МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: **«РОЗРОБКА ТОРГОВОЇ ОНЛАЙН-ПЛАТФОРМИ З АДАПТИВНИМ ДИЗАЙНОМ»**

Виконала: студентка 4 курсу, групи 6.1221
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)
освітньої програми Комп'ютерні науки
(назва освітньої програми)
Р.Л. Радчук
(ініціали та прізвище)
Керівник завідувачка кафедри комп'ютерних наук, професор, д.т.н Шило Г. М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Рецензент завідувач кафедри програмної інженерії доцент, к.ф.-м.н. Лісняк А.О.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти бакалавр

Спеціальність 122 Комп'ютерні науки
(шифр і назва)

Освітня програма Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук,
д.т.н., професор

_____ Шило Г.М.
(підпис)

" 23 " 12 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Радчук Руслані Леонідівні
(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка торгової онлайн-платформи з адаптивним дизайном

керівник роботи Шило Галина Миколаївна, д.т.н., професор
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

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

2. Строк подання студентом роботи 02.06.2025

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

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Теоретичні основи розробки туристичних маркетплейсів.
2. Проектування інформаційної системи.
3. Реалізація туристичного маркетплейсу

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
презентація

6. Консультанти розділів роботи:

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

7. Дата видачі завдання 23.12.2024

К А Л Е Н Д А Р Н И Й П Л А Н

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	30.12.2024	
2.	Збір вихідних даних.	10.01.2025	
3.	Обробка методичних та теоретичних	05.04.2025	
	джерел.		
4.	Розробка першого та другого розділу.	24.04.2025	
5.	Розробка третього розділу.	20.05.2025	
6.	Оформлення та нормоконтроль	02.06.2025	
	кваліфікаційної роботи бакалавра.		
7.	Захист кваліфікаційної роботи.	18.06.2025	

Студент _____ Р. Л. Радчук
(підпис) (ініціали та прізвище)

Керівник роботи _____ Г. М. Шило
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ О.Г.Спиця
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка торгової онлайн-платформи з адаптивним дизайном» 82 с., 38 рис., 4 табл., 20 джерел, 1 додаток.

АДАПТИВНИЙ ДИЗАЙН, АДМІНІСТРАТИВНА ПАНЕЛЬ, БАЗА ДАНИХ, ГОТЕЛЬНА ІНДУСТРІЯ, ІНТЕРФЕЙС КОРИСТУВАЧА, ІНФОРМАЦІЙНА СИСТЕМА, ПІДБІР ГОТЕЛІВ, ПОШУКОВА СИСТЕМА, СИСТЕМА БРОНЮВАННЯ, ТОРГОВА ОНЛАЙН-ПЛАТФОРМА, ТУРИСТИЧНИЙ МАРКЕТПЛЕЙС, WEB-РОЗРОБКА.

Мета дослідження: розробка інформаційної системи торгової онлайн-платформи з адаптивним дизайном на прикладі туристичного маркетплейсу з вбудованою системою підбору готелів за параметрами.

Об’єкт дослідження: процеси пошуку та бронювання номерів у готелях за допомогою онлайн-платформ.

Предмет дослідження: методи та технології створення інформаційної системи торгової онлайн-платформи з адаптивним дизайном та розширеними функціональними можливостями на прикладі туристичного маркетплейсу.

Методи дослідження: аналіз існуючих рішень, системний аналіз, об’єктно-орієнтоване проєктування, UML-моделювання, функціональне тестування.

Результати та їх новизна: розроблено інформаційну систему, що реалізує торгову онлайн-платформу з адаптивним дизайном на прикладі туристичного маркетплейсу із вбудованою системою підбору готелів за заданими параметрами. Система складається з користувацького інтерфейсу для пошуку та бронювання номерів у готелях та адміністративної панелі для управління інформацією про готелі та бронювання. Новизна роботи полягає в оптимізації алгоритмів пошуку за комбінованими параметрами, реалізації адаптивного дизайну та інтеграції різних джерел даних в єдину систему.

SUMMARY

Bachelor's Qualifying Theses «Development of an online trading platform with responsive design »: 82 pages, 38 figures, 4 tables, 20 references, 1 appendix.

RESPONSIVE DESIGN, ADMIN PANEL, DATABASE, HOTEL INDUSTRY, USER INTERFACE, INFORMATION SYSTEM, HOTEL SELECTION, SEARCH ENGINE, BOOKING SYSTEM, E-COMMERCE PLATFORM, TOURIST MARKETPLACE, WEB DEVELOPMENT.

Research objective: development of an information system for an e-commerce platform with responsive design, using a tourist marketplace with an integrated hotel selection system based on parameters as an example.

Object of research: processes of searching and booking hotel rooms using online platforms.

Subject of research: methods and technologies for creating an information system for an e-commerce platform with responsive design and enhanced functionality, using a tourist marketplace as an example.

Research methods: analysis of existing solutions, system analysis, object-oriented design, uml modeling, functional testing.

Results and novelty: an information system has been developed that implements an e-commerce platform with responsive design, using a tourist marketplace with an integrated hotel selection system based on specified parameters as an example. The system consists of a user interface for searching and booking hotel rooms and an administrative panel for managing hotel and booking information. The novelty of the work lies in optimizing search algorithms by combined parameters, implementing responsive design, and integrating various data sources into a single system.

ЗМІСТ

Завдання на кваліфікаційну роботу 2
Реферат 4
Summary 5
Вступ 8
1 Теоретичні основи розробки туристичних маркетплейсів 11
1.1 Аналіз сучасних систем бронювання готелів 11
1.2 Огляд існуючих рішень торгових онлайн-платформ 13
1.3 Технологічні вимоги до торгових онлайн-платформ 17
1.4 Визначення функціональних вимог до розроблюваної системи 19
1.5 Висновки до першого розділу 24
2 Проектування інформаційної системи 25
2.1 Розробка архітектури системи 25
2.1.1 Загальна архітектура системи 25
2.1.2 Архітектура клієнтської частини 27
2.1.3 Архітектура серверної частини 28
2.1.4 Вибір інструментів розробки 30
2.2 Проектування бази даних системи 32
2.3 Проектування функціональної структури системи 36
2.4 Висновки до другого розділу 40
3 Реалізація туристичного маркетплейсу 41
3.1 Структура файлів проєкту 41
3.2 Розробка серверної частини системи 42
3.2.1 Структура маршрутизації та обробка запитів 42
3.2.2 Взаємодія з базою даних 43
3.2.3 Реалізація контролерів 44
3.2.4 Управління сесіями та аутентифікація 45
3.3 Розробка клієнтської частини системи 46

3.3.1 Структура інтерфейсу користувача 46
3.3.2 Модульна організація представлень 47
3.3.3 Використання JavaScript для інтерактивності 47
3.3.4 Адаптивний дизайн 49
3.4 Реалізація основних функціональних можливостей 49
3.4.1 Інтерфейс для клієнтів 50
3.4.2 Інтерфейс для адміністраторів 56
3.5 Тестування системи та виправлення помилок 60
3.5.1 Методологія тестування 60
3.5.2 Функціональне тестування інтерфейсів 61
3.5.3 Виявлені проблеми та їх усунення 62
3.6 Висновки до третього розділу 63
4 Впровадження та оцінка результатів роботи системи 64
4.1 Аналіз зручності використання системи 64
4.1.1 Кросбраузерна та кросплатформна сумісність 64
4.1.2 Аналіз користувацького шляху 65
4.1.3 Доступність інтерфейсу 65
4.2 Перспективи розвитку та вдосконалення системи 66
4.2.1 Функціональне розширення системи 66
4.2.2 Технологічне вдосконалення 67
4.3 Висновки до четвертого розділу 67
Перелік посилань 71
Додаток А Вихідні коди веб-сайту 73

ВСТУП

Сфера електронної комерції, попри періодичні глобальні виклики, залишається однією з найбільш динамічних галузей світової економіки. З розвитком інформаційних технологій суттєво змінилися підходи до організації різноманітних бізнес-процесів та пошуку потрібної інформації. Сучасний користувач прагне самостійно здійснювати вибір, порівнюючи оптимальні варіанти за співвідношенням ціни та якості. Торгові онлайн-платформи стали невід'ємною частиною цього процесу, забезпечуючи зручний пошук та доступ до різних товарів і послуг.

Актуальність дослідження зумовлена стрімким зростанням онлайн-транзакцій у різних секторах економіки та необхідністю створення більш ефективних систем пошуку та підбору, які б враховували різноманітні потреби користувачів. Незважаючи на існування багатьох популярних платформ, користувачі часто стикаються з проблемами надмірної кількості варіантів, складності фільтрації за специфічними параметрами та незручних інтерфейсів, які погано адаптуються до різних пристроїв, що ускладнює процес вибору оптимального варіанту.

Аналіз існуючих рішень свідчить про те, що більшість сучасних торгових онлайн-платформ мають обмежені можливості персоналізації пошуку, недостатньо гнучкі фільтри або складні для сприйняття інтерфейси. Туристична галузь, зокрема системи бронювання готелів, є яскравим прикладом сфери, де ці обмеження особливо помітні.

Метою даної кваліфікаційної роботи є розробка інформаційної системи торгової онлайн-платформи з адаптивним дизайном на прикладі туристичного маркетплейсу з вбудованою системою підбору готелів за параметрами.

Для досягнення поставленої мети були визначені наступні завдання:

– провести аналіз існуючих торгових онлайн-платформ, зокрема в туристичній сфері, виявити їх переваги та недоліки;

- визначити функціональні та нефункціональні вимоги до розроблюваної системи;
- розробити архітектуру інформаційної системи торгової онлайн-платформи;
- розробити алгоритми підбору за параметрами з урахуванням пріоритетів користувачів;
- створити інтуїтивно зрозумілий користувацький інтерфейс з адаптивним дизайном для пошуку та здійснення операцій.

Об’єктом дослідження є процеси пошуку та бронювання номерів у готелях за допомогою онлайн-платформ як конкретний приклад функціонування торгової онлайн-платформи. Ці процеси включають взаємодію користувачів із системою пошуку, фільтрацію результатів, порівняння варіантів та здійснення операцій.

Предметом дослідження є методи та технології створення інформаційної системи торгової онлайн-платформи з адаптивним дизайном та розширеними функціональними можливостями на прикладі туристичного маркетплейсу.

В процесі виконання роботи були використані наступні методи дослідження: аналіз існуючих рішень, системний аналіз, об’єктно-орієнтоване проєктування, UML-моделювання, функціональне тестування. Для реалізації системи використовувалися сучасні веб-технології, зокрема HTML5, CSS3, JavaScript, PHP, MySQL, а також принципи адаптивного дизайну та UX/UI проєктування.

Практична значимість роботи полягає в створенні функціональної інформаційної системи, яка може бути впроваджена в діяльність різних бізнесів, зокрема туристичних агентств, готельних мереж, або використана як самостійний сервіс для кінцевих користувачів. Розроблена система дозволяє підвищити ефективність процесу пошуку та взаємодії, покращити комунікацію між клієнтами та представниками бізнесу, зробити процес вибору більш персоналізованим та інтуїтивно зрозумілим на будь-якому пристрої.

Структура роботи відповідає логіці дослідження та складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі представлено аналіз предметної області та огляд існуючих торгових онлайн-платформ на прикладі систем бронювання готелів. Другий розділ присвячено проєктуванню архітектури системи, розробці бази даних та алгоритмів підбору. У третьому розділі описано процес реалізації системи, її тестування та оптимізації. Четвертий розділ присвячений впровадженню та оцінці результатів роботи системи.

1 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ТУРИСТИЧНИХ МАРКЕТПЛЕЙСІВ

1.1 Аналіз сучасних систем бронювання готелів

Сучасні системи електронної комерції пройшли значну еволюцію від простих інтернет-магазинів до складних інформаційних платформ, які інтегрують різноманітні функції та можливості. У контексті торгових онлайн-платформ, які динамічно розвиваються, важливо розуміти характеристики та особливості систем, що забезпечують взаємодію між продавцями та потенційними покупцями. Туристична індустрія, зокрема системи бронювання готелів, є одним із яскравих прикладів таких платформ.

Основні типи систем електронної комерції, що представлені на сучасному ринку, можна класифікувати наступним чином:

Маркетплейси виступають посередниками між продавцями та покупцями, дозволяючи першим розміщувати свої пропозиції на єдиній платформі, а другим – обирати з широкого асортименту різних постачальників. Прикладом у туристичній сфері є системи бронювання готелів, такі як Booking.com, які з’єднують готелі та потенційних гостей.

Спеціалізовані платформи електронної комерції зосереджуються на певних категоріях товарів або послуг, пропонуючи специфічні функції для цього сегменту ринку. Наприклад, Airbnb спеціалізується на короткостроковій оренді житла.

Корпоративні онлайн-платформи розробляються компаніями для продажу власних товарів або послуг та управління взаємовідносинами з клієнтами. Прикладом у готельній індустрії є сайти готельних мереж, таких як Marriott чи Hilton.

Аналіз функціональних можливостей сучасних систем електронної комерції показує, що найбільш затребуваними є наступні функції:

Інтуїтивно зрозумілий пошук з гнучкими фільтрами, що дозволяє користувачам швидко знаходити відповідні варіанти з урахуванням таких факторів, як розташування, ціновий діапазон, рейтинг, характеристики, наявність додаткових послуг тощо.

Детальна інформація про пропозиції, що включає фотографії, описи, характеристики, відгуки попередніх клієнтів, політику скасування та повернення коштів, а також додаткову інформацію про надавача послуг.

Системи рейтингів та відгуків, які дозволяють потенційним клієнтам ознайомитися з досвідом інших користувачів та зробити більш інформований вибір.

Інтеграція з платіжними системами, що забезпечує безпечне та зручне проведення оплати.

Адаптивний дизайн, який стає все більш важливим з огляду на зростаючу кількість користувачів мобільних пристроїв, забезпечуючи оптимальне відображення контенту на екранах різних розмірів.

Персоналізація пропозицій, що базується на аналізі попередніх пошуків та транзакцій користувача, його вподобань та поведінки на сайті.

Однак, незважаючи на широкий функціонал, багато сучасних торгових онлайн-платформ мають певні обмеження. Серед найбільш помітних недоліків можна виділити перевантаженість інтерфейсу, що ускладнює навігацію; обмежені можливості фільтрації за специфічними параметрами; відсутність інтеграції з супутніми сервісами; недостатня персоналізація пропозицій; неоптимізований адаптивний дизайн; технічні проблеми при високому навантаженні.

Розуміння цих обмежень є важливим для розробки нової системи, яка зможе подолати існуючі недоліки та запропонувати користувачам більш зручний та ефективний досвід використання торгової онлайн-платформи, зокрема у сфері бронювання готелів [1].

1.2 Огляд існуючих рішень торгових онлайн-платформ

Ринок електронної комерції представлений різноманітними онлайн-платформами, які пропонують послуги з продажу товарів та послуг. У якості прикладу розглянемо туристичну галузь, зокрема платформи для бронювання готелів, які є показовим прикладом торгових онлайн-платформ. Аналіз існуючих рішень дозволяє виявити їх сильні та слабкі сторони, а також визначити потенційні напрямки для вдосконалення.

Booking.com є одним з найбільших у світі онлайн-сервісів для бронювання готелів (рис. 1.1). Сервіс пропонує доступ до більш ніж 28 мільйонів об’єктів розміщення у 226 країнах світу. Серед переваг Booking.com можна виділити: великий вибір варіантів розміщення, від готелів та хостелів до апартаментів та вілл; інтуїтивно зрозумілий інтерфейс з гнучкими фільтрами; програма лояльності Genius, яка пропонує знижки та додаткові переваги для постійних користувачів; детальні описи об’єктів розміщення з фотографіями та відгуками; підтримка багатьох мов та валют. Однак, система має і певні недоліки: часто висока комісія для власників житла, що може відображатися на кінцевій ціні для користувачів; обмежені можливості фільтрації за специфічними параметрами; перевантаженість інтерфейсу на певних сторінках [5].

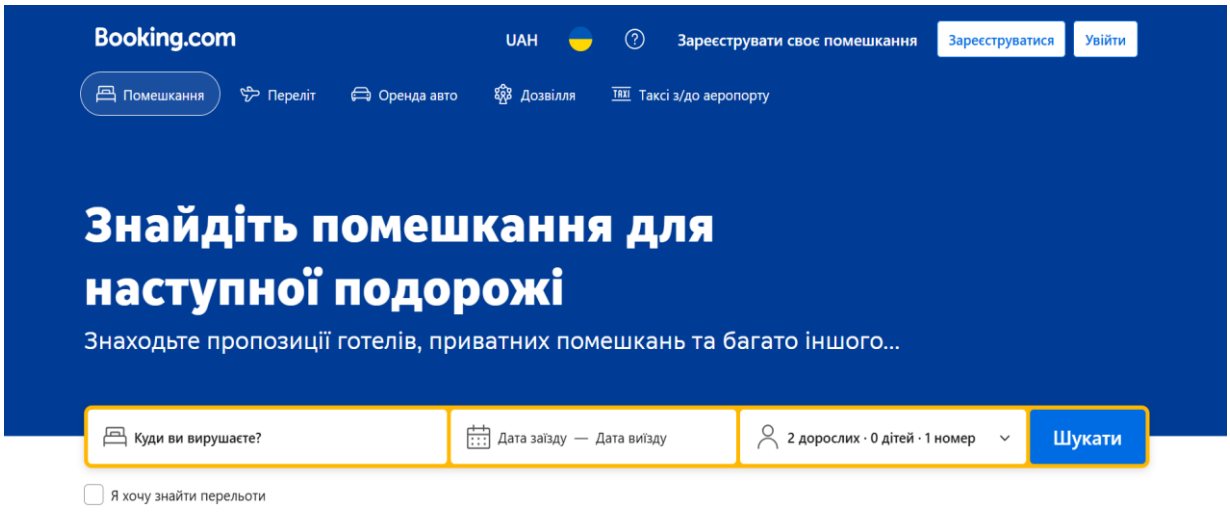


Рисунок 1.1 – Основна сторінка сайту Booking.com

Expedia Group (рис.1.2) володіє кількома туристичними платформами, включаючи Expedia, Hotels.com та інші. Ці платформи пропонують не лише бронювання готелів, але й комплексні туристичні пакети, що включають авіаквитки, оренду автомобілів та екскурсії. Серед переваг Expedia можна відзначити: можливість бронювання комплексних туристичних пакетів; програма лояльності з накопиченням балів за кожне бронювання; часті промо-акції та знижки; детальна інформація про готелі та визначні місця поруч. Проте, і тут є недоліки: вища вартість порівняно з деякими іншими сервісами; менш гнучкі опції фільтрації; складніша процедура скасування бронювання [2].

