

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 122 «Комп'ютерні науки»
(освітня програма «Комп'ютерна графіка та розробка ігор»)
на тему:

«Розробка веб-блогу ігор»

Виконав студент групи ЗКН-19-1
КОВАЛЬ Артем Андрійович

Керівник роботи:
КУЗЬМЕНКО Олександр Вікторович

Рецензент:
КОРОТУН Ольга Володимирівна

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

Марина ГРАФ

«05» січня 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу освітнього ступеня «бакалавр»
за спеціальністю 122 «Комп'ютерні науки»
(освітня програма «Комп'ютерна графіка та розробка ігор»)

Здобувач вищої освіти: **КОВАЛЬ Артем Андрійович**

Керівник роботи: **КУЗЬМЕНКО Олександр Вікторович**

Тема роботи: **«Розробка веб-блогу ігор»,**

затверджена наказом закладу вищої освіти від **«05» січня 2023 р., №11/с**

Термін здачі закінченої роботи: **«10» червня 2023 р.**

Вихідні дані роботи (зазначається предмет і об'єкт дослідження): об'єктом дослідження є процес проектування та розробки веб-блогу ігор. Предметом дослідження є використання рушія веб-технологій та об'єктно-орієнтованого підходу для створення блогу ігор.

Консультанти випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Дата	
		Завдання видав	Завдання прийняв

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі. Пошук, огляд та аналіз аналогічних розробок. Формулювання технічного завдання. Опрацювання літературних джерел.	2 лютого 2023 – 5 березня 2023	Виконано
2	Проектування структури системи	6 березня 2023 – 25 березня 2023	Виконано
3	Написання програмного коду	26 березня 2023 – 26 квітня 2023	Виконано
4	Тестування системи	27 квітня 2023 – 14 травня 2023	Виконано
5	Оформлення пояснювальної записки	15 травня 2023 – 1 червня 2023	
6	Попередній захист	1 червня 2023	Виконано

Здобувач вищої освіти **Артем КОВАЛЬ**

Керівник **Олександр КУЗЬМЕНКО**

РЕФЕРАТ

Кваліфікаційна робота складається з вихідного коду, SQL-файлу для імпорту структури БД та тестових даних, пояснювальної записки. Пояснювальна записка до випускної роботи має 68 сторінок друкованого тексту і містить 20 рисунків та 10 таблиць. Список використаних джерел містить 21 найменувань.

Метою кваліфікаційної роботи є розробка web-системи, що надаватиме можливість створення блогів з автоматичною категоризацією записів та забезпечуватиме зручний функціонал для їх просування.

Кваліфікаційна робота складається з програмного комплексу web-системи блогу та пояснювальної записки.

В пояснювальній записці до випускної кваліфікаційної роботи визначені головні завдання для розробки системи, проаналізовано аналоги розробленої системи. Наведено сценарії роботи програмного комплексу, загальну структуру програмного комплексу, опис алгоритмів взаємодії окремих модулів системи.

КЛЮЧОВІ СЛОВА: MVC, PHP, CMS, БД, DB, веб-блог, авторизація.

					КН.КР.Б – 122 – 23 – ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.	А.А. КОВАЛЬ				Розробка веб-блогу ігор Пояснювальна записка			Літ.	Арк.	Аркушів	
Керівник	О.В. КУЗЬМЕНКО									3	67
Рецензент	О.В.КОРОТУН							Житомирська політехніка, група ЗКН-19-1			
Н. контр.											
Зав. каф.	М.С. ГРАФ										

ABSTRACT

The qualification work consists of the source code, an SQL file for importing the database structure and test data, and an explanatory note. The explanatory note to the final thesis has 68 pages of printed text and contains 20 figures and 10 tables. The list of used sources contains 21 items.

The goal of the qualification work is the development of a web system that will provide the possibility of creating blogs with automatic categorization of entries and provide convenient functionality for their promotion.

The qualification work consists of the software complex of the blog web system and an explanatory note.

In the explanatory note to the final qualification work, the main tasks for the development of the system are defined, and analogues of the developed system are analyzed. The scenarios of the software complex, the general structure of the software complex, the description of the interaction algorithms of individual system modules are given.

KEYWORDS: MVC, PHP, CMS, DB, DB, weblog, authorization.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

ЗМІСТ

РЕФЕРАТ	4
ВСТУП	7
Розділ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ БЛОГІВ ТА ЗАСОБІВ ЇХ АНАЛІЗУ	8
1.1. Аналіз існуючого програмного забезпечення за тематикою роботи	8
1.2. Постановка задачі	11
1.3. Вибір архітектури web-орієнтованої системи	12
1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	14
Висновки до першого розділу	18
Розділ 2. ПРОЕКТУВАННЯ СИСТЕМИ	20
2.1. Визначення варіантів використання та структури системи	20
2.2. Розробка бази даних	23
2.3. Реалізація додатку web-орієнтованої системи	26
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ	31
3.1. Порядок встановлення та налаштування системи.	31
3.2. Опис роботи з програмним додатком	32
3.3. Тестування роботи системи	35
Висновки до третього розділу	38
ВИСНОВКИ	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	40
ДОДАТКИ	42

ВСТУП

Актуальність теми.

У сучасному світі відеоігри стали невід’ємною частиною майже кожної людини. Вони не тільки розважають нас, але й стають засобом втілення фантазій, випробовування нашої витривалості та розвитку наших навичок. Відеоігри надають гравцям захоплюючий та неповторний досвід вирішення задач у тяжких умовах та небезпечному середовищі. Деякі ігри вимагають від гравця витримки, стратегічного мислення та вміння приймати рішення в умовах обмежених ресурсів та небезпеки. У таких іграх гравець зазнає випробовування, де його навички у виживанні, збиранні ресурсів, створенні предметів, побудові споруд, взаємодії з вигаданим світом стають вирішальними для його успіху.

В наш час ігри та їх поняття зустрічаються у людей по всьому світу саме тому це грає велику роль, тому що технології пішли набагато вперед і саме тому ми зможемо належним чином відчувати все прекрасне що можуть дати ігри а ще люди люблять ділитися своїми досягненнями в інтернеті і саме тому для таких людей будь яка частина їх особливості дуже важлива, від матеріальних речей до фізичних. Люди люблять слідкувати за різними подіями які відбуваються в цих іграх і саме тому їм на допомогу приходять Блоги в яких можна сказати зберігається історія ігрової індустрії. Саме тому це частина життя людини яку просто так змінити неможливо.

Мета роботи є розробка власного веб-блогу та засобів для управління його контентом.

Завдання та дослідження: реалізувати cms-систему за допомогою патерну mvc, яка буде керувати сторінками та вмістом сайту.

Об’єктом дослідження є процес проектування та розробки веб-блогу ігор

Предметом дослідження є використання рушія веб-технологій та об’єктно-орієнтованого підходу для створення блогу ігор.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ БЛОГІВ ТА ЗАСОБІВ ЇХ АНАЛІЗУ

1.1. Аналіз існуючого програмного забезпечення за тематикою роботи

У ході пошуку існуючого програмного забезпечення було знайдено та проаналізовано такі веб сайти:

GameWAY

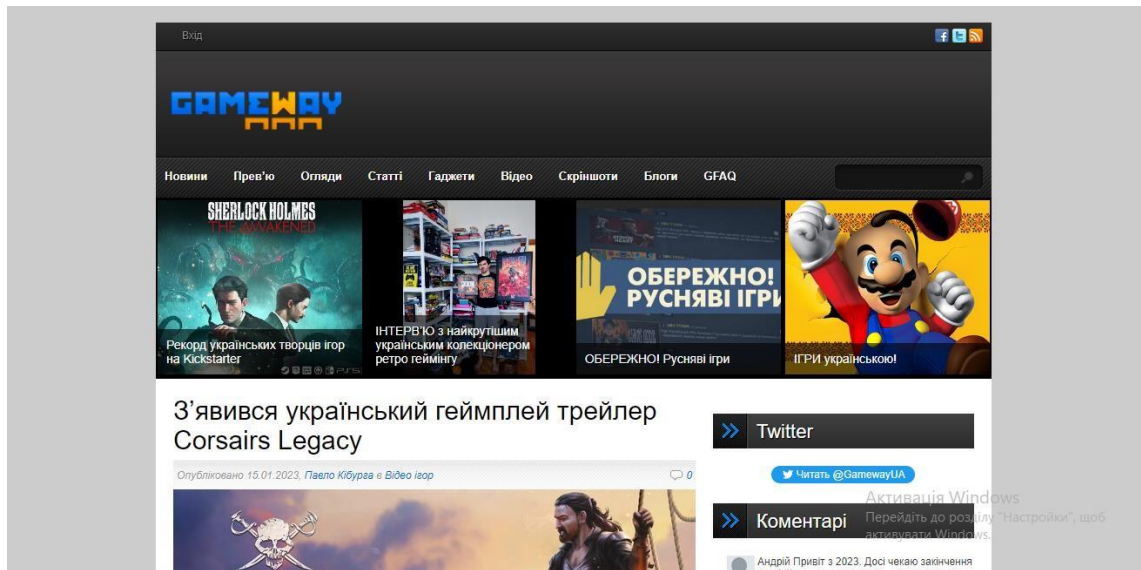


Рис 1.1 Головна сторінка сайту

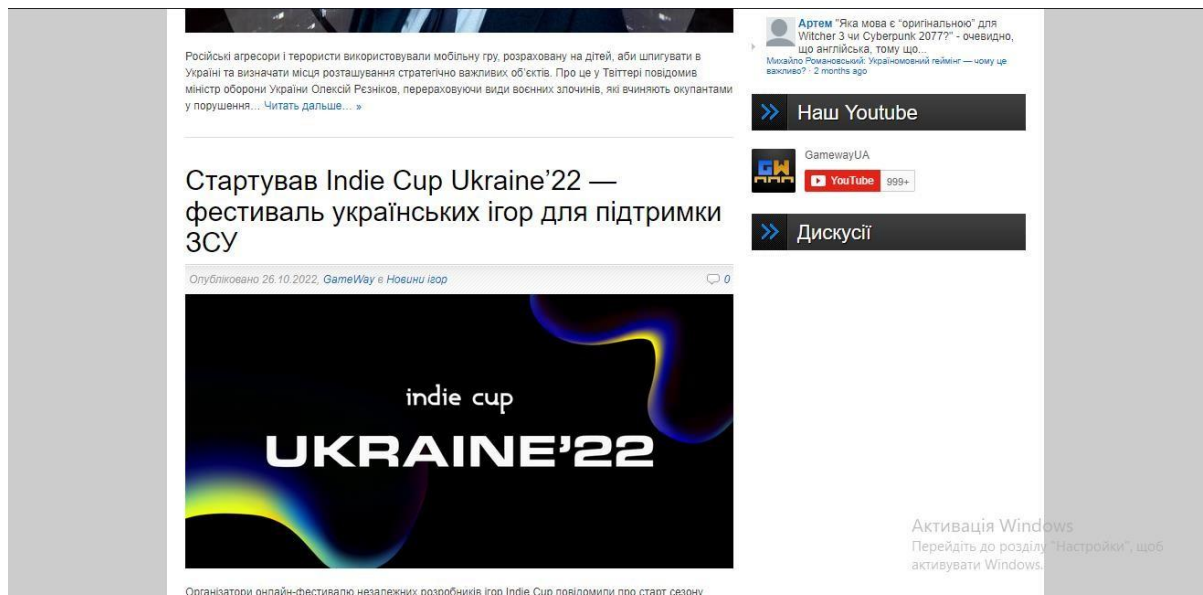


Рис 1.2 Блок новин сайту

Переваги:

- Зрозуміле меню

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

- Хороший дизайн

Недоліки:

- Складно знайти новину яка зацікавить
- GameGuru

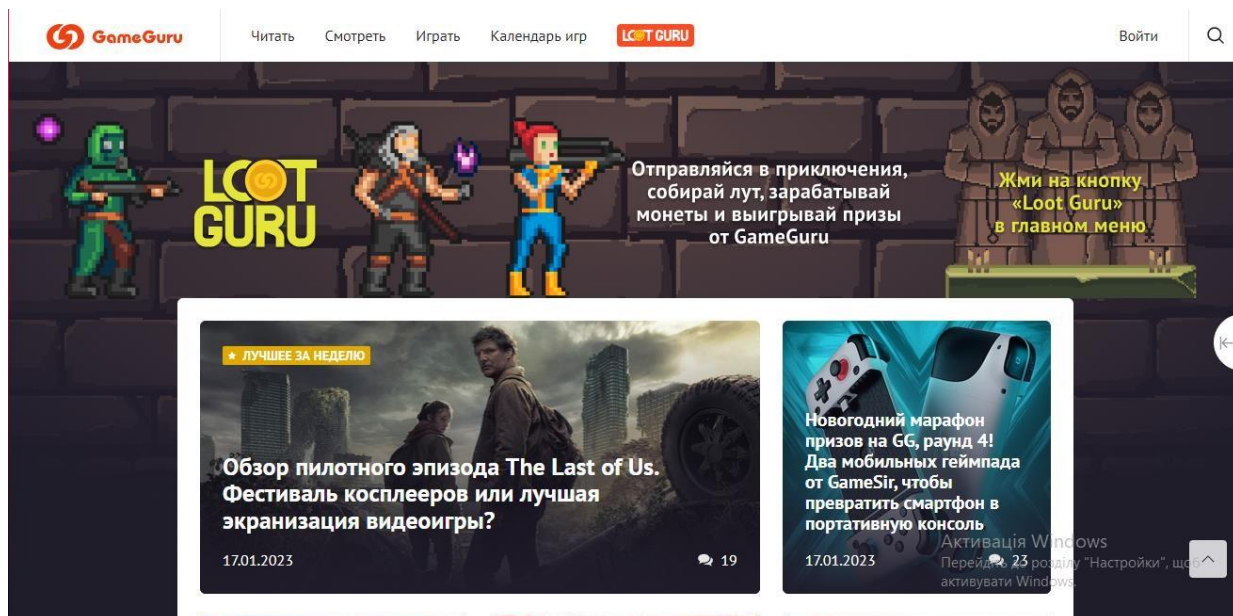


Рис 1.3 Головна сторінка сайту

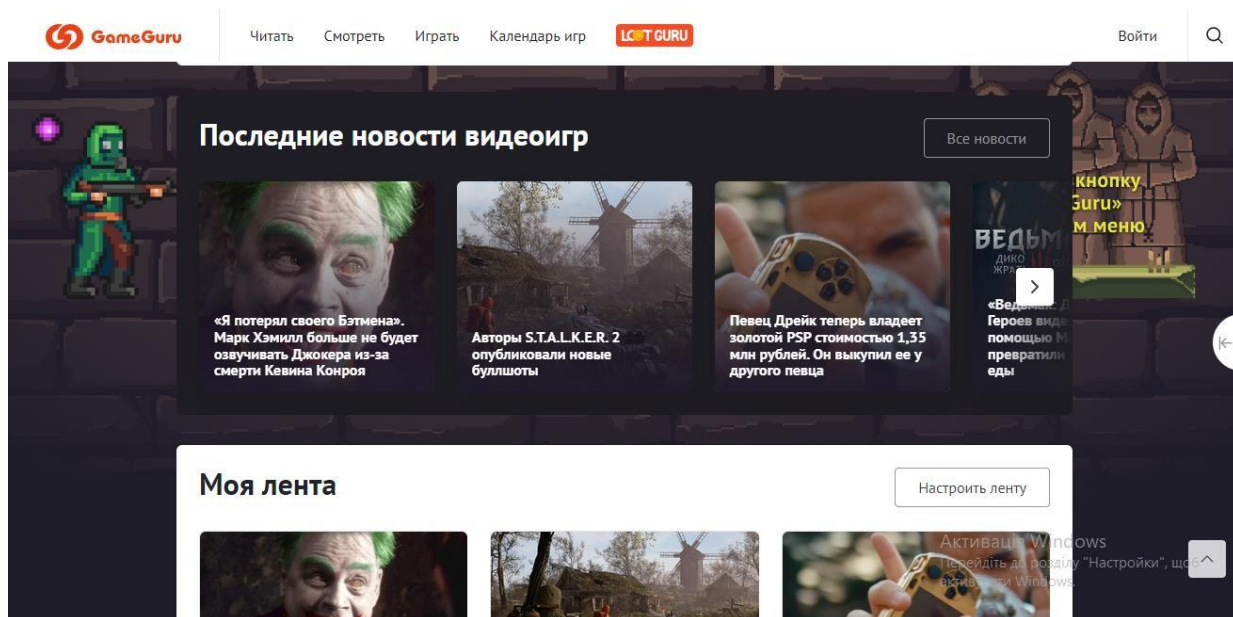


Рис 1.4 Новини веб сайту

Переваги:

