

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ВЕБЗАСТОСУНОК “БРОНІЮВАННЯ ГОТЕЛЬНИХ
НОМЕРІВ”

WEB APPLICATION “RESERVATION OF HOTEL ROOMS”

Виконав(ла): студент(ка) 4 курсу, групи КНТ-151

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ІШКОЛА Д.В.

(ПРИЗВИЩЕ та ініціали)

Керівник ФЕДОРОНЧАК Т.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОРОТКИЙ О.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Факультет КНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ШКОЛИ Дениса Володимировича
 (ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Вебзастосунок “Бронювання готельних номерів”.
Web Application “Reservation of Hotel Rooms”

керівник проєкту (роботи) к.т.н., доцент, ФЕДОРОНЧАК Тетяна Василівна,
 (науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 року № 209

2. Строк подання студентом проєкту (роботи) 02 червня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне
завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження.
2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ФЕДОРОНЧАК Т.В., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання « 14 » квітня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

_____ Денис ШКОЛА
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

_____ Тетяна ФЕДОРОНЧАК
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 94 с., 9 табл., 38 рис., 3 дод., 22 джерела.

БРОНЮВАННЯ ЖИТЛА, ВЕБЗАСТОСУНОК, PYTHON, SQLITE, DJANGO.

Об'єктом дослідження цієї роботи є процес розробки програмного забезпечення для бронювання готельних номерів.

Предмет дослідження – програмні засоби для організації допомоги процесу автоматизації бронювання готельних номерів.

Мета роботи – проектування й розробка вебзастосунку, що дозволяє здійснити процедуру онлайн-бронювання готельних номерів.

Матеріали, методи та технічні засоби: мова програмування Python, фреймворк Django, система управління базами даних SQLite.

Результати. Спроектовано та розроблено програмне забезпечення для виконання процесу онлайн-бронювання житла в готелях.

Висновки. Розроблено програмне забезпечення для бронювання житла в закладах тимчасового проживання, яке завдяки автоматизації підтримує діяльність готелей. Виконано функціональне, usability та безпекове тестування розробленого вебзастосунку для бронювання готельних номерів.

Галузь використання – засоби бронювання готельних номерів.

ABSTRACT

Explanatory note to the bachelor's diploma qualification work: 94 p., 9 table, 38 fig., 3 appendix, 22 sources.

ACCOMMODATION RESERVATION, WEB APPLICATION,
PYTHON, SQLITE, DJANGO.

The object of research of this work is the process of developing software for booking hotel rooms.

The subject of research is software for organizing assistance in the process of automating hotel room booking.

The purpose of the work is research and development of a web application that allows you to carry out the procedure of online hotel room booking.

Materials, methods and technical means: Python programming language, Django framework, SQLite database management system.

Results. Software has been designed and developed to perform the process of online hotel accommodation booking.

Conclusions. Software has been developed for booking accommodation in temporary accommodation establishments, which, thanks to automation, supports the activities of hotels. The developed web application for booking hotel rooms has been tested. Field of use – hotel room booking tools.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Поняття готелю. Програмне забезпечення для бронювання готельних номерів	10
1.2 Огляд вебзастосунків для бронювання готельних номерів	14
1.3 Порівняння вебзастосунків для бронювання готельних номерів	19
1.4 Висновки за розділом 1	21
2 Матеріали і методи.....	22
2.1 Вибір мови програмування	22
2.2 Вибір середовища розробки.....	24
2.3 Вибір фреймворку	28
2.4 Вибір бази даних	31
2.5 Висновок за розділом 2	34
3 Опис програми.....	35
3.1 Архітектура вебзастосунку для бронювання готельних номерів	35
3.2 Опис розробленої бази даних вебзастосунку для бронювання готельних номерів	44
3.3 Опис файлів, що реалізують логіку вебзастосунку для бронювання готельних номерів	48
3.4 Опис файлів, що реалізують інтерфейс користувача та конфігурацію вебзастосунку для бронювання готельних номерів	51
3.5 Висновки за розділом 3	53
4 Експлуатація, тестування та експериментальне дослідження програми	54

4.1 Призначення та умови застосування вебзастосунку	54
4.2 Опис процесу роботи з розробленим вебзастосунком для бронювання готельних номерів	54
4.3 Тестування роботи вебзастосунку для бронювання готельних номерів	62
4.4 Висновки за розділом 4	65
Висновки	66
Перелік джерел посилання	68
Додаток А Технічне завдання	70
Додаток Б Код програми	74
Додаток В Слайди презентації	86

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ПЗ – програмне забезпечення;

СУБД – система управління базами даних;

IDE – Integrated Development Environment,

UX – Uxer Experience Design;

API – Application Programming Interface;

HTML – HyperText Markup Language;

ORM – Object-Relational Mapping;

CSS – Cascading Style Sheets.

ВСТУП

Ритм життя людини в сучасному світі стає стрімкішим через зростання впливу інформаційного поля на людей, що підвищує загальну обізнаність населення та сприяє посиленню конкуренції в кожній сфері життєдіяльності окремо. Підвищення конкуренції посилює потребу людства в цифрових інструментах, що допомагають економити час, зокрема, час для планування відряджень та подорожей, в чому допоможуть вебзастосунки для бронювання готельних номерів.

Програмні засоби для онлайн-бронювання готельних номерів повинні відповідати людським вимогам до інструментів економії часу в швидкодії, надійності та зручності, тому можуть містити в собі функції перегляду всіх пропозицій для бронювання, пошуку необхідних готелів за допомогою фільтрів, оформлення та підтвердження бронювання обраного житла в будь-який час. Не менш важливим фактором необхідності існування вебзастосунків для бронювання готельних номерів є підтримка діяльності готелів завдяки наданню можливості цілодобового резервування житла, що значно підвищує дохід таких установ [1] - [5].

Одна з найголовніших переваг вебзастосунків для бронювання готельних номерів – наявність всіляких форм для зворотнього зв'язку, зокрема відгуків та служби підтримки. Завдяки відгукам та оцінкам, потенційні користувачі вебзастосунку зможуть обирати найкращі пропозиції для резервування та будуть повністю впевнені в тому, що бронювання було підтверджене завдяки присутності системи push-up сповіщень, а форма для спілкування з адміністрацією дозволить вирішити будь-яке питання стосовно бронювання [1] - [5].

Саме тому розробка вебзастосунку для бронювання готельних номерів є актуальною – інструменти для економії часу на планування мандрівок та відряджень стають необхідною частиною життя кожної людини, адже через збільшення кількості задач у людства, вільного часу стає все менше і менше.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття готелю. Програмне забезпечення для бронювання готельних номерів

Готелі - це будівлі, що складаються з облаштованих для життя людей кімнат, які в свою чергу називаються номерами. Такі установи складають основу третинного сектору світової економіки, надаючи послуги тимчасового проживання. Готелі прямо впливають на сферу туризму, підтримують розвиток культурного збагачення людства завдяки прийманню на своїх територіях людей з різних країн, забезпечують робочими місцями населення [1] - [5].

Для подальшого розуміння призначення і функціоналу програмних засобів для бронювання готельних номерів, необхідно визначити і розібрати, які види готелів існують, чим вони відрізняються та які особливості резервування мають.

Готелі з повним спектром послуг – установи, що пропонують гостям комплексний сервіс для проживання, тобто надають не тільки власний номер з усіма необхідними зручностями, а й інші можливості для проведення часу на території закладу. До таких можливостей можна віднести наявність вишуканих ресторанів високої кухні з різноманітних куточків планети, фітнес-центрів для занять спортом, а також приватних паркінгів для транспортних засобів і спеціальних приміщень для проведення заходів різного ступеня призначення. Такі установи ставлять своєю метою надати клієнтам універсальне рішення їх проблем на території свого закладу, гарантуючи приємне перебування та надання послуг в цілодобовому режимі. Готелі з повним спектром послуг бронюються не за тим самим принципом, як інші заклади тимчасового проживання : часто на пропозиціях з бронюванням номерів в таких готелях варіюється ціна, адже такі установи – одні з найпопулярніших варіантів для влаштування зустрічей політиків, дипломатів та інших світових діячів. Відповідно, під час розробки

вебзастосунку необхідно буде врахувати про готелі з повним спектром послуг таку особливість : бронювання номеру в таких установах має відбуватися тільки після подвійного підтвердження від адміністратора вебзастосунку [1] - [5].

Бутік-готелі – установи, що характеризуються компактним розміром будівлі загалом та номерами окремо, адже перш за все вони є пам'ятками культури. Такі заклади не схожі на готелі масового призначення через значно меншу кількість місць для проживання. Сервіс в таких установах набагато вище рівнем, ніж в інших готелях, бо до клієнта ставляться як до персонального гостя, що має відчувати всю унікальну атмосферу закладу та особливості місцевої культури. Бутік-готель, завдяки своєму історичному зовнішньому вигляду та дотриманню місцевих традицій, намагається повністю занурити клієнта в інший світ, тому умови і правила бронювання через вебзастосунок для такого закладу відрізняються від інших. Зокрема, такі установи часто пропонують спеціальні пропозиції, які стосуються конкретних номерів та подій, наприклад – варіанти проживання для закоханих пар під час свята Святого Валентина. Тому при розробці вебзастосунку для бронювання готельних номерів варто буде врахувати, що резервування кімнат у бутік-готелі має йти у вигляді спеціальної “пропозиції – пакету”, що містить назву конкретної події та номеру [1] - [5].

Бюджетні готелі – установи, що характеризуються наданням економічно вигідних та ефективних варіантів для тимчасового проживання. Ці готелі надають лише основний сервіс для життя клієнтів без надмірностей та додаткових можливостей для проведення часу, що наявні в інших типах закладів для тимчасового розміщення. Мешканці таких установ надають перевагу економії на місці тимчасового перебування, відповідно, єдині особливості, що варто врахувати під час розробки вебзастосунку для бронювання готельних номерів у бюджетних готеля – відносно висока доступність кімнат, низька ціна та майже повна відсутність сервісів [1] - [5].

Курортні готелі – установи, що характеризуються незвичним розташуванням своїх будівель, спрямованих на створення максимально приємних умов для відпочинку. Такі готелі є не просто місцями для сну - це місця для розслаблення, яке досягається розташуванням у відповідних куточках планети, наявності басейнів, пляжів на території установи, спациентрів. Розташування готелів пов'язане із сезонністю та прямо впливає на динамічне ціноутворення, наприклад, в теплих країнах вартість на номери в курортних будівлях буде значно вища влітку, ніж в інші пори року, тому цю особливість варто буде врахувати під час розроблення вебзастосунку для бронювання готельних номерів [1] - [5].

Готелі тривалого проживання – установи, що надають клієнтам всі необхідні умови для проживання на тривалий термін. Ці готелі намагаються створити максимальне відчуття рідної домівки кожного клієнта, тому пропонують свободу у багатьох побутових справах : мешканці можуть власноруч готувати їжу, прати білизну, влаштовувати свій індивідуальний простір на весь термін свого довготривалого перебування. Під час розробки вебзастосунку для бронювання готельних номерів варто буде врахувати таку особливість : готелі довготривалого проживання неможливо зарезервувати на короткий термін, мінімальний з можливих – один тиждень [1] - [5].

Функціонал ПЗ для бронювання готельних номерів можна розподілити на декілька різних видів: користувацький, для власників готелів, адміністративний.

Користувацький функціонал ПЗ для бронювання готельних номерів починається з перш за все із можливості перегляду всіх існуючих пропозицій для резервування : абсолютно кожен користувач може передивлятися сторінки варіантів для тимчасового проживання без задавання додаткових фільтрів. Це необхідно для загального ознайомлення потенційного клієнта з ПЗ, для проведення ним оцінки існуючого контенту вебзастосунку. Також користувачі мають всі умови для використання пошукових фільтрів, характер яких може відрізнятися за багатьма аспектами : наприклад, за типом готелю,

ціною, бажаною датою заїзду/виїзду, необхідною кількістю спальних місць в номері, наявністю повного або часткового сервісу обслуговування і так далі. Список фільтрів може змінюватися залежно від кожного програмного засобу, але чим більша кількість критеріїв для вибору – тим більший шанс того, що користувач стане потенційним клієнтом вебзастосунку та обраного готелю. Не менш важлива частина користувацького функціоналу ПЗ для бронювання готельних номерів – перегляд кожної пропозиції готелю окремо : людина має можливість ознайомитись із описом закладу тимчасового проживання, переглянути фотографії, прочитати відгуки та побачити оцінки. Обравши потрібну пропозицію для резервування, користувач може виконати процес бронювання готельного номеру, надавши всі необхідні особисті дані, та отримати відповідь у вигляді push-up сповіщення. За необхідності, деталі бронювання можна обговорити та узгодити з адміністрацією готелю або обраного вебзастосунку для резервування житла в закладах тимчасового проживання. Ще однією важливою функцією в таких ПЗ є наявність користувацького профілю, де наявна можливість змінювати особисті та платіжні дані, а також додавати бажані в майбутньому пропозиції для бронювання в список бажань. Також додатковою функцією для користувача є можливість зміни мови ПЗ на іншу задля підвищення комфорту користування [1] - [5].

Функціонал ПЗ для бронювання готельних номерів, що стосується власників готелів, перш за все надає можливість роботи з пропозиціями що доступні для резервування : зміна фотографій, опису, цін, налаштування типів номерів. Власники закладів тимчасового проживання мають повний контроль над пропозиціями своїх готелів, можуть не тільки змінювати деталі, а й взагалі повністю видаляти існуючі чи додавати нові варіанти для проживання. Не менш важливий аспект, що є необхідним для власників таких установ, - доступ до статистики : керівництво готелів може переглядати, в який саме час здійснюється найбільша кількість бронювань,

середню завантаженість номерів, що дозволяє змінювати стратегію розвитку своїх компаній [1] - [5].

Адміністративний функціонал ПЗ для бронювання номерів в основному має одне призначення – модерація всього, що відбувається на вебзастосунку : управління користувачами та власниками готелів, пропозиціями для бронювання та відгуками. Адміністратори таких вебзастосунків мають можливість блокувати користувачів та власників готелів за порушення правил платформи, перевіряти додані пропозиції бронювання готельних номерів на відповідність політиці вебзастосунку, видаляти нецензурні та невідповідні правилам платформи відгуки [1] - [5].

Отже, було визначено, що готелі - установи, що надають послуги тимчасового проживання на своїх територіях. Проаналізовано різні типи готелів, їх характеристики та особливості, що необхідно врахувати під час розробки ПЗ для бронювання готельних номерів. Відзначено, що особливості кожного типу готелю майже ніяк не впливають на основне призначення програмних засобів для бронювання готельних номерів, яке спрямоване на економію часу користувачів та організацію процесу онлайн-бронювання житла в закладах тимчасового проживання.

1.2 Огляд вебзастосунків для бронювання готельних номерів

Розглянемо та проаналізуємо вебзастосунки для бронювання готельних номерів.

Вебзастосунок Booking.com (рис 1.1) є одним з найпотужніших рішень для бронювання житла в закладах тимчасового проживання в режимі онлайн, адже містить близько 29 мільйонів пропозицій що доступні для резервування на 148 тисячах напрямків по всьому світу [6]. Цей вебзастосунок забезпечує активну взаємодію між користувачами та постачальниками послуг, при цьому зменшуючи ризики та час на оформлення бронювань, адже має чат підтримки і велику кількість

додаткових можливостей – оформлення білетів на літак, оренда автомобілів і замовлення таксі до аеропорту потрібного користувачеві міста. Важливо зазначити, що під час розробки вебзастосунку в цій дипломній роботі бакалавра, функціонально має бути реалізовано тільки роботу з бронюваннями готельних номерів.

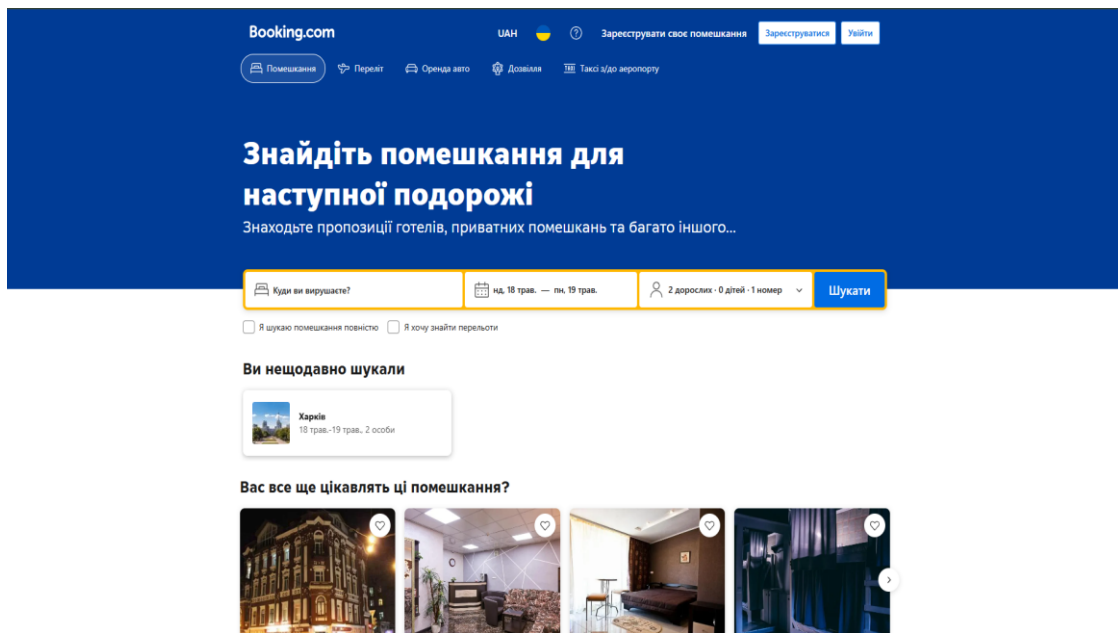


Рисунок 1.1 – Інтерфейс головної сторінки вебзастосунку Booking.com

Вебзастосунок Booking.com надає великий функціонал для всіх категорій користувачів - потенційних клієнтів, власників готелів та адміністраторів. Потенційні клієнти можуть переглядати всі існуючі пропозиції для бронювання без встановлення додаткових критеріїв пошуку, шукати необхідні варіанти завдяки великій кількості фільтрів різного типу та призначення, сортувати знайдені пропозиції за ціною та відгуками, переглядати сторінки готелів. Користувачі можуть здійснювати процес бронювання обраного готельного номеру, залишати відгуки для замовлених раніше пропозицій, змінювати дані в особистому профілі та в ньому ж відмінити заброньовані пропозиції, а також переглядати минулі бронювання завдяки історії замовлень. Для кожного потенційного клієнта цілодобово працює служба підтримки, що значно підвищує комфортність використання

вебзастосунку. Власники готелів можуть розміщувати свої пропозиції для бронювання, змінювати їх та видаляти, а адміністрація завдяки адмін-панелі може модерувати відгуки, блокувати користувачів згідно правил політики вебзастосунку.