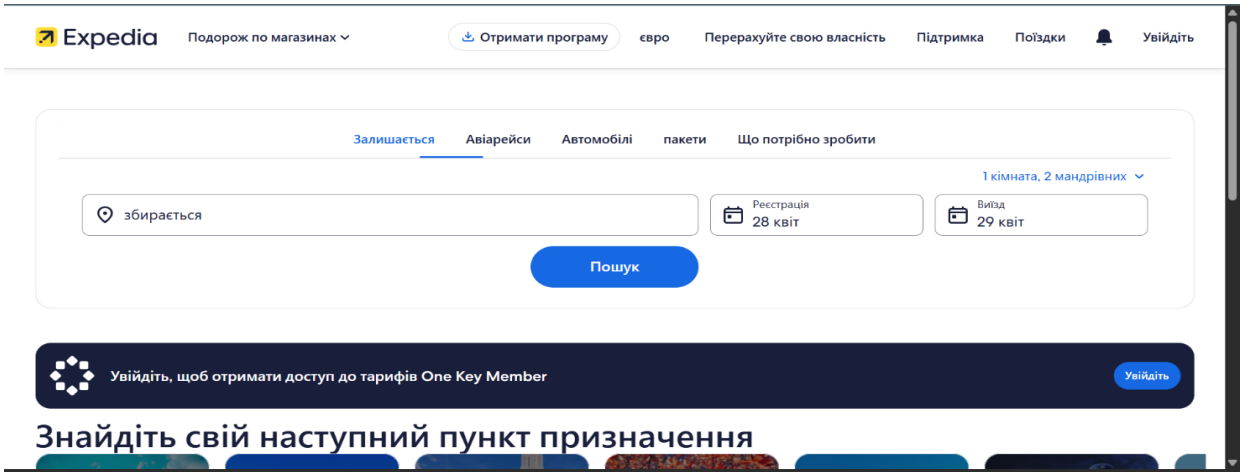


Рисунок 1.2 – Сторінка Expedia Group

Airbnb, хоча і не є традиційною платформою для бронювання готелів, становить значну конкуренцію на ринку туристичного розміщення. Сервіс спеціалізується на приватному житлі, від кімнат у квартирах до цілих будинків та унікальних варіантів розміщення (рис.1.3). Переваги Airbnb включають: унікальні варіанти розміщення, які часто не доступні на інших платформах; можливість спілкування з господарями та отримання локальних рекомендацій; часто нижчі ціни за тривале проживання; фокус на автентичному досвіді. Недоліки: менша стандартизація якості порівняно з готелями; можливі

проблеми з господарями або умовами проживання; обмежені можливості для бізнес-подорожей [3].

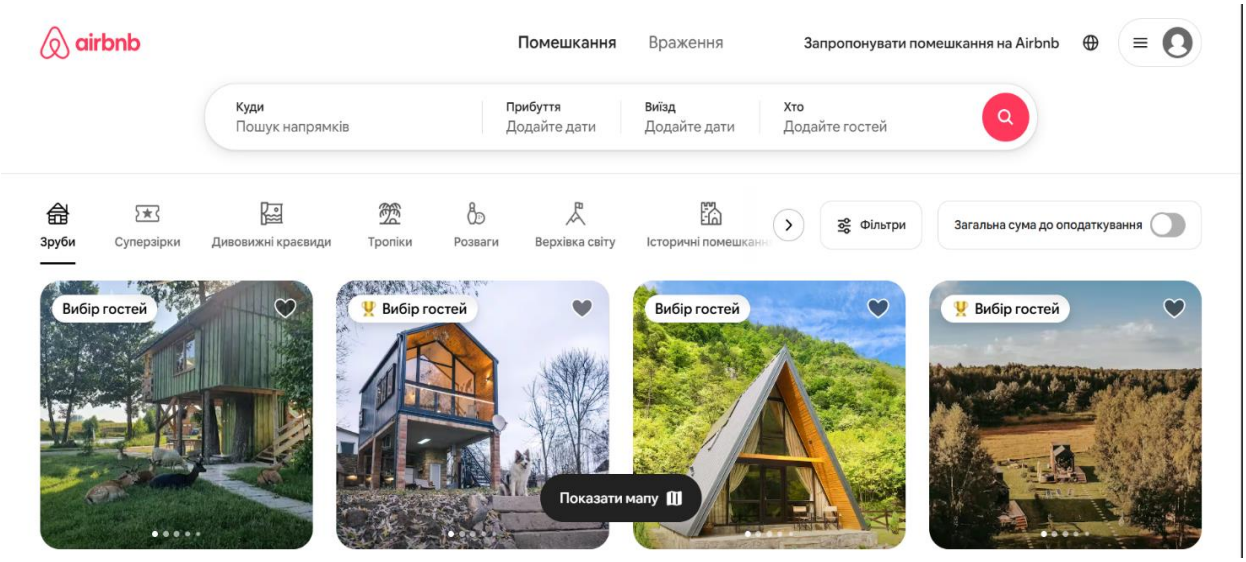


Рисунок 1.3 – Головна сторінка Airbnb

OtelMS.com спеціалізується на програмному забезпеченні для управління готелями, включаючи системи бронювання (рис 1.4). Цей сервіс відрізняється від попередніх тим, що орієнтований не на кінцевих споживачів, а на власників та менеджерів готелів.

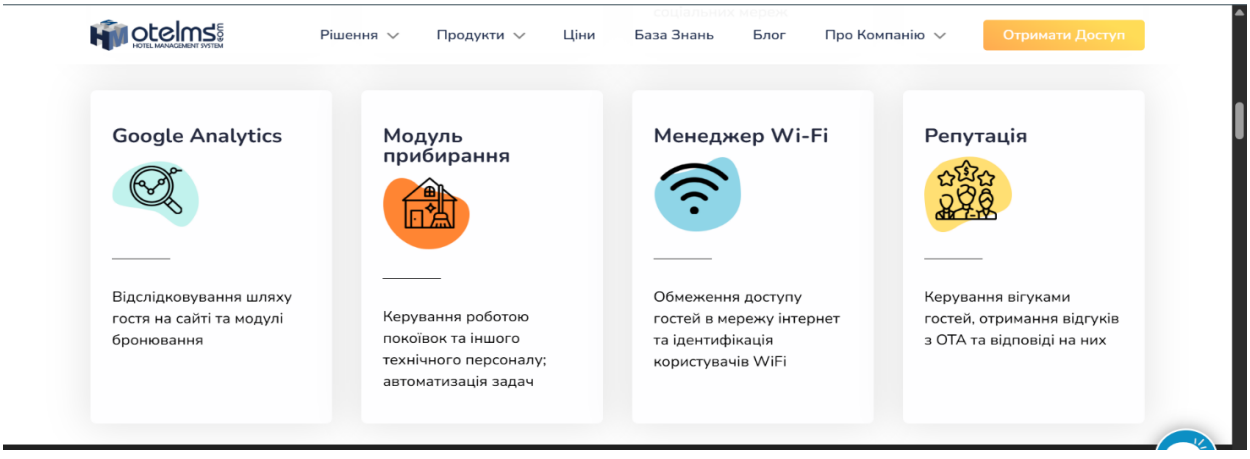


Рисунок 1.4 – Сторінка сайту OtelMS.com

Переваги OtelMS включають: комплексне рішення для управління готелем, включаючи бронювання, облік гостей, управління інвентарем тощо;

інтеграція з популярними туристичними платформами; аналітичні інструменти для оптимізації роботи готелю. Недоліки: висока вартість для малих готелів та хостелів; складність налаштування та інтеграції; фокус на B2B сегменті, а не на кінцевих споживачах [4].

На основі опису даних платформ було проведено порівняльну характеристику в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика програмних продуктів для бронювання готелів

Критерій	Booking.com	Expedia Group	Airbnb	OtelMS.com
Цільова аудиторія	Загальний споживач	Загальний споживач	Загальний споживач	B2B (готелі та хостели)
Масштаб	Глобальний (28+ млн об’єктів у 226 країнах)	Глобальний	Глобальний	Глобальний/локальний
Типи розміщення	Готелі, хостели, апартаменти, вілли	Готелі, апартаменти	Приватне житло, унікальні варіанти розміщення	Не застосовується (B2B)
Додаткові послуги	Обмежені	Комплексні туристичні пакети, авіаквитки, оренда авто, екскурсії	Локальні рекомендації, досвід	Управління готелем
Програма лояльності	Є (Genius)	Є (накопичення балів)	Відсутня	Не застосовується
Інтерфейс	Інтуїтивний, але часом перевантажений	Помірно зручний	Інтуїтивний	Складний (професійний)
Мовна підтримка	Багатомовний	Багатомовний	Багатомовний	Обмежена
Фільтрація	Гнучка, але з обмеженнями	Менш гнучка	Спеціалізована під різні типи житла	Професійно-орієнтована
Відгуки та рейтинги	Розвинена система	Розвинена система	Двостороння система оцінювання	Не застосовується

Продовження таблиці 1.1

Цінова політика	Варіативна, комісійна модель	Вища вартість	Часто нижчі ціни за тривале проживання	Висока вартість для бізнесу
Аналітичні інструменти	Обмежені	Обмежені	Обмежені	Розвинені
Інтеграція з іншими сервісами	Часткова	Висока (в межах Expedia Group)	Обмежена	Висока (з OTA-платформами)

Аналіз існуючих рішень показує, що, незважаючи на значну кількість платформ для бронювання готелів, існує потреба в системі, яка б поєднувала всі переваги існуючих рішень, зокрема: великий вибір варіантів розміщення; інтуїтивно зрозумілий інтерфейс з гнучкими фільтрами; персоналізовані рекомендації на основі вподобань користувача; інтеграція з місцевими сервісами та атракціями; прозора система рейтингів та відгуків.

Також важливо враховувати специфіку українського ринку туристичних послуг, який має свої особливості та потреби. Створення системи, яка б відповідала цим потребам, є перспективним напрямком розвитку туристичного бізнесу в Україні.

1.3 Технологічні вимоги до торгових онлайн-платформ

Розробка сучасної торгової онлайн-платформи вимагає врахування широкого спектру технологічних аспектів, які забезпечують її ефективне функціонування, масштабованість та безпеку. Ці вимоги формуються під впливом як технологічних тенденцій, так і потреб користувачів та бізнесу.

Архітектурні вимоги до торгових онлайн-платформ визначають загальну структуру та організацію компонентів системи. Сучасні рішення найчастіше базуються на мікросервісній архітектурі, яка забезпечує гнучкість, масштабованість та стійкість до відмов. Така архітектура дозволяє розділити систему на незалежні компоненти, кожен з яких відповідає за конкретну

функцію: каталог товарів, управління замовленнями, обробка платежів тощо. Важливим аспектом є також забезпечення високої доступності системи через використання балансування навантаження та розподілених баз даних.

Вимоги до баз даних включають вибір оптимальної моделі даних та систем управління базами даних (СУБД).

Вимоги до безпеки є критично важливими для систем електронної комерції, оскільки вони обробляють конфіденційні дані користувачів та фінансову інформацію. Необхідно забезпечити шифрування даних при передачі (використання HTTPS) та при зберіганні, особливо для платіжної інформації. Впровадження багатофакторної автентифікації та системи управління доступом (Role-Based Access Control) дозволяє захистити дані від несанкціонованого доступу. Регулярні аудити безпеки та відповідність міжнародним стандартам, таким як PCI DSS для систем, що обробляють платіжні карти, є обов’язковими умовами.

Інтеграційні вимоги визначають здатність системи взаємодіяти з іншими сервісами та платформами. Сучасні торгові онлайн-платформи повинні інтегруватися з різними API: платіжними системами для обробки транзакцій; логістичними сервісами для відстеження доставки; системами управління відносинами з клієнтами (CRM); сервісами геолокації для визначення місця доставки; соціальними мережами для спрощення реєстрації та поширення інформації.

Вимоги до користувацького інтерфейсу включають створення інтуїтивно зрозумілого, адаптивного дизайну, який забезпечує зручне використання системи на різних пристроях. Важливими аспектами є оптимізація швидкості завантаження сторінок, мінімізація кількості кроків для здійснення покупки, забезпечення доступності для користувачів з обмеженими можливостями відповідно до стандартів WCAG.

Вимоги до адаптивного дизайну передбачають, що інтерфейс платформи повинен коректно відображатися та функціонувати на пристроях з різними розмірами екранів: від великих настільних моніторів до маленьких смартфонів. Це досягається за допомогою таких технік, як гнучкі сітки та

макети, використання відносних одиниць вимірювання, медіа-запити для визначення характеристик пристрою, оптимізація зображень та контенту.

Вимоги до продуктивності та масштабованості передбачають можливість системи ефективно обробляти значну кількість запитів, особливо в пікові періоди, такі як сезонні розпродажі. Це досягається через горизонтальне масштабування серверної інфраструктури, оптимізацію запитів до бази даних, кешування часто запитуваних даних.

Отже, розробка сучасної торгової онлайн-платформи вимагає комплексного підходу до вирішення технологічних питань, враховуючи вимоги до архітектури, безпеки, інтеграції, адаптивного дизайну, користувацького інтерфейсу та продуктивності. Вибір оптимальних технологій та архітектурних рішень є ключовим фактором для створення конкурентоспроможної платформи.

1.4 Визначення функціональних вимог до розроблюваної системи

На основі аналізу існуючих рішень та технологічних тенденцій можна сформулювати функціональні вимоги до розроблюваної системи торгової онлайн-платформи з адаптивним дизайном на прикладі туристичного маркетплейсу. Функціональні вимоги описують конкретні функції та можливості, які система повинна надавати користувачам та адміністраторам.

Функціональні вимоги для користувачів системи передбачають розширені можливості пошуку та фільтрації за різноманітними параметрами. У контексті туристичного маркетплейсу користувачі повинні мати можливість здійснювати пошук за базовими параметрами, такими як місце розташування, дати заїзду та виїзду, кількість гостей, а також за додатковими параметрами: ціновий діапазон, рейтинг, відстань до певних об’єктів (центр міста, пляж, аеропорт тощо), наявність певних зручностей (басейн, спа, тренажерний зал, Wi-Fi, паркінг тощо), тип харчування, категорія готелю (кількість зірок).

Система повинна забезпечувати детальний перегляд інформації про пропозиції, включаючи загальний опис, фотогалерею, інформацію про номери та їх наявність, послуги та зручності, правила проживання та скасування бронювання, розташування на карті з можливістю перегляду найближчих визначних місць та транспортних вузлів, відгуки та рейтинги попередніх клієнтів.

Важливим функціональним елементом є процес здійснення транзакцій та управління ними. Система повинна забезпечувати зручний процес вибору та замовлення, заповнення інформації про клієнта, вибір способу оплати, підтвердження транзакції з отриманням підтверджувального електронного листа. Користувачі також повинні мати можливість переглядати історію своїх операцій, скасовувати або змінювати їх (відповідно до правил), залишати відгуки та рейтинги після отримання послуг.

Персональний кабінет користувача повинен включати можливість управління особистими даними, перегляд та управління операціями, збереження улюблених варіантів, історію пошуків, налаштування повідомлень та сповіщень.

Функціональні вимоги для продавців та адміністраторів магазинів/готелів включають можливість реєстрації та управління інформацією про продукт/об’єкт: заповнення загальної інформації, додавання фотографій, опис послуг та зручностей, встановлення правил та політик. У контексті туристичного маркетплейсу важливими функціями є управління наявністю номерів та тарифами: додавання різних типів номерів з детальним описом, встановлення базових тарифів та спеціальних пропозицій, управління календарем доступності, визначення обмежень (мінімальна тривалість проживання, обмеження на дати заїзду/виїзду тощо).

Адміністратори повинні мати можливість управління замовленнями/бронюваннями: перегляд нових запитів, підтвердження або відхилення запитів, комунікація з клієнтами, внесення змін, генерація звітів по операціях та доходах, управління відгуками клієнтів (відповіді на відгуки, вирішення проблемних ситуацій).

Функціональні вимоги для адміністраторів системи включають управління користувачами системи (реєстрація, деактивація, управління ролями та дозволами), управління контентом платформи (новини, статті, рекламні банери тощо), модерація контенту, створеного користувачами (відгуки, рейтинги, фотографії), управління системними налаштуваннями (валюти, мови, країни, категорії готелів тощо), моніторинг активності системи, виявлення та реагування на проблеми, генерація аналітичних звітів про роботу платформи.

Система підбору за параметрами є ключовою функціональною вимогою для розроблюваної системи. Вона повинна забезпечувати релевантні результати пошуку на основі вказаних користувачем параметрів, ранжування результатів з урахуванням вагових коефіцієнтів для різних параметрів, персоналізовані рекомендації на основі попередніх пошуків та транзакцій користувача, пропозиції альтернативних варіантів, якщо прямі результати пошуку обмежені, інтелектуальний підбір оптимальних варіантів з урахуванням співвідношення ціни та якості.

Додаткові функціональні вимоги можуть включати інтеграцію з сервісами для бронювання додаткових послуг (трансфер, екскурсії, оренда автомобіля тощо), систему повідомлень для комунікації між користувачами та представниками надавачів послуг, систему сповіщень про спеціальні пропозиції, зміни в бронюваннях тощо, багатомовний інтерфейс та підтримку різних валют, інтеграцію з соціальними мережами для спрощення реєстрації та поширення інформації.

Для більш наочного представлення функціональних вимог до системи торгової онлайн-платформи на прикладі туристичного маркетплейсу нижче наведено діаграми варіантів використання та потоків даних.

Діаграма варіантів використання (рис. 1.5) демонструє основні функціональні можливості системи з точки зору різних користувачів. Як видно з діаграми, звичайний користувач може здійснювати пошук за параметрами, переглядати інформацію про готелі, бронювати номери, управляти власними бронюваннями, залишати відгуки, використовувати особистий кабінет,

переглядати рейтинги та застосовувати фільтрацію під час пошуку. Адміністратор системи має розширені можливості, включаючи управління користувачами, додавання та редагування об’єктів, додавання номерів, модерацію контенту, налаштування системи, генерацію звітів та перегляд статистики використання платформи.

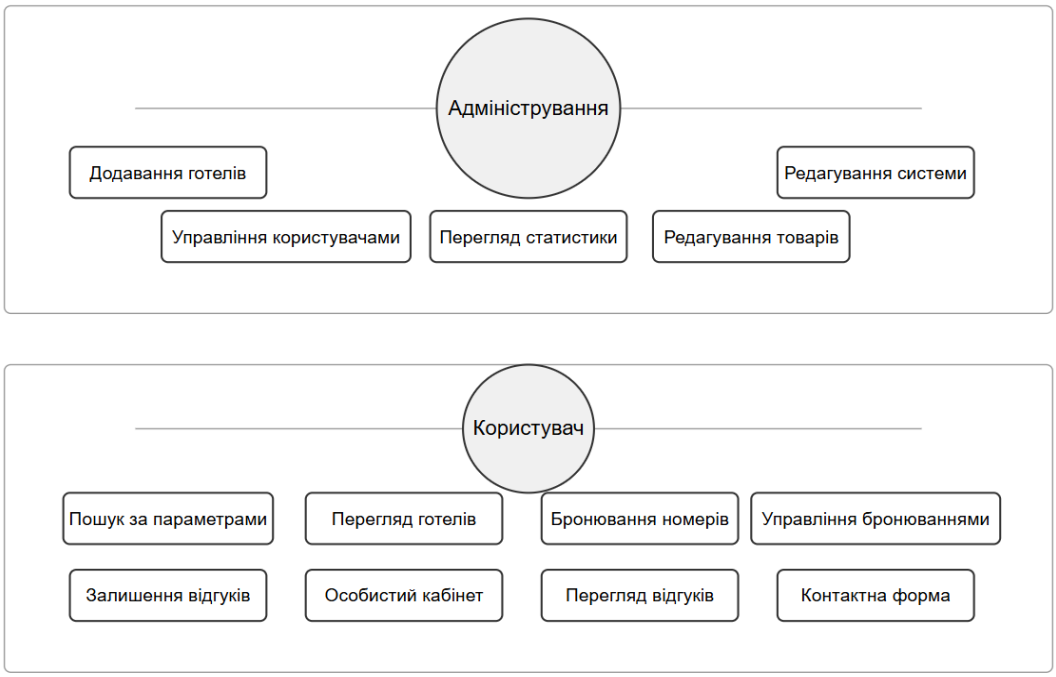


Рисунок 1.5 – Діаграма варіантів використання системи туристичного маркетплейсу

Діаграма потоків даних (рис. 1.6) відображає рух інформації між різними компонентами системи та її користувачами. Вона демонструє взаємодію користувача з системою пошуку та бронювання, збереження даних у базі даних, сповіщення адміністратора про нові бронювання, а також процеси управління контентом та генерації звітів з боку адміністратора. Така візуалізація допомагає краще зрозуміти принципи роботи системи та взаємозв’язки між її функціональними блоками.

