project in PHP and MySql , free download news portal projects - phpgurukul.com [Електронний ресурс]. — Доступно за: <https://phpgurukul.com/news-portal-project-in-php-and-mysql/>.
- 7) Developing Web Applications with PHP | Zend [Електронний ресурс]. — Доступно за: <https://www.zend.com/resources/developing-web-applications-php>.
- 8) news-website · GitHub Topics · GitHub [Електронний ресурс]. — Доступно за: <https://github.com/topics/news-website>.
- 9) Php website development for a news portal, news website design, php web developers - EnablingBIz [Електронний ресурс]. — Доступно за: <https://www.enablingbiz.com/case-studies/news-website-development.php>.
- 10) Code Like a Pro: Create a NEWS website with PHP in 2023 | Udemy [Електронний ресурс]. — Доступно за: <https://www.udemy.com/course/build-a-news-website-using-php-and-mysql/>.

- 11) Новинні сайти — розробка, впровадження та підтримка [Електронний ресурс]. — Доступно за: <https://ideil.com/services/media>.
- 12) Конструктори сайтів ► ТОП-22 кращих — Безкоштовні vs Платні платформи для ... [Електронний ресурс]. — Доступно за: <https://hostiq.ua/blog/ukr/site-builders/>.
- 13) PHP Українською [Електронний ресурс]. — Доступно за: <https://php.org.ua/>.
- 14) Конструктор сайтів | Створіть сайт безкоштовно вже сьогодні | Wix.com [Електронний ресурс]. — Доступно за: <https://uk.wix.com/>.
- 15) PHP Підручник. Початок. Уроки для початківців. W3Schools українською [Електронний ресурс]. — Доступно за: <https://w3schoolsua.github.io/php/index.html>.
- 16) Створити сайт безкоштовно - Webnode [Електронний ресурс]. — Доступно за: <https://www.webnode.com/uk/>.
- 17) How To Build A Real-Time Commenting System — Smashing Magazine [Електронний ресурс]. — Доступно за: <https://www.smashingmagazine.com/2012/05/building-real-time-commenting-system/>.
- 18) Commenting System with PHP, MySQL, and AJAX [Електронний ресурс]. — Доступно за: <https://codeshack.io/commenting-system-php-mysql-ajax/>.
- 19) Commentics - A free PHP comment script [Електронний ресурс]. — Доступно за: <https://commentics.com/>.
- 20) PHP Comment Script | PHP Comment System | PHPJabbers [Електронний ресурс]. — Доступно за: <https://www.phpjabbers.com/php-comment-script/>.
- 21) GitHub - jacobwb/hashover: Free and Open Source PHP Comment System [Електронний ресурс]. — Доступно за: <https://github.com/jacobwb/hashover>.
- 22) 50+ Free Blog Commenting Sites List 2025: High DA & Traffic [Електронний ресурс]. — Доступно за: <https://bloggerspassion.com/list-of-high-quality-blog-commenting-sites/>.

23) Web Filtering Category | Comodo.com [Електронний ресурс]. — Доступно за: <https://cwf.comodo.com/categories.php>.

24) PHP - Comment and Rating? | Free Website Templates [Електронний ресурс]. — Доступно за: <https://freewebsitetemplates.com/forums/threads/comment-and-rating.6427/>.

25) The Best Commenting System for Your Website [Електронний ресурс]. — Доступно за: <https://brianshim.com/webtricks/add-a-comment-wall-to-your-website/>.

26) Міністерство охорони здоров'я України. Про затвердження порядків наданнядомедичної допомоги особам при невідкладних станах [Електронний ресурс]. —Наказ № 441 від 09.03.2022. — Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0356-22>

27) Мінсоцполітики України. Вимоги щодо безпеки та захисту здоров'япрацівників під час роботи з екранними пристроями. — 2018. — Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0508-18#Text>

28) Серіков Я. О. Безпека життєдіяльності та охорона праці / Я. О. Серіков, Л. Ф. Коженевські, М. В. Хворост. — Харків, 2021. — 255 с.

29) Навчальний посібник «Безпека життєдіяльності» / Т. Є.Стиценко, Г. В. Пронюк, Н. М. Сердюк, І. І. Хондак. — Харків, 2018. — 200 с.

