МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: **«РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ЗБЕРІГАННЯ ЗОБРАЖЕНЬ МІСЦЕВОСТІ З ВИКОРИСТАННЯМ YII 2»**

Виконав: студент 4 курсу, групи 6.1211-2пі
спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)
освітньої програми програмна інженерія
(назва освітньої програми)
Т.С. Кочаров
(ініціали та прізвище)
Керівник доцент кафедри програмної інженерії, PhD Чопорова О.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Рецензент доцент кафедри фундаментальної та прикладної математики, доцент, к.ф.-м.н. Панасенко Є.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти бакалавр

Спеціальність 121 інженерія програмного забезпечення
(шифр і назва)

Освітня програма програмна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри програмної інженерії, к.ф.-м.н., доцент

_____ Лісняк А.О.
(підпис)

“ ____ ” _____ 2024 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Кочарову Тіграну Сергійовичу
(прізвище, ім’я та по-батькові)

1. Тема роботи Розробка інформаційної системи для зберігання зображень місцевості з використанням YII 2

керівник роботи Чопорова Оксана Володимирівна, PhD
(прізвище, ім’я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 20 » грудня 2024 року № 2001-с

2. Строк подання студентом роботи 02.06.2025 р.

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Дослідження і аналіз предметної області.
2. Проєктування структури інформаційної системи.
3. Реалізація застосунку.

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)
презентація за темою доповіді

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 26.12.2024 р.

К А Л Е Н Д А Р Н И Й П Л А Н

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	08.01.2025	
2.	Збір вихідних даних.	20.01.2025	
3.	Обробка методичних та теоретичних	14.02.2025	
	джерел.		
4.	Розробка першого та другого розділу.	31.03.2025	
5.	Розробка третього розділу.	19.05.2025	
6.	Оформлення та нормоконтроль	26.05.2025	
	кваліфікаційної роботи бакалавра.		
7.	Захист кваліфікаційної роботи.	18.06.2025	

Студент _____ Т.С. Кочаров
(підпис) (ініціали та прізвище)

Керівник роботи _____ О.В. Чопорова
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ А.В. Столярова
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка інформаційної системи для зберігання зображень місцевості з використанням YII 2»: 47 с., 31 рис., 9 джерел.

БАЗИ ДАНИХ, ЗОБРАЖЕННЯ, ЗОБРАЖЕННЯ МІСЦЕВОСТІ, ІНФОРМАЦІЙНА СИСТЕМА, FRAMEWORK, GII, PHP, YII 2.

Об’єктом дослідження кваліфікаційної роботи є використання мови програмування PHP та фреймворку Yii 2 для розробки програмного забезпечення та інформаційної сторінки.

Предметом дослідження кваліфікаційної роботи є можливості та інструменти для розробки інформаційної системи для зберігання зображень місцевості.

Мета кваліфікаційної роботи – розробити інформаційну систему для зберігання зображень місцевості, з використанням фреймворку Yii 2.

Методи дослідження – аналіз існуючих аналогічних рішень, розробка із застосуванням об’єктно-орієнтованого підходу, створення програмного забезпечення та його тестування.

Кваліфікаційна робота присвячена розробці інформаційної системи для збереження зображень місцевості. В її межах було розроблено діаграми та схеми програмного продукту, а також написано код, що реалізує основну функціональність. Отримані результати можуть слугувати основою для подальшого розвитку та використання при створенні масштабніших інформаційних систем для роботи із зображеннями місцевості. Це дозволяє застосувати проєкт у більш складних геоінформаційних рішеннях та розширити його функціонал.

SUMMARY

Bachelor’s qualifying paper “Development of an Information System for Storing Terrain Images Using YII 2”: 47 pages, 31 figures, 9 references.

DATABASES, FRAMEWORK, GII, IMAGE, INFORMATION SYSTEM, MAP, PHP, YII 2.

The object of the research is the use of the PHP programming language and the Yii 2 framework for software development and the creation of an informational webpage.

The subject of the research is the capabilities and tools for developing an information system for storing terrain images.

The aim the research is to develop an information system for storing terrain images using the Yii 2 framework.

The research methods include the analysis of existing similar solutions, development using an object-oriented approach, software creation, and testing.

This qualification thesis is dedicated to the development of an information system for storing terrain images. As part of the work, diagrams and system design models were created, along with the code that implements the core functionality of the software product. The obtained results can serve as a foundation for further development and use in the creation of larger-scale information systems for working with terrain imagery. This allows the project to be applied in more complex geoinformation solutions and to expand its functionality.

ЗМІСТ

Завдання на кваліфікаційну роботу……..2
Реферат 4
Summary 5
Вступ..... 7
1 Дослідження і аналіз предметної області 9
1.1 Загальний опис дослідження 9
1.2 Аналіз середовища розробки 12
1.3 Вибір СУБД для реалізації..... 13
1.4 Структура фреймворку..... 15
1.5 Реалізація MVC в Yii 17
1.6 Реалізація додатку з використанням Yii..... 20
1.7 Висновки до розділу 1 22
2 Проєктування структури інформаційної системи 23
2.1 Схематичне зображення роботи застосунку 23
2.2 Побудова функціональної схеми..... 26
2.3 Розробка бази даних..... 28
2.4 Висновки до розділу 2 30
3 Реалізація застосунку..... 31
3.1 Розробка та створення бази даних..... 31
3.2 Робота з фреймворком Yii..... 32
3.3 Налаштування та встановлення Gii..... 33
3.4 Підключення бази даних 34
3.5 Створення авторизації 38
3.6 JavaScript та CSS..... 39
3.7 Розробка інтерфейсу 41
3.8 Висновки до розділу 3 45
Висновки 46
Перелік посилань..... 47

ВСТУП

Інформаційні системи пов’язані з різними аспектами життя кожної людини. Їх застосовували вже тоді, коли комп’ютерів ще не було. Розвиток цих систем і перетворення їх в перспективні об’єкти проєктування та дослідження залежить від ефективності технічних комп’ютерних засобів і програмного забезпечення.

Сучасні технології геоінформаційних систем (ГІС-технології) відкривають широкі можливості щодо автоматизації обробки результатів досліджень територіальних об’єктів.

Інформаційна графіка або – це графічне візуальне подання інформації, даних або знань, призначених для швидкого та чіткого відображення комплексної інформації [1]. Вона може покращити сприйняття інформації, використовуючи графічні матеріали для того, щоб підвищити можливості зорової системи людини бачити моделі і тенденції [3]. Процес створення інфографіки можна розглядати як візуалізацію даних, створення інформаційних схем та моделей подання інформації [2].

Інфографіка поширювалась вже багато років, а теперішнє швидке зростання числа легких у використанні та безкоштовних інструментів для їх створення зробили інфографіку доступною для ширшого кола населення. Соціальні мережі також дали поштовх для розповсюдження персональної інфографіки серед людей у всіх куточках світу.

У друкованих виданнях інфографіку часто застосовують для демонстрації прогнозу погоди, а також для створення карт, схем вебсайтів і статистичних діаграм. Сучасні картографічні матеріали, зокрема маршрутні схеми транспортних систем, використовують принципи інфографіки для поєднання різних типів інформації: транспортних мереж, точок пересадки та локальних орієнтирів.

Громадські транспортні карти є яскравими прикладами інфографіки, як-от схеми метро Вашингтона чи Лондонської підземки. У місцях великого

скупчення людей, наприклад на пересадкових станціях, часто застосовують інтегровані навігаційні системи зі стандартизованими символами та стилізованими картами для зручності пасажирів.

1 ДОСЛІДЖЕННЯ І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальний опис дослідження

У кваліфікаційній роботі необхідно розробити інформаційну систему для збереження зображень місцевості, використовуючи фреймворк Yii 2. Вхідні дані включатимуть координати місцевості та масштаб карти, а вихідними стануть запис у базі даних і графічний файл із зображенням місцевості, доступний для завантаження.

Для розробки інформаційної системи буде використовуватися мова програмування PHP, проте для підвищення безпеки, якості коду та продуктивності буде застосований відповідний фреймворк. Останнім часом PHP-фреймворки набули значної популярності та стали основою для створення вебзастосунків. Їх використання дозволяє суттєво скоротити час розробки, зменшити навантаження на процес створення програмного забезпечення, уникнути проблеми повторюваного коду та забезпечити швидку розробку якісних програмних продуктів. Крім того, застосування PHP-фреймворків робить процес створення програм більш простим і функціональним.

Найпопулярніші PHP-фреймворки, які широко використовуються у веброзробці:

- Laravel – один із найпопулярніших фреймворків, який забезпечує простоту розробки складних вебдодатків завдяки зручному синтаксису та багатому набору функцій;
- Symfony – потужний фреймворк, який використовується для створення масштабованих вебдодатків та API (він має модульну структуру та підтримує повторне використання компонентів);
- CodeIgniter – легкий і швидкий фреймворк, який не потребує складної конфігурації та підходить для створення невеликих проєктів;

- Yii – високопродуктивний фреймворк, який добре підходить для розробки великих вебдодатків (він має вбудовані механізми безпеки та підтримує генерацію коду);
- Zend Framework (Laminas) – потужний фреймворк для корпоративних рішень, який забезпечує високу продуктивність та безпеку.

Ці фреймворки допомагають розробникам створювати якісні вебдодатки, спрощуючи процес розробки та підвищуючи продуктивність.

Zend framework – це PHP-фреймворк, що створений і підтримується компанією Zend, співробітники якої є безпосередніми авторами мови PHP [7].

Можливості:

- всі компоненти орієнтовані на PHP 5 та E_STRICT – сумісні;
- вбудований генератор коду;
- архітектура «використовуй тільки те, що необхідно» з мінімальними залежностями компонентів;
- використовує розширюваний шаблон проєктування MVC, підтримує макети і PHP-скрипти подання за замовчуванням;
- підтримує безліч різних баз даних, включаючи MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite, and Informix Dynamic Server;
- спеціальні класи для створення, відправлення, отримання email з допомогою mbox, Maildir, POP3 та IMAP4;
- гнучка система кешування з підтримкою безлічі сховищ.