- Добре стилізований сайт
- Зрозуміле меню

Недоліки:

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

- Неможливо дивитися новини без реєстрації
- GameInfo

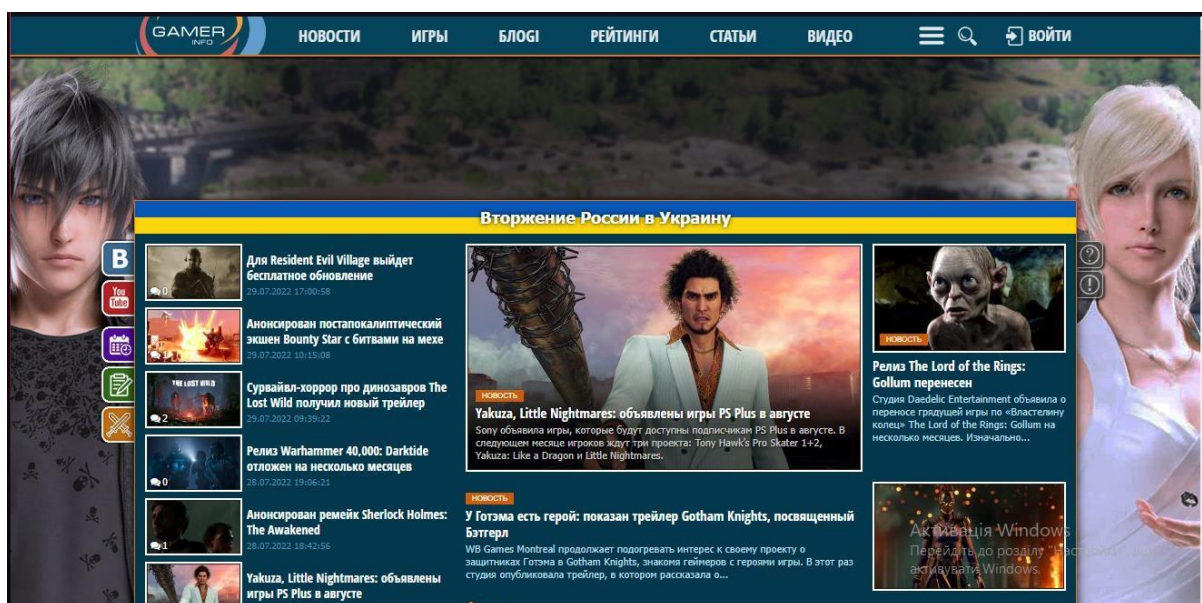


Рис 1.5 Головна сторінка сайту

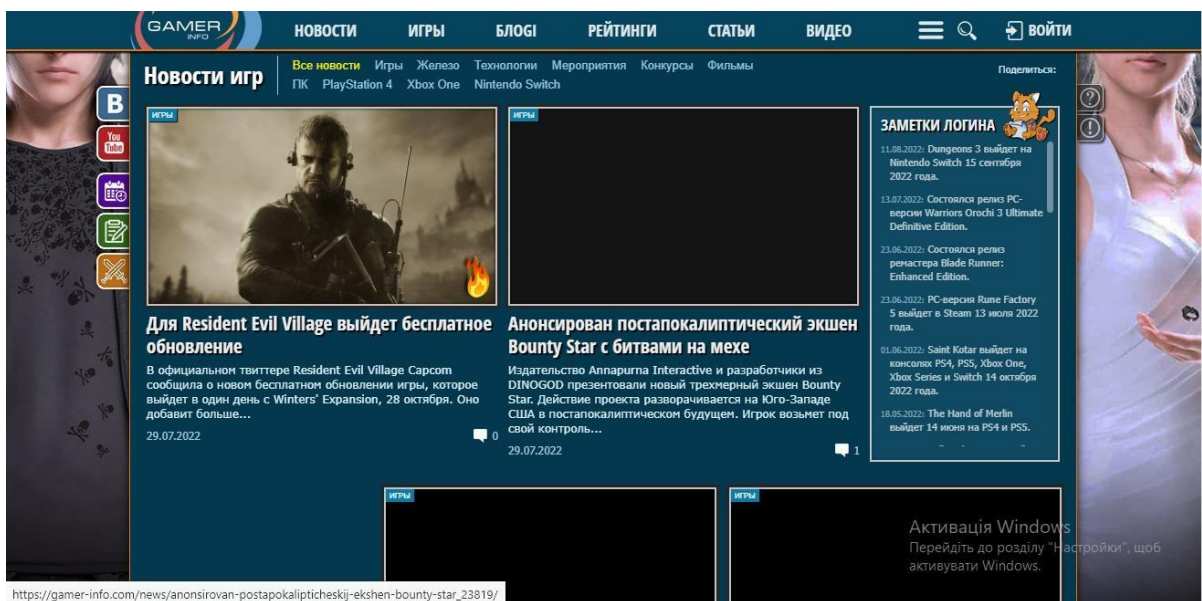


Рис 1.6 Новини веб сайту

Переваги:

- Є відео яке можна переглядати.
- Можна переглядати новини і бути не зареєстрованим

Недоліки:

- Повільне завантаження різних модулів сайту

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

1.2. Постановка задачі

Головною метою кваліфікаційної роботи є розробка web-сервісу для ведення блогів з автоматичною категоризацією контенту, що надаватиме користувачам дані веб-аналітики їх профілів. Розроблювана система повинна ввібрати у себе найкращі практики створення блогів та оминати недоліки уже існуючих систем.

Сервіс дозволить зручно отримати доступ до будь-якого запису блогу та до результатів аналізу блог-профілю з будь-якого пристрою, який підключений до мережі інтернет та має web-браузер.

Web-додаток повинен бути адаптованим до різних типів пристроїв (персональні комп'ютери, планшетні комп'ютери, смартфони).

Функції додатку повинні поступово розширюватися разом з етапами розробки. Проте серед основних можливостей можна виділити наступні:

- Можливість створення профілю та перегляду профілів інших користувачів.
- Можливість підписатися на профіль іншого користувача.
- Додавання, редагування, видалення, перегляд записів.
- Автоматичне групування записів за темою.
- Можливість пошуку записів за ключовими словами та темою.
- Адаптивний дизайн під різні роздільні здатності моніторів.
- Перегляд статистики власного профілю блогу.
- Перегляд результатів аналітики записів профілю блогу.

Основні етапи розробки проекту:

1. Налаштування середовища.
2. Проектування та створення бази даних.
3. Проектування та розробка ядра системи, в якому зосереджена бізнес логіка.
4. Збір даних для проведення навчання моделі класифікації.
5. Навчання моделі класифікації для розділу записів блогу за темами.
6. Розробка модулю аналітики.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

7. Розробка зовнішньої частини додатку.
8. Створення процесу забезпечення безпеки доступу за допомогою автентифікації та авторизації.
9. Включення можливостей JavaScript в проект (візуалізація ефектів, динамічне підвантаження контенту та інше.).
10. Тестування проекту, проведення code review, виправлення помилок після тестування, оптимізація коду.
11. Демонстрація проекту та його здача.

1.3. Вибір архітектури web-орієнтованої системи

Для розробки даної системи можна використати наступні шаблони:

- Model – View – Controller (модель – представлення – контролер)
- Model – View – Presenter (модель – представлення – пред'явник)
- Service – oriented architecture (сервісно – орієнтована архітектура)
- Model – View – View Model (модель – представлення – модель представлення)
- Microservices (мікросервісна)

Оптимальним для реалізації web-блогу буде використання архітектурного шаблону MVC (рис.1.7).

MVC – це патерн проектування веб-додатків, який включає кілька дрібніших шаблонів. При використанні MVC на три окремі компоненти розділені модель даних програми, інтерфейс користувача і логіка взаємодії користувача з системою, завдяки чому модифікація одного з цих компонентів надає мінімальний вплив на інші або не має його зовсім.

Основна мета застосування MVC полягає у розподілі даних та бізнес-логіки від візуалізації. За рахунок такого поділу підвищується можливість повторного використання програмного коду: наприклад, додати уявлення даних будь-якого існуючого маршруту не тільки у вигляді HTML, а й у форматах JSON, XML, PDF, XLSX стає дуже просто і не потребує змін шару бізнес-логіки вихідного маршруту. Також спрощується і супровід програмного коду: внесення

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

змін до зовнішнього вигляду, наприклад, не відбивається на бізнес-логіці, а зміни бізнес-логіки не торкаються візуалізації [15].

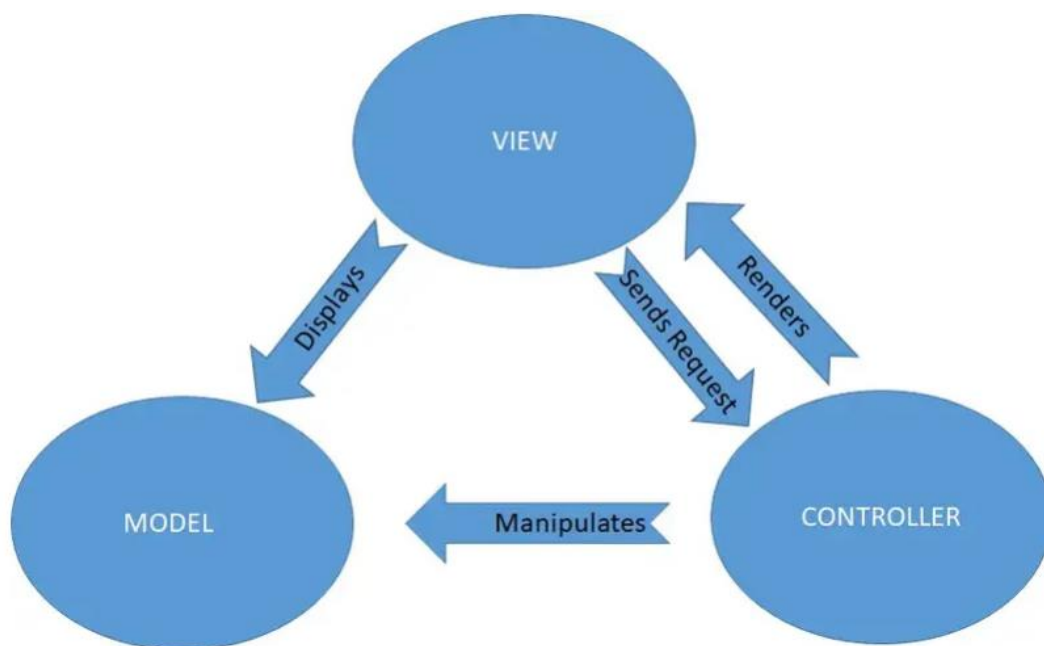


Рис. 1.7. MVC архітектура

Концепція MVC поділяє дані, представлення та обробку дій користувача на компоненти:

Модель / Model – це об’єкти, які містять оператори бізнес-логіки для читання або запису даних у базу даних. Ця модель може бути розроблена в різних формах залежно від вимог, наприклад, функціональні модулі, ВАРІ, класи тощо. Модель, створена як клас (клас моделі), є не що інше, як клас, який формує в ньому певні дані, які захочеться показати на екрані Web Dynpro. При цьому модель не містить у собі інформації про способи візуалізації даних або формати їх подання, а також не взаємодіє з користувачем безпосередньо.

Подання / View – відповідає за відображення інформації (візуалізацію). Одні й самі дані можуть представлятися різними способами й у різних форматах. Наприклад, колекцію об’єктів за допомогою різних уявлень можна представити на рівні інтерфейсу користувача як в табличному вигляді, так і списком; на рівні API можна експортувати дані як у JSON, так і в XML або XSLX.

Контролер працює як інтерфейс між моделлю (в якій зберігаються дані) і представленням (яке відображає дані). Це означає, що це використовується для

керування зв'язком між моделлю та представленням, де він приймає вхідні дані від користувачів і отримує потрібні дані від моделі та відображає дані у браузері.

Таким чином, можна зробити висновок, що даний архітектурний шаблон забезпечує всі технологічні можливості для побудови системи, що дозволяє реалізувати поставлені завдання.

1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Для реалізації web-орієнтованої блог платформи обрано мови програмування PHP та JavaScript. Також для розробки інтерфейсу додатку використовуватимуться HTML та CSS.

PHP – це поширена мова програмування загального призначення з відкритим вихідним кодом. PHP спеціально сконструйований для веб-розробок, і його код може впроваджуватися безпосередньо в HTML. Завданням мови PHP є забезпечення ефективного зв'язку веб-ресурсу із сервером та базами даних. При цьому ця мова надзвичайно проста в освоєнні та застосуванні. PHP чудово поєднується з HTML-кодом.

Головним чинником мови PHP є практичність. PHP надає програмісту засоби для швидкого та ефективного вирішення поставлених завдань. Вибір PHP обумовлений наступними важливими характеристиками [16]:

- традиційністю;
- простотою;
- ефективністю;
- безпекою;
- гнучкістю.

Для написання частини проекту, що міститиме інтелектуальний аналіз вирішено використовувати мову програмування Python.

Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднування

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах [17].

Розглянемо переваги Python для машинного навчання:

- Великий вибір бібліотек та фреймворків. Одна з основних причин, чому Python використовується для машинного навчання, полягає в тому, що у нього є безліч фреймворків, які спрощують процес написання коду і скорочують час на розробку.
- Зрозумілість. Python — найбільш високорівнева і зрозуміла мова, з якою зручно працювати. Завдяки його лаконічності та зручності читання він добре підходить для навчання розробці ПЗ. Простий синтаксис мови Python допомагає розробнику тестувати складні алгоритми з мінімальною тратою часу на їхню реалізацію.
- Гнучкість. Розробник має вибір між об'єктно-орієнтованим підходом і скриптами, Python допомагає поєднувати різні типи даних.
- Широка підтримка та якісна документація. Існує безліч корисних ресурсів про Python, на яких програміст може отримати допомогу та консультацію, перебуваючи на будь-якому етапі розробки.

Одним із основних інструментів при розробці web-додатку та моделей класифікації є середовище розробки. На сьогоднішній день на ринку інформаційних технологій існує велика кількість різних засобів розробки додатків, у кожного з яких є свої переваги. Тому для вибору найкращого варіанту для розробки web-блогу та модулю інтелектуального аналізу варто виділити наступні вимоги до інструментального засобу: робота з проектами, пошук по файлах проекту, підсвічування змін файлу і DiffTool, підтримка Git, автоматичне доповнення коду, автоформатування та підсвічування синтаксису, приємний інтерфейс, підтримка плагінів, швидка робота, доступність.

Серед тих середовищ розробки, що задовольняють дані потреби у Python та PHP розробників найбільшою популярністю користуються два інструменти:

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

WebStorm та Visual Studio Code. Для роботи над розроблюваною системою було обрано редактор VS Code (рис.1.9).

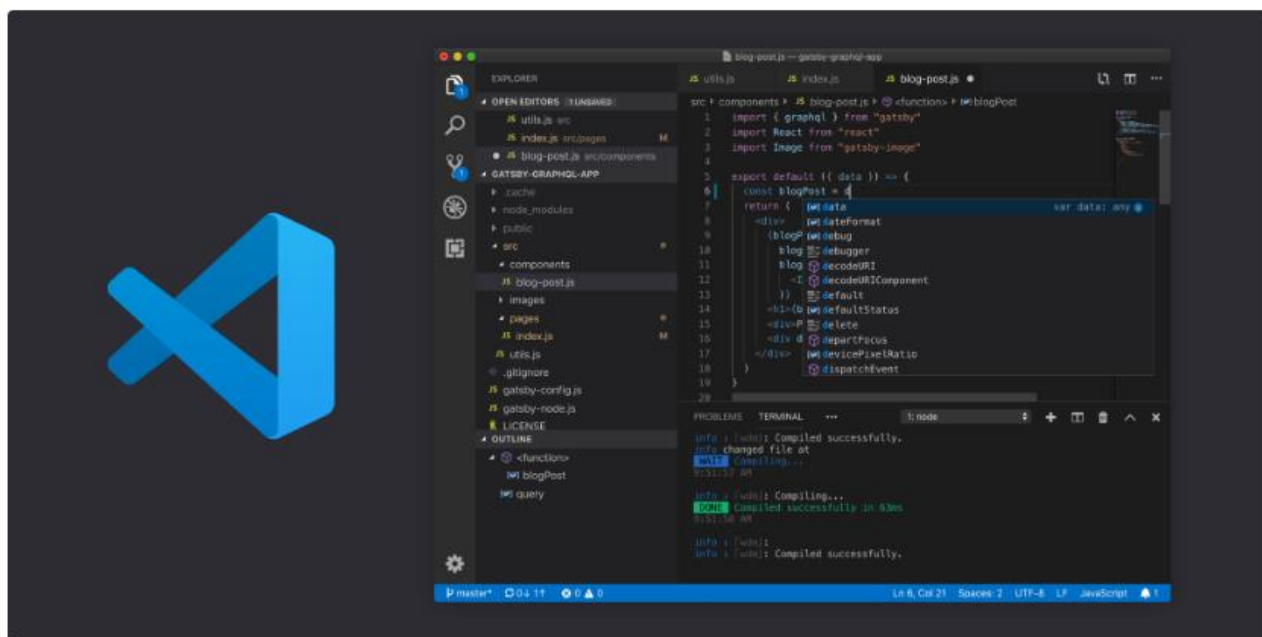


Рис. 1.9. Робоче вікно редактора VS Code

Вибір даного засобу обумовлений наступними перевагами:

- VS Code має високу швидкість запуску (відкриття проекту в даному редакторі можна назвати приємним, оскільки це відбувається без довгих індексацій та завантаження модулів);
- Редактор має приємний інтерфейс, який водночас стислий і зрозумілий (рис. 1.9);
- У VS Code добре реалізована робота з гітом (робота зводиться до написання тексту комітів і натискання кнопки синхронізації);
- Можливість розширення (VS плагіни додають функції, що забезпечують встановлення тем, підтримку нових мов і створення різноманітних інструментів);
- VS Code розповсюджується безкоштовно і доступний у версіях для платформ Windows, Linux і OS X.