Основні переваги використання Booking.com для бронювання готельних номерів :

- широкий вибір готелів: наявні пропозиції для резервування різних типів закладів тимчасового проживання в усіх країнах світу;
- зручний інтерфейс: пошуковий рядок доступний одразу на головній сторінці вебзастосунку, а фільтри розташовуються на лівій бічній панелі, що робить їх легкодоступними, економить час та відповідає ергономічним законам UX-дизайну [7];
- надійність відгуків: коментарі на пропозиціях, доступних для бронювання, можуть залишати лише ті користувачі, які дійсно проживали в цих готельних номерах;
- багатомовність та мультивалютність : вебзастосунок Booking.com підтримує 40 різних мов та десятки світових валют, що значно спрощує процес оформлення та підтвердження бронювання;
- цілодобовий зв'язок з адміністрацією : служба підтримки Booking.com працює цілодобово та має операторів, що володіють кожною з 40 мов, представлених на вебзастосунку.

Стосовно недоліків вебзастосунку для бронювання готельних номерів Booking.com можна виділити наявність прихованих зборів : кінцева вартість деяких пропозицій готелів для резервування іноді може зростати через туристичні збори, додаткові податки та плату за обслуговування, що вводить потенційного клієнта в оману та може призвести до відтоку користувачів з вебзастосунку.

Ще одним вебзастосунком для бронювання готельних номерів є Hotels24.ua (рис 1.2) – інструмент, що направлений на пошук кімнат в закладах тимчасового проживання виключно на території України та містить

інформацію про 4368 готелів [8]. Цей вебзастосунок не надає додаткових послуг та можливостей, на відміну від Booking.com, він спрямований конкретно на резервування готельних номерів.

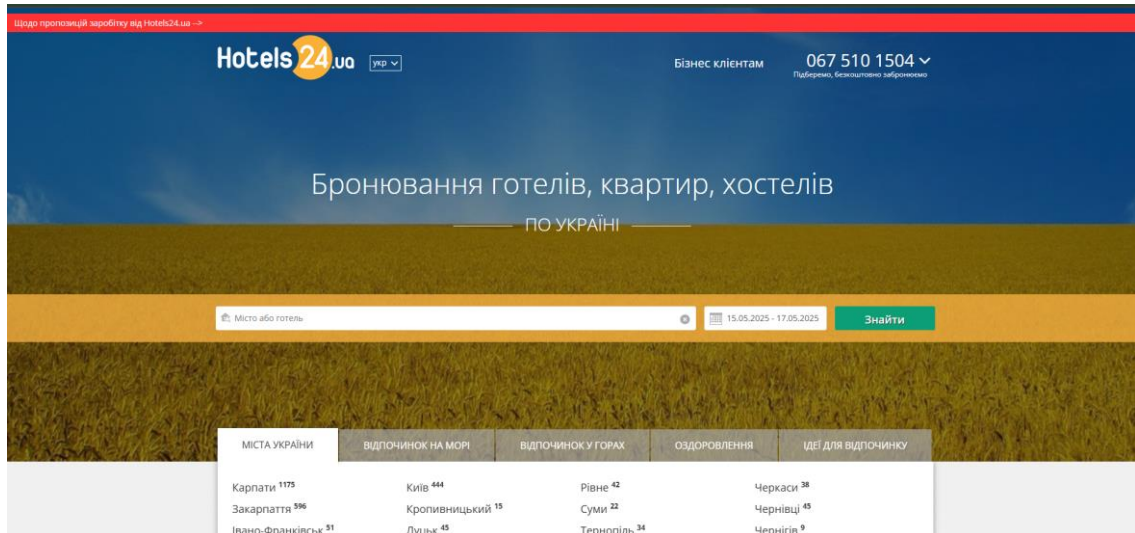


Рисунок 1.2 – Інтерфейс головної сторінки вебзастосунку Hotels24.ua

Функціонал вебзастосунку для бронювання готелів Hotels24.ua не є простим для розуміння, адже не містить панелі додавання готелів для їх господарів: додати пропозицію для бронювання готельного номеру все ще можливо, але для цього треба зв'язатися з адміністрацією вебзастосунку за допомогою мобільного телефону, що значно знижує швидкість обробки резервувань. Через відсутність окремої панелі керування для господарів готелів, додаванням нових варіантів тимчасового житла займається адміністрація вебзастосунку, що з одного боку займає більше часу для оформлення бронювань, а з іншого – гарантує більшу безпеку та надійність успішного резервування житла в установі тимчасового проживання, адже немає необхідності ще раз фільтрувати та змінювати існуючі варіанти для бронювання готельних номерів. Адміністрація вебзастосунку одразу правильно сформулює оголошення про пропозицію для тимчасового житла, дотримуючись всіх правил політики платформи, що надає користувачам більшої впевненості в успішності майбутнього візиту в готель.

Стосовно функцій для користувачів – вони не мають можливість переглядати всі існуючі готелі без додавання фільтрів, але все ще можуть шукати необхідні пропозиції завдяки великій кількості критеріїв для пошуку, сортувати за типом установи, ціною та популярністю, а також переглядати обрані варіанти та здійснювати процес бронювання. Єдиний суттєвий недолік у користувацьких функціях цього вебзастосунку, який можна відмітити вже зараз – необхідність підтвердження бронювання готельного номеру за допомогою мобільного телефону, що підвищує кількість необхідних кроків для перетворення користувача на клієнта.

Основні переваги використання Hotels24.ua для бронювання готельних номерів :

- безкоштовне бронювання: вебзастосунок не стягує ніяких комісій та прихованих податків і не потребує передоплати за резервування готельного номеру, що значно підвищує довіру в користувачів та розширює коло потенційних клієнтів;

- акції та знижки: адміністрація сервісу постійно вводить акційні пропозиції та надає знижки на перше бронювання на вебзастосунку, що дозволяє заощаджувати кошти користувачів.

Недоліки використання Hotels24.ua для бронювання готельних номерів :

- відсутність push-up сповіщень: про успішність підтвердження бронювання готельного номеру та всі можливі зміни в уже існуючому резервуванні, користувач дізнається за номером телефону, тобто цей вебзастосунок – лише інструмент для планування та оформлення майбутнього приїзду в заклад тимчасового проживання, для гарантування успішності бронювання необхідні інші маніпуляції. Це значно ускладнює виконання найголовнішого функціоналу таких вебзастосунків та сприяє відтоку користувачів;

- обмежена кількість можливих напрямків для резервування : вебзастосунок Hotels24.ua зосереджений перш за все на український ринок, що обмежує його використання для планування міжнародних подорожей;
- відсутність можливості цілодобового бронювання та чату з адміністрацією : оформити і підтвердити заявку на резервування готельного номеру та поставити питання адміністрації вебзастосунку можливо тільки в період з 8.00 до 24.00, що значно зменшує коло потенційних клієнтів.

1.3 Порівняння вебзастосунків для бронювання готельних номерів

При порівнянні вебзастосунку для бронювання готельних номерів, що розробляється в цій роботі, та вже існуючих аналогів для резервування житла в закладах тимчасового проживання, варто врахувати деякі особливості : основна відмінність між ними – в процесі додавання пропозицій резервування готельних номерів. У вебзастосунку, що створюється в даній роботі, варіанти житла в закладах тимчасового проживання додаються тільки адміністратором завдяки адмін-панелі. Тобто, немає необхідності в телефонних дзвінках та створенні спеціальної панелі керування вебзастосунком для власників готелей, на відміну від Booking.com та Hotels24.ua, що значно підвищує безпеку бронювання та впевненість потенційного клієнта в успішному резервуванні. Ще одна важлива особливість, що необхідна для порівняння вебзастосунків для бронювання готельних номерів – робота push-up сповіщень : якщо в Hotels24.ua вони взагалі відсутні, то в Booking.com та розробці, що створюється в даній роботі, вони несуть практичне значення та містять всю необхідну інформацію про успішність майбутніх бронювань. Зведені результати порівняння вебзастосунків для бронювання готельних номерів з урахуванням всіх необхідних особливостей наведено в таблиці 1.1.

Таблиця 1.1 – Порівняння вебзастосунків для бронювання готельних номерів

Критерій порівняння	Booking.com	Hotels24.ua	Розробка
Додавання пропозицій бронювання готельних номерів	+-	+-	+
Інформативність push-up сповіщень	+	-	+
Цілодобове оформлення бронювань	+	-	+
Вартість послуг вебзастосунку	Стягується комісія	Абсолютно безкоштовно	Абсолютно безкоштовно
Кількість фільтрів для пошуку	+	+	+
Різноманітність місць доступних для бронювання	+	-	-
Різноманітність мов та валют	+	-	-

Підводячи підсумки до таблиці 1.1, можна сказати, що під час створення вебзастосунку для бронювання готельних номерів в даній роботі варто реалізувати такі функціональні вимоги:

– реєстрація та авторизація користувачів, наявність персонального кабінету клієнта з можливістю зміни персональних даних та переглядом історії бронювань;

– налагодження роботи адмін-панелі для додавання, видалення та редагування сторінок бронювань, прийняття заявок на резервування готельного номеру, модерації відгуків та користувачів вебзастосунку;

- сортування та пошук пропозицій для бронювання за допомогою фільтрів, а також перегляд всіх можливих варіантів без додавання критеріїв пошуку;
- здійснення та підтвердження процесу бронювання користувачем, надання можливості залишити відгук та оцінку пропозиціям, що раніше резервував кожен користувач.

1.4 Висновки за розділом 1

Було проведено аналіз предметної області, перевірено роботу існуючих рішень проблеми онлайн-бронювання готельних номерів, зокрема Booking.com та Hotels24.ua, наведено їх переваги та недоліки, а основні результати зведено у вигляді таблиці.

Завдяки проведенню порівняльного аналізу існуючих аналогів вебзастосунків для бронювання готельних номерів сформульовано необхідні функціональні вимоги до розробки вебзастосунку в цій роботі. Також було сформульовано технічне завдання, яке враховує необхідність підтримки у вебзастосунку різних умов бронювання для кожного з типів готелів, а також наявність спеціальної форми для зв'язку з адміністрацією вебзастосунку.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

Перед початком аналізу основних мов програмування, що можна використати для розробки вебзастосунку для бронювання готельних номерів, варто визначитися з найважливішими параметрами, яким має відповідати обрана мова :

- синтаксис: процес написання коду має бути максимально простим, а читабельність рядків не повинна викликати труднощів, адже кількість спроектованих раніше функціональних вимог до вебзастосунку для бронювання готельних номерів є об'ємною. Важливо не зосереджувати всю увагу під час розробки ПЗ на синтаксичних особливостях мови програмування, бо створенням вебзастосунку займається одна людина, а не ціла команда;

- мінімальні вимоги в споживанні ресурсів серверної частини : обрана мова програмування має забезпечити найменше навантаження на апаратне забезпечення серверу, зокрема на оперативну пам'ять та процесор, при обробці великої кількості запитів. Це допоможе організувати швидку, цілодобову та безперебійну роботу вебзастосунку;

- підтримка фреймворків для створення вебзастосунків : наявність фреймворків значно зменшує час на розробку ПЗ, адже вони допомагають вирішувати стандартні задачі, наприклад, налагоджувати роботу з БД та створення API. Без фреймворку на реалізацію цих модулів піде певний час, тому краще взяти вже готові та прості рішення, що пропонують фреймворки, і налаштувати їх під свій вебзастосунок.

Після початкового перегляду визначених раніше параметрів для вибору мови програмування, було відібрано три потенційних кандидати – PHP, Python, JavaScript.

PHP – мова програмування сценаріїв, що була створена виключно для розробки вебзастосунків. PHP підтримує три різних парадигми

програмування – функціональну, процедурну та об’єктно-орієнтовану, що надає цій мові характеристику прототипної гнучкості. PHP-код виконується виключно на сервері, а в результаті генерації сервер надсилає браузеру HTML-документ, що дозволяє створювати динамічні сайти. Синтаксис PHP є доволі простим для розуміння, але має деякі власні особливості : оголошення змінних починаються з символу “ \$ “, функції оголошуються через ключове слово “function“, а весь код розміщується в HTML-тегах “<?php ... ?>“, що дозволяє динамічно генерувати дані в HTML-документах. Вимоги до серверних ресурсів у цієї мови є одними з найнижчих, адже PHP-скрипт виконується і одразу завершує роботу, що мінімізує тривалість утримання ресурсів. Також PHP підтримує багато фреймворків для створення вебзастосунків, що значно зменшить необхідний час для розробки [9].

Python – інтерпретована мова з динамічною типізацією, що підтримує імперативну, об’єктно-орієнтовану, функціональну, декларативну та ітеративну мови програмування. Python є мовою загального призначення, тому широко застосовується в сферах машинного навчання, ігрової розробки та інших. Синтаксис Python є найпростішим в порівнянні з усіма можливими мовами програмування, адже змінні оголошуються без зазначення типу, блоки коду визначаються за допомогою відступів, а кінці рядків не позначаються символом “ ; “ . Вимоги в споживанні серверних ресурсів Python є трохи більшими, ніж в PHP, адже код запускається на виконання в віртуальному середовищі CPython, що потребує додаткових але незначних витрат оперативної пам’яті [10].

JavaScript - інтерпретована мова з динамічною типізацією, що підтримує імперативну, об’єктно-орієнтовану, функціональну та асинхронну парадигми програмування. Ця мова в основному застосовується для розробки вебзастосунків та мобільних додатків. Синтаксис JavaScript важчий, ніж в розглянутих раніше мовах, але все одно є доволі простим для розуміння, адже в цій мові використовуються фігурні дужки для позначення блоків коду, підтримуються call-back функції. Вимоги в споживанні серверних

ресурсів JavaScript є середніми через асинхронну модель Node.js, яка обробляє тисячі запитів в одному потоці, а не в декількох [11].

Зведені результати порівняння мов програмування, що доступні для розробки вебзастосунку для бронювання готельних номерів, наведені в таблиці 2.1.

Таблиця 2.1 – Вибір мови розробки вебзастосунку для бронювання готельних номерів

Критерій для порівняння	PHP	Python	JavaScript
Синтаксис	+-	+	-
Вимоги в споживанні ресурсів серверної частини	+	-	+-
Підтримка фреймворків для створення вебзастосунків	+-	+	+

Під час аналізу результатів з таблиці 2.1, було зроблено висновок, що необхідна мова для розробки вебзастосунку для бронювання готельних номерів – Python, адже вона має найпростіший синтаксис та підтримує велику кількість фреймворків для створення вебзастосунків. Це значно скоротить час, необхідний на розробку такого ПЗ, що дозволить швидше приступити до процесу тестування вебзастосунку. Для усунення недоліку високого споживання ресурсів серверної частини вебзастосунку необхідно буде оптимізувати код ПЗ, наприклад, завдяки мінімізації звернень до БД та використанні кешування результатів запитів.

2.2 Вибір середовища розробки

Обираючи середовище розробки для створення вебзастосунку для бронювання готельних номерів, необхідно перш за все визначитися з основними критеріями відбору, яким має відповідати обране середовище.

Виходячи з пункту 2.1, можна зробити висновок, що основний критерій для IDE – підтримка мови програмування Python.

Також варто зазначити, що якщо обрані варіанти середовищ розробки мають платну та безкоштовну версію, то в рамках цієї роботи буде обрано безкоштовну версію.

Важливі критерії для вибору середовища розробки :

- підтримка фреймворків для створення вебзастосунків : обране IDE має надавати зручні інструменти для роботи з фреймворками, що допоможуть в створенні вебзастосунку та підтримуються мовою програмування Python, зокрема Django, Tornado та Flask;

- підтримка тестування : IDE має надати необхідний функціонал для автоматичного шаблонного генерування тестів, а також інструменти для їх запуску та візуалізації результатів. Підтримка цієї функції допоможе налагодити коректну роботу вебзастосунку, зокрема, перевірити процеси валідації даних користувача та правильного обчислення ціни бронювання готельного номеру в залежності від кількості обраних ночей;

- робота з базами даних: обране середовище розробки повинно мати інструменти для роботи з різними базами даних : підключення до БД, перегляд та редагування таблиць, запуск та доповнення SQL-запитів а також експорт баз даних.

Після початкового перегляду визначених критеріїв, було обрано два середовища розробки – PyCharm Community Editon та Visual Studio Code.

PyCharm Community Editon – це безкоштовне інтегроване середовище розробки, що було створено виключно для роботи з Python та має такі інструменти для роботи з цією мовою програмування : підтримка функцій автодоповнення, статичного аналізу коду програми. В це середовище розробки інтегровано відладчик для покрокового виконання коду, що дозволяє швидко знаходити помилки в тексті програми. На жаль, у PyCharm Community Editon повністю відсутня підтримка інструментів для роботи з БД, фреймворками для розробки вебзастосунків. Стосовно підтримки

функцій тестування – це середовище розробки підтримує вбудовану систему тестування unittest, проте інші способи тестування не підтримуються [12].

Visual Studio Code – IDE, що поєднує в собі функціонал текстового редактора та повноцінного середовища розробки. Visual Studio Code дуже швидко запускається через те, що без встановлених розширень, це середовище фактично являє собою текстовий редактор коду, а вже встановлені раніше пакети розширень запускаються динамічно, тобто лише за необхідністю. Процес встановлення пакетів розширення функціоналу середовища розробки відбувається безпосередньо в самому Visual Studio Code за допомогою відповідної вкладки “Extensions”, тому не викликає ніяких труднощів. Для повноцінної роботи з Python необхідно буде встановити відповідний плагін з аналогічною назвою, проте це не забере багато часу, але надасть широкий набір інструментів : підтримка функцій автодоповнення та рефакторингу, а також підсвічування синтаксису, що значно впливає на читабельність коду. Робота з базами даних, фреймворками та тестуванням в цьому середовищі розробки теж потребує встановлення необхідних розширень, але надає велику кількість можливостей : створення автозгенерованих тестів з повною візуалізацією даних, підключення до БД і редагування таблиць за допомогою графічного інтерфейсу користувача та через термінал, а також підтримка всіх необхідних фреймворків для розробки вебзастосунків завдяки взаємодії з ними через вбудований термінал. Єдиний мінус Visual Studio Code – відсутність офіційної підтримки від розробника, тобто не можна звернутися до служби підтримки з проханням отримати персональну допомогу. Але цей недолік є несуттєвим, оскільки це середовище розробки має велику за числом спільноту розробників, які обмінюються інформацією та допомагають один одному на форумах [13].

Зведені результати порівняння середовищ розробки, що розглядаються для створення вебзастосунку для бронювання готельних номерів, наведені в таблиці 2.2.

Таблиця 2.2 – Вибір середовища розробки для вебзастосунку “Бронювання готельних номерів”

Критерій для порівняння	PyCharm Community Editon	Visual Studio Code
Підтримка Python	+	+
Підтримка фреймворків для створення вебзастосунків	-	+
Підтримка інструментів для проведення тестування	+-	+
Підтримка інструментів для роботи з базами даних	-	+

Під час аналізу результатів з таблиці 2.2, було зроблено висновок, що необхідне середовище розробки для створення вебзастосунку для бронювання готельних номерів – Visual Studio Code, оскільки воно надає всі інструменти для роботи з Python, базами даних та тестуванням, підтримує фреймворки Django, Tornado та Flask, на відміну від PyCharm Community Editon. Звісно, платна версія PyCharm Pro надає такий самий функціонал, як і Visual Studio Code, але, як вже було зазначено раніше, в рамках виконання дипломної роботи бакалавра буде використано безкоштовні версії обраних середовищ розробки. Також потрібно відмітити, що через необхідність отримання всього необхідного функціоналу для створення вебзастосунку, в середовище розробки Visual Studio Code буде встановлено пакети розширень з такими назвами : “Python” – додає підтримку мови програмування Python та стандартні функції статичного аналізу коду, “Pylance” – підтримка функції швидкого автодоповнення коду, “SQLTools” – додає підтримку роботи з БД, “HTML CSS Support” – розширена підтримка функції автодоповнення коду CSS-класів у HTML-документах.