Недоліки:

- повільний без кешування;
- повільна швидкість розвитку;
- відсутній ORM.

Symfony – PHP фреймворк, який має велику бібліотеку класів, написаний на PHP 5. Архітектура має корисні компоненти та інструменти, призначені для створення складних вебдодатків. Symfony – це вільний каркас,

написаний на PHP 5, який використовує патерн Model-View-Controller (MVC). Symfony пропонує швидку розробку і керування вебдодатками, що дозволяє легко вирішувати рутинні завдання вебпрограміста [8].

Yii – це високопродуктивний компонентний PHP-фреймворк, призначений для швидкої розробки сучасних вебдодатків. Завдяки його компонентній структурі і відмінній підтримці кешування, фреймворк особливо підходить для розробки таких великих проєктів як портали, форуми, системи керування вмістом (CMS), інтернет-магазини або RESTful-додатки.

На даний момент існує дві версії Yii: 1.1 та 2.0. Версія 1.1 є попереднім поколінням і знаходиться у стані підтримки.

Версія 2.0 – це поточне покоління фреймворку, що використовує останні технології і протоколи, такі як Composer, PSR, простори імен, трейти і багато іншого. На цій версії будуть зосереджені основні зусилля кваліфікаційної роботи.

Для роботи Yii 2.0 необхідно мати PHP версії 5.4.0 та вище. Після встановлення Yii ми отримуємо як фреймворк, так і шаблон проєкту. Шаблон проєкту – це робочий проєкт Yii, в якому реалізовано деякий базовий функціонал, такий як система входу/виходу користувачів, форма зворотнього зв’язку [2]. Його код організовано в рекомендований спосіб.

Основні можливості і переваги Yii:

– інтерфейси DAO і ActiveRecord для роботи з базами даних (PDO);
– підтримка інтернаціоналізації;
– перехоплення і обробка помилок;
– україномовна документація;
– парадигма модель-вид-контролер;
– генерація базового PHP-коду для CRUD-операцій (скаффолдинг);
– підтримка тем оформлення для їх легкої зміни;
– використання AJAX і інтеграція з jQuery;
– низький поріг входження;
– автоматичне тестування.

На основі проведеного аналізу фреймворк Yii демонструє значну кількість можливостей і переваг, а також не має істотних недоліків. Саме тому він є найбільш відповідним вибором для реалізації інформаційної системи збереження зображень місцевості, порівняно з іншими популярними PHP-фреймворками.

1.2 Аналіз середовища розробки

Перед початком розробки інформаційної системи було проведено аналіз найбільш поширених середовищ розробки програмного забезпечення. Наведемо короткі відомості.

NetBeans – безкоштовне інтегроване середовище розробки (IDE), що підтримує широкий спектр мов програмування, включаючи PHP, Java, JavaScript, HTML5 та інші. Його можна встановити як для окремих мов, так і у повній конфігурації. Завдяки розширенням середовище можна адаптувати до потреб розробника.

JetBrains PhpStorm – платний редактор з розширеними можливостями аналізу коду, рефакторингу та автоматизованими засобами запобігання помилкам. Підтримує стандарти оформлення коду та інтеграцію з інструментами контролю якості, що робить його потужним рішенням для розробки сучасних PHP-додатків.

Qt Creator – середовище розробки для створення кросплатформових додатків з використанням бібліотеки Qt. Підтримує C++, QML та JavaScript, а також пропонує широкий набір інструментів для розробки системних програм та додатків.

З огляду на проведений аналіз, найбільш відповідним вибором для реалізації поставлених завдань є NetBeans, оскільки він оптимізований для розробки вебдодатків, є безкоштовним, а також забезпечує зручність та високу продуктивність у роботі.

1.3 Вибір СУБД для реалізації

База даних – це сукупність взаємозв’язаних даних, що зберігаються разом та відповідають наступним властивостям:

– дані логічно пов’язані між собою і несуть відповідну інформацію;
– структура баз даних звичайно відповідає тому специфічному набору даних, які вона містить;
– бази даних відображають тільки окремі аспекти реального світу, що дає змогу визначити їх як «мікросвіт».

Система управління базами даних (СУБД) поєднує відомості з різних джерел в одній реляційній базі даних. Створювані форми, запити і звіти дозволяють швидко й ефективно обновляти дані, отримувати відповіді на питання, здійснювати пошук потрібних даних, аналізувати дані, друкувати звіти, діаграми і поштові наклейки.

Розробка структури БД передбачає повний опис всіх атрибутів сутностей (полів таблиць). Кожна таблиця – масив з однорідних елементів, які прийнято називати записами. Запис – неподільна одиниця інформації в БД, але можна сформувати такий запит, щоб отримати якусь частину цієї інформації. Запис може містити в собі одну або кілька іменованих полів, що задаються при створенні таблиці. Кожне з полів має певний тип (наприклад, ціле число, текст та інші). У всі таблиці можна додавати записи, видаляти їх, робити пошук за таблицями і виводити потрібну інформацію. Всі ці дії виконуються за допомогою SQL (Structured Query Language) запитів SELECT, DELETE, CREATE, INSERT.

На сьогоднішній день існує досить багато систем управління базами даних. Було розглянуто деякі з них.

Oracle Database або Oracle RDBMS. Об’єктно-реляційна система управління базами даних компанії Oracle Corporation. Ця СУБД забезпечує ефективне, надійне і безпечне управління даними таких критично важливих для бізнесу додатків, як онлайнові середовища, виконує масштабну обробку

транзакцій (OLTP), сховища даних з високою інтенсивністю потоку запитів, а також ресурсномісткі інтернет-додатки. Редакція Oracle Database Enterprise Edition надає інструментальні засоби і функції, що забезпечують відповідність вимогам сучасних корпоративних додатків в області доступності та масштабованості. Ця редакція містить всі компоненти Oracle Database, а також допускає розширення за допомогою придбання додаткових модулів та програм.

Система Oracle Database дозволяє звертатися до даних з будь-якого додатку, розробленого із застосуванням технологій Microsoft. NET, Visual Studio та вебдодатків. Основною умовою є лише наявність справних бібліотек, що дають змогу підключатися до серверу бази даних [9].

Oracle. Oracle Database є комерційною СУБД, але є її безкоштовна версія, яку можна без проблем скачати прямо з офіційного сайту компанії.

Microsoft SQL Server. Система Microsoft SQL Server відштовхується від концепції платформи даних Майкрософт: вона спрощує управління будь-якими даними в будь-якому місці і в будь-який момент часу. Вона дозволяє зберігати в базах даних інформацію, отриману з структурованих, напівструктурованих і неструктурованих джерел, таких як зображення та музика. У SQL Server 2008 є великий набір інтегрованих служб, які розширюють можливості використання даних: можна складати запити, виконувати пошук, проводити синхронізацію, робити звіти, аналізувати дані. Всі дані зберігаються на основних серверах, що входять до складу центру обробки даних. До них здійснюється доступ з настільних комп’ютерів і мобільних пристроїв. Таким чином, можна повністю контролювати дані незалежно від того, де вони збережені.

Система SQL Server дозволяє звертатися до даних з будь-якого додатку, розробленого із застосуванням технологій Microsoft. NET та Visual Studio, а також в межах сервісно-орієнтованої архітектури і бізнес-процесів – через Microsoft BizTalk Server. SQL Server дозволяє створити надійну, продуктивну, інтелектуальну платформу, що відповідає всім вимогам по роботі з даними. SQL Server є комерційною СУБД.

PostgreSQL. Безкоштовна об’єктно-реляційна система управління базами даних. Система PostgreSQL заснована на ядрі, створеному безліччю розробників. У подібних випадках розумно зосередитися на оснащенні системи новими можливостями, але не займатися оптимальним їх втіленням, оскільки у випадку виникнення необхідності завжди можна буде повернутися до оптимізації відповідних ділянок коду.

MySQL. Безкоштовна система управління базами даних. MySQL є власністю компанії Oracle Corporation, що отримала її разом з поглиненою Sun Microsystems, що здійснює розробку і підтримку програми. Розповсюджується під GNU General Public License або під власною комерційною ліцензією. Крім цього, розробники створюють функціональність за замовленням ліцензійних користувачів.

MySQL є рішенням для малих і середніх додатків. Входить до складу серверів WAMP, LAMP і в портативні збірки серверів Denver, XAMPP. Зазвичай MySQL використовується як сервер, до якого звертаються локальні або віддалені клієнти, проте в дистрибутив входить бібліотека внутрішнього сервера, що дозволяє включати MySQL в автономні програми [4].

Оскільки інтерфейс системи буде вебдодатком, потрібно обрати СУБД, з якою можуть співпрацювати мови програмування для написання веб додатків. Вибір було зупинено на MySQL.

Висока продуктивність MySQL на вузькому колі задач – прямий наслідок її функціональної простоти.

1.4 Структура фреймворку

Yii реалізує архітектурний шаблон Model-View-Controller (MVC), який широко використовується у веброзробці. Ця модель дозволяє розділити бізнес-логіку та користувацький інтерфейс, що дає розробникам змогу модифікувати окремі частини програми без впливу на інші.

Модель (Model) відповідає за управління даними та бізнес-логіку програми.

Представлення (View) займається формуванням інтерфейсу користувача, наприклад, текстових елементів та форм введення.

Контролер (Controller) забезпечує зв’язок між моделлю і представленням, керуючи обробкою запитів.

Крім того, Yii використовує фронт-контролер, який виступає в ролі головного додатку (`application`), інкапсулюючи контекст обробки запиту. Він збирає інформацію про запит і передає її відповідному контролеру для подальшої обробки.

Ця архітектурна концепція робить розробку гнучкішою та простішою, особливо при масштабуванні вебдодатків.

