

МІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)
на тему:

«Каталог для малого бізнесу»

Виконав студент групи ВТ-22-2
МАКАРОВ Володимир Денисович

Керівник роботи:
ЄФРЕМОВ Юрій Миколайович

Рецензент:
ЧИЖМОТРЯ Олексій Володимирович

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

«ЗАТВЕРДЖУЮ»
Зав. кафедри інженерії
програмного забезпечення
Тетяна Вакалюк
«27» лютого 2026 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **МАКАРОВ Володимир Денисович**

Керівник роботи: **ЄФРЕМОВ Юрій Миколайович**

Тема роботи: **«Каталог малого бізнесу»,**

затверджена Наказом закладу вищої освіти від **«27» лютого 2026 р., №92/с**

Вихідні дані для роботи: Методи та засоби розробки веб-застосунку для збору, обробки та відображення інформації про компанії та послуги. Об'єктом дослідження є веб-застосунок каталогу, що забезпечує отримання, збереження та обробку даних про компанії, їх категорії, рейтинги та відгуки, а також їх представлення користувачам у зручній та інформативній формі.

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Єфремов Ю.М.	01.03.2026р	16.05.2026р
2	Єфремов Ю.М.	01.03.2026р	16.05.2026р
3	Єфремов Ю.М.	01.03.2026р	16.05.2026р

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з веб-застосунку та пояснювальної записки. Пояснювальна записка містить 72 сторінок, 17 ілюстрації, 22 таблиць, список використаних джерел та додатки.

Мета роботи – розробка та реалізація веб-застосунку каталогу для малого бізнесу, що забезпечує цифрове представництво компаній, автоматизацію процесу розміщення та управління оголошеннями, реалізацію механізму пошуку і фільтрації, а також систему відгуків і рейтингового оцінювання.

У роботі використано методи веб-розробки, проєктування реляційних баз даних, алгоритмізації, об'єктно-орієнтованого програмування, застосування архітектурного шаблону Model–View–Controller, а також принципи UX/UI дизайну та багаторівневого розмежування доступу (адміністратор, менеджер, користувач).

Отримані результати включають розроблену архітектуру веб-застосунку, спроектовану реляційну базу даних із забезпеченням цілісності та індексації, реалізацію модулів реєстрації та авторизації користувачів, управління оголошеннями та категоріями, механізму пошуку та фільтрації, а також системи відгуків і рейтингу з автоматичним обчисленням середнього значення оцінок. Реалізовано механізми інформаційної безпеки, зокрема хешування паролів, захист від SQL-ін'єкцій та контроль доступу відповідно до ролі користувача.

Рекомендовано впровадження розробленого веб-застосунку як платформи підтримки малого бізнесу для підвищення його видимості в мережі Інтернет, оптимізації взаємодії з клієнтами та покращення якості надання послуг.

КЛЮЧОВІ СЛОВА: ВЕБ-ЗАСТОСУНОК, КАТАЛОГ, MVC, БАЗА ДАНИХ, РЕЙТИНГ, ВІДГУКИ, ПОШУК, АВТОРИЗАЦІЯ, АДАПТИВНИЙ ІНТЕРФЕЙС

					ІПЗ.КР.Б – 121-26-ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Каталог для малого бізнесу	Літ.	Арк.	Аркуші
Розроб.		Макаров В. Д.						
Перевір.		Єфремов Ю. М.					3	69
Реценз.						ФІКТ, гр.ВТ-22-2		
Н. Контр.								
Зав.каф.		Вакалюк Т.А.						

ABSTRACT

The bachelor's final qualification work consists of a web application and an explanatory note. The explanatory note contains 72 pages, 17 illustrations, 22 tables, a list of sources used and an appendix.

The purpose of the work is to develop and implement a catalog web application for small businesses that provides digital representation of companies, automates the process of placing and managing ads, implements a search and filtering mechanism, as well as a feedback and rating system.

The work uses methods of web development, relational database design, algorithmization, object-oriented programming, the use of the Model–View–Controller architectural pattern, as well as the principles of UX/UI design and multi-level access delimitation (administrator, manager, user).

The results obtained include the developed architecture of the web application, the designed relational database with integrity and indexing, the implementation of user registration and authorization modules, ad and category management, search and filtering mechanism, as well as a feedback and rating system with automatic calculation of the average value of ratings. Information security mechanisms have been implemented, in particular password hashing, protection against SQL injections and access control according to the user role.

It is recommended to implement the developed web application as a small business support platform to increase its visibility on the Internet, optimize customer interaction and improve the quality of service provision.

KEYWORDS: WEB APPLICATION, CATALOG, MVC, DATABASE, RATING, FEEDBACK, SEARCH, AUTHORIZATION, ADAPTIVE INTERFACE

					ПЗ.КР.Б – 121-26-ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У СФЕРІ КАТАЛОГІВ ДЛЯ МАЛОГО БІЗНЕСУ	9
1.1 Постановка задачі дослідження	9
1.2 Аналіз існуючих програмних рішень каталогів.....	14
1.3 Вибір архітектури веб-застосунку каталогу	18
Висновки до першого розділу	21
РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБ-ЗАСТОСУНКУ КАТАЛОГУ ДЛЯ МАЛОГО БІЗНЕСУ	23
2.1 Визначення варіантів використання та структури системи	23
2.2 Розробка бази даних	33
2.3 Проєктування алгоритмів роботи веб-застосунку	37
2.4 Реалізація функціоналу веб-застосунку	40
Висновки до другого розділу	45
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБ-ЗАСТОСУНКОМ КАТАЛОГУ	47
3.1 Користувацький інтерфейс	47
3.2 Інтерфейс адміністратора та менеджера	54
3.3 Особливості роботи з оголошеннями, відгуками та рейтингом	58
3.4 Тестування веб-застосунку	60
Висновки до третього розділу.....	66
ВИСНОВКИ.....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	70
ДОДАТКИ	73

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

MVC — Model–View–Controller, архітектурний шаблон проєктування програмного забезпечення

OOP — Object-Oriented Programming, об'єктно-орієнтоване програмування

PHP — Hypertext Preprocessor, серверна мова програмування

MySQL — система керування реляційними базами даних

SQL — Structured Query Language, мова структурованих запитів до бази даних

HTML — HyperText Markup Language, мова розмітки веб-сторінок

CSS — Cascading Style Sheets, каскадні таблиці стилів

JS — JavaScript, мова програмування для створення інтерактивних веб-інтерфейсів

Bootstrap — фреймворк для розробки адаптивного користувацького інтерфейсу

HTTP — HyperText Transfer Protocol, протокол передачі гіпертексту

HTTPS — HyperText Transfer Protocol Secure, захищена версія протоколу HTTP

CRUD — Create, Read, Update, Delete, базові операції роботи з даними

UI — User Interface, користувацький інтерфейс

UX — User Experience, досвід користувача

DBMS — Database Management System, система керування базами даних

WampServer — програмне середовище для локальної розробки веб-застосунків на основі Windows, Apache, MySQL та PHP

AJAX — Asynchronous JavaScript and XML, технологія асинхронного обміну даними

API — Application Programming Interface, програмний інтерфейс застосунку

					ІІЗ.КР.Б – 121-26-ІІЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми пов'язана з тим, що малий бізнес активно бере участь у розвитку місцевої економіки, надає населенню необхідні товари й послуги та створює робочі місця. Водночас багато невеликих компаній мають обмежені можливості для просування в Інтернеті, через що потенційним клієнтам складніше знаходити інформацію про їхню діяльність. Не всі підприємці можуть дозволити собі розробку окремого сайту або постійну рекламну підтримку, тому веб-каталог є практичним рішенням для розміщення основних відомостей про компанію. Такий сервіс дає змогу систематизувати інформацію про підприємства, забезпечити пошук за напрямом діяльності, регіоном і ключовими словами, а також швидко надати користувачу доступ до контактів, адреси, опису послуг і вебсайту компанії. Для малого бізнесу це створює додаткову можливість бути представленим в онлайн-середовищі, підвищити впізнаваність і полегшити взаємодію з потенційними клієнтами.

Метою випускної кваліфікаційної роботи є розробка веб-застосунку каталогу малого бізнесу, який забезпечує додавання, редагування, пошук, фільтрацію та перегляд інформації про компанії у структурованому вигляді.

Для досягнення поставленої мети необхідно виконати такі завдання:

1. Проаналізувати особливості роботи каталогів та подібних інформаційних сервісів.
2. Обґрунтувати вибір технологій для створення веб-застосунку.
3. Спроектувати структуру бази даних для зберігання інформації про користувачів, компанії, категорії, міста, регіони та відгуки.
4. Реалізувати механізми реєстрації, авторизації та розмежування ролей користувачів.
5. Розробити функціонал додавання, редагування й видалення інформації про компанії.
6. Реалізувати пошук і фільтрацію компаній за категоріями та регіонами.
7. Створити адміністративну панель для керування категоріями, містами, регіонами, користувачами та оголошеннями.

					ІПЗ.КР.Б – 121-26-ІПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

8. Забезпечити базові механізми захисту даних, зокрема перевірку форм, хешування паролів і захист від CSRF-атак.

9. Провести тестування розробленого веб-застосунку.

Об'єктом дослідження є процес розміщення, організації та пошуку інформації про представників малого бізнесу у веб-застосунку.

Предметом дослідження є методи та програмні засоби створення веб-застосунку каталогу, зокрема проєктування бази даних, реалізація серверної логіки, інтерфейсу користувача, пошуку, фільтрації, адміністрування та механізмів безпеки.

					ПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

РОЗДІЛ 1. АНАЛІЗ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У СФЕРІ КАТАЛОГІВ ДЛЯ МАЛОГО БІЗНЕСУ

1.1 Постановка задачі дослідження

Сучасний етап розвитку інформаційного суспільства характеризується активною цифровізацією бізнес процесів та переходом значної частини економічної діяльності в онлайн-середовище. В умовах глобальної інформатизації підприємства будь-якого масштабу змушені адаптуватися до нових реалій ринку, де цифрова присутність перетворилася з конкурентної переваги на обов'язкову умову виживання. Малий бізнес, як один із ключових сегментів національної економіки, потребує доступних та ефективних інструментів для представлення своїх послуг і продукції в мережі Інтернет. Саме малі підприємства формують основу підприємницького середовища, забезпечують зайнятість населення та сприяють розвитку локальних економік, а тому цифрова підтримка цього сегменту є стратегічно важливим завданням як для окремих підприємств, так і для суспільства в цілому.

В умовах високої конкуренції та обмежених ресурсів підприємствам необхідні рішення, які дозволяють мінімізувати витрати на маркетинг і водночас забезпечити стабільний потік клієнтів. Відсутність належного цифрового представництва ставить малий бізнес у невигідне становище порівняно з великими корпораціями, що мають власні відділи маркетингу та значні бюджети на просування. Ця нерівність породжує системну проблему, вирішення якої потребує розроблення спеціалізованих інформаційних інструментів, орієнтованих саме на потреби малого підприємництва. Одним із найбільш ефективних та поширених таких інструментів є бізнес-каталог — інформаційна система, що дозволяє структурувати відомості про компанії за категоріями, видами діяльності та географічною ознакою, створюючи єдиний доступний простір для взаємодії між підприємцями та споживачами послуг.

Аналіз існуючих рішень у сфері каталогів свідчить про те, що більшість наявних платформ не відповідає потребам локального малого бізнесу. Одні системи є надто складними в налаштуванні та використанні, інші — комерційно

					ІІЗ.КР.Б – 121-26-ІІЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

орієнтованими на великі компанії з відповідними бюджетами. Частина рішень пропонує обмежений функціонал без можливості масштабування, тоді як інші вимагають значних технічних знань для впровадження та підтримки. Зазначені недоліки зумовлюють необхідність розроблення нового рішення, яке поєднуватиме функціональну повноту, зручність використання та доступність для широкого кола підприємців. Таке рішення повинно враховувати специфіку функціонування малого бізнесу, бути простим у впровадженні та не вимагати від власників підприємств глибоких технічних компетенцій. Об'єктом дослідження є процес цифрового представництва суб'єктів малого підприємництва в мережі Інтернет засобами веб-орієнтованих інформаційних систем. Предметом дослідження є методи, підходи та засоби проектування і розроблення веб-застосунку каталогу, що забезпечує ефективну взаємодію між підприємцями та споживачами їхніх послуг. Метою дослідження є комплексне проектування та програмна реалізація функціонального, зручного й безпечного веб-застосунку каталогу для малого бізнесу, що відповідає сучасним вимогам до якості програмного забезпечення.

Постановка задачі дослідження передбачає визначення основних функціональних та нефункціональних вимог до системи, що розробляється. Комплексне визначення вимог є фундаментальним етапом проектування будь-якої інформаційної системи, оскільки саме на цьому етапі закладається архітектурна основа майбутнього рішення та визначається вектор його розвитку. Недостатньо опрацьовані вимоги на початковому етапі, як правило, призводять до суттєвих переробок на пізніших стадіях розробки, що значно збільшує витрати часу та ресурсів.

Серед функціональних вимог до системи ключовими є: можливість реєстрації та авторизації користувачів із підтримкою кількох ролей; створення, редагування та видалення оголошень про бізнес; управління ієрархією категорій; здійснення пошуку та багатокритеріальної фільтрації; реалізація механізму залишення відгуків і рейтингової оцінки компаній. Крім того, система повинна передбачати можливість завантаження медіаматеріалів (фотографій, логотипів),

					ІІЗ.КР.Б – 121-26-ІІЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

управління профілями компаній, відстеження статистики переглядів оголошень та забезпечення системи сповіщень для зареєстрованих користувачів. Кожна з перелічених функцій безпосередньо впливає на цінність продукту для кінцевого користувача та визначає рівень його практичної корисності.

Нефункціональні вимоги охоплюють питання безпеки, продуктивності, масштабованості та зручності користування. До них належать: вимоги до надійності системи та часу відгуку сервера; забезпечення доступності платформи на рівні не нижче 99%; відповідність стандартам веб-доступності; коректне відображення інтерфейсу на пристроях різних типів і розмірів екрана. Нefункціональні вимоги, хоча й не описують конкретні функції системи, визначають якісні характеристики її роботи та суттєво впливають на задоволеність користувачів і загальне сприйняття продукту.

Важливим аспектом постановки задачі є визначення моделі розмежування прав доступу в системі. Ефективна модель управління доступом є невід'ємною складовою будь-якої багатокористувацької інформаційної системи, оскільки вона визначає межі дозволених дій для кожної категорії учасників та забезпечує цілісність і захищеність даних. Для ефективного управління контентом необхідно реалізувати багаторівневу рольову модель, що включає щонайменше три рівні: адміністратора, менеджера та звичайного користувача. Адміністратор здійснює загальне управління системою, включаючи налаштування категорій, модерацію контенту, управління обліковими записами та моніторинг активності платформи. Менеджер відповідає за оперативне адміністрування конкретних розділів каталогу, перевірку та публікацію нових оголошень, а також взаємодію з підприємцями в межах своєї зони відповідальності. Звичайний користувач отримує доступ до функцій перегляду, пошуку та взаємодії з оголошеннями, зокрема можливість залишати відгуки та зберігати обрані пропозиції. Кожна роль повинна мати чітко визначений набір повноважень, що сприятиме підтриманню порядку, безпеки та передбачуваності в інформаційному середовищі системи.

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

Задачею дослідження є також проєктування структури бази даних, яка повинна забезпечувати цілісність, узгодженість та ефективність зберігання інформації. Якісно спроектована база даних є основою надійної роботи всієї системи, оскільки визначає швидкість виконання запитів, зручність підтримки та потенціал масштабування у майбутньому. Необхідно визначити основні сутності системи, їхні атрибути та взаємозв'язки, а також реалізувати механізми обмеження доступу до даних та запобігання несанкціонованим змінам. Проєктування бази даних повинно охоплювати нормалізацію таблиць з метою усунення надлишковості даних, визначення первинних і зовнішніх ключів, а також розроблення індексів для прискорення виконання пошукових запитів.

Окремою задачею дослідження є забезпечення адаптивності інтерфейсу системи. Згідно зі статистичними даними, понад половина інтернет-трафіку генерується з мобільних пристроїв, що робить мобільну адаптацію обов'язковою умовою конкурентоспроможності будь-якого сучасного веб-застосунку. Система повинна коректно відображатися та повноцінно функціонувати на персональних комп'ютерах, планшетах і мобільних пристроях незалежно від розміру екрана та операційної системи користувача. Реалізація цієї вимоги передбачає застосування принципів адаптивного дизайну, зокрема підходу «mobile-first», гнучких сіток компоновання та медіазапитів, що дозволяють динамічно адаптувати макет сторінки до параметрів пристрою.

Важливим напрямом у рамках постановки задачі є визначення вимог до системи оцінювання якості послуг через механізм відгуків та рейтингу. В умовах інформаційної асиметрії між продавцями та покупцями такий функціонал відіграє роль соціального доказу, що суттєво впливає на прийняття рішень потенційними клієнтами. Відгуки формують довіру між користувачами та підприємцями, а також стимулюють підприємців до підвищення якості послуг. Водночас реалізація цієї функції потребує врахування можливих зловживань, зокрема публікації неправдивих або замовних відгуків. Тому система повинна передбачати механізми верифікації відгуків, обмеження частоти їх публікації

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

одним користувачем та надання адміністраторам інструментів модерації контенту.

Не менш важливою задачею є забезпечення якісного пошукового функціоналу. Рівень зручності та релевантності пошуку безпосередньо визначає користувацький досвід роботи з платформою, оскільки саме через пошук відбувається основна взаємодія між споживачем і каталогом. Система повинна забезпечувати швидкий та точний пошук за назвою компанії, категорією, ключовими словами або іншими параметрами. Це передбачає реалізацію ефективних алгоритмів обробки запитів, впровадження механізмів повнотекстового пошуку, автодоповнення запитів та фільтрації результатів за кількома критеріями одночасно, а також мінімізацію часу відповіді сервера на пошукові запити. Задача дослідження включає також визначення вимог до інформаційної безпеки системи. Надійність і захищеність є одними з ключових критеріїв практичної цінності будь-якого веб-застосунку, а будь-який інцидент безпеки здатний завдати непоправної шкоди репутації платформи та підірвати довіру користувачів. Система повинна забезпечувати захист персональних даних користувачів відповідно до чинних нормативних вимог, реалізовувати надійні механізми хешування паролів та управління сесіями, а також унеможлиблювати типові вектори атак, зокрема SQL-ін'єкції, міжсайтовий скриптинг та підробку міжсайтових запитів. Передача даних між клієнтом і сервером повинна здійснюватися виключно по захищеному протоколу.