2.3 Вибір фреймворку

Перед початком аналізу фреймворків, що можна використати для розробки вебзастосунку для бронювання готельних номерів, варто визначитися з найважливішими параметрами, яким має відповідати обраний фреймворк. Згідно пункту 2.1, основний критерій для вибору вебфреймворку – він має бути створений для мови програмування Python.

Важливі критерії для вибору вебфреймворку:

- продуктивність : обраний фреймворк повинен мати змогу обробляти велику кількість запитів та економно використовувати ресурси серверної частини вебзастосунку. При цьому, швидкість обробки запитів не є головним показником – якщо він здатен обробляти більшу кількість запитів при несуттєвому зниженні швидкості, то цього буде достатньо. Тобто, основний пункт продуктивності для фреймворку – підтримка масштабованості, що дозволить системі працювати стабільно без суттєвого зниження швидкості виконання при збільшенні навантаження : кількості користувачів та їх запитів, обсягів даних;

- швидкість розробки : фреймворк має надавати необхідні вбудовані модулі для прискорення процесу розробки вебзастосунку. До таких модулів можна віднести : наявність вбудованої адмін-панелі, шаблону авторизації користувача, початкових функцій для роботи з базами даних. Структура проєкту, що формується фреймворком, має бути максимально простою для розуміння, що значно прискорить процеси початкового ознайомлення та старту розробки вебзастосунку для бронювання готельних номерів;

- підтримка пакетів розширень функціоналу : обраний фреймворк має надавати можливість додавання пакетів розширення функціоналу, що будуть корисними при створенні вебзастосунку для бронювання готельних номерів, зокрема : для створення списків фільтрів пропозицій для резервування тимчасового житла, для розширення процесу авторизації користувача (наприклад, авторизація за електронною поштою) та для додавання функцій

текстового редактора безпосередньо у вебзастосунок, що допоможе зручно редагувати опис готелів.

Після початкового перегляду визначених критеріїв, було обрано три фреймворки – Django, Tornado та Flask.

Django – фреймворк для розробки вебзастосунків на Python, що заснований на різновидності архітектури Model-View-Controller, а точніше на Model-View-Template. В Django Model відповідає за структуру та логіку операцій в БД, Template – за представлення даних користувачу у вигляді HTML-шаблонів сторінок, View – за обробку запитів і реалізацію бізнес-логіки. Тобто views - це функції або класи, які отримують запити від користувачів, взаємодіють з models (базами даних) та повертають відповіді користувачам у вигляді веб-сторінок через templates. Django підтримує кешування та асинхронні запити, що позитивно впливає на його продуктивність, проте структура Django – синхронна, тому швидкість обробки запитів є суттєво нижчою, ніж в інших фреймворках. Швидкість розробки на Django є однією з найвищих, адже цей фреймворк надає велику кількість готових модулів для реалізації різноманітних функціональних вимог: система авторизації, адмін-панель, робота з шаблонами вебсторінок та БД, валідація форм – все це підтримується в Django. Також в цьому фреймворку наявна можливість встановлення великої кількості пакетів розширення функціоналу, зокрема “Django-allauth” для авторизації через електронну пошту, “Django-filter” для фільтрації запитів до списку готелів, “Django-crispy-forms” для кращої валідації HTML-форм [15].

Tornado – асинхронний вебфреймворк та вебсервер для Python, що був розроблений для обробки одночасної великої кількості з’єднань. Tornado не має шаблону архітектури, до якого його можна стовідсотково віднести, адже компоненти цього фреймворку є гнучкими. Стандартний алгоритм функціонування Tornado : користувач надсилає HTTP-запит, який порівнюється в компоненті Application з зареєстрованими маршрутами, визначений обробник запитів RequestHandler асинхронно обробляє

надісланий користувачем запит, повертаючи відповідь у вигляді HTML-сторінки. Продуктивність цього фреймворку є однією з найвищих, адже завдяки асинхронній моделі є можливість обробки декількох тисяч з'єднань одночасно без втрати швидкості та без зростання споживання ресурсів серверної частини. Недоліки Tornado – майже повна відсутність початкових шаблонів модулів та мала кількість пакетів розширення функціоналу, зокрема немає вбудованої моделі для взаємодії з БД та адмін-панелі [16].

Flask – легкий за споживанням ресурсів серверної частини фреймворк для Python, що дозволяє створювати гнучку структуру проєкту. Тобто, додаються лише необхідні компоненти, а алгоритм роботи виглядає так : Flask ініціалізується як об'єкт вебзастосунку, для кожного посилання створюється відповідна функція обробки, після чого завдяки шаблонам “Jinja 2” формується відповідь у вигляді HTML або JSON. Flask не оптимізовано під асинхронні задачі, тому його продуктивність є значно меншою, ніж у Tornado. Швидкість розробки вебзастосунку з використанням цього фреймворку є дуже швидкою, але для реалізації адмін-панелі доведеться власноруч створювати окремий компонент, що може принести деякі незручності. Фактично, вся робота з Flask будується на пакетах розширення функціоналу, яких можна додати велику кількість, проте це може забрати багато часу через майже повну відсутність базових функціональних компонентів. Flask ідеально підійшов би для вебзастосунків малого розміру, гнучкість структури яких необхідно було б контролювати вручну, але, у випадку з вебзастосунком для бронювання готельних номерів, кількості базового функціоналу цього фреймворку не вистачить, а на встановлення всіх необхідних пакетів розширення піде значна кількість часу [17].

Зведені результати порівняння різних фреймворків для створення вебзастосунку для бронювання готельних номерів наведено в таблиці 2.3.

Таблиця 2.3 – Вибір фреймворку вебзастосунку для бронювання готельних номерів

Критерій порівняння	Django	Tornado	Flask
Підтримка Python	+	+	+
Продуктивність	+/-	+	+
Швидкість розробки	+	-	+/-
Підтримка пакетів розширення функціоналу фреймворку	+/-	-	+

Під час аналізу результатів з таблиці 2.3, було зроблено висновок, що необхідний фреймворк для створення вебзастосунку для бронювання готельних номерів – Django, адже він має суттєву перевагу над іншими розглянутими фреймворками в швидкості розробки через наявність великої кількості початкових шаблонів для створення вебзастосунку. Звісно, продуктивність Django є меншою, ніж у інших розглянутих фреймворків, проте цю різницю неможливо порівняти з тією, що надає Django при початку процесу розробки ПЗ.

2.4 Вибір бази даних

Перед початком аналізу різних баз даних, що можна використати для розробки вебзастосунку для бронювання готельних номерів, варто визначитися з найважливішими параметрами, яким має відповідати обрана БД. Згідно пункту 2.3, основний критерій для вибору БД – вона має підтримуватися фреймворком Django нативно, без необхідності додаткового розширення вебзастосунку.

Інші важливі критерії для вибору бази даних :

– встановлення, налаштування і підтримка : обрана БД не повинна потребувати встановлення і налаштування окремого серверу для роботи з

нею, адже це забере велику кількість часу, яку краще використати для розробки основного функціоналу вебзастосунку;

- споживання ресурсів серверної частини : БД повинна мати мінімальні системні вимоги і не навантажувати серверну частину вебзастосунку;

- складність даних : оскільки в цьому вебзастосунку не буде багато таблиць та складних зв'язків, то вимоги до підтримки складних запитів суттєво знижені. Пріоритетними критеріями для вибору БД будуть простота налаштування та мінімальне споживання ресурсів серверної частини.

Після початкового перегляду визначених критеріїв, було обрано три бази даних – PostgreSQL, MySQL, SQLite.

PostgreSQL – об'єктно-реляційна система керування базами даних, яка створена для зберігання складних структурованих даних. PostgreSQL підтримує складні SQL-запити, а також роботу з типами даних JSON та XML і надає можливість створення користувацької структури даних, що робить її функціонал безмежно великим. Проте, вимоги до серверних ресурсів у цієї БД дуже високі, адже при обробці складних запитів використовується багатоядерність процесора та великий об'єм оперативної пам'яті. PostgreSQL також потребує налаштування окремо серверу для коректної роботи, адже ця БД працює за принципом отримання запитів на сервер [18].

MySQL – реляційна система керування базами даних, що працює за моделлю клієнт-сервер. MySQL запускається як серверний процес, тому потребує налаштування окремого серверу : користувач надсилає SQL-запити, а сервер обробляє їх і повертає результат. Ця БД надає підтримку для роботи з JSON, складними запитамі SQL, дозволяє одночасно обробляти велику кількість з'єднань. Проте, підтримка складних типів даних гірша, ніж в PostgreSQL, має обмеження з транзакціями в деяких типах таблиць. Стосовно вимог до ресурсів – вони нижчі, ніж в PostgreSQL, проте все одно є суттєвими, зокрема, є необхідність у виділенні пам'яті для кешування [19].

SQLite – вбудована реляційна база даних, що не потребує налаштування окремо серверу для коректної роботи. SQLite повністю інтегрується в програму завдяки підключення відповідного файлу у файлову систему вебзастосунку. Загалом, ця БД має менше функцій, ніж раніше розглянуті аналоги, зокрема – не підтримує одночасний багатокористувацький запис, має обмеженість у складності SQL-запитів. Проте, SQLite є найкращим варіантом для серверної частини, бо споживає мінімальну кількість ресурсів. Взагалом, дана БД ідеально підходить для проєктів невеликого розміру, що мають необхідність у швидкому розгортанні системи для прискорення процесу розробки [20].

Зведені результати порівняння різних баз даних для створення вебзастосунку для бронювання готельних номерів наведено в таблиці 2.4.

Таблиця 2.4 – Вибір бази даних вебзастосунку для бронювання готельних номерів

Критерій порівняння	PostgreSQL	MySQL	SQLite
Підтримка Django	+	+	+
Просте налаштування і встановлення	-	-	+
Споживання ресурсів серверної частини вебзастосунку	-	+-	+
Складність даних, що будуть зберігатися в базі	+	+-	-

Під час аналізу результатів з таблиці 2.4, було зроблено висновок, що необхідний база даних для створення вебзастосунку для бронювання готельних номерів – SQLite, адже вона підтримує прості SQL-запити, не потребує налаштування окремого серверу для коректної роботи та має

мінімальні вимоги до споживання ресурсів серверної частини вебзастосунку. Так, складність даних та інструментів для роботи з ними, що підтримуються в SQLite набагато нижча, ніж в інших розглянутих БД, але навіть такого простого набору інструментів вистачить для створення вебзастосунку в рамках цієї роботи.

2.5 Висновок за розділом 2

Під час проведення аналізу існуючих варіантів мов програмування, середовищ розробки, фреймворків та баз даних, що підійдуть для створення вебзастосунку для бронювання готельних номерів, було визначено :

- в якості мови програмування було обрано мову Python, адже вона має простий синтаксис, підтримує багато фреймворків для веброзробки;
- в якості середовища розробки було обрано Visual Studio Code , адже воно є абсолютно безкоштовним та підтримує мову програмування Python, а його функціонал легко розширюється за допомогою встановлення пакетів розширення функціональності;
- в якості фреймворку було обрано Django, адже він створений для мови програмування Python та має початковий набір шаблонів для створення необхідного функціоналу вебзастосунку, що дозволить швидко розробити ПЗ та приступити до процесу тестування;
- в якості бази даних було обрано SQLite, бо ця БД не потребує розгортання окремого серверу та не споживає багато ресурсів, при цьому надаючи мінімальний необхідний функціонал для розробки вебзастосунку для бронювання готельних номерів.

3 ОПИС ПРОГРАМИ

3.1 Архітектура вебзастосунку для бронювання готельних номерів

Архітектуру вебзастосунку для бронювання готельних номерів, що створюється на основі фреймворку Django [15], наведено на рисунку 3.1.

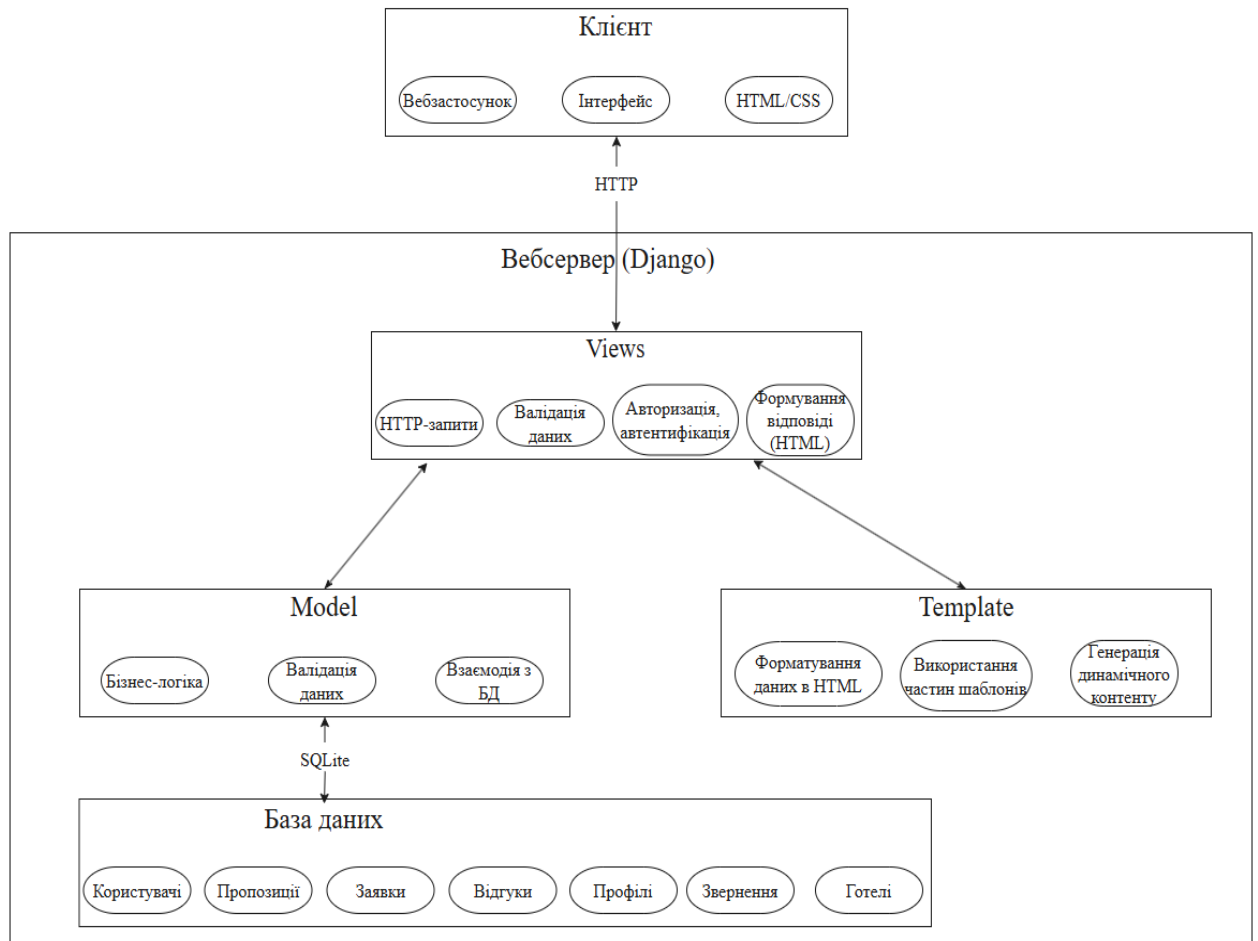


Рисунок 3.1 – Архітектура вебзастосунку для бронювання готельних номерів

Опис модулів, зображених на рисунку 3.1:

– клієнт : відповідає за взаємодію користувача з вебзастосунком. Цей модуль надсилає HTTP-запити до сервера, отримуючи відповідь у вигляді HTML-документів. Тобто, клієнт відображає весь інтерфейс вебзастосунку [15];

- вебсервер (Django): являє собою частину, де реалізована бізнес-логіка, обробка запитів та робота з БД. Сервер складається з трьох різних модулів, кожен з яких має своє призначення [15];

- views (подання): обробляє HTTP-запити від клієнта, викликає відповідні моделі для отримання та збереження даних, повертає в клієнт відповідь у вигляді HTML-документів. Тобто, в цьому модулі виконуються всі логічні операції вебзастосунку [15];

- model (моделі) : являє собою структуру даних у вигляді класів Python, що визначають, яким чином зберігаються дані в БД. Фактично, даний модуль відповідає за створення таблиць , оновлення та валідацію даних в таблицях БД [15];

- template (шаблони) : відповідає за генерування HTML-сторінок на основі даних, що отримуються з модуля Views [15];

- база даних : зберігає всю інформацію, що необхідна для роботи вебзастосунку. Модуль Model звертається до БД через ORM, відповідно, дані можуть оновлюватися, фільтруватися та змінюватися [15].

Загальний потік роботи вебзастосунку для бронювання готельних номерів можна сформулювати за розробленою раніше архітектурою вебзастосунку таким чином: користувач надсилає HTTP-запит з браузера, який отримує модуль Views. Views в свою чергу викликає модуль Model, щоб отримати або змінити дані в модулі БД, а потім передає необхідні дані в модуль Template. Template формує відповідь у вигляді HTML-документу, який повертається в якості відповіді у клієнт.

Для більшого розуміння основного призначення та функції вебзастосунку – здійснення процесу бронювання готельного номеру, необхідно описати загальний перелік дій, що є можливими під час роботи з вебзастосунком.

Загальний опис дій, що можуть виконувати користувачі під час роботи з вебзастосунком для бронювання готельних номерів, представлений у вигляді діаграми прецедентів, наведено на рисунку 3.2.