Життєвий цикл додатку в середовищі Yii при обробці користувальницького запиту складається з таких ключових етапів:

- попередня ініціалізація (CApplication::preinit()) – виконується базова підготовка додатку до роботи;
- ініціалізація обробника помилок – налаштовується механізм обробки можливих помилок;
- реєстрація компонентів ядра – підключаються основні системні компоненти;
- завантаження конфігурації – додаток отримує параметри з конфігураційного файлу;
- ініціалізація додатку (CApplication::init()) – завершується основне налаштування додатку;
- реєстрація поведінок – додаються додаткові функціональні можливості;
- завантаження статичних компонентів – підключаються необхідні статичні елементи;
- виклик події `onBeginRequest` – запускається механізм початкової обробки запиту;

– обробка запиту – здійснюється отримання та інтерпретація даних користувача;
– збір інформації про запит – аналізуються параметри запиту для подальшої обробки;
– створення контролера – визначається відповідний контролер для обробки запиту;
– запуск контролера – активується контролер для виконання необхідного завдання;
– виклик події onEndRequest – завершується обробка запиту, і формується відповідь.

Ця структурована послідовність забезпечує ефективне управління користувацькими запитами та гарантує стабільну роботу вебдодатку.

1.5 Реалізація MVC в Yii

Yii містить набір ядрових компонентів, які забезпечують необхідні функціональні можливості для більшості вебдодатків. Наприклад, компонент `CHttpRequest` відповідає за збір інформації про користувацький запит, надаючи дані про URL, cookies та параметри запиту. Встановлюючи властивості компонентів, можна гнучко змінювати поведінку Yii відповідно до потреб розробки.

Ключові компоненти, які використовує клас CWebApplication:

– CAssetManager – керує публікацією та збереженням файлів ресурсів (asset files);
– CAuthManager – реалізує систему контролю доступу на основі ролей (RBAC);
– CCache – забезпечує кешування даних для підвищення продуктивності додатку;
– CClientScript – управляє клієнтськими скриптами (JavaScript та CSS);

- CPhpMessageSource – надає можливості локалізації та перекладу системних повідомлень Yii;
- CDbConnection – управляє підключенням до бази даних;
- CErrorHandler – обробляє неперехоплені помилки та винятки PHP;
- CFormatter – форматує дані для правильного відображення;
- CHttpRequest – надає інформацію про параметри користувацького запиту;
- CSecurityManager – забезпечує безпеку, включаючи шифрування та хешування;
- CHttpSession – управляє користувацькими сесіями;
- CStatePersister – дозволяє зберігати глобальний стан додатку;
- CUrlManager – виконує парсинг і формування URL;
- CWebUser – надає ідентифікаційну інформацію поточного користувача;
- CThemeManager – керує оформленням та темами додатку.

Завдяки цим компонентам Yii пропонує гнучкість, безпеку та продуктивність, що робить його одним із найпотужніших фреймворків для веброзробки.

Контролер (controller) – це екземпляр класу CController або успадкованого від нього класу. Він створюється об’єктом додатку тоді, коли користувач робить відповідний запит. Під час запуску контролер виконує відповідну дію, що зазвичай передбачає створення відповідних моделей і рендеринг необхідних представлень. У найпростішому випадку дія – це метод класу контролера, назва якого починається з action.

У контролера є дія за замовчуванням, яка виконується у випадку, коли користувач не вказує дію при запиті. За замовчуванням ця дія називається index. Змінити її можна шляхом встановлення значення CController::defaultAction.

Модель (model) – це екземпляр класу CModel або класу, успадкованого від нього. Модель використовується для зберігання даних або застосованих до них бізнес-правил.

Модель становить собою окремий об’єкт даних. Це може бути запис таблиці бази даних або HTML-форма з полями для вводу даних. Кожне поле об’єкту даних представляється атрибутом моделі, кожний атрибут має мітку та може бути перевірений на коректність за допомогою набору правил.

Yii надає два типи моделей: модель форми та Active Record. Обидва типи є розширенням базового класу CModel.

Модель форми – це екземпляр класу CFormModel. Вона використовується для зберігання даних, введених користувачем. Як правило, ми отримуємо ці дані, обробляємо, а потім позбавляємося від них. Наприклад, на сторінці авторизації модель такого типу може бути використана для представлення інформації про ім’я користувача та пароль. Детальний опис роботи з формами подано у розділі «Робота з формами».

Active Record (AR) – це шаблон проєктування, який використовується для абстрагування доступу до бази даних у об’єктно-орієнтованій формі. Кожен об’єкт AR являє собою екземпляр класу CActiveRecord або класу, успадкованого від нього, та представляє окремий рядок у таблиці бази даних. Поля цього рядку відповідають властивостям AR-об’єкту.

Представлення – це PHP-скрипт, який складається переважно з елементів користувальницького інтерфейсу. Він може охоплювати вирази PHP, проте рекомендується, щоб ці вирази не змінювали дані і залишалися відносно простими. Згідно з концепцією розділення логіки та представлення, більша частина коду логіки повинна бути розміщена у контролері або моделі, а не у скрипті представлення.

Представлення має ім’я, яке використовується з метою ідентифікації файлу скрипта представлення у процесі рендерингу. Ім’я представлення повинно співпадати із назвою файлу представлення. Наприклад, для представлення edit відповідний файл скрипта повинен називатися edit.php. Щоб відобразити представлення необхідно викликати метод CController::render(), вказавши ім’я представлення, при цьому метод спробує виявити відповідний файл у директорії protected/views/ControllerID.

Всередині скрипта представлення екземпляр контролера доступний через $this. Таким чином, ми можемо звернутися до властивостей контролера із коду представлення: $this->propertyName.

1.6 Реалізація додатку з використанням Yii

Загальний процес створення вебдодатків за допомогою Yii включає такі основні етапи:

- створення структури директорій – утиліта Yii допомагає автоматизувати цей процес, створюючи необхідні папки та базові файли;
- конфігурація додатку – редагування конфігураційного файлу програми, що може вимагати створення спеціальних компонентів, наприклад, керування користувачами;
- створення класу моделі – кожен тип даних має відповідну модель (для швидкої генерації моделей Active Record можна використовувати інструмент Gii);
- розробка класу контролера – кожен тип запиту обробляється відповідним контролером (взаємодія між моделлю та представленням забезпечує гнучкість; Gii також допомагає автоматизувати цей процес);
- створення дій та представлень (views) – основний етап, на якому формується функціональність та користувацький інтерфейс додатку;
- конфігурація фільтрів – налаштування механізмів обробки запитів у контролерах;
- створення тем оформлення – якщо потрібна кастомізація дизайну;
- переклад повідомлень – важливий етап при локалізації додатку;
- оптимізація кешування – аналіз даних та представлень, що можуть бути закешовані для підвищення продуктивності;

– налаштування продуктивності та розгортання – фінальне тестування додатку перед запуском у продакшн.

Цей процес забезпечує структурованість, гнучкість та ефективність при розробці вебдодатків.

Процес обробки користувальницького запиту в Yii проходить через кілька ключових етапів, що забезпечують коректну взаємодію кожного компонента з системою:

– запит користувача – відправляється через URL, після чого вебсервер обробляє його та запускає скрипт ініціалізації (index.php);
– створення екземпляра додатку – скрипт index.php створює об’єкт додатку та розпочинає його виконання;
– отримання параметрів запиту – компонент request збирає всю інформацію, зокрема URL, cookies та інші параметри запиту;
– визначення контролера та дії – компоненти urlManager та router визначають, який контролер та метод відповідають за обробку запиту (наприклад, якщо запит пов’язаний з постами, контролером буде PostController, а дією – show);
– створення контролера – система ініціалізує відповідний контролер (PostController) для обробки запиту;
– застосування фільтрів – перед виконанням дії застосовуються різні фільтри (контроль доступу, логування, кешування);
– отримання даних – дія (actionShow) виконує запит до бази даних, отримуючи об’єкт моделі `Post` з ID = 1;
– формування представлення (view) – викликається уявлення show, яке отримує модель Post для подальшого відображення;
– рендеринг сторінки – представлення обробляє атрибути моделі та формує візуальний інтерфейс;
– підключення віджетів – в процесі формування сторінки можуть підключатися додаткові інтерактивні елементи;
– вставка уявлення в макет – остаточне представлення інтегрується в макет сторінки (layout);

– завершення обробки запиту – дія завершує рендеринг сторінки та виводить отриманий результат користувачеві.

Ця послідовність забезпечує чітку організацію запитів та ефективну взаємодію між моделлю, контролером та представленням.

1.7 Висновки до розділу 1

У першому розділі кваліфікаційної роботи проведено аналіз основних засобів та методів створення інформаційної системи для збереження зображень місцевості. У результаті для реалізації було обрано такі технології: мову програмування PHP, фреймворк Yii2, скриптову мову JavaScript, мову розмітки HTML5 та каскадні таблиці стилів CSS3.

Оптимальним середовищем розробки визначено NetBeans, оскільки воно найбільш пристосоване для створення вебдодатків, забезпечує зручність та високу швидкість роботи.

Також у цьому розділі розглянуто фреймворк Yii, наведено його основні теоретичні аспекти, сформульовано задачу та визначено перелік функцій, які необхідно реалізувати у програмному продукті. Завдяки широким можливостям, Yii суттєво спрощує процес розробки, роблячи його ефективним, швидким і зручним для розробників.

2 ПРОЄКТУВАННЯ СТРУКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

На основі аналізу поставленої задачі та актуальності розробки сформовано перелік ключових вимог до програмного продукту.

Система повинна включати:

– вибір місцевості на карті та можливість її деталізації;
– збереження координат і масштабу у базі даних для подальшого використання;
– завантаження графічного файлу із зображенням обраної місцевості;
– механізм авторизації: система реєстрації користувачів та вхід до системи.

Розробка буде здійснена на PHP із застосуванням фреймворку Yii 2, що забезпечить ефективну обробку запитів та структуровану взаємодію з базою даних. Для запуску додатку буде достатньо сучасного веббраузера (Google Chrome, Opera, Mozilla Firefox, Edge). Вхідними даними слугуватимуть координати місцевості та масштаб мапи, а вихідними – запис у базі даних і графічний файл із зображенням території. Ця структура дозволить створити гнучку, ефективну та масштабовану інформаційну систему.

