

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА
ПРИРОДОКОРИСТУВАННЯ

«До захисту допущена»

Зав. кафедри прикладної математики

д.т.н. професор Турбал Ю.В.

«__»_____2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Проектування та розробка вебдодатку для оренди житла засобами
.NET»

Виконав: Ейсмонт Віктор Володимирович

студент навчально-наукового інституту кібернетики, інформаційних технологій
та інженерії

група КНз-51

Керівник: к.т.н., доцент Демчук Олена Станіславівна

Рівне-2024

Національний університет водного господарства та природокористування

НН інститут кібернетики, інформаційних технологій та інженерії

Кафедра комп'ютерних наук та прикладної математики

Освітньо-кваліфікаційний рівень **бакалавр**

Галузь знань 12 «Інформаційні технології»

Спеціальність 122 «Комп'ютерні науки»

Спеціалізація _____

ЗАТВЕРДЖУЮ

Завідувач кафедри

“ _____ ” _____ 20__ року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Ейсмонту Віктору Володимировичу

1. Тема роботи ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБДОДАТКУ ДЛЯ ОРЕНДИ ЖИТЛА ЗАСОБАМИ .NET керівник роботи доцент Демчук О.С. затверджені наказом вищого навчального закладу від «22» квітня 2024 року С №-525
2. Термін подання роботи студентом 19.06.2024
3. Вихідні дані до роботи технології, що необхідні для розробки сайту для оренди житла.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ. РОЗДІЛ 2. ПРОЄКТУВАННЯ САЙТУ. РОЗДІЛ 3. РОЗРОБКА САЙТУ.
5. Перелік графічного матеріалу (з точним зазначення обов'язкових креслень) мультимедійна презентація.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 1	Доцент Демчук О.С.	01.05.2024	
Розділ 2	Доцент Демчук О.С.	10.06.2024	
Розділ 3	Доцент Демчук О.С.	14.06.2024	

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вивчення літератури	01.11.2024 – 11.01.2024	
2	Формулювання задачі	10.03.2024 – 17.03.2024	
3	Вибір стеку технологій	18.03.2024 – 20.03.2024	
4	Аналіз даних	22.03.2024 – 28.03.2024	
5	Розробка структури сайту	28.03.2024 – 10.04.2024	
6	Розробка інтерфейсу	12.04.2024 – 10.05.2024	
7	Тестування програми	11.05.2024 – 15.05.2024	
8	Написання звіту	15.05.2024 – 30.05.2024	
9	Підготовка мультимедійної презентації	10.06.2024 – 15.06.2024	
10	Підготовка до виступу	15.06.2024 – 18.06.2024	

Студент

_____ Ейсмонт В.В.
 (підпис) (прізвище та ініціали)

Керівник роботи

_____ Демчук О.С.
 (підпис) (прізвище та ініціали)

ЗМІСТ

РЕФЕРАТ	5
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	6
ВСТУП.....	7
РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Застосування сучасних інформаційних технологій у сфері оренди житла та бронювання рейсів.....	10
1.2 Огляд аналогічних застосунків	12
1.2.1 Airbnb.....	14
1.2.2 Booking	15
1.2.3 Agoda	16
1.3 Порівняльний аналіз сайтів-аналогів	17
1.4 Постановка задачі.....	21
1.5 Висновки першого розділу.....	22
РОЗДІЛ 2 ПРОЄКТУВАННЯ САЙТУ	24
2.1 Вибір стеку технологій	24
2.1.1 ASP.NET MVC.....	24
2.1.2 Entity Framework.....	26
2.1.3 MS SQL.....	28
2.2 Опис функціоналу системи	29
2.2.1 Функціонал неавторизованого користувача.....	31
2.2.2 Функціонал для авторизованого користувача.....	31
2.2.3 Функціонал для модератора.....	32
2.3 Розробка вимог за допомогою діаграми прецедентів.....	32
2.4 Проектування архітектури сайту	36
2.5 Архітектура бекенд частини	37
2.6 Висновки до другого розділу	40
РОЗДІЛ 3 РОЗРОБКА ДОДАТКУ	41
3.1 Розробка структури бази даних	41
3.2 Програмна реалізація додатку	43
3.3 Інтерфейс додатку	48
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58

РЕФЕРАТ

Кваліфікаційна робота: 49 с., 25 малюнків, 2 таблиці, 7 джерел.

Мета роботи: розробка функціонального вебсайту для оренди житла, бронювання польотів та дозвілля, який забезпечить користувачам зручність у використанні. Завдання роботи охоплюють аналіз потреб і вимог користувачів, проектування та створення веб-додатку, його тестування та впровадження отриманого рішення.

Об'єкт досліджування – розробка сайту для оренди житла.

Методи дослідження – технології ASP.NET MVC, C#, Entity Framework, MS SQL, JavaScript, HTML, CSS, Bootstrap, jQuery та Razor, з дотриманням архітектури MVC.

Результатом кваліфікаційної роботи є сайт для оренди житла під час подорожей, розроблений із використанням сучасних інформаційних технологій. Цей сайт забезпечує зручний інтерфейс для користувачів, дозволяючи переглядати доступні житлові об'єкти, здійснювати бронювання та відстежувати статуси своїх орендних операцій. Підготовлено детальну документацію з описом вимог та архітектури системи. Розроблений сайт дозволяє користувачам легко орендувати житло, а постачальникам — публікувати доступні апартаменти.

Ключові слова: ASP.NET MVC, C#, Entity Framework, MS SQL, JavaScript, HTML, CSS, Bootstrap, jQuery та Razor, **ОRENДА ЖИТЛА.**

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ASP.NET MVC – архітектурний шаблон для розробки веб-додатків на платформі .NET, що розділяє додаток на три основні компоненти: Model (Модель), View (Подання) та Controller (Контролер).

C# – об'єктно-орієнтована мова програмування, розроблена компанією Microsoft, яка використовується для розробки різноманітних програм, включаючи веб-додатки, мобільні додатки та серверні програми.

Entity Framework ORM – об'єктно-реляційний відображувач для .NET, що спрощує роботу з базами даних, дозволяючи розробникам працювати з даними у вигляді об'єктів.

MS SQL – реляційна система управління базами даних (RDBMS), розроблена Microsoft, яка підтримує зберігання та управління даними.

JavaScript – мова програмування, яка використовується для додавання інтерактивності до веб-сторінок, таких як анімації, форми, обробка подій та динамічні зміни контенту.

HTML – мова розмітки, що використовується для створення та структурування контенту на веб-сторінках.

CSS – мова стилів, що використовується для опису зовнішнього вигляду HTML-документів, включаючи оформлення, кольори, шрифти та розташування елементів.

Bootstrap – фреймворк для розробки фронтенду, що включає готові компоненти HTML, CSS та JavaScript для створення адаптивних та сучасних веб-інтерфейсів.

jQuery – бібліотека JavaScript, яка спрощує роботу з DOM, обробку подій, анімації та AJAX-запити, забезпечуючи кросбраузерну сумісність.

Вступ

Актуальність теми. Розробки веб-додатку для оренди житла, бронювання рейсів та організації екскурсій є безсумнівно актуальною в сучасних умовах. З розвитком цифрових технологій та глобалізацією, попит на онлайн-сервіси для подорожей значно зріс. Люди шукають зручні та ефективні способи планування своїх поїздок, включаючи бронювання житла, квитків на рейси та екскурсійних турів. Традиційні методи бронювання вже не відповідають потребам сучасних користувачів, які прагнуть швидкості, простоти та доступності інформації. Веб-додаток, подібний до Booking.com, може запропонувати користувачам зручний та інтерактивний інтерфейс для здійснення всіх необхідних бронювань в одному місці.

Такий додаток надає можливість миттєвого доступу до інформації про доступні варіанти житла, рейси та екскурсії, що значно зекономить час користувачів. Крім того, інтеграція сучасних технологій, таких як ASP.NET MVC, C#, Entity Framework, MS SQL, JavaScript, HTML, CSS, Bootstrap, jQuery та Razor, дозволяє створити надійний та функціональний продукт. Використання архітектури MVC забезпечує розділення логіки додатку, що сприяє більш ефективному управлінню кодом та його тестуванню. Це також полегшує підтримку та розвиток додатку у майбутньому.

Враховуючи, що подорожі стали більш доступними для широкого загалу, зростає необхідність у додатках, які дозволяють легко та швидко знайти й забронювати необхідні послуги. Такі додатки забезпечують прозорість процесу бронювання, що є важливим фактором для користувачів. Додаток для оренди житла, рейсів та екскурсій може також сприяти розвитку туристичної галузі, надаючи платформу для постачальників послуг для розміщення своїх пропозицій. Це створює додаткові можливості для малого та середнього бізнесу в туристичній сфері.

Крім того, інтеграція з платіжними системами дозволяє здійснювати безпечні та швидкі транзакції, що додає зручності користувачам. Можливість

залишати відгуки та рейтинги підвищує довіру до сервісу та допомагає іншим користувачам у прийнятті рішень. Актуальність теми також підкреслюється тим, що під час пандемії COVID-19 багато людей вважають за краще планувати свої поїздки онлайн, мінімізуючи особисті контакти.

Загалом, розробка такого додатку сприяє покращенню якості обслуговування туристів, забезпечує більшу гнучкість та адаптивність у процесі планування подорожей. Це актуально не лише для індивідуальних мандрівників, а й для туристичних агентств, які можуть використовувати такі платформи для покращення своїх послуг. Враховуючи всі ці фактори, тема розробки веб-додатку для оренди житла, бронювання рейсів та екскурсій є надзвичайно актуальною та відповідає сучасним вимогам ринку туристичних послуг.