Для досягнення поставленої мети необхідно вирішити такий комплекс задач: проаналізувати існуючі рішення у сфері каталогів та визначити їхні переваги і недоліки; сформулювати повний перелік функціональних і нефункціональних вимог до системи; обґрунтувати вибір архітектурного підходу та спроектувати загальну структуру застосунку; розробити модель бази даних із визначенням усіх сутностей, зв'язків та обмежень; реалізувати основні функціональні модулі системи; забезпечити адаптивність інтерфейсу та захищеність даних; провести тестування розробленого застосунку і оцінити відповідність результатів висунутим вимогам.

					ІІЗ.КР.Б – 121-26-ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Практична значущість дослідження полягає у створенні готового до впровадження інструменту цифрового представництва для суб'єктів малого підприємництва. Реалізація поставленої задачі дозволить створити функціональну, зручну та безпечну інформаційну систему, що стане ефективним інструментом підтримки малого підприємництва в умовах цифрової економіки, сприятиме підвищенню їхньої конкурентоспроможності та розширенню клієнтської бази, а також забезпечить подальшу інтеграцію малого бізнесу в сучасний інформаційний простір.

1.2 Аналіз існуючих програмних рішень каталогів

Сучасний ринок цифрових сервісів пропонує значну кількість платформ, що виконують функції онлайн-каталогів підприємств та бізнес-довідників. Такі системи забезпечують зручний інструмент для пошуку компаній, товарів та послуг, а також надають підприємствам можливість представлення власної діяльності в мережі Інтернет[1]. Каталоги відіграють важливу роль у розвитку цифрової економіки, оскільки забезпечують ефективну взаємодію між бізнесом та споживачами.

Для визначення вимог до майбутнього веб-застосунку доцільно провести аналіз існуючих рішень. У межах дослідження розглянуто два популярні сервіси каталогів: Google Business Profile та UA-REGION.

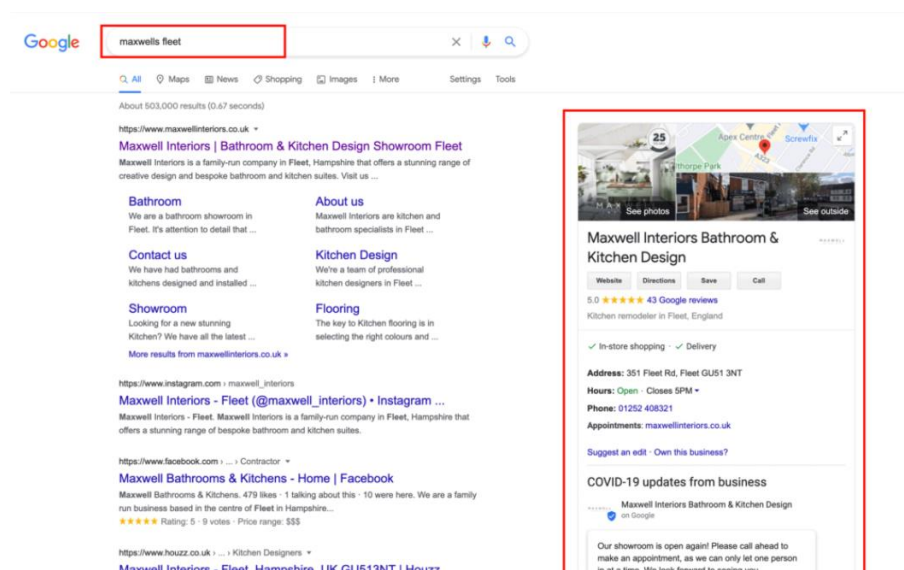


Рис.1.1. Сторінка Google Business Profile

Google Business Profile є одним із найпопулярніших сервісів для представлення бізнесу в Інтернеті. Платформа дозволяє компаніям створювати власні профілі, які відображаються у результатах пошуку Google та на карті Google Maps.

Функціональні можливості сервісу охоплюють створення та редагування профілю компанії, відображення контактної інформації, адреси та графіка роботи, інтеграцію з картами Google Maps, додавання фотографій та опису діяльності. Окрім цього, платформа містить розвинену систему відгуків та рейтингів, що дозволяє користувачам залишати коментарі й оцінки, а власникам бізнесу — відстежувати статистику переглядів та взаємодій з профілем. До основних переваг сервісу належать безкоштовне розміщення інформації, тісна інтеграція з пошуковою системою Google, висока довіра з боку користувачів, значна охоплена аудиторія та наявність функціоналу збору відгуків[2]. Ці характеристики роблять Google Business Profile привабливим інструментом для малого бізнесу, що шукає доступний спосіб підвищити свою впізнаваність в онлайн-середовищі.

Водночас платформа має суттєві недоліки. Можливості кастомізації профілю є обмеженими, а власники бізнесу повністю залежать від політики та алгоритмів Google, які можуть змінюватися без попередження. Модерація відгуків є складною та не завжди ефективною, що відкриває можливості для зловживань. Крім того, відсутність повного контролю над структурою каталогу унеможливорює гнучке налаштування системи під специфічні потреби окремих підприємців.

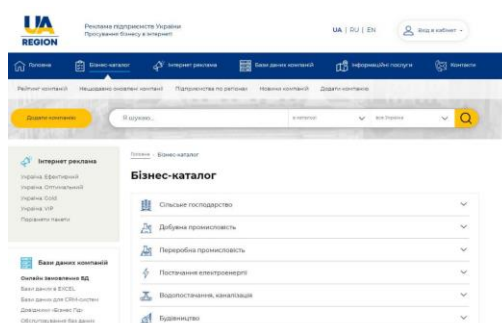


Рис.1.2. сторінка UA-REGION

					ІІЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

UA-REGION є українським онлайн-каталогом підприємств, який надає інформацію про компанії, організації та установи України. Платформа функціонує як електронний довідник, що дозволяє користувачам знаходити підприємства за категоріями діяльності або регіонами. Функціональні можливості сервісу включають пошук підприємств за категоріями та регіонами або містами, відображення контактних даних компаній і перегляд опису їхньої діяльності. Додатково платформа надає можливість публікації новин та інформаційних матеріалів від імені компанії, що дозволяє підприємствам підтримувати актуальність свого представництва в каталозі.

Серед переваг платформи варто відзначити орієнтацію на український ринок, наявність великої бази підприємств, можливість пошуку за регіональною ознакою та загальну простоту використання. Ці характеристики роблять UA-REGION зручним інструментом для користувачів, які шукають інформацію про вітчизняні компанії в конкретному географічному регіоні. Проте платформа має низку суттєвих недоліків[3]. Інтерфейс сервісу є візуально застарілим і не відповідає сучасним стандартам веб-дизайну. Функціонал взаємодії з користувачами залишається обмеженим, а механізми відгуків фактично відсутні, що унеможливорює формування довіри між споживачами та підприємствами на основі реального досвіду. Додатково платформа погано адаптована для перегляду на мобільних пристроях, що суттєво знижує її доступність в умовах домінування мобільного інтернет-трафіку.

Проведений аналіз сучасних програмних рішень, що виконують функції каталогів та онлайн-довідників підприємств, дозволяє зробити висновок, що кожна з розглянутих платформ має власні особливості функціонування, а також певні переваги та недоліки. Досліджені сервіси відрізняються за рівнем функціональності, орієнтацією на різні ринки, можливостями взаємодії з користувачами та рівнем технологічної реалізації. Глобальні платформи, такі як Google Business Profile, характеризуються високим рівнем інтеграції з пошуковими системами, значною аудиторією користувачів та високим рівнем довіри з боку споживачів. Водночас такі сервіси мають обмежені можливості

					ІПЗ.КР.Б – 121-26-ІЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

кастомізації, а також передбачають залежність бізнесу від політики зовнішньої платформи. Це означає, що власник компанії не має повного контролю над способом представлення своєї інформації або функціональністю сервісу.

Регіональні онлайн-каталоги, зокрема UA-REGION, орієнтовані на локальний ринок і дозволяють шукати підприємства за категоріями або регіонами. Проте їхній інтерфейс і функціональність часто поступаються сучасним вимогам до зручності, адаптивності та інтерактивності, таблиця 1.1.

Таблиця 1.1.

Порівняльна характеристика існуючих каталогів

Критерій	Google Business Profile	UA-REGION
Тип платформи	Глобальний сервіс для представлення бізнесу у пошуку та на картах	Український онлайн-довідник підприємств
Географічне охоплення	Міжнародне, доступний у більшості країн світу	Переважно Україна
Інтеграція з іншими сервісами	Тісна інтеграція з Google Search та Google Maps	Обмежена інтеграція з іншими сервісами
Реєстрація компанії	Безкоштовна реєстрація з підтвердженням бізнесу	Реєстрація підприємства у каталозі
Відображення контактної інформації	Адреса, телефон, сайт, графік роботи	Адреса, телефон, опис діяльності
Система відгуків та рейтингів	Є система оцінювання та коментарів користувачів	Обмежена або відсутня система відгуків
Пошук та фільтрація	Пошук за назвою, категоріями, геолокацією	Пошук за категоріями та регіонами
Мобільна адаптивність	Повністю адаптований для мобільних пристроїв	Обмежена або часткова адаптація
Інтерфейс користувача	Сучасний та інтуїтивно зрозумілий	Порівняно застарілий дизайн
Рівень кастомізації профілю	Обмежений набір налаштувань	Базове редагування інформації
Статистика та аналітика	Доступна статистика переглядів і взаємодій	Практично відсутня
Контроль над платформою	Залежність від політики Google	Частковий контроль з боку адміністраторів каталогу
Основні переваги	Велика аудиторія, інтеграція з картами, висока довіра користувачів	Орієнтація на український бізнес, простота використання
Основні недоліки	Обмежена кастомізація, залежність від зовнішньої платформи	Застарілий інтерфейс, обмежений функціонал

Порівняльний аналіз показує, що Google Business Profile забезпечує значно ширші можливості для представлення бізнесу та взаємодії з користувачами завдяки інтеграції з пошуковою системою та картографічним сервісом. Водночас платформа має обмежені можливості кастомізації та створює залежність бізнесу від політики зовнішнього сервісу. UA-REGION, у свою чергу, орієнтований на локальний український ринок і дозволяє здійснювати пошук підприємств за регіонами та категоріями діяльності. Проте платформа поступається сучасним сервісам за рівнем функціональності, інтерфейсу та інтерактивності[4].

Отже, аналіз існуючих систем підтверджує доцільність розроблення власного веб-застосунку каталогу, який поєднуватиме сучасний інтерфейс, зручні механізми пошуку, систему ролей користувачів та високий рівень інформаційної безпеки.

1.3 Вибір архітектури веб-застосунку каталогу

Вибір архітектури веб-застосунку є одним із ключових рішень, що приймаються на початкових етапах проєктування. Від нього залежить структура системи, зручність її підтримки, масштабування та подальшого розвитку. Під час розробки каталогу для малого бізнесу необхідно було знайти баланс між гнучкістю архітектури та простотою реалізації. Перед вибором архітектури було розглянуто кілька поширених підходів, зокрема монолітну архітектуру, мікросервісну модель, клієнт-серверну взаємодію та шаблон Model–View–Controller. Кожен із цих підходів має свої переваги та сферу застосування залежно від масштабу проєкту та вимог до системи. Монолітна архітектура передбачає розміщення всієї логіки застосунку в єдиній кодовій базі. Такий підхід є досить простим у розгортанні та зручним для невеликих проєктів. Проте за відсутності чіткої структури код може швидко ускладнюватися, що ускладнює підтримку та розвиток системи. Мікросервісна архітектура, навпаки, передбачає поділ застосунку на незалежні сервіси, які взаємодіють між собою через API. Цей підхід забезпечує високу масштабованість та активно використовується у великих корпоративних системах. Однак для проєкту масштабу каталогу така

архітектура є надмірно складною та потребує значних ресурсів для реалізації і підтримки.

Клієнт-серверна модель лежить в основі більшості сучасних веб-застосунків. У цій моделі браузер виступає клієнтом, який надсилає запити до сервера, а сервер обробляє їх та повертає відповідь. Саме в рамках цієї моделі реалізується обраний у даній роботі архітектурний шаблон MVC. Шаблон Model–View–Controller передбачає розподіл функціональності системи між трьома компонентами: моделлю, представленням та контролером. Такий підхід дозволяє розділити бізнес-логіку, роботу з даними та інтерфейс користувача, що значно спрощує підтримку та розвиток системи. Компонент Model відповідає за роботу з даними[5]. У межах даного проєкту він забезпечує збереження інформації про компанії, відгуки користувачів, категорії та облікові записи. Модель взаємодіє з базою даних та надає необхідні дані іншим компонентам системи.

Компонент View відповідає за відображення інформації користувачеві. Представлення реалізовані у вигляді HTML-шаблонів, які отримують підготовлені дані від контролера та відображають їх на сторінці. Такий підхід дозволяє відокремити логіку інтерфейсу від серверної логіки. Компонент Controller виконує роль посередника між моделлю та представленням. Він приймає HTTP-запити від користувача, обробляє їх, викликає відповідні методи моделі та передає отримані дані до шаблонів для відображення. У межах проєкту кожна дія користувача, наприклад перегляд каталогу, пошук компаній або додавання відгуку, обробляється за цією схемою: запит надходить до контролера, який звертається до моделі для отримання даних, після чого передає їх до представлення для формування HTML-сторінки. Для серверної частини застосунку обрано мову програмування PHP версії 8.x. Основними причинами такого вибору є її орієнтованість на веб-розробку, широка підтримка хостинг-провайдерами та висока продуктивність сучасних версій мови. Для роботи з базою даних використовується розширення PDO (PHP Data Objects). Його основною перевагою є підтримка підготовлених запитів, що дозволяє ефективно

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

захищати застосунок від SQL-ін'єкцій. Крім того, PDO забезпечує універсальний інтерфейс для взаємодії з різними системами управління базами даних.

Клієнтська частина інтерфейсу побудована із використанням CSS-фреймворку Bootstrap 5, що дозволяє реалізувати адаптивний дизайн та забезпечити коректне відображення сайту на різних пристроях. Маршрутизація запитів реалізована за принципом єдиної точки входу. Усі запити перенаправляються на файл index.php, який аналізує URL-адресу та визначає відповідний контролер і метод для виконання. Система автентифікації реалізована з використанням PHP-сесій[6]. Після входу користувача у систему у сесії зберігається його ідентифікатор та роль доступу. У системі передбачено три ролі: адміністратор, менеджер та звичайний користувач. Адміністратор має повний доступ до керування системою, менеджер може редагувати оголошення та категорії, а користувач має можливість створювати власні оголошення та залишати відгуки.

Структура проекту відповідає принципам MVC. Основна бізнес-логіка розміщена у директорії app, що містить папки models, views та controllers. Публічні файли, такі як стилі, скрипти та зображення, розміщені у директорії public, а конфігураційні параметри системи винесені у папку config. Для локальної розробки використовується пакет WAMP Server, який включає веб-сервер Apache, мову PHP та систему управління базами даних MySQL. Це значно спрощує налаштування середовища розробки та тестування застосунку. Безпека системи забезпечується використанням підготовлених SQL-запитів, хешуванням паролів за допомогою алгоритму bcrypt, захистом від міжсайтового скриптингу шляхом екранування даних та використанням CSRF-токенів у формах. Таким чином, використання архітектурного шаблону MVC дозволяє створити структурований, масштабований та безпечний веб-застосунок каталогу, що відповідає сучасним вимогам веб-розробки. Підсумовуючи, можна впевнено стверджувати, що архітектура MVC у поєднанні з технологічним стеком PHP/MySQL/Bootstrap стала оптимальним вибором для реалізації веб-застосунку каталогу.

					ІПЗ.КР.Б – 121-26-ІЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

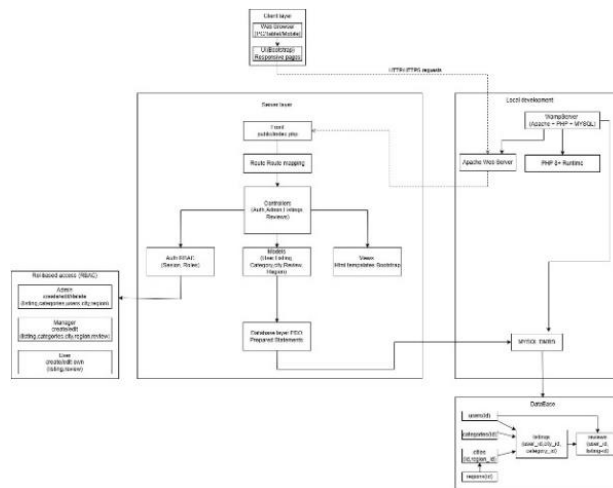


Рис.1.3. Структурна архітектура каталогу з використанням шаблону Model–View–Controller

Вона забезпечує чіткий поділ відповідальностей, підтримує рольову модель доступу, дозволяє безпечно працювати з даними та легко розширювати функціонал. Все це робить систему придатною, як основу для реального продукту, що може бути розгорнутий і використаний малими підприємствами, рис.1.3.

Висновки до першого розділу

У першому розділі було розглянуто теоретичні та прикладні аспекти використання інформаційних технологій у сфері каталогів для малого бізнесу. Проаналізовано сучасні тенденції цифровізації підприємницької діяльності та визначено роль веб-застосунків як інструменту представлення компаній у мережі Інтернет. Встановлено, що онлайн-каталоги створюють для малого бізнесу додаткові можливості для підвищення видимості, розширення клієнтської бази та спрощення взаємодії з потенційними споживачами.

У процесі дослідження було сформульовано постановку задачі, яка передбачає розроблення веб-застосунку каталогу з урахуванням функціональних і нефункціональних вимог. Визначено необхідність реалізації реєстрації та авторизації користувачів, багаторівневої системи ролей, управління оголошеннями, категоріями, містами й регіонами, пошуку та фільтрації, а також системи відгуків і рейтингового оцінювання. Окремо окреслено вимоги до безпеки, продуктивності, адаптивності та зручності користувацького інтерфейсу.

Проведений аналіз існуючих програмних рішень дозволив виявити їхні переваги та недоліки. Встановлено, що глобальні платформи мають широку аудиторію та високий рівень довіри, проте обмежують можливості кастомізації та повного контролю над функціональністю. Регіональні каталоги є більш орієнтованими на локальний ринок, однак часто поступаються сучасним вимогам до дизайну, адаптивності та інтерактивності. Це обґрунтовує доцільність створення власного веб-застосунку каталогу малого бізнесу.