В якості підходу реалізації системи управління базами даних обрано MySQL.

Вибір MySQL обумовлений наступними сильними сторонами механізму:

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

- Широкий функціонал. Система MySQL має практично весь необхідний інструментарій, який може знадобитися в реалізації практично будь-якого проекту.
- Безпека. Система створена таким чином, що багато вбудованих функцій безпеки в ній працюють за умовчанням.
- Масштабованість. Будучи дуже універсальною СУБД, MySQL однаково легко може бути використана для роботи і з малими, і з великими обсягами даних.
- Швидкість. Висока продуктивність системи забезпечується за рахунок спрощення деяких стандартів, що використовуються в ній.
- Простота у використанні. MySQL досить легко встановлюється, а наявність безлічі плагінів та допоміжних програм спрощує роботу з базами даних.

Важливим інструментальним засобом при розробці web-додатку є також інструмент для розробки бази даних. В таких цілях найкраще використовувати офіційну систему для розробки та адміністрування – MySQL Workbench (рис. 1.10).

MySQL Workbench – це уніфікований візуальний інструмент для архітекторів баз даних та розробників БД. MySQL Workbench надає можливість моделювання даних, розробку SQL та комплексні інструменти адміністрування для конфігурації сервера, адміністрування користувачів, резервного копіювання та багато іншого [18]. MySQL Workbench доступний на Windows, Linux та Mac OS X.

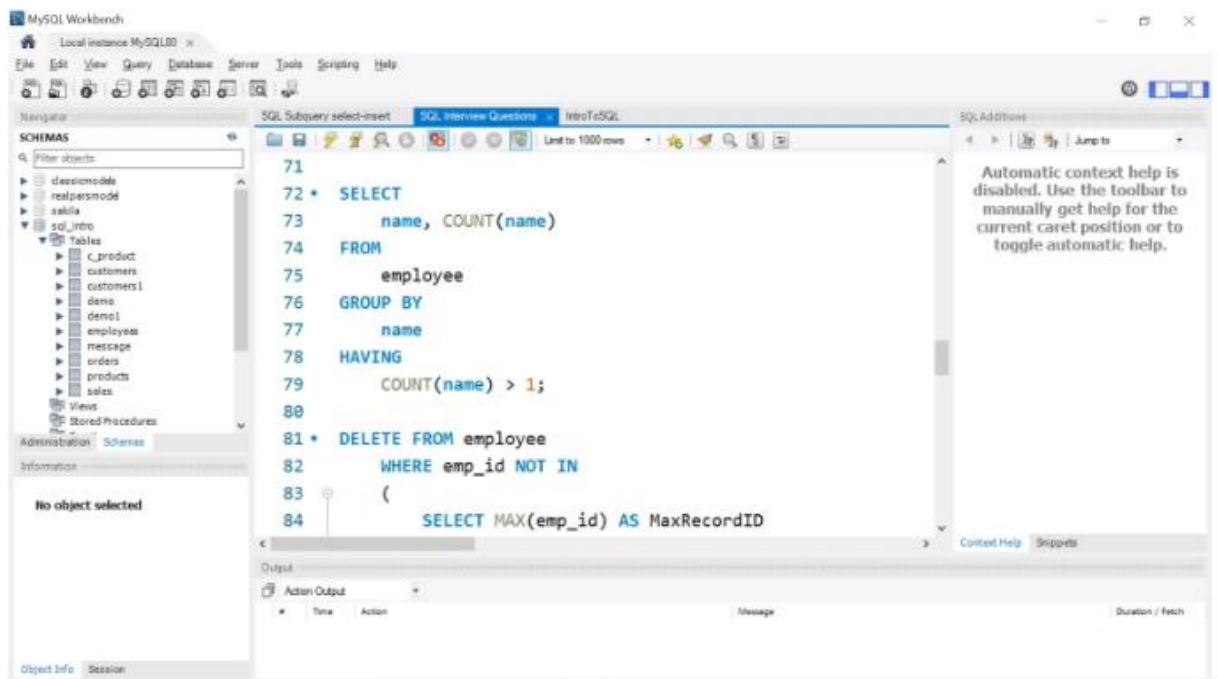


Рис. 1.10. – MySQL Workbench

MySQL Workbench забезпечує усі необхідні для роботи з MySQL можливості:

- Перегляд, додавання та видалення бази даних і таблиці.
- Редагування даних у таблиці в візуальному режимі.
- Дозволяє наочно представити модель бази даних в графічному вигляді.
- Наочний і функціональний механізм установки зв'язків між таблицями, в тому числі «багато до багатьох» із створенням таблиці зв'язків.
- Reverse Engineering — відновлення структури таблиць з вже існуючої на сервері БД (зв'язки відновлюються в InnoDB, при використанні MyISAM зв'язки необхідно встановлювати вручну).

Зручний редактор SQL запитів, що дозволяє відразу ж відправляти їх серверові і отримати відповідь у вигляді таблиці.

Висновки до першого розділу

У даному розділі було поставлено задачу та проаналізовано методи та способи вирішення. Були досліджені аналоги та визначено їх переваги та недоліки.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

Огляд аналогів показав, що головними функціями майбутньої блог-системи має бути: можливість створення та редагування профіля, можливість створення, читання та редагування постів.

Було визначено оптимальну структуру та MVC архітектуру додатку, що представлені у першому підрозділі, у тому числі на прикладі діаграми компонентів системи.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ

2.1. Визначення варіантів використання та структури системи

Система функціонування web-блог створюється з метою реалізації додатку загального користування, що надаватиме весь необхідний функціонал для ведення блогу та інформацію, яка може бути необхідною для просування блогу, в зручній та зрозумілій формі. В залежності від того який користувач користується веб-блогом, існують різні алгоритми роботи веб сторінки. Якщо користувач неавторизований, йому доступні наступні функції:

1. Перегляд головної сторінки сайту.
2. Перегляд доданих блогів від інших користувачів.
3. Перегляд останнього блогу
4. Реєстрація та авторизація.
5. Детальна інформація про блог

Загальну схему роботи сайту для неавторизованого користувача наведено нарисунку 2.1.

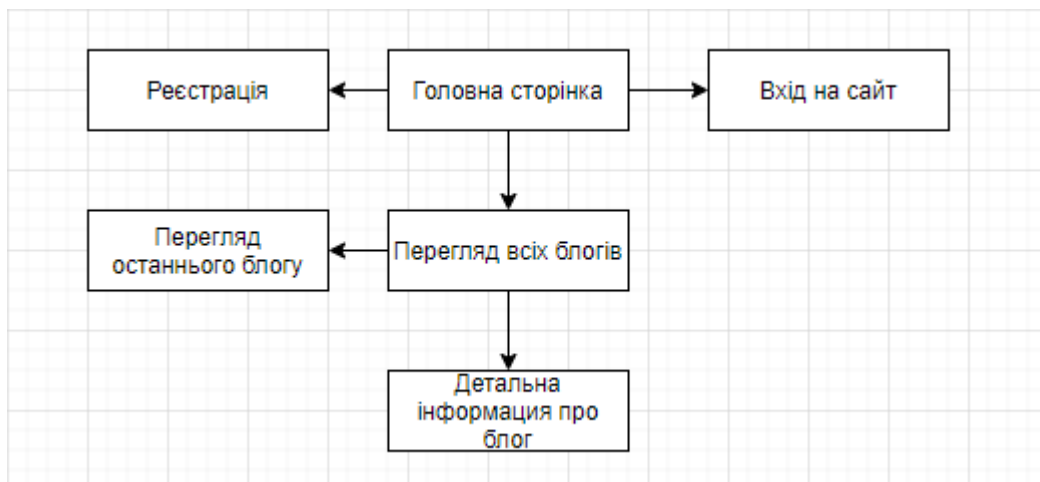


Рис 2.1 Схема роботи сайту для неавторизованого користувача

Розглянемо дії, до яких користувач має доступ при реєстрації та авторизації:

1. Перегляд головної сторінки сайту.
2. Перегляд доданих блогів від інших користувачів.
3. Перегляд останнього блогу
4. Можливість вийти з акаунту

5. Детальна інформація про блог
6. Можливість додати блог
7. Можливість видалити доданий блог
8. Можливість редагувати доданий блог

Загальну схему роботи сайту для авторизованого користувача наведено на рисунку 2.2.



Рис 2.2 Схема роботи сайту для авторизованого користувача

Також на сайті присутній аккаунт адміністратора, який має повний контроль над контентом сайту, та може робити певні речі, яких не може робити звичайний користувач. Розглянемо дії доступні адміністратору:

1. Додавання, видалення та редагування будь якого товару.
2. Перегляд повної Інформації про користувачів

Загальну схему роботи сайту для адміністратора наведено на рисунку 2.3.

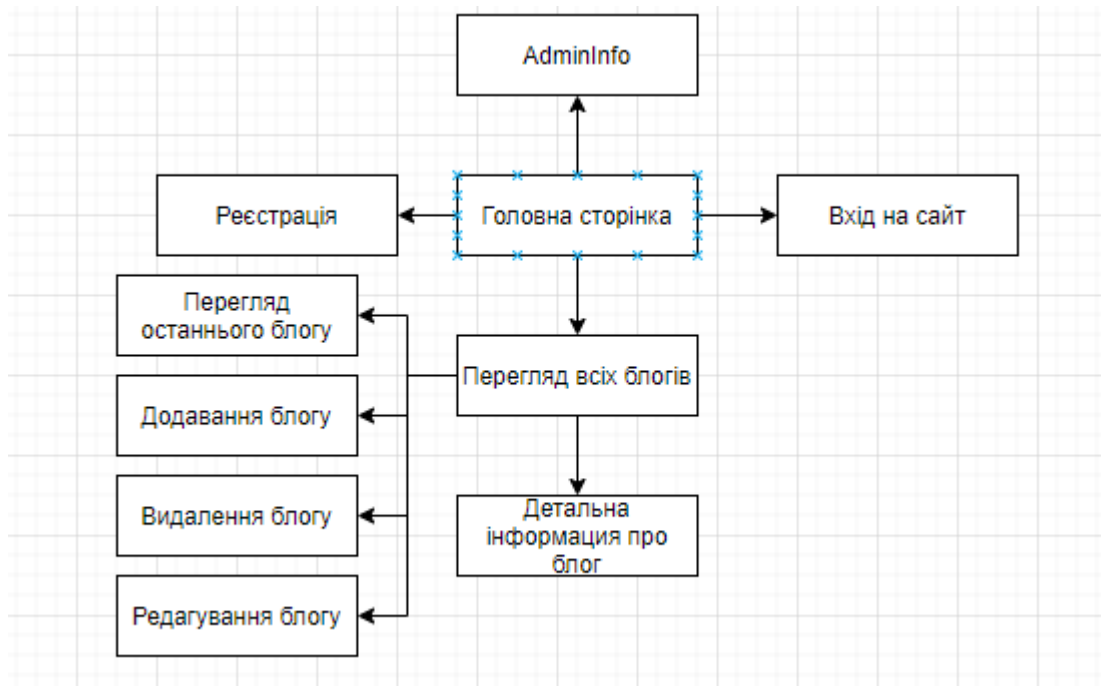


Рис 2.3 Схема роботи сайту коли користувач адмін

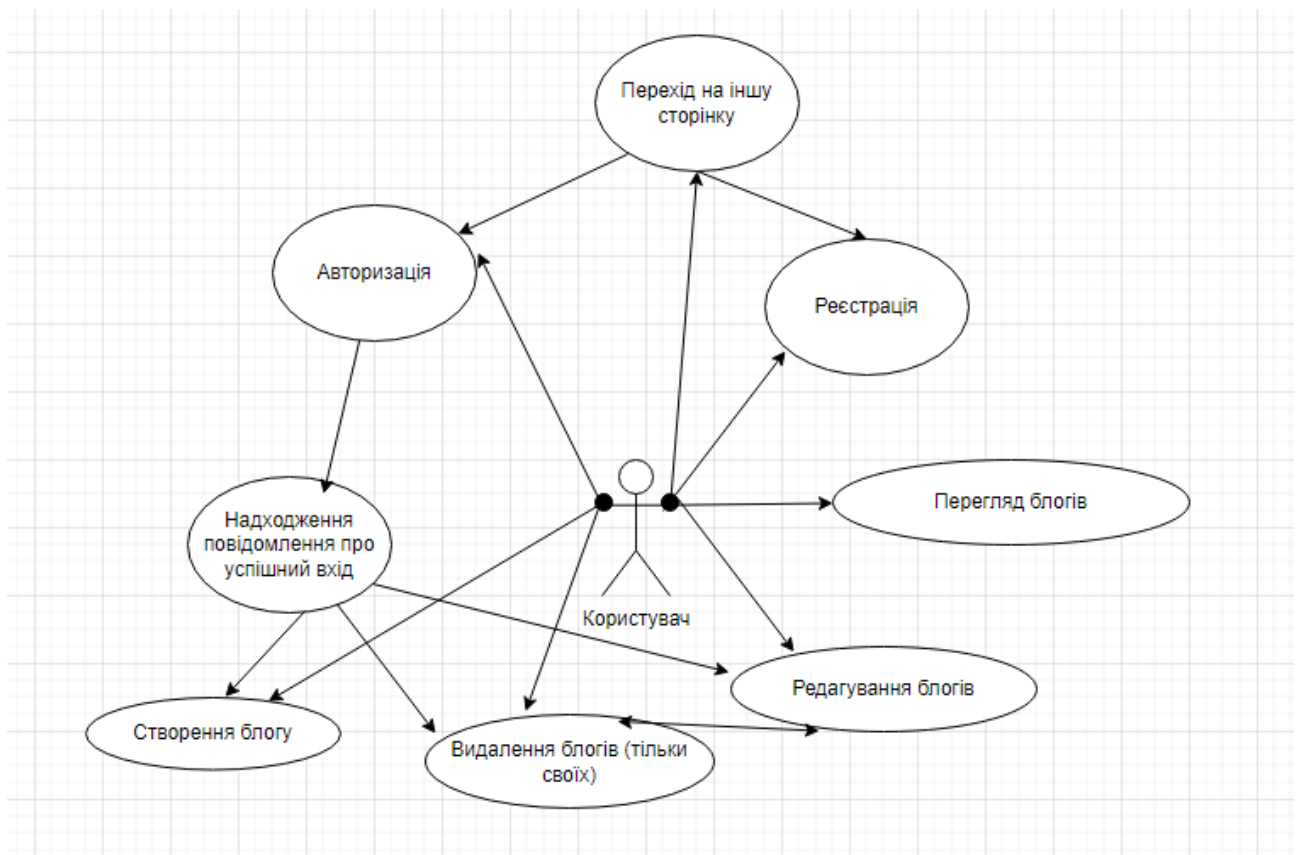


Рис. 2.4 Варіанти використання

На рисунку 2.4 відображено можливості користувача при користуванні сайту.

2.2. Розробка бази даних та архітектури системи

Невід’ємною частиною функціонування системи є робота та обробка інформації, а саме створення, оновлення, видалення та вибірка даних. Для реалізації цих задач було спроектовано базу даних (рис. 2.4). База містить у собі наступні 6 таблиць: User, Posts, Comments, Category, Follow, Likes.