Мета кваліфікаційної роботи. Метою цієї кваліфікаційної роботи є розробка функціонального веб-додатку для оренди житла, бронювання рейсів та організації екскурсій, який забезпечить зручність і ефективність використання для користувачів. Додаток повинен інтегрувати сучасні технології для створення надійного та інтуїтивно зрозумілого інтерфейсу, що дозволяє користувачам швидко знаходити та бронювати необхідні послуги. Основна увага приділяється аналізу потреб користувачів та визначенню технічних вимог для створення оптимальної архітектури системи. Веб-додаток має бути розроблений з використанням технологій ASP.NET MVC, C#, Entity Framework, MS SQL, JavaScript, HTML, CSS, Bootstrap, jQuery та Razor. Важливим аспектом є забезпечення безпеки даних та інтеграція з платіжними системами для безпечного здійснення транзакцій. Також передбачено проведення тестування для гарантування високої якості продукту та підготовка детальної технічної документації. Розроблений додаток має сприяти покращенню сервісу в туристичній галузі, забезпечуючи користувачам зручний та швидкий доступ до необхідних послуг під час подорожей.

Особистий внесок випускника. Внесок у розробку WEB -додатку для оренди житла, бронювання рейсів та організації екскурсій є значним і включає кілька ключових аспектів. Перш за все, самостійно проведено аналіз ринку та дослідження потреб цільової аудиторії, що дозволило визначити основні функції та вимоги до додатку. На основі цього аналізу розроблено детальний проект архітектури системи, включаючи базу даних та логіку роботи серверної та клієнтської частин.

Реалізовано серверну частину додатку за допомогою ASP.NET MVC та C#, використовуючи Entity Framework для взаємодії з базою даних MS SQL. Також розроблено клієнтську частину, застосовуючи сучасні технології, такі як HTML, CSS, Bootstrap, JavaScript та jQuery, що забезпечило створення зручного та інтуїтивно зрозумілого інтерфейсу для користувачів. Важливою частиною моєї роботи було забезпечення безпеки додатку, включаючи інтеграцію з платіжними системами та захист від можливих атак.

Крім того, самостійно проведено тестування додатку, виявлено та виправлено помилки, що дозволило забезпечити високу якість кінцевого продукту. У процесі розробки я створив детальну технічну документацію, яка включає опис архітектури системи, технічні вимоги, інструкції з налаштування та використання додатку. Також підготував інструкції для користувачів і адміністраторів системи, що полегшить подальше використання та обслуговування додатку.

Загалом, мій особистий внесок охоплює всі етапи розробки додатку, від аналізу та проектування до реалізації, тестування та документування, що сприяло створенню якісного та функціонального продукту, який відповідає сучасним вимогам ринку туристичних послуг.

РОЗДІЛ 1

ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Застосування сучасних інформаційних технологій у сфері оренди житла та бронювання рейсів

Індустрія оренди, що охоплює різні сектори, такі як житло, транспортні послуги та організація дозвілля, зазнала значного зростання і популярності в останні роки. Це можна пояснити кількома факторами, серед яких зміна споживчих уподобань, фінансові міркування та перехід до економіки спільного користування. Оренда пропонує людям гнучкість у доступі до активів без довгострокових зобов'язань або фінансового тягаря власності. Зручність та економічна ефективність оренди знайшли відгук у широкого кола споживачів, що призвело до різкого зростання попиту на послуги оренди в різних сферах.

Історичне коріння оренди житла можна простежити до початку 20-го століття, коли з'явилися перші готелі та пансіонати, що пропонували короткострокове проживання для туристів та ділових людей. З розвитком індустрії гостинності та зростанням доступності подорожей, послуги з оренди житла набули популярності серед ширшої клієнтської бази. Від готелів до приватних квартир, оренда житла стала більш доступною та різноманітною, надаючи можливість кожному знайти оптимальний варіант проживання.

Аналогічно, послуги бронювання рейсів та організації дозвілля також зазнали значного розвитку. Від появи перших авіакомпаній до сучасних онлайн платформ бронювання, ці послуги стали невід'ємною частиною туристичного ринку. Все більше людей шукають зручні та швидкі способи планування своїх поїздок та відпочинку, і бронювання рейсів стало простішим завдяки технологічним інноваціям.

Розвиток послуг оренди житла, бронювання рейсів та організації дозвілля був додатково пришвидшений соціальними змінами, включаючи посилення урбанізації, зростання туризму та збільшення наявного доходу. Урбанізація

сприяє збільшенню попиту на короткострокове житло, особливо серед молодих професіоналів та туристів. Зростання туризму також стимулює попит на різноманітні послуги, такі як оренда житла та бронювання рейсів, адже люди хочуть мати зручний доступ до всіх необхідних сервісів під час подорожей.

Ці фактори створили попит на тимчасове житло та зручні способи бронювання подорожей і дозвілля, що призвело до створення великих онлайн платформ, які задовольняють різноманітні потреби клієнтів. Такі платформи, як Booking.com, Airbnb, Expedia, дозволяють користувачам легко знаходити та бронювати житло, рейси та екскурсії з будь-якої точки світу. Вони надають зручні інтерфейси та широкий вибір пропозицій, що робить процес бронювання швидким і простим.

Різноманітність пропозицій та легкість доступу до них зробили ці послуги привабливими для широкого кола споживачів, що сприяло подальшому зростанню популярності орендних сервісів у сучасному світі. Користувачі цінують можливість швидко знайти потрібні послуги, порівняти ціни та умови, а також зробити бронювання в декілька кліків. Це дозволяє економити час та кошти, а також надає більше можливостей для організації подорожей та відпочинку.

Таким чином, індустрія оренди житла, бронювання рейсів та організації дозвілля продовжує зростати завдяки зміні споживчих уподобань, соціальним та економічним змінам, а також технологічним інноваціям. Цей сектор стає все більш важливим для сучасного споживача, пропонуючи зручні та гнучкі рішення для задоволення різноманітних потреб під час подорожей та відпочинку.

Практичне застосування веб-додатку для оренди житла, бронювання рейсів та організації дозвілля охоплює широкий спектр можливостей для користувачів і постачальників послуг. Для користувачів додаток значно спрощує процес планування подорожей, дозволяючи швидко знайти та забронювати житло, вибрати рейси та організувати дозвілля в одному місці. Це

забезпечує гнучкість та зручність, оскільки користувачі можуть порівнювати різні варіанти, знаходити найвигідніші пропозиції та заощаджувати кошти. Безпечні транзакції та актуальна інформація про доступні послуги підвищують довіру до додатку, а оперативність інформації допомагає користувачам приймати обґрунтовані рішення.

Для постачальників послуг додаток є ефективним інструментом залучення нових клієнтів та управління своїми пропозиціями. Інтерфейс дозволяє зручно оновлювати інформацію про наявність та ціни, відстежувати бронювання і отримувати зворотний зв'язок від клієнтів, що сприяє покращенню якості послуг. Підвищення видимості в цифровому просторі та можливості проведення маркетингових кампаній допомагають збільшити кількість бронювань і розширити ринок збуту.

Таким чином, веб-додаток для оренди житла, бронювання рейсів та організації дозвілля є потужним інструментом, який сприяє підвищенню зручності для користувачів та ефективності для постачальників послуг. Він інтегрує різні сервіси в одному місці, полегшує процеси планування та бронювання, а також підтримує розвиток економіки спільного користування, що стає все більш популярною в сучасному світі.

1.2 Огляд аналогічних застосунків

Ринок веб-сайтів для оренди житла, бронювання рейсів та організації дозвілля за останні роки зазнав значного зростання завдяки великому попиту на доступні та зручні рішення для подорожей та відпочинку. Цей ринок є динамічним та постійно розвивається, пропонуючи широкий спектр послуг для задоволення потреб клієнтів у бронюванні житла, авіарейсів та розважальних заходів. Він стає все більш популярним як серед приватних осіб, так і серед підприємств, забезпечуючи зручні та ефективні рішення для організації подорожей та відпочинку.

Сайти для оренди житла, бронювання рейсів та організації дозвілля надають різноманітний функціонал, який спрощує процес пошуку та бронювання послуг, забезпечуючи комфорт користувачів. Вони пропонують можливість швидкого та зручного пошуку доступного житла за різними критеріями, такими як тип житла, місце розташування та тривалість перебування. Крім того, вони часто надають можливість здійснити бронювання та оплату онлайн, що забезпечує швидку та безпечну транзакцію для користувачів. Багато сайтів також пропонують додаткові сервіси, такі як страхування подорожей та цілодобова підтримка клієнтів, що сприяє забезпеченню високої якості обслуговування.

Однак, на ринку сайтів для оренди житла, бронювання рейсів та організації дозвілля також існують деякі недоліки, які варто враховувати. Наприклад, деякі сайти можуть мати обмежений асортимент житла, що може звужувати вибір для користувачів та не задовольнити їхні потреби. Також не всі сайти гарантують однакову якість послуг, що може призвести до незадоволеності користувачів. Крім того, вартість оренди житла та бронювання рейсів може варіюватися на різних сайтах, і існує ризик непередбачених додаткових витрат або прихованих платежів, що може вплинути на вибір користувачів. Також більшість сайтів не надають можливість користувачам публікувати власні пропозиції щодо оренди житла.

Отже, при вході на ринок сайтів для оренди житла, бронювання рейсів та організації дозвілля важливо уважно розглянути та вирішити ці недоліки, забезпечуючи широкий вибір послуг, високу якість обслуговування та прозорість у вартості оренди.

Перейдемо до огляду існуючих сайтів, що надають можливість орендувати житло, бронювати рейси та організувати дозвілля.