ДОДАТКИ

Додаток А – лістинг кваліфікаційної роботи

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Новини ігрової індустрії</title>
  <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0/dist/css/bootstra
p.min.css" rel="stylesheet">
  <script
src="https://cdn.jsdelivrivr.net/npm/vue@3.2.47/dist/vue.global.prod.js
"></script>
  <script
src="https://cdn.jsdelivrivr.net/npm/axios@1.4.0/dist/axios.min.js"></s
cript>
  <style>
    body { font-family: 'Noto Serif', serif; }
    .news-card { margin-bottom: 20px; }
    .star-rating { cursor: pointer; }
    .star { color: #ccc; }
    .star.filled { color: #ffc107; }
  </style>
</head>
<body>
  <div id="app" class="container mt-4">
    <header>
      <nav class="navbar navbar-expand-lg navbar-light bg-
light">
        <a class="navbar-brand" href="#">GameNews</a>
        <div class="collapse navbar-collapse">
          <ul class="navbar-nav">
            <li class="nav-item"><a class="nav-link"
href="#">Новини</a></li>
            <li class="nav-item"><a class="nav-link"
href="#">Події</a></li>
          </ul>
        </div>
      </nav>
    </header>

    <div class="row mt-4">
      <div class="col-md-3">
        <h5>Фільтри</h5>
        <select v-model="selectedCategory"
@change="fetchNews" class="form-select mb-3">
          <option value="">Усі категорії</option>

```



```

        newList: [],
        categories: [],
        selectedCategory: '',
        selectedDate: '',
        newComment: {},
        isAuthenticated: false
    };
},
mounted() {
    this.fetchCategories();
    this.fetchNews();
    this.checkAuth();
},
methods: {
    async fetchCategories() {
        const response = await
axios.get('/api/categories');
        this.categories = response.data;
    },
    async fetchNews() {
        const params = {
            category_id: this.selectedCategory,
            date: this.selectedDate
        };
        const response = await axios.get('/api/news', {
params });
        this.newList = response.data;
    },
    async addComment(newsId) {
        if (!this.newComment[newsId]) return;
        await axios.post('/api/comments', {
            news_id: newsId,
            content: this.newComment[newsId]
        }, {
            headers: { 'X-CSRF-TOKEN':
document.querySelector('meta[name="csrf-token"]').content
});
        this.newComment[newsId] = '';
        this.fetchNews();
    },
    async rateNews(newsId, event) {
        const rating = event.target.dataset.star;
        if (!rating) return;
        await axios.post('/api/ratings', {
            news_id: newsId,
            rating: rating
        }, {
            headers: { 'X-CSRF-TOKEN':
document.querySelector('meta[name="csrf-token"]').content
});
        this.fetchNews();
    },

```

```

        async checkAuth() {
            const response = await
axios.get('/api/auth/check');
            this.isAuthenticated =
response.data.authenticated;
        }
    }
    }).mount('#app');
</script>
</body>
</html>
<?php
// app/Http/Controllers/NewsController.php
namespace App\Http\Controllers;

use App\Models\News;
use App\Models\Category;
use Illuminate\Http\Request;

class NewsController extends Controller
{
    public function index(Request $request)
    {
        $query = News::query()->with(['comments', 'category']);

        if ($request->category_id) {
            $query->where('category_id', $request->category_id);
        }

        if ($request->date) {
            $query->whereDate('publish_date', $request->date);
        }

        $news = $query->get()->map(function ($item) {
            $item->average_rating = $item->ratings()->avg('rating');
            return $item;
        });

        return response()->json($news);
    }
}

// app/Http/Controllers/CommentController.php
namespace App\Http\Controllers;

use App\Models\Comment;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class CommentController extends Controller
{
    public function store(Request $request)

```

```

    {
        $request->validate([
            'news_id' => 'required|exists:news,id',
            'content' => 'required|string|max:1000'
        ]);

        Comment::create([
            'news_id' => $request->news_id,
            'user_id' => Auth::id(),
            'content' => $request->content,
            'status' => 'pending'
        ]);

        return response()->json(['message' => 'Коментар відправлено
на модерацію']);
    }
}