Особливу увагу було приділено вибору архітектурного підходу до реалізації системи. На основі аналізу можливих рішень визначено доцільність використання шаблону Model–View–Controller, який забезпечує розмежування логіки роботи з даними, обробки запитів і відображення інформації користувачу. Таким чином, перший розділ сформував теоретичну та практичну основу для подальшого проєктування структури бази даних, розробки функціональних модулів і реалізації програмного продукту.

					ІПЗ.КР.Б – 121-26-ІПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБ-ЗАСТОСУНКУ КАТАЛОГУ ДЛЯ МАЛОГО БІЗНЕСУ

2.1 Визначення варіантів використання та структури системи

Проектування веб-застосунку каталогу для малого бізнесу передбачає визначення основних функціональних можливостей системи, ролей користувачів та взаємодії між окремими компонентами. На цьому етапі важливо встановити, які саме дії повинна підтримувати система, які категорії користувачів будуть із нею працювати та які об'єкти формують її внутрішню структуру. Такий підхід дає змогу не лише сформулювати вимоги до майбутнього програмного продукту, а й створити основу для подальшого проектування архітектури, бази даних і програмних модулів.

Основною метою розроблення системи є створення веб-застосунку, який забезпечить зручне цифрове представництво підприємств малого бізнесу, дозволить користувачам знаходити потрібні компанії за категоріями, переглядати інформацію про них, а також взаємодіяти з платформою через механізм відгуків і рейтингу[7]. Водночас система повинна підтримувати адміністрування контенту, керування обліковими записами та розмежування прав доступу між різними категоріями користувачів. Для опису функціональних можливостей системи доцільно використати діаграму варіантів використання, яка відображає основних акторів та їхню взаємодію з веб-застосунком.

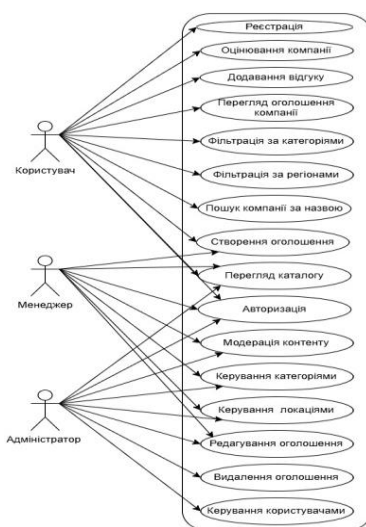


Рис.2.1. Діаграма варіантів використання веб-застосунку каталогу

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

У системі передбачено три основні актори: адміністратор, менеджер та користувач. Кожен із них має власний набір дій і прав доступу відповідно до ролі в системі, рис.2.1.

Наведена діаграма демонструє, що найбільш базові функції системи доступні всім користувачам, зокрема перегляд каталогу, пошук компаній і перегляд профілів. Зареєстрований користувач також може створювати власні оголошення, залишати відгуки та оцінювати компанії. Менеджер має розширені повноваження, що дозволяють редагувати оголошення, керувати категоріями та здійснювати модерацію контенту. Найширші можливості має адміністратор, який, окрім зазначених функцій, може видаляти оголошення та керувати користувачами системи. Для більшої наочності права доступу акторів доцільно подати у вигляді таблиці 2.2.

Таблиця 2.2.

Актори системи та їхні функції

Актор	Опис ролі	Основні можливості
Користувач	Зареєстрований або незареєстрований відвідувач платформи	Перегляд каталогу, пошук компаній, фільтрація за категоріями, перегляд профілів компаній, реєстрація, авторизація, створення власних оголошень, додавання відгуків та оцінок
Менеджер	Уповноважений користувач, який здійснює часткове адміністрування контенту	Усі можливості користувача, а також редагування оголошень, керування категоріями, керування локаціями, модерація контенту
Адміністратор	Користувач із повним доступом до системи	Усі можливості менеджера, а також видалення оголошень, керування користувачами, повне адміністрування системи

З таблиці видно, що система реалізує багаторівневу модель доступу, яка дозволяє чітко розмежувати функції між різними категоріями учасників. Такий підхід забезпечує безпеку, спрощує адміністрування та зменшує ризик несанкціонованих дій[8]. Окрім зовнішньої взаємодії користувачів із системою, важливим етапом проєктування є визначення її внутрішньої структури. Для

цього використовується діаграма класів, яка відображає основні сутності предметної області, їхні атрибути та взаємозв'язки між ними. Опис діаграми варіантів використання наведено в таблицях 2.3 – 2.14.

Таблиця 2.3

Опис прецеденту «Авторизація користувача»

Назва варіанту використання	Авторизація користувача
Основні актори	Адміністратор, менеджер, користувач
Короткий опис	Користувач вводить електронну пошту та пароль для входу в систему. Після успішної перевірки облікових даних система надає доступ до функціоналу відповідно до ролі користувача.

Таблиця 2.4

Опис прецеденту «Реєстрація користувача»

Назва варіанту використання	Реєстрація користувача
Основні актори	Користувач
Короткий опис	Новий користувач заповнює форму реєстрації, вказуючи ім'я, електронну пошту та пароль. Система перевіряє коректність даних, створює обліковий запис і автоматично авторизує користувача.

Таблиця 2.5

Опис прецеденту «Вихід із системи»

Назва варіанту використання	Вихід із системи
Основні актори	Адміністратор, менеджер, користувач
Короткий опис	Авторизований користувач завершує роботу із системою. Після виконання дії обліковий запис виходить із сесії, а користувач перенаправляється на головну сторінку.

Таблиця 2.6

Опис прецеденту «Перегляд каталогу оголошень»

Назва варіанту використання	Перегляд каталогу оголошень
Основні актори	Гість, користувач, менеджер, адміністратор
Короткий опис	Користувач переглядає список оголошень у каталозі. Система виводить перелік доступних оголошень з урахуванням пошукового запиту та обраної категорії.

Таблиця 2.7

Опис прецеденту «Перегляд оголошення»

Назва варіанту використання	Перегляд оголошення
Основні актори	Гість, користувач, менеджер, адміністратор
Короткий опис	Користувач відкриває сторінку конкретного оголошення для перегляду детальної інформації про компанію, а також відгуків і рейтингу. Якщо користувач авторизований, система додатково може відобразити його власний відгук.

Таблиця 2.8

Опис прецеденту «Створення оголошення»

Назва варіанту використання	Створення оголошення
Основні актори	Користувач, менеджер, адміністратор
Короткий опис	Авторизований користувач створює нове оголошення, заповнюючи форму з даними компанії, описом, контактною інформацією, категорією та локацією. Після перевірки введених даних система зберігає оголошення та відображає його сторінку.

Таблиця 2.9

Опис прецеденту «Редагування оголошення»

Назва варіанту використання	Редагування оголошення
Основні актори	Власник оголошення, менеджер, адміністратор
Короткий опис	Авторизований користувач, який є власником оголошення, або користувач із розширеними правами може змінювати інформацію про оголошення. Після збереження змін система оновлює дані та відображає актуальну сторінку оголошення.

Таблиця 2.10

Опис прецеденту «Додавання або оновлення відгуку»

Назва варіанту використання	Додавання або оновлення відгуку
Основні актори	Користувач
Короткий опис	Авторизований користувач залишає оцінку та текстовий відгук про компанію. Якщо відгук уже існує, система оновлює його; якщо ні — створює новий запис.

Таблиця 2.11

Опис прецеденту «Видалення відгуку»

Назва варіанту використання	Видалення відгуку
Основні актори	Користувач, менеджер, адміністратор
Короткий опис	Авторизований користувач може видалити власний відгук, а користувачі з адміністративними правами можуть видаляти відгуки в межах модерації контенту. Після цього система прибирає відгук зі сторінки оголошення.

Таблиця 2.12

Опис прецеденту «Керування категоріями»

Назва варіанту використання	Керування категоріями
Основні актори	Менеджер, адміністратор
Короткий опис	Користувач із роллю менеджера або адміністратора переглядає список категорій, додає нові категорії та редагує наявні. Видалення категорії доступне лише адміністратору.

Таблиця 2.13

Опис прецеденту «Керування користувачами»

Назва варіанту використання	Керування користувачами
Основні актори	Адміністратор
Короткий опис	Адміністратор переглядає список користувачів, створює нові облікові записи, змінює ролі користувачів та видаляє їх, за винятком власного облікового запису.

Таблиця 2.14

Опис прецеденту «Видалення оголошення адміністратором»

Назва варіанту використання	Видалення оголошення адміністратором
Основні актори	Адміністратор
Короткий опис	Адміністратор видаляє оголошення із системи в межах адміністрування та модерації контенту. Після виконання дії система прибирає оголошення зі списку доступних записів.

Для відображення внутрішньої структури веб-застосунку каталогу використовується діаграма класів, яка показує основні сутності системи, їхні методи та зв'язки між ними. Дана діаграма дає змогу формалізувати логіку предметної області та визначити, які об'єкти беруть участь у роботі системи.

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

У розробленому веб-застосунку основними класами є User, Listing, Category та Review[9]. Саме вони відображають ключові сутності системи: користувачів, оголошення, категорії та відгуки. Клас User призначений для роботи з користувачами системи. Він забезпечує пошук користувача за електронною поштою або ідентифікатором, створення нового облікового запису, отримання списку всіх користувачів, зміну ролі та видалення користувача. Цей клас є основою реалізації механізмів автентифікації та авторизації, оскільки містить інформацію про роль користувача в системі: admin, manager або user.

Клас Listing представляє оголошення компанії в каталозі. Він містить методи для пошуку оголошень за заданими фільтрами, отримання окремого оголошення, виведення оголошень конкретного користувача, створення нового запису, редагування існуючого та його видалення. Також цей клас використовується для відображення повного списку оголошень в адміністративній частині системи. Клас Category відповідає за роботу з категоріями каталогу. Він забезпечує отримання списку всіх категорій, створення нових категорій, оновлення та видалення існуючих. Додатково в цьому класі реалізовано формування унікального символьного ідентифікатора slug, який використовується для коректного представлення категорій у системі. Клас Cities відповідає за місцезнаходження бізнесів. Він забезпечує отримання списку всіх міст, створення нових, оновлення та видалення існуючих. Додатково в цьому класі реалізовано обирання областей класом Region де є списки областей, який формує детальну інформацію у каталозі.

Клас Review призначений для роботи з відгуками користувачів. Він дозволяє отримувати відгуки для конкретного оголошення, знаходити відгук певного користувача, додавати або оновлювати відгук, а також видаляти його. Саме цей клас забезпечує функціонування підсистеми рейтингу та зворотного зв'язку. Між наведеними класами існують логічні зв'язки[10]. Один користувач може створювати декілька оголошень, тому між класами User і Listing існує зв'язок типу один до багатьох. Одна категорія може містити багато оголошень, отже між класами Category і Listing також встановлюється зв'язок один до

багатьох. Одне оголошення може мати багато відгуків, тому між класами Listing і Review існує аналогічний зв'язок. Водночас один користувач може залишати декілька відгуків, що формує зв'язок один до багатьох між класами User і Review.

Таким чином, діаграма класів відображає внутрішню логіку веб-застосунку каталогу, демонструє основні сутності системи та взаємозв'язки між ними. Вона є важливим етапом проєктування, оскільки створює основу для побудови структури бази даних і подальшої реалізації програмних модулів.

Таблиця 2.15

Опис класів та основних методів веб-застосунку

Клас	Призначення	Основні методи
User	Зберігання та обробка даних користувачів системи	findByEmail(), findById(), create(), all(), setRole() deleteById()
Listing	Робота з оголошеннями компаній у каталозі	search(), find(), my(), create(), update(), delete(), allAdmin()
Category	Керування категоріями каталогу	all(), slugify(), create(), update(), delete()
Region	Керування регіонами каталогу	all(), create(), update(), delete()
Cities	Керування містами каталогу	all(), create(), update(), delete()
Review	Робота з відгуками користувачів	forListing(), userReview(), upsert(), delete()

На рисунку 2.2 наведено діаграму класів веб-застосунку каталогу. Вона відображає основні сутності системи, їх функціональне призначення та взаємозв'язки між ними.

Центральними класами є User, Listing, Category, Region, Cities та Review, які забезпечують реалізацію основних функцій системи: керування користувачами, оголошеннями, категоріями та відгуками.

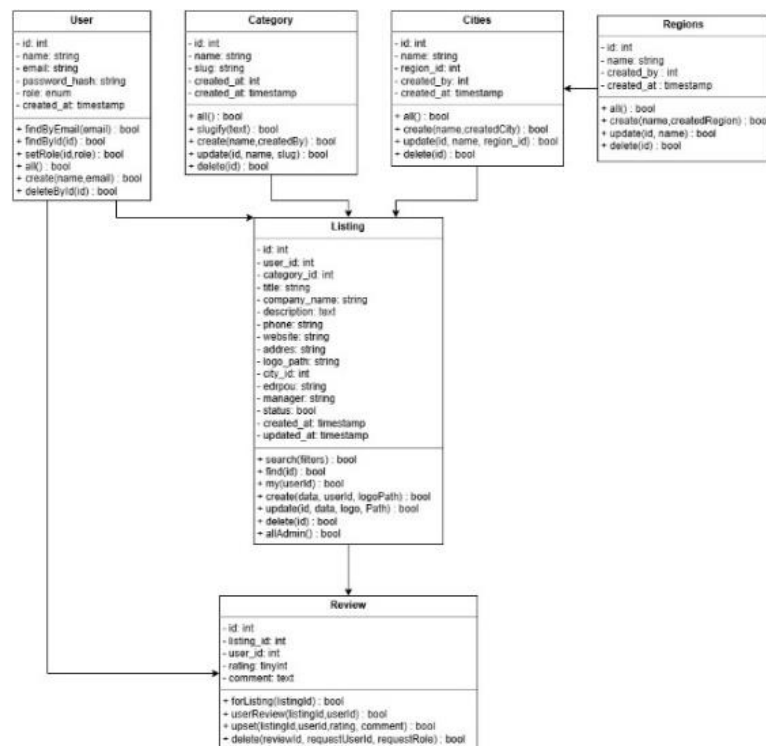


Рисунок 2.2 – Діаграма класів веб-застосунку каталогу

Функціональні вимоги до системи були сформовані з урахуванням потреб усіх категорій користувачів веб-застосунку каталогу. Для адміністратора, як внутрішнього користувача системи, веб-застосунок повинен забезпечувати повний контроль над управлінням платформою. Зокрема, адміністратор має можливість переглядати всі оголошення, видаляти або редагувати їх у разі порушення правил платформи, керувати категоріями бізнесу, створювати нові категорії та редагувати існуючі. Крім того, адміністратор повинен мати доступ до управління обліковими записами користувачів, змінювати їх ролі, створювати нові облікові записи та видаляти існуючі[11].

Менеджер, як ще один внутрішній користувач системи, повинен мати можливість авторизуватися у веб-застосунку, переглядати список оголошень, редагувати інформацію про компанії, а також керувати категоріями бізнесу. Менеджер може здійснювати модерацію контенту та контролювати актуальність інформації, що розміщується у каталозі. Водночас рівень доступу менеджера є обмеженим порівняно з адміністратором, оскільки він не має можливості видаляти користувачів або змінювати їх ролі.

Для користувачів системи, які є основною категорією учасників платформи, передбачено можливість реєстрації та авторизації, створення власного профілю, а також розміщення оголошень про компанії або послуги. Користувач повинен мати можливість створювати нові оголошення, редагувати або видаляти власні записи, переглядати каталог компаній, здійснювати пошук за ключовими словами та фільтрувати результати за категоріями. Крім того, система повинна забезпечувати можливість перегляду детальної інформації про компанію, а також залишення відгуків і оцінювання якості послуг.

Окрім функціональних вимог, були визначені також нефункціональні вимоги до системи, які включають продуктивність, надійність, безпеку та масштабованість веб-застосунку. У частині зручності використання передбачено, що новий користувач повинен мати змогу швидко освоїти роботу із системою без необхідності додаткового навчання. Інтерфейс веб-застосунку повинен бути інтуїтивно зрозумілим, адаптивним та забезпечувати зручну навігацію між основними розділами каталогу.

Щодо надійності функціонування, система повинна забезпечувати стабільну роботу та доступність для користувачів у будь-який час. Веб-застосунок повинен коректно обробляти запити користувачів, зберігати інформацію в базі даних та забезпечувати захист від втрати даних у випадку помилок або технічних збоїв. Передбачено використання механізмів резервного копіювання та контролю цілісності даних. Продуктивність системи передбачає швидке виконання основних операцій, таких як пошук оголошень, відкриття сторінки компанії або збереження відгуку користувача. Час відображення сторінок повинен бути мінімальним, а обробка запитів до бази даних має здійснюватися ефективно завдяки використанню оптимізованих SQL-запитів та індексів. У частині безпеки система повинна забезпечувати захист персональних даних користувачів та запобігати несанкціонованому доступу. Для цього передбачено використання механізмів автентифікації та авторизації, хешування паролів із використанням алгоритму bcrypt, перевірку вхідних даних, а також

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

захист від поширених веб-загроз, таких як SQL-ін'єкції та міжсайтовий скриптинг[12].

Щодо масштабованості, веб-застосунок повинен мати можливість подальшого розвитку та розширення функціоналу без суттєвих змін у структурі системи. Архітектура застосунку повинна дозволяти додавання нових модулів, інтеграцію з іншими сервісами та підтримку зростаючої кількості користувачів і обсягу даних. Це забезпечує довгострокову ефективність роботи системи та її адаптацію до майбутніх потреб[13] користувачів.

2.2 Розробка бази даних

База даних є ключовим компонентом будь-якого веб-застосунку, оскільки саме вона забезпечує зберігання, обробку та доступ до інформації. Для реалізації веб-застосунку бізнес-каталогу необхідно використовувати систему керування базами даних, яка забезпечує високу продуктивність, надійність та масштабованість. Перед вибором СУБД було проведено аналіз найбільш поширених систем керування базами даних, які використовуються у веб-розробці. Найбільш популярними серед них є MySQL, PostgreSQL та SQLite. Кожна з цих систем має власні переваги та обмеження, що необхідно враховувати при розробці інформаційної системи. Порівняння систем керування базами даних наведено в таблиці 2.1.

Таблиця 2.1

Порівняння СУБД для реалізації веб-застосунку

СУБД	Переваги	Недоліки
MySQL	висока швидкість роботи, простота адміністрування, широка підтримка у веб-розробці, інтеграція з PHP	обмежені можливості для складної аналітики
PostgreSQL	висока надійність, підтримка складних SQL-запитів, розширені можливості роботи з даними	складніше налаштування та адміністрування
SQLite	простота використання, не потребує окремого сервера	Обмежена продуктивність при великій кількості користувачів

У результаті проведеного аналізу було обрано MySQL, оскільки дана СУБД є оптимальним рішенням для веб-застосунків, що використовують мову програмування PHP. Вона забезпечує достатню продуктивність, стабільність роботи та простоту інтеграції з веб-сервером.