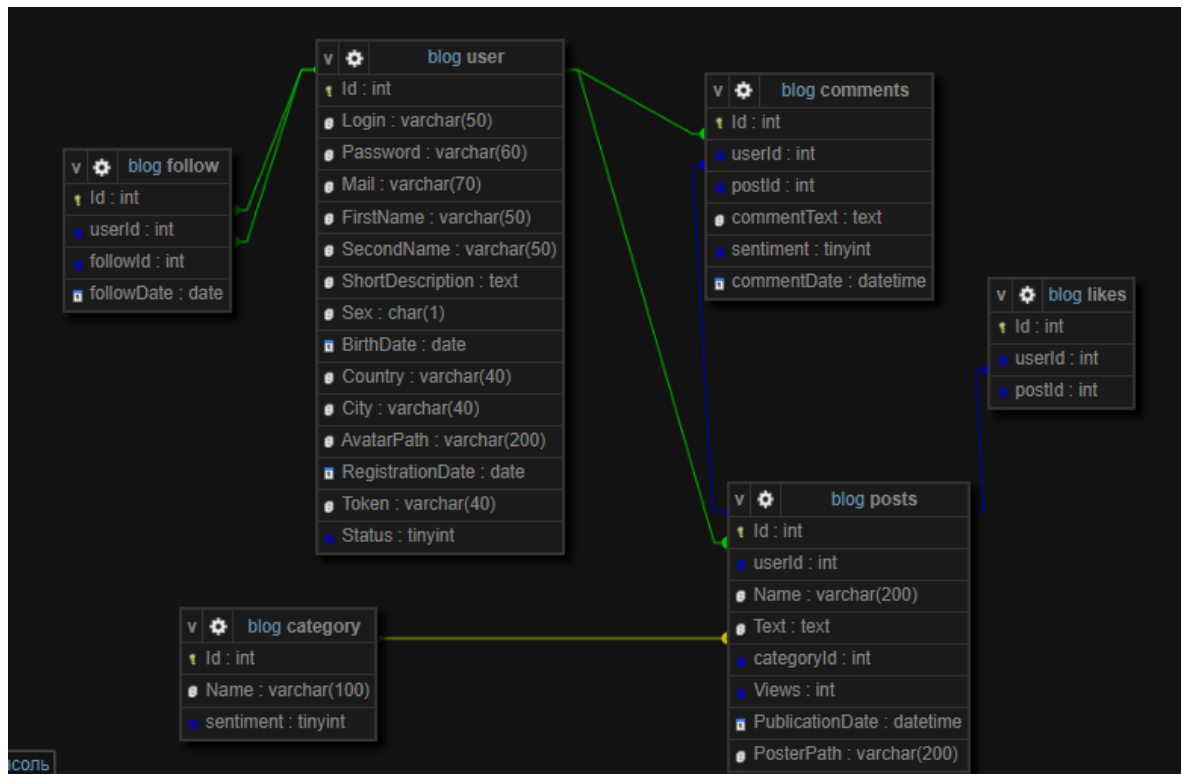


Рис. 2.4. Структура баз даних системи

Потрібно розробити CMS- систему, для цього було обрано патерн MVC. Цей патерн дозволяє розділити всю систему сайту на три категорії: моделі, вигляди, контролери. Тому проект було розділено на три каталоги (рис. 2.5).

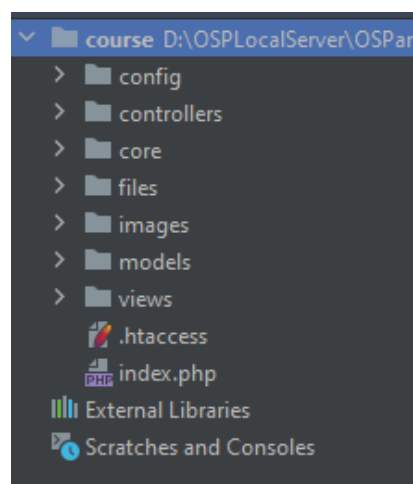


Рис 2.5 Структура проекту

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

При розробці системи було використано об'єктно-орієнтований підхід, за роботу сторінки відповідають класи. Проект має каталог Core - це каталог в якому знаходяться важливі для функціонування класи(рис. 2.6).

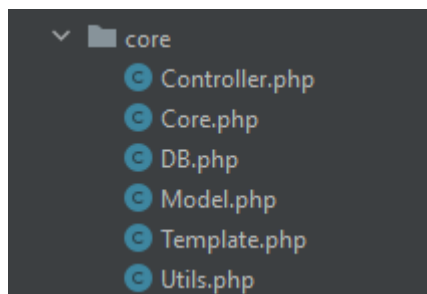


Рис 2.6 Каталог Core

Основне функціонування на себе бере каталог Core, як ми бачимо він містить багато класів, нижче перераховані їх назви та функції.

1. Controller - батьківський клас для усіх контролерів.
2. Core - містить методи для виконання основних задач системи.
3. DB - цей клас відповідає за роботу з базою даних, у ньому описані основні методи для роботи з базою даних.
4. Model - батьківський клас для всіх моделей.
5. Template - відповідає за відображення частин сторінки.
6. Utils - містить в собі додаткові методи.

Розглянемо каталог Models в якому знаходиться 2 класи(рис. 2.7).

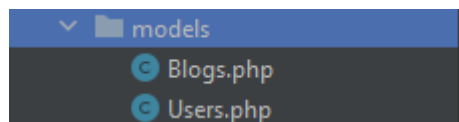


Рис 2.7 Каталог Моделс

Усі класи працюють з базою даних роблячи, редагування, додавання, видалення. Blogs відповідає за блоги, Users за користувачів.

Розглянемо каталог Controllers у якому знаходяться усі контролери які є необхідними для функціонування сайту(рис. 2.8).

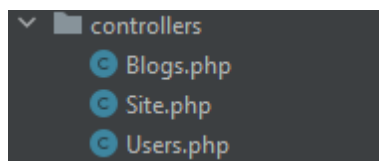


Рис 2.8 Каталог контролерів

У каталозі Controllers містяться 3 контроллери, розглянемо кожен із них. Blogs - контроллер який відповідає за роботу із сумками, їх додавання, видалення, редагування, відображення, перевірку параметрів, та інші методи.

Site - контроллер який відповідає за виведення інформації на потрібну сторінку.

Users - контроллер який відповідає за авторизацію, реєстрацію, перевірку даних користувача, виходом із аккаунту, та забезпечує наявність аккаунту адміністратора.

Розглянемо каталог Views який зберігає всі сторінки HTML для їх подальшого використання контроллерами(рис. 2.9).

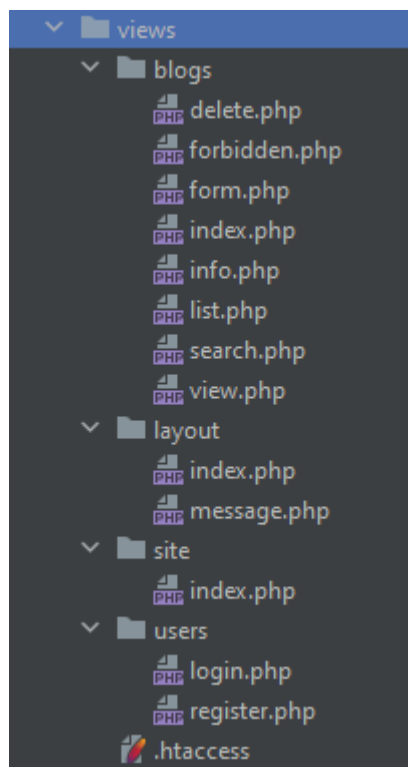


Рис 2.9 Каталог ViewsРозглянемо кожний каталог.

blogs - містить 7 сторінок.

Delete - яка відповідає за видалення товару.

Forbidden - сторінка яка повідомляє якщо доступ до сторінки заборонено.

Form - сторінка для редагування, додавання нового запису.

Index - сторінка для виведення контенту користувачам. View - сторінка для перегляду детальної інформації щодо певного блогу.

Каталог layout містить файл index - який є основним шаблоном сайту для

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

всіх сторінок. Message - відображає певне повідомлення.

Каталог Site містить сторінку з контентом для головної сторінки сайту. Каталог users містить файл Register який відповідає за реєстрацію користувача, та файл login який відповідає за авторизацію користувача.

2.3. Реалізація додатку web-орієнтованої системи

Почнемо із файлу index.php, який підключає файл config, та файл Core.

```
<?php
include ('config/config.php');
include('core/Core.php');

$core = \core\Core::getInstance();
$core->init();
$core->run();
$core->done();|
```

Файл config містить інформацію про підключення до бази даних та зі змінноюBlogsCount яка відповідає за кількість виведених блогів на сайті.

```
<?php
$config =
[
    'Database' =>
    [
        'Server' => 'localhost',
        'Username' => 'root',
        'Password' => '',
        'Database' => 'DBBs'
    ],
    'BlogsCount' => 10
];
```

Спочатку метод Run отримує назву контроллера через суперглобальний масивGET. Далі йде перевірка чи записано у URL адресу назву методу, якщо не записано то змінна fullActionName буде дорівнювати action index. Далі йдуть перевірки, на існування методів та контроллерів. Якщо виникне помилка при перевірці існування методів, або контроллерів, користувач отримає помилку.

```
public function run()
```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    $path = $_GET["path"];
    $pathParth = explode('/', $path);
    $classname =
    ucfirst($pathParth[0]);if
    (empty($classname))
        $fullclassname =
        'controllers\\Site';else
        $fullclassname = 'controllers\\'.$classname;
    $methodname =
    ucfirst($pathParth[1]);if
    (empty($methodname))
        $fullmethodname =
        'actionIndex';else
        $fullmethodname =
        "action".$methodname;
    if(class_exists($fullclassname)) {
        $controller = new $fullclassname();
        if (method_exists($controller, $fullmethodname)) {
            $method = new
            \ReflectionMethod($fullclassname,$fullmethodname);
            $paramsarray = [];
            foreach ($method -> getParameters() as $parameter)
            {
                array_push($paramsarray,isset($_GET[$parameter->name]) ?
                $_GET[$parameter->name] : null);
            }
            $result = $method-
            >invokeArgs($controller,$paramsarray);if
            (is_array($result))
            {
                self::$mainTemplate->setParams($result);
            }
        }
        else
            throw new \Exception('404 not found');
    }
    else
        throw new \Exception('404 not found');
}

```

Розглянемо код роботи одного із контроллерів. Для прикладу візьмемо контроллер блогів, та методу який нас цікавить, нехай буде actionadd.

```

public function actionAdd()
{
    $titleForbidden = 'Доступ
заборонено';if ($this->user ==
null) {
        return $this->render('forbidden',
        null, ['Title' =>
        $titleForbidden]);
    }
    $title = 'Додавання
блory';if ($this-
>isPost()) {
        $result = $this->blogModel-
        >addBlog($_POST);if ($result['error']
        === false) {
            $allowed_types = ['image/png',
            'image/jpeg'];
            if(is_file($_FILES['File']['tmp_name']
            ) &&
            in_array($_FILES['File']['type'], $allowed_types)){

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        switch($_FILES['File']['type']
            '){ case 'image/png':
                $extension =
                'png';break;
            default:
                $extension = 'jpg';
        }
        $name =
        $result['id'].'_'.uniqid().'.'.$extension;
        move_uploaded_file($_FILES['File']['tmp_name'],
'files/blogs/'.$name);

        $this->blogModel->ChangePhoto($result['id'], $name);
    }
    return $this->renderMessage('ok', 'Блог успішно додано',
        null, ['Title' => $title]);
} else {
    $message = implode('<br/>',
        $result['messages']); return $this-
    >render('form', ['model'=> $_POST], [
        'Title' => $title,
        'MessageText' =>
        $message,
        'MessageClass' =>
        'danger']);
    }
}
else{
    return $this->render('form', ['model'=>
        $_POST], ['Title' => $title]);
}
}

```

В даному методі відбувається перевірка чи користувач є зареєстрованим якщо так тоді користувач коли водить дані в формі щоб створити блог він натискає створити і метод перевіряє як саме вони передаються якщо методом пост тоді метод починає обробляти їх далі та додавати їх в базу даних. Потім починає обробку фото. Потім за допомогою Методу addBlog додається інша інформація та перевіряється на наявність помилок та їх виведення при наявності.

Розглянемо наступний метод addBlog.

```

public function addBlog($row)
{
    $userModel = new \models\Users();
    $user = $userModel-
    >GetCurrentUser(); if ($user ==
    null)
    {
        $result = [
            'error'
            => true,
            'messages' => ['Користувач не аутентифікований']
        ];
        return $result;
    }
    $validateResult = $this-
    >Validate($row); if

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        (is_array($validateResult)){
            $result = [
                'error' => true,
                'messages' => $validateResult
            ];
            return $result;
        }
        $fields = ['title','text','short_text'];
        $RowFiltered = Utils::ArrayFilter($row,$fields);
        $RowFiltered['datetime'] = date('Y-m-d H:i:s');
        $RowFiltered['user_id'] = $user['id'];
        $RowFiltered['photo'] = 'photo';
        $id = \core\Core::getInstance()->getDB()-
            >insert('Blogs',$RowFiltered);return [
'error' => false,'id' => $id

```

Спочатку ми перевіряємо чи увійшов користувач на сайт, та тільки потім даємо можливість додавати запис. Після успішної перевірки даних методом Validate ми можемо переходити до додавання записів. Всі записи крім дати, фото, ім'я користувача та його логін з паролем, додаються за тими даними що ввів користувач.

Далі розглянемо як працює контроллер Users. Та його Метод actionLogin .

```

public function actionLogin()
{
    $title = 'Вхід на сайт'; if(isset($_SESSION['account']))
        return $this->renderMessage('ok', 'Ви уже увійшли на сайт', null,['Title' => $title]);
    if ($this->isPost())
    {
        $user =
        $this->usersModel->AuthUser($_POST['user'],$_POST['password']); if (!empty($user))
        {
            $_SESSION['account'] = $user;
            return $this->renderMessage('ok', 'Ви успішно увійшли на сайт',null, ['Title' => $title]);
        }
        else
        {
            return $this->render('login',null,[
                'Title'=>$title,
                'MessageText'=>'Неправильний логін або пароль',
                'MessageClass'=>'danger'
            ]);
        }
    }else {
        $params = ['Title'=>$title];
        return $this->render('login',null,$params);
    }
}

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

}

Даний метод використовує інший метод під назвою AuthUser для того щоб перевірити правильність введених даних або ж повідомити що авторизація пройшла успішно.

Висновки до другого розділу

У другому розділі було розглянуто застосовані при розробці архітектури патерни проектування додатку. Також розроблено структуру бази даних для збереження усієї необхідної інформації для повноцінної роботи системи.

В ході виконання розділу було знайдено та підготовлено дані для машинного навчання та створено модель класифікації публікацій блогу.

В останньому пункті наведено детальний розгляд основних аспектів реалізації web-блогу, зокрема описано принципи роботи маршрутизації, деяких компонентів системи та файлову структуру додатку.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ

3.1. Порядок встановлення та налаштування системи.

Розроблена платформа – це web-сайт. Сторінки сайтів - це набір текстових файлів, розмічених мовою HTML. Ці файли розуміються і обробляються браузером та виводяться на засіб відображення користувача (монітор, екран, принтер або синтезатор мови). Тому, перегляд сторінок в Інтернеті не потребує надпотужної комп'ютерної системи чи оптоволоконного каналу зв'язку .

– Проте, оскільки частина контенту генерується за допомогою мови програмування JavaScript, а сценарії JavaScript надсилаються сервером для виконання на боці клієнта (тобто саме на комп'ютері користувача). Та через те що дописи блогу можуть включають у себе мультимедійні матеріали, все-таки можна виділити деякі вимоги до апаратної частини та інтернет-з'єднання для комфортного перегляду web-блогу в браузері:

– Мінімальні вимоги до комп'ютера: процесор 1,5 ГГц, оперативна пам'ять 512 Мб, вільне місце на диску 350 Мб, операційна система Windows 7 або вище / Mac OS X 10.6 або вище / Linux (Ubuntu 10.04 + / Debian 6 + / OpenSuSE 11.3 + / Fedora Linux 14).

– Мінімальні вимоги до смартфона: Android – операційна система Android 5.0 (Lollipop) або вище, iOS – версія iOS 12.2 або вище.

– Дисплей (монітор, екран) повинен мати розподільну здатність щонайменше 1024 точки (пікселі) по горизонталі.

– Підключення до мережі Інтернет що забезпечує прийом даних близько 1 Мегабіт на секунду (при роботі з текстами такої швидкості цілком достатньо).