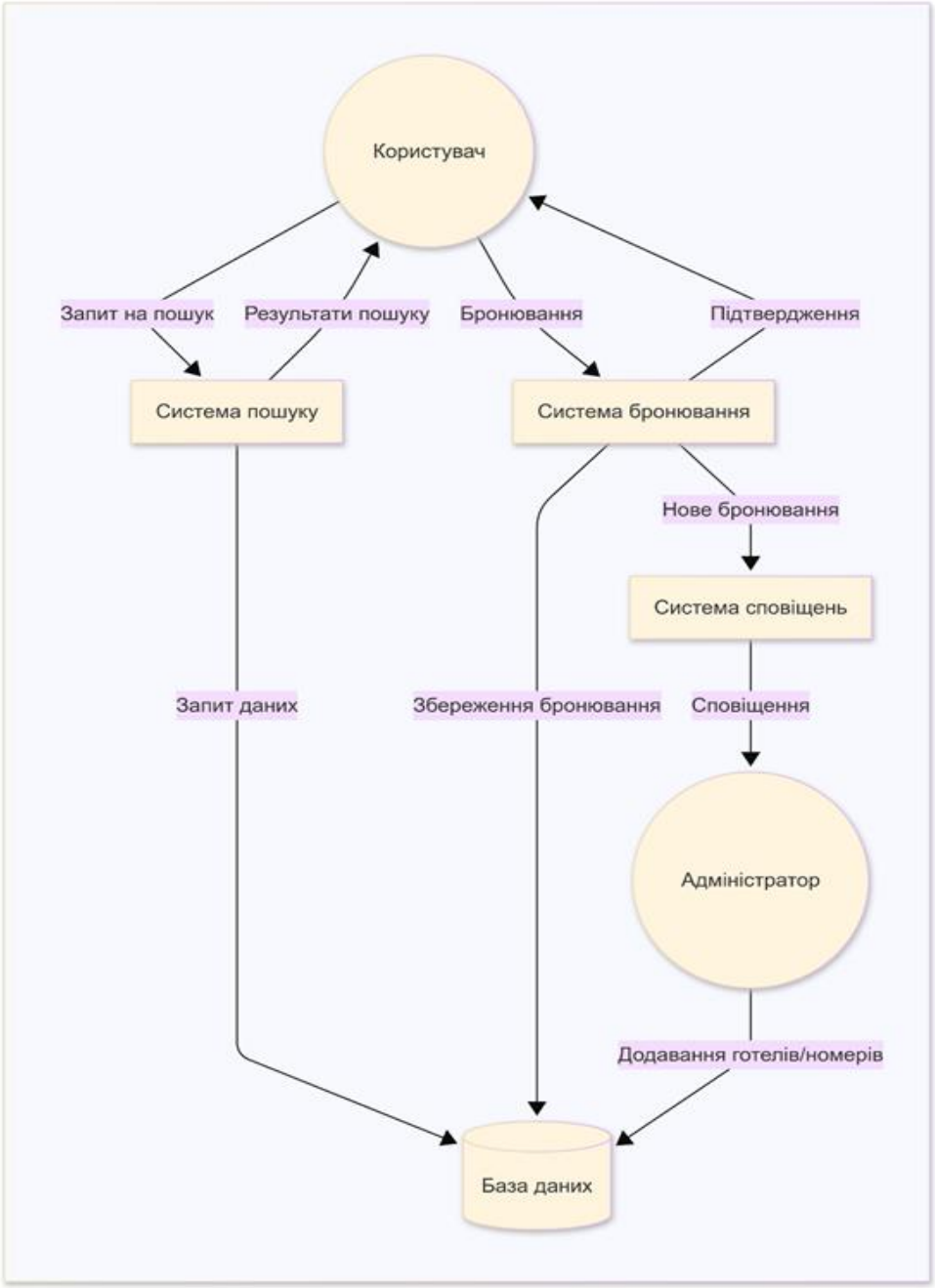


Рисунок 1.6 – Діаграма потоків даних в системі туристичного маркетплейсу

Таким чином, візуальне представлення функціональних вимог у вигляді діаграм суттєво доповнює текстовий опис вимог до системи, забезпечуючи більш комплексне розуміння її функціональності. Діаграми наочно демонструють основні сценарії використання системи різними категоріями користувачів та показують взаємодію між різними компонентами торгової онлайн-платформи на прикладі туристичного маркетплейсу.

Визначені функціональні вимоги формують основу для проєктування архітектури системи та розробки детальних специфікацій для кожного модуля торгової онлайн-платформи з адаптивним дизайном. Їх реалізація дозволить створити повноцінну платформу, яка відповідає потребам всіх зацікавлених сторін: клієнтів, продавців та адміністраторів систем

1.5 Висновки до першого розділу

У першому розділі було проведено комплексне дослідження теоретичних основ розробки торгових онлайн-платформ з адаптивним дизайном на прикладі туристичного маркетплейсу. Аналіз сучасних систем електронної комерції дозволив виявити їх основні типи, функціональні можливості та обмеження. Було встановлено, що сучасний ринок представлений різноманітними типами платформ, кожна з яких має свої переваги та недоліки.

Огляд існуючих рішень на ринку електронної комерції, включаючи такі платформи як Booking.com, Expedia, Airbnb та OtelMS.com на прикладі туристичної сфери, дозволив визначити їх сильні та слабкі сторони. Аналіз показав, що, незважаючи на значну кількість доступних платформ, існує потреба в системі, яка б поєднувала всі переваги існуючих рішень, мала якісний адаптивний дизайн та враховувала специфіку українського ринку.

Дослідження технологічних вимог до торгових онлайн-платформ виявило ключові аспекти, які необхідно враховувати при розробці сучасної системи: архітектурні рішення, вимоги до баз даних, безпеки, інтеграції з іншими системами, користувацького інтерфейсу, адаптивного дизайну, продуктивності та масштабованості.

На основі проведеного аналізу було сформульовано детальні функціональні вимоги. Ці вимоги охоплюють потреби всіх категорій користувачів системи: клієнтів та адміністраторів платформи.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Розробка архітектури системи

Туристичний маркетплейс представляє собою комплексне рішення для пошуку та бронювання готелів, побудоване на основі сучасної клієнт-серверної архітектури. Вона інтегрує передові технології веб-розробки з потужними алгоритмами пошуку та фільтрації для досягнення оптимальної продуктивності, надійності та зручності використання. Завдяки такому підходу, система здатна, оптимізувати результати відбору за різними параметрами та надавати користувачам зручний інтерфейс для взаємодії з усіма функціями системи.

2.1.1 Загальна архітектура системи

Архітектура маркетплейсу (рис 2.1) втілює модульний підхід до розробки, що забезпечує гнучкість і масштабованість системи.

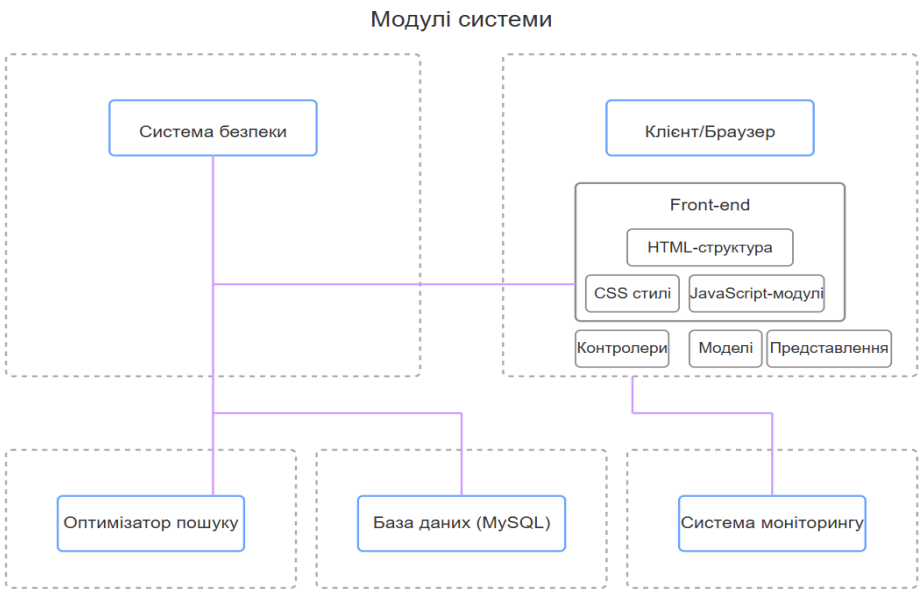


Рисунок 2.1 – Загальна архітектура системи

Система складається з ряду ключових компонентів, кожен з яких відіграє важливу роль у забезпеченні повноцінного функціонування платформи. Клієнтська частина (Front-end) відповідає за відображення інтерфейсу та організацію взаємодії з користувачем, реалізована з використанням HTML, CSS, JavaScript для створення інтуїтивно зрозумілого та адаптивного інтерфейсу. Без цього компонента неможлива ефективна взаємодія користувачів з системою та візуалізація важливих даних.

Серверна частина (Back-end) функціонує як головний координаційний центр, обробляючи запити від клієнтської частини та керуючи бізнес-логікою системи. Реалізована на PHP, що забезпечує обробку даних, взаємодію з базою даних та формування відповідей для клієнтської частини. Ця частина відповідає за всю логіку роботи системи та обробку даних.

База даних виступає сховищем інформації, забезпечує цілісність даних та швидкий доступ до них. Використовується MySQL, що забезпечує надійне зберігання та ефективний доступ до всіх даних системи, включаючи інформацію про користувачів, готелі, бронювання та параметри пошуку.

Підсистема моніторингу працює в режимі реального часу, забезпечуючи постійний нагляд за станом системи, включаючи доступність готелів, оновлення цін та наявність вільних номерів. Ця підсистема є критично важливою для оперативного реагування на будь-які зміни стану бронювань або непередбачувані ситуації.

Модуль оптимізації пошуку є ключовим інтелектуальним компонентом системи, який використовує складні алгоритми для визначення найбільш релевантних готелів з урахуванням заданих користувачем параметрів, таких як місце розташування, цінова категорія, рейтинг, наявність додаткових послуг та інших важливих факторів.

Підсистема безпеки пронизує всі рівні архітектури, застосовуючи багатошаровий підхід до захисту даних та функціональності від несанкціонованого доступу, кібератак та інших загроз. Вона включає аутентифікацію та авторизацію користувачів, шифрування даних, захист від SQL-ін’єкцій та інших типів атак, а також моніторинг підозрілої активності.

2.1.2 Архітектура клієнтської частини

Клієнтська частина (рис. 2.2) побудована за принципом динамічного веб-додатку, що забезпечує зручний та інтерактивний інтерфейс користувача.

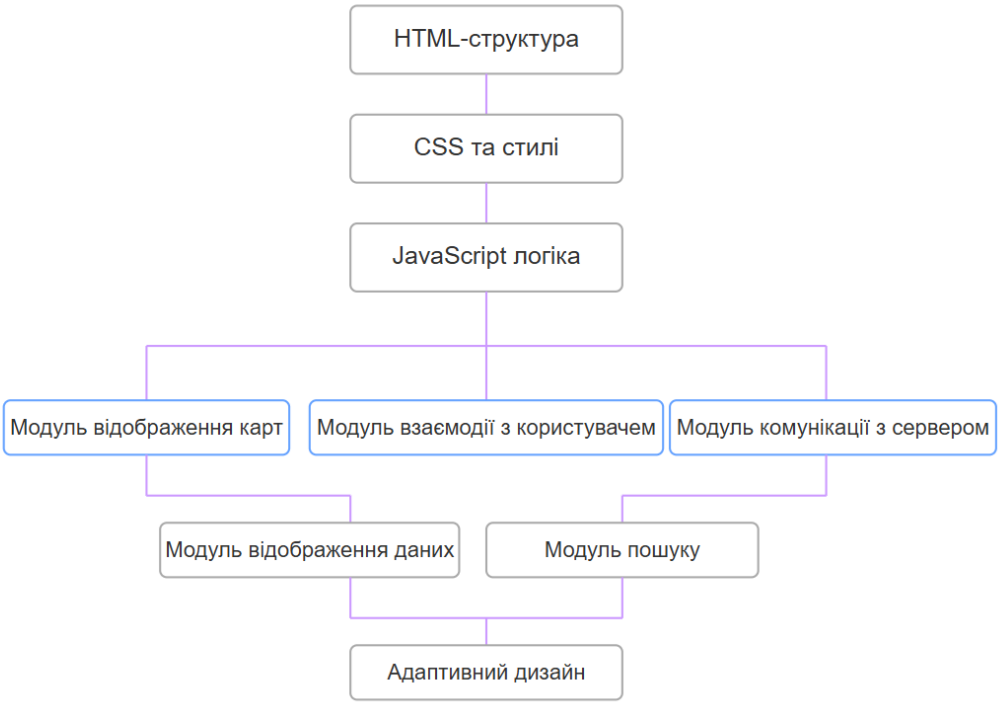


Рисунок 2.2 – Архітектура клієнтської частини

Клієнтська частина системи спроектована таким чином, щоб забезпечити максимально зручний та ефективний досвід користувача незалежно від пристрою, що використовується для доступу до системи. HTML-структура визначає основний каркас сторінок та розташування елементів інтерфейсу, забезпечуючи семантичну розмітку для кращої доступності та SEO-оптимізації. Вона реалізована з використанням сучасних практик HTML5, що дозволяє створити структуровану основу для всіх сторінок системи.

CSS стилі відповідають за візуальне оформлення інтерфейсу. Використання CSS дозволяє організувати стилі в модульну структуру,

спрощуючи їх підтримку та розширення. Це включає використання змінних для єдиної колірної схеми, міксинів для повторюваних елементів стилів, а також вкладеність селекторів для кращої читабельності коду. Такий підхід значно спрощує підтримку та модифікацію стилів інтерфейсу.

JavaScript-модулі забезпечують динамічну функціональність інтерфейсу, включаючи різноманітні компоненти для забезпечення повноцінної роботи клієнтської частини. Модуль відображення карт інтегрує географічні дані для відображення місцезнаходження готелів та туристичних атракцій, використовуючи сучасні картографічні API та бібліотеки. Модуль взаємодії з користувачем обробляє введені дані та дії користувача, забезпечуючи миттєвий відгук інтерфейсу на взаємодію. Модуль комунікації з сервером відповідає за відправку запитів та обробку відповідей від серверної. Модуль відображення даних представляє отримані від сервера дані у зручній для користувача формі, включаючи каталоги готелів, детальні сторінки, фотогалереї та системи фільтрації.

Адаптивний дизайн забезпечує коректне відображення інтерфейсу на пристроях з різними розмірами екранів, від мобільних телефонів до великих моніторів. Це досягається за допомогою медіа-запитів CSS, гнучких сіток та адаптивних зображень. Такий підхід дозволяє користувачам ефективно працювати з системою незалежно від використовуваного пристрою.

2.1.3 Архітектура серверної частини

Серверна частина системи (рис. 2.3) реалізована на PHP з використанням архітектурного патерну MVC (Model-View-Controller), що забезпечує чіткий розподіл відповідальності між різними компонентами та високу гнучкість системи.

Модель–вигляд–контролер – архітектурний шаблон, який використовується під час проєктування та розробки програмного забезпечення.



Рисунок 2.3 – Архітектура серверної частини

Цей шаблон ділить систему на три взаємопов’язані частини: модель даних, представлення (інтерфейс користувача) і модуль керування. Він використовується для відділення даних (моделі) від інтерфейсу користувача (перегляду), щоб мінімізувати вплив змін в інтерфейсі користувача на операції з даними, а модель даних можна змінити без зміни інтерфейсу користувача. Метою цього шаблону є досягнення гнучкого дизайну програмного забезпечення для полегшення подальших змін або розширень програми та забезпечення можливості повторного використання окремих компонентів програми. Крім того, використання цього шаблону у великих системах допомагає організувати його структуру та полегшити розуміння шляхом зменшення складності.

Архітектура серверної частини системи базується на паттерні MVC, що забезпечує високу модульність, гнучкість та підтримуваність коду. Контролери займають центральне місце в цій архітектурі, обробляючи запити від клієнтської частини, валідуючи вхідні дані та координуючи взаємодію між іншими компонентами системи. Кожен контролер відповідає за обробку

запитів, пов’язаних з конкретним функціональним модулем системи, таким як управління користувачами, готелями або бронюваннями.

Моделі містять бізнес-логіку системи та відповідають за взаємодію з базою даних. Модель користувачів управляє даними про користувачів системи, включаючи їхні профілі, ролі та права доступу. Модель готелів відповідає за управління даними про готелі, включаючи їхні характеристики, наявність номерів та цінову політику. Модель бронювань керує інформацією про бронювання, включаючи дати, типи номерів та статуси оплати.

Сервіси реалізують складну бізнес-логіку, яка може бути використана різними контролерами. Вони забезпечують повторне використання коду та спрощують підтримку системи. Сервіс пошуку відповідає за підбір готелів за параметрами з урахуванням різних факторів, таких як місце розташування, ціна, рейтинг та наявність додаткових послуг. Сервіс моніторингу забезпечує збір та аналіз даних про стан бронювань у реальному часі. Сервіс безпеки відповідає за аутентифікацію та авторизацію користувачів, а також за захист даних від несанкціонованого доступу.

Представлення (Views) відповідають за формування HTML-відповідей для клієнтської частини. Цей компонент забезпечує відокремлення логіки від презентації, що спрощує підтримку та розвиток системи.

Утиліти містять допоміжні функції та класи, які використовуються в різних частинах системи. Вони включають функції для валідації даних, форматування результатів та логування помилок та інших важливих подій в системі.

2.1.4 Вибір інструментів розробки

Для ефективної розробки та розгортання системи було обрано професійні інструменти, які забезпечують автоматизацію рутинних операцій та підвищують продуктивність розробки.

Visual Studio Code (VS Code) (рис. 2.4) став основним інструментом розробки для всіх учасників проектної команди. Цей потужний редактор коду від Microsoft обраний завдяки низці ключових переваг. Розширена система плагінів дозволила налаштувати середовище розробки під специфічні потреби проекту. Зокрема, використані розширення для підтримки JavaScript, React, TypeScript, які значно спростили процес написання та рефакторингу коду.



Рисунок 2.4 – VS Code

Вбудовані інструменти налагодження (debugging) VS Code забезпечили ефективну роботу з кодом, дозволяючи розробникам швидко ідентифікувати та виправляти потенційні помилки. Функція живого перезавантаження (live reload) суттєво прискорила процес розробки інтерфейсних компонентів.

Figma стала ключовим інструментом для дизайну інтерфейсу (рис 2.5). Цей Cloud-based інструмент для UX/UI проектування дозволив команді дизайнерів створити комплексний дизайн з інтерактивними прототипами. Collaborative функціонал Figma забезпечує ефективну комунікацію між дизайном та розробкою, дозволяючи миттєво синхронізувати бачення проекту.