Структура бази даних веб-застосунку каталогу складається з шістьох основних таблиць:

- users – зберігає інформацію про користувачів системи
- cities – містить міста каталогу
- regions – містить регіони каталогу
- categories – містить категорії каталогу
- listings – містить оголошення компаній
- reviews – зберігає відгуки користувачів

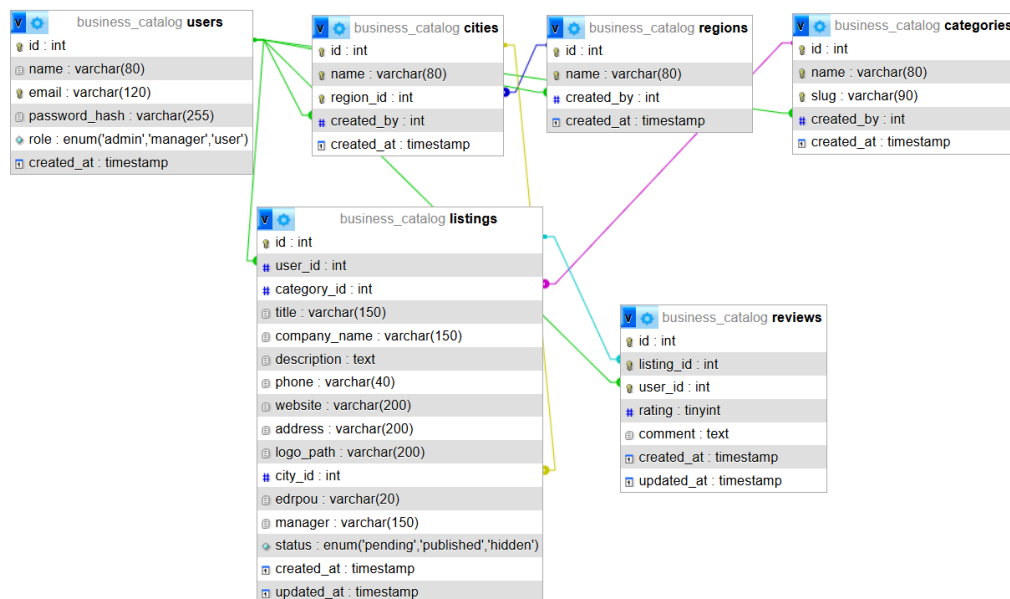


Рисунок 2.2 Схема бази даних веб-застосунку каталогу

На рисунку 2.2 представлено структуру бази даних системи, яка відображає таблиці та зв'язки між ними. Основні зв'язки реалізовані через зовнішні ключі, що забезпечують цілісність даних та правильну організацію взаємозв'язків між сутностями.

Структура сутностей бази даних

Таблиця	Назва поля	PK	FK	Опис
users	id	+	-	Унікальний ідентифікатор користувача
	name	-	-	Ім'я користувача
	email	-	-	Електронна пошта користувача
	Password_hash	-	-	Хешований пароль користувача
	role	-	-	Роль користувача в системі
	created_at	-	-	Дата створення облікового запису
categories	id	+	-	Унікальний ідентифікатор категорії
	name	-	-	Назва категорії
	Slug	-	-	Символьний ідентифікатор категорії
	created_by	-	-	Користувач, який створив категорію
	created_at	-	-	Дата створення категорії
regions	id	+	-	Унікальний ідентифікатор регіону
	name	-	-	Назва регіону
	created_by	-	-	Користувач, який створив регіон
	created_at	-	-	Дата створення регіону
cities	id	+	-	Унікальний ідентифікатор міста
	name	-	-	Назва міста
	region_id	-	+	Ідентифікатор регіону, до якого належить місто
	created_by	-	-	Користувач, який створив місто
	created_at	-	-	Дата створення міста
listings	id	+	-	Унікальний ідентифікатор оголошення
	user_id	-	+	Ідентифікатор користувача, який створив оголошення
	category_id	-	+	Ідентифікатор категорії оголошення
	title	-	-	Заголовок оголошення
	company_name	-	-	Назва компанії

продовж. табл. 2.2

	description	-	-	Опис діяльності компанії
	phone	-	-	Контактний номер телефону
	website	-	-	Вебсайт компанії
	address	-	-	Адреса компанії
	logo_path	-	-	Шлях до логотипу компанії
	city_id	-	+	Ідентифікатор міста, до якого належить оголошення
	edrpou	-	-	Код ЄДРПОУ компанії
	manager	-	-	Керівник компанії
	status	-	-	Статус оголошення
	created_at	-	-	Дата створення оголошення
	updated_at	-	-	Дата останнього оновлення оголошення
reviews	id	+	-	Унікальний ідентифікатор відгуку
	listing_id	-		Ідентифікатор оголошення, до якого належить відгук
	user_id	-	+	Ідентифікатор користувача, який залишив відгук
	rating	-	-	Оцінка компанії
	comment	-	-	Текстовий коментар користувача
	created_at	-	-	Дата створення відгуку
	updated_at	-	-	Дата останнього оновлення відгуку

Таблиця 2.3

Зв'язки між сутностями бази даних

Батьківська таблиця	Дочірня таблиця	Тип зв'язку
users	listings	1 : M
categories	listings	1 : M
regions	cities	1 : M
cities	listings	1 : M
users	reviews	1 : M
listings	reviews	1 : M

Такі зв'язки забезпечують логічну структуру бази даних та дозволяють ефективно організувати зберігання інформації у системі. Таким чином, кожна таблиця бази даних відіграє окрему важливу роль у функціонуванні веб-застосунку каталогу. Таблиця users забезпечує зберігання інформації про користувачів системи та реалізує механізм автентифікації й розмежування прав доступу відповідно до ролей[14]. Таблиця categories використовується для організації структури каталогу та дозволяє класифікувати компанії за видами діяльності. Таблиці cities, regions використовуються для організації структури каталогу та дозволяє надати інформацію компанії за локацією. Таблиця listings є центральним елементом бази даних, оскільки містить основну інформацію про компанії та їх оголошення, які відображаються у каталозі. Таблиця reviews, у свою чергу, забезпечує можливість зберігання відгуків і рейтингових оцінок користувачів, що дозволяє формувати репутацію компаній та підвищує довіру до платформи.

Завдяки використанню зв'язків між таблицями забезпечується цілісність даних і логічна структура бази даних. Така організація дозволяє ефективно виконувати пошук інформації, обробляти запити користувачів та підтримувати взаємодію між різними елементами системи. У результаті сформована структура бази даних створює надійну основу для стабільної роботи веб-застосунку та подальшого розширення його функціональних можливостей.

2.3 Проєктування алгоритмів роботи веб-застосунку

Проєктування алгоритмів роботи веб-застосунку є важливим етапом створення інформаційної системи, оскільки саме алгоритмічна логіка визначає порядок обробки запитів, взаємодію компонентів архітектури та коректність функціонування всієї системи. Якість розроблених алгоритмів безпосередньо впливає на продуктивність застосунку, рівень безпеки та зручність користувацького досвіду, тому їх ретельне проєктування є невід'ємною складовою загального процесу розробки[15]. Для веб-застосунку каталогу алгоритми повинні забезпечувати швидку обробку даних, безпечну аутентифікацію користувачів, коректне управління правами доступу та

ефективну взаємодію з базою даних. Кожен алгоритм системи проєктується з урахуванням принципів мінімальної складності, передбачуваності поведінки та стійкості до некоректних вхідних даних або зловмисних дій з боку користувачів.

Загальна логіка роботи застосунку базується на клієнт-серверній моделі. Ця модель є фундаментальним архітектурним принципом сучасних веб-застосунків, що забезпечує чітке розмежування між інтерфейсом користувача та серверною логікою обробки даних. Користувач надсилає HTTP-запит через браузер, сервер приймає запит, маршрутизує його до відповідного контролера, який виконує бізнес-логіку та формує відповідь у вигляді HTML-сторінки. Такий підхід дозволяє централізувати обробку запитів, ефективно розподіляти відповідальність між компонентами системи та забезпечувати однорідність поведінки застосунку незалежно від типу пристрою, що використовує клієнт. Таким чином, кожен алгоритм функціонування системи можна розглядати як послідовність чітко визначених кроків від моменту отримання запиту до повернення результату клієнту, причому кожен крок цієї послідовності виконує конкретну функцію та може бути протестований незалежно від інших[16].

Алгоритм обробки запитів починається з центральної точки входу — файлу маршрутизації. Централізована маршрутизація є однією з ключових переваг MVC-архітектури, оскільки вона забезпечує єдину точку контролю над усіма вхідними запитами та дозволяє реалізувати глобальні перевірки безпеки до передачі управління відповідному контролеру. Після отримання URL-адреси система визначає відповідний контролер і метод, який відповідає за обробку конкретної дії. Такий підхід дозволяє централізувати логіку доступу та уникнути дублювання коду, а також суттєво спрощує подальше розширення функціоналу системи, оскільки додавання нових маршрутів не потребує змін у вже існуючих компонентах застосунку.

Реалізація функціоналу веб-застосунку каталогу передбачає створення програмних модулів, що забезпечують взаємодію користувача із системою та обробку даних на сервері. Основні функції системи реалізовані у вигляді контролерів і моделей відповідно до архітектури MVC. Контролери

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

відповідають за обробку HTTP-запитів, моделі — за роботу з базою даних, а представлення забезпечують відображення інформації користувачеві. Кожен модуль виконує окрему функцію, що спрощує підтримку коду та дозволяє надалі розширювати можливості системи без значних змін у її структурі.

Основними функціональними можливостями системи є:

- реєстрація користувачів
- авторизація та керування сесіями
- створення та редагування оголошень
- пошук і перегляд компаній
- додавання та редагування відгуків
- адміністрування користувачів, категорій та локацій

Кожна з цих функцій реалізована у вигляді окремих програмних методів.

Функція авторизації реалізована у контролері `AuthController` та забезпечує перевірку введених користувачем облікових даних. Після перевірки створюється користувацька сесія. Виконується перевірка CSRF-токена, пошук користувача за електронною поштою та перевірка відповідності введеного пароля хешу, що зберігається у базі даних. У разі успішної авторизації створюється сесія користувача.

Функція створення оголошення реалізована у контролері `ListingsController` та забезпечує додавання нового запису до таблиці `listings`.

Даний метод перевіряє авторизацію користувача, виконує перевірку CSRF-токена, обробляє завантаження логотипу та зберігає нове оголошення у базі даних.

Функція додавання відгуку реалізована у контролері `ReviewsController`.

Метод перевіряє авторизацію користувача, обробляє введені дані та додає або оновлює відгук у таблиці `reviews`.

Для наочного відображення логіки роботи функції авторизації доцільно використати діаграму активності, рис.2.3.

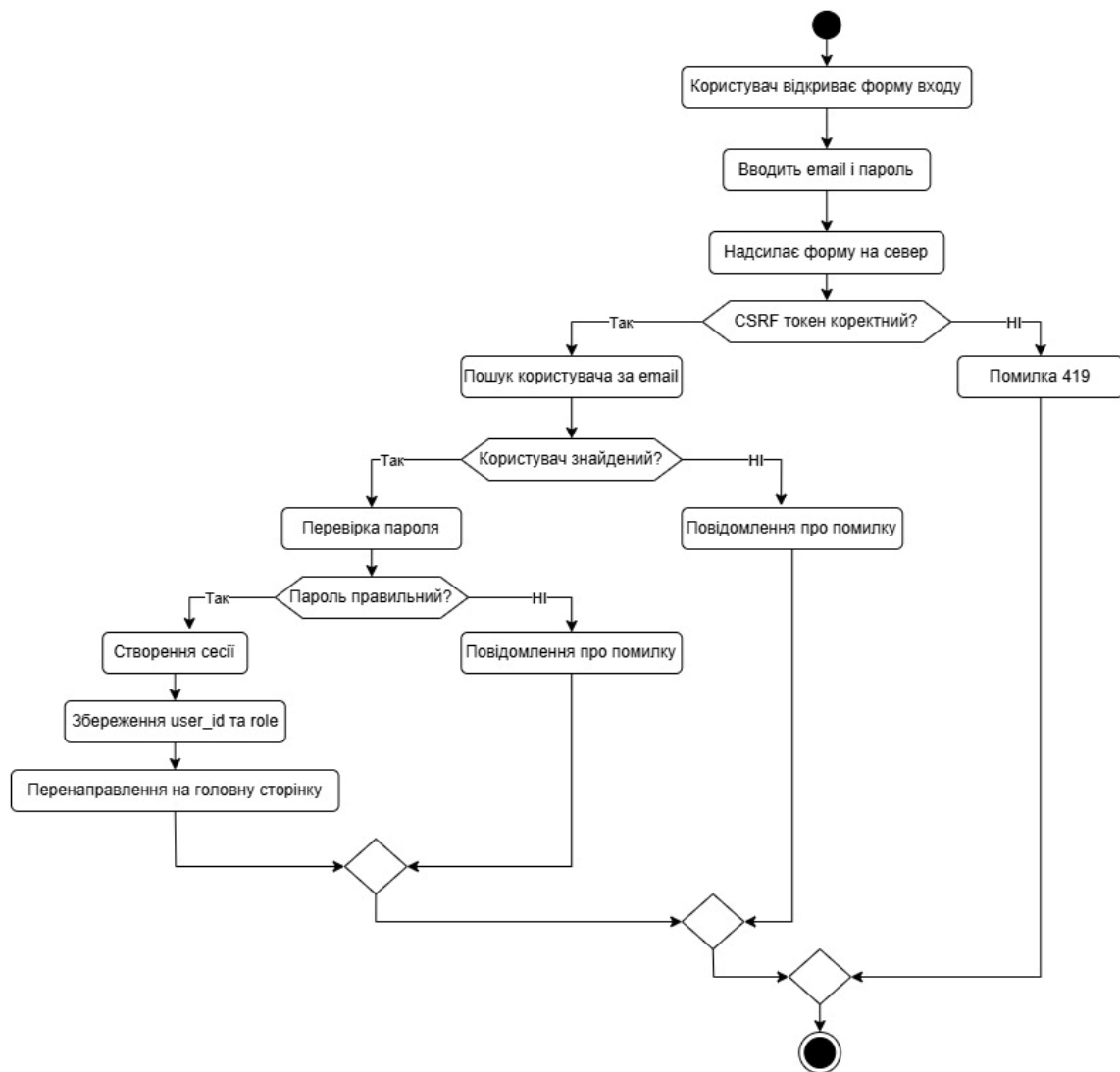


Рисунок 2.3 – Діаграма активності процесу авторизації користувача

Дана діаграма відображає послідовність дій, які виконуються під час входу користувача в систему.

Таким чином, реалізація функціоналу веб-застосунку каталогу базується на використанні архітектури MVC та модульного підходу до розробки програмного забезпечення. Основні функції системи реалізовані у вигляді окремих методів контролерів, які забезпечують обробку запитів користувачів, перевірку прав доступу та взаємодію з базою даних. Використання діаграм активності дозволяє наочно представити логіку роботи основних процесів системи та спрощує розуміння алгоритмів її функціонування.

2.4 Реалізація функціоналу веб-застосунку

Реалізація функціоналу веб-застосунку каталогу здійснювалася відповідно до сформованих вимог та обраної архітектури Model–View–Controller.

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

Програмна реалізація виконана мовою PHP із використанням об'єктно-орієнтованого підходу, що забезпечило структурованість коду, його модульність та зручність подальшого розширення. Інтерфейс користувача реалізовано із застосуванням HTML, CSS та фреймворку Bootstrap, що дозволило забезпечити адаптивність і сучасний дизайн системи[17].

Структурно веб-застосунок складається з кількох логічних модулів: модуль автентифікації та авторизації, модуль управління оголошеннями, модуль категорій, модуль міст і регіонів, модуль відгуків і рейтингу, а також адміністративна панель. Кожен модуль реалізовано у вигляді окремого контролера та відповідних моделей і шаблонів відображення. Такий підхід дозволив розділити функціональність системи на незалежні компоненти, що значно спрощує підтримку програмного коду та подальше масштабування застосунку.

Модуль автентифікації забезпечує реєстрацію нових користувачів, вхід до системи та вихід із неї. Реєстрація передбачає перевірку коректності введених даних, унікальності електронної пошти та хешування пароля з використанням алгоритму bcrypt. Після успішної реєстрації користувач може здійснити вхід до системи, під час якого створюється сесія із збереженням ідентифікатора та ролі доступу[15]. Для реалізації автентифікації використано сесійний механізм PHP, який дозволяє зберігати інформацію про авторизованого користувача між HTTP-запитами. Після успішного входу система автоматично визначає роль користувача та відкриває доступ до відповідного функціоналу.

Механізм авторизації реалізовано шляхом перевірки ролі користувача перед виконанням дій, що потребують відповідних прав. Для цього в системі передбачено три ролі: адміністратор, менеджер та користувач. Адміністратор має повний доступ до всіх функцій системи, включаючи управління категоріями, містами, регіонами та оголошеннями інших користувачів. Менеджер може створювати та редагувати оголошення й категорії, а звичайний користувач має право додавати власні оголошення та залишати відгуки. Така система

розмежування доступу реалізує модель RBAC та дозволяє обмежити виконання адміністративних операцій лише уповноваженим користувачам.

Модуль управління оголошеннями реалізує повний цикл операцій CRUD (Create, Read, Update, Delete). Користувач після авторизації може створити нове оголошення, заповнивши форму з інформацією про компанію, категорію, контактні дані та опис діяльності. Дані проходять перевірку на коректність і зберігаються в базі даних через модель, що виконує параметризований SQL-запит. Під час створення оголошення користувач також може завантажити логотип компанії, який автоматично зберігається на сервері, а шлях до нього записується у базу даних.

Перегляд оголошень реалізовано через формування списку компаній із можливістю фільтрації за категоріями, регіонами та містами, а також пошуку за назвою компанії або ключовими словами. Для цього використовується динамічний SQL-запит із врахуванням параметрів пошуку. Результати передаються до шаблону відображення та формуються у вигляді карток компаній із відображенням рейтингу, категорії, міста та основної контактної інформації. Для покращення інтерфейсу картки компаній мають адаптивний дизайн і автоматично підлаштовуються під різні розміри екранів.

На сторінці конкретного оголошення реалізовано відображення повної інформації про компанію. Користувач може переглянути опис діяльності, контактні дані, адресу, сайт, рейтинг та відгуки інших користувачів. Додатково реалізовано блок «Досьє компанії», де відображаються ЄДРПОУ та керівник компанії. Якщо вказано адресу, система автоматично формує інтегровану Google Maps карту, що дозволяє користувачам швидко визначити місцезнаходження підприємства.