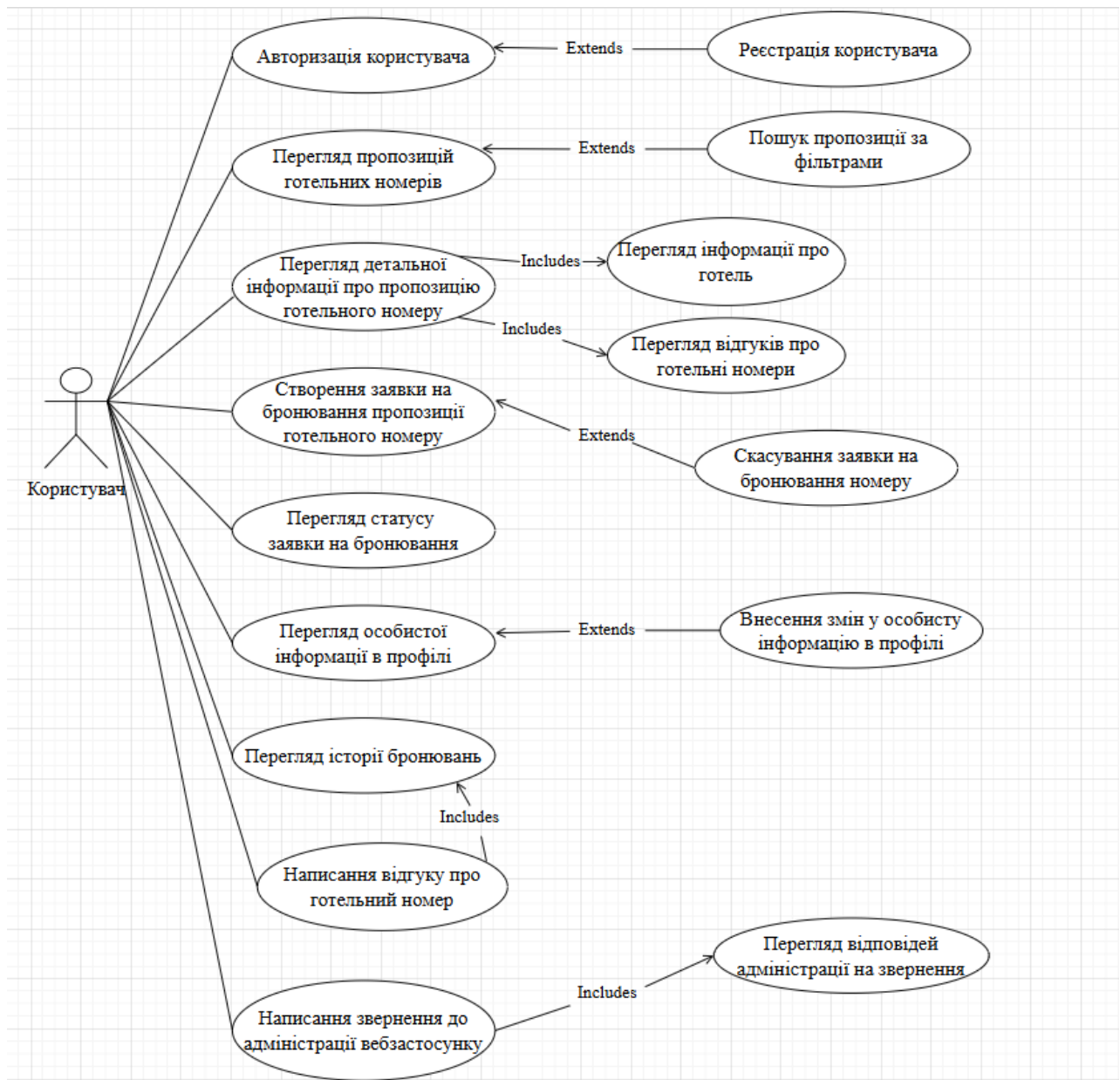


Рисунок 3.2 – Діаграма прецедентів користувача для застосунку
“Бронювання готельних номерів”

Основний потік сценаріїв роботи користувача з вебзастосунком для бронювання готельних номерів:

- авторизація користувача;
- перегляд пропозицій готельних номерів;
- перегляд детальної інформації про пропозицію готельного номеру, що включає сценарії перегляду відгуків про готельні номери та інформації про готель, адже користувачеві необхідно бачити не тільки характеристики

обраного готельного номеру, а й оцінки готелю та номерів взагалом на основі відгуків інших користувачів;

- створення заявки на бронювання готельного номеру;
- перегляд статусу заявки на бронювання;
- перегляд особистої інформації в профілі;
- перегляд історії бронювань;
- написання відгуку про готельний номер, що включає сценарій перегляду історії бронювань, адже користувач може писати відгук лише до тих пропозицій готельних номерів, в яких він сам жив;
- написання звернення до адміністрації вебзастосунку, що включає перегляд відповідей адміністрації, адже є необхідність у забезпеченні зворотнього зв'язку.

Альтернативні сценарії основного потоку роботи користувача з вебзастосунком для бронювання готельних номерів:

- реєстрація користувача, що є альтернативним сценарієм для авторизації користувача, коли він не має зареєстрованого профілю для виконання процесу авторизації;
- пошук пропозиції за фільтрами, що є альтернативним сценарієм для перегляду пропозицій готельних номерів, коли в користувача є необхідність знайти конкретну пропозицію за вказаними характеристиками, а не переглядати всі можливі пропозиції та обирати власноруч без фільтрів;
- скасування заявки на бронювання готельного номеру, що є альтернативним сценарієм для створення заявки на бронювання пропозиції готельного номеру, коли користувач має необхідність відмінити вже існуючу заявку на бронювання;
- внесення змін у особисту інформацію в профілі, що є альтернативним сценарієм для перегляду особистої інформації в профілі, коли треба замінити особисті дані.

Загальний опис дій, що можуть виконувати адміністратори вебзастосунку під час роботи з вебзастосунком для бронювання готельних номерів, представлений у вигляді діаграми прецедентів, наведено на рисунку 3.3.

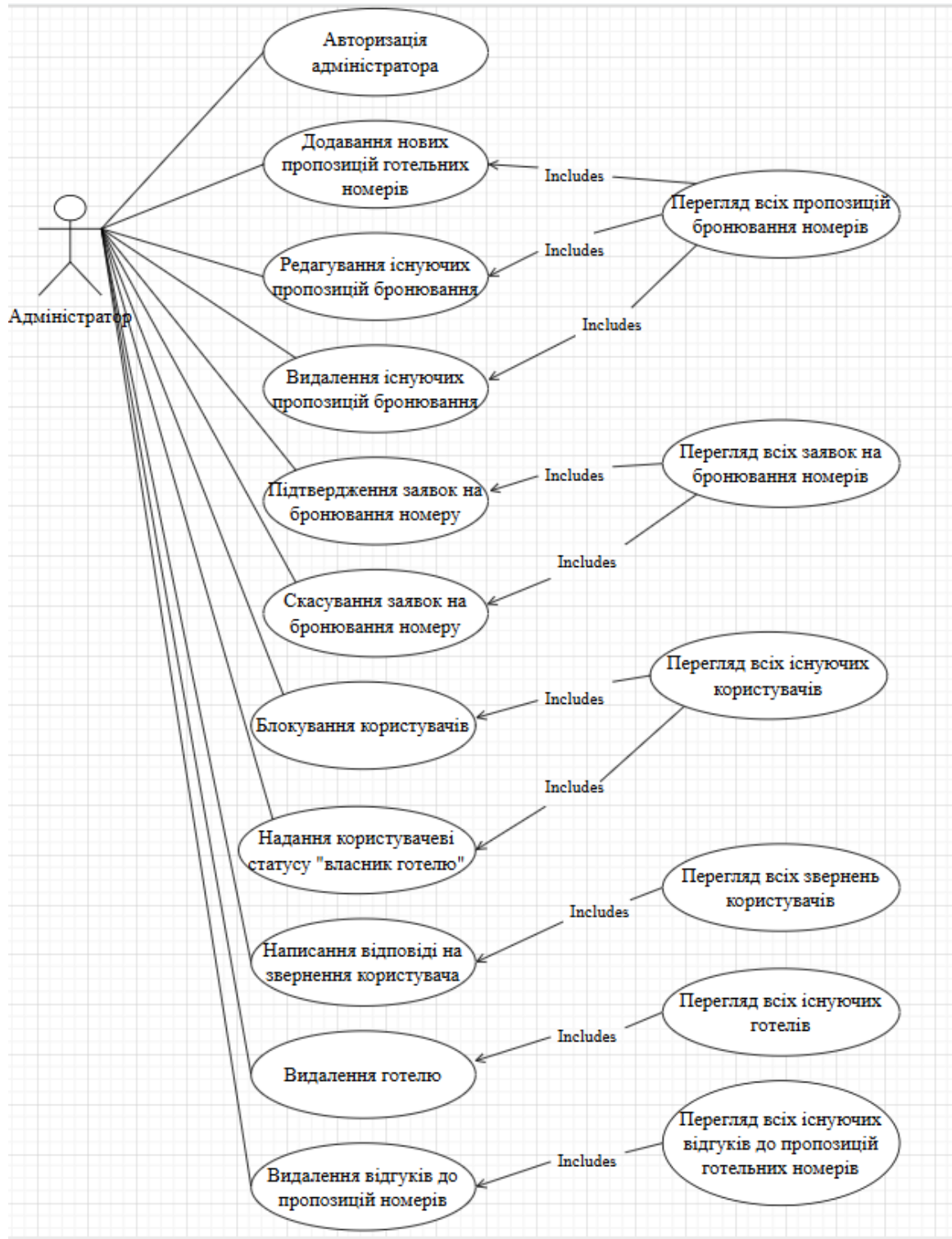


Рисунок 3.3 – Діаграма прецедентів адміністратора

Основний потік сценаріїв роботи адміністратора з вебзастосунком для бронювання готельних номерів:

- авторизація адміністратора;
- додавання нових пропозицій готельних номерів, що включає сценарій перегляду всіх пропозицій бронювання номерів для уникнення дублювання пропозицій;
- редагування існуючих пропозицій бронювання, що включає сценарій перегляду всіх пропозицій бронювання номерів;
- видалення існуючих пропозицій бронювання, що включає сценарій перегляду всіх пропозицій бронювання номерів;
- підтвердження заявок на бронювання номеру, що включає сценарій перегляду всіх заявок на бронювання готельних номерів;
- скасування заявок на бронювання номеру, що включає сценарій перегляду всіх заявок на бронювання готельних номерів;
- блокування користувачів, що включає сценарій перегляду всіх існуючих користувачів;
- надання користувачеві статусу “Власник готелю”, що включає сценарій перегляду усіх існуючих користувачів;
- написання відповіді на звернення користувача, що включає сценарій перегляду всіх звернень користувачів;
- видалення готелю, що включає сценарій перегляду всіх існуючих готелей;
- видалення відгуків до пропозицій номерів, що включає сценарій перегляду всіх існуючих відгуків до пропозицій готельних номерів.

Примітка : адміністратор не може реєструватися, адже він визначений у вебзастосунку заздалегідь, що виключає можливість реєстрації інших адміністраторів у вебзастосунку для бронювання готельних номерів.

Загальний опис дій, що можуть виконувати власники готелів під час роботи з вебзастосунком для бронювання готельних номерів, представлений у вигляді діаграми прецедентів, наведено на рисунку 3.4.

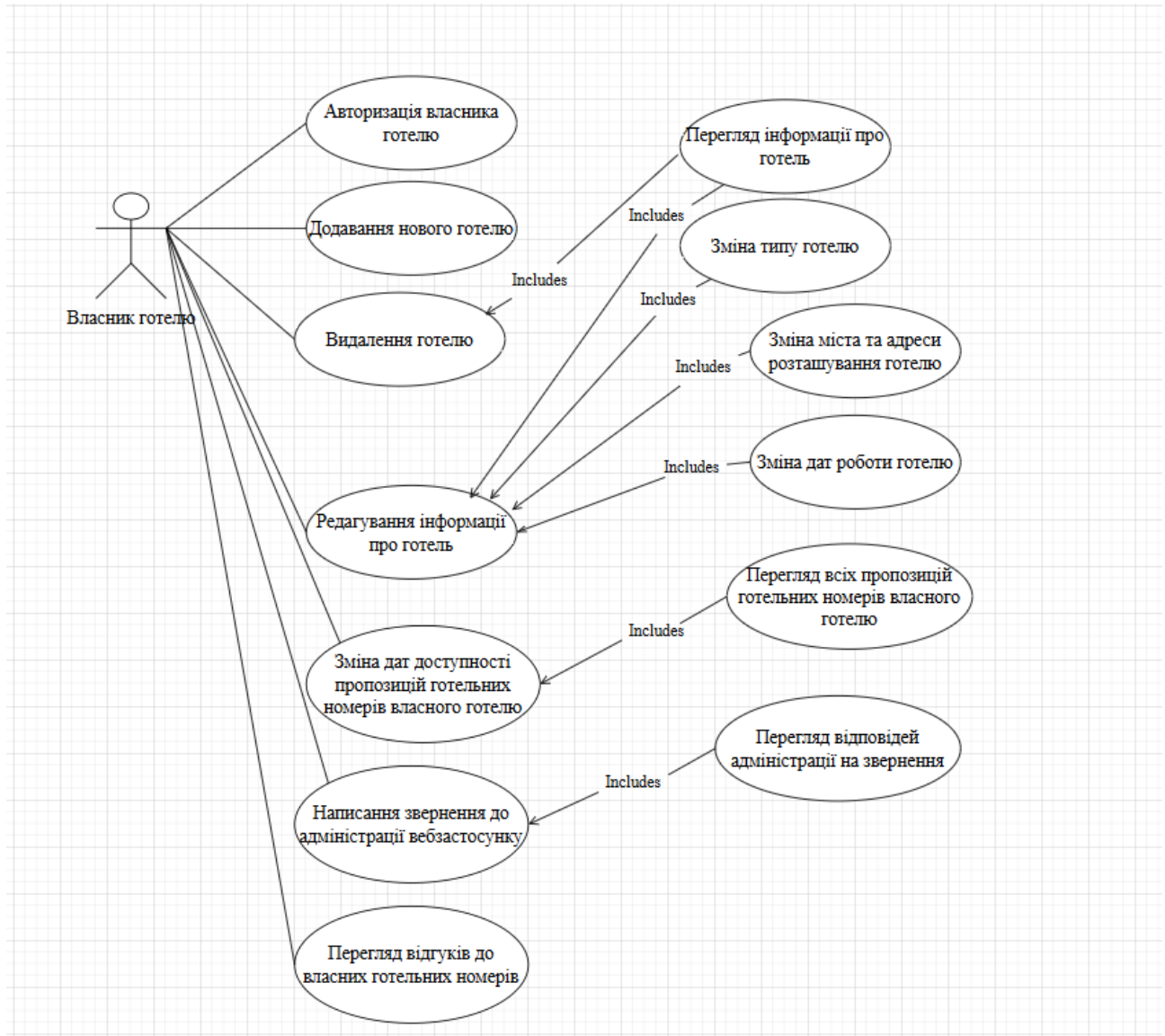


Рисунок 3.4 – Діаграма прецедентів власника готелю

Основний потік сценаріїв роботи власника готелю з вебзастосунком для бронювання готельних номерів:

- авторизація власника готелю;
- додавання нового готелю;
- видалення готелю;

- редагування інформації про готель, що включає сценарій перегляду інформації, зміни типу, зміни міста та адреси, а також зміну дат роботи готелю, що дозволяє власникові повністю керувати своїм готелем;
- зміна дат доступності пропозицій готельних номерів власного готелю, що включає сценарій перегляду всіх пропозицій готельних номерів у власному готелі;
- написання звернення до адміністрації вебзастосунку, що включає сценарій перегляду відповідей адміністрації на звернення для необхідності забезпечення зворотнього зв'язку;
- перегляд відгуків до власних готельних номерів.

Варто зазначити, що в діаграмах прецедентів власника готелю та адміністратора немає альтернативних сценаріїв роботи з вебзастосунком для бронювання готельних номерів, адже їх основне і єдине завдання – постійно редагувати готелі, пропозиції та інші об'єкти, а не просто переглядати їх. Перегляд інформації у випадку з власниками готелів та адміністратором вебзастосунку – це лише інструмент, без якого є неможливим виконання своїх основних обов'язків, для яких і необхідне їх існування в рамках вебзастосунку.

Для більшого розуміння логіки роботи вебзастосунку для бронювання готельних номерів, необхідно навести загальні стани, в яких може перебувати пропозиція для бронювання готельного номеру. Діаграму станів для об'єкту “Пропозиція готельного номеру” наведено на рисунку 3.5.

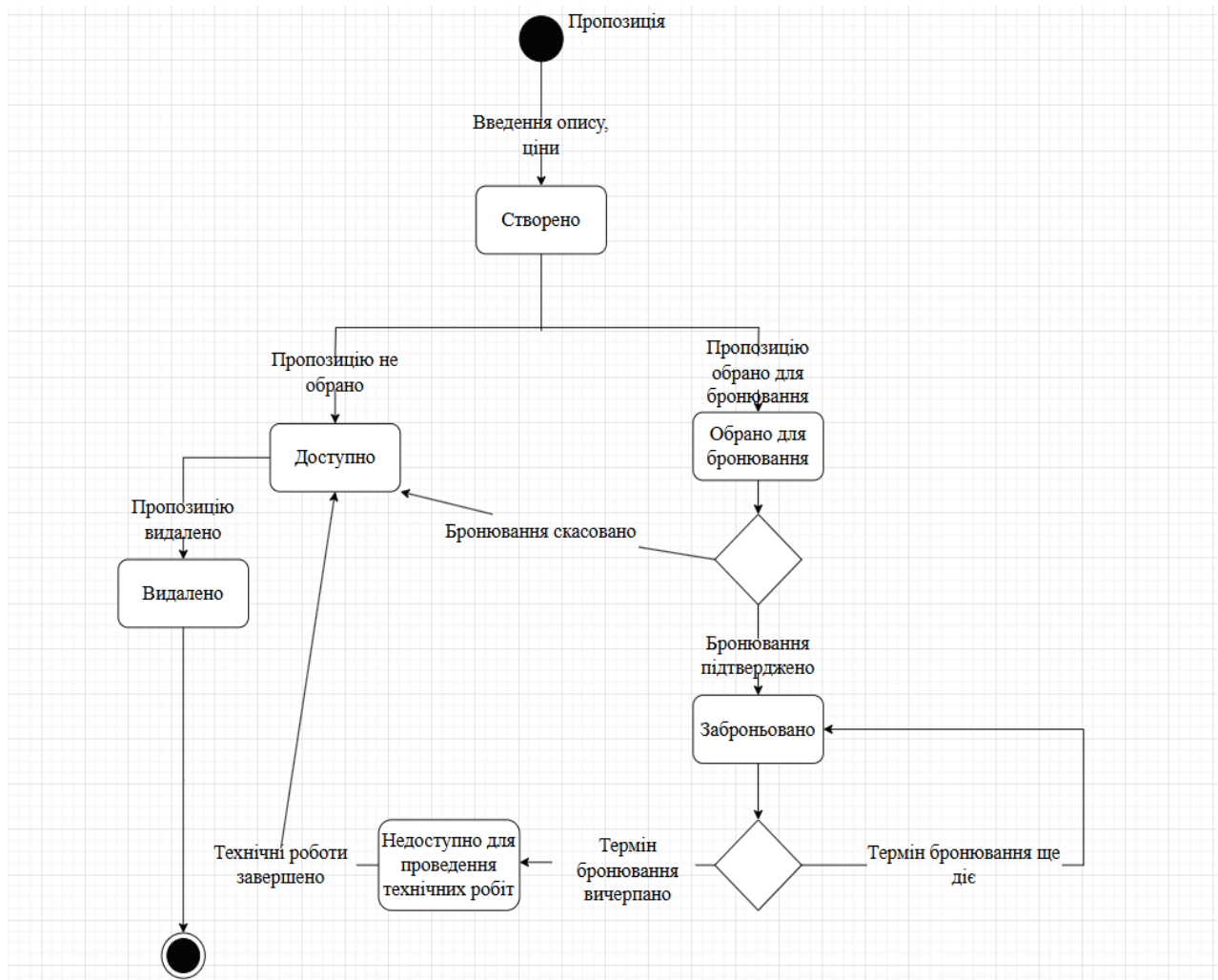


Рисунок 3.5 – Діаграма станів для об'єкту “Пропозиція готельного номеру”

Отже, після введення ціни, типу та опису готельного номеру, пропозиція набуває стану “Створено”. Якщо її ще незарезервовано – то вона має стан “Доступно”, якщо вона обрана для бронювання – отримує статус “Обрано для бронювання”. У випадку підтвердження бронювання, пропозиція отримує стан “Заброньовано” і зникає з переліку пропозицій для бронювання на термін її резервації, але після скасування бронювання готельного номеру пропозиція повертається до стану “Доступно”. Якщо термін бронювання ще діє, то пропозиція залишається у статусі “Заброньовано”, але якщо термін вичерпано - то вона переходить до стану “Недоступно для проведення технічних робіт”, що зроблено для того, аби привести номер у належний вигляд для бронювання після виселення минулих клієнтів, термін дії технічних робіт - 1 день. Відповідно, після завершення

технічних робіт, пропозиція повертається до стану “Доступно” і знову з’являється у переліку пропозицій, що доступні для потенційного резервування. Також пропозицію для бронювання можна видалити, що може зробити адміністратор вебзастосунку, відповідно, вона набуде стану “Видалено”.