1.2.1 Airbnb

Airbnb є відомою платформою для оренди житла, яка надає користувачам можливість знайти тимчасове житло у будь-якому куточку світу. Хоча цей сервіс не є спеціалізованим для оренди транспортних засобів або організації дозвілля, він пропонує широкий спектр варіантів житла, включаючи квартири, будинки та унікальні місця проживання, такі як будинки на деревах або замки.

Основні переваги платформи Airbnb:

- Широкий вибір: Airbnb забезпечує велику кількість пропозицій щодо оренди житла. Користувачі можуть знайти житло різних типів та цінових категорій, що відповідають їхнім потребам.
- Безпосередні контакти: На Airbnb можна зв'язатися безпосередньо з власником житла, що зручно для уточнення умов оренди, деталей та узгодження домовленостей.
- Можливість переглядів: Airbnb надає можливість переглядати відгуки та рейтинги житла від попередніх орендарів, що допомагає приймати обґрунтовані рішення.
- Гнучкість: Платформа Airbnb дозволяє власникам житла самостійно встановлювати умови оренди, такі як тривалість, ціна та додаткові вимоги. Користувачі можуть шукати варіанти, що найкраще відповідають їхнім вимогам та бюджету.

Основні недоліки:

- Відповідальність: Airbnb несе обмежену відповідальність за угоди між орендарями та власниками житла. Користувачам необхідно самостійно перевіряти інформацію про власників та умови оренди для забезпечення безпеки та надійності.
- Конкуренція: Завдяки великій кількості пропозицій, на Airbnb може бути значна конкуренція між орендарями. Користувачам потрібно

бути готовими до швидкого реагування та конкурувати за найкращі пропозиції.

- Комісії: Airbnb стягує комісію як з гостей, так і з господарів, що може збільшити загальну вартість оренди.
- Варіативність якості житла: Якість житла може значно варіюватися, оскільки кожна пропозиція унікальна, і не всі господарі забезпечують однаковий рівень обслуговування.

1.2.2 Booking

Booking.com є провідною платформою для онлайн-бронювання житла, яка надає користувачам можливість знайти та забронювати житло по всьому світу. Платформа пропонує широкий вибір варіантів, включаючи готелі, квартири, будинки для відпочинку, а також унікальні місця для проживання, такі як будинки на деревах, вілли та інші.

Основні переваги платформи Booking:

- Широкий вибір: Booking забезпечує великий вибір житла по всьому світу. Користувачі можуть знайти різні варіанти розміщення, від бюджетних до люксових, що відповідають їхнім потребам та бюджету.
- Безпосередні контакти: На Booking можна зв'язатися безпосередньо з готелями та власниками житла для уточнення умов бронювання, деталей та узгодження домовленостей.
- Можливість переглядів: Booking надає можливість переглядати відгуки та рейтинги житла від попередніх гостей, що допомагає приймати обґрунтовані рішення.
- Гнучкість: Платформа Booking.com дозволяє користувачам самостійно встановлювати умови бронювання, такі як тривалість, ціна та додаткові вимоги. Користувачі можуть шукати варіанти, що найкраще відповідають їхнім вимогам та бюджету.

Основні недоліки:

- **Відповідальність:** Booking несе обмежену відповідальність за угоди між орендарями та власниками житла. Користувачам необхідно самостійно перевіряти інформацію про власників та умови оренди для забезпечення безпеки та надійності.
- **Конкуренція:** Завдяки великій кількості пропозицій, на Booking може бути значна конкуренція між орендарями. Користувачам потрібно бути готовими до швидкого реагування та конкурувати за найкращі пропозиції.
- **Комісії:** Booking стягує комісію з господарів житла, що може впливати на загальну вартість оренди для гостей.
- **Варіативність якості житла:** Якість житла може значно варіюватися, оскільки кожна пропозиція унікальна, і не всі власники забезпечують однаковий рівень обслуговування.

1.2.3 Agoda

Agoda є відомою платформою для онлайн-бронювання житла, яка надає користувачам можливість знайти та забронювати житло в різних куточках світу. Платформа спеціалізується на готелях, курортах, апартаментах та інших типах проживання, надаючи користувачам зручний інструмент для планування своїх подорожей.

Основні переваги платформи Agoda:

- **Широкий вибір:** Agoda забезпечує великий вибір житла по всьому світу. Користувачі можуть знайти різні варіанти розміщення, від бюджетних до люксових, що відповідають їхнім потребам та бюджету.
- **Безпосередні контакти:** На Agoda можна зв'язатися безпосередньо з готелями та власниками житла для уточнення умов бронювання, деталей та узгодження домовленостей.

- **Можливість переглядів:** Agoda надає можливість переглядати відгуки та рейтинги житла від попередніх гостей, що допомагає приймати обґрунтовані рішення.
- **Гнучкість:** Платформа Agoda дозволяє користувачам самостійно встановлювати умови бронювання, такі як тривалість, ціна та додаткові вимоги. Користувачі можуть шукати варіанти, що найкраще відповідають їхнім вимогам та бюджету.

Основні недоліки:

- **Відповідальність:** Agoda несе обмежену відповідальність за угоди між орендарями та власниками житла. Користувачам необхідно самостійно перевіряти інформацію про власників та умови оренди для забезпечення безпеки та надійності.
- **Конкуренція:** Завдяки великій кількості пропозицій, на Agoda може бути значна конкуренція між орендарями. Користувачам потрібно бути готовими до швидкого реагування та конкурувати за найкращі пропозиції.
- **Комісії:** Agoda стягує комісію з господарів житла, що може впливати на загальну вартість оренди для гостей.
- **Варіативність якості житла:** Якість житла може значно варіюватися, оскільки кожна пропозиція унікальна, і не всі власники забезпечують однаковий рівень обслуговування.

1.3 Порівняльний аналіз сайтів-аналогів

Метою розробки веб-сайту для оренди житла, бронювання рейсів та організації дозвілля є вирішення існуючих проблем у цій індустрії та забезпечення ефективною і зручною платформи для користувачів. Для цього необхідно провести оцінку поточних веб-сайтів та визначити найбільш значущі проблеми, які потребують вирішення. У цьому дослідженні ми аналізуємо

ринок і проблеми трьох відомих веб-сайтів: Airbnb, Booking.com та Agoda. Використовуючи систему оцінювання, ми прагнемо виявити їхні сильні та слабкі сторони, визначити пріоритетні напрямки для покращення та забезпечити створення надійного і зручного веб-сайту для користувачів.

Система оцінювання, застосована в цьому дослідженні, використовує шкалу від 1 до 4, де кожен бал означає наступне:

«1» — характеристика має значні недоліки, що перешкоджають користувацькому досвіду та не відповідають основним вимогам.

«2» — характеристика відповідає мінімальним очікуванням, але має простір для вдосконалення для підвищення задоволеності користувачів.

«3» — характеристика відповідає очікуванням користувача та забезпечує задовільний досвід.

«4» — характеристика перевершує очікування користувача, демонструючи видатну продуктивність і забезпечуючи винятковий користувацький досвід.

Критерії оцінювання:

- Зручність реєстрації та входу: Простота процесу реєстрації нового користувача та авторизації.
- Дизайн та естетика: Зовнішній вигляд сайту, привабливість та сучасність дизайну.
- Швидкість завантаження: Час, необхідний для повного завантаження сторінки.
- Безпека транзакцій: Рівень захисту особистих даних та фінансових операцій.
- Асортимент пропозицій: Кількість та різноманітність доступних варіантів житла, рейсів та послуг.
- Інтерактивні функції: Наявність інтерактивних карт, чатів підтримки, калькуляторів тощо.

- Мобільна адаптація: Оптимізація сайту для мобільних пристроїв та наявність мобільного додатку.
- Зворотній зв'язок: Можливість залишати відгуки, оцінки та коментарі, а також отримувати відповіді від служби підтримки.

Аналіз сайтів для оренди житла, бронювання рейсів та організації дозвілля:

Таблиця 1.1

Аналіз сайтів для оренди житла, бронювання рейсів та організації дозвілля

Критерії	Airbnb	Booking	Agoda	Вага
Зручність реєстрації та входу	4	3	3	0,1
Дизайн та естетика	4	3	2	0,1
Швидкість завантаження	3	4	3	0,15
Безпека транзакцій	4	4	3	0,2
Асортимент пропозицій	4	4	3	0,2
Інтерактивні функції	3	3	3	0,1
Мобільна адаптація	4	4	3	0,1
Зворотній зв'язок	3	4	3	0,05

У таблиці вище кожній характеристиці присвоєно вагу, яка відображає її відносну важливість. Вага – це числове значення в межах від 0 до 1, яке відображає пріоритет кожної оцінюваної характеристики в загальній оцінці. Це дозволяє систематично порівнювати різні аспекти та визначити їхню значущість для загальної продуктивності та користувацького досвіду.

Оцінки для Airbnb, Booking.com та Agoda надані на основі їхніх показників за кожною характеристикою. Ці бали дозволяють комплексно порівняти та оцінити можливості сайтів, а також визначити їхню загальну придатність для користувачів. Розрахуємо фінальні оцінки за допомогою формули $\sum(s*w)$, де s – оцінка характеристики сайту, а w – її вага:

Таблиця 1.2

Оцінки характеристик сайтів для оренди житла, бронювання рейсів та організації дозвілля

Сайт	Оцінки
Airbnb	3,75
Booking	3,7
Agoda	3,0

Отже, оцінка веб-сайтів з оренди житла, бронювання рейсів та організації дозвілля, включаючи Airbnb, Booking.com та Agoda, надала цінну інформацію про їхні сильні та слабкі сторони. За результатами оцінки очевидно, що сайт Airbnb виявився найкращим, оскільки відповідає найважливішим характеристикам з точки зору безпеки транзакцій, асортименту пропозицій та зручності реєстрації. З іншого боку, сайт Agoda має деякі недоліки, особливо у дизайні та естетиці, а також у зворотньому зв'язку.