Таким чином розроблена web-платформа не вимагає спеціалізованого апаратного забезпечення, додаткових налаштувань та засобів для розгортання окрім стандартних для усіх сучасних девайсів та браузерів.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

3.2. Опис роботи з програмним додатком

При вході на сайт буде завантажено головну сторінку з якої користувач зможе перемикатися на інші(рис 3.1)

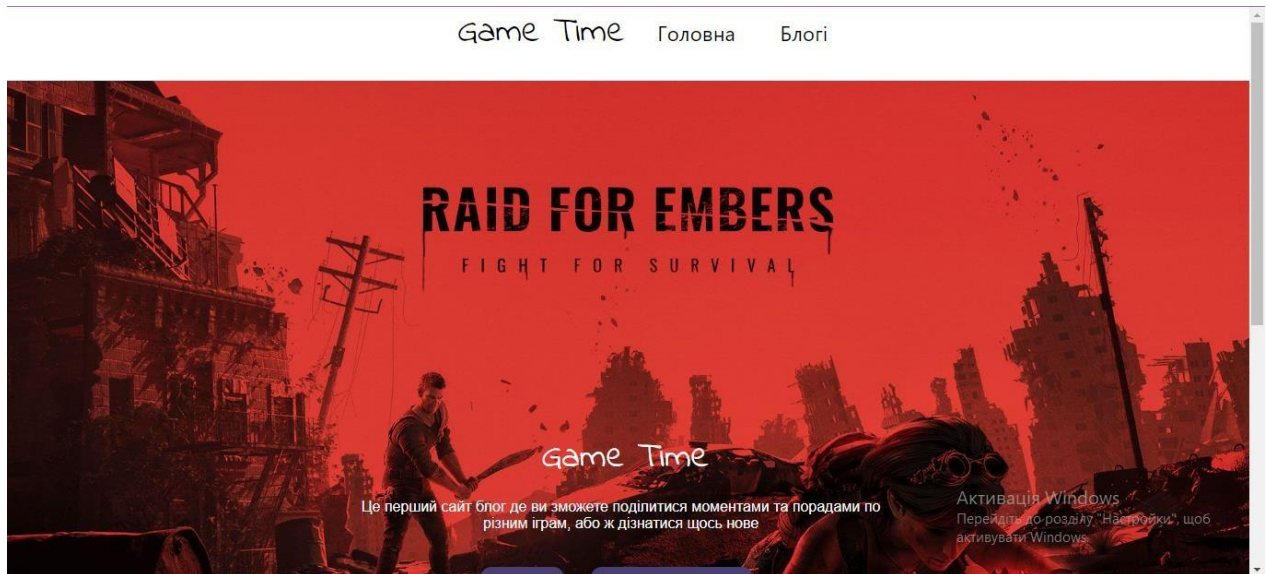


Рис 3.1 Головна сторінка сайту

Коли користувач захоче зареєструватися він натисне на кнопку та перейде на форму реєстрації(рис 3.2)

Рис 3.2 Форма реєстрації

Якщо користувач зареєстрований або тільки що це зробив йому слід увійти на сайт(рис3.3)

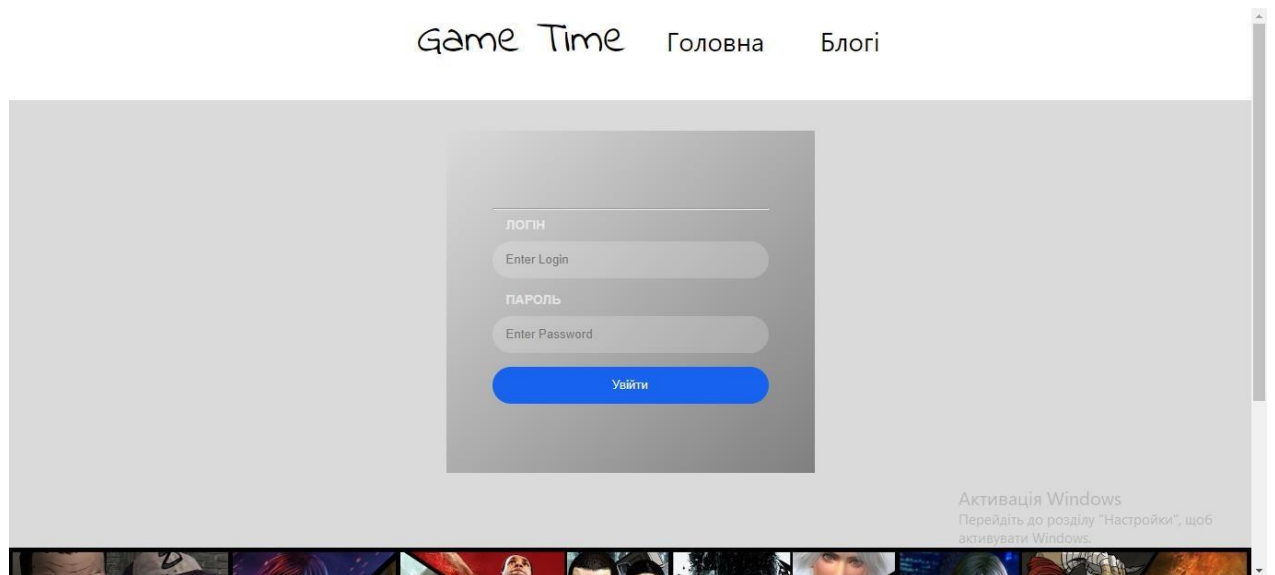


Рис 3.3 Форма для входу в акаунт

При успішному вході виведеться відповідне повідомлення (рис 3.4)

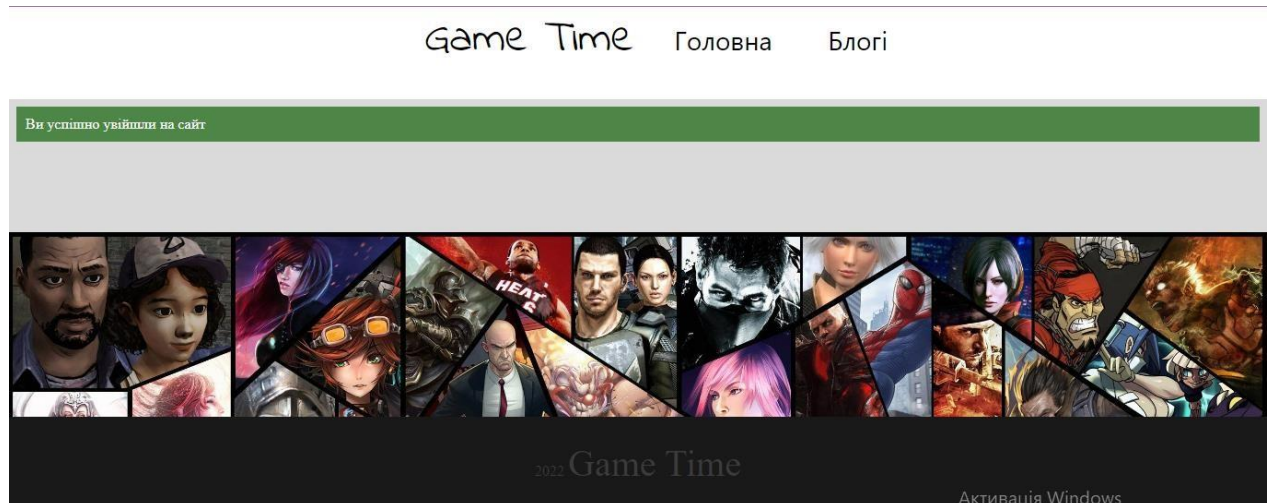


Рис 3.4 Повідомлення про вхід в акаунт

Після входу в акаунт ми можемо переглядати блоги додавати їх та редагувати видаляти тільки ті які ми створювали (рис. 3.5 - 3.7)

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

Game Time Головна Блогі

ЗАГОЛОВОК БЛОГУ

Введіть заголовок блогу

ВВЕДІТЬ КОНТЕНТ БЛОГУ

ВВЕДІТЬ КОРОТКИЙ ОПИС БЛОГУ

ФОТОГРАФІЯ БЛОГУ

Виберіть файл

Файл не вибран

Зберегти

Активация Windows

Перейдіть до розділу "Настройки", щоб активувати Windows.

Рис 3.5 Форма створення блогу

Game Time Головна Блогі

ЗАГОЛОВОК БЛОГУ

bloha

ВВЕДІТЬ КОНТЕНТ БЛОГУ

d1ch0154

ВВЕДІТЬ КОРОТКИЙ ОПИС БЛОГУ

xD bruh777

ФОТОГРАФІЯ БЛОГУ

Виберіть файл

Файл не вибран

Зберегти

Активация Windows

Перейдіть до розділу "Настройки", щоб активувати Windows.

Рис 3.6 Редагування створеного блогу

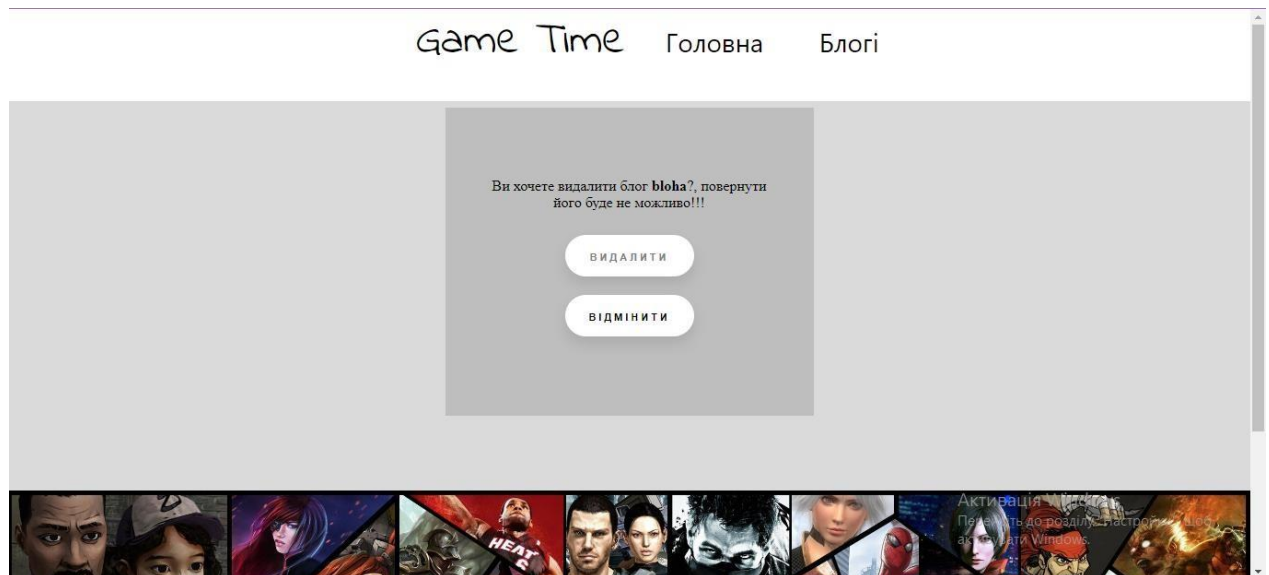


Рис 3.7 Видалення створеного блогу

Якщо ж користувач не хоче створювати блоги, а лише переглядати він також може це зробити потрібно обрати який сподобався та натиснути кнопку Reedall (рис3.8)



Рис 3.8 Повна інформація про блог

3.3. Тестування роботи системи

Тестування, як заключний етап розробки web-сайту, грає життєво важливу роль в процесі створення високоякісного програмного забезпечення. Для перевірки основних показників розробленого блогу проведено тестування сайту за допомогою інструменту Google Lighthouse.

Google Lighthouse — це інструмент аудиту з відкритим вихідним кодом, який допомагає розробникам підвищити продуктивність і доступність своїх веб-

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

проектів. Його можна запустити на будь-якій веб-сторінці, загальнодоступній або яка вимагає аутентифікації. Google Lighthouse здійснює перевірку продуктивності, доступності та відповідності стандартам і найкращим практикам.

Для перевірки обрано головну сторінку web-системи, оскільки вона відвідується користувачами найчастіше та містить найбільше контенту.

Результат тестування головної сторінки показав досить хороші результати (рис. 3.9): продуктивність – 97/100, доступність – 91/100, відповідність найкращим практикам – 87/100.

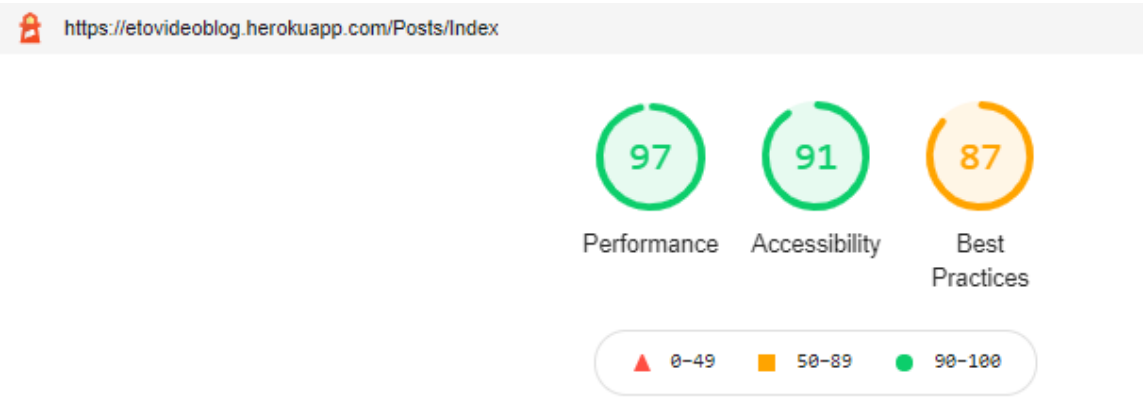


Рис 3.9. Результат перевірки в Google Lighthouse

Перейдемо до тестування сайту та подивимося на функції детальніше. Спочатку коли ми відкриваємо інформацію про блог ми можемо більш детальніше продивитися картинку(рис3.10)