3.2 Опис розробленої бази даних вебзастосунку для бронювання готельних номерів

Розроблена база даних для вебзастосунку “Бронювання готельних номерів”, що зображена у вигляді схеми, наведена на рисунку 3.6.

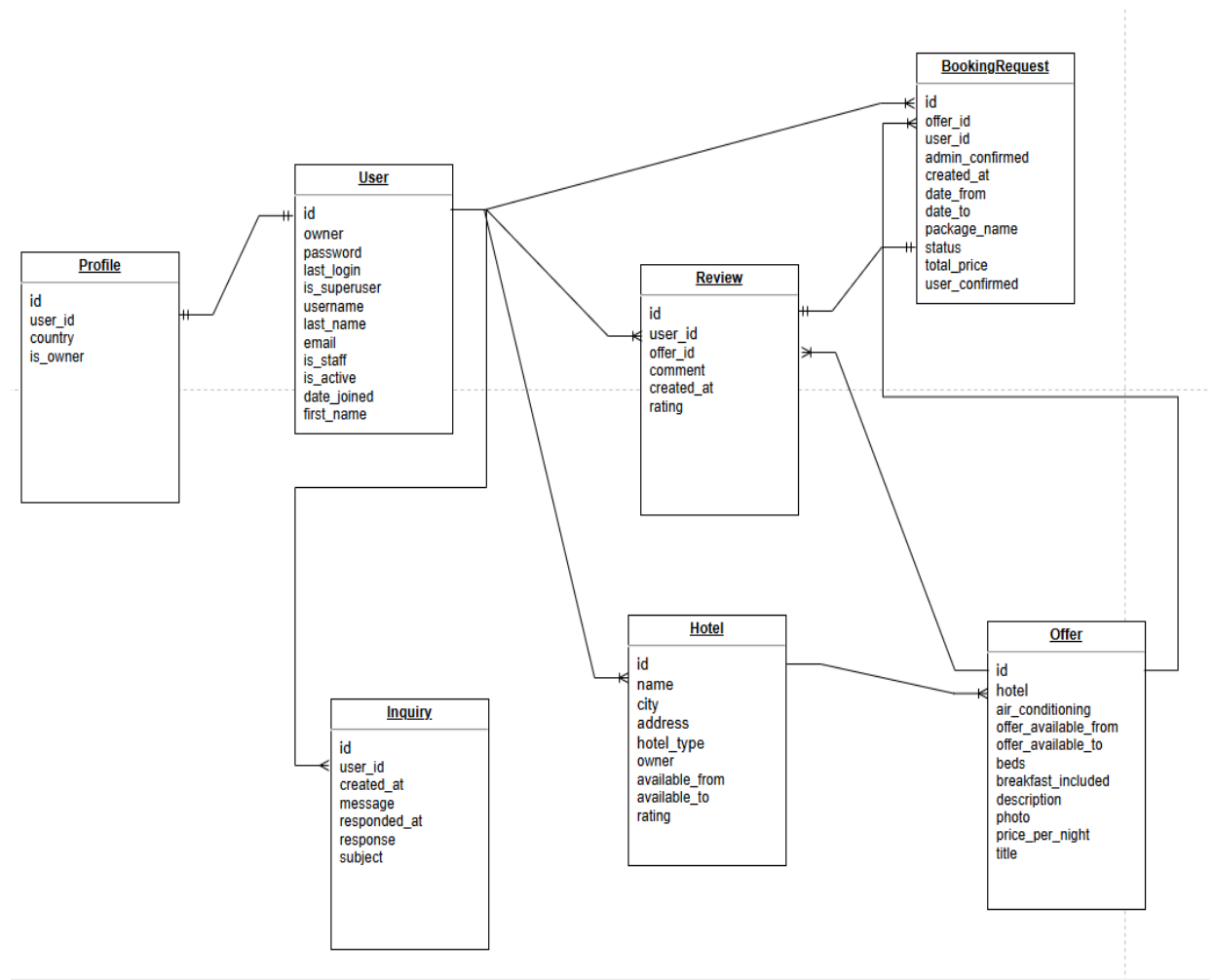


Рисунок 3.6 – Зображення розробленої БД у вигляді схеми

Короткий опис полів всіх таблиць створеної БД для вебзастосунку “Бронювання готельних номерів” наведено в таблиці 3.1.

Таблиця 3.1 – Призначення полів з таблиць БД

Назва таблиці	Назва поля	Призначення
User	id	Унікальний номер користувача
	owner	Визначає, чи є користувач власником готелю
	password	Пароль користувача. У БД пароль зберігається у вигляді хешу завдяки вбудованому захисту Django
	last_login	Остання дата та час, коли користувач заходив зі свого акаунту у вебзастосунок
	is_superuser	Визначає, чи є користувач адміністратором вебзастосунку
	username	Нікнейм користувача
	last_name	Прізвище користувача
	email	Електронна пошта користувача
	is_staff	Визначає, чи має користувач доступ до деяких функцій адмін-панелі
	is_active	Визначає, чи зайшов зараз цей користувач у вебзастосунок через свій профіль
	date_joined	Дата реєстрації користувача у вебзастосунку
	first_name	Ім'я користувача
Profile	id	Унікальний номер створеного профілю
	user_id	Унікальний номер користувача
	is_owner	Визначає, чи є користувач власником

Продовження таблиці 3.1

Назва таблиці	Назва поля	Призначення
Inquiry	id	Унікальний номер звернення користувача
	user_id	Унікальний номер користувача
	created_at	Дата створення звернення користувачем
	message	Текст звернення користувача
	responded_at	Дата відповіді адміністратора на звернення користувача
	responce	Текст відповіді адміністратора на звернення користувача
	subject	Тема звернення користувача
Booking-Request	id	Унікальний номер заявки на бронювання
	offer_id	Унікальний номер пропозиції для бронювання
	user_id	Унікальний номер користувача
	admin_confirmed	Поле для готелей з повним спектром послуг для здійснення процесу подвійного підтвердження бронювання готельного номеру
	created_at	Дата створення заявки на бронювання номеру
	date_from	Дата, з якої користувач бажає забронювати готельний номер
	date_to	Дата, до якої користувач бажає здійснити бронювання
	package_name	Поле для бутік-готелей, що дозволяє оформити бронювання у вигляді “Назва події + номер”

Продовження таблиці 3.1

Назва таблиці	Назва поля	Призначення
Booking-Request	status	Статус заявки на бронювання
	total_price	Ціна на готельний номер з урахуванням типу готелю та вказаної кількості ночей
	user_confirmed	Поле для готелів з повним спектром послуг, що допомагає здійснити процедуру подвійного бронювання
Review	id	Унікальний номер відгуку
	offer_id	Унікальний номер пропозиції для бронювання
	user_id	Унікальний номер користувача
	comment	Текст відгуку
	created_at	Дата створення відгуку
	rating	Оцінка пропозиції для бронювання
Offer	id	Унікальний номер пропозиції для бронювання готельного номеру
	hotel	Назва готелю, з якого зроблена пропозиція про бронювання готельного номеру
	offer_available_from	Дата, з якої пропозиція стає доступною
	offer_available_to	Дата, до якої пропозиція є доступною
	beds	Кількість спальних місць в номері
	breakfast_included	Чи є сніданок в пропозиції обраного номеру
	air_conditioning	Чи є в пропозиції обраного номеру кондиціонер
	description	Опис пропозиції для бронювання
	photo	Фото пропозиції для бронювання

Продовження таблиці 3.1

Назва таблиці	Назва поля	Призначення
Offer	price_per_night	Вартість однієї ночі в готельному номері
	title	Назва пропозиції для бронювання
Hotel	id	Унікальний номер готелю
	name	Назва готелю
	city	Назва міста, в якому розташований готель
	address	Адреса, за якою розташований готель
	hotel_type	Тип готелю
	owner	Унікальний номер власника готелю
	available_from	Дата, з якої працює готель
	available_to	Дата, до якої працює готель
	rating	Рейтинг готелю

3.3 Опис файлів, що реалізують логіку вебзастосунку для бронювання готельних номерів

Перед початком опису файлів варто зазначити, що фрагменти коду, який міститься в цих файлах, будуть розміщені в Додатку Б – Код програми. Логіка роботи вебзастосунку для бронювання готельних номерів поділяється на дві групи : бізнес-логіка – всі правила для ефективної роботи вебзастосунку, службова логіка – дозволяє фреймворку Django успішно організувати бізнес-логіку. Список файлів, що реалізують всю логіку роботи вебзастосунку для бронювання готельних номерів, наведено на рисунку 3.7.

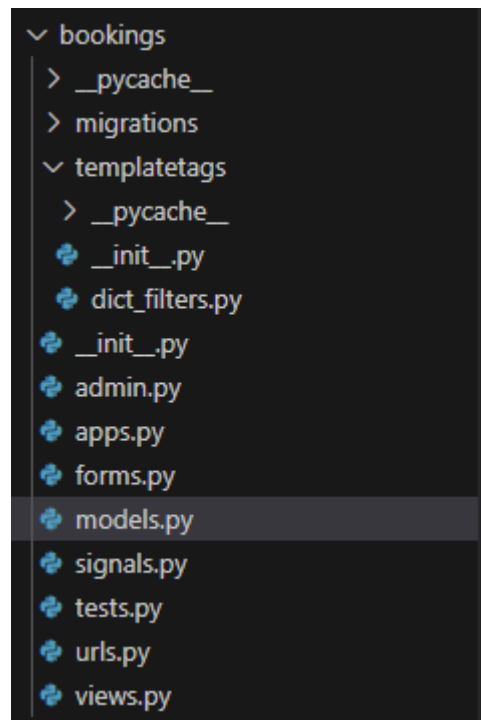


Рисунок 3.7 – Файли, що реалізують логіку роботи вебзастосунку для бронювання готельних номерів

Короткий опис кожного з файлів, що реалізують бізнес-логіку:

– Models.py: файл, в якому описуються моделі, що визначають структуру БД. Згідно рисунку 3.4, створено сім моделей : User – автоматично створена Django модель для зберігання даних про користувачів, Profile – модель для визначення структури профілю користувача, Offer – модель для пропозицій бронювання готельних номерів, BookingRequest – структура заявок на бронювання готельних номерів, Review – вигляд відгуків до пропозицій, Inquiry – модель для форми спілкування користувачів з адміністрацією вебзастосунку, Hotel – модель для зберігання даних про готель [15];

– Forms.py: містить в собі всі форми для введення, перевірки даних, що вводить користувач. Під час розробки вебзастосунку було реалізовано такі форми : RegistrationForm – реєстрація користувача на сайті, EditProfileForm – зміна персональних даних в особистому профілі користувача, BookingForm – здійснення процесу бронювання обраної пропозиції готельного номеру в якій інтегровано сторонній сервіс Stripe для

здійснення процесу оплати, ReviewForm – написання відгуку до пропозиції готельного номеру, InquiryForm – звернення користувачів до адміністрації вебзастосунку;

- Signals.py : файл для автоматизації дій при зміні даних, був реалізований для створення профілю при реєстрації користувача (create_profile), та для автоматичного оновлення даних в БД про користувача після зміни інформації в профілі (save_profile) [15];

- Views.py : обробка запитів та підготовка даних до подальшого відображення. Під час розробки вебзастосунку для бронювання готельних номерів було реалізовано такі функції : Profile_views - функція для перегляду профілю, що надає можливість залишити питання для адміністрації вебзастосунку через форму InquiryForm та залишити відгук до пропозицій що вже були підтверджені адміністратором та в яких користувач гарантовано проживав завдяки ReviewForm. Edit_profile_view – дозволяє змінювати персональні дані користувача, а Payment_details_view – оновлювати платіжні дані в профілі. Login_view та Register_view реалізують логіку авторизації та реєстрації на вебзастосунку, Search_results являє собою фільтри та сортування на сторінці з пошуком пропозицій. Offer_detail відкриває однойменну HTML-сторінку для перегляду всієї інформації про обрану пропозицію, а Book визначає всю логіку здійснення процесу бронювання готельного номеру [15].

Короткий опис кожного з файлів, що реалізують службову логіку:

- Urls.py : визначає, які саме адреси обробляються конкретними функціями [15];

- Admin.py : файл для адмін-панелі, в якому реалізовано всі необхідні дії для адміністрування вебзастосунком : додавання пропозицій та їх видалення, зміна вже існуючих варіантів для резервування готельних номерів, формування відповідей на звернення користувачів, прийняття та відхилення заявок на бронювання, видалення відгуків та блокування користувачів [15].

Детальний опис процесу оплати готельного номеру в формі BookingForm завдяки використанню стороннього сервісу Stripe : Stripe – це платіжна платформа, що дозволяє легко інтегрувати систему онлайн-платежів у вебзастосунки через використання API-ключа. Всі платіжні дані під час оплати шифруються, а також підтримуються картки різних типів. Оскільки під час виконання цієї роботи було поставлено вимогу у абсолютній безкоштовності всіх ресурсів для успішної роботи вебзастосунку, ніякі платежі зі списанням грошових коштів проводитись не будуть : Stripe надає тестові платіжні дані від карток, які дозволять продемонструвати процес оплати готельного номеру абсолютно безкоштовно [22].

3.4 Опис файлів, що реалізують інтерфейс користувача та конфігурацію вебзастосунку для бронювання готельних номерів

Список файлів та директорій, в яких реалізується представлення даних користувачеві та загальні налаштування вебзастосунку, наведено на рисунку 3.8.

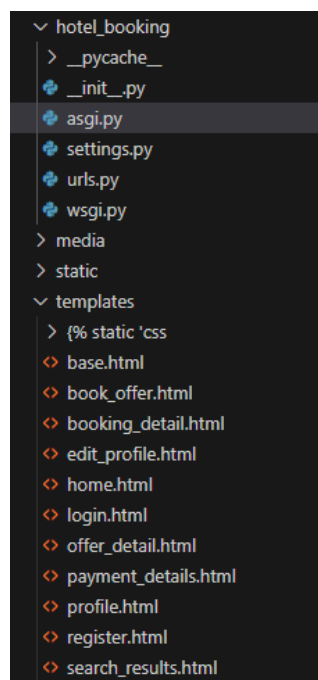


Рисунок 3.8 – Файли, що реалізують інтерфейс користувача та налаштування

Короткий опис файлів, в яких реалізовано необхідні налаштування вебзастосунку для бронювання готельних номерів :

- Settings.py : в ньому містяться встановлені Apps для роботи Django, а також необхідні механізми для захисту даних, реалізації шаблонів сторінок, БД. Фактично, це основний файл для налаштувань проєкту, в якому також знаходяться підключені директорії статичних файлів, валідатори для перевірки даних. Все, що Django може надати в якості шаблонів та механізмів для початку швидкої розробки вебзастосунку, підключається саме в цьому файлі [15];

- Urls.py : в цьому файлі знаходяться глобальні маршрути, включаючи ті, що знаходяться у файлі з аналогічною назвою, який реалізує службову логіку. Це центральна точка маршрутизації, яка розподіляє запити на відповідні частини вебзастосунку [15];

- Wsgi.py, asgi.py : інтерфейси для серверів Web Server Gateway Interface та Asynchronous Server Gateway Interface, але вони не використовуються в цій роботі через використання вбудованого вебсерверу від Django [15].

Короткий опис файлів, в яких реалізовано інтерфейс користувача:

- Base.html : базовий шаблон, що містить в собі необхідні кнопки навігації та в подальшому наслідується кожним HTML-документом;

- Home.html : шаблон для відображення стартової сторінки вебзастосунку;

- Login.html : реалізація сторінки входу користувача у вебзастосунок;

- Register.html : реєстрація користувача у вебзастосунку, автоматичне створення профілю;

- Profile.html : шаблон для відображення сторінки профілю користувача;

- Edit_profile.html : шаблон для зміни персональних даних в профілі користувача;

- Payment_details.html: сторінка для зміни платіжних даних;

- Search_results.html : сторінка для відображення результатів пошуку пропозицій для бронювання готельних номерів за фільтрами;
- Offer_detail.html : перегляд повної інформації про обраний варіант для резервування;
- Booking_detail.html : сторінка для відображення деталей бронювання.

В директорії Static знаходяться CSS файли для підтримки роботи файлів які реалізують інтерфейс користувача, а в директорії Media знаходяться фотографії пропозицій для бронювання.

3.5 Висновки за розділом 3

Проаналізувавши архітектуру вебзастосунку на фреймворку Django та створивши діаграми прецедентів, станів та базу даних, було розроблено вебзастосунок “Бронювання готельних номерів”. Реалізовано всі необхідні функціональні вимоги, зокрема : пошук за фільтрами (за наявністю сніданку, кондиціонеру, кількості спальних місць і типом готелю) та сортування результатів (за популярністю, датою заселення і виселення), перегляд та бронювання готельних номерів з урахуванням таких особливостей : готелі з повним спектром послуг потребують подвійного підтвердження, номери в бутік-готелях резервуються за допомогою “пакету-пропозиції” згідно структурі “назва події + назва номеру” , кімнати в бюджетних готелях не мають фільтрів що стосуються сервісу, номери в курортних готелях мають динамічне ціноутворення завдяки сезонності, а номери в готелях тривалого проживання можливо зарезервувати тільки на термін від 7 днів. Також реалізовано зміну даних в профілі користувача, підтримку історії бронювань для можливості написання відгуків, форму для зв’язку з адміністрацією вебзастосунку, здійснення модерації та адміністрування через створену адмін-панель. Здійснення процесу оплати готельного номеру без списання коштів реалізовано через сторонній платіжний сервіс Stripe.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення та умови застосування вебзастосунку

Розроблений вебзастосунок “Бронювання готельних номерів” створений для спрощення процесу бронювання житла в закладах тимчасового проживання за рахунок автоматизації. Вебзастосунок для бронювання дозволяє шукати варіанти готельних номерів та резервувати їх, отримувати відповіді на всі питання від адміністрації сайту, писати відгуки й залишати оцінки на пропозиціях з раніше заброньованими готельними номерами.

Для використання вебзастосунку “Бронювання готельних номерів” необхідно мати таке апаратне забезпечення:

- процесор з мінімум двома ядрами, тактова частота яких є не нижчою за 1.5 ГГц, а також з вбудованим графічним ядром;
- від 2 гб оперативної пам’яті;
- рекомендована вимога : наявність SSD-накопичувача для прискорення роботи системи, проте для мінімальних вимог вистачить будь-якого накопичувача з мінімальним місцем 5 гб.

В якості програмного забезпечення, необхідно мати операційну систему Windows 10 або аналоги macOS та Linux, а також встановлений один з браузерів Chrome, Firefox, Edge, Safari. Для розгортання системи, необхідно мати встановлений Python, середовище розробки Visual Studio Code, а також налаштовані через термінал пакети для роботи з Django, SQLite, Stripe.

4.2 Опис процесу роботи з розробленим вебзастосунком для бронювання готельних номерів

Використання вебзастосунку “Бронювання готельних номерів” є одночасно зручним і для користувачів, і для адміністраторів, адже весь

процес роботи відбувається через використання навігаційних кнопок та форм, що спрощує процес розуміння функціонування вебзастосунку.