Розробляючи сайт для оренди житла, бронювання рейсів та організації дозвілля, важливо зосередитись на характеристиках з більшою вагою, які суттєво впливають на користувацький досвід і загальну продуктивність. Такий підхід гарантує, що отриманий веб-сайт не тільки відповідає очікуванням користувачів, але й перевершує їх у важливих аспектах, як-от безпека транзакцій, асортимент пропозицій та швидкість завантаження, про що свідчить успіх сайту Airbnb.

Крім того, процес оцінювання підкреслює важливість створення спеціалізованого веб-сайту, оптимізованого для конкретних завдань оренди житла, бронювання рейсів та організації дозвілля. Така спеціалізація забезпечує більш індивідуальний і безперешкодний користувацький досвід, сприяючи ефективному вибору житла, прозорому ціноутворенню і спрощенню процесу бронювання. Враховуючи ці висновки в процесі розробки, метою є створення

веб-сайту, який не тільки перевершить своїх конкурентів, але й забезпечить клієнтам найкращий та найприємніший досвід користування.

Таким чином, оцінка служить цінним орієнтиром для розробки надійного і орієнтованого на користувача веб-сайту для оренди житла, бронювання рейсів та організації дозвілля.

1.4 Постановка задачі

Мета проєкту – розробити зручний та ефективний веб-сайт для оренди житла, бронювання рейсів та організації дозвілля.

Цілі проєкту:

- Забезпечити інтуїтивний користувацький інтерфейс з простим процесом реєстрації та входу, оптимізованим для мобільних пристроїв.
- Покращити асортимент житла, рейсів та дозвілля, забезпечивши фільтрацію та сортування результатів, а також актуальність інформації.
- Підвищити безпеку та довіру, забезпечивши захист персональних даних та транзакцій, систему відгуків та оцінок, а також цілодобову підтримку клієнтів.
- Впровадити інтерактивні карти, комунікацію між орендодавцями та орендарями, калькулятори вартості та онлайн-бронювання.

Основні етапи розробки:

- Дослідження ринку та аналіз конкурентів для визначення сильних та слабких сторін.
- Розробка технічного завдання, проектування користувацького інтерфейсу.
- Впровадження основних функцій, тестування на різних пристроях, забезпечення безпеки даних.

- Запуск сайту, збір відгуків користувачів, підтримка та оновлення функціоналу.

Очікувані результати:

- Надійний, зручний та безпечний веб-сайт, що залучає широку аудиторію.
- Високий рівень сервісу та прозорість, що забезпечують задоволення користувачів.

Ключові показники ефективності (KPI):

- Кількість зареєстрованих користувачів.
- Рівень задоволеності користувачів.
- Час завантаження сторінок.
- Обсяг заброньованих пропозицій.
- Рівень безпеки даних.

1.5 Висновки до першого розділу

У першій частині роботи було розглянуто основні критерії, за якими можна порівнювати веб-сайти для оренди житла, бронювання рейсів та організації дозвілля з точки зору розробки. Було визначено, що такі критерії, як швидкість завантаження, безпека, мобільна сумісність, зручність інтерфейсу та використання сучасних технологій, є ключовими для забезпечення успішного та задовільного користувацького досвіду.

Детальний аналіз кожного з цих критеріїв дозволив виявити сильні та слабкі сторони різних платформ. Було встановлено, що найуспішніші сайти мають зручні та інтуїтивно зрозумілі інтерфейси, а використані технології відповідають сучасним стандартам веб-розробки.

Загалом, використання цих критеріїв дозволяє провести об'єктивне порівняння між різними веб-сайтами для оренди житла, бронювання рейсів та

організації дозвілля. Це допомагає обрати найбільш оптимальну та високоякісну платформу для користувачів, яка забезпечує зручність, надійність та задоволення від використання.

РОЗДІЛ 2

ПРОЄКТУВАННЯ САЙТУ

2.1 Вибір стеку технологій

2.1.1 ASP.NET MVC

ASP.NET MVC (Model-View-Controller) є одним з основних фреймворків для розробки веб-додатків, що базується на технології ASP.NET, розробленій компанією Microsoft. Цей фреймворк надає потужні інструменти та можливості для створення динамічних, масштабованих і підтримуваних веб-додатків, використовуючи архітектуру MVC.

Архітектура MVC: ASP.NET MVC реалізує патерн проектування Model-View-Controller, який розділяє додаток на три основні компоненти:

- **Model (Модель):** Відповідає за керування даними, бізнес-логікою і правилами додатку.
- **View (Представлення):** Відповідає за відображення даних користувачу. Представлення отримує дані з моделі та відображає їх у відповідному форматі.
- **Controller (Контролер):** Відповідає за обробку запитів користувачів, взаємодію з моделлю для отримання або обробки даних та передачу цих даних до представлення.

Тестування та підтримка TDD: ASP.NET MVC підтримує розробку через тестування (TDD), що дозволяє легко створювати юніт-тести для різних частин додатку, включаючи контролери та моделі.

Масштабованість та гнучкість: Завдяки архітектурі MVC, ASP.NET MVC дозволяє легко розширювати та масштабувати додатки, що робить його ідеальним для великих корпоративних рішень.

SEO-дружність: Фреймворк дозволяє створювати веб-сторінки з чистими та дружніми для SEO URL, що покращує видимість додатку у пошукових системах.

Робота з JavaScript-фреймворками: ASP.NET MVC легко інтегрується з популярними JavaScript-фреймворками, такими як Angular, React, Vue.js, що дозволяє створювати інтерактивні та динамічні веб-додатки.

Переваги використання ASP.NET MVC такі:

- Розділення завдань: Завдяки розділенню на модель, представлення та контролер, розробники можуть чітко розподілити завдання між різними командами, що підвищує ефективність та продуктивність роботи.
- Легкість тестування: Розробка через тестування (TDD) стає простішою завдяки чіткій структурі фреймворку, що дозволяє легко створювати та підтримувати тестові набори.
- Висока продуктивність: ASP.NET MVC забезпечує високу продуктивність завдяки оптимізації обробки запитів і відображення даних.
- Гнучкість у розробці: Фреймворк дозволяє легко інтегрувати різноманітні сторонні бібліотеки та фреймворки, що надає розробникам більшу гнучкість у створенні додатків.
- Підтримка від Microsoft: Як частина екосистеми Microsoft, ASP.NET MVC отримує регулярні оновлення та підтримку, що забезпечує стабільність та безпеку додатків.

Недоліки ASP.NET MVC наступні.

Крута крива навчання: Для новачків у програмуванні або тих, хто не знайомий з архітектурою MVC, ASP.NET MVC може мати круту криву навчання.

Відносно висока вартість: Використання ASP.NET MVC може бути дорожчим у порівнянні з деякими іншими фреймворками, особливо через ліцензійні вимоги та інструменти Microsoft.

ASP.NET MVC є потужним фреймворком для розробки веб-додатків, що забезпечує високу продуктивність, гнучкість та масштабованість. Завдяки чіткій архітектурі MVC, підтримці TDD та можливості інтеграції з сучасними JavaScript-фреймворками, ASP.NET MVC залишається популярним вибором для розробників, що створюють складні та динамічні веб-додатки. Його переваги, включаючи підтримку від Microsoft, легкість тестування та SEO-дружність, роблять його привабливим для корпоративних рішень та великих проектів.

2.1.2 Entity Framework

Entity Framework (EF) — потужний ORM (Object-Relational Mapping) фреймворк для .NET, який значно спрощує роботу з базами даних, дозволяючи розробникам взаємодіяти з даними у вигляді об'єктів .NET. Завдяки своїм широким можливостям та гнучкості, Entity Framework отримав велику популярність у сфері розробки додатків.

Розглянемо основні характеристики Entity Framework. Entity Framework надає розробникам можливість працювати з базами даних, використовуючи об'єкти .NET, що значно зменшує кількість необхідного SQL-коду. EF автоматично генерує SQL-запити для взаємодії з базою даних на основі операцій, виконаних над об'єктами, що значно спрощує процес розробки та знижує ризик виникнення помилок.

EF підтримує кілька підходів до роботи з даними, включаючи Code First, Database First та Model First. Це забезпечує гнучкість у виборі методу розробки, що найкраще підходить для конкретного проекту.

- Code First підхід дозволяє створювати моделі даних, використовуючи код C#, а потім автоматично генерувати схему бази даних.
- Database First підхід використовується, коли розробники вже мають готову базу даних, і вони бажають створити моделі об'єктів на основі існуючої схеми.
- Model First підхід дозволяє створювати моделі даних у вигляді діаграм, а потім генерувати схему бази даних та код об'єктів на основі цих діаграм.

Однією з ключових особливостей Entity Framework є підтримка LINQ (Language Integrated Query), що дозволяє розробникам писати запити до бази даних, використовуючи синтаксис C#. Це забезпечує інтуїтивно зрозумілий та потужний спосіб взаємодії з даними, покращуючи продуктивність та зручність роботи.

Існує кілька вагомих причин для вибору Entity Framework для роботи з базами даних у додатках .NET. По-перше, EF значно знижує складність та обсяг коду, необхідного для роботи з базами даних, автоматизуючи процеси генерації SQL-запитів та управління схемою бази даних. Це дозволяє розробникам зосередитися на бізнес-логіці додатку, не витрачаючи час на написання та налагодження SQL-коду.