Ключові переваги Figma для проекту включали:

– створення деталізованих дизайн-систем;

– генерація specs для розробників;
– швидке прототипування інтерфейсів;
– можливість спільної роботи в режимі реального часу.

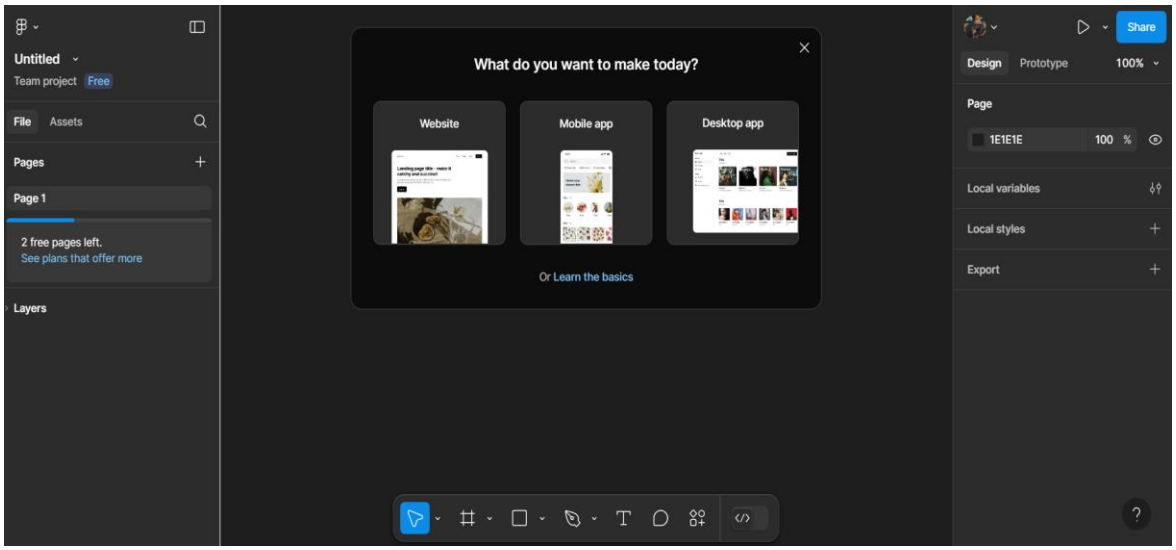


Рисунок 2.5 – Інтерфейс Figma

Використання цих інструментів дозволило забезпечити високу якість коду, спростити процес розробки та розгортання системи, а також зменшити кількість помилок та час на їх виправлення. Інтеграція VS Code та Figma створила ефективне середовище для розробки сучасного веб-додатку.

2.2 Проектування бази даних системи

Проектування бази даних є ключовим етапом у розробці інформаційної системи туристичного маркетплейсу, після ефективної організації даних функцій на продуктивність, надійність та масштабованість системи в усьому. Перед вибором конкретної системи управління базами даних (СУБД) був проведений порівняльний аналіз, який зображений в таблиці 2.1, найбільших розширених рішень на ринку: MySQL, PostgreSQL та MongoDB.

Таблиця 2.1 – Порівняльний аналіз системи управління базами даних

Критерій	MySQL	PostgreSQL	MongoDB
Тип СУБД	Реляційна	Реляційна	Документоорієнтована
Масштабованість	+	++	+++
Швидкість операцій читання	+++	++	+++
Швидкість операції запису	++	+	+++
Цілісність даних	+++	+++	+
Транзакційність	++	+++	+
Підтримка складних запитів	++	+++	+
Гнучкість схеми даних	+	++	+++
Зручність розгортання	+++	++	++
Підтримка спільноти	+++	++	++
Вартість обслуговування	+	++	++
Документація	+++	+++	++

На основі проведеного аналізу для реалізації бази даних туристичного маркетплейсу була вибрана СУБД MySQL, яка забезпечує оптимальну продуктивність підключення, надійність та зручності використання. Основними перевагами MySQL для даного проекту є:

- висока швидкість операції читання, що особливо важливо для системи пошуку покупок, де більшість запитів є саме операціями читання;
- хороша підтримка транзакцій, що забезпечує цілісність даних при операціях бронювання;
- широка підтримка спільноти та детальна документація, що пропонує розробку та системи підтримки;
- низька вартість обслуговування та простота розгортання.

Для організації даних у системі була розроблена реляційна модель, яка відображена на схемі бази даних і складається з наступних основних таблиць:

а) hotels – містить інформацію про готелі, включаючи:
   – id: унікальний ідентифікатор готелю (int);
   – name: назва готелю (varchar);
   – type: тип готелю – enum(‘resort’, ‘city’, ‘boutique’, ‘beach’);
   – address: адреса розташування (varchar);
   – city: місто розташування (varchar);
   – description: детальний опис готелю (text);
   – amenities: доступні зручності в форматі JSON;
   – image_url: посилання на зображення готелю (varchar);
   – created_at: дата створення запису (timestamp);

б) rooms – зберігає дані про номери в готелях:
   – id: унікальний ідентифікатор номера (int);
   – hotel_id: ідентифікатор готелю, до якого належить номер (int, зовнішній ключ);
   – name: назва або тип номера (varchar);
   – price: вартість проживання (decimal);
   – description: опис номера (text);
   – image_url: посилання на зображення номера (varchar);
   – created_at: дата створення запису (timestamp);

в) users – містить інформацію про користувачів системи:
   – id: унікальний ідентифікатор користувача (int);
   – name: ім’я користувача (varchar);
   – email: електронна пошта (varchar);
   – phone: номер телефону (varchar);
   – password: зашифрований пароль (varchar);
   – role: роль в системі (varchar);
   – registration_date: дата реєстрації (timestamp);

г) bookings – зберігає дані про бронювання:
   – id: унікальний ідентифікатор бронювання (int);

- user_id: ідентифікатор користувача, який здійснив бронювання (int, зовнішній ключ);
- hotel_id: ідентифікатор готелю (int, зовнішній ключ);
- room_id: ідентифікатор номера (int, зовнішній ключ);
- check_in: дата заїзду (date);
- check_out: дата виїзду (date);
- created_at: дата створення бронювання (timestamp);

д) reviews – зберігає відгуки користувачів про готелі:

- id: унікальний ідентифікатор відгуку (int);
- user_id: ідентифікатор користувача, який залишив відгук (int, зовнішній ключ);
- hotel_id: ідентифікатор готелю, для якого написаний відгук (int, зовнішній ключ);
- rating: рейтинг готелю (int);
- review: текст відгуку (text);
- created_at: дата створення відгуку (timestamp).

Схема бази даних (рис. 2.7) включає відповідні зв’язки між таблицями через зовнішні ключі, що забезпечують цілісність даних та можливість ефективного виконання складних запитів. Наприклад:

- зв’язок між таблицями hotels і rooms через поле hotel_id;
- зв’язок між таблицями users і bookings через поле user_id;
- зв’язок між таблицями hotels і reviews через поле hotel_id;
- зв’язок між таблицями users і reviews через поле user_id;
- складний зв’язок між таблицями bookings, rooms і hotels.

Для забезпечення ефективної роботи з даними створені індекси для полів, що часто використовуються в запитах: id користувачів, готелів та номерів, а також дати бронювання, що значно прискорює операції пошуку та фільтрації.

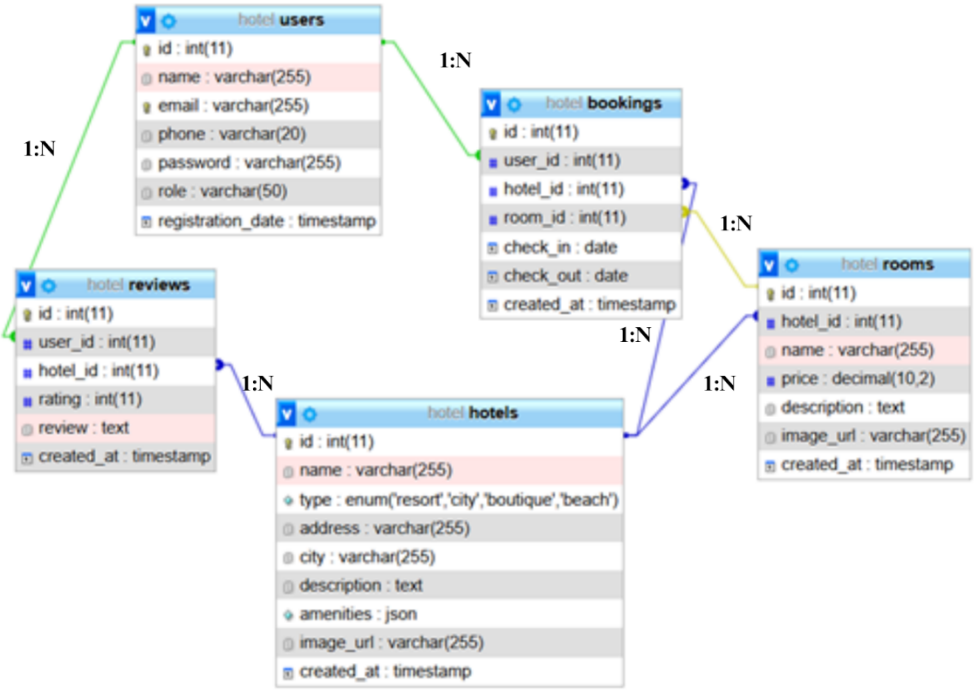


Рисунок 2.7 – Структура бази даних

Таким чином, розроблена структура бази даних повністю відповідає функціональним вимогам до інформаційної системи туристичного маркетплейсу та створює надійну основу для реалізації всіх функцій системи.

2.3 Проектування функціональної структури системи

Функціональна структура інформаційної системи туристичного маркетплейсу визначає основні модулі та підсистеми, які забезпечують виконання всіх функціональних вимог. Така структуризація дозволяє ефективно організувати розробку, тестування та подальшу підтримку системи (рис. 2.8).

Інформаційна система туристичного маркетплейсу складається з наступних функціональних модулів, розподілених між front-end та back-end частинами:

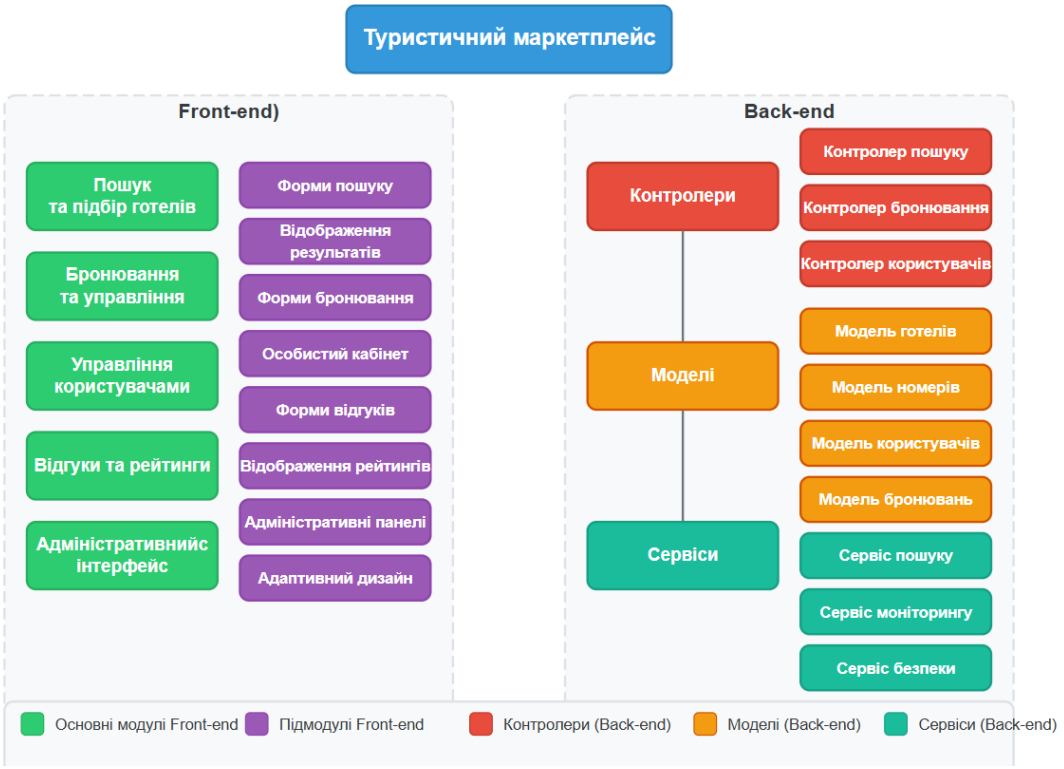


Рисунок 2.8 – Діаграма структури модулів системи

Модуль пошуку та підбору готелів є ключовим компонентом системи, який реалізує основну функціональність для кінцевих користувачів. На front-end частині цей модуль представлений інтерфейсом з формами пошуку та компонентами відображення результатів. На back-end частині модуль включає логіку обробки запитів пошуку та доступ до даних про готелі. Функціонально модуль охоплює підсистему базового пошуку за основними параметрами, підсистему розширеного пошуку з додатковими фільтрами, підсистему інтелектуального підбору та підсистему персоналізованих рекомендацій.

Модуль бронювання та управління резерваціями забезпечує процес створення та управління бронюваннями. Front-end частина містить інтерфейси для вибору номерів та заповнення форм бронювання, тоді як back-end частина включає логіку обробки запитів на бронювання та роботу з даними про резервації. Модуль реалізує підсистему створення нового бронювання та підсистему управління існуючими бронюваннями.

Модуль управління користувачами відповідає за реєстрацію, автентифікацію та управління профілями. На front-end представлений

формами входу, реєстрації та інтерфейсом особистого кабінету. На back-end реалізується обробка даних користувачів та управління сесіями. Модуль включає підсистему реєстрації та входу, підсистему управління профілем, підсистему ролей та дозволів, а також підсистему історії дій користувача.

Модуль управління готелями призначений для адміністраторів та власників готелів. Front-end включає адміністративні інтерфейси для управління інформацією про готелі, а back-end забезпечує обробку та зберігання цих даних. Функціонально модуль включає підсистему створення та редагування інформації про готель, підсистему управління номерами, підсистему управління цінами та доступністю, а також підсистему аналітики.

Модуль відгуків та рейтингів забезпечує функціональність для збору та відображення відгуків користувачів. Front-end представлений інтерфейсами для перегляду та додавання відгуків, back-end забезпечує обробку та зберігання відгуків. Модуль включає підсистему додавання нових відгуків, підсистему модерації відгуків та підсистему розрахунку та відображення рейтингів.

Модуль адміністрування системи призначений для адміністраторів платформи і забезпечує управління системою в цілому. Front-end складається з панелі адміністратора, а back-end реалізує логіку управління системою. Модуль включає підсистему управління користувачами та їх ролями, підсистему налаштування параметрів системи та підсистему моніторингу активності.

Модуль комунікацій забезпечує взаємодію між користувачами системи. Front-end представлений інтерфейсами для обміну повідомленнями, а back-end обробляє та зберігає комунікації.

Модуль інтеграції із зовнішніми системами відповідає за взаємодію з іншими платформами та сервісами. Реалізується переважно на back-end через взаємодію з зовнішніми API. Модуль включає підсистему інтеграції з сервісами геолокації та підсистему інтеграції з соціальними мережами.

На діаграмі взаємозв’язків (рис. 2.9) видно, що front-end компоненти взаємодіють з back-end компонентами через API запити. Back-end компоненти включають обробники запитів та логіку доступу до даних. Така структура відповідає описаній раніше архітектурі системи, забезпечуючи чіткий розподіл відповідальності між компонентами.

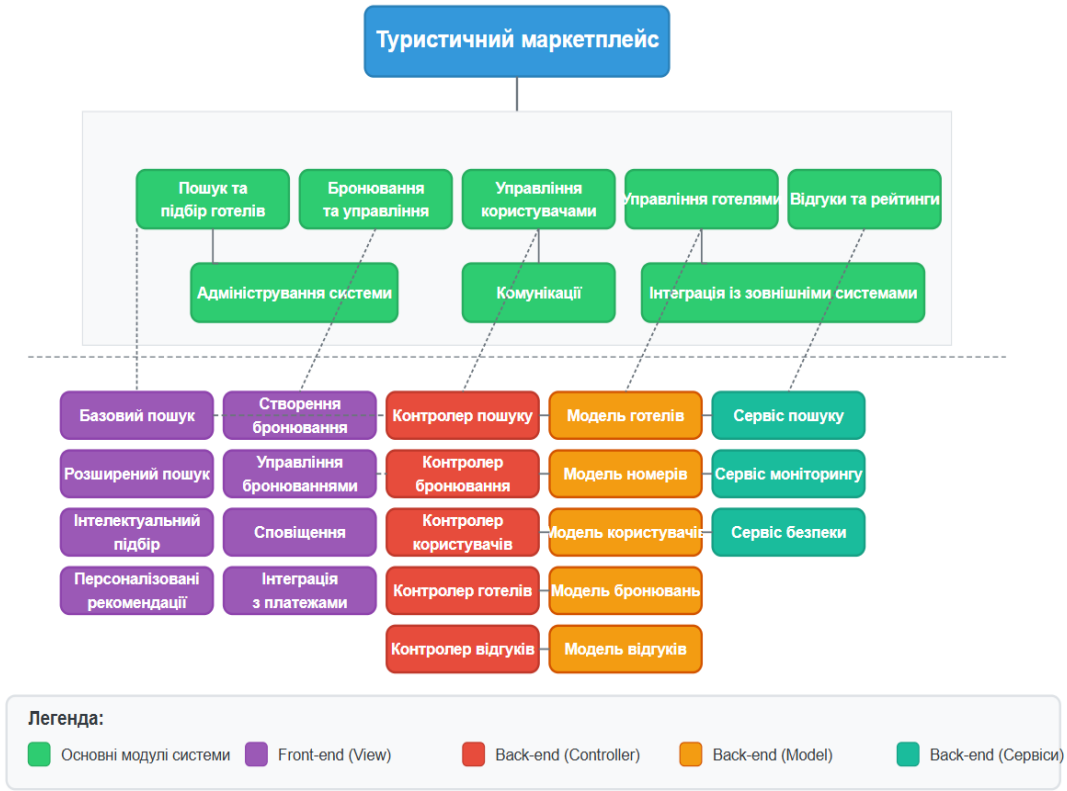


Рисунок 2.9 – Діаграма взаємозв’язків між модулями

Back-end частина системи включає:

– обробники запитів, які отримують дані від front-end, валідують їх та координують виконання операцій;
– бізнес-логіку, яка реалізує функціональність системи;
– доступ до бази даних для збереження та отримання інформації;
– сервіси пошуку, моніторингу та безпеки, які забезпечують додаткову функціональність.

Для забезпечення високої якості та надійності системи кожен модуль проходить окреме тестування, а також інтеграційне тестування у взаємодії з

іншими модулями. Це дозволяє виявити та усунути проблеми на ранніх стадіях розробки.

Таким чином, спроектована функціональна структура інформаційної системи туристичного маркетплейсу забезпечує повну реалізацію всіх визначених функціональних вимог та створює надійну основу для ефективної розробки та подальшого розвитку системи. Розподіл компонентів на front-end та back-end узгоджується з описаною архітектурою та забезпечує модульність, масштабованість та підтримуваність системи.

2.4 Висновки до другого розділу

У другому розділі проведено проєктування інформаційної системи туристичного маркетплейсу з вбудованою системою підбору готелів. Розроблена архітектура забезпечує реалізацію функціональних вимог, визначених у першому розділі.