Головна сторінка вебзастосунку для бронювання готельних номерів містить пошуковий рядок та кнопку авторизації, що дозволяє користувачеві швидко зрозуміти, які необхідно виконати дії для подальшого процесу перегляду та бронювання житла в закладах тимчасового проживання. Вигляд головної сторінки вебзастосунку для бронювання готельних номерів наведено на рисунку 4.1.

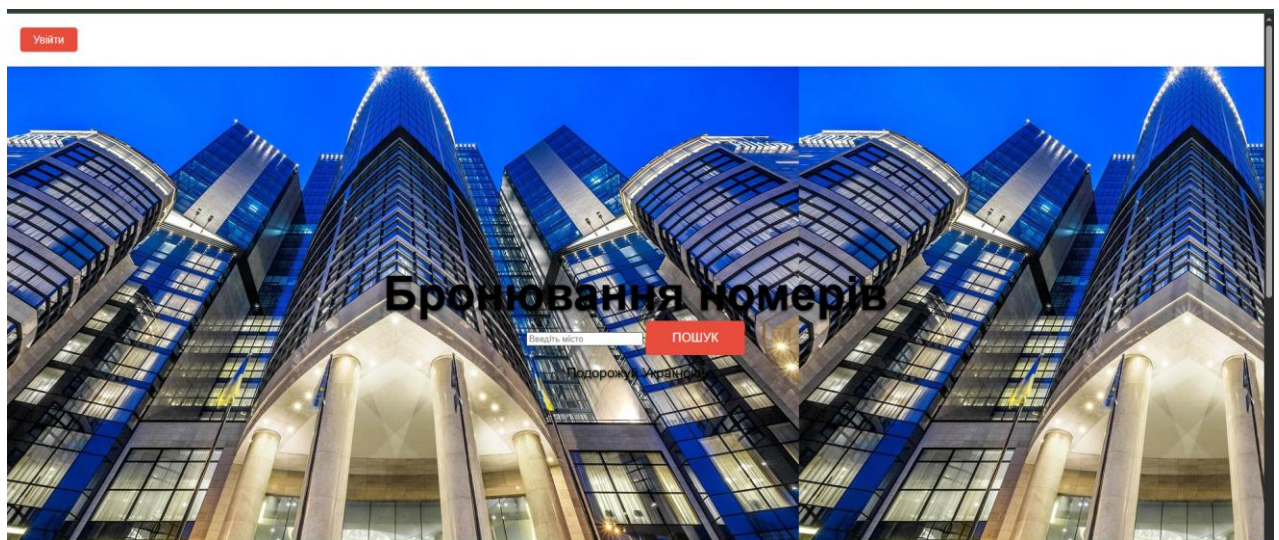


Рисунок 4.1 – Вигляд головної сторінки вебзастосунку

Натискаючи на кнопку входу, користувач бачить перед собою форму авторизації, або, якщо він ще не має власного профілю на вебзастосунку, є можливість зареєструватись. Вигляд форми авторизації наведено на рисунку 4.2.

Увійти до акаунту

Username:

Password:

[Увійти](#)

Не маєте акаунту? [Зареєструйтесь](#)

Рисунок 4.2 – Форма авторизації

Форму реєстрації, у випадку відсутності в користувача свого профілю, наведена на рисунку 4.3.

Реєстрація

Ім'я користувача:

Required. 150 characters or fewer. Letters, digits and @./+/-/_ only.

Ім'я:

Прізвище:

Електронна пошта:

Пароль:

Підтвердження паролю:

[Зареєструватися](#)

У вас вже є акаунт? [Увійти](#)

Рисунок 4.3 – Форма реєстрації

Увійшовши до свого профілю, користувач бачить всі свої дані, а також кнопки для зміни особистих та платіжних даних, виходу з профілю. Також наявна панель для зворотнього зв'язку з адміністрацією, на якій також зображені минулі звернення користувача. Якщо користувач має статус власника готелю, то в його профілі буде кнопка “Перегляд готелю”, де він зможе змінювати всю інформацію про готель, а також найголовніше – дати, за якими перевіряється доступність пропозицій для бронювання. Першу частину сторінки профілю зображено на рисунку 4.4.

Рисунок 4.4 – Сторінка профілю, перша частина

Нижче, під формою з даними профіля, наведено історію бронювань готельних номерів користувача з можливістю залишити відгук до тих резервувань, статус яких підтверджений і час бронювання вже вичерпано. Це зроблено для наявності на сторінках тільки правдивих відгуків, від людей, які дійсно вже проживали в цих номерах. Друга частина сторінки профілю зображена на рисунку 4.5.

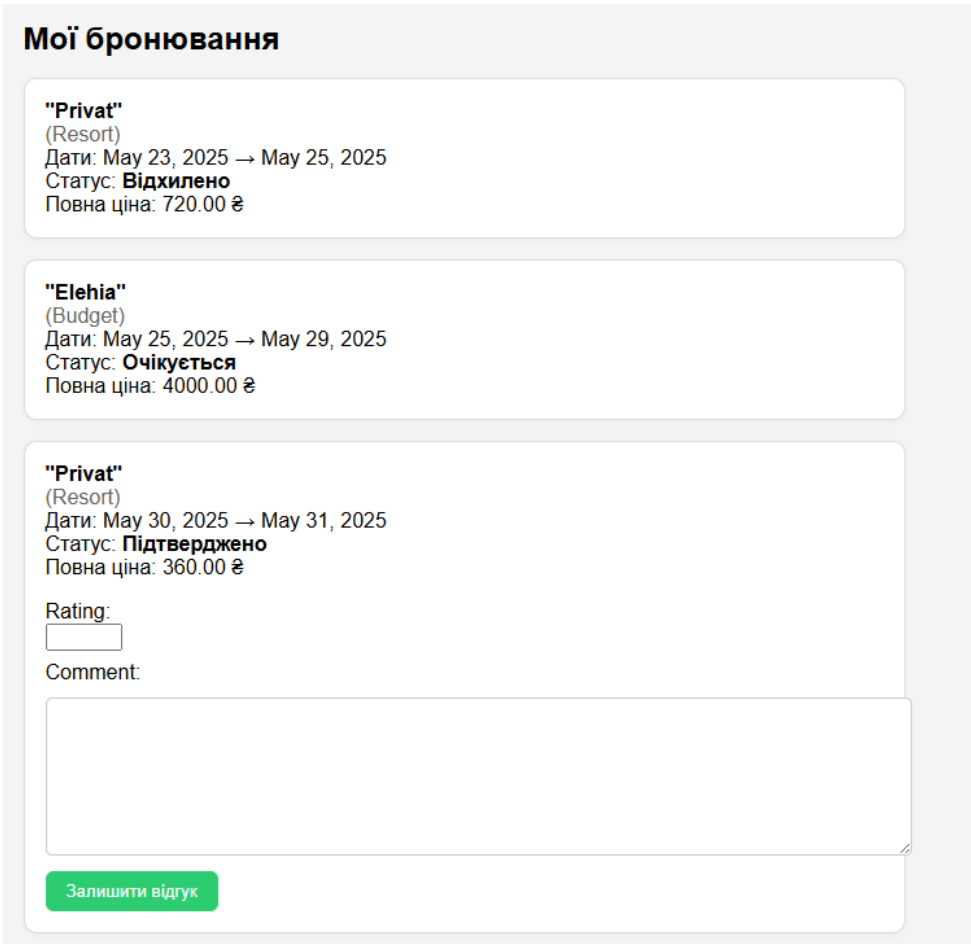


Рисунок 4.5 – Сторінка профілю, друга частина

Увівши на головній сторінці в пошуковий рядок назву міста або готелю, відкриється сторінка пропозицій за зазначеними фільтрами. Інші можливі фільтри будуть доступні на бічній панелі, а для відкриття повної інформації про будь-яку з пропозицій необхідно натиснути на будь-яке місце картки пропозиції. Сторінка пошуку зображена на рисунку 4.6.

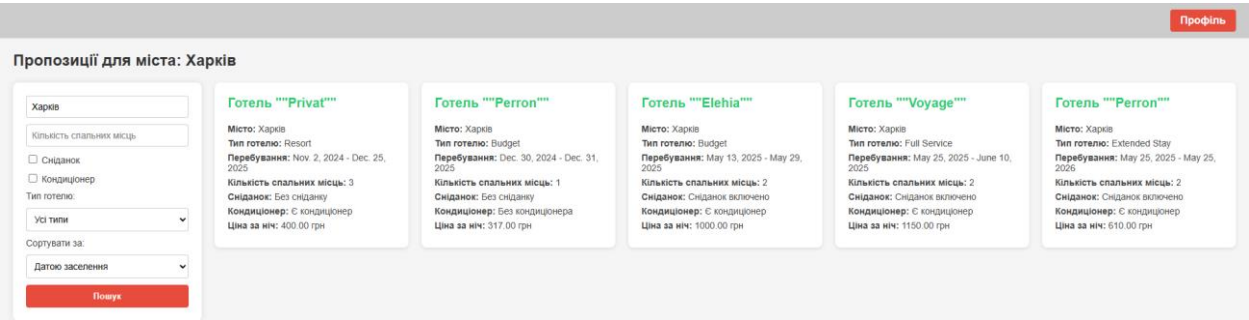


Рисунок 4.6 – Сторінка пошуку пропозицій

Натиснувши на картку пропозиції зі сторінки пошуку, відкривається повна інформація про варіант готельного номеру, а також є можливість забронювати пропозицію, натиснувши відповідну кнопку. Сторінка повної інформації про пропозицію наведена на рисунку 4.7.

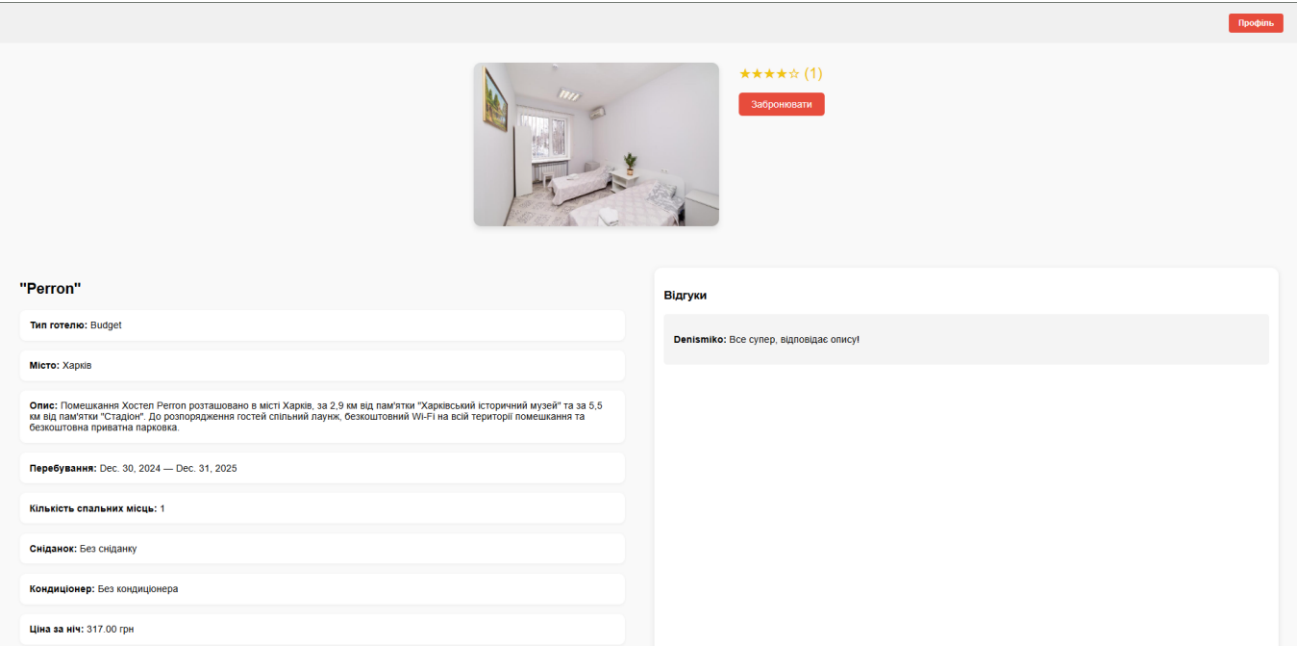


Рисунок 4.7 – Сторінка з повною інформацією про готельний номер

Натиснувши на кнопку бронювання зі сторінки детальної інформації про пропозицію, відкривається форма для бронювання, вигляд якої зображено на рисунку 4.8.

Бронювання: "Privat"

- Тип готелю: Resort
- Місто: Харків
- Сніданок включено: Ні
- Кондиціонер: Так

Date from

2025-06-02

Date to

2025-06-05

Package name

Кількість ночей

This field is required

Назва пропозиції: "Privat"

Номер пропозиції: 1

Ціна за ніч: 400.00 грн

Забронювати

Рисунок 4.8 – Вигляд форми для бронювання

Увівши всі необхідні дані для бронювання, користувача переводить на сторінку оплати номеру завдяки платіжному сервісу Stripe, де він має ввести платіжні дані. Вигляд форми оплати бронювання готельного номеру наведено на рисунку 4.9.

← TEST MODE

Бронювання: "Privat"

UAH 1,920.00

Оплата с **link**

Or

Email

newmiko778@gmail.com

Card information

1234 1234 1234 1234

MM / YY

CVC

Cardholder name

Full name on card

Country or region

Ukraine

☐ Securely save my information for 1-click checkout
Pay faster on this site and everywhere Link is accepted.

Pay

Powered by stripe

Terms Privacy

Рисунок 4.9 – Оплата готельного номеру через Stripe

Здійснивши оплату готельного номеру, користувача переводить на сторінку з оголошенням результату, вигляд сторінки інформування про оплату бронювання наведено на рисунку 4.10. Варто зазначити, що ніяких коштів при здійсненні оплати готельного номеру не списується, адже використовуються тестові картки від Stripe для перевірки роботи системи.

Home Logout

Оплата пройшла успішно!

Дякуємо за ваше бронювання. Підтвердження надіслано вам на пошту.

Перейти в профіль

Оплата пройшла успішно!

Рисунок 4.10 – Успішність оплати готельного номеру

Оформивши заявку на бронювання та здійснивши оплату через Stripe, після натискання кнопки переходу до профілю на сторінці з інформуванням про оплату, користувач переходить до сторінки профілю, де він бачить статус свого бронювання та інформаційне повідомлення про успішність оформлення заявки. Також користувачеві в профіль приходять сповіщення про успішну зміну даних, підтвердження і відхилення бронювань. Загальний вигляд повідомлень вебзастосунку наведено на рисунку 4.11.

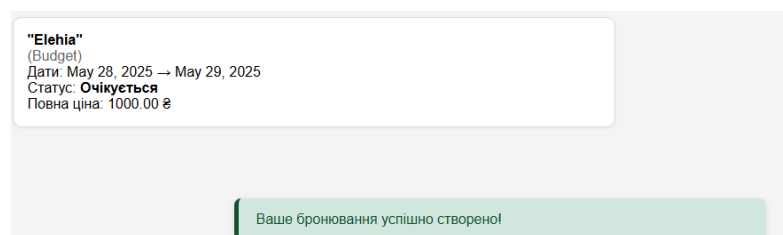


Рисунок 4.11 – Загальний вигляд повідомлень у вебзастосунку

Робота адміністратора здійснюється завдяки адмін-панелі від Django, функціонал якої було розширено для надання можливості додавати пропозиції, видаляти і змінювати існуючі варіанти для бронювання, а також блокувати користувачів і відповідати на їх питання. Приклад роботи адмін-панелі для зміни існуючої пропозиції готельного номеру наведено на рисунку 4.12.

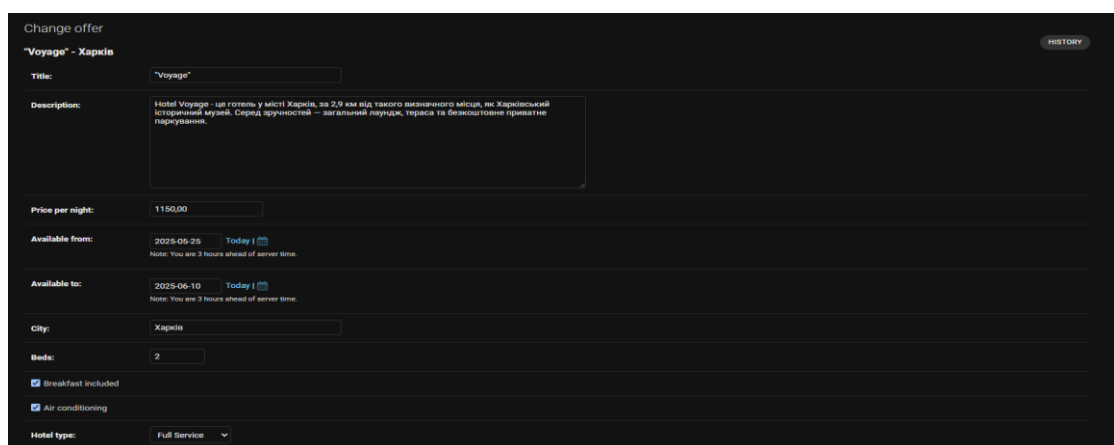


Рисунок 4.12 – Зміна існуючої пропозиції через адмін-панель

4.3 Тестування роботи вебзастосунку для бронювання готельних номерів

Для проведення функціонального тестування вебзастосунку для бронювання готельних номерів було створено чек-лист, вигляд якого наведено в таблиці 4.1.

Таблиця 4.1 – Чек-лист функціонального тестування

Назва функції	Google Chrome 136.0.7103.114	Microsoft Edge 136.0.3240.76	Mozilla Firefox 138.0.4
Перевірити реєстрацію у вебзастосунку за умови заповнення валідними даними	Passed	Passed	Passed
Перевірити блокування процесу бронювання готельного номеру без платіжних даних	Passed	Passed	Passed
Перевірити роботу функції написання відгуку тільки для пропозицій зі статусом “Підтверджено”	Passed	Passed	Passed
Перевірити роботу кнопок зміни особистих даних в профілі користувача	Passed	Passed	Passed
Перевірити роботу пошуку пропозицій за назвою міста	Passed	Passed	Passed
Перевірити роботу фільтрів	Passed	Passed	Passed

Для виконання функціонального тестування було проведено підготовку: налаштування середовищ – трьох різних браузерів, а також написання автоматизованих тестів на Pytest-Django [21] для здійснення простих процесів, зокрема, для авторизації, реєстрації та перевірки роботи процесу бронювання. Проведене функціональне тестування допомогло з'ясувати, що всі функціональні вимоги до вебзастосунку було реалізовано повністю, а їх робота є передбачуваною та коректною.

Чек-лист тестування зручності використання вебзастосунку для бронювання готельних номерів наведено в таблиці 4.2.

Таблиця 4.2 – Чек-лист тестування зручності

Назва параметру	Google Chrome 136.0.7103.114	Microsoft Edge 136.0.3240.76	Mozilla Firefox 138.0.4
Сайт швидко завантажується (не більше 3 сек)	Passed	Passed	Passed
Кожна дія користувача супроводжується повідомленнями	Passed	Passed	Passed
Кнопки та поля форм мають зрозумілі назви	Passed	Passed	Passed
Інформація на сторінках читається легко	Passed	Passed	Passed
Навігація в профілі користувача є інтуїтивно зрозумілою	Passed	Passed	Passed
Одноманітність інтерфейсу	Passed	Passed	Passed

Для виконання тестування зручності використання вебзастосунку для бронювання готельних номерів нічого не треба було налаштовувати, адже всі браузери вже готові до роботи після виконання функціонального тестування. Usability-тестування було виконано повністю вручну, адже вебзастосунком мають користуватись люди, поведінка яких є непередбачуваною ніяким скриптом. У випадку тестування на зручність використання, суб'єктивний досвід кожного користувача є набагато ціннішим, ніж результати роботи написаних автоматичних тестів.