По-друге, Entity Framework забезпечує високу продуктивність завдяки підтримці оптимізації запитів та кешування. EF використовує відкладену та нехайну завантаження даних для мінімізації кількості запитів до бази даних, що покращує продуктивність додатків. Крім того, EF підтримує механізми трасування та профілювання запитів, що дозволяє розробникам виявляти та усувати вузькі місця у продуктивності.

Крім того, Entity Framework забезпечує високу гнучкість та розширюваність завдяки підтримці різних конфігураційних параметрів та можливості інтеграції з іншими компонентами .NET. Розробники можуть

налаштовувати поведінку EF, використовуючи анотації даних або Fluent API, що дозволяє точно відповідати вимогам конкретного проекту.

2.1.3 MS SQL

MS SQL Server (Microsoft SQL Server) — це потужна реляційна система управління базами даних (RDBMS), розроблена корпорацією Microsoft. Завдяки своїй високій продуктивності, надійності та широким можливостям, MS SQL Server є одним із провідних рішень для роботи з базами даних у корпоративних середовищах.

MS SQL Server пропонує розробникам широкий набір інструментів та функцій для ефективного управління даними. Однією з ключових особливостей є T-SQL (Transact-SQL) — розширення SQL, яке включає додаткові можливості для програмування, зокрема, контроль потоку, обробку винятків і змінні. T-SQL дозволяє створювати складні запити, зберігані процедури та тригери, що робить роботу з базою даних більш гнучкою та потужною.

MS SQL Server підтримує різні режими зберігання даних, включаючи індексоване збереження (columnstore indexes), яке оптимізоване для великих аналітичних запитів і забезпечує високу швидкість обробки даних. Це особливо корисно для бізнес-аналітики та обробки великих обсягів даних.

Однією з важливих функцій MS SQL Server є реплікація даних, що дозволяє копіювати дані між серверами та забезпечує високу доступність і відмовостійкість системи. Реплікація підтримує кілька методів, таких як транзакційна реплікація, знімова реплікація та реплікація злиття, що дозволяє вибрати найбільш підходящий варіант для конкретних потреб.

Розглянемо переваги використання MS SQL Server.

MS SQL Server забезпечує високу продуктивність завдяки різним механізмам оптимізації запитів, управління пам'яттю та паралельної обробки.

Алгоритми оптимізації запитів автоматично вибирають найефективніші шляхи виконання запитів, мінімізуючи час відповіді та знижуючи навантаження на систему. Крім того, MS SQL Server підтримує технологію in-memory OLTP, яка дозволяє зберігати та обробляти таблиці у пам'яті, що значно прискорює обробку транзакцій.

Надійність та безпека є важливими аспектами MS SQL Server. Система підтримує шифрування даних, аутентифікацію та авторизацію на основі ролей, що забезпечує захист даних від несанкціонованого доступу. Крім того, MS SQL Server пропонує механізми резервного копіювання та відновлення даних, що дозволяє забезпечити безперервність бізнес-процесів навіть у разі відмови обладнання або програмного збою.

MS SQL Server має потужні інструменти для адміністрування та моніторингу баз даних, такі як SQL Server Management Studio (SSMS) та SQL Server Profiler. Ці інструменти дозволяють адміністраторам ефективно керувати базами даних, відстежувати продуктивність системи, виявляти та усувати проблеми.

Широка інтеграція з іншими продуктами Microsoft є ще однією перевагою MS SQL Server. Він легко інтегрується з такими інструментами, як Visual Studio, Power BI, та Azure, що дозволяє створювати комплексні рішення для аналізу та візуалізації даних, розробки додатків та управління інфраструктурою.

2.2 Опис функціоналу системи

Сайт для бронювання житла, рейсів та дозвілля буде працювати наступним чином: неавторизований користувач заходить на наш сайт. Користувач може обрати тип послуги (житло, рейси, дозвілля) і переглянути доступні пропозиції.

Перегляд пропозицій здійснюється за кількома категоріями.

Житло.

Користувач може вибрати місто та знайти доступне житло яке сподобається, або знайти конкретне місце за пошуком. На основі параметрів система покаже список доступного житла з фільтрами за ціною, типом житла, зручностями та іншими характеристиками. Картки з житлом міститимуть фото, опис, правила, локацію та ціну.

Рейси.

Користувач може ввести місце відправлення та призначення, відсортувати за ціною, або авіалініями. Відобразатимуться доступні рейси з фільтрами за авіакомпанією, ціною. Картки з рейсами міститимуть деталі про час відправлення та прибуття, тривалість польоту, пересадки та ціну.

Дозвілля.

Дозвілля є додатковою фічею при бронюванні житла. Воно пропонується при відображенні деталей житла. Або ж можна вибрати місто та дати проведення заходу або активності. Відобразатимуться доступні варіанти з фільтрами за типом активності, ціною, тривалістю. Картки з дозвіллям міститимуть фото, опис та ціну.

Бронювання.

Існують обмеження для неавторизованих користувачів.

Незареєстровані або неавторизовані користувачі можуть переглядати всі доступні пропозиції, але не можуть здійснювати бронювання. При спробі забронювати користувач буде перенаправлений на сторінку авторизації або реєстрації.

На сторінці реєстрації Користувачі можуть створити обліковий запис, ввівши необхідні дані, такі як електронна пошта, повне ім'я, електронну пошту та пароль. Після реєстрації користувачеві буде доступний весь функціонал.

На сторінці авторизації зареєстровані користувачі можуть увійти, ввівши свою електронну адресу та пароль. Після авторизації користувачі отримують повний доступ до функціоналу бронювання.

Додаткові функції для зареєстрованих користувачів наступні.

Можливість бронювання.

Зареєстровані та авторизовані користувачі можуть завершити процес бронювання, підтвердити та оплатити.

Особистий кабінет.

Зареєстровані користувачі можуть переглядати та керувати своїми бронюваннями, а також переглянути історію.

2.2.1 Функціонал неавторизованого користувача

1. Перегляд доступних позицій по житлу, рейсах та дозвіллю.
2. Використання фільтрів для пошуку.
2. Перегляд деталей житла, рейсів.
4. Перегляд списку доступних заходів, таких як концерти, екскурсії, вистави тощо.
5. Реєстрація в системі з введенням необхідних даних (електронна пошта, повне ім'я та пароль).
6. Перегляд основної інформації про сайт.

2.2.2 Функціонал для авторизованого користувача

1. Доступ до головної сторінки для вибору місця та дати оренди
2. Здійснення бронювання вибраного житла, рейсів та дозвілля.

3. Перегляд історії бронювань.
4. Можливість підтвердження бронювання.
5. Доступ до особистого кабінету.
6. Можливість оплати послуг онлайн.
7. Зміна інформації у профілі користувача.
8. Зміна пароля.

2.2.3 Функціонал для модератора

1. Перехід до панелі керування після входу в систему.
2. Додавання нового житла, рейсів та дозвілля.
3. Доступ до списку всіх користувачів та модераторів без можливості змінювати їхню інформацію.
4. Можливість змінювати доступні позиції.

2.3 Розробка вимог за допомогою діаграми прецедентів

Діаграма прецедентів – це графічне відображення взаємодії між акторами (користувачами або зовнішніми системами) і розглянутою системою. Вона демонструє різні варіанти використання або функціональні можливості системи у стислому та зрозумілому вигляді. Діаграми прецедентів відображають поведінку системи з точки зору користувача, показуючи різні дії, які можуть бути виконані, та взаємозв'язки між ними.

Формування вимог за допомогою діаграми прецедентів зосереджене на фіксації та документуванні функціональних вимог до системи з точки зору її користувачів. Вона акцентує увагу на розумінні учасників, їхніх цілей і взаємодій, необхідних для досягнення цих цілей. Діаграма виступає як план для

проектування системи, надаючи візуальне представлення, що полегшує проектування та розробку застосунку.

Процес створення діаграми прецедентів включає збір інформації від зацікавлених сторін, включаючи кінцевих користувачів. Такий підхід допомагає виявити та зафіксувати ключові вимоги, які могли б залишитися непоміченими. Він сприяє активній участі та гарантує, що система відповідатиме потребам і очікуванням цільових користувачів.

Діаграми прецедентів допомагають визначити межі та сферу застосування системи, визначаючи учасників і їхні відповідні цілі. Вони надають загальний огляд функціональних можливостей системи, що дозволяє проектним групам визначати пріоритети вимог, ефективно розподіляти ресурси та керувати обмеженнями проекту. Також вони встановлюють простежуваний зв'язок між вимогами користувачів і функціональністю системи, що дозволяє відстежувати і контролювати хід реалізації, гарантуючи, що всі визначені варіанти використання враховані і виконані в процесі розробки.

Враховуючи всі наведені вимоги та функції, необхідно створити діаграму прецедентів для виявлення функціональних можливостей системи та поведінки системи з точки зору користувача. Актори на наступних діаграмах включатимуть: неавторизованого користувача, авторизованого користувача, постачальника та адміністратора системи.

На рисунку 2.1 зображено функціональні можливості неавторизованого користувача на нашому сайті.