Загальна архітектура базується на клієнт-серверній моделі, включаючи Front-end, Back-end, базу даних, підсистеми моніторингу, оптимізації пошуку та безпеки. Така структура забезпечує чіткий розподіл функціональності між компонентами.

Клієнтська частина побудована як динамічний веб-додаток з адаптивним дизайном. Серверна частина реалізована за патерном MVC, що забезпечує гнучкість системи. Для бази даних обрано MySQL, розроблено оптимізовану структуру таблиць з необхідними індексами.

Функціональна структура включає модулі пошуку та підбору готелів, бронювання, управління користувачами та готелями, відгуків, платежів, адміністрування, комунікацій та інтеграції із зовнішніми системами.

Створене проєктне рішення охоплює всі аспекти розробки системи та забезпечує високу продуктивність, масштабованість, безпеку та зручність використання.

3 РЕАЛІЗАЦІЯ ТУРИСТИЧНОГО МАРКЕТПЛЕЙСУ

3.1 Структура файлів проєкту

Структура файлів (див. рис. 3.1) проєкту відображає використання архітектурного патерну MVC та забезпечує модульність і масштабованість системи.



Рисунок 3.1 – Структура файлів проєкту

Основні компоненти структури проєкту:

– index.php – точка входу в додаток, реалізує маршрутизацію запитів
– model/ – містить файли моделей для роботи з даними
– controller/ – містить контролери для обробки запитів
– views/ – містить представлення для відображення даних
– templates/ – шаблони, спільні для різних сторінок (навігація, футер)
– DB/ – скрипти для роботи з базою даних

Така структура забезпечує чіткий розподіл відповідальності між компонентами системи та спрощує її подальшу підтримку та розвиток.

3.2 Розробка серверної частини системи

Серверна частина туристичного маркетплейсу реалізована з використанням PHP, що забезпечує чіткий розподіл відповідальності між компонентами системи та підвищує її гнучкість і масштабованість.

3.2.1 Структура маршрутизації та обробка запитів

Основною точкою входу до системи є файл index.php, який виконує функцію маршрутизатора (див рис. 3.2). Він аналізує вхідний URI запиту та перенаправляє його до відповідного контролера.

```
session_start();

require_once './model/model.php';
require_once './controller/controller.php';

$uri = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);

if ('/' === $uri) {
    mainPage();
    return:
```

Рисунок 3.2 – Код маршрутизатора

Такий підхід дозволяє організувати логічне розділення функціональності та забезпечити гнучкість у розвитку системи. Для кожної сторінки або функціонального блоку визначено окремий маршрут та відповідний контролер, що відповідає принципам MVC архітектури.

Система маршрутизації розділяє запити за ролями користувачів. Наприклад, для доступу до адміністративних функцій використовується функція перевірки ролі (див. рис. 3.3).

Ця функція перевіряє, чи має поточний користувач необхідну роль, і в разі відсутності прав доступу перенаправляє на сторінку авторизації.

```
function checkUserRole($required_role, $url_page) {
    if (!isset($_SESSION['user_role']) || $_SESSION['user_role'] !==
$required_role) {
        header('Location: ' . $url_page);
        exit();
```

Рисунок 3.3 – Функція перевірки ролі

3.2.2 Взаємодія з базою даних

Для зберігання та управління даними в системі використовується реляційна база даних MySQL. Взаємодія з базою даних реалізована через модельний шар відповідно до архітектури MVC. Кожна модель відповідає за роботу з певною сутністю системи (готелі, номери, користувачі, бронювання).

Наприклад, для отримання списку готелів використовується відповідна функція в моделі (див. рис.3.4):

```
function getHotels($limit = null, $city = '', $room_type = '')
{
    $connect = db_connect();

    // Базовий SQL-запит
    $query = "SELECT DISTINCT h.* FROM hotels h";

    // Додати умови для фільтрації
    $conditions = [];

    if (!empty($city)) {
        $conditions[] = "h.city LIKE '%" . mysqli_real_escape_string($connect,
$city) . "%'";
    }

    if (!empty($room_type)) {
        $query .= " JOIN rooms r ON h.id = r.hotel_id";
        $conditions[] = "r.name = '" . mysqli_real_escape_string($connect,
$room_type) . "'";
    }
```

Рисунок 3.4 – Модель отримання списку готелів

А для додавання нового готелю (див. рис 3.5).

```
function saveHotel($name, $type, $adress, $city, $description, $amenities,
$fileUpload)
{
    $connect = db_connect();

    $query = "INSERT INTO hotels (name, type, address, city, description,
amenities,image_url) VALUES ('$name', '$type', '$adress', '$city', '$description',
'$amenities', '$fileUpload')";
    return mysqli_query($connect, $query);
}
```

Рисунок 3.5 – Модель додавання нового готелю

Для забезпечення безпеки даних використовуються підготовлені запити (prepared statements), що захищають від SQL-ін’єкцій та інших видів атак.

3.2.3 Реалізація контролерів

Контролери в системі виконують функцію посередника між моделями та представленнями. Вони обробляють запити користувачів, взаємодіють з моделями для отримання або зміни даних та передають результати до представлень для формування відповіді.

Розглянемо приклад контролера для керування готелями адміністратором(див. рис 3.6).

```
function deleteHotel($id)
{
    $connect = db_connect();

    $query = "DELETE FROM hotels WHERE id = $id";
    return mysqli_query($connect, $query);
}
```

Рисунок 3.6 – Приклад контролера для керування готелями

Такий підхід дозволяє чітко розділити відповідальність між різними компонентами системи та забезпечити її гнучкість і масштабованість.

3.2.4 Управління сесіями та аутентифікація

Для забезпечення персоналізованого досвіду користувачів та захисту даних у системі реалізовано механізм сесій та аутентифікації. Сесія ініціалізується на початку виконання скрипту:

```
session_start();
```

Для авторизації користувачів використовується механізм порівняння введених даних з інформацією в базі даних з використанням безпечного хешування паролів (див. рис.3.7).

```
function profileUser()
{
    $user_id = $_SESSION['user_id'];
    $profile = getUserById($user_id);
    $bookings = getBookingsByUserId($user_id);
    // var_dump($bookings);
    $profile = getUserById($user_id);
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $name = $_POST['first-name'];
        $email = $_POST['email'];
        $phone = $_POST['phone'];

        updateUser($user_id, $name, $email, $phone);
        header('Location: ' . $_SERVER['HTTP_REFERER']);
    }
    require_once './views/profile.php';
}
```

Рисунок 3.7 – Код авторизації користувачів

Такий підхід дозволяє зберігати стан користувача між запитами та забезпечувати доступ до функціональності відповідно до його ролі.

3.3 Розробка клієнтської частини системи

Клієнтська частина туристичного маркетплейсу реалізована з використанням сучасних веб-технологій для забезпечення зручності та інтерактивності інтерфейсу користувача.

3.3.1 Структура інтерфейсу користувача

Інтерфейс користувача побудовано на основі HTML5 з використанням CSS для стилізації та JavaScript для забезпечення інтерактивності. Для спрощення розробки та підтримки використовується фреймворк Tailwind CSS, що дозволяє створювати адаптивний дизайн за допомогою утилітарних класів.

Приклад структури сторінки адміністративної панелі(див. рис. 3.8).

```
<div class="min-h-screen flex flex-col justify-between">
    <?php require_once './templates/nav.php'; ?>

    <!-- Main Content -->
    <div class="w-full max-w-7xl mx-auto py-6 sm:px-6 lg:px-8">
        <!-- Header -->
        <div class="px-4 py-5 sm:px-0 flex justify-between items-center">
            <h1 class="text-2xl font-semibold text-gray-900">Управління
Готелями</h1>
            <a href="/admin/create/hotel"
                class="bg-blue-600 text-white px-4 py-2 rounded-md hover:bg-
blue-700 focus:outline-none focus:ring-2 focus:ring-blue-500 focus:ring-offset-2">
                <i class="fas fa-plus mr-2"></i>
                Додати готель
            </a>
<div class="mt-8 bg-white shadow rounded-lg overflow-hidden">
            <?php if (empty($hotels)): ?>
                <div class="p-6 text-center text-gray-500">
                    <p>Наразі немає жодного готелю.</p>
                </div>
```

Рисунок 3.8 – Структура сторінки адміністративної панелі

Такий підхід забезпечує єдиний стиль для всіх сторінок системи та спрощує підтримку і розвиток інтерфейсу.

3.3.2 Модульна організація представлень

Для забезпечення повторного використання коду та спрощення підтримки інтерфейсу користувача використовується модульна організація представлень. Спільні елементи інтерфейсу, такі як навігаційне меню та підвал сторінки, виокремлені в окремі файли та підключаються до кожної сторінки(див. рис.3.9).

```
php
<?php require_once './templates/nav.php'; ?>
<!-- Main content -->
<?php require_once './templates/footer.php'; ?>
```

Рисунок 3.9 – Підключення окремих файлів

Такий підхід дозволяє уникнути дублювання коду та забезпечити єдиний стиль для всіх сторінок системи.

3.3.3 Використання JavaScript для інтерактивності

Для забезпечення інтерактивності інтерфейсу користувача використовується JavaScript. Наприклад, для пошуку готелів за параметрами реалізовано динамічну форму пошуку, яка дозволяє користувачам вказати локацію, тип номеру, дати заїзду та виїзду (див. рис.3.10).

JavaScript також використовується для валідації форм перед відправкою даних на сервер, що покращує досвід користувача та зменшує навантаження на сервер (див. рис 3.11):

```
<form id="searchForm" class="space-y-4">
    <div class="grid grid-cols-1 md:grid-cols-2 gap-4">
        <div>
            <label for="location" class="block text-sm font-medium text-gray-
700">Локація</label>
            <input type="text" id="location" name="location" class="mt-1 block w-
full rounded-md border-gray-300 shadow-sm">
        </div>
        <div>
            <label for="roomType" class="block text-sm font-medium text-gray-
700">Тип номеру</label>
            <select id="roomType" name="roomType" class="mt-1 block w-full rounded-
md border-gray-300 shadow-sm">
                <option value="">Будь-який</option>
                <option value="standard">Стандарт</option>
                <option value="family">Сімейний</option>
                <option value="luxury">Люкс</option>
                <option value="presidential">Презедентський</option>
            </select>
        </div>
    </div>
    <div class="grid grid-cols-1 md:grid-cols-2 gap-4">
        <div>
            <label for="checkIn" class="block text-sm font-medium text-gray-
700">Дата заїзду</label>
            <input type="date" id="checkIn" name="checkIn" class="mt-1 block w-full
rounded-md border-gray-300 shadow-sm">
        </div>
        <div>
            <label for="checkOut" class="block text-sm font-medium text-gray-
700">Дата виїзду</label>
            <input type="date" id="checkOut" name="checkOut" class="mt-1 block w-
full rounded-md border-gray-300 shadow-sm">
        </div>
    </div>
    <button type="submit" class="w-full bg-blue-600 text-white py-2 px-4 rounded-md
hover:bg-blue-700">Знайти готелі</button>
</form>
```

Рисунок 3.10 – Код пошуку готелів за параметрами

```
document.getElementById('searchForm').addEventListener('submit', function(event) {
    const checkIn = document.getElementById('checkIn').value;
    const checkOut = document.getElementById('checkOut').value;

    if (checkIn && checkOut && new Date(checkIn) >= new Date(checkOut)) {
        event.preventDefault();
        alert('Дата виїзду повинна бути пізніше дати заїзду');
    }
});
```

Рисунок 3.11 – Код валідації форм

3.3.4 Адаптивний дизайн

Для забезпечення зручності використання системи на різних пристроях реалізовано адаптивний дизайн. Використання фреймворку Tailwind CSS дозволяє легко створювати інтерфейс, який автоматично адаптується до різних розмірів екрану (див. рис.3.12).

```
<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-4">
    <!-- Card content -->
</div>
```

Рисунок 3.12 – фреймворку Tailwind CSS

У наведеному прикладі на малих екранах картки відображатимуться в один стовпець, на середніх – у два, а на великих – у чотири стовпці. Такий підхід забезпечує оптимальне використання доступного простору на екрані та зручність використання системи на різних пристроях.

Такий підхід забезпечує зручний та інтуїтивно зрозумілий інтерфейс для управління системою.

Таким чином, клієнтська частина туристичного маркетплейсу реалізована з використанням сучасних веб-технологій та принципів розробки для забезпечення зручності, інтерактивності та адаптивності інтерфейсу користувача. Використання модульної організації представлень та фреймворку Tailwind CSS спрощує підтримку та розвиток системи.

3.4 Реалізація основних функціональних можливостей

Розглянемо реалізацію основних функціональних можливостей туристичного маркетплейсу HotelBook з вбудованою системою підбору готелів за параметрами. Система розроблена з урахуванням потреб двох основних типів користувачів: клієнтів та адміністраторів.

3.4.1 Інтерфейс для клієнтів

Клієнтський інтерфейс є основним компонентом системи, через який кінцеві користувачі взаємодіють з маркетплейсом при пошуку та бронюванні готелів. Головна сторінка (рис. 3.13) включає логотип і назву сервісу HotelBook, головне навігаційне меню з розділами "Головна", "Готелі", "Загальна інформація", "Правила та умови", а також кнопки для входу та реєстрації у системі.

Центральним елементом головної сторінки є великий банер із зображенням готелю та заголовком "Знайдіть Свій Ідеальний Готель", що одразу транслює основну цінність сервісу. Під банером розміщений текстовий блок із коротким описом переваг послуги: "Забронюйте найкращі готелі та номери за найбільш конкурентоспроможними цінами. Насолоджуйтесь відпочинком у нашій преміальній підборці помешкань". Для швидкого доступу до основних функцій на головній сторінці розміщені кнопки "Пошук готелів" та "Дізнатися більше".

Для використання повного функціоналу маркетплейсу користувачі повинні пройти процедуру реєстрації (рис. 3.14). Форма реєстрації включає поля для введення основної інформації: ім’я, електронної пошти, паролю та підтвердження паролю. Система передбачає валідацію введених даних для забезпечення безпеки облікових записів.

Зареєстровані користувачі можуть входити в систему через форму авторизації (рис. 3.14), що запитує електронну пошту та пароль. Для зручності передбачено функцію "Запам’ятати мене" та відновлення забутого паролю.

Після авторизації користувач отримує доступ до особистого кабінету (рис. 3.15), де може управляти своїм профілем та переглядати історію взаємодій з системою. Розділ "Особиста інформація" дозволяє редагувати основні дані профілю: ім’я, електронну пошту та телефон. Внесені зміни зберігаються після натискання кнопки "Зберегти зміни".

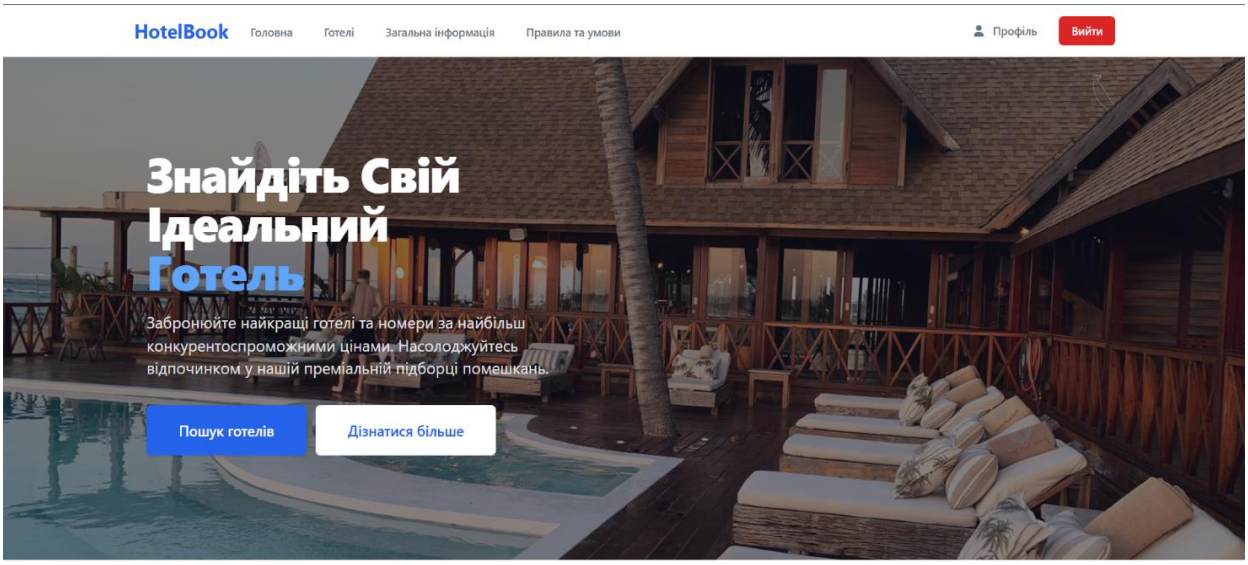


Рисунок 3.13 – Головна сторінка сайту

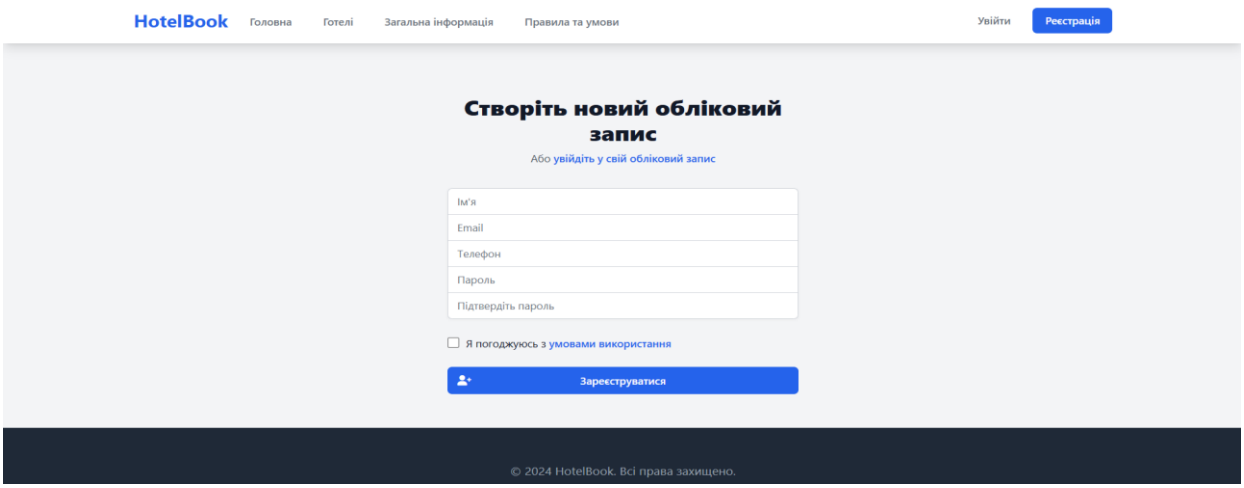


Рисунок 3.14 – Форми реєстрації

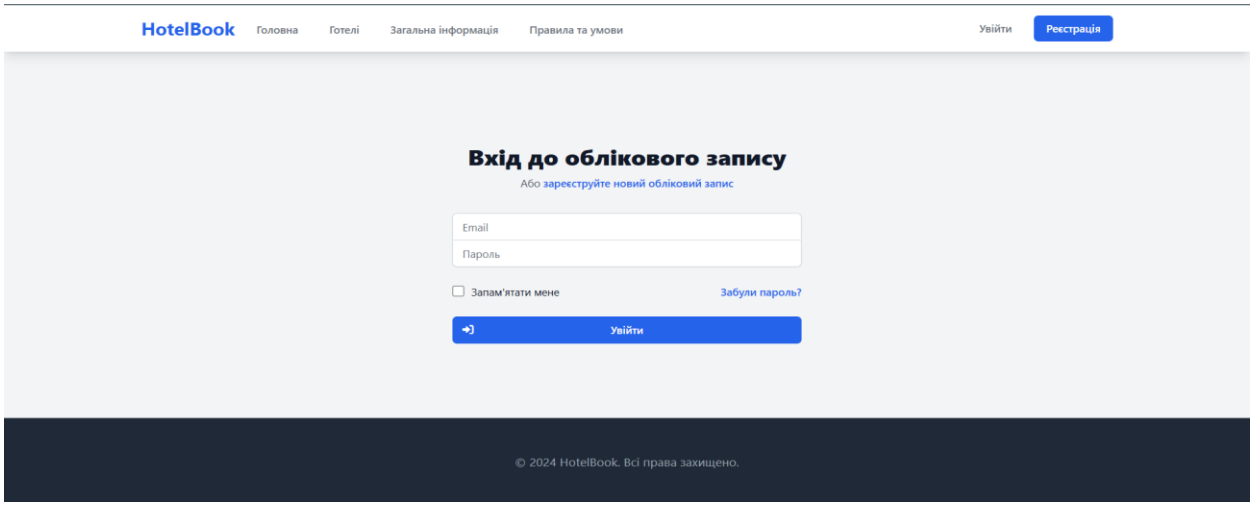


Рисунок 3.15 – Форми авторизації

Важливою частиною особистого кабінету (рис. 3.16) є розділ "Останні бронювання", де користувач може переглядати історію своїх бронювань готелів, їх статус та деталі. Якщо користувач ще не здійснював бронювань, система відображає повідомлення "У вас ще немає бронювань".