Редагування оголошення можливе лише для його власника або адміністратора. Перед виконанням операції система перевіряє права доступу, після чого дозволяє оновити дані. Видалення записів здійснюється з урахуванням правил каскадного видалення, що забезпечує автоматичне

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

очищення пов'язаних відгуків. Це дозволяє підтримувати цілісність бази даних та уникати появи неактуальних записів.

Модуль категорій дозволяє адміністраторам та менеджерам створювати нові категорії діяльності, редагувати їх назви та видаляти за відсутності пов'язаних записів. Для кожної категорії формується унікальний slug, що використовується у формуванні URL-адрес та забезпечує SEO-оптимізацію[18]. Аналогічним чином реалізовано модулі міст і регіонів, які дозволяють структурувати компанії за географічним розташуванням та реалізувати пошук за локацією.

Система пошуку реалізована шляхом використання оператора LIKE або FULLTEXT-індексації в MySQL. Користувач може ввести назву компанії або ключове слово, після чого система повертає релевантні результати. Додатково реалізовано фільтрацію за категорією та регіоном, що дозволяє звузити вибірку відповідно до сфери діяльності та місцезнаходження бізнесу. Для підвищення швидкодії до таблиць бази даних були додані індекси для найчастіше використовуваних полів.

Модуль відгуків забезпечує можливість залишення оцінки та текстового коментаря лише зареєстрованими користувачами. Після додавання відгуку автоматично обчислюється середній рейтинг компанії. У разі повторної спроби залишити відгук для тієї самої компанії система оновлює попередній запис замість створення нового. Такий підхід дозволяє уникнути дублювання відгуків і забезпечує коректність рейтингової системи.

Для відображення рейтингу використовується система зірок, яка автоматично формується відповідно до середнього значення оцінок. Середній рейтинг компанії обчислюється SQL-запитом із використанням агрегатних функцій AVG та COUNT. Отримане значення передається до шаблону сторінки та відображається користувачу у візуальному вигляді.

Адміністративна панель реалізована у вигляді окремого інтерфейсу з розширеним функціоналом. Вона надає можливість переглядати список усіх оголошень, редагувати або видаляти їх, керувати категоріями, містами та

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

регіонами, а також здійснювати модерацію контенту[19]. Для забезпечення зручності адміністрування інтерфейс побудовано із застосуванням Bootstrap-компонентів. Панель адміністрування містить окремі сторінки для управління користувачами, категоріями, оголошеннями та локаціями.

У процесі реалізації функціоналу особлива увага приділялася забезпеченню безпеки. Усі запити до бази даних виконуються з використанням підготовлених виразів PDO Prepared Statements, що запобігає SQL-ін'єкціям. Вхідні дані проходять валідацію та санітизацію для уникнення XSS-атак. Паролі користувачів зберігаються виключно у вигляді хешів. Для захисту форм використано механізм CSRF-токенів, який дозволяє запобігати несанкціонованому виконанню HTTP-запитів від імені користувача.

З метою підвищення зручності користування реалізовано адаптивний інтерфейс, який автоматично підлаштовується під розмір екрана пристрою. Це забезпечує коректне відображення як на персональних комп'ютерах, так і на мобільних пристроях. Для цього використовувались адаптивні сітки Bootstrap, flex-компоненти та медіа-запити CSS.

Тестування функціоналу проводилося в локальному середовищі WampServer. Було перевірено коректність роботи всіх модулів, обробку виняткових ситуацій та взаємодію з базою даних. Окремо тестувалися процеси реєстрації та авторизації, створення оголошень, додавання відгуків, фільтрації та роботи адміністративної панелі. Також проводилася перевірка прав доступу для різних ролей користувачів.

Результати тестування підтвердили відповідність реалізованого функціоналу визначеним вимогам. Веб-застосунок коректно обробляє запити користувачів, забезпечує стабільну взаємодію з базою даних та підтримує багаторівневу систему доступу. Використання архітектури MVC дозволило створити структурований програмний код, придатний до подальшого розвитку та масштабування.

Таким чином, реалізація функціоналу веб-застосунку каталогу забезпечує повний цикл роботи з оголошеннями, категоріями, містами, регіонами та

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

відгуками, підтримує багаторівневу систему доступу та відповідає сучасним вимогам до безпеки й адаптивності веб-систем. Створений застосунок може бути використаний як ефективний інструмент цифрового представництва малого бізнесу та основа для подальшого розвитку і масштабування.

Висновки до другого розділу

У другому розділі було розглянуто процес проєктування та реалізації веб-застосунку каталогу малого бізнесу. Насамперед визначено основні варіанти використання системи, ролі користувачів та їхні функціональні можливості. Було встановлено, що система повинна підтримувати роботу трьох основних категорій користувачів: адміністратора, менеджера та звичайного користувача. Для кожної ролі визначено окремий набір прав доступу, що дозволяє забезпечити контроль дій у системі та запобігти несанкціонованому використанню адміністративних функцій.

У межах проєктування структури системи було описано основні прецеденти роботи веб-застосунку, зокрема авторизацію, реєстрацію, перегляд каталогу, створення та редагування оголошень, додавання відгуків, керування категоріями та користувачами. Також було визначено основні класи системи, їх призначення, методи та взаємозв'язки. Це дозволило сформулювати логічну модель майбутнього програмного продукту та підготувати основу для реалізації його функціональних модулів.

Окрему увагу було приділено розробці бази даних. Для зберігання інформації обрано реляційну СУБД MySQL, яка забезпечує стабільну роботу з даними, підтримку зовнішніх ключів та ефективне виконання запитів. У процесі розробки було визначено основні таблиці бази даних: users, categories, regions, cities, listings та reviews. Між таблицями встановлено логічні зв'язки типу «один до багатьох», що дозволяє забезпечити цілісність даних і коректну взаємодію між користувачами, оголошеннями, категоріями, локаціями та відгуками.

Також у другому розділі було розглянуто алгоритми роботи веб-застосунку. Описано загальну логіку обробки HTTP-запитів, взаємодію контролерів, моделей і представлень відповідно до архітектури MVC. Особливу

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

увагу приділено алгоритмам авторизації користувачів, створення оголошень, обробки відгуків, перевірки прав доступу та взаємодії з базою даних. Використання діаграми активності дозволило наочно подати послідовність дій під час входу користувача в систему.

У процесі реалізації функціоналу було створено основні модулі веб-застосунку: модуль автентифікації та авторизації, модуль управління оголошеннями, модуль категорій, міст і регіонів, модуль відгуків і рейтингу, а також адміністративну панель. Реалізований функціонал забезпечує повний цикл роботи з оголошеннями, підтримує пошук і фільтрацію компаній, дозволяє залишати відгуки та формувати середній рейтинг компаній. Крім того, у системі реалізовано рольову модель доступу, що дозволяє розмежувати можливості адміністратора, менеджера та користувача.

Важливою складовою реалізації стала інформаційна безпека. У веб-застосунку використано хешування паролів, перевірку вхідних даних, підготовлені SQL-запити для захисту від SQL-ін'єкцій, санітизацію даних для запобігання XSS-атакам, а також CSRF-токени для захисту форм. Це підвищує надійність системи та забезпечує безпечну взаємодію користувачів із веб-застосунком.

Таким чином, у другому розділі було сформовано структурну, логічну та програмну основу веб-застосунку каталогу малого бізнесу. Розроблена база даних, описані алгоритми роботи та реалізовані функціональні модулі створюють підґрунтя для подальшого аналізу користувацького інтерфейсу, адміністративної частини системи та тестування працездатності веб-застосунку.

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБ-ЗАСТОСУНКОМ КАТАЛОГУ

3.1 Користувацький інтерфейс

Користувацький інтерфейс веб-застосунку каталогу є одним із ключових компонентів системи, оскільки саме через нього здійснюється взаємодія користувача з функціональними можливостями застосунку. Під час розроблення інтерфейсу було враховано принципи зручності використання (usability), адаптивності, логічної структурованості елементів та мінімізації когнітивного навантаження на користувача. Інтерфейс реалізовано з використанням HTML, CSS та фреймворку Bootstrap, що забезпечує сучасний вигляд сторінок і коректне відображення на пристроях із різною роздільною здатністю екрана[20]. Адаптивна верстка дозволяє автоматично змінювати розташування елементів залежно від ширини екрану, що забезпечує комфортне користування як на персональних комп'ютерах, так і на мобільних пристроях.

Головна сторінка веб-застосунку містить перелік компаній, представлених у вигляді карток. Кожна картка відображає назву компанії, категорію діяльності, короткий опис, середній рейтинг та кнопку переходу до детальної інформації. Візуальне виділення категорій за допомогою м'яких кольорових акцентів покращує сприйняття інформації та сприяє швидкій орієнтації користувача, рис.3.1.

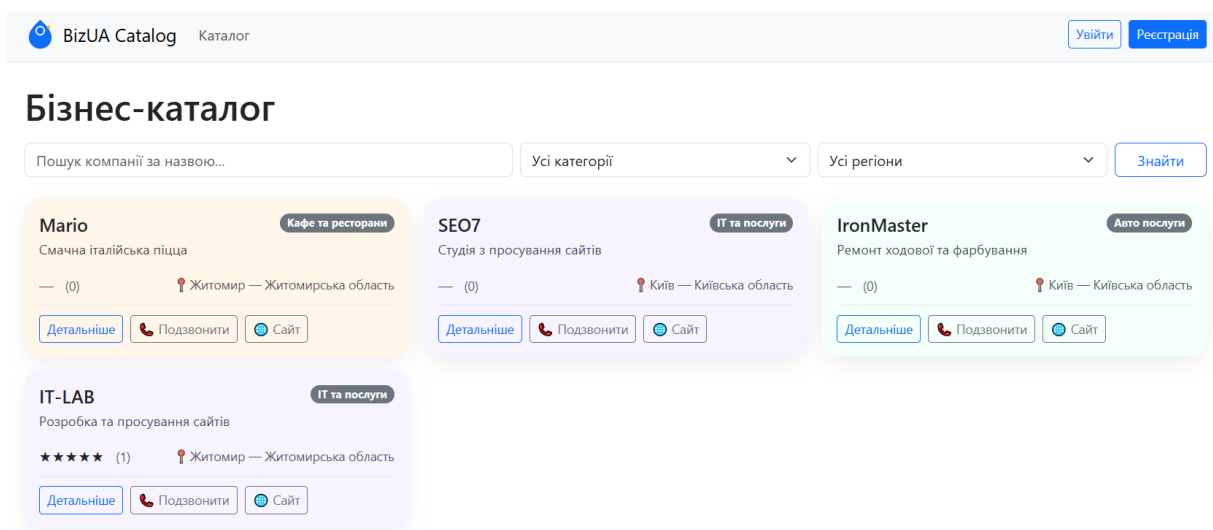


Рис.3.1. Інтерфейс програми

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

У верхній частині сторінки розташовано навігаційне меню, яке забезпечує доступ до основних розділів системи: головної сторінки, форми додавання компанії, сторінки входу та реєстрації, рис.3.2.

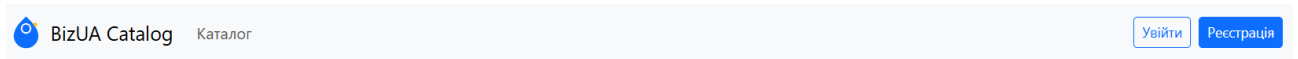


Рис.3.2. Навігаційне меню

Для авторизованих користувачів у меню відображається ім'я користувача та доступ до особистого кабінету або адміністративної панелі залежно від ролі, рис.3.3.

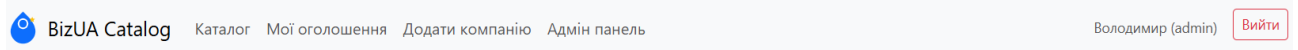


Рис.3.3. Приклад панелі адміна

Після успішної реєстрації та авторизації користувача відбувається динамічна зміна навігаційної панелі веб-застосунку. Якщо до входу в систему у верхній частині сторінки відображаються кнопки «Увійти» та «Реєстрація», то після створення облікового запису інтерфейс адаптується відповідно до статусу авторизованого користувача. У навігаційній панелі з'являються додаткові пункти меню, які надають доступ до функціональних можливостей системи. Зокрема, користувач отримує доступ до розділів «Мої оголошення» та «Додати компанію». Це забезпечує швидкий перехід до створення нового запису або управління вже доданими компаніями. Такий підхід відповідає принципу персоналізації інтерфейсу та покращує зручність взаємодії з системою.

У правій частині панелі відображається ім'я користувача із зазначенням його ролі (наприклад, Admin або User), а також кнопка «Вийти», що забезпечує завершення сесії. Відображення ролі користувача є важливим елементом інтерфейсу, оскільки воно візуально підтверджує рівень доступу до функціоналу системи. У випадку входу адміністратора або менеджера до навігаційного меню додається пункт «Адмін», який надає доступ до адміністративної панелі. Ця панель містить інструменти управління контентом, включаючи редагування та видалення оголошень, а також створення нових категорій діяльності.

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

Таким чином, після реєстрації користувач бачить оновлену навігаційну структуру, яка відображає його повноваження в системі. Інтерфейс автоматично адаптується відповідно до ролі, забезпечуючи персоналізований доступ до функціоналу. Така реалізація підвищує ергономічність системи та створює зрозумілу логіку взаємодії з веб-застосунком.

Важливим елементом інтерфейсу є форма пошуку, що дозволяє здійснювати пошук компаній за назвою. Пошукове поле доповнюється випадаючим списком категорій, що реалізує функцію фільтрації. Результати відображаються миттєво після виконання запиту, що забезпечує оперативність взаємодії, рис.3.4.

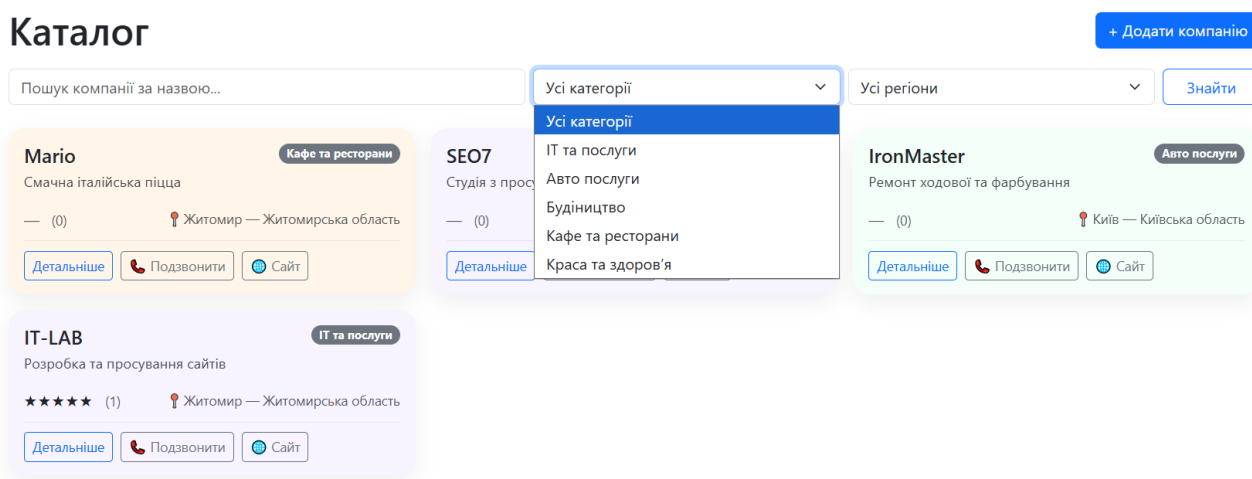


Рис.3.4. Панель пошуку компаній з робочим фільтром

Пошукове поле дозволяє здійснювати текстовий пошук за назвою компанії. Після введення запиту та натискання кнопки «Знайти» система формує SQL-запит із використанням оператора LIKE або механізму FULLTEXT-пошуку, залежно від конфігурації. У результаті відображаються лише ті записи, які відповідають введеному критерію. Випадаючий список категорій реалізує додатковий механізм фільтрації. Користувач може обрати одну із запропонованих категорій, наприклад «ІТ та послуги», «Авто послуги», «Будівництво», «Кафе та ресторани», «Краса та здоров'я» тощо. Після вибору категорії система автоматично обмежує вибірку компаній лише відповідною сферою діяльності. Якщо обрано пункт «Усі категорії», фільтрація не застосовується.

Фільтр реалізовано через передачу параметра `category_id` у GET-запиті. Контролер отримує значення параметра та формує відповідну умову у SQL-запиті (`WHERE category_id = ...`)[21]. Таким чином забезпечується динамічне оновлення списку компаній без зміни структури сторінки. Відображення результатів здійснюється у вигляді адаптивних карток компаній. Кожна картка містить назву підприємства, короткий опис послуг, позначку категорії, середній рейтинг із графічним відображенням зірок, кількість відгуків, а також місто розташування. Використання м'яких кольорових відтінків для різних категорій покращує візуальну диференціацію записів і підвищує зручність сприйняття інформації.

У випадку, якщо жоден запис не відповідає заданим параметрам, система відображає повідомлення «Нічого не знайдено». Такий підхід забезпечує зворотний зв'язок із користувачем та запобігає виникненню непорозумінь щодо відсутності результатів. Поєднання текстового пошуку та категоріальної фільтрації формує гнучкий механізм навігації в межах каталогу. Реалізований функціонал дозволяє швидко знаходити релевантні компанії, що особливо важливо для користувачів, які здійснюють пошук послуг у конкретній сфері діяльності.

Таким чином, інтерфейс пошуку та фільтрації є невід'ємною складовою користувацького досвіду, оскільки забезпечує ефективну взаємодію з інформаційним наповненням веб-застосунку та сприяє підвищенню його практичної цінності. Сторінка детального перегляду компанії містить розширену інформацію про підприємство, включаючи опис діяльності, контактні дані, адресу, веб-сайт та рейтинг. Відгуки користувачів відображаються у структурованому вигляді із зазначенням оцінки та текстового коментаря. Такий формат забезпечує прозорість і довіру до інформації, рис.3.5.