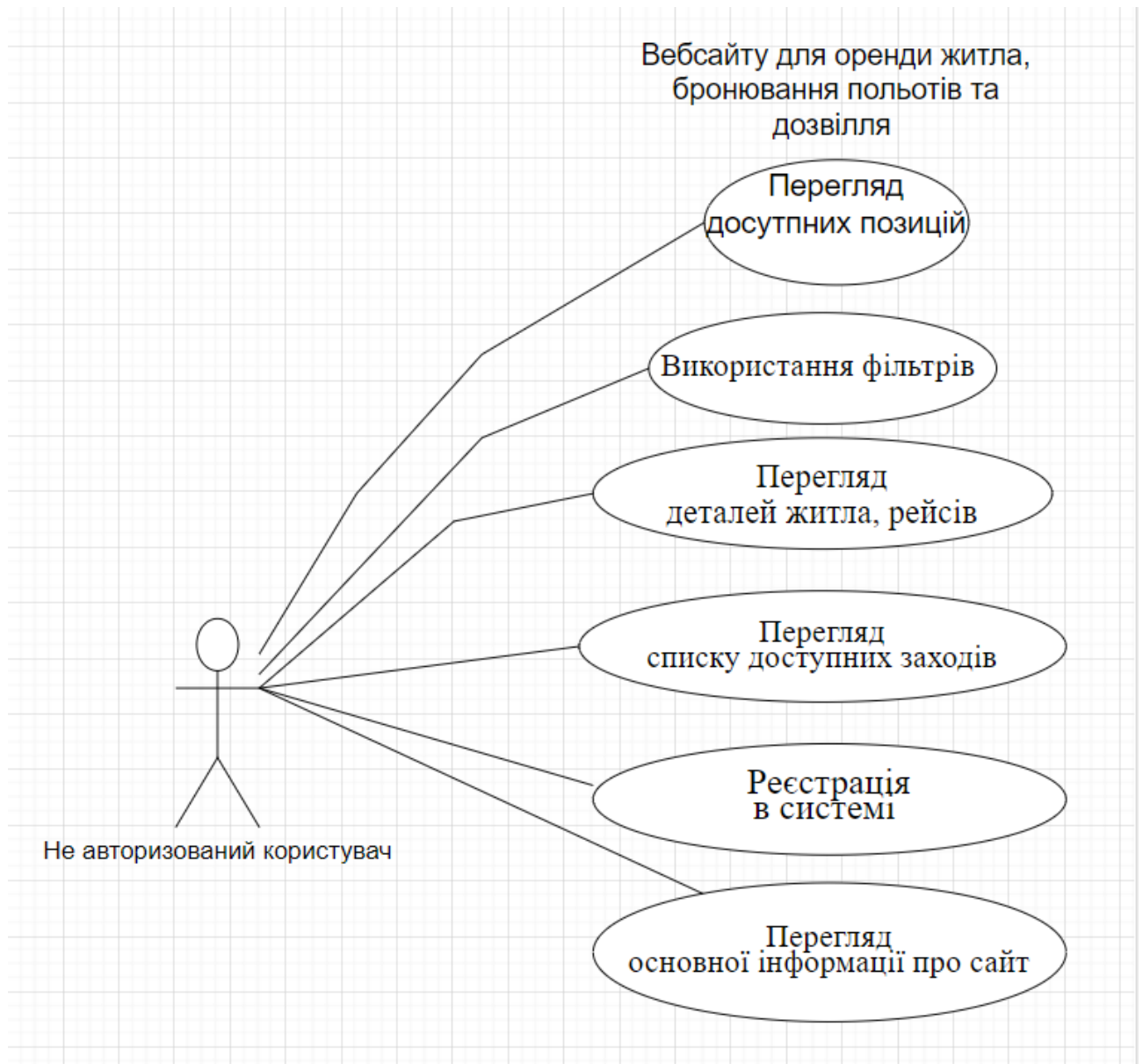


Рис. 2.1. Діаграма прецедентів неавторизованого користувача і сайту

На рисунку 2.2 представлені функціональні можливості авторизованого користувача на сайті. Для цього типу користувача функцій більше, оскільки це дозволяє прив'язати всі введені ним дані до його профілю. На рисунках 2.3 показані способи використання сайту модератором.

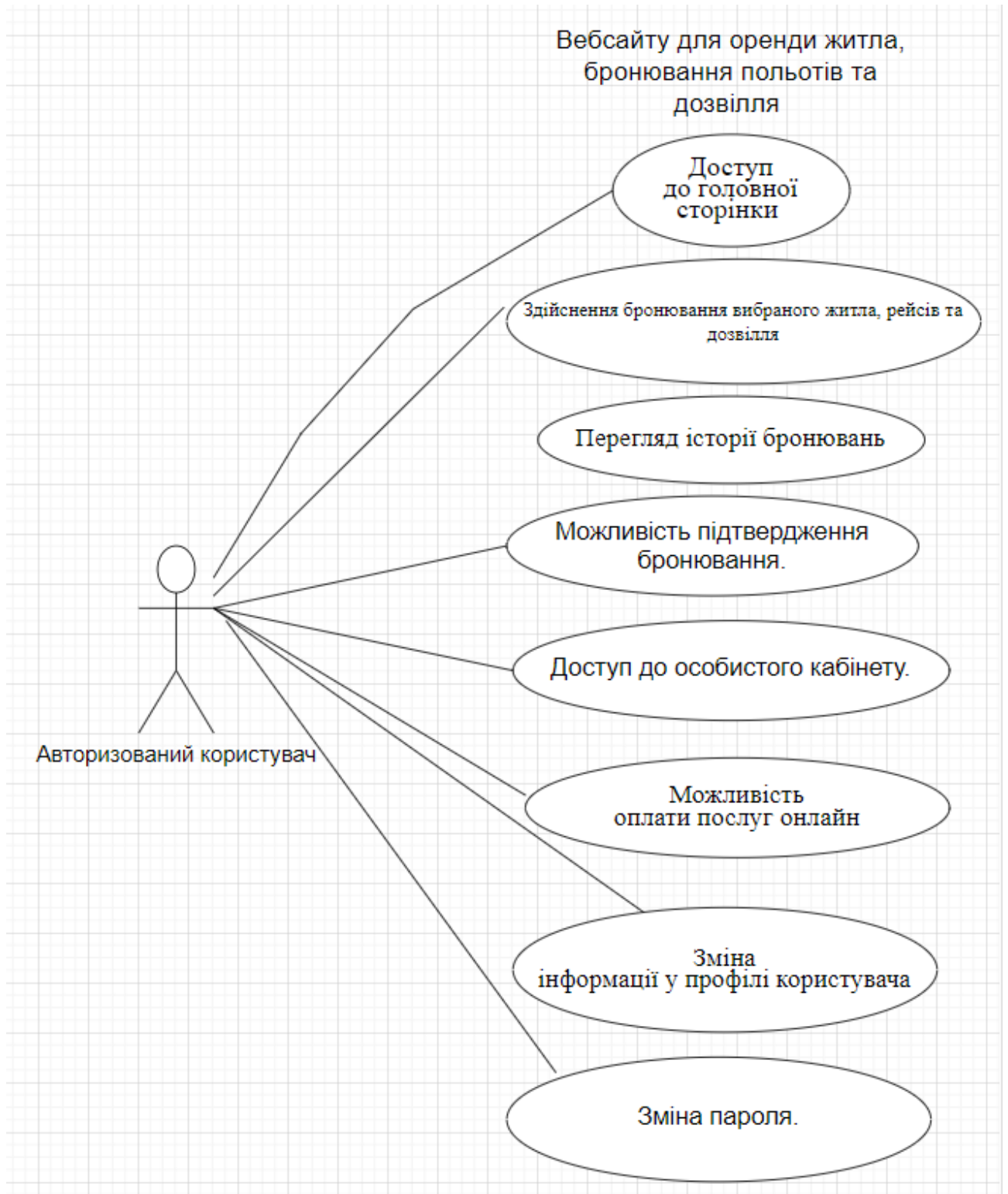


Рис. 2.2. Зображено діаграму, що показує, як авторизований користувач взаємодіє з сайтом.