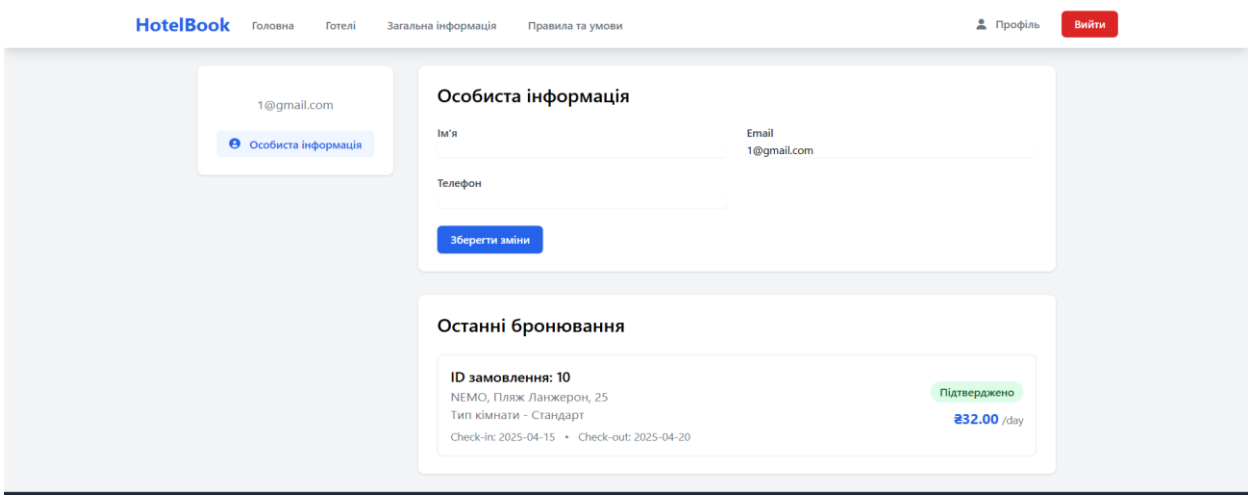


Рисунок 3.16 – Особистий кабінету користувача

Функціональність пошуку готелів є ключовим елементом маркетплейсу, що дозволяє користувачам швидко знайти оптимальний варіант розміщення відповідно до їхніх потреб. Інтерфейс пошуку включає розширені можливості фільтрації за різними параметрами:

– локація (місто, регіон або конкретний район);
– тип номеру з вибором категорій: Будь-який, Стандарт, Сімейний, Люкс, Президентський;
– дата заїзду з інтерактивним календарем;
– дата виїзду з інтерактивним календарем.

Система пошуку (рис. 3.17) працює в режимі реального часу, оновлюючи результати відповідно до встановлених фільтрів. Це дозволяє користувачам швидко коригувати параметри пошуку та миттєво бачити нові результати.

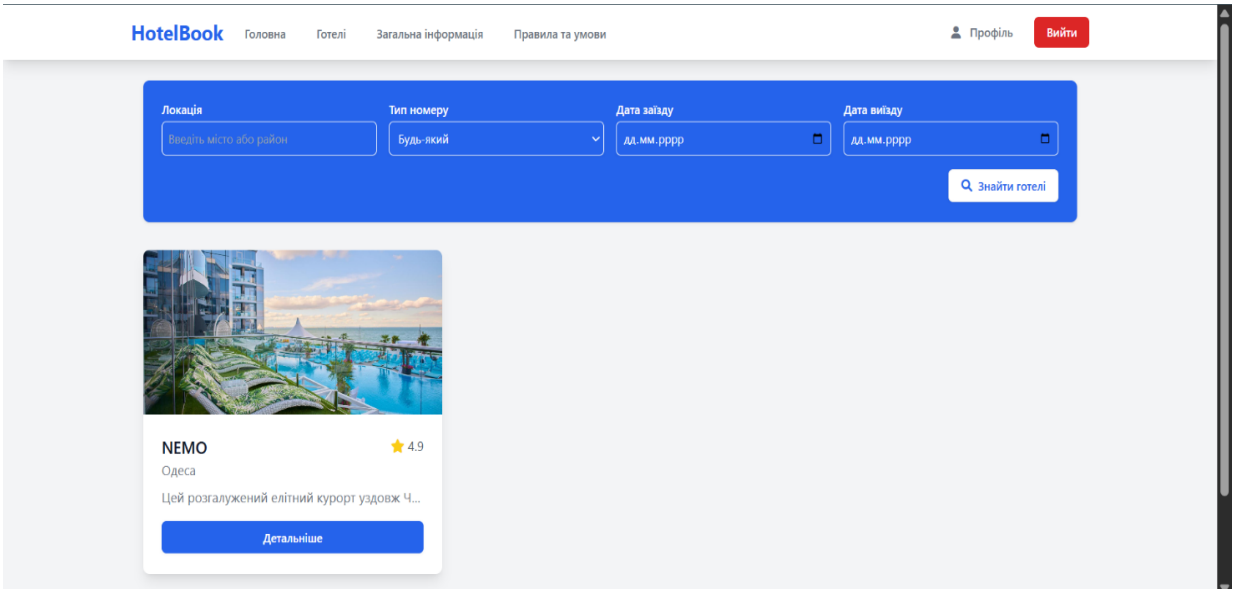


Рисунок 3.17 – Система пошуку та фільтрації

Результати пошуку (рис.3.18) відображаються у вигляді карток готелів, які містять:

– назву і фотографію готелю;
– рейтинг і кількість відгуків від попередніх гостей;
– коротку інформацію про розташування;
– кнопку для переходу до детальної інформації.

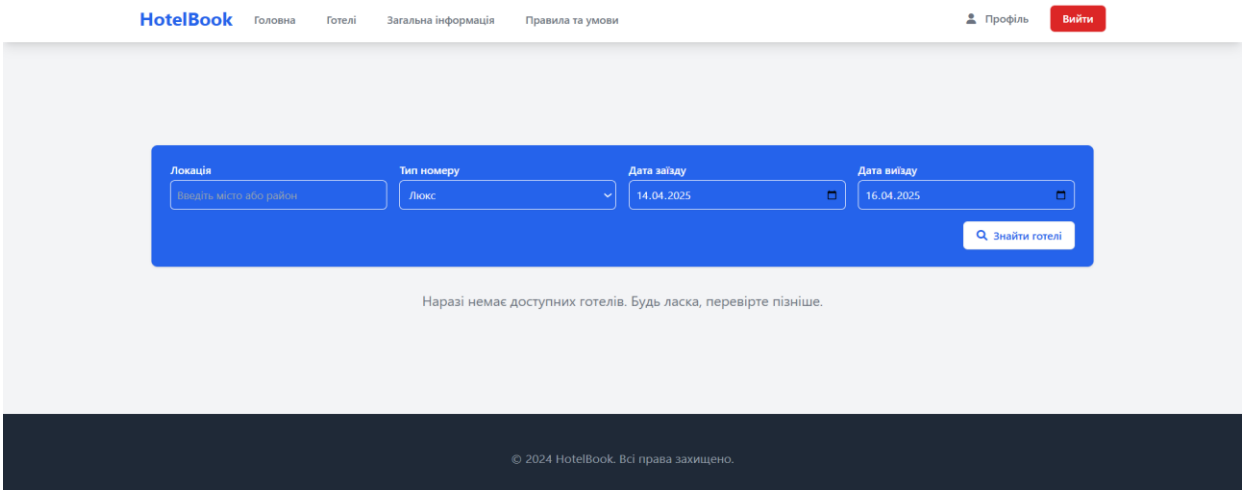


Рисунок 3.18 – Результати пошуку

Система бронювання номерів реалізована у вигляді інтуїтивно зрозумілого інтерфейсу, що дозволяє користувачеві обрати конкретний номер у вибраному готелі. На сторінці готелю представлено детальну інформацію про доступні номери з описами, фотографіями та цінами (рис.3.19).

Для кожного типу номера (Стандарт, Сімейний, Люкс, Президентський) надається інформація про:

– площу номера;
– кількість та тип ліжок;
– максимальну кількість гостей;
– наявні зручності (кондиціонер, телевізор, міні-бар, сейф тощо);
– вартість за ніч.

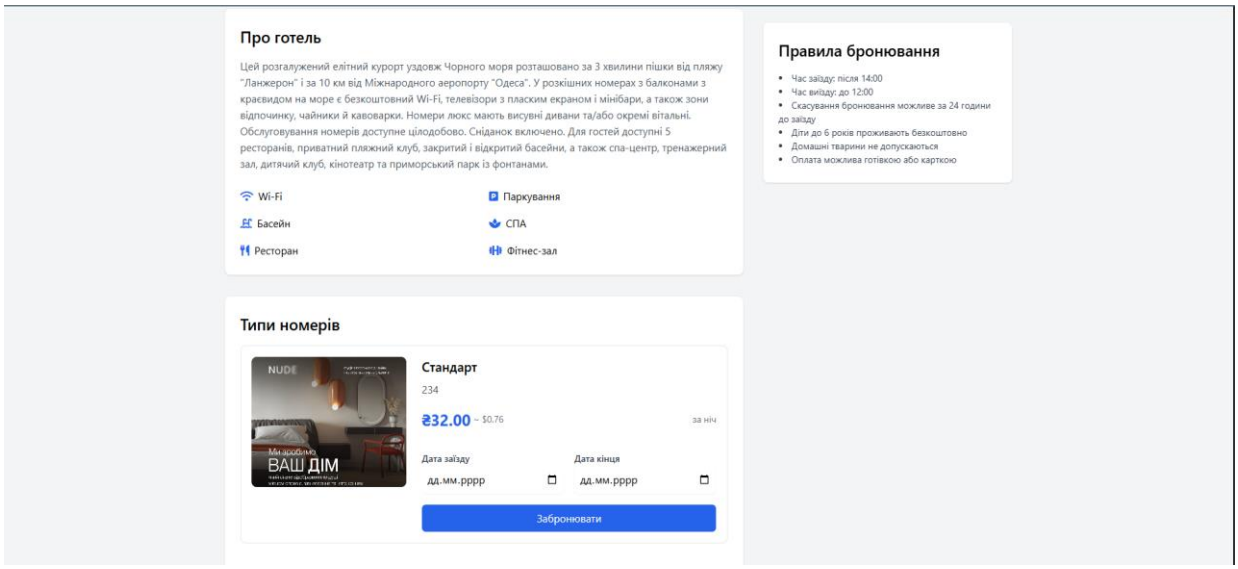


Рисунок 3.19 – Сторінка опису готелю

Процес бронювання реалізовано як послідовність кроків:

а) вибір дати заїзду та виїзду через календар;
б) ознайомлення з правилами бронювання;
в) підтвердження бронювання.

Правила бронювання чітко вказані на сторінці та включають інформацію про:

– час заїзду (після 14:00);

– час виїзду (до 12:00);
– можливість скасування бронювання за 24 години до заїзду;
– умови проживання дітей (діти до 6 років проживають безкоштовно);
– правила щодо домашніх тварин (не допускаються);
– способи оплати (готівкою або карткою).

Після завершення бронювання користувач отримує підтвердження з деталями резервації, яке також відображається в особистому кабінеті у розділі "Останні бронювання".

Важливим елементом маркетплейсу є система відгуків (рис. 3.20), що дозволяє користувачам залишати оцінки та коментарі про готелі, в яких вони зупинялися. Форма відгуку включає оцінювання за 5-бальною шкалою та текстове поле для детального опису досвіду проживання.

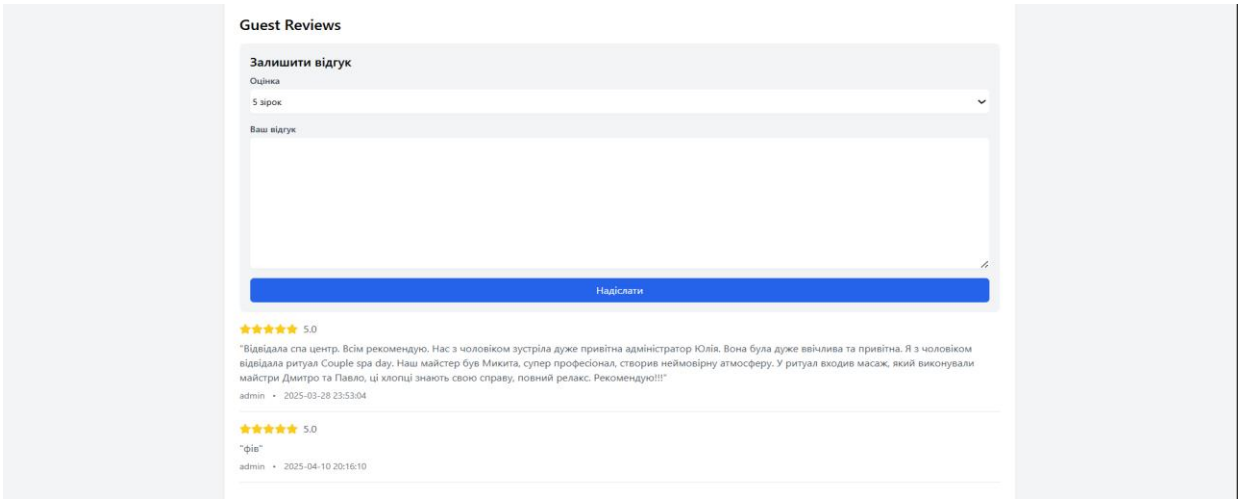


Рисунок 3.20 – Сторінка системи відгуків

Відгуки інших користувачів відображаються на сторінці готелю та допомагають потенційним гостям зробити більш інформований вибір. Система також розраховує середній рейтинг готелю на основі всіх отриманих оцінок.

3.4.2 Інтерфейс для адміністраторів

Адміністративний інтерфейс (рис. 3.21) надає доступ до управління всіма аспектами функціонування маркетплейсу. Панель адміністратора містить розширене меню навігації та включає функціональні модулі для ефективного керування контентом платформи.

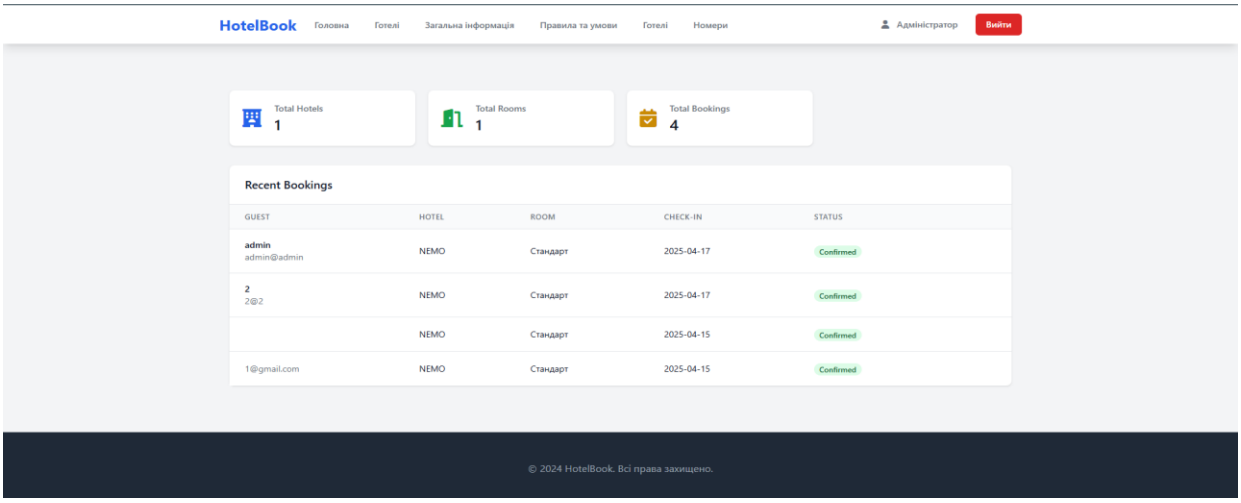


Рисунок 3.21 – Панель моніторингу адміністратора

Після авторизації адміністратор отримує доступ до спеціалізованої панелі управління, яка містить вкладку "Адміністратор" з оглядовою інформацією про стан системи:

– загальна кількість готелів у системі;
– загальна кількість номерів у всіх готелях;
– загальна кількість бронювань;
– таблиця останніх бронювань з інформацією про гостя, готель, номер, дату реєстрації та статус бронювання.

Ця інформація дозволяє адміністратору швидко оцінити поточний стан маркетплейсу та активність користувачів.

Адміністративний інтерфейс відображає всі доступні готелі (рис. 3.22) у вигляді зручної таблиці з колонками "Готель", "Місцезнаходження", "Статус",

"Номерів" та "Дії". Для кожного готелю доступна опція видалення, а також можливість додавання та управління номерами.