2.1 Схематичне зображення роботи застосунку

На рисунку 2.1 зображується алгоритм роботи інформаційної системи для зберігання зображень місцевості.

Алгоритм збереження місцевості у базу даних включає кілька ключових етапів:

– введення вхідних даних – користувач вводить координати місцевості та масштаб карти, супроводжувані відповідними повідомленнями;
– перевірка коректності введених даних – система аналізує отриману інформацію, перевіряючи її на відповідність заданим параметрам;

– обробка даних – координати та масштаб передаються в систему для подальшого використання;
– створення графічного файлу – система генерує або отримує відповідне зображення місцевості на основі введених даних;
– додавання даних у базу – координати та масштаб записуються у базу даних для подальшого використання;
– збереження зображення – графічний файл, що відображає місцевість, зберігається та стає доступним для завантаження;
– відображення результату – користувач отримує підтвердження успішного збереження даних та доступ до зображення.

Ця послідовність забезпечує логічну та ефективну обробку даних і гарантує правильне збереження інформації у базі.

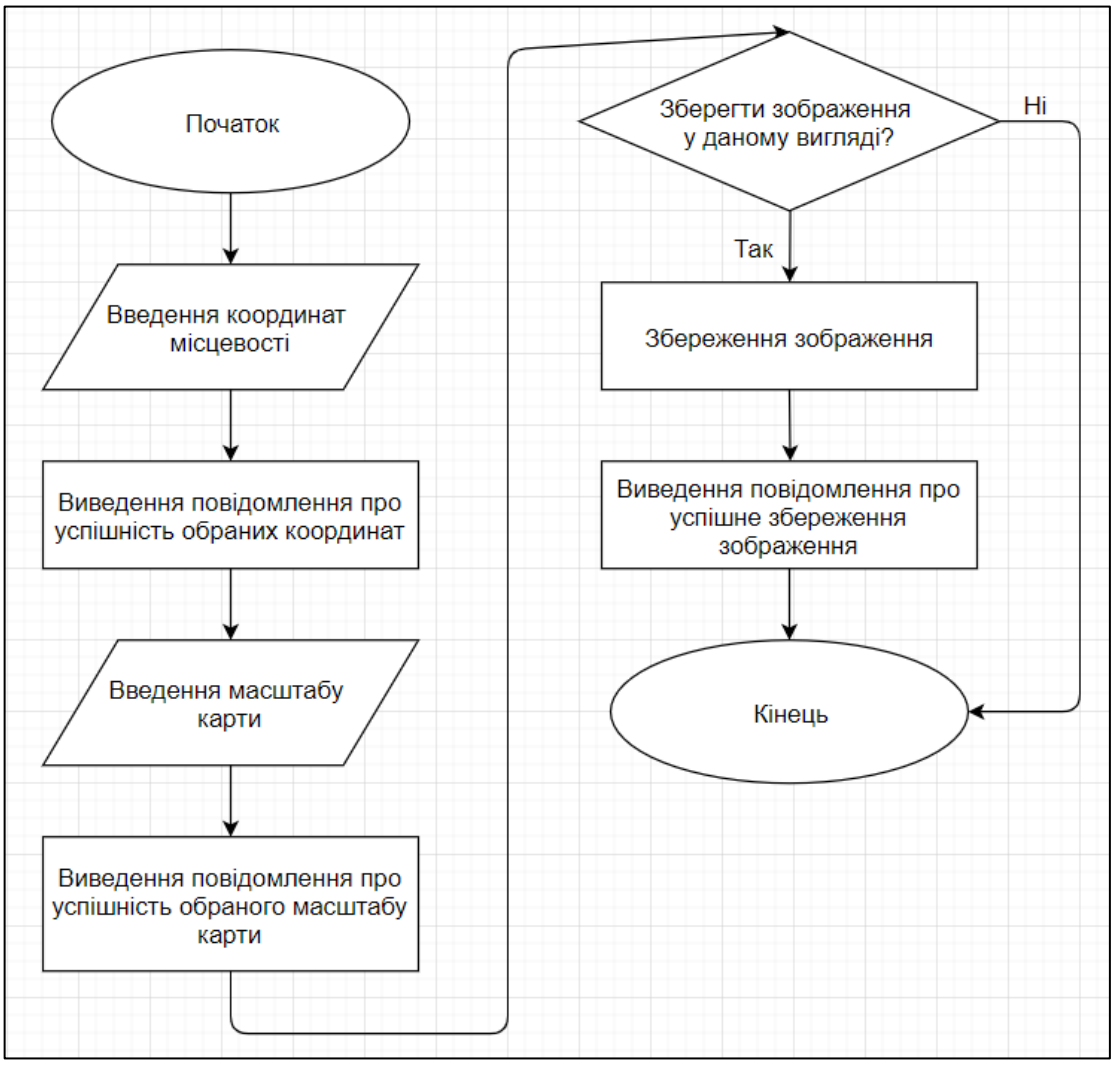


Рисунок 2.1 – Алгоритм роботи

На рисунку 2.2 зображена діаграма станів, яка дозволяє зобразити всю роботу інформаційної системи для зберігання зображень місцевості.

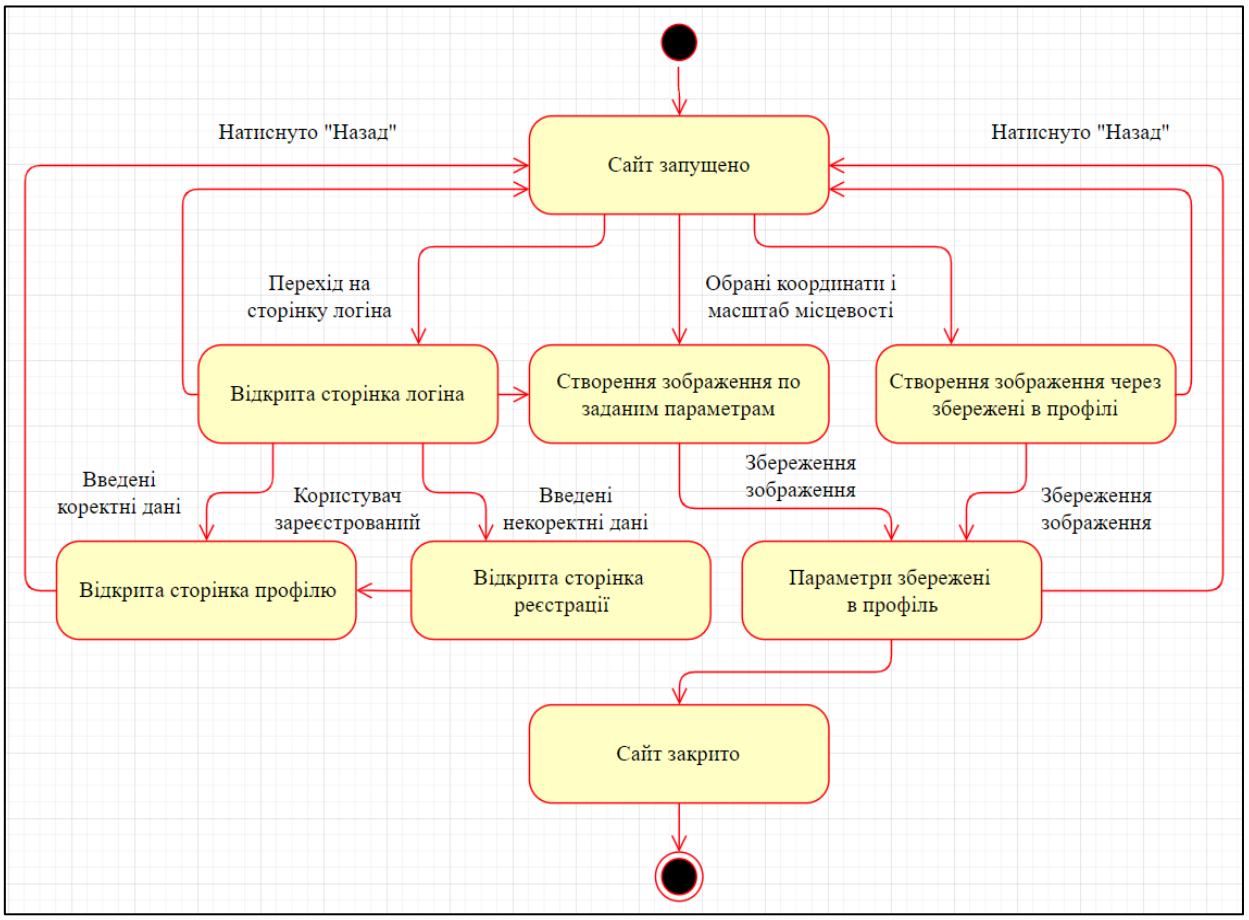


Рисунок 2.2 – Діаграма станів

Діаграма ілюструє вісім ключових станів у процесі взаємодії з інформаційною системою.

– запуск сайту – користувач відкриває вебдодаток;
– перехід на сторінку логіну – рекомендується авторизуватись або зареєструватись для отримання доступу до функціоналу;
– авторизація або реєстрація – користувач вводить дані для входу або проходить реєстрацію, створюючи новий акаунт;
– перехід до роботи із системою – після успішної авторизації відкривається можливість роботи з основними функціями;
– вибір місцевості для збереження – користувач може самостійно вибрати потрібну територію на карті;

– використання історії пошуку – можливість вибору місцевості з раніше здійснених запитів;
– збереження місцевості – координати та масштаб фіксуються у базі даних, а зображення території формується для завантаження;
– підтвердження дії – користувач отримує повідомлення про успішне виконання операції.

Цей логічний процес забезпечує зручність і структурованість при роботі з системою.

2.2 Побудова функціональної схеми

На рисунку 2.3 зображено функціональну схему, що містить систему зберігання зображення місцевості та підсистеми для роботи з картами, графічним файлом та авторизацією.