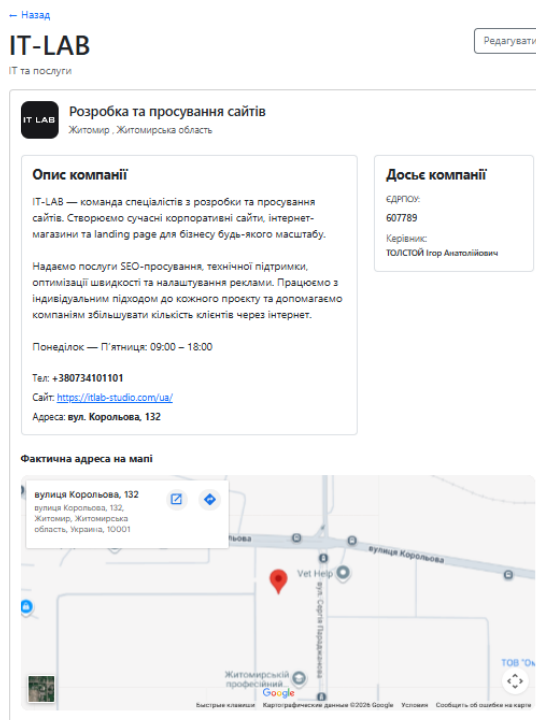


Рис.3.5. Сторінка детального перегляду компанії

На рисунку 3.5 зображено сторінку детального перегляду компанії в межах веб-застосунку бізнес-каталогу. У верхній частині сторінки відображається назва компанії **IT-LAB**, що подається великим шрифтом для візуального акцентування. Під назвою зазначено категорію діяльності («ІТ та послуги»), що дозволяє користувачу швидко ідентифікувати сферу послуг підприємства[22].

Нижче представлено інформаційний блок із логотипом компанії та коротким переліком основних послуг, зокрема розробка та просування сайтів. Така структурована подача інформації забезпечує швидке ознайомлення з профілем діяльності компанії. У лівій частині сторінки знаходиться блок «Опис компанії» опис підприємства, який містить деталі щодо якості послуг, умов обслуговування, графіку роботи та додаткові переваги. Це дозволяє користувачу отримати повне уявлення про рівень сервісу.

Контактна інформація подається у зручному форматі та включає номер телефону, адресу веб-сайту, місто розташування. Дані структуровано в окремих рядках, що покращує читабельність і спрощує доступ до необхідної інформації. У правій частині сторінки розташовано блок «Досьє компанії». Він містить інформацію код ЄДРПОУ та ініціали керівника компанії. Даний блок

					ІПЗ.КР.Б – 121-26-ІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

призначений для швидкого перегляду ключових відомостей про організацію користувачами веб-сервісу. Також на нижній частині знаходиться блок «Фактична адреса на мапі». Тут показує розташування на GoogleMaps, щоб користувач побачив де знаходиться данна компанія.

Відгуки

Коментар

Рекомендую цю компанію для створення вашого продукту.

Рейтинг

5 ★

[Зберегти відгук](#) [Видалити мій відгук](#)

Тільки зареєстровані користувачі можуть залишати відгуки. 1 відгук на 1 оголошення.

Володимир 5 ★

Рекомендую цю компанію для створення вашого продукту.

2026-05-10 13:02:01 [Видалити](#)

Рис.3.6. Блок для залишення відгуку

На рис.3.6 блок для залишення відгуку під оголошення компанії. У цьому блоці є попередження про те, що залишати відгуки можуть лише зареєстровані користувачі та що дозволяється один відгук на одне оголошення. Це реалізує бізнес-правило системи та запобігає дублюванню оцінок. Нижче відображається вже залишений відгук із зазначенням імені користувача (у даному випадку Admin), тексту коментаря та дати публікації. Для користувача, який є автором відгуку або має адміністративні права, передбачена можливість видалення запису. Сторінка детального перегляду компанії поєднує інформаційний блок, контактні дані та інтерактивний модуль відгуків, що забезпечує комплексне представлення підприємства та створює прозорий механізм оцінювання якості послуг. Представлена структура сприяє формуванню довіри до інформації та підвищує ефективність взаємодії користувачів із веб-застосунком[23].

Форма додавання або редагування компанії побудована з використанням стандартних елементів введення: текстових полів, випадаючих списків та кнопок підтвердження. Перед відправленням даних система перевіряє коректність заповнення обов'язкових полів. У разі виникнення помилки користувач отримує відповідне повідомлення із поясненням, рис.3.7.

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

Додати компанію

Назва компанії

Категорія

ІТ та послуги

Заголовок оголошення

Напр.: Ремонт квартир під ключ у Львові

Опис

Опишіть послуги, досвід, графік роботи...

Телефон

+380...

Сайт

https://...

Місто

Оберіть місто

ЄДРПОУ

6077759

Керівник

Сидоренко Василь Дмитрович

Адреса

Вулиця, будинок...

Логотип (jpg/png/webp)

Виберіть файл

Файл не вибран

Опублікувати

Скасувати

Рис.3.7. Форма додавання компанії

Реєстрація та авторизація реалізовані через окремі сторінки з мінімалістичним дизайном. Форми містять лише необхідні поля, що спрощує процес створення облікового запису та входу до системи. Повідомлення про помилки або успішні дії відображаються у вигляді сповіщень (alert-компонентів Bootstrap), рис.3.8.

← → ↻ 📄

localhost/business-catalog/public/login

🔍 ☆ 📱 🛡️ 🗑️ 🧑

📺 YouTube

📖 Learn | Mate acad...

📄 ClickUp

🌐 Canva

📦 Node

👤 My Profile

🔗 Тв Підтримка Link...

📁 Все закладки

BizUA Catalog

Каталог

Увійти

Реєстрація

Невірний email або пароль.

Вхід

Email

Admin@local.com

Пароль

Volodia1234

👁️

Увійти

Немає акаунта? [Зареєструватися](#)

Рис.3.8. Форма реєстрація та авторизація

Особливу увагу приділено доступності інтерфейсу. Контрастність кольорів, розмір шрифту та відступи між елементами відповідають базовим принципам ергономіки. Кнопки мають достатній розмір для зручного натискання на сенсорних екранах. Інтерфейс підтримує логічну послідовність дій

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

користувача. Після виконання операції додавання або редагування система автоматично перенаправляє користувача на сторінку перегляду результату. Це забезпечує зрозумілий сценарій взаємодії та підвищує загальну зручність користування.

Для покращення візуального сприйняття інформації використано іконографіку та рейтингову систему у вигляді зірок. Середній рейтинг компанії відображається числовим значенням та графічним індикатором, що дозволяє швидко оцінити якість послуг.

Інтерфейс адміністративної частини відрізняється розширеним функціоналом, проте зберігає єдину стилістику оформлення. Це забезпечує цілісність дизайну та спрощує перехід між різними ролями користувачів.

Важливим аспектом реалізації є обробка виняткових ситуацій на рівні інтерфейсу. У випадку відсутності результатів пошуку або порожнього каталогу система відображає відповідне повідомлення, що інформує користувача про стан даних. Користувацький інтерфейс веб-застосунку бізнес-каталогу реалізовано з урахуванням сучасних вимог до зручності, адаптивності та інтуїтивності. Він забезпечує комфортну взаємодію користувачів із системою та сприяє ефективному використанню функціональних можливостей веб-застосунку.

3.2 Інтерфейс адміністратора та менеджера

У межах розробленого веб-застосунку реалізовано окремі інтерфейси для адміністратора та менеджера, які формуються динамічно залежно від ролі авторизованого користувача. Такий підхід дозволяє забезпечити розмежування прав доступу та надати кожній категорії користувачів відповідний функціональний інструментарій. Інтерфейс адміністратора є найбільш розширеним і надає повний контроль над вмістом системи. Після авторизації користувача з роллю admin у навігаційній панелі з'являється пункт «Адмін», що відкриває доступ до адміністративної частини застосунку.

Адміністративна панель є центральним інструментом керування веб-застосунком. Вона призначена для контролю основних інформаційних об'єктів системи: оголошень, категорій, міст, регіонів і користувачів. Усі розділи

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

адміністративної частини згруповані у вигляді окремих карток або сторінок, що спрощує навігацію та дає змогу швидко перейти до потрібного модуля керування. Така організація інтерфейсу дозволяє адміністратору оперативно виконувати службові дії без необхідності переглядати зайві розділи системи.

Адміністратор має можливість переглядати всі оголошення незалежно від автора, редагувати та видаляти будь-які компанії, створювати, редагувати та видаляти категорії діяльності, керувати містами й регіонами, здійснювати модерацію контенту, а також видаляти некоректні або спам-відгуки. Крім того, адміністратор має доступ до управління користувачами системи. Він може переглядати список зареєстрованих користувачів, змінювати їхні ролі, створювати нові облікові записи та видаляти користувачів, якщо це не суперечить логіці безпеки системи. Наприклад, у застосунку передбачено обмеження, яке не дозволяє адміністратору видалити власний обліковий запис.

Інтерфейс адміністративної панелі побудований у єдиній стилістиці із загальним дизайном системи, що забезпечує візуальну цілісність веб-застосунку. Основні дії реалізовані у вигляді кнопок керування, розташованих поруч із відповідними записами. Наприклад, біля кожної категорії, міста або оголошення відображаються елементи для редагування, збереження змін або видалення. Такий підхід спрощує управління контентом і зменшує кількість зайвих переходів між сторінками. Адміністратор також бачить у верхній частині сторінки власне ім'я та роль, що підтверджує його повноваження в системі.

Окрему роль в адміністративному інтерфейсі відіграють повідомлення про результат виконання дій. Після створення, редагування або видалення запису система відображає відповідне інформаційне повідомлення. Наприклад, після успішного додавання категорії або міста користувач бачить повідомлення про успішне виконання операції. Якщо дія не може бути виконана, наприклад через те, що категорія вже використовується в оголошеннях, система виводить повідомлення про помилку. Це забезпечує зрозумілий зворотний зв'язок і допомагає адміністратору контролювати стан виконаних операцій.

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

Доступ до адміністративних функцій додатково контролюється на рівні серверної логіки, що запобігає несанкціонованому використанню адміністративних інструментів. Навіть якщо звичайний користувач спробує вручну перейти за адресою адміністративної сторінки, система перевірить його роль і заборонить доступ у разі відсутності відповідних прав. Таким чином, захист реалізовано не лише на рівні інтерфейсу, а й на рівні контролерів, що підвищує загальну безпеку веб-застосунку.

Інтерфейс менеджера реалізований із обмеженим, але розширеним порівняно зі звичайним користувачем функціоналом. Після входу користувача з роллю manager у навігаційній панелі відображаються пункти «Мої оголошення», «Додати компанію» та, за наявності відповідних прав, можливість управління категоріями, містами й регіонами. Менеджер виконує роль проміжного адміністратора, який може допомагати підтримувати актуальність інформації в каталозі, але не має повного контролю над усією системою.

Менеджер має можливість створювати нові оголошення, редагувати власні записи, керувати категоріями та локаціями в межах визначених правил доступу. Його функціональність орієнтована на підтримку структури каталогу та модерацію інформації без доступу до критично важливих адміністративних можливостей. На відміну від адміністратора, менеджер не має права керувати користувачами системи, змінювати їхні ролі або видаляти облікові записи. Також рівень його доступу обмежується відповідно до принципу мінімально необхідних привілеїв.

Принцип мінімально необхідних привілеїв означає, що кожен користувач отримує лише ті права, які потрібні йому для виконання визначених функцій. Такий підхід зменшує ризик випадкового або навмисного пошкодження даних. Наприклад, менеджер може брати участь у підтримці актуальності каталогу, але не може впливати на облікові записи інших користувачів. Адміністратор, у свою чергу, має повний доступ, оскільки відповідає за загальне керування системою.

У верхній частині інтерфейсу менеджера відображається повідомлення про успішний вхід у систему, що підтверджує активність сесії. Поруч із ім'ям

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

користувача зазначено його роль, а також розміщено кнопку «Вийти», яка забезпечує завершення роботи із застосунком. Інтерфейс менеджера візуально не відрізняється від загального користувацького інтерфейсу, проте містить додаткові елементи керування, що з’являються відповідно до ролі. Такий підхід дозволяє зберегти єдину концепцію дизайну та водночас забезпечити функціональну диференціацію доступу, рис. 3.9.

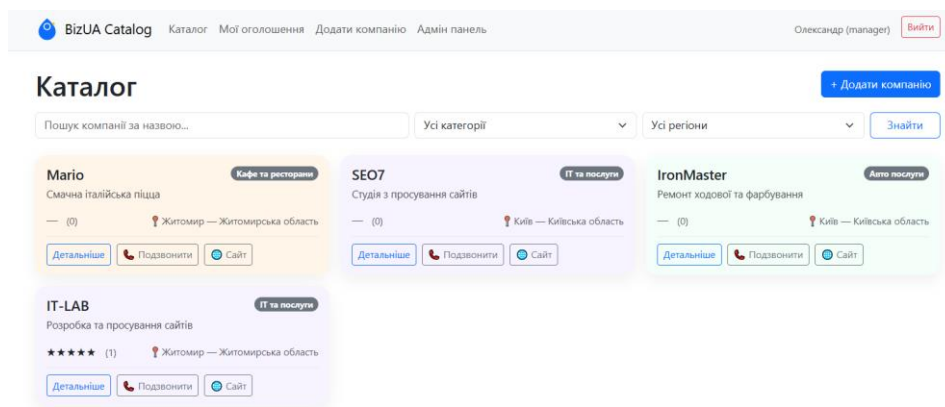


Рис.3.9. Інтерфейс менеджера

Важливою особливістю реалізації інтерфейсів адміністратора та менеджера є те, що всі службові дії виконуються через форми із захистом від CSRF-атак. Це означає, що кожна операція створення, редагування або видалення запису супроводжується перевіркою спеціального токена. Такий механізм запобігає виконанню небажаних запитів від імені авторизованого користувача та підвищує рівень інформаційної безпеки системи.

Також у системі передбачено підтвердження небезпечних дій, зокрема видалення записів. Перед видаленням категорії, міста, оголошення або користувача система може відображати діалог підтвердження. Це знижує ризик випадкового видалення важливих даних і робить роботу адміністратора або менеджера більш контрольованою. Такий підхід є важливим для веб-застосунків, у яких користувачі працюють з великою кількістю інформаційних записів.

Таким чином, реалізація окремих інтерфейсів адміністратора та менеджера забезпечує гнучке управління веб-застосунком, підтримує багаторівневу модель доступу та підвищує безпеку системи. Розмежування прав реалізовано як на рівні

інтерфейсу, так і на рівні серверної логіки, що гарантує коректність функціонування та запобігає несанкціонованим діям. Завдяки цьому веб-застосунок залишається зручним для звичайних користувачів і водночас надає достатній набір інструментів для ефективного адміністрування каталогу.

3.3 Особливості роботи з оголошеннями, відгуками та рейтингом

Функціонування веб-застосунку каталогу базується на взаємодії трьох ключових інформаційних об'єктів: оголошень (компаній), відгуків користувачів та системи рейтингового оцінювання. Саме ці компоненти формують основну цінність сервісу, забезпечуючи інформаційну наповненість каталогу та механізм зворотного зв'язку[24]. Оголошення є центральним елементом системи та відображають профіль конкретної компанії або підприємця. Створення оголошення здійснюється через спеціальну форму, яка містить обов'язкові та додаткові поля: назву компанії, заголовок, опис діяльності, категорію, контактні дані, адресу, місто та посилання на веб-сайт. Перед збереженням дані проходять серверну валідацію, що забезпечує коректність та повноту інформації. Після успішного створення оголошення воно зберігається у базі даних та відображається на головній сторінці каталогу у вигляді картки. Кожна картка містить основні відомості про компанію, середній рейтинг, кількість відгуків та кнопки швидкого доступу до детального перегляду, дзвінка або переходу на сайт. Редагування оголошення можливе лише для його власника або адміністратора. Перед відкриттям форми редагування система перевіряє право доступу, що реалізує принцип контролю повноважень. У разі видалення оголошення автоматично видаляються пов'язані з ним відгуки відповідно до встановлених правил цілісності даних.

Система відгуків реалізована як механізм оцінювання якості послуг компаній. Залишати відгуки можуть лише авторизовані користувачі. Для кожного оголошення передбачено обмеження — один користувач може залишити лише один відгук. Це запобігає штучному завищенню або заниженню рейтингу. Форма відгуку складається з двох основних компонентів: вибору рейтингу за п'ятибальною шкалою та текстового коментаря. Після збереження

відгук відображається у списку з указанням імені автора, дати публікації та тексту повідомлення. Якщо користувач уже залишав відгук, система дозволяє його редагування або видалення. Адміністратор має додаткові повноваження щодо модерації відгуків. У разі порушення правил користування або виявлення некоректного змісту відгук може бути видалений. Такий механізм забезпечує підтримку якості контенту та захист репутації компаній, рис.3.9.

Рис.3.9. Форма рейтингу

Рейтинг компанії формується на основі середнього арифметичного значення всіх отриманих оцінок. Після кожного додавання або редагування відгуку система виконує перерахунок середнього рейтингу. Для цього використовується агрегатна функція AVG у запиті до бази даних. Отриманий показник відображається як числове значення та графічний індикатор у вигляді зірок. Поруч із рейтингом вказується кількість відгуків, що дозволяє користувачеві оцінити репрезентативність оцінки. Такий підхід забезпечує прозорість та об'єктивність системи оцінювання. Особливістю реалізації є тісна інтеграція оголошень і відгуків. Під час перегляду сторінки компанії користувач бачить повну інформацію разом із блоком рейтингу та відгуків. Зміни, внесені до відгуків, миттєво відображаються в загальному рейтингу компанії, що забезпечує актуальність даних.

Система також підтримує логічну послідовність дій: після створення або редагування оголошення користувач автоматично перенаправляється на сторінку детального перегляду. Після збереження відгуку сторінка оновлюється з відображенням нового рейтингу, рис. 3.10.

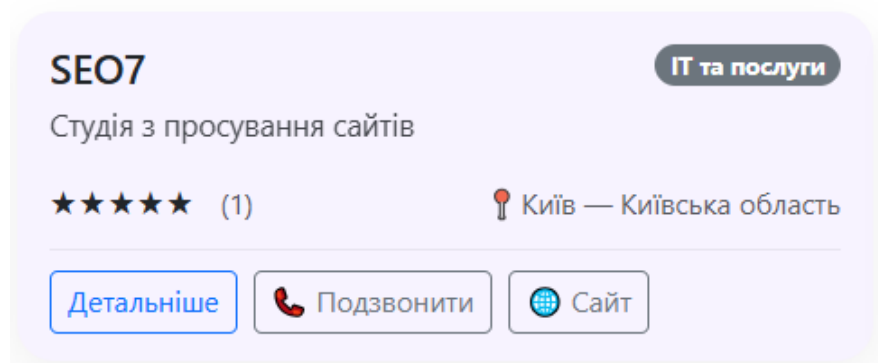


Рис.3.10. Приклад компанії з одним відгуком і 5 зірками

Для запобігання некоректному введенню даних реалізовано перевірку допустимого діапазону рейтингу (від 1 до 5). Усі текстові дані проходять санітизацію для запобігання XSS-атакам. Запити до бази даних виконуються із використанням підготовлених виразів, що виключає можливість SQL-ін'єкцій. Таким чином, особливості роботи з оголошеннями, відгуками та рейтингом формують комплексний механізм представлення інформації та зворотного зв'язку в межах веб-застосунку. Реалізована система забезпечує прозорість оцінювання, контроль доступу та актуальність даних, що підвищує довіру користувачів і ефективність функціонування каталогу.