Чек-лист тестування безпеки вебзастосунку для бронювання готельних номерів наведено в таблиці 4.3.

Таблиця 4.3 – Чек-лист тестування безпеки

Назва параметру	Google Chrome 136.0.7103.114	Microsoft Edge 136.0.3240.76	Mozilla Firefox 138.0.4
Перевірити блокування доступу до даних профілю без авторизації	Passed	Passed	Passed
Перевірити зберігання паролів у вигляді хешів	Passed	Passed	Passed
Перевірити вихід з профілю користувача при завершенні сесії	Passed	Passed	Passed
Перевірити приховування паролю користувача при його введенні у формах	Passed	Passed	Passed
Перевірити, що користувач не може змінювати бронювання інших	Passed	Passed	Passed

Для виконання кожного з етапів чек-листа тестування безпеки було написано окремий тест на Pytest-Django [21], що дозволив дійти таких висновків – розроблений вебзастосунок для бронювання готельних номерів повністю відповідає базовим вимогам до безпеки вебзастосунків. Зокрема, у вебзастосунку є захист від вкрадення паролів, даних бронювань та платіжної інформації кожного з користувачів.

4.4 Висновки за розділом 4

Розроблений вебзастосунок для бронювання готельних номерів був детально описаний : визначено його важливість та спосіб застосування, апаратні та програмні характеристики що необхідні для коректної роботи. Було показано покроковий алгоритм роботи з вебзастосунком, кожна з дій підкріплювалася відповідним описом. Проведення функціонального тестування, тестувань зручності використання та безпеки довело, що розроблений вебзастосунок повністю відповідає створеним раніше вимогам, успішно реалізує процес бронювання готельного номеру онлайн.

ВИСНОВКИ

В ході виконання роботи було розроблено вебзастосунок “Бронювання готельних номерів”, який дозволяє швидко здійснювати процес оформлення заявок на резервування готельних номерів в режимі онлайн завдяки надаванню адміністраторам вебзастосунку повноважень власників готелей.

Результатом детального аналізу специфіки проблеми та існуючих рішень стали сформульовані вимоги до функціоналу вебзастосунку. Провівши дослідження різних матеріалів та методів, що можуть допомогти досягти цілі у вигляді реалізації всіх визначених вимог, було прийнято рішення створення вебзастосунку на мові програмування Python з використанням фреймворку Django на основі бази даних SQLite в середовищі розробки Visual Studio Code , що забезпечило прискорення процесу розробки для подальшого початку тестування.

Створений вебзастосунок для бронювання готельних номерів дозволяє переглядати пропозиції готельних номерів що доступні для резервування, оформлювати заявки та здійснювати оплату бронювань через платіжний сервіс Stripe, писати відгуки та ставити оцінки завдяки реалізації історії бронювань в профілі користувача. В свою чергу, адміністратори завдяки створеній панелі можуть модерувати користувачами, змінювати дані у варіантах для резервування тимчасового житла, видаляти їх та додавати нові, а також спілкуватися з користувачами через окрему форму для звернень.

Зробивши тестування зручності використання, безпеки та функціоналу, було зроблено висновок, що робота з розробленим вебзастосунком для бронювання готельних номерів є інтуїтивно зрозумілою, безпечною та передбачуваною, адже всі функції працюють коректно.

Практична цінність розробленого вебзастосунку “Бронювання готельних номерів” полягає у методі делегування повноважень :

адміністратори беруть на себе функціонал власників закладів тимчасового проживання. Це дозволяє одразу створювати правильні та чітко сформульовані пропозиції для резервування житла на основі даних від власників готелів про готель та часові рамки доступності кожної з його пропозицій, відповідно, підвищуючи безпеку процесу бронювань та зменшуючи час, необхідний на модерацію пропозицій готельних номерів, адже адміністратори завдяки розробленій панелі одразу створюють вірні варіанти житла без необхідності внесення правок. Також такий підхід значно підвищує довіру клієнтів до вебзастосунку для бронювання готельних номерів, адже вони впевнені, що їх заявка створена на пропозицію, яка майже не потребує додаткових уточнень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. 8 Types of Hotels: A Detailed Guide [Electronic resource]. – Access mode: <https://foodnhotelasia.com/glossary/horeca/types-of-hotels/>.
2. Hotel | History, Types, Characteristics, & Facts [Electronic resource]. – Access mode: <https://www.britannica.com/topic/hotel>.
3. Hotel automation: Implementation guide for the future [Electronic resource]. – Access mode: <https://www.siteminder.com/r/hotel-automation/#:~:text=Hotel%20automation%20is%20the%20use,guests%20through%20automated%20messaging%20platforms>.
4. Why You Need an Online Booking Hotel System Immediately [Electronic resource]. – Access mode: [https://www.newbook.cloud/online-booking-hotel-system/#:~:text=An%20online%20hotel%20booking%20management%20system%20enables%20guests%20to%20book,different%20from%20your%20hotel's%20l](https://www.newbook.cloud/online-booking-hotel-system/#:~:text=An%20online%20hotel%20booking%20management%20system%20enables%20guests%20to%20book,different%20from%20your%20hotel's%20location)ocation.
5. 5 Reasons to Use Online Booking System for Your Hotel Reservation [Electronic resource]. – Access mode: <https://queueme.io/blog/5-Reasons-to-Use-Online-Booking-System-for-Your-Hotel-Reservation.html>.
6. Booking.com - Готелі | Booking.com [Електрон. ресурс]. – Режим доступу: https://www.booking.com/index.uk.html?lang=uk&soz=1&lang_changed=1.
7. Топ-7 законів у UX-дизайні [Електрон. ресурс]. – Режим доступу : <https://cases.media/article/top-7-zakoniv-u-ux-dizaini?srsltid=AfmBOoqfm3YF3YbxZDA-Q7KnShpS4VloFsrYmqWFW66yxhkmAkm7soKi>.
8. Hotels24.ua [Електрон. ресурс]. – Режим доступу : <https://hotels24.ua/uk/>.
9. PHP : Language Reference - Manual [Electronic resource]. – Access mode: <https://www.php.net/manual/en/langref.php>.

10. Python Reference [Electronic resource]. – Access mode: https://www.w3schools.com/python/python_reference.asp.
11. Node.js - Introduction to Node.js [Electronic resource]. – Access mode: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>.
12. Quick start guide | PyCharm documentation [Electronic resource]. – Access mode: <https://www.jetbrains.com/help/pycharm/quick-start-guide.html>.
13. Tutorial: Get Started with Visual Studio Code [Electronic resource]. – Access mode: <https://code.visualstudio.com/docs/getstarted/getting-started>.
14. Introduction | Sublime Text Community Documentation [Electronic resource]. – Access mode: <https://docs.sublimetext.io/guide/>.
15. Django documentation [Electronic resource]. – Access mode: <https://docs.djangoproject.com/en/5.2/>.
16. Tornado Web Server - Tornado 6.5 documentation [Electronic resource]. – Access mode: <https://www.tornadoweb.org/en/stable/>.
17. Welcome to Flask - Flask Documentation (3.1 x) [Electronic resource]. – Access mode: <https://flask.palletsprojects.com/en/stable/>.
18. PostgreSQL 16.9 Documentation [Electronic resource]. – Access mode: <https://www.postgresql.org/files/documentation/pdf/16/postgresql-16-A4.pdf>.
19. MySQL: Understanding What It Is and How It's Used [Electronic resource]. – Access mode: <https://www.oracle.com/middleeast/mysql/what-is-mysql/>.
20. About SQLite [Electronic resource]. – Access mode: <https://www.sqlite.org/about.html>.
21. Pytest-Django Documentation [Electronic resource]. – Access mode: <https://pytest-django.readthedocs.io/en/latest/>.
22. Stripe Documentation [Electronic resource]. – Access mode: <https://docs.stripe.com/>.

ДОДАТОК А
Технічне завдання

Вступ

Розробка вебзастосунку “Бронювання готельних номерів” потребує аналізу проблем існуючих систем. Попри велику кількість замовлень, ці сервіси не завжди забезпечують надійність бронювань через сумнівну якість оголошень та обмежені можливості модерації.

Мета проєкту – спростити процес резервування та підвищити його безпеку завдяки делегуванню повноважень: адміністратор керує створенням пропозицій і обробкою заявок через спеціальну панель, а власники готелів лише надають інформацію про свої заклади.

A.1 Підстави для розробки

Вебзастосунок “Бронювання готельних номерів” розробляється на підставі завдання на дипломну кваліфікаційну роботу бакалавра, затверджене наказом № 209 Національного Університету “Запорізька Політехніка” від 28 квітня 2025 року.

A.2 Призначення розробки

Вебзастосунок “Бронювання готельних номерів” має бути створений для здійснення прискореного процесу пошуку та бронювання житла в закладах тимчасового проживання з використанням методу делегування повноважень.

А.3 Основні вимоги до програми

А.3.1 Вимоги до функціональних характеристик

Для успішного досягнення мети розробки вебзастосунку для бронювання готельних номерів, необхідно забезпечити відповідність таким функціональним вимогам:

- забезпечення безпечного проведення процесу реєстрації та авторизації користувачів і адміністраторів;
- забезпечення обов'язковості використання профілю користувача, в якому є необхідний функціонал для зміни персональних даних, а також форма зв'язку з адміністрацією вебзастосунку та форма для написання відгуків до раніше заброньованих пропозицій, що мають підтверджений статус;
- здійснення процесу знаходження пропозицій для бронювання готельних номерів завдяки наявності пошукового рядка на головній сторінці вебзастосунку;
- виконання розширеного пошуку завдяки бічній панелі фільтрів;
- відтворення процесу бронювання за допомогою реалізації форми для резервування готельних номерів, а також проведення оплати за обране готельне житло завдяки сторонньому сервісу для проведення оплат;
- створення адміністративної панелі для надання можливості адміністраторам додавати пропозиції бронювання готельних номерів, видаляти та змінювати існуючі варіанти житла, блокувати користувачів та видаляти відгуки, формувати відповіді на запитання користувачів.

А.3.2 Умови експлуатації

Для коректної роботи вебзастосунку “Бронювання готельних номерів” необхідно забезпечити виконання таких умов експлуатації : завантаження і налаштування Django-застосунку для можливості забезпечення необхідного

вигляду бази даних за рахунок використання міграцій, завантажена база даних SQLite, а також інстальований модуль Decimal з мови програмування Python.

A.3.3 Вимоги до надійності

Вебзастосунок для бронювання готельних номерів має забезпечити безпеку даних завдяки використанню методів хешування паролів, захисту особистої інформації в профілі.

A.3.4 Вимоги до інтерфейсу

Вигляд вебзастосунку повинен бути інтуїтивно зрозумілим та чітким, повинен не викликати зайвих питань та виконувати першочергові функції завдяки швидкому доступу до кнопок і інструментів навігації.

A.3.5 Вимоги до складу та параметрів технічних засобів

Для коректної роботи вебзастосунку “Бронювання готельних номерів” необхідно таке апаратне забезпечення :

- не менше 2 гб оперативної пам’яті, а також 5 гб вільного простору на накопичувачі будь-якого типу;
- процесор з двома ядрами, тактова частота яких є не меншою за 1.5 ГГц.

A.4 Вхідні дані до роботи

Вхідними даними до вебзастосунку “Бронювання готельних номерів” є особиста інформація при реєстрації, вихідними – дані про створені бронювання, відгуки та оцінки.

ДОДАТОК Б**Код програми**

```

from django.shortcuts import render, get_object_or_404, redirect
from django.contrib.auth.decorators import login_required
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.forms import AuthenticationForm
from django.contrib.auth.models import User
from django.contrib import messages
from .forms import InquiryForm
from django.db.models import Q
from django.http import HttpResponseRedirect
from django.urls import reverse
from decimal import Decimal
from .models import Offer, BookingRequest, Review, Profile, Inquiry
from .forms import ReviewForm
from .forms import (
    EditProfileForm,
    RegistrationForm,
    PaymentDetailsForm,
    BookingForm,
    ReviewForm
)

def logout_view(request):
    logout(request)
    messages.success(request, 'Ви успішно вийшли із системи!')
    return redirect('home')

@login_required

def profile_view(request):
    user = request.user

```

```

bookings = user.bookingrequest_set.select_related('offer')

review_forms = {}
inquiry_form = InquiryForm()
inquiries = Inquiry.objects.filter(user=user).order_by('-created_at')

if request.method == 'POST':
    if 'inquiry_submit' in request.POST:
        inquiry_form = InquiryForm(request.POST)
        if inquiry_form.is_valid():
            inquiry = inquiry_form.save(commit=False)
            inquiry.user = user
            inquiry.save()
            messages.success(request, 'Ваше звернення надіслано
адміністрації!')
            return redirect('profile')
        else:
            booking_id = request.POST.get('booking_id')
            booking = get_object_or_404(BookingRequest, id=booking_id,
user=user)

            if booking.status == 'approved' and not
Review.objects.filter(offer=booking.offer, user=user).exists():
                form = ReviewForm(request.POST)
                if form.is_valid():
                    review = form.save(commit=False)
                    review.user = user
                    review.booking = booking
                    review.offer = booking.offer
                    review.save()

```

```

        messages.success(request, 'Відгук успішно додано!')
        return redirect('profile')
    else:
        review_forms[booking.id] = form

for booking in bookings:
    if booking.status == 'approved' and not
Review.objects.filter(offer=booking.offer, user=user).exists():
    if booking.id not in review_forms:
        review_forms[booking.id] = ReviewForm()

return render(request, 'profile.html', {
    'bookings': bookings,
    'review_forms': review_forms,
    'inquiry_form': inquiry_form,
    'inquiries': inquiries,
})

def edit_profile_view(request):
    profile, created = Profile.objects.get_or_create(user=request.user)

    if request.method == 'POST':
        form = EditProfileForm(request.POST, instance=request.user)
        if form.is_valid():
            form.save()
            profile.country = request.POST.get('country')
            profile.card_number_last4 =
request.POST.get('card_number_last4')
            profile.save()

```

```

        messages.success(request, 'Профіль успішно оновлено!')
        return redirect('profile')
    else:
        form = EditProfileForm(instance=request.user)

    return render(request, 'edit_profile.html', {'form': form, 'profile': profile})

@login_required
def payment_details_view(request):
    profile, created = Profile.objects.get_or_create(user=request.user)

    if request.method == 'POST':
        form = PaymentDetailsForm(request.POST)
        if form.is_valid():
            card_number = form.cleaned_data['card_number']
            profile.card_number_last4 = card_number[-4:]
            profile.save()
            messages.success(request, 'Платіжні дані оновлено!')
            return redirect('profile')
        else:
            form = PaymentDetailsForm()

    return render(request, 'payment_details.html', {'form': form})

def login_view(request):
    if request.method == 'POST':
        form = AuthenticationForm(request, data=request.POST)
        if form.is_valid():
            username = form.cleaned_data['username']
            password = form.cleaned_data['password']

```

```

        user = authenticate(request, username=username,
password=password)
        if user is not None:
            login(request, user)
            messages.success(request, 'Ви успішно увійшли!')
            return redirect('home')
        else:
            messages.error(request, 'Невірне ім'я користувача або пароль')
    else:
        messages.error(request, 'Невірне ім'я користувача або пароль')
    else:
        form = AuthenticationForm()
    return render(request, 'login.html', {'form': form})

def register_view(request):
    if request.method == 'POST':
        form = RegistrationForm(request.POST)
        if form.is_valid():
            user = form.save(commit=False)
            user.set_password(form.cleaned_data['password'])
            user.save()
            messages.success(request, 'Реєстрація успішна! Увійдіть у
систему.')
            return redirect('login')
        else:
            messages.error(request, 'Помилка реєстрації. Спробуйте ще раз.')
    else:
        form = RegistrationForm()
    return render(request, 'register.html', {'form': form})

```

```
def home(request):
    return render(request, 'home.html')

def search_results(request):
    city = request.GET.get('city', '')
    beds = request.GET.get('beds')
    breakfast_included = request.GET.get('breakfast_included')
    air_conditioning = request.GET.get('air_conditioning')
    sort_by_date = request.GET.get('sort_by_date', 'available_from')
    hotel_type = request.GET.get('hotel_type')

    offers = Offer.objects.all()

    if city:
        offers = offers.filter(city__icontains=city)

    if beds:
        offers = offers.filter(beds=beds)

    if hotel_type:
        offers = offers.filter(hotel_type=hotel_type)

    budget_offers = offers.filter(hotel_type='budget')
    non_budget_offers = offers.exclude(hotel_type='budget')

    if breakfast_included:
        non_budget_offers =
non_budget_offers.filter(breakfast_included=True)

    if air_conditioning:
```

```

        non_budget_offers = non_budget_offers.filter(air_conditioning=True)

    offers = (budget_offers | non_budget_offers).order_by(sort_by_date)

    return render(request, 'search_results.html', {'offers': offers})

def offer_detail(request, offer_id):
    offer = get_object_or_404(Offer, id=offer_id)
    return render(request, 'offer_detail.html', {'offer': offer})

@login_required
def book(request, offer_id):
    offer = get_object_or_404(Offer, id=offer_id)

    if request.method == 'POST':
        form = BookingForm(request.POST, offer=offer)
        if form.is_valid():
            booking = form.save(commit=False)
            booking.user = request.user
            booking.offer = offer

            nights = form.cleaned_data['nights']
            date_from = form.cleaned_data['date_from']
            date_to = form.cleaned_data['date_to']

            if offer.hotel_type == 'resort':
                price_per_night = offer.dynamic_price(date_from)
            else:

```

```

price_per_night = offer.price_per_night

booking.total_price = price_per_night * Decimal(nights)

if offer.hotel_type == 'full_service':
    booking.status = 'requires_confirmation'
    booking.save()
    messages.success(request, 'Бронювання надіслано. Очікується
подвійне підтвердження.')

elif offer.hotel_type == 'boutique':
    booking.package_name =
form.cleaned_data.get('package_name') or f'Подія-{offer.title}'
    booking.status = 'pending'
    booking.save()
    messages.success(request, 'Ваш пакет для бутік-готелю
збережено.')

elif offer.hotel_type == 'extended_stay':
    if (date_to - date_from).days < 7:
        messages.error(request, 'Extended stay: мінімальний термін
бронювання - 7 днів.')
        return redirect('offer_detail', offer_id=offer.id)
    booking.status = 'pending'
    booking.save()
    messages.success(request, 'Бронювання для extended stay
створено.')