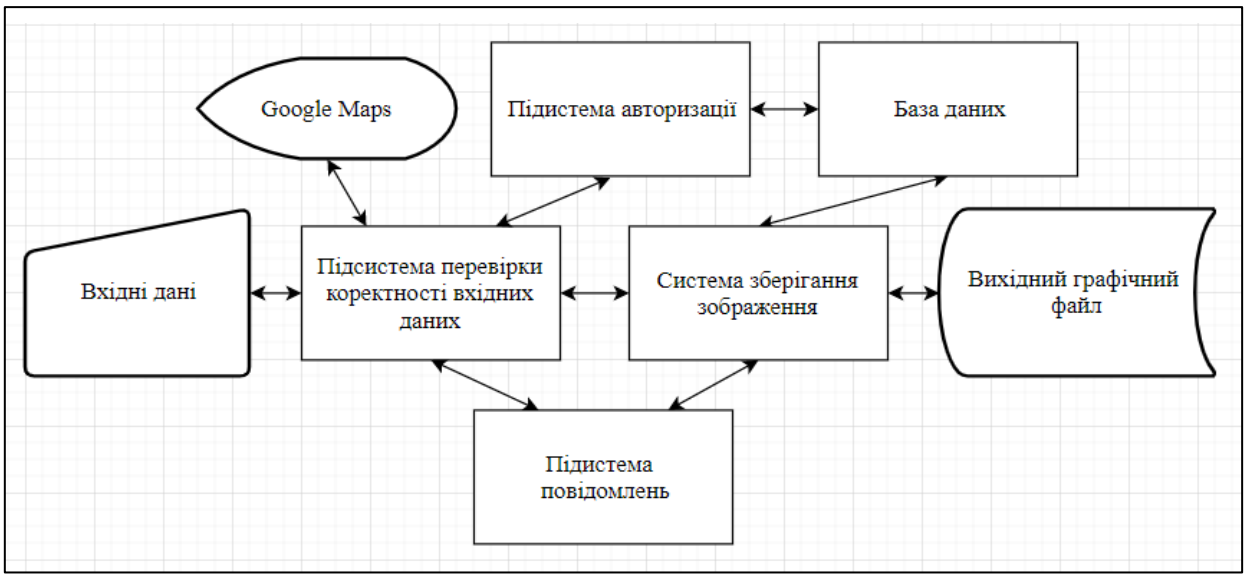


Рисунок 2.3 – Функціональна схема

Підсистеми авторизації та зберігання зображень мають тісний зв’язок із базою даних, оскільки вони відповідають за ідентифікацію користувачів та збереження графічних матеріалів.

Підсистема авторизації забезпечує розпізнавання користувачів та контроль доступу.

Підсистема збереження зображень відповідає за збереження вибраної місцевості, включаючи координати та масштаб.

Підсистема повідомлень сповіщає користувачів про всі важливі події та зміни у системі.

Вхідними даними слугують координати та масштаб місцевості, які інтегруються із Google Maps для точного визначення області.

На виході система, окрім запису в базу даних, при необхідності генерує графічний файл, що містить зображення обраної території.

На діаграмі прецедентів (рис. 2.4) зображено взаємодію користувача з інформаційною системою для збереження зображень місцевості.

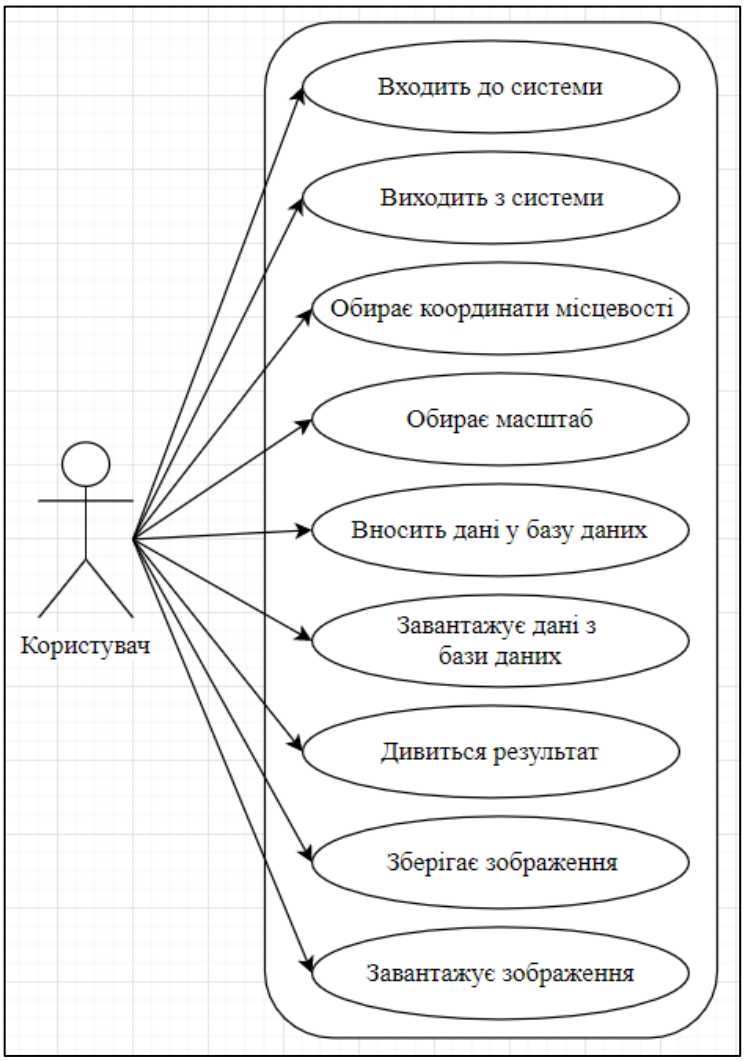


Рисунок 2.4 – Діаграма прецедентів

Користувач може обирати координати та масштаб місцевості для збереження. Входити та виходити із системи, забезпечуючи авторизацію. Вносити та завантажувати дані до бази даних. Перевіряти правильність виконаних дій у системі. Завантажувати графічні файли із зображенням місцевості на свій пристрій. У системі немає адміністратора або інших акторів, окрім користувача, оскільки вона вже зібрана, і внесення змін можливе лише програмістом.

Ця модель взаємодії забезпечує простоту використання та логічну структуру системи.

2.3 Розробка бази даних

На діаграмі бази даних (рис. 2.5) відображено зв’язок між таблицями users і locations, що організований за принципом один до багатьох.

Це означає, що один користувач може зберігати декілька локацій у системі. Кожна локація пов’язана з одним конкретним користувачем через відповідний ідентифікатор. Така структура забезпечує гнучке управління збереженими місцевостями та дозволяє легко отримувати список всіх точок, прив’язаних до конкретного акаунту.

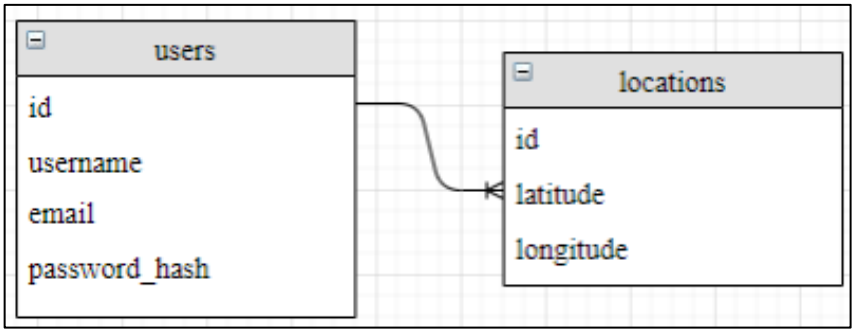


Рисунок 2.5 – Схема бази даних

На рисунку 2.6 зображено діаграму класів, що містить у собі класи User та Location.

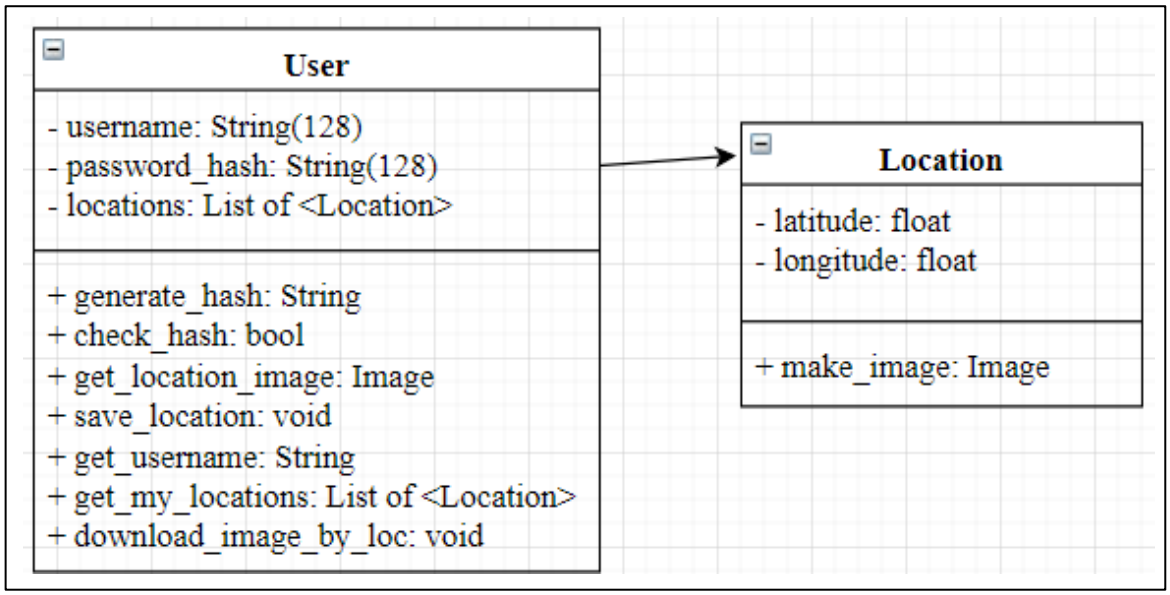


Рисунок 2.6 – Діаграма класів

Процес взаємодії користувача з вебдодатком відбувається наступним чином:

– передача запиту – користувач вводить дані у вебформу та надсилає AJAX POST-запит на сервер;
– обробка запиту – сервер приймає запит через обробник /getimage, передає дані в метод класу Location;
– отримання зображення – на основі координат та масштабу обробник генерує зображення місцевості;
– відображення результату – після створення зображення обробник on_success AJAX-запиту відображає картинку та оновлює внутрішній вміст без перезавантаження сторінки;
– збереження даних – якщо користувач залогінений, його координати та масштаб зберігаються у базі даних;
– повторний доступ – користувач може переглянути список запитів, вибрати потрібну локацію та повторно отримати її без введення даних вручну.