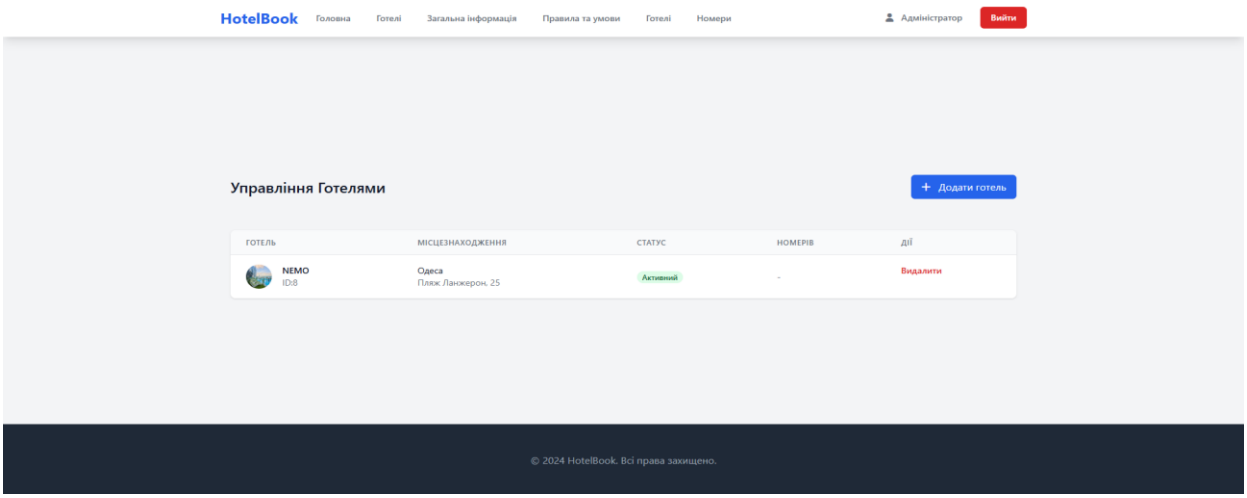


Рисунок 3.22 – Таблиця управління готелями

Функціональність додавання нових готелів є ключовою для адміністратора маркетплейсу (рис. 3.23). Система пропонує структурований інтерфейс для внесення всієї необхідної інформації про новий об’єкт розміщення, розділений на логічні блоки:

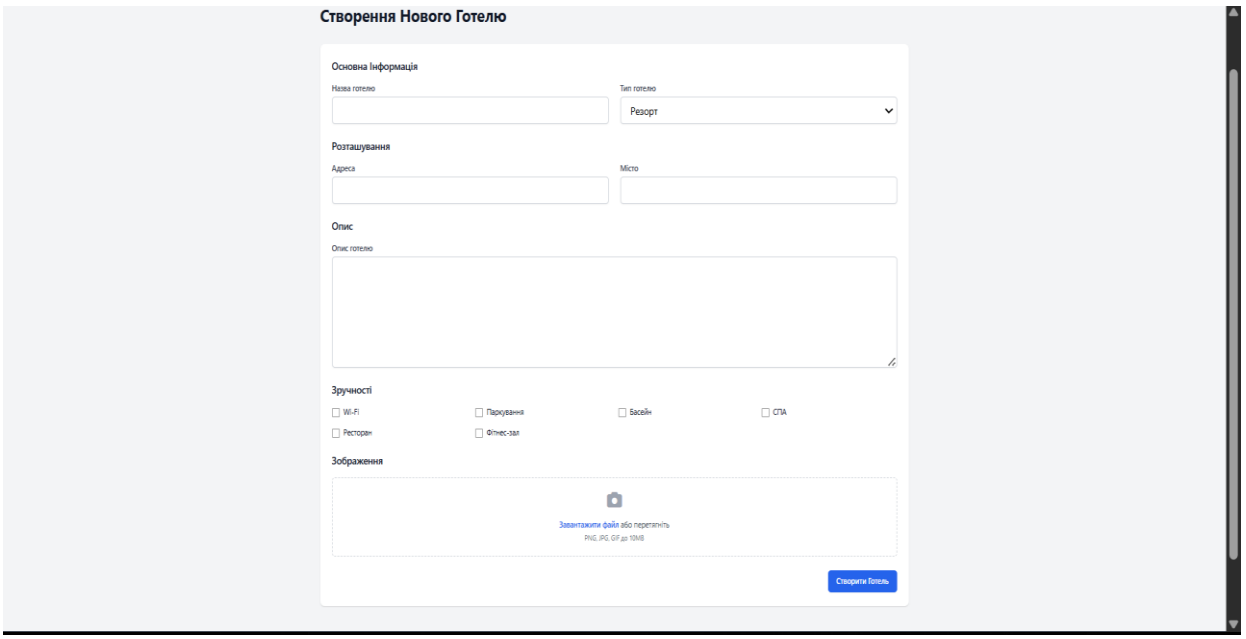


Рисунок 3.23 – Сторінка додавання нового готелю

Основна інформація:

– назва готелю (унікальна назва для ідентифікації в системі);
– тип готелю (вибір з варіантів: Резорт, Міський, Бутик, Пляжний);
– розташування (поля для введення точної адреси).

Опис готелю:

– розгорнутий опис особливостей та переваг готелю;
– інформація про історію, концепцію та унікальні пропозиції.

Інтерактивний список доступних зручностей з можливістю вибору:

– Wi-Fi;
– паркування;
– басейн;
– СПА;
– ресторан;
– фітнес-зал;
– кондиціонер;
– конференц-зали;
– трансфер з/до аеропорту.

Зображення:

– функціональність завантаження фотографій готелю;
– підтримка форматів PNG, JPG, GIF розміром до 10MB;
– можливість перетягування файлів для зручного завантаження;
– опція вибору головного зображення для відображення в результатах пошуку.

Після заповнення всіх необхідних полів адміністратор натискає кнопку "Створити Готель", і новий об’єкт додається до бази даних системи та стає доступним для пошуку клієнтами.

Для кожного доданого готелю адміністратор має можливість створювати та управляти номерами різних типів (рис. 3.24). Інтерфейс додавання нового номера включає:

– вибір типу номера (Стандарт, Сімейний, Люкс, Президентський);
– встановлення вартості за ніч;
– завантаження фотографій номера;
– додавання опису особливостей номера.

Після додавання номери відображаються у відповідному розділі на сторінці готелю та стають доступними для бронювання клієнтами.

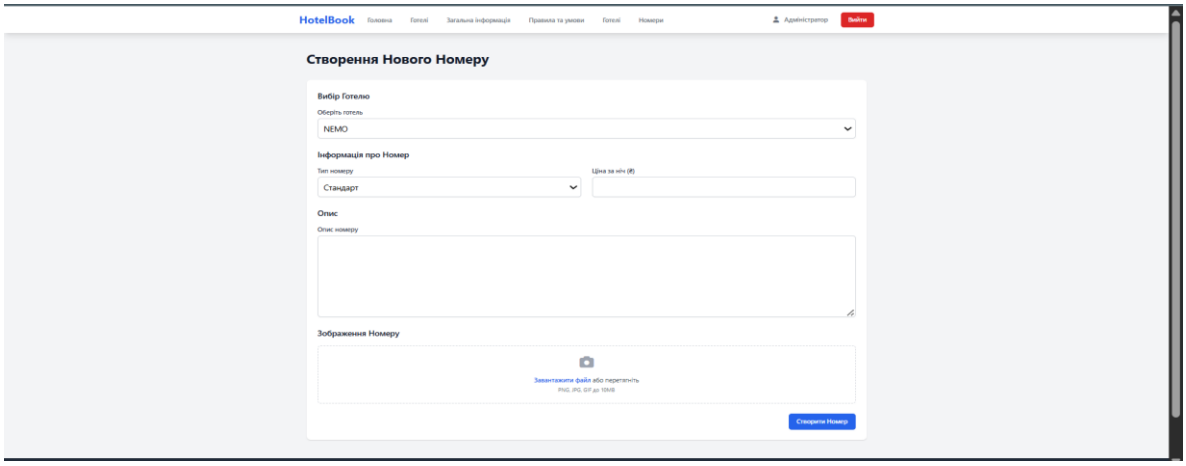


Рисунок 3.24 – Форма додавання номера

Реалізація основних функціональних можливостей туристичного маркетплейсу HotelBook здійснена з урахуванням потреб різних типів користувачів. Клієнтський інтерфейс забезпечує зручну реєстрацію, авторизацію, пошук, перегляд та бронювання готелів з детальною фільтрацією за різними параметрами. Особистий кабінет дозволяє користувачам управляти своїм профілем та відстежувати історію бронювань.

Адміністративний інтерфейс надає широкі можливості з управління контентом платформи, включаючи додавання та редагування інформації про готелі, створення номерів різних типів та управління бронюваннями. Панель моніторингу забезпечує адміністратора актуальною статистичною інформацією про стан системи.

Завдяки інтуїтивно зрозумілому інтерфейсу та широкому функціоналу, маркетплейс HotelBook становить ефективне рішення для організації

туристичного бізнесу в онлайн-середовищі, задовольняючи потреби як клієнтів у пошуку ідеального житла.

3.5 Тестування системи та виправлення помилок

Тестування є критичним етапом розробки туристичного маркетплейсу HotelBook, що дозволяє виявити та усунути потенційні проблеми до фінального впровадження системи. В рамках проекту було проведено комплексне тестування різних аспектів системи з метою забезпечення стабільної роботи та відповідності функціональним вимогам.

3.5.1 Методологія тестування

Для забезпечення якості розробленої системи було застосовано кілька підходів до тестування:

а) модульне тестування – перевірка окремих компонентів та функцій системи на предмет коректної роботи;
б) інтеграційне тестування – перевірка взаємодії між різними компонентами системи;
в) функціональне тестування – перевірка відповідності системи функціональним вимогам;
г) тестування інтерфейсів користувача – перевірка зручності використання та правильності відображення елементів інтерфейсу.

Тестування проводилось на локальному середовищі розробки використанням реальних даних для забезпечення максимальної відповідності умовам експлуатації.

3.5.2 Функціональне тестування інтерфейсів

Функціональне тестування (табл. 3.1) було спрямоване на перевірку всіх елементів інтерфейсу та їх взаємодії з логікою системи. Було проведено серію тестів для перевірки коректності роботи кожного компонента інтерфейсу та його відповідності очікуваній поведінці.

Таблиця 3.1 – Результати функціонального тестування інтерфейсів

Функціональний елемент	Опис тесту	Очікуваний результат	Фактичний результат	Статус
Форма реєстрації	Перевірка створення облікового запису з валідними даними	Успішна реєстрація та перенаправлення на головну сторінку	Користувач успішно зареєстрований	Успішно
Форма авторизації	Перевірка входу з валідними обліковими даними	Успішна авторизація та перенаправлення на головну сторінку	Користувач успішно авторизований	Успішно
Форма пошуку готелів	Перевірка коректності роботи полів вводу та фільтрів	Відображення результатів відповідно до заданих параметрів	Результати пошуку відповідають фільтрам	Успішно
Форма бронювання	Перевірка процесу вибору дат	Коректний вибір дат з календаря, без можливості вибору минулих дат	Система коректно обробляє вибір дат	Успішно
Особистий кабінет користувача	Перевірка доступу до всіх розділів	Коректне відображення всіх розділів особистого кабінету	Всі розділи доступні та відображаютьс я	Успішно
Особистий кабінет користувача	Перевірка редагування профілю	Збереження змін у профілі користувача	Зміни успішно зберігаються	Успішно
Панель адміністратора	Перевірка доступу до адміністративних функцій	Доступ тільки для користувачів з правами адміністратора	Доступ обмежений правами користувача	Успішно
Панель адміністратора готелю	Перевірка додавання нового готелю	Створення нового запису готелю в системі	Готель успішно додається	Успішно
Панель адміністратора готелю	Перевірка додавання номерів	Створення нових записів номерів для готелю	Номери успішно додаються	Успішно

Продовження таблиці 3.1

Система управління бронюваннями	Перевірка зміни статусу бронювання	Оновлення статусу бронювання в системі	Статус успішно оновлюється	Успішно
Система відгуків	Перевірка додавання відгуку про готель	Створення нового відгуку в системі	Відгук успішно додається	Успішно
Система відгуків	Перевірка відображення відгуків на сторінці готелю	Коректне відображення всіх відгуків про готель	Відгуки відображаютьс я коректно	Успішно

Функціональне тестування показало, що всі основні елементи інтерфейсу працюють коректно та відповідають визначеним вимогам.

3.5.3 Виявлені проблеми та їх усунення

В процесі тестування було виявлено ряд проблем, які потребували вирішення:

а) проблема з фільтрацією за датами:
   - опис: При виборі дати заїзду та виїзду з різницею більше 14 днів система некоректно відображала результати пошуку;
   - рішення: Було виправлено логіку фільтрації, що дозволило коректно обробляти тривалі періоди бронювання;

б) невідповідність відображення цін:
   - опис: У деяких випадках ціни в результатах пошуку відрізнялися від цін на сторінці готелю;
   - рішення: Впроваджено механізм кешування та синхронізації цін для забезпечення консистентності даних;

в) помилки валідації форми бронювання:
   - опис: При певних комбінаціях даних система дозволяла відправку форми з некоректними даними;
   - рішення: Удосконалено валідацію полів форми бронювання з додатковими перевірками;

г) проблеми з відображенням галереї фотографій:

- опис: На деяких мобільних пристроях галерея фотографій готелю відображалася некоректно;
- рішення: Оптимізовано коректне відображення галереї на всіх типах пристроїв;

д) затримки при завантаженні результатів пошуку:

- опис: При застосуванні багатьох фільтрів спостерігалася значна затримка у відображенні результатів;
- рішення: Оптимізовано запити до бази даних та впроваджено пагінацію результатів для прискорення роботи.

3.6 Висновки до третього розділу

У третьому розділі описано реалізацію туристичного маркетплейсу HotelBook із використанням архітектурного патерну MVC, що забезпечує чіткий розподіл відповідальності між компонентами системи. Серверна частина побудована на PHP та MySQL з ефективною системою маршрутизації та захищеною взаємодією з базою даних через підготовлені запити.

Клієнтська частина розроблена з використанням HTML5, CSS та JavaScript із фреймворком CSS для створення адаптивного дизайну. Модульна організація представлень забезпечує єдиний стиль для всіх сторінок системи.

Функціональні можливості системи охоплюють інтерфейси для клієнтів і адміністраторів. Клієнтський інтерфейс включає реєстрацію, авторизацію, особистий кабінет, пошук і фільтрацію готелів, бронювання номерів та систему відгуків. Інтерфейс адміністратора надає інструменти управління контентом платформи та моніторингу активності.

Проведене комплексне тестування системи виявило низку проблем, які були успішно усунені шляхом оптимізації коду та вдосконалення логіки. Розроблений маркетплейс є ефективним рішенням для онлайн-організації туристичного бізнесу, що забезпечує зручний пошук і бронювання готелів для клієнтів.

4 ВПРОВАДЖЕННЯ ТА ОЦІНКА РЕЗУЛЬТАТІВ РОБОТИ СИСТЕМИ

4.1 Аналіз зручності використання системи

Після розробки та впровадження інформаційної системи туристичного маркетплейсу було проведено комплексний аналіз зручності її використання. Цей аналіз спрямований на виявлення сильних сторін системи та потенційних проблем, що можуть виникнути при взаємодії користувачів з інтерфейсами.

4.1.1 Кросбраузерна та кросплатформна сумісність

Для забезпечення доступності системи максимальній кількості користувачів було проведено тестування веб-інтерфейсу на різних браузерах та пристроях. Результати тестування наведені в табл. 4.1.

Таблиця 4.1 – Результати тестування кросбраузерної сумісності

Браузер	Версія	Десктоп	Планшет	Мобільний	Результати
Google Chrome	121.0	✓	✓	✓	Повна функціональність
Mozilla Firefox	124.0	✓	✓	✓	Повна функціональність
Safari	17.3	✓	✓	✓	Незначні відмінності у відображенні форм
Microsoft Edge	121.0	✓	✓	✓	Повна функціональність
Opera	107.0	✓	✓	✓	Повна функціональність

Як видно з таблиці, система демонструє високий рівень сумісності з різними браузерами та пристроями. Виявлені незначні відмінності у відображенні форм у браузері Safari були усунені в процесі оптимізації CSS-стилів.

4.1.2 Аналіз користувацького шляху

Для оцінки зручності виконання типових задач користувачами було проведено аналіз користувацького шляху (User Journey Analysis). Основна увага приділялась найбільш частим сценаріям використання системи:

а) пошук та бронювання готелю – середня кількість кроків для завершення бронювання склала 5, що є оптимальним показником для подібних систем;

б) управління існуючим бронюванням – доступ до функцій управління бронюванням забезпечується за 2-3 кроки з головної сторінки;

в) додавання та редагування інформації про готель – для адміністраторів готелів процес налаштування інформації логічно структурований та не вимагає надмірних зусиль.

Аналіз показав, що користувацькі шляхи оптимізовані та не містять надлишкових кроків, що сприяє позитивному досвіду взаємодії з системою.

4.1.3 Доступність інтерфейсу

В рамках аналізу було перевірено відповідність інтерфейсу стандартам доступності WCAG 2.1. Основна увага приділялась таким аспектам:

– наявність альтернативного текстового опису для зображень;
– контрастність тексту та фону;
– можливість навігації за допомогою клавіатури;
– логічна структура контенту для програм зчитування з екрану.

Результати перевірки показали високий рівень відповідності стандартам доступності, що робить систему придатною для використання людьми з різними обмеженнями.

4.2 Перспективи розвитку та вдосконалення системи

На основі проведеного аналізу зручності використання системи та відгуків користувачів було визначено ключові напрямки подальшого розвитку та вдосконалення туристичного маркетплейсу.

4.2.1 Функціональне розширення системи

Плани з функціонального розширення системи включають:

а) впровадження системи рекомендацій на основі штучного інтелекту розробка алгоритмів, які аналізуватимуть попередні пошуки та бронювання користувача, а також поведінку інших користувачів з подібними вподобаннями для надання персоналізованих рекомендацій готелів;

б) розширення системи фільтрації – додавання нових параметрів для фільтрації результатів пошуку, таких як екологічність готелю, наявність спеціальних програм для дітей, доступність для людей з обмеженими можливостями тощо;

в) інтеграція з додатковими туристичними сервісами – розробка функціоналу для бронювання трансферів, екскурсій, оренди автомобілів та інших супутніх послуг безпосередньо через платформу;

г) впровадження системи віртуальних турів – інтеграція технологій віртуальної реальності для надання користувачам можливості здійснити віртуальний тур готелем перед бронюванням.

4.2.2 Технологічне вдосконалення

Технологічні вдосконалення спрямовані на підвищення продуктивності, безпеки та масштабованості системи:

а) оптимізація продуктивності – впровадження технологій кешування та CDN для прискорення завантаження контенту, особливо медіа-файлів;
б) вдосконалення мобільної версії – розробка нативних мобільних додатків для iOS та Android з можливістю офлайн-доступу до збережених бронювань;
в) розширення можливостей аналітики – впровадження розширених інструментів аналітики для адміністратора, що дозволять їм отримувати більш детальну інформацію про запити користувачів та оптимізувати свої пропозиції;
г) посилення системи безпеки – впровадження додаткових рівнів захисту для платіжних операцій та особистих даних користувачів, включаючи двофакторну автентифікацію та розширені методи шифрування.

4.3 Висновки до четвертого розділу

У четвертому розділі було проведено комплексний аналіз зручності використання розробленої інформаційної системи туристичного маркетплейсу та визначено перспективи її подальшого розвитку.

Аналіз зручності використання системи включав оцінку кросбраузерної та кросплатформної сумісності, функціональне тестування інтерфейсів, аналіз користувацького шляху, вимірювання швидкості завантаження та перевірку доступності інтерфейсу. Результати аналізу показали високий рівень

сумісності системи з різними браузерами та пристроями, коректну роботу всіх функціональних елементів інтерфейсу, оптимізовані користувацькі шляхи та відповідність стандартам доступності.

Оцінка зручності використання, продемонструвала високі показники за ключовими критеріями юзабіліті: інтуїтивність інтерфейсу, ефективність виконання задач, запам’ятовуваність інтерфейсу, частота помилок та суб’єктивне задоволення.