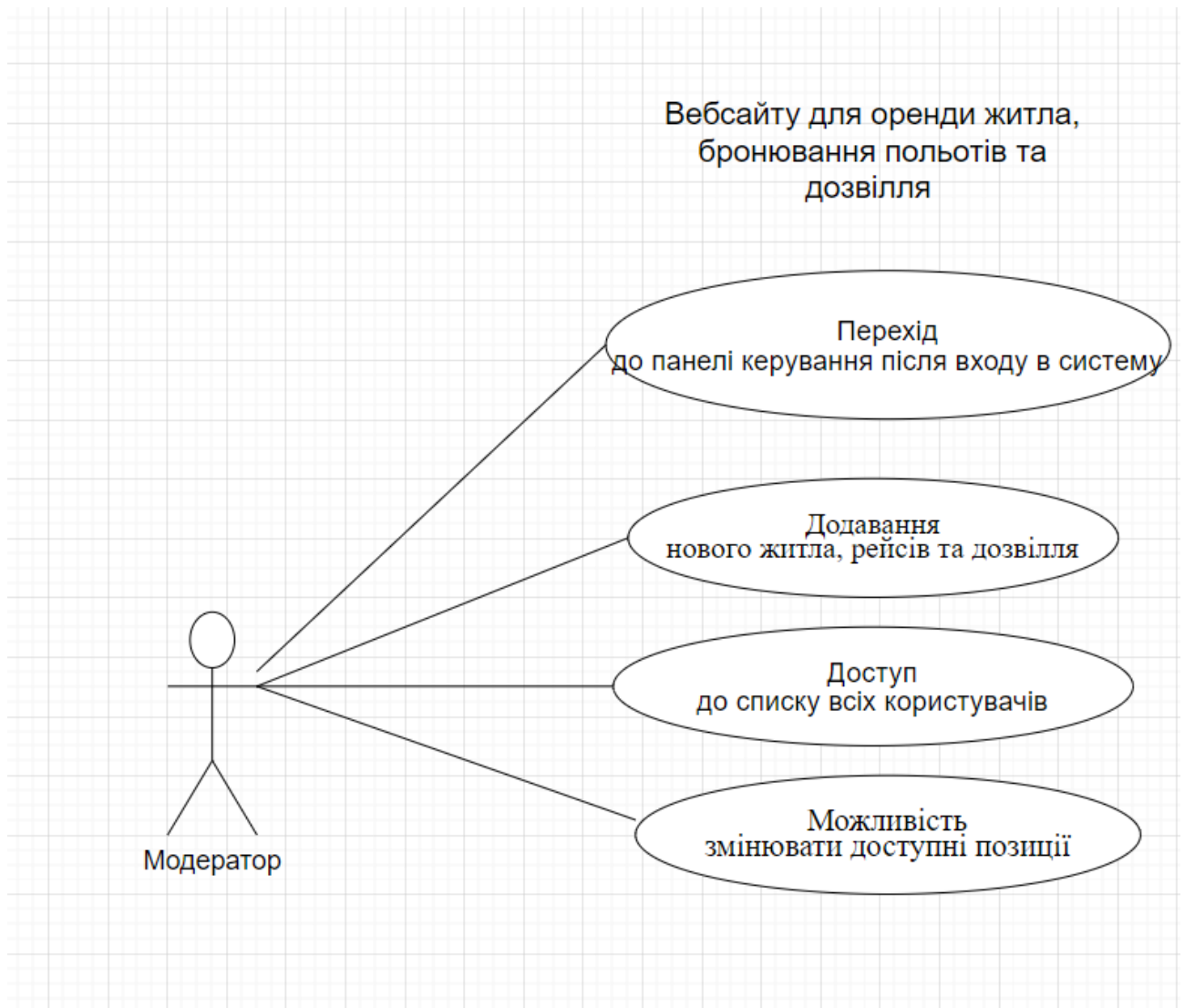


Рис. 2.3. Діаграма взаємодії модератора з сайтом

2.4 Проектування архітектури сайту

Архітектура сайту для оренди житла, бронювання рейсів та дозвілля грає важливу роль у забезпеченні його функціональності, масштабованості та ефективності.

Фронтенд частина сайту, побудована на основі ASP.NET MVC, формує інтерфейс для користувачів. Вона використовує HTML, CSS та JavaScript для створення візуально привабливого та інтуїтивно зрозумілого інтерфейсу. Цей інтерфейс дозволяє користувачам шукати житло, бронювати рейси та обирати розважальні заходи.

Інтерфейс адміністратора надає засоби для управління вмістом сайту. Він містить функції для додавання та редагування житла, рейсів та розважальних заходів, а також для керування користувачами та бронюваннями.

Бекенд сайту відповідає за обробку запитів користувачів та управління базою даних. Він використовує технології ASP.NET MVC та Entity Framework для взаємодії з базою даних MS SQL Server. Бекенд реалізує бізнес-логіку, обробку запитів від клієнтів та забезпечує безперебійну роботу сайту.



Рис. 2.4. Діаграма структури сайту

2.5 Архітектура бекенд частини

Архітектура бекенд частини сайту для оренди житла, бронювання рейсів та дозвілля побудована на основі ASP.NET MVC і включає в себе кілька ключових компонентів Models, Views, Controllers. На рисунку 2.5 представлена діаграма шаблону MVC.

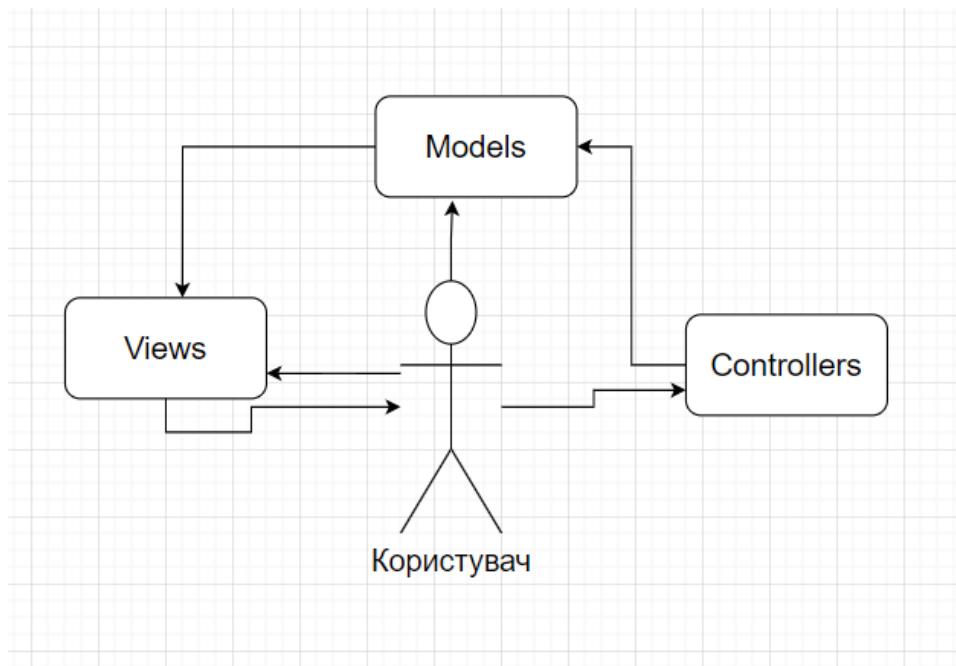


Рис. 2.6. Діаграма MVC патерну.

Контролери (Controllers) відповідають за обробку HTTP-запитів і виконання відповідних дій. Вони отримують дані від користувачів, обробляють їх за допомогою сервісів та взаємодіють з моделями для збереження або отримання інформації з бази даних. Наприклад, контролери можуть включати такі класи, як HomeController, HotelController, та LandmarkController.

Моделі (Models) визначають структуру даних, які використовуються в додатку. Вони взаємодіють з Entity Framework для збереження і отримання даних з бази даних. Прикладами моделей можуть бути User, Property, Flight.

Видошаблони (Views) відповідають за відображення даних, отриманих від контролерів, у вигляді HTML-сторінок. Вони використовують Razor-синтаксис для динамічного створення HTML-коду на основі даних моделей. Типовими шаблонами є Index.cshtml, EditReservation.cshtml, та Details.cshtml.

Логіка бізнес-процесів винесена в окремі сервіси (Services), що забезпечує розділення обов'язків і полегшує тестування. Сервіси виконують обробку даних

і взаємодіють з моделями та репозиторіями для виконання операцій з базою даних. Прикладами сервісів є FlightService, HotelService, та LandmarkService.

Репозиторії (Repositories) відповідають за доступ до бази даних і використовують Entity Framework для виконання CRUD-операцій (створення, читання, оновлення, видалення) з даними. Наприклад, це може бути Repository.

Entity Framework є ORM (об'єктно-реляційним маппером), який використовується для взаємодії з базою даних MS SQL Server. Він дозволяє працювати з даними як з об'єктами, забезпечуючи зручний і ефективний спосіб доступу до бази даних. Моделі даних налаштовані через клас DbContext, який визначає набори даних (DbSet) і конфігурації для роботи з таблицями бази даних.

MS SQL Server є реляційною базою даних, яка використовується для зберігання інформації про користувачів, житло, рейси, дозвілля та інші дані, необхідні для функціонування сайту. База даних містить таблиці, індекси, збережені процедури та тригери, які забезпечують ефективне зберігання та доступ до даних.

Мідлвари (Middleware) є компонентами, які обробляють запити на шляху від клієнта до контролера і назад. Вони можуть виконувати автентифікацію, авторизацію, логування, обробку помилок та інші функції. Прикладами мідлварів є AuthenticationMiddleware та ErrorHandlingMiddleware.

Файли конфігурації (Configuration), такі як appsettings.json, використовуються для зберігання налаштувань додатка. Це включає налаштування бази даних, налаштування логування, параметри автентифікації та інші параметри.

Архітектура бекенд частини забезпечує чітке розділення обов'язків, що полегшує розробку, тестування та підтримку додатку. Використання сучасних технологій ASP.NET MVC, Entity Framework та MS SQL Server дозволяє створити надійний та масштабований веб-додаток.

2.6. Висновки до другого розділу

У другому розділі було детально розглянуто архітектуру бекенд частини сайту для оренди житла, бронювання рейсів та дозвілля. Було визначено, що використання ASP.NET MVC як основної технології дозволяє створити структуровану та легко підтримувану систему. Контролери забезпечують обробку запитів користувачів, моделі визначають структуру даних, а видошаблони відповідають за відображення інформації.

Виділення логіки бізнес-процесів в окремі сервіси сприяє розділенню обов'язків, що покращує читабельність і тестованість коду. Використання репозиторіїв для доступу до бази даних через Entity Framework забезпечує ефективну і зручну роботу з даними. Використання MS SQL Server як основної бази даних забезпечує надійне зберігання та швидкий доступ до необхідної інформації.

Застосування мідлварів для обробки запитів дозволяє реалізувати такі важливі аспекти, як автентифікація, авторизація та обробка помилок. Використання файлів конфігурації для зберігання налаштувань додатку дозволяє гнучко керувати параметрами системи.

Таким чином, розроблена архітектура бекенд частини сайту є надійною, масштабованою та легко підтримуваною. Вона забезпечує необхідний функціонал для користувачів, дозволяючи їм зручно орендувати житло, бронювати рейси та організувати дозвілля. Використання сучасних технологій та підходів дозволяє створити високоякісний продукт, який відповідає всім вимогам сучасних веб-додатків.