Цей механізм забезпечує швидку та зручну взаємодію з інформаційною системою, дозволяючи користувачам миттєво отримувати необхідні зображення місцевості.

2.4 Висновки до розділу 2

У другому розділі кваліфікаційної роботи було виконано ключові етапи проєктування інформаційної системи. Сформовано технічне завдання, що визначає цілі та обмеження розробки. Визначено вхідні та вихідні дані, необхідні для функціонування системи. Створено макет шаблону інтерфейсу, що забезпечує зручність взаємодії користувача із системою. Складено основні вимоги, що охоплюють функціонал та особливості програмного продукту. Сформовано алгоритм роботи для коректного виконання завдань у системі. Розроблено діаграми та схеми, які структуровано описують архітектуру програмного продукту. Ці етапи дозволяють створити логічну та ефективну систему, що відповідає поставленим завданням.

3 РЕАЛІЗАЦІЯ ЗАСТОСУНКУ

Для оптимізації розробки проєкту та зручності були використані такі інструменти:

- Open Server – серверна платформа, яка забезпечує локальне розгортання вебдодатків та їх тестування;
- PhpMyAdmin – графічний інтерфейс для керування MySQL, що значно спрощує роботу з базами даних;
- Sublime Text – потужний редактор програмного коду, що забезпечує швидке та ефективне написання скриптів.

Кожен із цих інструментів має унікальні можливості, які допомагають розробникам прискорити процес розробки, тестування та налагодження вебдодатку.

3.1 Розробка та створення бази даних

PhpMyAdmin – це вебдодаток з відкритим кодом, створений на PHP, що забезпечує інтуїтивно зрозумілий інтерфейс для адміністрування СУБД MySQL.

Завдяки вебінтерфейсу, користувач може керувати сервером MySQL без необхідності вводити SQL-команди вручну, запускати SQL-запити, виконувати зміни у базі даних, переглядати структуру та вміст таблиць, що значно спрощує адміністрування.

Популярність PhpMyAdmin серед веброзробників пояснюється зручністю, швидкістю роботи та гнучкістю у керуванні базою даних.

Цей інструмент буде використовуватися у створеному вебсайті для ефективного управління базою даних, що робить його незамінним у розробці.

3.2 Робота з фреймворком Yii

Встановити Yii можна двома способами:

– через менеджер пакунків Composer – цей метод є бажанішим, оскільки дозволяє легко встановлювати нові розширення та оновлювати Yii однією командою;
– за допомогою архіву – завантаження архіву з офіційного сайту, розпакування та налаштування конфігурації [2].

Встановлення через Composer має наступні етапи:

– завантажити та встановити Composer-Setup.exe;
– переконатись, що використовується остання версія Composer (composer self-update);
– виконати команду для встановлення Yii у директорію basic;
– встановити composer-asset-plugin, щоб керувати залежностями Bower та NPM.

Встановлення через архів має такі етапи:

– завантажити архів з офіційного сайту Yii;
– розпакувати архів у вебдоступну директорію;
– редагувати конфігураційний файл (config/web.php), зокрема встановити cookieValidationKey.

Після успішного встановлення необхідно налаштувати вебсервер або скористатися вбудованим сервером PHP, виконавши команду php yii serve. Ця команда запускає додаток без необхідності налаштовувати Apache чи Nginx.

Також Yii надає два шаблони проєкту:

– базовий шаблон – містить мінімальний набір функцій для швидкого старту;
– розширений шаблон – більше підходить для командної розробки складних додатків.

3.3 Налаштування та встановлення Gii

Встановлювати та налаштовувати розширення Gii відбувається за допомогою Composer, для цього виконується команда, як зображено на рисунку 3.1.

```
php composer.phar require --dev --prefer-dist yiisoft/yii2-gii
//або додати
"yiisoft/yii2-gii": "~2.0.0"
//до секції require-dev файлу composer.json.
```

Рисунок 3.1 – Встановлення Gii за допомогою Composer

Коли розширення Gii встановлено, його необхідно підключити до файлу конфігурації інформаційної системи (рис. 3.2).

```
return [
    'bootstrap' => ['gii'],
    'modules' => [
        'gii' => 'yii\gii\Module',
        // ...
    ],
```

Рисунок 3.2 – Підключення Gii

Доступ до Gii можна буде отримати за допомогою localhost. В базовому шаблоні структура конфігурації інакша, тому Gii необхідно сконфігурувати у config/web.php, як це зображено на рисунку 3.3.

```
// ...
if (YII_ENV_DEV) {
    // налаштування конфігурації для середовища розробки
    $config['bootstrap'][] = 'debug';
    $config['modules']['debug'] = 'yii\debug\Module';
    $config['bootstrap'][] = 'gii';
    $config['modules']['gii'] = 'yii\gii\Module';
}
```

Рисунок 3.3 – Конфігурування Gii

Таким чином, щоб налаштувати IP-адреси, необхідно зробити як показано нижче на рисунку 3.4.

```
if (YII_ENV_DEV) {
    $config['bootstrap'][] = 'debug';
    $config['modules']['debug'] = 'yii\debug\Module';
    $config['bootstrap'][] = 'gii';
    $config['modules']['gii'] = [
        'class' => 'yii\gii\Module',
        'allowedIPs' => ['127.0.0.1', '::1', '192.168.0.*', '192.168.178.20'],
    ];
}
```

Рисунок 3.4 – Налаштування IP-адрес для доступу

3.4 Підключення бази даних

Для початку потрібно налаштувати Yii для роботи з базою даних (рис. 3.5).

```
<?php
return array(
'connectionString'=>'mysql:host=localhost;dbname=mehodical,
'emulatePrepare' => true,
'username' => 'root',
'password' => '',
'charset' => 'utf8', ); ?>
```

Рисунок 3.5 – Налаштування з’єднання з базою даних

Фреймворк Yii містить вбудований генератор коду Gii, який значно спрощує процес створення основних компонентів вебдодатку. За допомогою Gii можна генерувати моделі на основі таблиць у базі даних та створювати CRUD-контролери, що реалізують ключові операції: Create (додавання запису), Read (перегляд запису), Update (редагування запису), Delete (видалення запису).

Для активації Gii у Yii необхідно внести зміни до конфігураційного файлу (`config/web.php`), додавши опис підключення модуля Gii (рис. 3.6).

Gii автоматизує рутинні завдання, прискорюючи розробку та забезпечуючи структурований код, що робить його незамінним для створення вебдодатків.

```
'Gii' => array (
'Class' => 'system.gii.GiiModule',
'Password' => 'generate',
'IpFilters' => array ( '127.0.0.1', ':: 1'),
),
```

Рисунок 3.6 – Активація кодогенератора Gii

Для налаштування з’єднання з базою даних у Yii необхідно виконати кілька ключових кроків:

- створити базу даних – Yii підтримує SQLite, MySQL, PostgreSQL, MSSQL, Oracle, але для інформаційної системи збереження зображень місцевості використовується MySQL;
- створити клас Active Record – він відповідатиме за взаємодію з таблицями бази даних;
- визначити дію (Action) – необхідно налаштувати контролер, що буде обробляти запити до БД;
- створити представлення (View) – воно забезпечить візуальне відображення отриманих даних;
- відредагувати конфігурацію БД – у файлі config/db.php потрібно встановити відповідні параметри.

Після цього Yii зможе автоматично взаємодіяти з базою даних (рис. 3.7).

Файл config/db.php є стандартним засобом конфігурації на основі файлів, який містить параметри, необхідні для створення та ініціалізації об’єкта yii\db\Connection. Через цей об’єкт здійснюється взаємодія з базою даних та виконання SQL-запитів.

```
return
[
 'class' => 'yii\db\Connection',
 'dsn' => 'mysql:host=localhost;dbname=supportmad_claim',
 'username' => 'root',
 'password' => '',
 'charset' => 'utf8'];
```

Рисунок 3.7 – Бібліотека PDO для роботи з базою даних

Конфігурація config/db.php інтегрується у головний файл config/web.php, де визначаються правила ініціалізації додатку.

Для забезпечення адаптивного дизайну необхідно завантажити та встановити розширення Yiistrap2, яке розширює можливості оформлення інтерфейсу (рис. 3.8). Далі необхідно налаштувати систему доступу на основі ролей (рис. 3.9).

```
'import'=>array(
'bootstrap.helpers.*',
'bootstrap.behaviors.*',
'bootstrap.components.*',
'bootstrap.form.*',
'bootstrap.widgets.*',
),
'components'=>array(
'bootstrap' => array(
'class' => 'bootstrap.components.TbApi',
),
```

Рисунок 3.8 – Встановлення доповнення Yiiastrap (main.php)

```
'components'=>array(
'user' => array(
class' => 'WebUser',
),
'authManager' => array(
'class' => 'PhpAuthManager',
'defaultRoles' => array('guest'),
),
),
```

Рисунок 3.9 – Налаштування системи доступа на основі ролей (main.php)

Після цього необхідно створити контролер, налаштувати правила URL для відображення та увімкнути підтримку Json [6] (рис. 3.10).

```
namespace app\controllers;
use yii\rest\ActiveController;
class UserController extends ActiveController
{
    public $modelClass = 'app\models\User';
}
```

Рисунок 3.10 – Створення контролеру

Клас контролера успадковується від yii\rest\ActiveController. Якщо задати modelClass як app\models\User, тим самим вказавши контролера, до якої моделі йому необхідно звертатися для редагування або вибірки даних (рис. 3.11).

```
'urlManager' => [
    'enablePrettyUrl' => true,
    'enableStrictParsing' => true,
    'showScriptName' => false,
    'rules' => [
        ['class' => 'yii\rest\UrlRule', 'controller' => 'user'],
    ],
]
```

Рисунок 3.11 – Налаштування URL

Налаштування вище додають правило для контролера user, яке надає доступ до даних користувача через красиві URL і логічні дієслова HTTP.