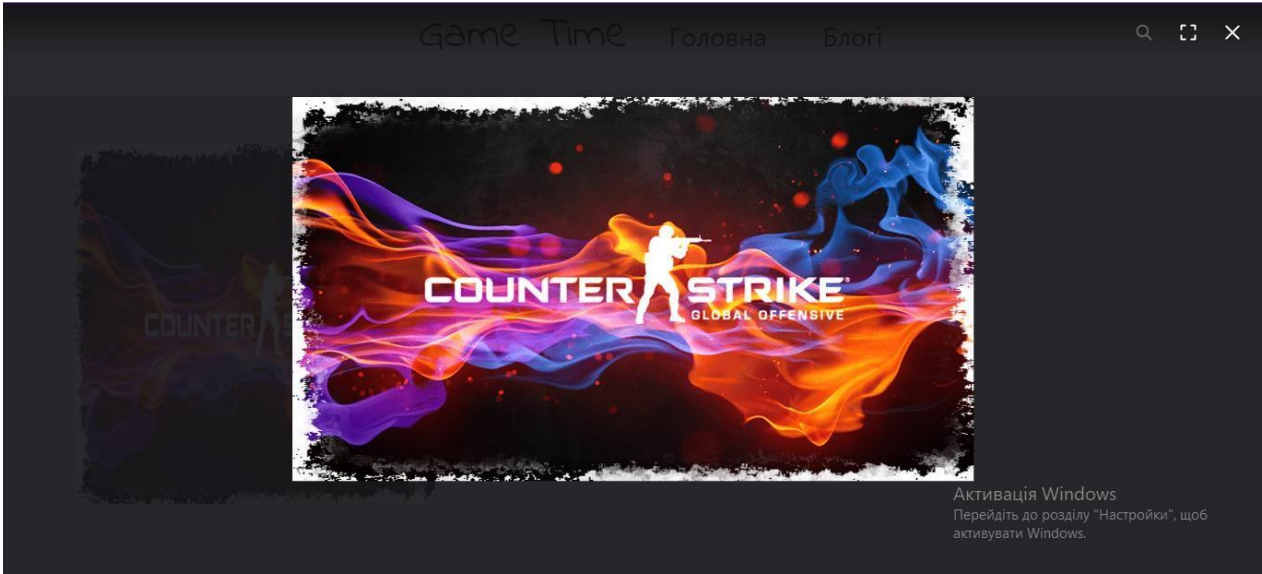


Рис 3.10 Збільшена картинка

Також коли ми не створювали блог ми не можемо його редагувати та видаляти

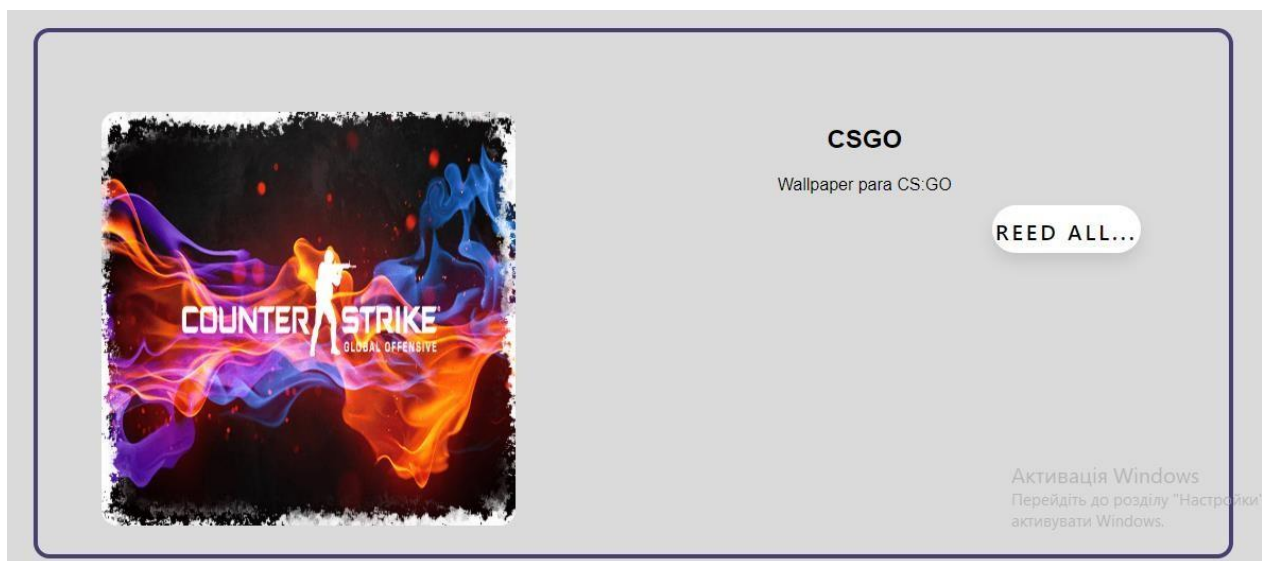


Рис 3.11 Перегляд блогу іншого користувача

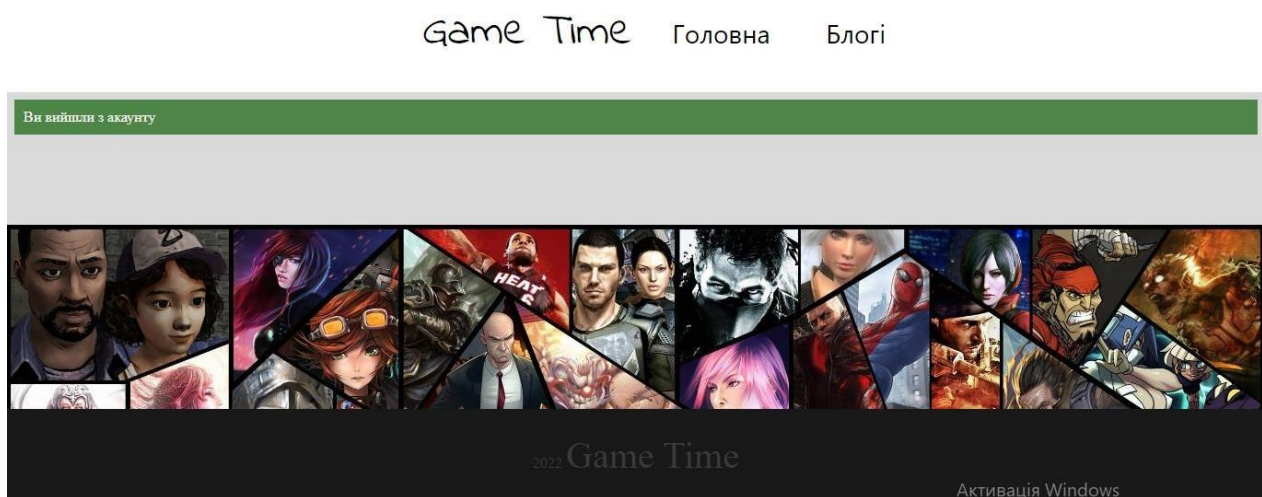


Рис 3.12 Вихід з акаунту

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

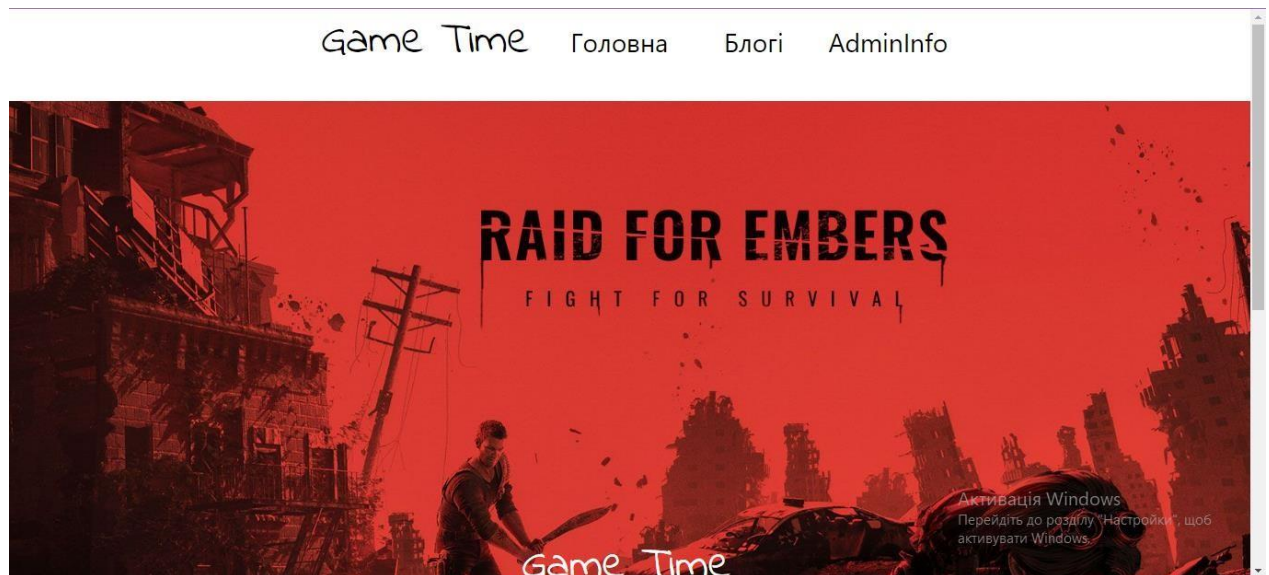


Рис 3.13 Вхід в акаунт адміна нова вкладка admininfo

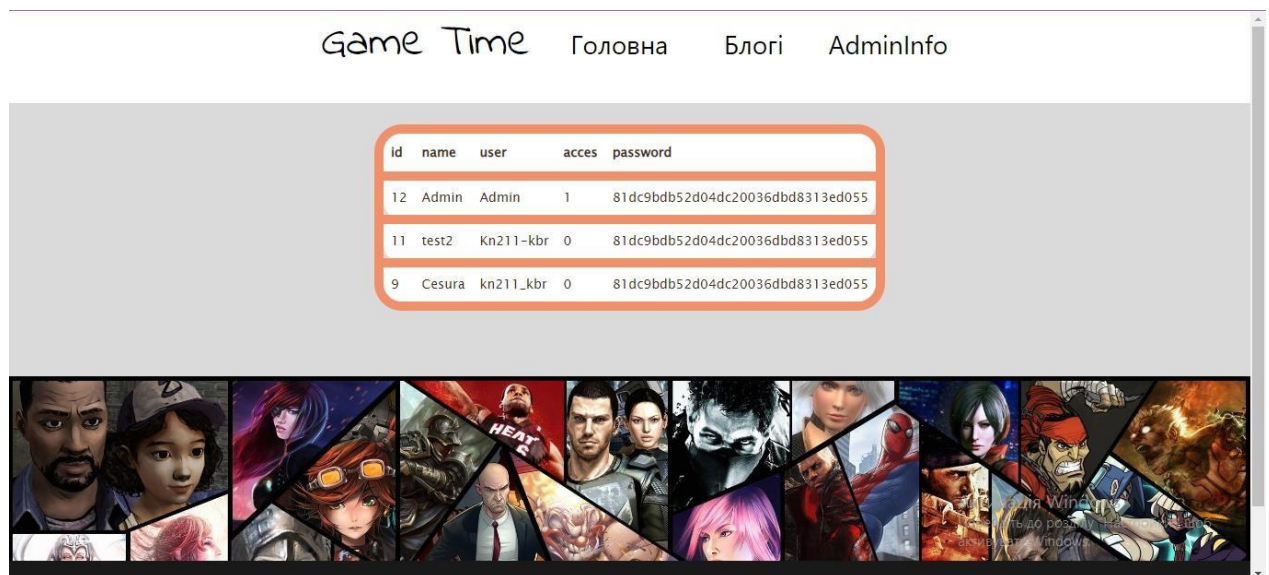


Рис 3.14 Інформація про всіх зареєстрованих користувачів

Висновки до третього розділу

В даному розділі було наведено порядок встановлення та налаштування додатку. Було детально зображено структури інтерфейсу додатку та принципи взаємодії з ним. Покроково пояснено (з відповідними зображеннями) як використовувати основні функції системи.

У третьому підрозділі описано результати тестування додатку за допомогою інструменту Google Lighthouse. Перевірка основних характеристик показала хороші результати.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було спроектовано та реалізовано блог-платформу. Для цього було вирішено наступні завдання:

1. Проведений аналіз предметної області та поглиблене вивчення тематики дозволив визначити основні задачі, що необхідні для реалізації блог-платформи. Огляд наявних аналогів показав, що головними функціями та рисами майбутньої системи мають бути: простота у використанні; наявність статистики публікацій: коментарів, переглядів; наявність статистики підписників (по віковій категорії, країнам та за статтю); можливість експорту даних в форматі документу.

2. Визначившись з оптимальною архітектурою MVC, обрано необхідний та достатній набір програмних продуктів для реалізації поставлених задач.

3. Було проаналізовано функціональні та нефункціональні вимоги до блогу, розроблено структуру бази даних для збереження усієї необхідної інформації для повноцінної роботи системи.

4. Розглянуто алгоритми основних процесів системи та наведено детальний розгляд головних аспектів реалізації web-додатку.

5. Також було описано порядок встановлення і налаштування додатку та проведено тестування системи.

В результаті проведеної роботи було вирішено поставлену задачу та розроблено систему веб-блогу ігор. Реалізований програмний комплекс готовий до використання.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Обробка природної мови [Електронний ресурс] – Режим доступу: <https://artsandculture.google.com/entity/m05flf?hl=uk>.
2. Аналіз настроїв із використанням бібліотек НЛП [Електронний ресурс] – Режим доступу: <https://ichi.pro/ru/analiz-nastroenij-s-ispol-zovaniem-bibliotek-nlp-245981930982197>.
3. Просування сайту [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Просування_сайту.
4. Instagram [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Instagram>.
5. Кращі сервіси для веб-аналітики [Електронний ресурс] – Режим доступу: <https://www.ukraine.com.ua/uk/blog/web-analytics/luchshie-servisi-dlya-veb-analitiki.html>.
6. Що таке веб-аналітика, а також сервіси та системи для її проведення [Електронний ресурс] – Режим доступу: https://medianation.ru/blog/analitika/chto-takoe-web-analitika-a-takzhe-servisy-i-sistemy-dlya-eye-provedeniya/#yakor_2.
7. Матеріали десятої міжнародної наукової конференції студентів та молодих вчених [Електронний ресурс] – Режим доступу: <https://sci.ldubgd.edu.ua/bitstream/123456789/7584/1/%D0%9C%D0%A22020.pdf>.
8. Переваги PHP [Електронний ресурс] – Режим доступу: http://site-konstruktor.com.ua/php/perevagy_php/index.html.
9. MySQL Workbench, візуальний інструмент для проектування баз даних [Електронний ресурс] – Режим доступу: <https://ubunlog.com/uk/бази-даних-робочого-середовища-mysql/>.
10. Вимоги до інтерфейсів інформаційних систем [Електронний ресурс] – Режим доступу: <https://lib.chmnu.edu.ua/pdf/posibnuku/313/7.pdf>.
11. Породжувальні патерни проектування [Електронний ресурс] – Режим доступу: <https://refactoring.guru/uk/design-patterns/creational-patterns>.
12. Поведінкові патерни проектування [Електронний ресурс] – Режим доступу: <https://refactoring.guru/uk/design-patterns/behavioral-patterns>.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

13. Kaggle [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Kaggle>.

14. sklearn.neighbors.KNeighborsClassifier [Електронний ресурс] – Режим доступу: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>.

15. Модуль pickle [Електронний ресурс] – Режим доступу: <https://pythoner.name/pickle>.

16. jQuery UI [Електронний ресурс] – Режим доступу: <http://jqueryui.com/>.

17. PHP MAILER [Електронний ресурс] – Режим доступу: <https://github.com/PHPMailer/PHPMailer>.

18. HTML Styles - CSS Tutorial [Електронний ресурс] – Режим доступу: <https://www.w3schools.com/Html/>.

19. Що таке сайт? [Електронний ресурс] – Режим доступу: <https://2ip.ua/ua/blog/website>.

20. Що означає середня швидкість інтернету. Який швидкості інтернету досить [Електронний ресурс] – Режим доступу: <https://androidas.ru/which-means-the-average-speed-of-the-internet-what-is-the-speed-of-the-internet/>.