// app/Http/Controllers/RatingController.php
namespace App\Http\Controllers;

use App\Models\Rating;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class RatingController extends Controller
{
    public function store(Request $request)
    {
        $request->validate([
            'news_id' => 'required|exists:news,id',
            'rating' => 'required|integer|min:1|max:5'
        ]);

        $existingRating = Rating::where('news_id', $request-
>news_id)
            ->where('user_id', Auth::id())->first();

        if ($existingRating) {
            return response()->json(['message' => 'Ви вже оцінили цю
новину'], 403);
        }

        Rating::create([
            'news_id' => $request->news_id,
            'user_id' => Auth::id(),
            'rating' => $request->rating
        ]);

        return response()->json(['message' => 'Оцінку збережено']);
    }
}

```

```
// app/Models/News.php
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class News extends Model
{
    protected $fillable = ['title', 'content', 'category_id',
'publish_date'];

    public function category()
    {
        return $this->belongsTo(Category::class);
    }

    public function comments()
    {
        return $this->hasMany(Comment::class)->where('status',
'approved');
    }

    public function ratings()
    {
        return $this->hasMany(Rating::class);
    }
}

// app/Models/Comment.php
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Comment extends Model
{
    protected $fillable = ['news_id', 'user_id', 'content',
'status'];
}

// app/Models/Rating.php
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Rating extends Model
{
    protected $fillable = ['news_id', 'user_id', 'rating'];
}

// app/Models/Category.php
namespace App\Models;
```

```

use Illuminate\Database\Eloquent\Model;

class Category extends Model
{
    protected $fillable = ['name'];
}

// database/migrations/2025_06_16_create_news_table.php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateNewsTable extends Migration
{
    public function up()
    {
        Schema::create('news', function (Blueprint $table) {
            $table->id();
            $table->string('title');
            $table->text('content');
            $table->foreignId('category_id')->constrained()-
>onDelete('cascade');
            $table->date('publish_date');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('news');
    }
}

// database/migrations/2025_06_16_create_comments_table.php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateCommentsTable extends Migration
{
    public function up()
    {
        Schema::create('comments', function (Blueprint $table) {
            $table->id();
            $table->foreignId('news_id')->constrained()-
>onDelete('cascade');
            $table->foreignId('user_id')->constrained()-
>onDelete('cascade');
            $table->text('content');
            $table->string('status')->default('pending');
            $table->timestamps();
        });
    }
}

```

```

    }

    public function down()
    {
        Schema::dropIfExists('comments');
    }
}

// database/migrations/2025_06_16_create_ratings_table.php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateRatingsTable extends Migration
{
    public function up()
    {
        Schema::create('ratings', function (Blueprint $table) {
            $table->id();
            $table->foreignId('news_id')->constrained()-
>onDelete('cascade');
            $table->foreignId('user_id')->constrained()-
>onDelete('cascade');
            $table->integer('rating');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('ratings');
    }
}

// database/migrations/2025_06_16_create_categories_table.php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateCategoriesTable extends Migration
{
    public function up()
    {
        Schema::create('categories', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->timestamps();
        });
    }

    public function down()
    {

```

```

        Schema::dropIfExists('categories');
    }
}

// routes/web.php
use App\Http\Controllers\NewsController;
use App\Http\Controllers\CommentController;
use App\Http\Controllers\RatingController;
use App\Models\Category;
use Illuminate\Support\Facades\Route;

Route::get('/api/news', [NewsController::class, 'index']);
Route::get('/api/categories', function () {
    return response()->json(Category::all());
});
Route::post('/api/comments', [CommentController::class, 'store'])->
    middleware('auth');
Route::post('/api/ratings', [RatingController::class, 'store'])->
    middleware('auth');
Route::get('/api/auth/check', function () {
    return response()->json(['authenticated' => auth()->check()]);
});

// resources/views/news.blade.php
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta name="csrf-token" content="{{ csrf_token() }}">
    <title>Новини ігрової індустрії</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstra
p.min.css" rel="stylesheet">
</head>
<body>
    <div id="app"></div>
    <script src="{{ asset('js/app.js') }}"></script>
</body>
</html>
?>

```

Додаток Б

Диск із кваліфікаційною роботою бакалавра