3.4 Тестування веб-застосунку

Тестування веб-застосунку каталогу проводилося з метою перевірки коректності реалізації функціональних можливостей, стабільності роботи системи, відповідності визначеним вимогам та забезпечення інформаційної безпеки. Комплексний підхід до тестування є невід'ємною складовою процесу розробки якісного програмного продукту, оскільки дозволяє виявити та усунути помилки до моменту введення системи в промислову експлуатацію. Своєчасне виявлення дефектів на етапі тестування суттєво знижує вартість їх усунення порівняно з виправленням помилок, виявлених вже після розгортання системи у виробничому середовищі. Процес тестування здійснювався у локальному середовищі розробки на базі WampServer із використанням браузера Google Chrome, що забезпечило контрольоване середовище для відтворення

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

різноманітних сценаріїв використання системи та перевірки її реакції на нестандартні вхідні дані.

На першому етапі було виконано функціональне тестування, яке передбачало перевірку працездатності основних модулів системи. Функціональне тестування є базовим видом перевірки програмного забезпечення, спрямованим на підтвердження того, що кожна функція застосунку працює відповідно до визначених вимог і специфікацій. Зокрема, перевірено процеси реєстрації та авторизації користувачів, створення та редагування оголошень, додавання та видалення відгуків, а також механізм фільтрації й пошуку компаній[25]. Тестові сценарії охоплювали як типові шляхи використання системи, так і граничні випадки, що дозволило всебічно оцінити поведінку застосунку в різноманітних ситуаціях. Результати тестування підтвердили коректність обробки введених даних і правильність формування відповідей сервера, а виявлені в процесі тестування незначні невідповідності були оперативно усунені ще до завершення етапу розробки.

Окрему увагу приділено тестуванню системи розмежування прав доступу. Перевірка коректності реалізації механізму авторизації є критично важливим аспектом тестування будь-якої багатокористувацької системи, оскільки вразливості у цьому компоненті можуть призвести до серйозних порушень безпеки та компрометації конфіденційних даних. Було перевірено можливість доступу до адміністративної панелі лише для користувачів із відповідною роллю, а також коректність обмеження доступу до функцій редагування та видалення оголошень для користувачів, які не є їх власниками. Тестування проводилося шляхом спроб отримати доступ до захищених ресурсів з облікових записів із різними рівнями повноважень — від незареєстрованого відвідувача до звичайного користувача та менеджера[26]. Спроби доступу до адміністративних функцій без належних повноважень завершувалися поверненням помилки доступу з відповідним кодом HTTP 403, що свідчить про правильну реалізацію механізму контролю прав і коректне реагування системи на несанкціоновані запити.

					ІІЗ.КР.Б – 121-26-ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

Під час тестування модуля оголошень перевірено повний цикл CRUD-операцій. Перевірка повного циклу операцій зі створення, читання, оновлення та видалення записів є стандартною практикою тестування модулів, що взаємодіють із базою даних, оскільки дозволяє підтвердити коректність усіх можливих сценаріїв роботи з даними. Створення нового запису супроводжувалося його коректним відображенням у каталозі з усіма введеними атрибутами, включаючи назву компанії, опис, категорію та контактну інформацію. Редагування оголошення призводило до миттєвого оновлення інформації на сторінці детального перегляду, що підтверджує коректну роботу механізму передачі даних між формою редагування та базою даних. Видалення оголошення призводило до автоматичного видалення пов'язаних відгуків відповідно до встановлених правил каскадного видалення у базі даних, що підтверджує коректність реалізації обмежень цілісності на рівні фізичної моделі даних.

Тестування системи відгуків включало перевірку обмеження "один користувач — один відгук". Коректна реалізація цього обмеження є важливою умовою достовірності рейтингової системи, оскільки можливість залишати кількох відгуків від одного користувача відкриває шляхи для маніпуляцій із репутаційними оцінками компаній. Під час повторної спроби залишити відгук для тієї самої компанії система коректно виконувала оновлення існуючого запису замість створення нового, зберігаючи при цьому всі попередні налаштування та оновлюючи лише змінені поля. Також було ретельно перевірено правильність обчислення середнього рейтингу після додавання або зміни оцінки, включаючи граничні випадки — наприклад, ситуацію з одним відгуком або максимальною та мінімальною можливими оцінками. Результати перевірки підтвердили, що агрегований рейтинг компанії коректно перераховується у режимі реального часу після кожної зміни оцінки, забезпечуючи актуальність відображуваної інформації для всіх користувачів платформи[27].

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

Проведено тестування механізму пошуку та фільтрації. Ефективність пошукового функціоналу є одним із ключових показників якості каталогу, оскільки саме через пошук відбувається основна взаємодія кінцевого споживача з платформою при пошуку потрібних послуг або компаній. Введення ключових слів у пошуковому рядку забезпечувало відображення релевантних результатів, причому система коректно обробляла як точні збіги, так і часткові відповідності в назвах та описах компаній. Вибір категорії у випадяючому списку обмежував вибірку компаній відповідно до заданого параметра, а комбінування текстового пошуку з фільтрацією за категорією дозволяло суттєво звузити результати до найбільш релевантних записів. У випадку відсутності результатів за введеними критеріями система відображала інформативне повідомлення, що надає користувачу зрозумілий зворотній зв'язок і допомагає скоригувати параметри пошуку для отримання кращих результатів.

Тестування безпеки включало перевірку стійкості системи до SQL-ін'єкцій шляхом введення спеціальних символів та конструкцій мови SQL у поля форм. SQL-ін'єкція є однією з найбільш поширених та небезпечних вразливостей веб-застосунків, тому перевірка захисту від цього типу атак є обов'язковою складовою комплексного тестування безпеки. Завдяки використанню підготовлених запитів із параметризованими значеннями некоректні SQL-конструкції не впливали на структуру або вміст бази даних, підтверджуючи надійність реалізованого захисного механізму. Додатково перевірено коректність обробки HTML-тегів та JavaScript-коду у текстових полях для запобігання XSS-атакам, при цьому система коректно екранувала потенційно небезпечний контент перед його відображенням на сторінці. Також було проведено перевірку захисту від підробки міжсайтових запитів шляхом спроб відправки форм без валідного токена захисту, що підтвердило коректну реалізацію відповідного механізму безпеки[28].

Під час навантажувального тестування було перевірено швидкодію системи при збільшенні кількості записів у базі даних. Навантажувальне тестування дозволяє оцінити поведінку системи в умовах, наближених до

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

реальних сценаріїв експлуатації, та виявити потенційні вузькі місця у продуктивності ще до введення застосунку в промислову експлуатацію. До тестових таблиць було додано значну кількість тестових записів, після чого проводилися заміри часу відповіді сервера при виконанні типових запитів — пошуку, фільтрації та отримання деталей оголошення. Використання індексів у таблицях забезпечило стабільну швидкість пошуку та фільтрації навіть за умови зростання обсягу даних, що підтверджує коректність прийнятих рішень щодо оптимізації структури бази даних. Час відповіді сервера залишався в межах допустимих значень, що відповідає визначеним нефункціональним вимогам до продуктивності системи.

Також проведено тестування адаптивності інтерфейсу[29] на різних типах пристроїв та розмірах екранів. Перевірка кросплатформної сумісності є невід'ємною складовою тестування сучасних веб-застосунків, враховуючи різноманіття пристроїв та браузерів, що використовуються сучасними інтернет-користувачами. Тестування проводилося з використанням інструментів емуляції мобільних пристроїв вбудованих у браузер Google Chrome, а також шляхом безпосередньої перевірки на реальних мобільних пристроях із різними розмірами екрана. Сторінки коректно відображалися на різних розмірах екранів, включаючи мобільні пристрої з невеликими дисплеями, планшети та широкоформатні монітори персональних комп'ютерів. Розташування елементів автоматично змінювалося відповідно до ширини екрану, навігаційне меню коректно трансформувалося у компактну форму на мобільних пристроях, а таблиці та форми адаптувалися до вузьких екранів без втрати функціональності, що підтверджує ефективність використання адаптивного фреймворку Bootstrap.

Для перевірки коректності роботи веб-застосунку було проведено функціональне тестування основних модулів системи. Тестування виконувалося шляхом реалізації типових сценаріїв взаємодії користувача із системою. Основні результати тестування наведені у таблицях 3.1–3.4.

					ІІЗ.КР.Б – 121-26-ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

Таблиця 3.1.

Тестування функціоналу авторизації користувача

№	Тестовий сценарій	Вхідні дані	Очікуваний результат	Фактичний результат
1	Вхід з правильними даними	email, пароль	Користувач успішно входить у систему	Успішний вхід
2	Вхід з неправильним паролем	email, неправильний пароль	Відображається повідомлення про помилку	Повідомлення про помилку
3	Вхід з неіснуючим email	неіснуючий email	Система повідомляє про помилку авторизації	Повідомлення відображено
4	Вхід без заповнення полів	порожні поля	Система не дозволяє виконати вхід	Повідомлення про помилку

Таблиця 3.2.

Тестування CRUD-операцій оголошень

№	Тестовий сценарій	Дія користувача	Очікуваний результат	Фактичний результат
1	Створення оголошення	Заповнення форми та збереження	Оголошення додається у каталог	Оголошення створено
2	Перегляд оголошення	Відкриття сторінки оголошення	Відображаються дані компанії	Дані відображено
3	Редагування оголошення	Зміна інформації	Дані оголошення оновлюються	Дані змінено
4	Видалення оголошення	Натискання кнопки видалення	Оголошення видаляється з каталогу	Оголошення видалено

Таблиця 3.3.

Тестування функціоналу відгуків

№	Тестовий сценарій	Вхідні дані	Очікуваний результат	Фактичний результат
1	Додавання відгуку	рейтинг + коментар	Відгук зберігається у базі даних	Відгук додано
2	Оновлення відгуку	новий рейтинг	Оновлюється існуючий запис	Відгук оновлено
3	Додавання другого відгуку	той самий користувач	Відгук оновлюється, а не створюється новий	Обмеження працює
4	Видалення відгуку	натискання кнопки видалення	Відгук видаляється	Відгук видалено

Таблиця 3.4.

Тестування пошуку та фільтрації

№	Тестовий сценарій	Вхідні дані	Очікуваний результат	Фактичний результат
1	Пошук за назвою	ключове слово	Відображаються релевантні компанії	Результати відображено
2	Пошук за категорією	вибір категорії	Показуються компанії вибраної категорії	Фільтр працює
3	Пошук за регіоном	вибір регіона	Показуються компанії вибраного регіона	Фільтр працює
4	Комбінований пошук	ключове слово + категорія + регіон	Відображаються релевантні результати	Результати коректні
5	Пошук без результатів	випадковий текст	Відображається повідомлення про відсутність результатів	Повідомлення відображено

Результати тестування засвідчили, що веб-застосунок[30] відповідає визначеним функціональним та нефункціональним вимогам. Усі ключові модулі працюють стабільно, обробка помилок реалізована коректно з відображенням зрозумілих повідомлень для користувача без розкриття чутливої технічної інформації, а механізми безпеки забезпечують надійний захист даних користувачів від актуальних загроз. Виявлені в процесі тестування незначні недоліки були оперативно усунені, після чого повторна перевірка підтвердила коректність внесених виправлень. Таким чином, проведене комплексне тестування підтвердило працездатність і надійність розробленого веб-застосунку каталогу, що дозволяє рекомендувати його до практичного використання та подальшого впровадження як ефективного інструменту цифрового представництва малого бізнесу в сучасному інформаційному середовищі.

Висновки до третього розділу

У третьому розділі було детально розглянуто особливості реалізації інтерфейсу та порядок роботи з веб-застосунком каталогу. Проаналізовано структуру користувацького інтерфейсу, механізми навігації, адаптивність дизайну та принципи організації взаємодії між користувачем і системою. У процесі дослідження встановлено, що реалізований інтерфейс відповідає

					ІІЗ.КР.Б – 121-26-ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

сучасним вимогам до веб-застосунків, зокрема забезпечує інтуїтивну зрозумілість, логічну послідовність дій та адаптивність для різних типів пристроїв. Використання фреймворку Bootstrap дозволило створити уніфікований дизайн і забезпечити коректне відображення елементів на екранах різної роздільної здатності.

Окрему увагу приділено реалізації інтерфейсів адміністратора та менеджера. Розмежування прав доступу відображається не лише на рівні серверної логіки, а й у структурі навігаційної панелі та доступних функціональних елементів. Такий підхід підвищує безпеку системи та забезпечує чітке розподілення ролей у межах веб-застосунку. Було проаналізовано особливості роботи з оголошеннями, відгуками та рейтинговою системою. Реалізований механізм додавання, редагування та видалення записів, а також автоматичний перерахунок середнього рейтингу забезпечують актуальність і достовірність інформації. Наявність системи відгуків формує довіру користувачів та підвищує практичну цінність каталогу.

Проведене тестування підтвердило стабільність функціонування всіх модулів системи, коректність обробки даних та відповідність визначеним вимогам. Перевірка механізмів безпеки, розмежування доступу та обробки помилок засвідчила надійність розробленого програмного рішення. Таким чином, у третьому розділі обґрунтовано ефективність реалізованого інтерфейсу та продемонстровано працездатність веб-застосунку каталогу. Отримані результати підтверджують готовність системи до практичного використання та створюють фундамент для формування загальних висновків бакалаврської роботи.

ВИСНОВКИ

У випускній кваліфікаційній роботі бакалавра розроблено та реалізовано веб-застосунок каталогу для малого бізнесу, що забезпечує цифрове представництво компаній, механізм пошуку послуг та систему зворотного зв'язку у вигляді відгуків і рейтингу. Актуальність теми зумовлена зростанням ролі онлайн-платформ у розвитку малого підприємництва та необхідністю створення доступних інструментів для представлення бізнесу в мережі Інтернет. У першому розділі здійснено аналіз інформаційних технологій у сфері каталогів, розглянуто існуючі програмні рішення та визначено їх переваги й недоліки. На основі проведеного аналізу обґрунтовано вибір архітектури веб-застосунку, орієнтованої на клієнт-серверну модель та принципи MVC. Це дозволило сформулювати концептуальну основу для подальшого проєктування системи.

У другому розділі виконано проєктування веб-застосунку, визначено функціональні та нефункціональні вимоги, розроблено логічну й фізичну моделі бази даних, а також сформовано алгоритми роботи системи. Реалізовано структуру реляційної бази даних із використанням MySQL, що забезпечує цілісність, узгодженість та ефективність обробки інформації. Описані алгоритми реєстрації, авторизації, управління оголошеннями та відгуками дозволили формалізувати логіку роботи застосунку. У третьому розділі розглянуто особливості реалізації користувацького інтерфейсу, інтерфейсу адміністратора та менеджера, а також механізми роботи з оголошеннями, відгуками та рейтинговою системою. Проведене тестування підтвердило працездатність системи, коректність обробки даних та ефективність реалізованих механізмів безпеки. Інтерфейс застосунку відповідає сучасним вимогам до адаптивності та зручності використання.

У результаті виконаної роботи досягнуто поставленої мети — створено функціональний веб-застосунок каталогу, що забезпечує багаторівневу систему доступу, підтримує пошук і фільтрацію компаній, реалізує механізм відгуків та рейтингового оцінювання, а також відповідає вимогам інформаційної безпеки. Практичне значення роботи полягає у можливості використання розробленого

					ІПЗ.КР.Б – 121-26-ІПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

застосунку як готового рішення для підтримки малого бізнесу або як основи для подальшого розширення функціоналу. Перспективами подальших досліджень є інтеграція із зовнішніми сервісами (платіжними системами, картографічними API), впровадження аналітичних модулів та розгортання системи у хмарному середовищі. Таким чином, результати бакалаврської роботи підтверджують ефективність застосування сучасних веб-технологій для створення інформаційних систем підтримки малого бізнесу та демонструють можливість їх практичного впровадження.

					ІІЗ.КР.Б – 121-26-ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Laudon K. C., Traver C. G. *E-commerce 2023–2024: Business, Technology and Society*. 18th ed. Pearson, 2023.
2. Google. *Business Profile Help*. Google, 2026. URL: <https://support.google.com/business/>
3. UA-REGION. *Український бізнес-каталог підприємств*. 2026. URL: <https://www.ua-region.com.ua/>
4. Google. *Google Maps Platform Documentation*. Google, 2026. URL: <https://developers.google.com/maps/>
5. Bourque P., Fairley R. E. *Guide to the Software Engineering Body of Knowledge SWEBOOK Guide*. Version 4.0. IEEE Computer Society, 2024.
6. OWASP Foundation. *Access Control Cheat Sheet*. 2026. URL: <https://cheatsheetseries.owasp.org/>
7. Freeman E., Robson E. *Head First Design Patterns*. 2nd ed. Sebastopol : O'Reilly Media, 2021.
8. Object Management Group. *Unified Modeling Language Specification*. 2024. URL: <https://www.omg.org/spec/UML/>
9. ISO/IEC/IEEE 12207:2017. *Systems and software engineering — Software life cycle processes*. Geneva : ISO, 2017.
10. OWASP Foundation. *OWASP Top 10:2021*. 2021. URL: <https://owasp.org/Top10/>
11. Laudon K. C., Laudon J. P. *Management Information Systems: Managing the Digital Firm*. 17th ed. Pearson, 2021.
12. Fowler M. *Patterns of Enterprise Application Architecture*. Boston : Addison-Wesley, 2003.
13. PHP Manual. *Sessions*. The PHP Group, 2026. URL: <https://www.php.net/manual/en/book.session.php>
14. Coronel C., Morris S. *Database Systems: Design, Implementation, & Management*. 14th ed. Boston : Cengage, 2023.