Для того щоб API міг приймати дані в форматі JSON, треба налаштувати властивість parsers у компонента request application component на використання yii\web\JsonParser JSON даних на вході (рис. 3.12).

```
'request' => [
    'parsers' => [
        'application/json' => 'yii\web\JsonParser',
    ]
]
```

Рисунок 3.12 – Включення JSON на прийом даних

3.5 Створення авторизації

Компонент user відповідає за управління статусом аутентифікації користувача та вимагає вказаний identity class, що містить логіку аутентифікації. У конфігурації інформаційної системи identity class для user визначено як app\models\User (рис. 3.13).

```
return [
    'components' => [
        'user' => [
            'identityClass' => 'app\models\User',
        ],
    ],];
```

Рисунок 3.13 – Використання класу user

Фільтри контролю доступу (ACF) – це простий спосіб керування доступом, який найкраще підходить для додатків із базовими вимогами до контролю. Як випливає з їхньої назви, ACF – це фільтри, які можна приєднувати до контролера або модуля як поведінку [5]. Вони перевіряють набір правил доступу, щоб гарантувати, що користувач має необхідні права для виконання запитаної дії (рис. 3.14).

У базовому класі yii\web\User його використовують як компонент додатка.

Щоб отримати identity поточного користувача, можна скористатися виразом Yii::$app->user->identity. Він поверне екземпляр identity class, який представляє поточного аутентифікованого користувача, або null, якщо користувач не аутентифікований (наприклад, гість).

Код з рисунку 3.15 демонструє, як отримати іншу інформацію, пов’язану з аутентифікацією, з yii\web\User.

Метод yii\web\User::login() встановлює identity поточного користувача в yii\web\User. Якщо сесії увімкнені, identity буде зберігатися в сесії, що забезпечить збереження статусу аутентифікації протягом усього часу її дії.

```
{
    public function behaviors()
    {
        return [
            'access' => [
                'class' => AccessControl::className(),
                'only' => ['login', 'logout', 'signup'],
                'rules' => [
                    [
                        'allow' => true,
                        'actions' => ['login', 'signup'],
                        'roles' => ['?'],
                    ],
                    [
                        'allow' => true,
                        'actions' => ['logout'],
                        'roles' => ['@'],
                    ],
                ],
            ],
        ];}}
```

Рисунок 3.14 – Використання ACF-фільтру

```
$identity = Yii::$app->user->identity;
$id = Yii::$app->user->id;
$isGuest = Yii::$app->user->isGuest;
$identity = User::findOne(['username' => $username]);
Yii::$app->user->login($identity);
```

Рисунок 3.15 – Використання компоненту user

Якщо увімкнено вхід на основі cookie (так званий «запам’ятай мене»), то identity також буде збережена в cookie, що дозволяє відновити статус аутентифікації користувача протягом всього часу життя cookie.

3.6 JavaScript та CSS

Вбудовані скрипти відіграють важливу роль у налаштуванні динамічно генерованого коду та невеликих багаторазово використовуваних фрагментів інтерфейсу, які входять до складу віджетів.

Щоб додати такі скрипти, можна скористатися методом registerJs(), який дозволяє реєструвати JavaScript-код безпосередньо у представленнях Yii. Це забезпечує зручний спосіб інтеграції сценаріїв у систему без необхідності ручного підключення окремих файлів (рис. 3.16).

```
$this->registerJs(
    "$('#myButton').on('click', function() { alert('Button clicked!'); });",
    View::POS_READY,
    'my-button-handler'
);
```

Рисунок 3.16 – Реєстрація вбудованого скрипту

Зовнішній скрипт може бути доданий так, як продемонстровано на рисунку 3.17. Зареєструвати CSS-файл можна, як показано на рисунку 3.18.

```
$this->registerJsFile(
    '@web/js/main.js',
    ['depends' => [\yii\web\JqueryAsset::className()]]
);
```

Рисунок 3.17 – Реєстрація зовнішнього скрипту

```
$this->registerCssFile("@web/css/themes/black-and-white.css", [
    'depends' => [\yii\bootstrap\BootstrapAsset::className()],
    'media' => 'print',
], 'css-print-theme');
```

Рисунок 3.18 – Реєстрація CSS-файлу

Обробник помилок (errorHandler) реєструється як компонент додатка. Для його використання застосовується відповідна команда, яка дозволяє перехоплювати та обробляти виняткові ситуації, забезпечуючи коректне реагування на помилки в системі.

У Yii errorHandler автоматично обробляє помилки та винятки, перенаправляючи їх на визначену сторінку або логуючи інформацію для

подальшого аналізу. Це дозволяє покращити стабільність та зручність взаємодії з додатком.

Якщо потрібно налаштувати поведінку обробника, це можна зробити через конфігураційний файл, задаючи відповідні параметри для errorHandler (рис. 3.19).

```
return [
    'components' => [
        'errorHandler' => [
            'maxSourceLines' => 20,
        ],
    ], ];
```

Рисунок 3.19 – Використання обробника помилок

3.7 Розробка інтерфейсу

На рисунку 3.20 представлена головна сторінка інформаційної системи для збереження зображень місцевості.

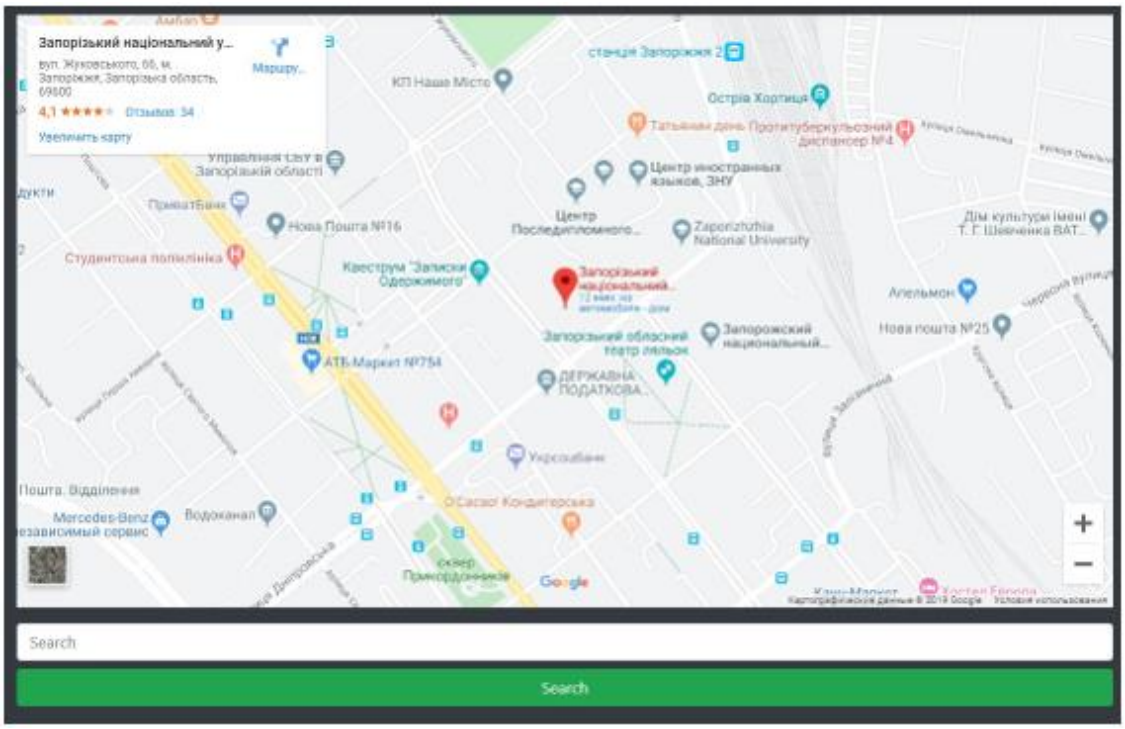


Рисунок 3.20 – Головна сторінка інформаційної системи

Сторінка містить інтегровану карту Google Maps, на якій можна вибрати потрібну локацію для додавання до бази даних. Вибрані координати та масштаб автоматично зберігаються в буфері перед внесенням у базу.

На рисунку 3.21 зображений пошук по сайту інформаційної системи. На рисунку 3.22 показано, як будуть виглядати вже збережені в базу даних зображення місцевості.



Рисунок 3.21 – Пошук по сайту

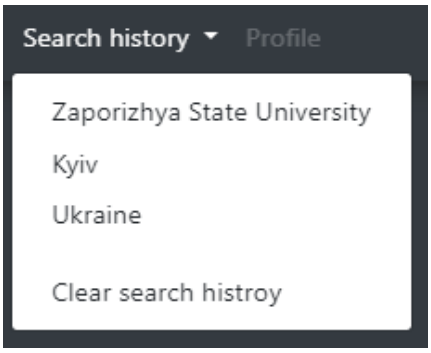


Рисунок 3.22 – Історія пошуку

На рисунку 3.23 зображено меню сайту інформаційної системи, яке включає перехід на головну сторінку, доступ до профілю користувача та історію пошуку. Остання містить раніше збережені зображення місцевості, що були вибрані та додані до бази даних.

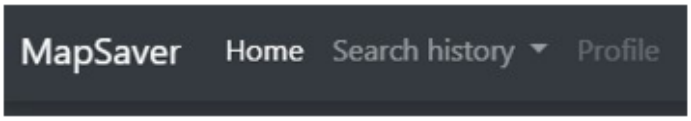


Рисунок 3.23 – Меню сайту

Оскільки система для збереження зображень працює з базами даних, у ній передбачено механізми реєстрації та авторизації користувачів. Процес реєстрації та авторизації в інформаційній системі ілюструється на

рисунках 3.24 та 3.25. Він забезпечує доступ користувачів до функціоналу платформи, дозволяючи зберігати та переглядати зображення місцевості відповідно до їхніх прав доступу.