21. Lighthouse [Електронний ресурс] – Режим доступу: <https://developers.google.com/web/tools/lighthouse>.

ДОДАТКИ

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

Config

```

<?php
$config =[
    'Database' =>[
        'Server' => 'localhost','Username' =>
        'root', 'Password' => '', 'Database' =>
        'DBBs'
    ],
    'BlogsCount' => 10
];

<?php

namespace controllers;

use core\Controller;

class Blogs extends Controller
{
    protected $user;
    protected
    $blogModel;
    protected
    $userModel;
    public function __construct()
    {
        $this->userModel = new \models\Users();
        $this->blogModel = new \models\Blogs();
        $this->user = $this->userModel->GetCurrentUser();
    }

    public function actionIndex()
    {
        global $Config;
        $title = 'Блоги';
        $Blogs = $this->blogModel->GetBlogs($Config['BlogsCount']);
        return $this->render('index',['blogs'=>$Blogs,['Title'=>$title]);
    }

    public function actionView()
    {
        $id = $_GET['id'];
        $blogs = $this->blogModel->GetBlogsById($id);
        $title = $blogs['title'];
    }
}

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return $this->render('view',['model'=>$blogs,['Title'=>$title]);
    }

    public function actionSearch()
    {
        $query = $this->blogModel->searchBlog($_POST['search']);
        $row = mysqli_fetch_assoc($query);
        return
    $this->render('search',['blogs'=>$row,['Title'=>'Пошук']);
    }

    public function actionInfo()
{
    global $Config;
    $title = 'AdminInfo';

    $Users = $this->blogModel->GetUser($Config['BlogsCount']);
    return $this->render('info',['users'=>$Users,['Title'=>$title]);
}
    public function actionAdd()
    {
        $titleForbidden = 'Доступ
заборонено';if ($this->user ==
null) {
            return $this->render('forbidden',
                null, ['Title' =>
                    $titleForbidden]);
        }
        $title = 'Додавання
        блогу';if ($this-
        >isPost()) {
            $result = $this->blogModel-
            >addBlog($_POST);if ($result['error']
            === false) {
                $allowed_types = ['image/png',
                'image/jpeg'];
                if(is_file($_FILES['File']['tmp_name']
                ) &&
                in_array($_FILES['File']['type'], $allowed_types)){
                    switch($_FILES['File']['ty
                    pe']){ case
                    'image/png':
                        $extension =
                        'png';break;
                    default:
                        $extension = 'jpg';
                    }
                    $name =
                    $result['id'].'_'.uniqid().'.'.$extension;
                    move_uploaded_file($_FILES['File']['tmp_nam
                    e'],
                    'files/blogs/'.$name);
                    $this->blogModel->ChangePhoto($result['id'], $name);
                }
                return $this->renderMessage('ok', 'Блог успішно додано',
                null, [
                    'Title' => $title]);
            } else {
                $message = implode('<br/>',
                $result['messages']); return $this-

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        >render('form', ['model'=> $_POST], [
            'Title' => $title,
            'MessageText' =>
                $message,
            'MessageClass' =>
                'danger']);
    }
}
else{
    return $this->render('form', ['model'=>
        $_POST], ['Title' => $title]);
}
}
public function actionChange()
{
    $id = $_GET['id'];
    $blog = $this->blogModel->GetBlogsById($id);
    $title = 'Редарування
    блять';if ($this-
    >isPost()) {
        $result = $this->blogModel-
        >UpdateBlogs($_POST, $id);if ($result ===
        true) {
            $allowed_types = ['image/png',
            'image/jpeg'];if
            (is_file($_FILES['File']['tmp_name'])
            &&
            in_array($_FILES['File']['type'], $allowed_types)) {
                switch
                    ($_FILES['File']['type']
                    ) {case 'image/png':
                        $extension =
                        'png';break;
                    default:
                        $extension = 'jpg';
                }
                $name = $id . '_' . uniqid() . '.' . $extension;

                move_uploaded_file($_FILES['File']['tmp_name'],
                'files/blogs/' . $name);
                $this->blogModel->ChangePhoto($id, $name);
            }
            return $this->renderMessage('ok', 'Блог оновлено',
            null, ['Title' => $title,
            'NameContent' => $title
            ]);
        } else {
            $message = implode('<br/>', $result);
            return $this->render('form', ['model' =>
                $blog], ['Title' => $title,
                'NameContent' =>
                $title,
                'MessageText' =>
                $message,
                'MessageClass' =>
                'danger'
            ]);
        }
    }
}
return $this->render('form', null, ['Title' =>
$title, 'NameContent' => $title]);
}
public function actionDelete()
{

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $title = 'Видалення блогу';
        $id = $_GET['id'];
        if (isset($_GET['confirm']) && $_GET['confirm'] == 'yes')
        {
            if ($this->blogModel->DeleteBlog($id)) {
                header('Location: /blogs/');
            }else
                return $this->renderMessage('error','Помилка видалення
новини',null,[
                    'Title' =>
                        $title ,
                    'NameContent' =>
                        $title
                ]);
        }
        $blog = $this->blogModel->GetBlogsbyId($id);
        return $this->render('delete',['model'=>$blog,['Title'=>$title]);
    }
    public function actionlist()
    {
        echo 'actionlist';
    }
}

<?php

namespace

controllers;use

core\Controller

;

class Site extends Controller
{
    public function actionIndex()
    {
        return $this->
            >render('index',null,[
                'Title' => 'Game Time',
                'NameContent' =>
                'Stalker', 'Content' =>
                'Lorem ipsum'
            ]);
    }
}

<?php

namespace

controllers;use

core\Controller

;

class Users extends Controller
{
    protected

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$usersModel;
function
construct()
{
    $this->usersModel = new \models\Users();
}
public function actionLogout()
{
    $title = 'Вихід з
акаунту';
unset($_SESSION['acou
nt']);
return $this->renderMessage('ok', 'Ви вийшли з
акаунту', null, ['Title' => $title]);
}
public function actionLogin()
{
    $title = 'Вхід на сайт';
if(isset($_SESSION['acoun
t']))
    return $this->renderMessage('ok', 'Ви уже увійшли на
сайт', null, ['Title' => $title]);
    if ($this->isPost())
    {
        $user =
$this->usersModel->AuthUser($_POST['user'],$_POST['password'])
; if (!empty($user))
        {
            $_SESSION['account'] = $user;
return $this->renderMessage('ok', 'Ви успішно
увійшли на сайт', null, ['Title' => $title]);
        }
        else
        {
            return $this->render('login',null,[
                'Title'=>$title,
                'MessageText'=>'Неправильний логін або
пароль', 'MessageClass'=>'danger'
            ]);
        }
    }else {
        $params = ['Title'=>$title];
        return $this->render('login',null,$params);
    }
}
public function actionRegister()
{
    if ($this->isPost()) {
        $result = $this->usersModel->AddUsers($_POST);if ($result === true)
        {
return $this->renderMessage('ok', 'Користувач успішно створений', null, ['Title' => 'Реєстрація на сайті']);
        }
    }
    else {
        $message = implode("<br/>",$result); return $this->render('register',null,
        [

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'Title'=>'Реєстрація на сайті',
        'MessageText'=>$message,
        'MessageClass'=>'danger'
    );
    }
}
else {
    $params = ['Title'=>'Реєстрація на сайті']; return $this->render('register',null,$params);
}
}
}

<?php
namespace
    ace
    core;

class Controller
{
    public function isPost()
    {
        if ($_SERVER['REQUEST_METHOD'] == 'POST')
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public function isGet()
    {
        if ($_SERVER['REQUEST_METHOD'] == 'GET')
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public function postFilter($fields)
    {
        return Utils::ArrayFilter($_POST,$fields);
    }

    public function render($viewName, $localParams = null, $globalParams =null)
    {
        $tpl = new Template();
        if (is_array($localParams))

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$tpl->setParams($localParams);
if (!is_array($globalParams))

    $globalParams = [];

    $moduleName = strtolower((new
    \ReflectionClass($this))->getShortName());

    $globalParams['Content'] =
$tpl->render("views/{$moduleName}/{ $viewName }.php"); return
    $globalParams;

}

public function renderMessage($type,$message,$localParams =null,$globalParams = null)

```

```

switch ($type)
{
    case 'ok':
        $tpl-
        >setParam('MessageClass','success');
        break;
    case 'error':
        $tpl-
        >setParam('MessageClass','danger');
        break;
    case 'info':
        $tpl-
        >setParam('MessageClass','info');
        break;
}
if (!is_array($globalParams))
    $globalParams = [];
$globalParams['Content'] = $tpl-
>render("views/layout/message.php");return $globalParams;
}
}

```

```

<?php

namespace

e core;

class

Core
{
    private static
    $instance; private
    static $mainTemplate;
    private static $db;
    private function __construct()
    {
        global $config;
        spl_autoload_register('\core\Core::
        autoload'); self::$db = new
        DB($config['Database']['Server'],
            $config['Database']['Username'],
            $config['Database']['Password'],
            $config['Database']['Database']);
    }
    public static function getInstance()
    {

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if(empty(self::$instance))
        { self::$instance =
          new Core();return
          self::getInstance();
        }
        else
            return self::$instance;
    }
    public function getDB()
    {
        return self::$db;
    }
    public function init()
    {
        session_start();
        self::$mainTemplate = new Template();
    }
    public function run()

        $fullclassname =
        'controllers\\Site';else
        $fullclassname = 'controllers\\'.$classname;
        $methodname =
        ucfirst($pathParth[1]);if
        (empty($methodname))
        $fullmethodname =
        'actionIndex';else
        $fullmethodname =
        "action".$methodname;
        if(class_exists($fullclassname)) {
            $controller = new $fullclassname();
            if (method_exists($controller, $fullmethodname)) {
                $method = new
                \ReflectionMethod($fullclassname,$fullmetho
                dname);
                $paramsarray = [];
                foreach ($method -> getParameters() as $parameter)
                {
                    array_push($paramsarray,isset($_GET[$parameter->name])
                    ?
                    $_GET[$parameter->name] : null);
                }
                $result = $method-
                >invokeArgs($controller,$paramsarray);if
                (is_array($result))
                {
                    self::$mainTemplate->setParams($result);
                }
            }
            else
                throw new \Exception('404 not found');
        }
        else
            throw new \Exception('404 not found');
    }
    public function done()
    {
        self::$mainTemplate->display("views/layout/index.php");
    }
    public static function __autoload($className)
    {
        $filename =
        $className.".php";if
        (is_file($filename))

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        {
            include ($filename);
        }
    }
}

<?php

namespace

e Core;

use models\Users;

class DB
{
    protected $pdo;

    public function __construct($server, $login, $password, $database)
    {
        $this->pdo = new
        \PDO("mysql:host={$server};dbname={$database};charset=UTF8", $login,
        $password);

        public function select($table, $fields="*", $where = null, $orderby =
        null,
        $limit=null, $offset=null)
        {
            $fieldsStr = "*";
            if (is_string($fields)) {
                $fieldsStr = $fields;
            }
            if (is_array($fields)) {
                $fieldsStr = implode(' ', $fields);
            }
            $sql = "SELECT {$fieldsStr} FROM {$table}";
            if (is_array($where) && count($where) > 0) {
                $whereParts = [];
                foreach ($where as $key => $value) {
                    $whereParts [] = "{$key} = ?";
                }
                $whereStr = implode(' AND ', $whereParts);
                $sql .= ' WHERE ' . $whereStr;
            }
            if (is_string($where)) {
                $sql .= ' WHERE ' . $where;
            }
            if (is_array($orderby)) {
                $orderByParts = [];
                foreach ($orderby as $key => $value) {
                    $orderByParts [] = "{$key} {$value}";
                }
                $sql .= ' ORDER BY ' . implode(' ', $orderByParts);
            }
            if (!empty($limit)) {
                if (!empty($offset)) {
                    $sql .= " LIMIT {$offset}, {$limit}";
                } else {
                    $sql .= " LIMIT {$limit}";
                }
            }
            $sth = $this->pdo->prepare($sql);

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (is_array($where) && count($where) > 0) {
            $sth->execute(array_values($where));
        } else {
            $sth->execute();
        }
        return $sth->fetchAll();
    }

    public function insert($table, $row){
        $fieldsStr = implode(' ', array_keys($row));
        $valuesParts = [];
        foreach ($row as $key => $value){
            $valuesParts []= '?';
        }
        $valuesStr = implode(' ', $valuesParts);

        $sql = "INSERT INTO {$table} ({$fieldsStr}) VALUES ({$valuesStr})";
        $sth = $this->pdo->prepare($sql);
        $sth->execute(array_values($row));
        ; return $this->pdo->lastInsertId();
    }

    public function delete($table, $where = null){
        $sql = "DELETE FROM {$table}";
        if (is_array($where) && count($where) > 0) {
            $whereParts = [];
            foreach ($where as $key => $value) {

                $whereParts [] = "{$key} = ?";

            }
            $whereStr = implode(' AND ', $whereParts);
            $sql .= ' WHERE ' . $whereStr;
        }
        if (is_string($where)) {
            $sql .= ' WHERE ' . $where;
        }
        $sth = $this->pdo->prepare($sql);
        if (is_array($where) && count($where) > 0) {
            $sth->execute(array_values($where));
        } else {
            $sth->execute();
        }
    }

    public function update($table, $newRow, $where){
        $sql = "UPDATE {$table} SET ";
        $setParts = [];
        $paramsArr = [];

        foreach($newRow as $key=> $value){
            $setParts [] = "{$key} = ?";
            $paramsArr [] = $value;
        }
        $sql .= implode(' ', $setParts);
        if (is_array($where) && count($where) > 0) {
            $whereParts = [];
            foreach ($where as $key => $value) {
                $whereParts [] = "{$key} = ?";
                $paramsArr [] = $value;
            }
            $whereStr = implode(' AND ', $whereParts);

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $sql .= ' WHERE ' . $whereStr;
    }
    if (is_string($where)) {
        $sql .= ' WHERE ' . $where;
    }
    $sth = $this->pdo->prepare($sql);
    $sth->execute($paramsArr);
}

}

<?php

namespace

    core;

class

Template
{
    protected $parameters;
    public function __construct()
    {
        $this->parameters = [];
    }
    public function setParam($name,$value)
    {
        $this->parameters[$name] = $value;
    }
    public function getParam($name)
{
        return $this->parameters[$name];
    }
    public function setParams($array)
    {
        foreach ($array as $key => $value)
            $this->parameters[$key] = $value;
    }
    public function render($path)
    {
        extract($this->parameters); ob_start();
        include ($path);
        $html = ob_get_contents();
        ob_end_clean();
        return $html;
    }
    public function display($path)
    {
        echo $this->render($path);
    }
}

```

Core Utils

```

<?php

namespace

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

```

e Core;

class
Utils
{
    public static function ArrayFilter($row,$fields)
    {
        $newRow = [];
        foreach ($fields as
            $field) if
                (isset($row[$fiel
                    d]))
                    $newRow[$field] =
                        $row[$field];return $newRow;
    }
}

<?php

namespace models;

use
Core\Mod
el; use
Core\Uti
ls; use
Imagick;

class Blogs extends Model
{
    public function ChangePhoto($id, $file){
        $folder = 'files/blogs/';
        $file_path = pathinfo($folder.$file);
        $file_big = $file_path['filename'].'_b';
        $file_medium = $file_path['filename'].'_m';
        $file_small = $file_path['filename'].'_s';
        $bags = $this->GetBlogsById($id);
        if(is_file($folder.$bags['Photo'].'_b.jpg') &&
is_file($folder.$file)){
            unlink($folder.$bags['Photo'].'_b.jpg');

                if(is_file($folder.$bags['Photo'].'_m.j
pg') &&is_file($folder.$file)){
                    unlink($folder.$bags['Photo'].'_m.jpg');
                }
                if(is_file($folder.$bags['Photo'].'_s.j
pg') &&is_file($folder.$file)){
                    unlink($folder.$bags['Photo'].'_s.jpg');
                }
                $bags['Photo'] = $file_path['filename'];
                $im_b = new Imagick();
                $im_b->readImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.$file);
                $im_b->cropThumbnailImage(600,500,true);

                $im_b-
>writeImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.'/'.$file_big.'.jpg')
;

                $im_m = new Imagick();
                $im_m->readImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.$file);
                $im_m->cropThumbnailImage(300,300,true);

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$im_m-
>writeImage($_SERVER['DOCUMENT_ROOT'].'/.'.$folder.'/'.$file_medium.'.jpg');

    $im_s = new Imagick();
    $im_s->readImage($_SERVER['DOCUMENT_ROOT'].'/.'.$folder.$file);
    $im_s->cropThumbnailImage(250,250,true);

$im_s-
>writeImage($_SERVER['DOCUMENT_ROOT'].'/.'.$folder.'/'.$file_small.'.jpg');
);

    unlink($folder.$file);
    $this->Updateblogs($bags, $id);
}
public function searchBlog($searchs)
{
    $connect =
mysqli_connect('localhost','root','','DBBs') or
die('error');
    $search = explode(" ", $searchs);
    $count = count($search);
    $array = array();
    $i = 0;
    foreach ($search as $key) {
        $i++;
        if ($i<$count) $array [] = "CONCAT('title','text') LIKE
'%" . $key . "%' ";else $array [] = "CONCAT('title','text') LIKE '%" . $key . "%'";
    }
    $sql = "SELECT * FROM Blogs WHERE ".implode(" ", $array);
    $query =
mysqli_query($connect,$sql);return
$query;
}
public function addBlog($row)
{
    $userModel = new \models\Users();
    $user = $userModel-
>GetCurrentUser();if ($user ==
null)
    {
        $result = [
            'error'
            => true,
            'messages' => ['Користувач не аутентифікований']
        ];
        return $result;
    }
    $validateResult = $this-
>Validate($row);if
(is_array($validateResult)){
        $result = [
            'error' => true,
            'messages'=> $validateResult
        ];
        return $result;
    }
    $fields = ['title','text','short_text'];
    $RowFiltered = Utils::ArrayFilter($row,$fields);
    $RowFiltered['datetime'] = date('Y-m-d H:i:s');
    $RowFiltered['user_id'] = $user['id'];
    $rowFiltered['photo'] = 'photo';
    $id =
\core\Core::getInstance()->getDB()-

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        >insert('Blogs',$RowFiltered); return [
            'error' =>
            false,'id'
            => $id
        ];
    }
    public function GetBlogs($count)
    {
        return
        \core\Core::getInstance()->getDB()-
        >select('Blogs','*',null,['datetime' =>'DESC'],$count);
    }
    public function GetUser($count)
    {
        return
        \core\Core::getInstance()->getDB()-
        >select('Users','*',null,['id' =>'DESC'],$count);
    }
    public function GetBlogsbyId($id)
    {
        $blog =
        \core\Core::getInstance()->getDB()-
        >select('Blogs','*',['id'=>$id]); if (!empty($blog))
            return
            $blog[0];else
            return null;
    }
    public function DeleteBlog($id)
    {
        $blog = $this->GetBlogsbyId($id);
        $userModel = new \models\Users();
        $user = $userModel->GetCurrentUser();
        if (empty($blog) || empty($user) || $user['user'] !=
            'Admin')return false;
        else
        \core\Core::getInstance()->getDB()-
        >delete('Blogs',['id'=>$id]); return true;
    }
    public function UpdateBlogs($row,$id)
    {
        $userModel = new \models\Users();
        $user = $userModel-
        >GetCurrentUser();if ($user ==
        null)
            return false;
        $validateResult = $this-
        >Validate($row);if
        (is_array($validateResult))
            return $validateResult;
        $fields = ['title','text','short_text','photo'];
        $RowFiltered = Utils::ArrayFilter($row,$fields);
        $RowFiltered['datetime_change'] = date('Y-m-d H:i:s');

        \core\Core::getInstance()->getDB()-
        >update('Blogs',$RowFiltered,['id'=>$id]); return true;
    }
    public function Validate($row)

    {
        $errors = [];
        if (empty($row['title']))
            $errors [] = 'Поле title не може бути
            порожнім';if (empty($row['text']))
            $errors [] = 'Поле text не може бути

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        порожнім';if (empty($row['short_text']))
            $errors [] = 'Поле short_text не може бути
        порожнім';if (count($errors) > 0)
            return
        $errors;else
            return true;
    }
}