					ІІЗ.КР.Б – 121-26-ІІЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

15. Pressman R. S., Maxim B. R. *Software Engineering: A Practitioner's Approach*. 9th ed. New York : McGraw-Hill Education, 2020.
16. Nixon R. *Learning PHP, MySQL & JavaScript*. 7th ed. Sebastopol : O'Reilly Media, 2025.
17. PHP Manual. *Password Hashing*. The PHP Group, 2026. URL: <https://www.php.net/manual/en/book.password.php>
18. MySQL 8.4 Reference Manual. Oracle Corporation, 2026. URL: <https://dev.mysql.com/doc/refman/8.4/en/>
19. Bootstrap Documentation. Bootstrap Team, 2026. URL: <https://getbootstrap.com/docs/5.3/>
20. W3C. *Web Content Accessibility Guidelines (WCAG) 2.2*. W3C Recommendation, 2023. URL: <https://www.w3.org/TR/WCAG22/>
21. MySQL Documentation. *Full-Text Search Functions*. Oracle Corporation, 2026. URL: <https://dev.mysql.com/doc/>
22. MDN Web Docs. *HTML, CSS and JavaScript documentation*. Mozilla, 2026. URL: <https://developer.mozilla.org/>
23. Nielsen Norman Group. *User Reviews and Ratings UX*. 2024. URL: <https://www.nngroup.com/>
24. Stair R., Reynolds G. *Principles of Information Systems*. 14th ed. Boston : Cengage, 2021.
25. ISO/IEC 25010:2023. *Systems and software engineering — Systems and software Quality Requirements and Evaluation*. Geneva : ISO, 2023.
26. OWASP Foundation. *Authentication Cheat Sheet*. 2026. URL: <https://cheatsheetseries.owasp.org/>
27. MySQL 8.4 Reference Manual. *Aggregate Functions*. Oracle Corporation, 2026. URL: <https://dev.mysql.com/doc/>
28. OWASP Foundation. *Cross Site Scripting Prevention Cheat Sheet*. 2026. URL: <https://cheatsheetseries.owasp.org/>

					ІІЗ.КР.Б – 121-26-ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

29. Bootstrap Documentation. *Layout and responsive design*. Bootstrap Team, 2026. URL: <https://getbootstrap.com/docs/5.3/layout/>
30. Duckett J. *PHP & MySQL: Server-side Web Development*. Indianapolis : Wiley, 2022.

					ІПЗ.КР.Б – 121-26-ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

Index.php

```

<?php
declare(strict_types=1);
session_start();

require_once __DIR__ . '/../app/core/Database.php';
require_once __DIR__ . '/../app/core/Controller.php';
require_once __DIR__ . '/../app/core/Auth.php';
require_once __DIR__ . '/../app/core/Router.php';
require_once __DIR__ . '/../app/core/Csrf.php';
require_once __DIR__ . '/../app/core/Validator.php';

spl_autoload_register(function($class){
    $paths = [
        __DIR__ . '/../app/controllers/' . $class . '.php',
        __DIR__ . '/../app/models/' . $class . '.php',
        __DIR__ . '/../app/core/' . $class . '.php',
    ];
    foreach ($paths as $p) {
        if (file_exists($p)) { require_once $p; return; }
    }
});

$router = new Router();

/* Public */
$router->get('/', 'ListingsController@index');
$router->get('/listing', 'ListingsController@show'); // ?id=1
$router->get('/categories', 'CategoriesController@index');

/* Auth */
$router->get('/login', 'AuthController@loginForm');
$router->post('/login', 'AuthController@login');
$router->get('/register', 'AuthController@registerForm');
$router->post('/register', 'AuthController@register');
$router->post('/logout', 'AuthController@logout');

/* User */
$router->get('/my-listings', 'ListingsController@my');
$router->get('/listing-create', 'ListingsController@createForm');
$router->post('/listing-create', 'ListingsController@create');
$router->get('/listing-edit', 'ListingsController@editForm');
$router->post('/listing-edit', 'ListingsController@edit');

$router->post('/review-save', 'ReviewsController@save');
$router->post('/review-delete', 'ReviewsController@delete');
* Admin/Manager */
$router->get('/admin', 'AdminController@dashboard');

```

```

$router->get('/admin-listings', 'AdminController@listings');
$router->post('/admin-listing-delete', 'AdminController@deleteListing');

$router->get('/admin-categories', 'AdminController@categories');
$router->post('/admin-category-create', 'AdminController@createCategory');
$router->post('/admin-category-update', 'AdminController@updateCategory');
$router->post('/admin-category-delete', 'AdminController@deleteCategory');

$router->get('/admin-regions', 'AdminController@regions');
$router->post('/admin-region-create', 'AdminController@createRegion');
$router->post('/admin-region-update', 'AdminController@updateRegion');
$router->post('/admin-region-delete', 'AdminController@deleteRegion');

$router->get('/admin-cities', 'AdminController@cities');
$router->post('/admin-city-create', 'AdminController@createCity');
$router->post('/admin-city-update', 'AdminController@updateCity');
$router->post('/admin-city-delete', 'AdminController@deleteCity');

$router->get('/admin-users', 'AdminController@users');
$router->post('/admin-user-role', 'AdminController@setUserRole');
$router->post('/admin-user-create', 'AdminController@createUser');
$router->post('/admin-user-delete', 'AdminController@deleteUser');

$router->dispatch(
    $_SERVER['REQUEST_METHOD'],
    parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH)
);

```

Models/city.php

```
<?php
```

```

class City {

    public static function all(): array {
        $pdo = Database::pdo();

        return $pdo->query("SELECT cities.*, regions.name AS region_name FROM cities JOIN
regions ON regions.id = cities.region_id ORDER BY cities.name ASC")->fetchAll();
    }

    public static function create(string $name, int $regionId, int $createdBy): int {
        $pdo = Database::pdo();

        $st = $pdo->prepare("INSERT INTO cities(name,region_id,created_by) VALUES(?,?,?)");
        $st->execute([$name, $regionId, $createdBy]);

        return (int)$pdo->lastInsertId();
    }

    public static function update(int $id, string $name, int $regionId): void {
        $pdo = Database::pdo();

        $st = $pdo->prepare("UPDATE cities SET name=?, region_id=? WHERE id=?");
    }
}

```

```

    $st->execute([$name, $regionId, $id]);
}

public static function delete(int $id): void {
    $pdo = Database::pdo();

    $st = $pdo->prepare("DELETE FROM cities WHERE id=?");
    $st->execute([$id]);
}
}

```

Controllers/AdminController.php

```
<?php
```

```

class AdminController extends Controller {

    public function dashboard(): void {
        Auth::requireRole(['admin', 'manager']);
        $this->view('admin/dashboard', []);
    }

    public function listings(): void {
        Auth::requireRole(['admin', 'manager']);
        $listings = Listing::allAdmin();
        $this->view('admin/listings', compact('listings'), 'admin');
    }

    public function deleteListing(): void {
        Auth::requireRole(['admin']);
        if (!Csrf::check($_POST['_csrf'] ?? null)) { http_response_code(419); echo "CSRF
token mismatch"; return; }

        $id = (int)($_POST['id'] ?? 0);
        if ($id > 0) Listing::delete($id);

        $this->flash('success', 'Оголошення видалено.');
```

`$this->redirect('/admin-listings');`

```

    }

    public function categories(): void {
        Auth::requireRole(['admin', 'manager']);
        $categories = Category::all();
        $this->view('admin/categories', compact('categories'), 'admin');
    }

    public function createCategory(): void {
        Auth::requireRole(['admin', 'manager']);
        if (!Csrf::check($_POST['_csrf'] ?? null)) { http_response_code(419); echo "CSRF
token mismatch"; return; }

        $name = trim($_POST['name'] ?? '');
        if ($name === '') {
            $this->flash('danger', 'Назва категорії не може бути порожньою.');
```

`$this->redirect('/admin-categories');`

```

        return;
    }

    try {
        Category::create($name, (int)Auth::user()['id']);
        $this->flash('success', 'Категорію додано.');
    } catch (Throwable $e) {
        $this->flash('danger', 'Не вдалося додати категорію (можливо, вже існує).');
    }

    $this->redirect('/admin-categories');
}

public function updateCategory(): void {
    Auth::requireRole(['admin', 'manager']);
    if (!Csrf::check($_POST['_csrf'] ?? null)) { http_response_code(419); echo "CSRF
token mismatch"; return; }

    $id = (int)($_POST['id'] ?? 0);
    $name = trim($_POST['name'] ?? '');
    $slug = trim($_POST['slug'] ?? '');

    if ($id <= 0 || $name === '') {
        $this->flash('danger', 'Некоректні дані для редагування категорії.');
        $this->redirect('/admin-categories');
        return;
    }

    if ($slug === '') $slug = $name;

    try {
        Category::update($id, $name, $slug);
        $this->flash('success', 'Категорію оновлено.');
    } catch (Throwable $e) {
        $this->flash('danger', 'Не вдалося оновити категорію (можливо, slug вже
існує).');
    }

    $this->redirect('/admin-categories');
}

public function deleteCategory(): void {
    Auth::requireRole(['admin']);
    if (!Csrf::check($_POST['_csrf'] ?? null)) { http_response_code(419); echo "CSRF
token mismatch"; return; }

    $id = (int)($_POST['id'] ?? 0);
    if ($id <= 0) {
        $this->redirect('/admin-categories');
        return;
    }

    try {

```

```

        Category::delete($id);
        $this->flash('success', 'Категорію видалено.');
```

} catch (Throwable \$e) {

```

        $this->flash('danger', 'Не вдалося видалити категорію (можливо, вона
використовується в оголошеннях).');
```

}

```

        $this->redirect('/admin-categories');
```

}

```

public function regions(): void {
    Auth::requireRole(['admin','manager']);
    $regions = Region::all();
    $this->view('admin/regions', compact('regions'), 'admin');
```

}

```

public function createRegion(): void {
    Auth::requireRole(['admin','manager']);

    if (!Csrf::check($_POST['_csrf'] ?? null)) {
        http_response_code(419);
        echo "CSRF token mismatch";
        return;
    }

    $name = trim($_POST['name'] ?? '');

    if ($name === '') {
        $this->flash('danger', 'Назва регіону не може бути порожньою.');
```

\$this->redirect('/admin-regions');

```

        return;
    }

    try {
        Region::create($name, (int)Auth::user()['id']);
        $this->flash('success', 'Регіон додано.');
```

} catch (Throwable \$e) {

```

        $this->flash('danger', 'Не вдалося додати регіон.');
```

}

```

    $this->redirect('/admin-regions');
```

}

```

public function updateRegion(): void {
    Auth::requireRole(['admin','manager']);

    if (!Csrf::check($_POST['_csrf'] ?? null)) {
        http_response_code(419);
        echo "CSRF token mismatch";
        return;
    }

    $id = (int)($_POST['id'] ?? 0);
    $name = trim($_POST['name'] ?? '');

```

```

if ($id <= 0 || $name === '') {
    $this->flash('danger', 'Некоректні дані для редагування регіону.');
```

\$this->redirect('/admin-regions');
 return;
 }

 try {
 Region::update(\$id, \$name);
 \$this->flash('success', 'Регіон оновлено.');

} catch (Throwable \$e) {
 \$this->flash('danger', 'Не вдалося оновити регіон.');

}

 \$this->redirect('/admin-regions');
}

public function deleteRegion(): void {
 Auth::requireRole(['admin']);

 if (!Csrf::check(\$_POST['_csrf'] ?? null)) {
 http_response_code(419);
 echo "CSRF token mismatch";
 return;
 }

 \$id = (int)(\$_POST['id'] ?? 0);

 if (\$id <= 0) {
 \$this->redirect('/admin-regions');
 return;
 }

 try {
 Region::delete(\$id);
 \$this->flash('success', 'Регіон видалено.');

} catch (Throwable \$e) {
 \$this->flash('danger', 'Не вдалося видалити регіон.');

}

 \$this->redirect('/admin-regions');
}

public function cities(): void {
 Auth::requireRole(['admin', 'manager']);

 \$cities = City::all();
 \$regions = Region::all();

 \$this->view('admin/cities', compact('cities', 'regions'), 'admin');

}

public function createCity(): void {
 Auth::requireRole(['admin', 'manager']);

```

if (!Csrf::check($_POST['_csrf'] ?? null)) {
    http_response_code(419);
    echo "CSRF token mismatch";
    return;
}

$name = trim($_POST['name'] ?? '');
$regionId = (int)($_POST['region_id'] ?? 0);

if ($name === '' || $regionId <= 0) {
    $this->flash('danger', 'Заповніть місто та регіон.');
    $this->redirect('/admin-cities');
    return;
}

try {
    City::create($name, $regionId, (int)Auth::user()['id']);
    $this->flash('success', 'Місто додано.');
```

```

} catch (Throwable $e) {
    $this->flash('danger', 'Не вдалося додати місто (можливо, вже існує)');
```

```

}

$this->redirect('/admin-cities');
}

public function updateCity(): void {
    Auth::requireRole(['admin', 'manager']);

    if (!Csrf::check($_POST['_csrf'] ?? null)) {
        http_response_code(419);
        echo "CSRF token mismatch";
        return;
    }

    $id = (int)($_POST['id'] ?? 0);
    $name = trim($_POST['name'] ?? '');
    $regionId = (int)($_POST['region_id'] ?? 0);

    if ($id <= 0 || $name === '' || $regionId <= 0) {
        $this->flash('danger', 'Некоректні дані для редагування міста.');
```

```

        $this->redirect('/admin-cities');
        return;
    }

    try {
        City::update($id, $name, $regionId);
        $this->flash('success', 'Місто оновлено.');
```

```

    } catch (Throwable $e) {
        $this->flash('danger', 'Не вдалося оновити місто.');
```

```

    }

    $this->redirect('/admin-cities');
}

```

```

public function deleteCity(): void {
    Auth::requireRole(['admin']);

    if (!Csrf::check($_POST['_csrf'] ?? null)) {
        http_response_code(419);
        echo "CSRF token mismatch";
        return;
    }

    $id = (int)($_POST['id'] ?? 0);

    if ($id <= 0) {
        $this->redirect('/admin-cities');
        return;
    }

    try {
        City::delete($id);
        $this->flash('success', 'Місто видалено.');
```

```

    } catch (Throwable $e) {
        $this->flash('danger', 'Не вдалося видалити місто.');
```

```

    }

    $this->redirect('/admin-cities');
}

public function users(): void {
    Auth::requireRole(['admin']);
    $users = User::all();
    $this->view('admin/users', compact('users'), 'admin');
```

```

}

public function setUserRole(): void {
    Auth::requireRole(['admin']);
    if (!Csrf::check($_POST['_csrf'] ?? null)) { http_response_code(419); echo "CSRF
token mismatch"; return; }

    $id = (int)($_POST['id'] ?? 0);
    $role = $_POST['role'] ?? 'user';
    if (!in_array($role, ['admin', 'manager', 'user'], true)) $role = 'user';

    if ($id > 0) User::setRole($id, $role);

    $this->flash('success', 'Роль оновлено.');
```

```

    $this->redirect('/admin-users');
}

public function createUser(): void {
    Auth::requireRole(['admin']);
    if (!Csrf::check($_POST['_csrf'] ?? null)) { http_response_code(419); echo "CSRF
token mismatch"; return; }

    $name = trim($_POST['name'] ?? '');

```

```

$email = trim($_POST['email'] ?? '');
$password = (string)($_POST['password'] ?? '');
$role = $_POST['role'] ?? 'user';

if ($name === '' || $email === '' || strlen($password) < 12) {
    $this->flash('danger', 'Заповніть ім'я, email і пароль (мін. 6 символів).');
    $this->redirect('/admin-users');
    return;
}

if (!in_array($role, ['admin','manager','user'], true)) $role = 'user';

try {
    User::create($name, $email, $password, $role);
    $this->flash('success', 'Користувача додано.');
```

```

} catch (Throwable $e) {
    $this->flash('danger', 'Не вдалося додати користувача (можливо, email вже існує).');
```

```

}

$this->redirect('/admin-users');
}

public function deleteUser(): void {
    Auth::requireRole(['admin']);
    if (!Csrf::check($_POST['_csrf'] ?? null)) { http_response_code(419); echo "CSRF token mismatch"; return; }

    $id = (int)($_POST['id'] ?? 0);
    if ($id <= 0) {
        $this->redirect('/admin-users');
        return;
    }

    $me = Auth::user();
    if ($me && (int)$me['id'] === $id) {
        $this->flash('warning', 'Не можна видалити свій акаунт.');
```

```

        $this->redirect('/admin-users');
        return;
    }

    User::deleteById($id);

    $this->flash('success', 'Користувача видалено.');
```

```

    $this->redirect('/admin-users');
}
}

```

Views/dashboard.php

```

<?php
$config = require __DIR__ . '/../../../config/config.php';
$base = $config['app']['base_url'] ?? '';
$user = Auth::user();
?>

```

```

<div class="container py-4">
  <h1 class="mb-2">Адмін-панель</h1>
  <div class="text-muted mb-4">Ви увійшли як: <strong><?= htmlspecialchars($u['name'])
?></strong> (<?= htmlspecialchars($u['role']) ?>)</div>

  <div class="row g-3">
    <div class="col-md-6 col-lg-4">
      <div class="card h-100">
        <div class="card-body">
          <h5>Оголошення</h5>
          <p class="text-muted mb-3">Перегляд і керування списком оголошень.</p>
          <a class="btn btn-outline-primary" href="<?= $base ?>/admin-listings">Відкрити</a>
        </div>
      </div>
    </div>

    <div class="col-md-6 col-lg-4">
      <div class="card h-100">
        <div class="card-body">
          <h5>Категорії</h5>
          <p class="text-muted mb-3">Додавання нових категорій (admin/manager).</p>
          <a class="btn btn-outline-primary" href="<?= $base ?>/admin-
categories">Відкрити</a>
        </div>
      </div>
    </div>

    <div class="col-md-6 col-lg-4">
      <div class="card h-100">
        <div class="card-body">
          <h5>Міста</h5>
          <p class="text-muted mb-3">Додавання нових міст (admin/manager).</p>
          <a class="btn btn-outline-primary" href="<?= $base ?>/admin-cities">Відкрити</a>
        </div>
      </div>
    </div>

    <div class="col-md-6 col-lg-4">
      <div class="card h-100">
        <div class="card-body">
          <h5>Регіони</h5>
          <p class="text-muted mb-3">Додавання нових регіонів (admin/manager).</p>
          <a class="btn btn-outline-primary" href="<?= $base ?>/admin-regions">Відкрити</a>
        </div>
      </div>
    </div>

    <?php if ($u['role']==='admin'): ?>
    <div class="col-md-6 col-lg-4">
      <div class="card h-100">
        <div class="card-body">
          <h5>Користувачі</h5>
          <p class="text-muted mb-3">Керування ролями користувачів (тільки admin).</p>
          <a class="btn btn-outline-primary" href="<?= $base ?>/admin-users">Відкрити</a>
        </div>
      </div>
    </div>
  </div>

```

```
    </div>
  </div>
</div>
<?php endif; ?>
</div>
</div>
```