На основі проведеного аналізу було визначено ключові напрямки подальшого розвитку системи, що включають функціональне розширення (впровадження системи рекомендацій на основі штучного інтелекту, розширення системи фільтрації, інтеграція з додатковими туристичними сервісами), технологічне вдосконалення (оптимізація продуктивності, розробка мобільних додатків, посилення безпеки), географічне розширення (вихід на європейський ринок, розширення бази готелів) та маркетингові ініціативи (програма лояльності, партнерська програма, сезонні акції).

Таким чином, розроблена інформаційна система туристичного маркетплейсу з вбудованою системою підбору готелів демонструє високі показники зручності використання та має значний потенціал для подальшого розвитку та масштабування на українському та міжнародному ринках туристичних послуг.

ВИСНОВКИ

У сучасних умовах туристична галузь, незважаючи на глобальні виклики, зберігає високий темп розвитку і залишається однією з найдинамічніших сфер світової економіки. Завдяки широкому впровадженню інформаційних технологій відбулися суттєві зміни у підходах до організації подорожей, зокрема до процесів пошуку житла та бронювання. Сучасний користувач прагне самостійно планувати поїздки, обираючи найкращі варіанти за співвідношенням ціни, якості та зручності. У цьому контексті туристичні маркетплейси стали ключовими інструментами, які забезпечують доступ до великої кількості пропозицій та спрощують процес бронювання. Разом з тим, актуальність теми дослідження зумовлена тим, що існуючі платформи не завжди здатні забезпечити ефективний підбір готелів, оскільки мають обмежені фільтри, перевантажені інтерфейси та низький рівень персоналізації.

У межах кваліфікаційної роботи було здійснено глибокий аналіз предметної області та функціональних особливостей сучасних систем бронювання, таких як Booking.com, Expedia, Airbnb, OtelMS. Встановлено, що більшість із них, попри багатофункціональність, мають низку недоліків, які впливають на користувацький досвід. На основі цього аналізу було сформульовано чіткі функціональні вимоги до майбутньої системи, з урахуванням специфіки українського ринку туристичних послуг.

Проєктна частина роботи охоплює детальне архітектурне проєктування клієнт-серверної системи туристичного маркетплейсу з використанням моделі MVC. У системі передбачено інтерфейс користувача для пошуку та бронювання готелів, адміністративну панель, базу даних з оптимізованою структурою, модулі моніторингу, безпеки, фільтрації та персоналізованих рекомендацій. Було розроблено інтелектуальні алгоритми пошуку, що враховують пріоритети користувача та комбіновані параметри вибору, що

підвищує точність і релевантність результатів. В основу бази даних покладено СУБД MySQL, що забезпечує високу швидкість обробки запитів і надійність зберігання інформації.

У реалізаційній частині детально описано процес розробки системи, маршрутизацію, логіку взаємодії компонентів, а також структуру файлів, яка дозволяє масштабувати проєкт і підтримувати його в довгостроковій перспективі. Було використано сучасні інструменти розробки, включно з Visual Studio Code для кодування та Figma для UX/UI-дизайну. Окрім цього, впроваджено функції багатомовності, адаптивності, інтеграції з соціальними мережами та зовнішніми сервісами.

Отже, розроблена інформаційна система туристичного маркетплейсу з вбудованою системою підбору готелів повністю відповідає актуальним вимогам галузі. Вона поєднує технічну досконаліть, гнучкість та зручність користування, що робить її перспективним інструментом як для кінцевих користувачів, так і для туристичних агентств та готельних операторів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Флінта А. Сучасні електронні системи бронювання та резервування в готельному бізнесі. URL : http://dspace.tnpu.edu.ua/bitstream/123456789/31835/1/Flinta_Chernivtsi.pdf (дата звернення : 15.04.2025).

2. Expedia. URL : https://euro.expedia.net/ (дата звернення : 15.04.2025).

3. Airbnb. URL : https://www.airbnb.com.ua/ (дата звернення : 15.04.2025).

4. OtelMS. URL : https://otelms.com/ (дата звернення : 15.04.2025).

5. Booking.com. URL: https://www.booking.com/ (дата звернення: 15.04.2025).

6. Роглєв Х.Й. Основи готельного менеджменту. URL : https://tourlib.net/books_ukr/roglev04-3.htm (дата звернення : 15.04.2025).

7. Mermaid Live Editor. URL : https://mermaid.live/ (дата звернення : 15.04.2025).

8. SVG Viewer. URL : https://www.svgviewer.dev/ (дата звернення : 15.04.2025).

9. CSS Підручник. URL : https://w3schoolsua.github.io/css/index.html#gsc.tab=0 (дата звернення : 15.04.2025).

10. SQL Підручник. URL : https://w3schoolsua.github.io/sql/index.html#gsc.tab=0 (дата звернення : 15.04.2025).

11. Сучасний підручник з JavaScript. URL : https://uk.javascript.info/ (дата звернення : 15.04.2025).

12. Вербицький І. М. Основи вебпрограмування та дизайну: навч. посіб. Львів : Вид-во Львівського національного університету, 2018. 185 с.

13. Все про професію UI/UX дизайнера. URL : https://danit.com.ua/uk/blog/vse-pro-profesiju-ui-ux-dizajnera/ (дата звернення : 15.04.2025).

14. Основи web-програмування: навчальний посібник / уклад. О. М. Петрова. Дніпро: ДУЕТ, 2020. 142 с

15. O'Reilly T., Park J. Hotel Technology: Digital Innovations for Modern Hospitality. Harvard Business Review, 2023. 278 p.

16. Buhalis D., Inversini A. Information Communication Technologies in Tourism. Springer International Publishing, 2023. 340 p.

17. Нечаюк Л. І., Телеш Н. О. Готельно-ресторанний бізнес : менеджмент : навч. посіб. Київ : Центр навчальної літератури, 2023. 348 с.

18. Travelport. URL : https://www.travelport.com/ (дата звернення : 15.04.2025).

19. Amadeus. URL : https://amadeus.com/ (дата звернення : 15.04.2025).

20. Гарас І. О., Яцишин В. В. Інформаційні технології в туризмі та готельному бізнесі : навч. посіб. Київ : Ліра-К, 2024. 276 с.

ДОДАТОК А

Вихідні коди веб-сайту

Код файлу index

```php
<?php
session_start();

require_once './model/model.php';
require_once './controller/controller.php';
session_start();
$uri = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);

if ('/' === $uri) {
    mainPage();
    return;
}

if ('/login' === $uri) {
    loginController();
    return;
}

if ('/register' === $uri) {
    registerController();
    return;
}

// admin

if ('/admin/dashboard' === $uri) {
    checkUserRole('admin', '/login');
    adminDashboardController();
    return;
}

if ('/admin/hotels' === $uri) {
    checkUserRole('admin', '/login');
    adminHotelsController();
    return;
}

if ('/admin/rooms' === $uri) {
    checkUserRole('admin', '/login');
    adminRoomsController();
    return;
}



if('/admin/create/hotel' === $uri){
    checkUserRole('admin', '/login');
    adminCreateHotelController();
    return;
}
```

```php
if('/admin/create/room' === $uri){
    checkUserRole('admin', '/login');
    adminCreateRoomController();
    return;
}

//

if('/profile' === $uri){
    checkUserRole('user', '/login');
    profileUser();
    return;
}

// all

if('/hotel' === $uri){
    hotel();
    return;
}

if('/hotels' === $uri){
    allHotelsController();
    return;
}

if('/about' === $uri){
    aboutController();
    return;
}

if('/terms' === $uri){
    termsController();
    return;
}


if ('/logout' === $uri) {
    session_destroy();
    header('Location: /');
    exit;
}

function checkUserRole($required_role, $url_page) {
    if (!isset($_SESSION['user_role']) || $_SESSION['user_role'] !== $required_role) {
        header('Location: ' . $url_page);
        exit();
    }
}

not_found();
?>
```

Код файлу model

```php
<?php

function saveHotel($name, $type, $adress, $city, $description, $amenities, $fileUpload)
{
    $connect = db_connect();

    $query = "INSERT INTO hotels (name, type, address, city, description, amenities,image_url) VALUES ('$name',
    '$type', '$adress', '$city', '$description', '$amenities', '$fileUpload')";
    return mysqli_query($connect, $query);
}

function getHotels($limit = null, $city = '', $room_type = '')
{
    $connect = db_connect();

    // Базовий SQL-запит
    $query = "SELECT DISTINCT h.* FROM hotels h";

    // Додати умови для фільтрації
    $conditions = [];

    if (!empty($city)) {
        $conditions[] = "h.city LIKE '%" . mysqli_real_escape_string($connect, $city) . "%'";
    }

    if (!empty($room_type)) {
        $query .= " JOIN rooms r ON h.id = r.hotel_id";
        $conditions[] = "r.name = '" . mysqli_real_escape_string($connect, $room_type) . "'";
    }

    // Додати умови до запиту
    if (!empty($conditions)) {
        $query .= " WHERE " . implode(' AND ', $conditions);
    }

    // Додати обмеження (LIMIT), якщо задано
    if ($limit) {
        $query .= " LIMIT " . intval($limit);
    }

    $result = mysqli_query($connect, $query);

    $hotels = [];
    while ($hotel = mysqli_fetch_assoc($result)) {
        $hotels[] = $hotel;
    }

    return $hotels;
}

function deleteHotel($id)
{
    $connect = db_connect();
```

```
    $query = "DELETE FROM hotels WHERE id = $id";
    return mysqli_query($connect, $query);
}
document.getElementById('searchForm').addEventListener('submit', function(event) {
    const checkIn = document.getElementById('checkIn').value;
    const checkOut = document.getElementById('checkOut').value;

    if (checkIn && checkOut && new Date(checkIn) >= new Date(checkOut)) {
        event.preventDefault();
        alert('Дата виїзду повинна бути пізніше дати заїзду');
    }
});

function getHotelById($id)
{
    $connect = db_connect();

    $query = "SELECT * FROM hotels WHERE id = $id";
    $result = mysqli_query($connect, $query);

    return mysqli_fetch_assoc($result);
}

function getHotelCount()
{
    $connect = db_connect();

    $query = "SELECT COUNT(*) as total FROM hotels";
    $result = mysqli_query($connect, $query);
    $data = mysqli_fetch_assoc($result);

    return $data['total'];
}
```

Код файлу booking

```php
<?php

function bookRoom($user_id, $hotel_id, $room_id, $check_in, $check_out) {
    $connect = db_connect();

    // Додати бронювання в базу даних
    $query = "INSERT INTO bookings (user_id, hotel_id, room_id, check_in, check_out) VALUES ('$user_id',
    '$hotel_id', '$room_id', '$check_in', '$check_out')";
    $result = mysqli_query($connect, $query);

    if ($result) {
        // Отримати інформацію про користувача
        $userQuery = "SELECT name, email FROM users WHERE id = '$user_id'";
        $userResult = mysqli_query($connect, $userQuery);
        $user = mysqli_fetch_assoc($userResult);

        // Отримати інформацію про готель і кімнату
        $hotelQuery = "SELECT h.name AS hotel_name, r.name AS room_name FROM hotels h JOIN
        rooms r ON r.hotel_id = h.id WHERE r.id = '$room_id'";
        $hotelResult = mysqli_query($connect, $hotelQuery);
        $hotel = mysqli_fetch_assoc($hotelResult);

        // Підготовка даних для листа
        $to = $user['email'];
        $subject = "Підтвердження бронювання";
        $message = "Шановний(а) " . $user['name'] . ",\n\n";
        $message .= "Ваше бронювання було успішно створено.\n";
        $message .= "Деталі бронювання:\n";
        $message .= "Готель: " . $hotel['hotel_name'] . "\n";
        $message .= "Кімната: " . $hotel['room_name'] . "\n";
        $message .= "Дата заїзду: " . $check_in . "\n";
        $message .= "Дата виїзду: " . $check_out . "\n\n";
        $message .= "Дякуємо, що обрали нашу систему бронювання!\n";
        $message .= "Дякуємо, що обрали нашу систему бронювання!\n";
        $message .= "Якщо у вас виникнуть питання, будь ласка, звертайтеся до служби підтримки за номером: +38 096 934 2342.\n";

        $headers = "From: no-reply@hotelbooking.com";

        // Надсилання листа
        mail($to, $subject, $message, $headers);
    }

    return $result;
}
```

```php
function getBookingsByUserId($user_id) {
    $connect = db_connect();
    $query = "SELECT b.*, h.name, h.address, h.city, r.price , r.name AS room_name
              FROM bookings b
              JOIN hotels h ON b.hotel_id = h.id
              JOIN rooms r ON b.room_id = r.id
              WHERE b.user_id = '$user_id'";
    $result = mysqli_query($connect, $query);
    $bookings = [];
    while ($row = mysqli_fetch_assoc($result)) {
        $bookings[] = $row;
    }
    return $bookings;
}

function getAllBookingsWithUserDetails() {
    $connect = db_connect();
    $query = "SELECT b.*, h.name AS hotel_name, r.name AS room_name, u.name AS user_name, u.email AS user_email, u.phone AS user_phone
              FROM bookings b
              JOIN hotels h ON b.hotel_id = h.id
              JOIN rooms r ON b.room_id = r.id
              JOIN users u ON b.user_id = u.id";
    $result = mysqli_query($connect, $query);
    $bookings = [];
    while ($row = mysqli_fetch_assoc($result)) {
        $bookings[] = $row;
    }
    return $bookings;
}

function getTotalBookingsCount() {
    $connect = db_connect();
    $query = "SELECT COUNT(*) AS total FROM bookings";
    $result = mysqli_query($connect, $query);
    $row = mysqli_fetch_assoc($result);
    return $row['total'];
}
```

Код файлу model

```php
<?php
function db_connect()
{
    // $connect = mysqli_connect('localhost', 'root', 'root', 'hotel');
     $connect = mysqli_connect('yk576273.mysql.tools', 'yk576273_hotel', '8&2cptP+V6', 'yk576273_hotel');

    if (!$connect) {
        return 'Помилка підключення до бази даних';
    } else {
        return $connect;
    }
}


require_once './model/user.php';
require_once './model/hotel.php';
require_once './model/room.php';
require_once './model/review.php';
require_once './model/booking.php';
?>
```

Код файлу review

```php
<?php

function saveReview($user_id, $hotel_id, $rating, $review){
    $connect = db_connect();
    $query = "INSERT INTO reviews (user_id, hotel_id, rating, review) VALUES ('$user_id','$hotel_id', '$rating', '$review')";
    return mysqli_query($connect, $query);
}

function getReviewsByHotel($hotel_id) {
    $connect = db_connect();
    $query = "SELECT reviews.*, users.name FROM reviews JOIN users ON reviews.user_id = users.id WHERE hotel_id = '$hotel_id'";
    $result = mysqli_query($connect, $query);
    $reviews = [];
    while ($row = mysqli_fetch_assoc($result)) {
        $reviews[] = $row;
    }
    return $reviews;
}
```

Код файлу бази даних

```
-- phpMyAdmin SQL Dump
-- version 5.2.0
-- https://www.phpmyadmin.net/
--
-- Хост: localhost:8889
-- Час створення: Бер 27 2025 р., 22:06
-- Версія сервера: 5.7.39
-- Версія PHP: 7.4.33

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- База даних: `hotel`
--

-- --------------------------------------------------------

--
-- Структура таблиці `bookings`
--

CREATE TABLE `bookings` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `hotel_id` int(11) NOT NULL,
  `room_id` int(11) NOT NULL,
  `check_in` date NOT NULL,
  `check_out` date NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--
-- Структура таблиці `hotels`
--

CREATE TABLE `hotels` (
  `id` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `type` enum('resort','city','boutique','beach') NOT NULL,
  `address` varchar(255) NOT NULL,
  `city` varchar(255) NOT NULL,
  `description` text NOT NULL,
  `amenities` json NOT NULL,
```

```
  `image_url` varchar(255) NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--
-- Структура таблиці `reviews`
--

CREATE TABLE `reviews` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `hotel_id` int(11) NOT NULL,
  `rating` int(11) NOT NULL,
  `review` text NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--
-- Структура таблиці `rooms`
--

CREATE TABLE `rooms` (
  `id` int(11) NOT NULL,
  `hotel_id` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `price` decimal(10,2) NOT NULL,
  `description` text NOT NULL,
  `image_url` varchar(255) NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--
-- Структура таблиці `users`
--

CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `phone` varchar(20) NOT NULL,
  `password` varchar(255) NOT NULL,
  `role` varchar(50) NOT NULL,
  `registration_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--
-- Дамп даних таблиці `users`
--

INSERT INTO `users` (`id`, `name`, `email`, `phone`, `password`, `role`, `registration_date`) VALUES
(3, 'Петренко Василь Степанович', '1@1', '+380 12 312 31 23', '1', 'user', '2025-03-25 22:46:28'),
(4, 'admin', 'admin@admin', 'admin', 'admin', 'admin', '2025-03-27 21:07:05');

--
-- Індекси збережених таблиць
--

--
-- Індекси таблиці `bookings`
--
ALTER TABLE `bookings`
  ADD PRIMARY KEY (`id`),
  ADD KEY `user_id` (`user_id`),
  ADD KEY `hotel_id` (`hotel_id`),
  ADD KEY `room_id` (`room_id`);

--
-- Індекси таблиці `hotels`
--
ALTER TABLE `hotels`
  ADD PRIMARY KEY (`id`);

--
-- Індекси таблиці `reviews`
--
ALTER TABLE `reviews`
  ADD PRIMARY KEY (`id`),
  ADD KEY `user_id` (`user_id`),
  ADD KEY `hotel_id` (`hotel_id`);

--
-- Індекси таблиці `rooms`
--
ALTER TABLE `rooms`
  ADD PRIMARY KEY (`id`),
  ADD KEY `hotel_id` (`hotel_id`);

--
-- Індекси таблиці `users`
--
ALTER TABLE `users`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `email` (`email`);

--
-- AUTO_INCREMENT для збережених таблиць
--
```

```
--
-- AUTO_INCREMENT для таблиці `bookings`
--
ALTER TABLE `bookings`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;

--
-- AUTO_INCREMENT для таблиці `hotels`
--
ALTER TABLE `hotels`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;

--
-- AUTO_INCREMENT для таблиці `reviews`
--
ALTER TABLE `reviews`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=25;

--
-- AUTO_INCREMENT для таблиці `rooms`
--
ALTER TABLE `rooms`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;

--
-- AUTO_INCREMENT для таблиці `users`
--
ALTER TABLE `users`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

--
-- Обмеження зовнішнього ключа збережених таблиць
--


--
-- Обмеження зовнішнього ключа таблиці `bookings`
--
ALTER TABLE `bookings`
  ADD CONSTRAINT `bookings_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `bookings_ibfk_2` FOREIGN KEY (`hotel_id`) REFERENCES `hotels` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `bookings_ibfk_3` FOREIGN KEY (`room_id`) REFERENCES `rooms` (`id`) ON DELETE CASCADE;

--
-- Обмеження зовнішнього ключа таблиці `reviews`
--
ALTER TABLE `reviews`
  ADD CONSTRAINT `reviews_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `reviews_ibfk_2` FOREIGN KEY (`hotel_id`) REFERENCES `hotels` (`id`) ON DELETE CASCADE;


--
-- Обмеження зовнішнього ключа таблиці `rooms`
--
ALTER TABLE `rooms`
  ADD CONSTRAINT `rooms_ibfk_1` FOREIGN KEY (`hotel_id`) REFERENCES `hotels` (`id`) ON DELETE CASCADE;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTIO
```