```

```

else:

    booking.status = 'pending'
    booking.save()
    messages.success(request, 'Ваше бронювання успішно
створено!')

    return redirect('profile')
else:
    form = BookingForm(offer=offer)

preview_price = None
if form.is_bound and form.is_valid():
    nights = form.cleaned_data['nights']
    date_from = form.cleaned_data['date_from']
    if offer.hotel_type == 'resort':
        price_per_night = offer.dynamic_price(date_from)
    else:
        price_per_night = offer.price_per_night
    preview_price = price_per_night * Decimal(nights)

return render(request, 'book_offer.html', {
    'form': form,
    'offer': offer,
    'preview_price': preview_price,
})

def admin_booking_detail(request, booking_id):

```

```

booking = get_object_or_404(BookingRequest, id=booking_id)

if request.method == 'POST':
    form = BookingForm(request.POST, instance=booking,
offer=booking.offer)
    if form.is_valid():
        form.save()
        return
HttpResponseRedirect(reverse('admin:bookingrequest_changelist'))
else:
    form = BookingForm(instance=booking, offer=booking.offer)

return render(request, 'admin/booking_detail.html', {'form': form,
'booking': booking})

```

@login_required

```

def add_review(request):
    if request.method == 'POST':
        offer_id = request.POST.get('offer_id')
        rating = request.POST.get('rating')
        comment = request.POST.get('comment')
        offer = get_object_or_404(Offer, id=offer_id)

        Review.objects.create(
            user=request.user,
            offer=offer,
            rating=rating,
            comment=comment
        )
        messages.success(request, 'Ваш відгук успішно додано!')

```

```
        return redirect('offer_detail', offer_id=offer.id)
    return redirect('home')

def submit_review(request, booking_id):
    booking = get_object_or_404(BookingRequest, id=booking_id,
user=request.user)
    if booking.status != 'approved' or hasattr(booking, 'review'):
        return redirect('profile')

    if request.method == 'POST':
        rating = int(request.POST.get('rating'))
        comment = request.POST.get('comment')

        Review.objects.create(
            booking=booking,
            user=request.user,
            rating=rating,
            comment=comment
        )
    return redirect('profile')
```

ДОДАТОК В
Слайди презентації

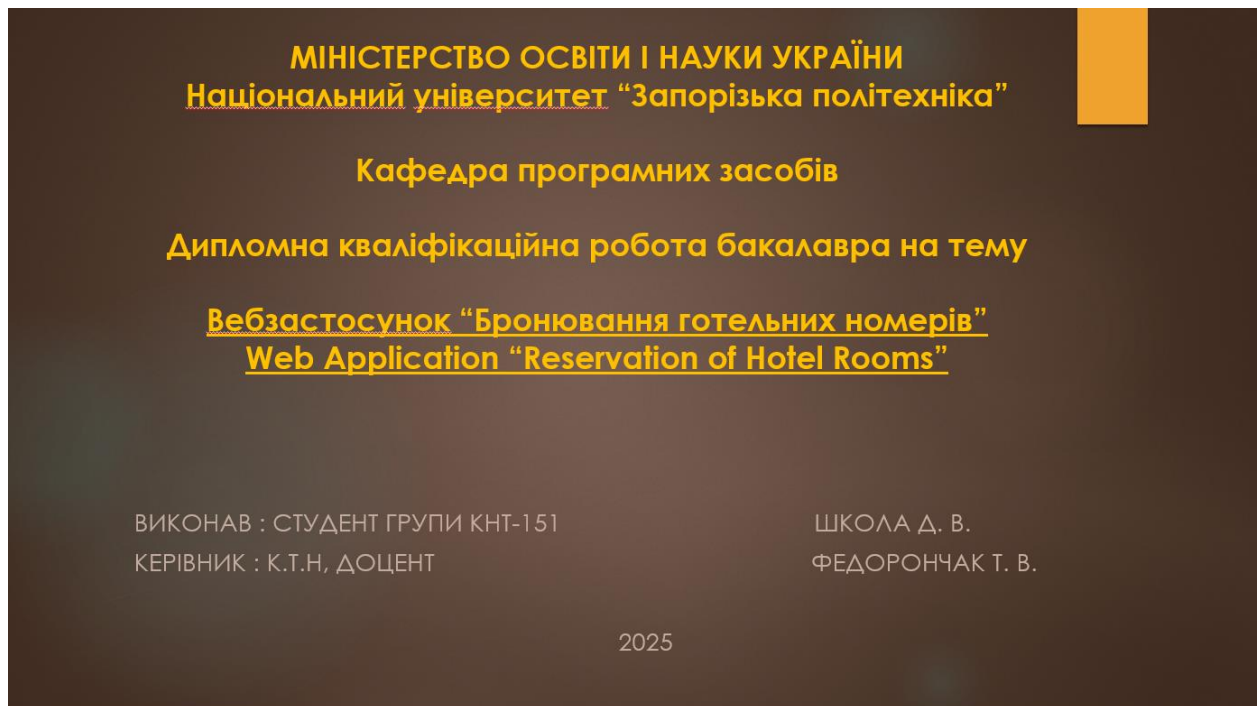


Рисунок В.1 – Титульний слайд

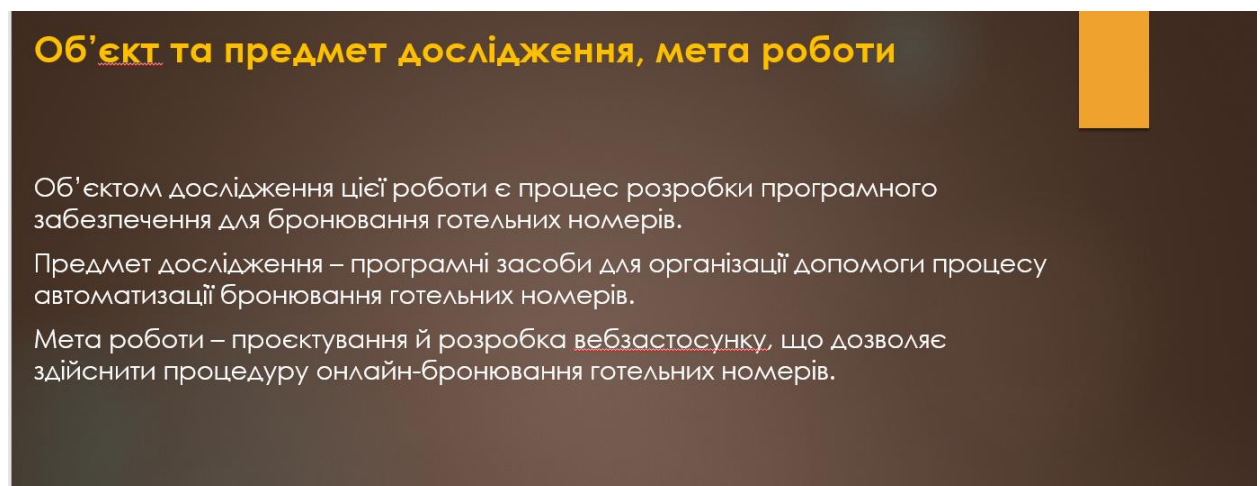


Рисунок В.2 – Мета роботи

Завдання, що необхідно розв'язати в роботі

- Виконати аналіз існуючих аналогів вебзастосунків для бронювання готельних номерів, розробити функціональні вимоги;
- Обрати матеріали та методи створення вебзастосунку, порівнявши різні можливі варіанти засобів;
- Виконати проектування вебзастосунку для бронювання готельних номерів;
- Реалізувати вебзастосунок для бронювання готельних номерів, виконати тестування.

Рисунок В.3 – Завдання

Порівняння аналогів вебзастосунків для бронювання готельних номерів

Критерій порівняння	Booking.com	Hotels24.ua	Розробка
Додавання пропозицій бронювання готельних номерів	+-	+-	+
Інформативність push-up сповіщень	+	-	+
Цілодобове оформлення бронювань	+	-	+
Вартість послуг вебзастосунку	Стагується комісія	Абсолютно безкоштовно	Абсолютно безкоштовно
Кількість фільтрів для пошуку	+	+	+
Різноманітність місць доступних для бронювання	+	-	-
Різноманітність мов та валют	+	-	-

Рисунок В.4 – Порівняння аналогів

Вибір мови програмування

Критерій для порівняння	PHP	Python	JavaScript
Синтаксис	+/-	+	-
Вимоги в споживанні ресурсів серверної частини	+	-	+/-
Підтримка фреймворків для створення вебзастосунків	+/-	+	+

Рисунок В.5 – Порівняння мов програмування

Вибір середовища розробки

Критерій для порівняння	PyCharm Community Edition	Visual Studio Code
Підтримка Python	+	+
Підтримка фреймворків для створення вебзастосунків	-	+
Підтримка інструментів для проведення тестування	+/-	+
Підтримка інструментів для роботи з базами даних	-	+

Рисунок В.6 – Порівняння середовищ розробки

Вибір фреймворку

Критерій порівняння	Django	Tornado	Flask
Підтримка Python	+	+	+
Продуктивність	+~	+	+
Швидкість розробки	+	-	+~
Підтримка пакетів розширення функціоналу фреймворку	+~	-	+

Рисунок В.7 – Порівняння фреймворків

Вибір бази даних

Критерій порівняння	PostgreSQL	MySQL	SQLite
Підтримка Django	+	+	+
Просте налаштування і встановлення	-	-	+
Споживання ресурсів серверної частини вебзастосунку	-	+~	+
Складність даних, що будуть зберігатися в базі	+	+~	-

Рисунок В.8 – Порівняння баз даних

Архітектура вебзастосунку на основі Django

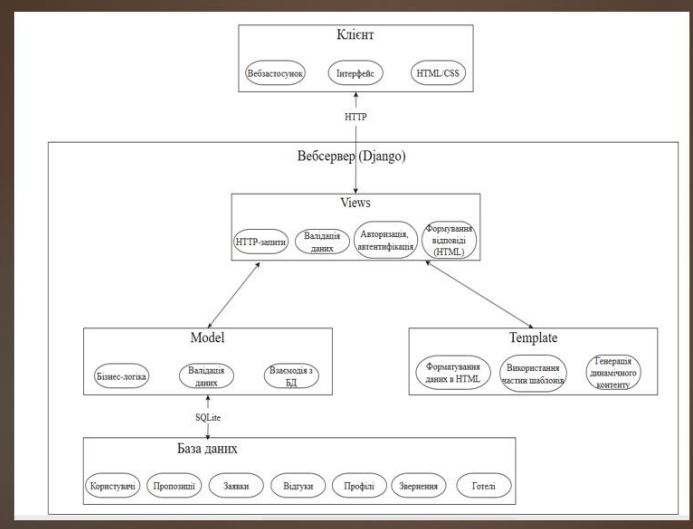


Рисунок В.9 – Архітектура вебзастосунку

Діаграми прецедентів користувача і адміністратора:

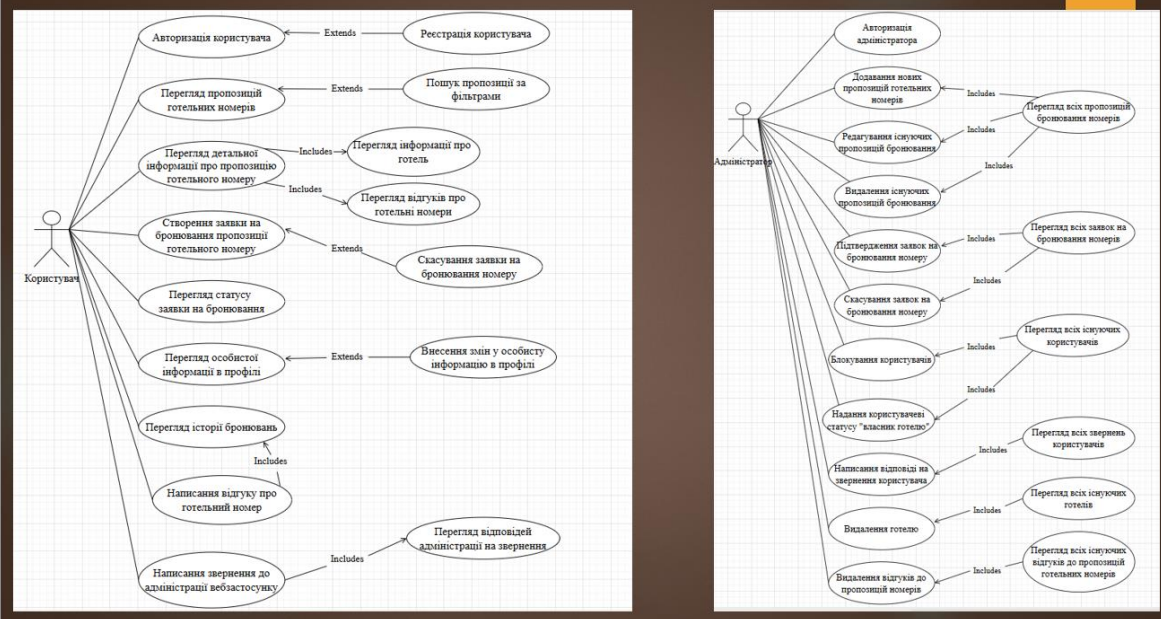
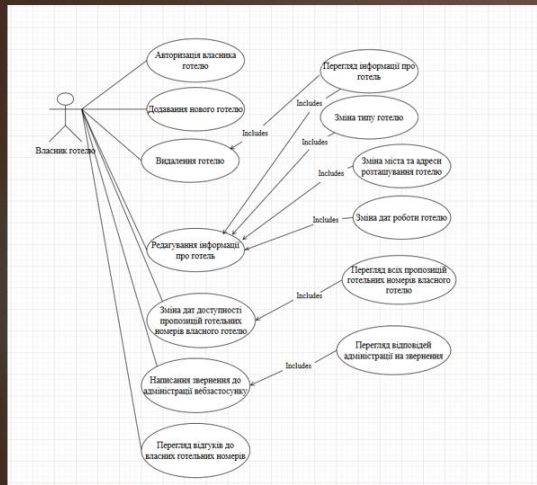


Рисунок В.10 – Діаграми прецедентів

Діаграма прецедентів власника готелю



Діаграма станів

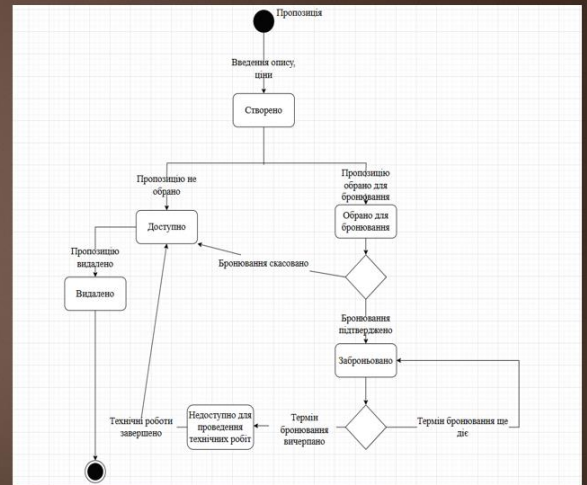


Рисунок В.11 – Інші корисні діаграми

Схема Баз Даних

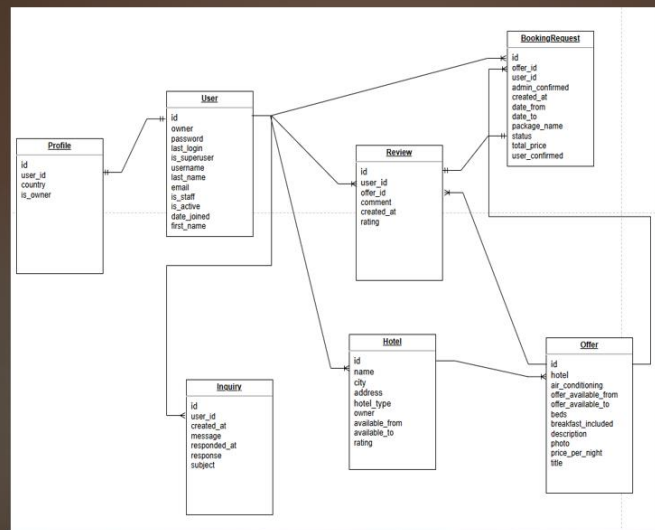


Рисунок В.12 – Схема бази даних

Сторінка профілю

[illegible]

Рисунок В.13 – Сторінка профілю

Повна інформація про готельний номер

[illegible]

Вигляд форми для бронювання

[illegible]

Оплата номеру через Stripe

UAH 1,920.00

Бронирование "Private"

1234 1234 1234 1234

MMJ / YY

CVC

1234 1234 1234 1234

1234 1234 1234 1234

Card information

Cardholder name

Full name on card

Country or region

Ukraine

☐ Securely save my information for 1-click checkout

Pay later on this site and everywhere Link is accepted.

Pay

Powered by  digiqa

[Terms](#) [Privacy](#)

Рисунок В.14 – Форми з інформацією про готелі та бронювання



Рисунок В.15 – Чек-листи тестувань

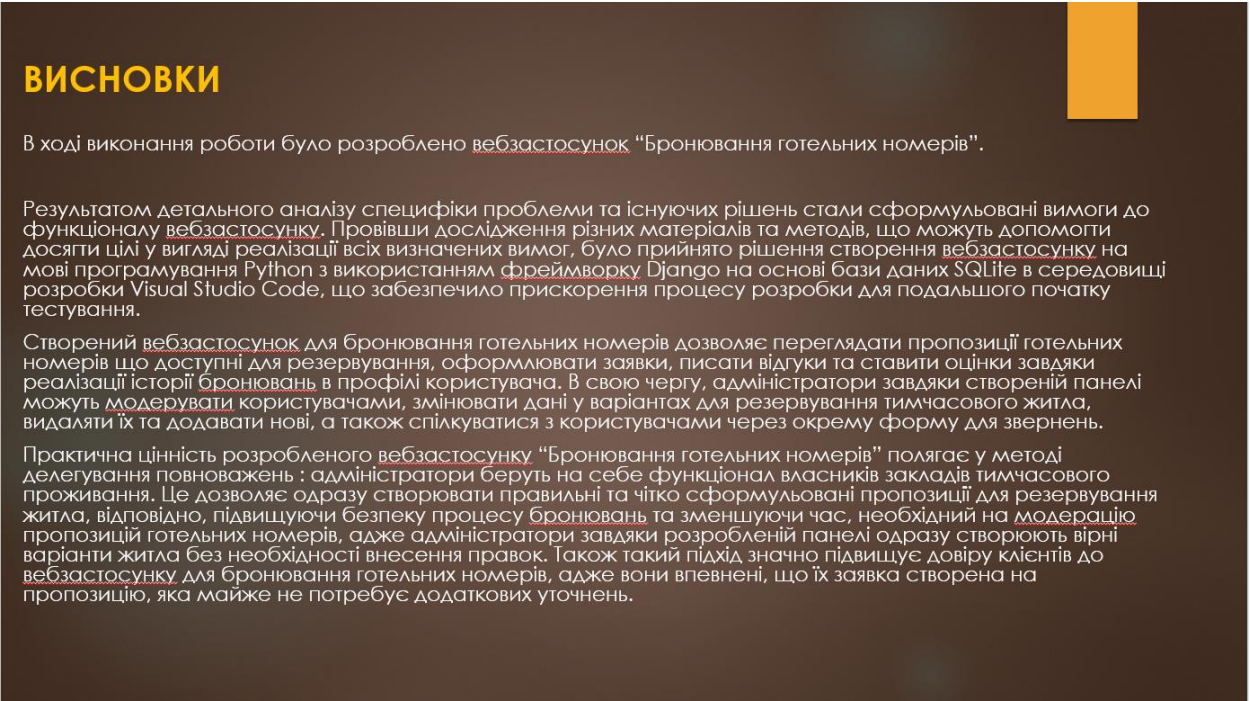


Рисунок В.16 – Висновки