<?php

namespace models;

use
Core\Mod
el;use
Core\Uti
ls;

class Users extends Model
{
    public function Validate($formRow)
    {
        $errors = [];
        if (empty($formRow['user']))
            $errors [] = 'Поле логін не може бути порожнім';
        $user = $this->
        >GetUserByLogin($formRow['user']);
        if(!empty($user))
            return $errors [] = 'Користувач з вказаним логіном
        існує';if (empty($formRow['password']))
            $errors [] = 'Поле пароль не може бути
        порожнім';if ($formRow['password'] !=
        $formRow['password2'])
            $errors [] = 'Паролі не
        співпадають';if
        (empty($formRow['nickname']))
            $errors [] = 'Поле nickname не може бути
        порожнім';if (count($errors) > 0)
            return
        $errors;else
            return true;
    }
    public function AddUsers($userRow)
    {
        $validateResult = $this->
        >Validate($userRow);if
        (is_array($validateResult))
            return $validateResult;
        $fields = ['user','password','nickname'];
        $userRowFiltered = Utils::ArrayFilter($userRow,$fields);
        $userRowFiltered['password'] = md5($userRowFiltered['password']);
        \core\Core::getInstance()->getDB()-
        >insert('Users',$userRowFiltered); return true;
    }
    public function IsUserAuth()
    {
        return isset($_SESSION['account']);
    }
    public function GetCurrentUser()
    {
        if ($this->IsUserAuth())

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return $_SESSION['account'];else
        return null;
    }
    public function AuthUser($login,$password)
    {
        $password = md5($password);
        $users = \core\Core::getInstance()->getDB()->select('Users','*',[
            'user'=>$login, 'password'=>$password
        ]);
        if (count($users) > 0)
        {
            $user = $users[0];return
            $user;
        } else return false;
    }
    public function GetUserByLogin($login)
    {
        $rows = \core\Core::getInstance()->getDB()->select('Users','*',['user'
=> $login]);
        if (count($rows) > 0)return
            $rows[0];
        else
            return null;
    }
}

<div class="test" style="margin-top: 8%">
<p>
    Ви хочете видалити блог <b><?=$model['title'] ?></b>?, повернути йогобуде не можливо!!!
</p>
<p>
    <a class="floating-button-red" href="/blogs/delete?id=<?=$model['id'] ?>&confirm=yes"
>Видалити</a>
    <a class="floating-button-blue" href="<?=$
$_SERVER['HTTP_REFERER'] ?>" >Відмінити</a>
</p>
</div>
<div class="alert-danger" style="margin-top: 8%">
    Доступ заборонено, або недостатньо прав для використання
</div>
<form class="transparent" method="post" action="" enctype="multipart/form-data" >
    <div class="form-inner" style="margin-top: 8%">
        <hr>
        <label for="title"><b>Заголовок блогу</b></label>

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <input type="text" placeholder="Введіть заголовок блогу"
value="<?=$model['title'] ?>" name="title">
        <label for="psw-repeat"><b>Введіть короткий опис блогу</b></label>
        <textarea type="text" name="short_text"><?=$model['short_text'] ?></textarea>
        <hr>

        <label for="text"><b>Фотографія блогу</b></label>
        <input type="file" accept="image/jpeg , image/png" name="photo" id="photo">
        <hr>
        <? if(is_file('files/blogs/'.$model['photo'].'_b.jpg')) : ?>
        
        <? endif; ?>
        <input type="submit" value="Зберегти">
    </div>
</form>

<?php
    $userModel = new \models\Users();
    $user = $userModel->GetCurrentUser();
    ?>
    <? if($userModel->IsUserAuth() || $user['user'] == 'Admin') : ?>
        <div class="Title" style="display: flex; padding-top: 6%;">
            <a href="/blogs/add" class="floating-button-green">Створити блог</a>
        </div>
        <div class="Title" style="display: flex; padding-top: 1%">
            <hr style="background-color: #4a3f73;width: 39%;height: 1%;">
            <div class="Caption">Блогі</div>
            <hr style="background-color: #4a3f73;width: 37%;height: 1%;">
        </div>
    <? else: ?>
    <div class="Title" style="display: flex; padding-top: 10%; ">
        <hr style="background-color: #4a3f73;width: 39%;height: 1%;">
        <div class="Caption">Блогі</div>
        <hr style="background-color: #4a3f73;width: 37%;height: 1%;">
    </div>
    <? endif; ?>
    <?php foreach ($blogs as $blog) : ?>
        <div class="blogs" style="display: flex; padding-top: 8%;margin-right:auto; margin-left:
auto; width: 1130px">
            <div class="BackgroundBlog" style="background-color: #dadada;border:4px solid
#4a3f73;width: 1133px; height: 500px;border-radius: 15px">
                <div class="ImageBlog">
                    <? if ($blog['id'] == 25) : ?>

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <? endif; ?>

        <? if ($blog['id'] == 26) : ?>

        <? endif; ?>

        <? if ($blog['id'] == 27) : ?>

            
all...</a>

        <? if($user['user'] == 'Admin' || $blog['user_id'] ==

$user['id'] ): ?>

            <a class="floating-button-green" href="/blogs/change?id=<?=$blog['id'] ?>"
style="position: absolute;display:inline-block;margin-left: 24%;margin-top: 12%;font-family: 'Segoe UI';font-size:
20px">edit</a>

            <a class="floating-button-red" href="/blogs/delete?id=<?=$blog['id'] ?>"
style="position: absolute;display:inline-block;margin-left: 24%;margin-top: 22%;font-family: 'Segoe UI';font-size:
20px">delete</a>

        <? endif; ?>

    </div>

</div>

</div>

</div>

<?php endforeach; ?>

<table>

    <tr>

        <th>id</th>

        <th>name</th>

        <th>user</th>

        <th>acces</th>

        <th>password</th>

    </tr>

    <? foreach ($users as $user) :?>

    <tr>

        <td><?=$user['id'] ?></td>

        <td><?=$user['nickname']?></td>

        <td><?=$user['user'] ?></td>

        <td><?=$user['acces'] ?></td>

        <td><?=$user['password'] ?></td>

    </tr>

    <? endforeach; ?>

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

</table>

Views/blogs/search

<?php

\$userModel = new \models\Users();

\$user = \$userModel->GetCurrentUser();

?>

<div class="Title" style="display: flex; padding-top: 10%">

<hr style="background-color: #4a3f73;width: 39%;height: 1%;">

<div class="Caption">Блог</div>

<hr style="background-color: #4a3f73;width: 37%;height: 1%;">

</div>

<div class="blogs" style="display: flex; padding-top: 8%;margin-right:auto; margin-left:auto; width: 1130px">

<div class="BackgroundBlog" style="background-color: #dadada;border:4px solid #4a3f73;width: 1133px; height: 500px;border-radius: 15px">

<div class="ImageBlog">

<? if (\$blogs['id'] == 25) : ?>

<? endif; ?>

</div>

<div class="NameBlog"><h2><?=\$blogs['title'] ?></h2>

<div class="ContentText"><?=\$blogs['short_text'] ?></div>

<div class="ReadAll"><a class="floating-button-blue" href="/blogs/view?id=<?=\$blogs['id'] ?>" style="position: absolute;display:inline-block;margin-left: 24%;font-family: 'Segoe UI';font-size: 20px">Readall...

<? if(\$user['user'] == 'Admin' ||\$blogs['user_id'] ==

\$user['id']): ?>

<a class="floating-button-green" href="/blogs/change?id=<?=\$blogs['id'] ?>" style="position: absolute;display:inline-block;margin-left: 24%;margin-top: 12%;font-family: 'Segoe UI';font-size: 20px">edit

<a class="floating-button-red" href="/blogs/delete?id=<?=\$blogs['id'] ?>" style="position: absolute;display:inline-block;margin-left: 24%;margin-top: 22%;font-family: 'Segoe UI';font-size: 20px">delete

<? endif; ?>

</div>

</div>

</div>

</div>

<div class="ImageBlog" style="margin-top: 8%">

<? if (\$model['id'] == 25) : ?>

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <a href="https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcTnkXX1msb3FcwUKdveOb4VJ_8dlsezqUlqEQ&usqp
=CAU" data-fancybox="gallery">

        </a>

    <? endif; ?>

    <? if ($model['id'] == 26) : ?>

        <a href="https://escorenews.com/media/news/n29361.jpeg" data-
fancybox="gallery">

            </a>

        <? endif; ?>

    <? if ($model['id'] == 27) : ?>

        <a href="https://i.pinimg.com/originals/f2/12/75/f21275716a197d57bfeaa6e7c065cb2
9.png" data-fancybox="gallery">

            </a>

        <? endif; ?>

    </div>

    <div class="NameBlog"><h2><?=$model['title'] ?></h2>

        <div class="ContentText"><?=$model['text'] ?></div>

    </div>

?>

<!DOCTYPE html>
<html>
    <head>
        <title><?=$Title ?></title>
        <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/@fancyapps/ui/dist/fancybox.css"/>
        <script
src="https://cdn.jsdelivrivr.net/npm/@fancyapps/ui@4.0/dist/fancybox.umd.js">
        </s cript>
        <style>
            .StockImage{
                background-image: url("https://p19-
marketing-va.bytedgame.com/img/g-marketing-assets-
va/2022_10_18_09_19_18/kv2_s391770.jpg~q80.jpg");
                height: 862px;
                background-size:
                cover;
                background-repeat: no-
repeat;position:
                absolute;
            }
            a {
                outline: none;

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        text-
        decoration:
        none;padding:
        2px 1px 0;
    }

a:link {
    color: gray;
}

a:visited {
    color: black;
}

a:focus {
    border-bottom: 1px solid;
}

a:hover {
    border-bottom: 1px
    solid;color: gray;
}

a:active {
    color: gray;
}
.Search,.S
ubmit{
    width:
    428px;
    height
    :
    48px;
    font-family:
    inherit;font-
    size: 1rem;
    font-weight: 400;
    line-height: 1.5;
    color: #000000;
    background-color:
    #dadada; background-
    clip: padding-box;
    border: 1px solid
    #bdbdbd; border-
    radius: 10px;
    transition: border-color 0.15s ease-in-out, box-shadow
    0.15s
    ease-in-out;
}

.Search::placeholder,.Submit::placeholder{

    color: #212529;
    opacity: 0.7;
}

.Search:focus,
.Submit:focus{
    color: #212529;
    background-color:
    #fff; border-
    color: #bdbdbd;
    outline: 0;
    box-shadow: 0 0 0 0.2rem rgba(158, 158, 158, 0.25);
}

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    .SearchBTn,.JoinBTn,.RegistrationBTn,.SubmitBTn {size: 20px;
    background-color:
    #4a3f73;border-
    color: #4a3f73;
    color: white;
    font-family:
    Arial; font-
    size: 20px;
    display: inline-
    block; box-
    sizing: border-
    box;padding: 0
    25px;
    margin: 0 15px
    15px 0;outline:
    none;
    border-
    radius: 15px;
    height: 46px;
    line-
    height:
    46px;font-
    weight:
    600;
    text-decoration: none;
    box-shadow: 0 4px 6px rgb(65 132 144 / 10%), 0 1px 3px
    0 0 / 8%);
    cursor:
    pointer;
    user-
    select:
    none;
    appearance:
    none;
    touch-action:
    manipulation;
    vertical-align: top;
    transition: box-
    shadow 0.2s;
    }
    .SearchBTn:active,.JoinBTn:active,.RegistrationBTn:active,.Submit
    BTn:active {
        background-color: #9a80a6;
    }
    .SearchBTn:hover,.JoinBTn:hover,.RegistrationBTn:hover,.SubmitB
    Tn:
    :hover {
        box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px
    50px 0rgba(0,0,0,0.19);
    }
    .RegistrationInf
    o { margin-
    left: auto;
    margin-
    right: auto;
    width:
    735px;
    padding-top:
    543px;
    }

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        .Caption {
            color: black;
            font-family:
            "Segoe UI";font-
            style: italic;
            font-size:
            60px;
            position:
            absolute;
            padding-
            left: 45%;
        }
        .ImageBlog{
            width:
            390px;
            height:
            390px;
background-size: cover;

            background-repeat: no-
            repeat;border-radius:
            15px;
            margin-left: 4%;
            margin-top: 5%;
            display:
            inline-block;
        }
        .NameBlog{
            text-align:
            center;
            width:
            500px;
            height:
            350px;
            display:
            inline-block;
            position:
            absolute;
            margin-left:
            7%;
            margin-top:
            5%; font-
            family:
            Arial;
        }
        .ImageFooter {
            background
            -image:
            url("https://phonoteka.org/uploads/posts/2021-04/1618607889_4-
            phonoteka_org- p-fon-igrovoi-tematiki-5.jpg");
            width:
            100%;
            height:
            300px;
            background-size:
            cover; background-
            repeat: no-repeat;
        }
    </style>
<style>
    * {
        box-sizing: border-box;
    }
    .transparent {

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        position:
        relative;
        max-width:
        400px;
        padding:
        60px 50px;
        margin: 50px
        auto 0; margin-
        top: 10%;
        background-size:
        cover;
    }
    .transparent:before {
        content: "";
        position:
        absolute;
        top: 0;
        left: 0;
        right: 0;
        bottom: 0;
        background: linear-gradient(to right bottom, #dadada,
        gray);
    }
    .form-inner {
        position: relative;
    }
    .form-inner h3 {
        position:
        relative;
        margin-top:
        0; color:
        white;
        font-family: 'Roboto',
        sans-serif;font-weight:
        300;
        font-size: 26px;
        text-transform: uppercase;
    }
    .form-inner
    h3:after {
        content: "";
        position:
        absolute;
        left: 0;

        height:
        2px;
        width:
        :
        60px;
        background: #1762EE;
    }
    .form-inner
    label {
        display:
        block;
        padding-
        left: 15px;
        font-family: 'Roboto',
        sans-serif;color: rgba(255,
        255, 255, .6); text-

```

bottom: -6px;

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        transform: uppercase;
        font-size: 14px;
    }
    .form-inner
        input {
            display:
            block;
            width:
            100%;
            padding: 0
            15px; margin:
            10px 0 15px;
            border-width:
            0; line-
            height: 40px;
            border-
            radius: 20px;
            color: white;
            background: rgba(255, 255,
            255, .2);font-family:
            'Roboto', sans-serif;
        }
    .form-inner
        input[type="checkbox"] {
            position: absolute;
            opacity: 0;
        }
    #custom-
        checkbox+label
        1 {position:
        relative;
        margin: 20px
        0;
        text-
        transform:
        none;cursor:
        pointer;
    }
    #custom-
        checkbox+label:before
        {content: "";
        display:
        inline-block;
        width: 20px;
        height:
        20px;
        margin-
        right: 10px;
        vertical-align:
        text-top;
        background: white;
    }
    #custom-
        checkbox:checked+label:before
        {background: #1762EE;
    }
    #custom-
        checkbox:checked+label:after
        {content: "";
        position:
        absolute;
        width: 2px;
        height:
        10px;
        background:
        white;
    }

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        2px;
        left:
        20px;
        top:
        9px;
        background: white;
        box-shadow: 2px 0 0 white, 4px 0 0 white, 4px -2px 0
white, 4px -4px 0 white, 4px -6px 0 white, 4px -8px 0 white;
        transform: rotate(45deg);
    }
    .form-inner
        input[type="submit"] {
            background: #1762EE;
        }
    </style>
</style>
.floating-button-green,.floating-button-blue,.floating-button-red

<form class="transparent" method="post" action="">
    <div class="form-inner" class="container" style="margin-top: 8%">
        <hr>
        <label for="email"><b>Логін</b></label>
        <input type="text" placeholder="Enter Login"
name="user" value="<?=$_POST['user'] ?>">

        <label for="psw"><b>Пароль</b></label>
        <input type="password" placeholder="Enter Password"
name="password">

        <input type="submit" value="Увійти">
    </div>
</form>

<form class="transparent" method="post" action="">
    <div class="form-inner" class="container" style="margin-top: 8%">
        <hr>
        <label for="email"><b>Логін</b></label>
        <input id="username" type="text" placeholder="Enter
Login" name="user" value="<?=$_POST['user'] ?>">

        <label for="psw"><b>Пароль</b></label>
        <input id="password" type="password" placeholder="Enter
Password" name="password">

        <label for="psw-repeat"><b>Повторіть пароль</b></label>
        <input id="password2" type="password" placeholder="Repeat
Password" name="password2">
        <hr>

        <label for="login"><b>Ім'я</b></label>
        <input id="username2" type="text" placeholder="Введіть ім'я"
name="nickname" value="<?=$_POST['nickname'] ?>">

        <input type="submit" value="Зареєструватися">
    </div>
</form>

```