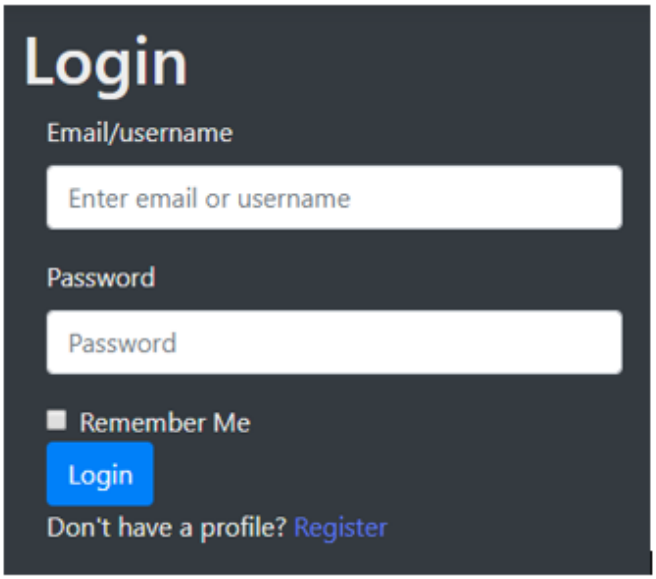


Рисунок 3.24 – Вхід на сайт

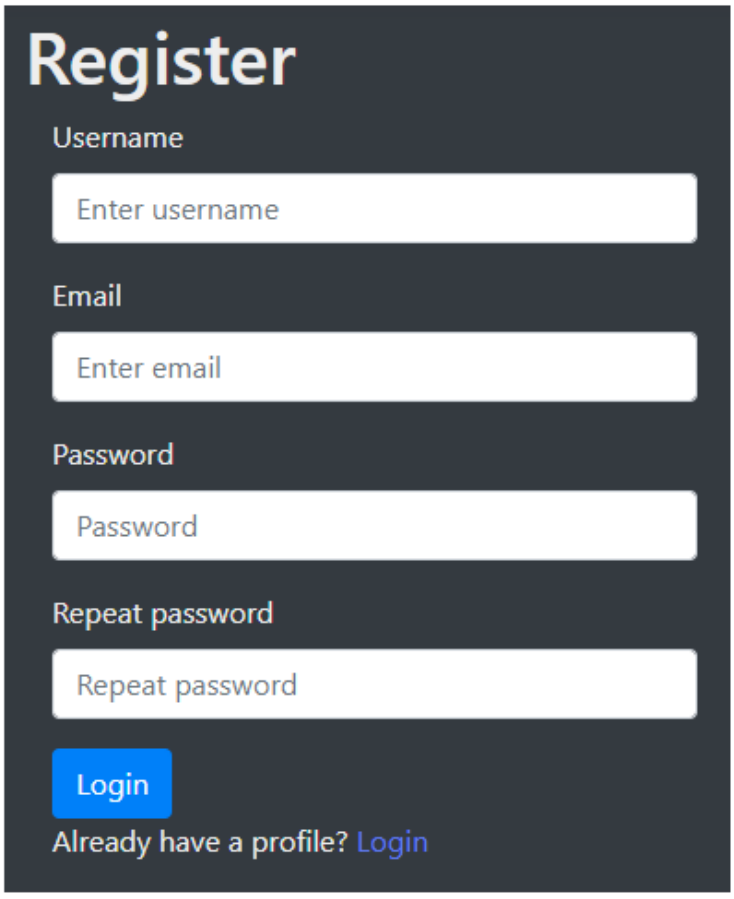


Рисунок 3.25 – Реєстрація на сайті

На етапі розробки інформаційної системи одним із ключових аспектів є перевірка її працездатності.

Під час тестування перевіряється відповідність програмного продукту вимогам технічного завдання та його технічним характеристикам. Система має крос-платформну сумісність, що забезпечує її коректне відображення на пристроях із різною роздільною здатністю екрана. Випробування проводилося на планшеті, телефоні, ноутбуці та широкоформатному екрані, і в усіх випадках система функціонувала без помилок.

Крім того, система є кросбраузерною, що гарантує її правильну роботу у всіх популярних веббраузерах. Тестування проводилося в Google Chrome, Mozilla Firefox, Opera, Safari, Internet Explorer 11 та Edge. Виявлених відмінностей у відображенні та роботі не було – функціонал залишався стабільним, а всі прогнозовані результати відповідали отриманим. Жодних зависань чи труднощів у використанні не виявлено, а логічна структура інтерфейсу залишилася незмінною.

Оцінка швидкості завантаження головної сторінки залежно від швидкості інтернет-з’єднання:

– 100 Мб/с – 1,23 с;
– 20 Мб/с – 3,55 с;
– 10 Мб/с – 8,78 с;
– 1 Мб/с – 11,76 с.

Отримані результати підтверджують швидку роботу системи.

Серед гіперпосилань не виявлено неактивних чи неправильних – усі переходи між сторінками здійснюються швидко та коректно. Статичні та динамічні зображення відображаються правильно, фон автоматично адаптується до розмірів екрана. Всі заголовки сторінок відображаються коректно, іконка сайту присутня на кожній сторінці. Зображення місцевості завантажуються плавно, без втрати кольорів та ефектів руху.

Також не було знайдено грубих синтаксичних чи орфографічних помилок. Текст легко читати й сприймати.

Зважаючи на результати тестування, можна зробити висновок, що інформаційна система повністю готова до використання – вона працює швидко, стабільно та якісно.

3.8 Висновки до розділу 3

У третьому розділі кваліфікаційної роботи було розглянуто реалізацію інформаційної системи для збереження зображень місцевості та її інтерфейс.

Для розробки вебсайту використовувався фреймворк Yii 2, що забезпечує всі ключові особливості проєкту, побудованого за його допомогою. Архітектура сайту заснована на трирівневій моделі Model–View–Controller (MVC), а всі дані зберігаються у базі даних.

Для покращення інтерфейсу було використано доповнення YiiStrap, яке є аналогом фреймворку Bootstrap для Front-End розробки.

Система пройшла тестування на різних пристроях з різними операційними системами, що підтвердило її повну готовність до використання без критичних помилок.

Також була створена інструкція користувача, яка детально описує етапи роботи з інформаційною системою, забезпечуючи легкість освоєння її функціоналу.

ВИСНОВКИ

У кваліфікаційній роботі було розроблено інформаційну систему для збереження зображень місцевості з використанням фреймворку Yii 2.

Для реалізації було обрано мову програмування PHP, а також Yii 2, JavaScript, HTML 5 та CSS 3.

Завдання кваліфікаційної роботи виконано в повному обсязі:

– розроблено та описано алгоритми основних модулів системи;
– здійснено програмну реалізацію та проведено тестування;
– підтверджено коректність роботи програмного продукту та забезпечено можливість його подальшої підтримки.

Перспективи розвитку – систему можна легко розширити, додавши нові функції, наприклад:

– інші види карт;
– розширений набір вхідних та вихідних даних;
– інтеграцію з іншими програмними продуктами для збереження зображень місцевості.

Основні результати кваліфікаційної роботи:

– виконано аналіз сучасного стану питання, що підтвердило доцільність розробки нового продукту;
– проведено дослідження існуючих рішень, визначено їхні переваги та недоліки, що підтвердило актуальність створення нової інформаційної системи;
– розроблено графічну та структурну схему програмного продукту;
– проведено тестування на різних пристроях та при різних роздільних здатностях, що підтвердило адаптивність та кросплатформеність системи.

Отримані результати свідчать про успішну розробку інформаційної системи, її стабільність, ефективність та перспективність у подальшому використанні та розвитку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Yii Framework Wiki : Yii for beginners 2. *YiiFramework*. URL: https://www.yiiframework.com/wiki/462/yii-for-beginners-2 (дата звернення: 15.03.2025).

2. Yii2 Practical-B App. *Krajee*. URL: https://demos.krajee.com/app-practical-b#google_vignette (дата звернення: 15.03.2025).

3. Васильєв О. Основи теорії систем і системного аналізу : навч. посіб. Київ : Видавництво «Ліра-К», 2022. 368 с. URL: https://lira-k.com.ua/files/contents/13029_1.pdf?srsltid=AfmBOoo8liWdX9Dg24aFwFo62wqYJq2oL6Pr1KnXeK1pV-8-pWy5lmMX (дата звернення: 16.03.2025).

4. Що таке фреймворк: пояснюємо простими словами. 2023. *BrainLab*. URL: https://brainlab.com.ua/uk/blog-uk/shho-take-frejmvork-poyasnyuyemo-prostymy-slovamy (дата звернення: 24.03.2025).

5. Популярні PHP фреймворки: як вибрати. 2024. *FoxmindEd IT School*. URL: https://foxminded.ua/php-freimvorky/ (дата звернення: 24.03.2025).

6. Що таке фреймворк: пояснюємо простими словами. 2022. *Wezom*. URL: https://wezom.com.ua/ua/blog/scho-take-freymvork-poyasnyujemo-prostimi-slovami (дата звернення: 26.04.2025).

7. Денисенко А. Фреймворки у веб-розробці – що це, які існують і для чого потрібні. 2022. *Highload*. URL: https://highload.tech/uk/frejmvorki-u-veb-rozrobtsi-shho-tse-yaki-isnuyut-i-dlya-chogo-potribni/ (дата звернення: 26.04.2025).

8. Фреймворк в програмуванні. *QALight*. URL: https://qalight.ua/baza-znaniy/frejmvork-v-programuvanni/ (дата звернення: 27.04.2025).

9. Oracle Database. *Wikipedia*. URL: https://en.wikipedia.org/wiki/Oracle_Database (дата звернення: 12.05.2025).