РОЗДІЛ 3

РОЗРОБКА ДОДАТКУ

3.1 Розробка структури бази даних

Розробка веб-сайту з оренди житла, бронювання рейсів та організації дозвілля вимагає добре продуманої та ефективної структури бази даних для зберігання та управління різними сутностями та їхніми зв'язками. Для роботи з базою даних MS SQL на нашому сервері будемо використовувати підхід Code First від Entity Framework Core.

Entity Framework Core (EF Core) – це сучасна платформа ORM (Object-Relational Mapping), яка дозволяє розробникам працювати з базою даних за допомогою об'єктів .NET. Підхід Code First дозволяє спочатку створювати класи моделі в коді, а потім використовувати ці моделі для створення бази даних. EF Core автоматично генерує схеми бази даних на основі визначених класів, що спрощує процес розробки та управління базою даних.

Схеми бази даних для ключових сутностей:

- Hotel
- HotelReservation
- Landmark
- LandmarkReservation
- Flight
- FlightReservation
- ApplicationUser
- HotelOwner
- PaymentHistory

На рисунку 3.1. зображено діаграму частини сутностей бази даних, які були показані вище.



Рис. 3.1. Діаграма взаємозв'язків сутностей в базі даних.

У наведеному прикладі коду зображено приклад, як відбувається процес бронювання житла. Подібним чином відбуваються і бронювання інших послуг.

```
[HttpPost]
0 references
public async Task<IActionResult> Reserve(HotelReservationInputModel model)
{
    if (await hotelService.RoomExistsAsync(model.Room_Id) == false)
    {
        ModelState.AddModelError(nameof(model.Room_Id), "Кімнати не існує!");
    }

    DateTime startDate = DateTime.ParseExact(model.StartDate, DateTimeFormat, CultureInfo.InvariantCulture);
    DateTime endDate = DateTime.ParseExact(model.EndDate, DateTimeFormat, CultureInfo.InvariantCulture);

    if(endDate <= startDate)
    {
        ModelState.AddModelError(nameof(model.EndDate), "Дата завершення має бути щонайменше через один день після дати початку!");
    }

    if (!ModelState.IsValid)
    {
        var room = await hotelService.GetRoomAsync(model.Room_Id);
        model.Rooms = await hotelService.GetRoomsAsync(room.Hotel_Id);
        return View(model);
    }

    string userId = User.GetUserId();

    _ = await hotelService.ReserveAsync(model, userId);

    return RedirectToAction(nameof(Verify));
}
```

Рис. 3.2. Приклад коду бронювання.

3.2 Програмна реалізація додатку

Реалізація програми для оренди житла, бронювання рейсів та дозвілля передбачає створення комплексного веб-додатку з використанням технологій ASP.NET MVC, MS SQL та Entity Framework. Основні кроки реалізації включають налаштування середовища розробки, створення моделей даних, розробку контролерів, реалізацію логіки бізнес-процесів та створення представлень для користувачів.

Для налаштування служб і підключення до бази даних у додатку ASP.NET Core MVC необхідно налаштувати залежності та конфігурацію у файлі Program.cs.

Це включає налаштування служб Entity Framework Core для взаємодії з базою даних MS SQL та налаштування аутентифікації та авторизації користувачів.

Розглянемо процес налаштування залежностей. У файлі Program.cs додаємо залежності для використання Entity Framework Core та Identity:

```
0 references
public static async Task Main(string[] args)
{
    var builder = WebApplication.CreateBuilder(args);

    builder.Services.AddApplicationDbContext(builder.Configuration);
    builder.Services.AddApplicationIdentity(builder.Configuration);

    builder.Services.AddControllersWithViews(options =>
    {
        options.Filters.Add<AutoValidateAntiforgeryTokenAttribute>();
    });

    builder.Services.AddApplicationServices();
    builder.Services.AddMemoryCache();

    var app = builder.Build();

    if (app.Environment.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseMigrationsEndPoint();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error/500");
        app.UseStatusCodePagesWithReExecute("/Home/Error", "{statusCode}");
        app.UseHsts();
    }

    app.UseHttpsRedirection();
    app.UseStaticFiles();

    app.UseRouting();
}
```

Рис. 3.3. Налаштування служб в Program.cs.

Тут ми додаємо службу для ApplicationDbContext, яка використовує SQL Server для підключення до бази даних. Рядок підключення до бази даних береться з конфігурації в даному випадку винесено в окремий файл. Зображено на рисунку 3.4.

```

1 reference
public static IServiceCollection AddApplicationDbContext(this IServiceCollection services, IConfiguration config)
{
    var connectionString = config.GetConnectionString("DefaultConnection");

    services.AddDbContext<RentApartmentsDbContext>(options =>
        options.UseSqlServer(connectionString));

    services.AddDatabaseDeveloperPageExceptionFilter();

    services.AddScoped<IRepository, Repository>();

    return services;
}

```

Рис. 3.4. Підключення до бази даних

Для налаштування файлу appsettings.json в ASP.NET Core, де визначаються конфігураційні параметри, такі як підключення до бази даних, ось приклад того, як це може виглядати для вашого проекту з використанням Entity Framework Core та MS SQL. Налаштування файлу appsettings.json зображено на рисунку 3.5.

```

tps://json.schemastore.org/appsettings.json
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=EISMONT;Database=BookingSystemDb;Trusted_Connection=True;MultipleActiveResultSets=true",
    "BookingSystemDbContextConnection": "Server=EISMONT;Database=BookingSystem;Trusted_Connection=True;MultipleActiveResultSets=true"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
}

```

Рис. 3.5. Налаштування файлу appsettings.json

Конструктор класу контексту бази даних у ASP.NET Core з використанням Entity Framework Core містить кілька ключових елементів. Основною метою цього конструктора є налаштування з'єднання з базою даних та інших параметрів, необхідних для коректної роботи з базою даних у додатку. Нижче на рисунку 3.6. зображено, як виглядає конструктор класу контексту бази даних в ASP.NET Core з використанням Entity Framework Core.

```

30 references
public class RentApartmentsDbContext : IdentityDbContext<ApplicationUser>
{
    7 references
    public RentApartmentsDbContext(DbContextOptions<RentApartmentsDbContext> options)
        : base(options)
    {
    }
    0 references
    protected override void OnModelCreating(ModelBuilder builder)
    {
        builder.ApplyConfiguration(new ContinentConfiguration());
        builder.ApplyConfiguration(new CountryConfiguration());
        builder.ApplyConfiguration(new CityConfiguration());
        #region
        builder.ApplyConfiguration(new UserConfiguration());
        builder.ApplyConfiguration(new UserClaimsConfiguration());

        base.OnModelCreating(builder);
    }
}

```

Рис 3.6. Конструктор класу контексту бази даних

Контролери в архітектурі веб-додатків відповідають за обробку запитів від клієнтів і відправку відповідей назад. Вони дозволяють взаємодіяти з даними та бізнес-логікою додатка. Контролери мають методи, які відповідають різним HTTP-запитам, таким як GET, POST, PUT, DELETE. Ці методи приймають дані від клієнта, обробляють їх і повертають відповідь, часто у вигляді HTML-сторінки, JSON-об'єкта або іншого формату, залежно від типу запиту та потреб додатка. Контролери також виконують функції валідації вхідних даних, авторизації користувачів і керування переадресаціями на різні сторінки або дії в додатку. Вони є ключовим елементом в паттерні Model-View-Controller (MVC), де вони відповідають за логіку взаємодії моделей (даних) та представлень (відображень) додатка. Приклад контроллера зображено на рисунку 3.7. де зображено контроллер HotelController та метод EditReservation.

```

3 references
public class HotelController : BaseController
{
    private readonly IHotelService hotelService;
    private readonly ILandmarkService landmarkService;
    private readonly ILogger<HotelController> logger;
    0 references
    public HotelController(IHotelService hotelService,
        ILandmarkService landmarkService,
        ILogger<HotelController> logger)
    {
        this.hotelService = hotelService;
        this.landmarkService = landmarkService;
        this.logger = logger;
    }

    [HttpGet]
    0 references
    public async Task<IActionResult> EditReservation(string id)
    {
        if (await hotelService.HotelReservationExistsAsync(id) == false)
        {
            return BadRequest();
        }

        string userId = User.GetUserId();

        try
        {
            var model = await hotelService.GetForEditReservationAsync(id, userId);

            return View(model);
        }
        catch (UnauthorizedAccessException uae)
        {
            logger.LogError(uae, "Hotel/EditReservation[GET]");

            return Unauthorized();
        }
    }
}

```

Рис. 3.7. Приклад контролера HotelController та метод EditReservation

Представлення (Views) в архітектурі веб-додатків відображають інформацію користувачеві та забезпечують взаємодію з ним через інтерфейс. Вони генерують HTML-код для відображення структури сторінки, включаючи тексти, кнопки, форми та інші елементи. Розмітка сторінки визначається за допомогою HTML, який відповідає за структуру контенту. CSS використовується для стилізації елементів, забезпечення їхнього оформлення, розміщення та вигляду на сторінці.

У розробці веб-додатків з використанням ASP.NET MVC, представлення створюються з використанням розширення файлу .cshtml (для C#) або .vbhtml (для VB.NET) з розміткою Razor. Razor дозволяє вбудовувати код C# або

VB.NET безпосередньо в HTML-сторінку, що дозволяє відображати динамічні дані на сторінці.

Для відображення даних на сторінці можна використовувати модель вигляду (ViewModel), яка містить необхідні дані для конкретного представлення. Це дозволяє окремо визначати, які саме дані потрібно відобразити на сторінці, що зменшує залежність представлення від структури моделі даних.

Також важливим аспектом представлень є використання макетів (Layouts), які дозволяють створювати загальний вигляд для всіх сторінок додатку. Макети дозволяють зберегти єдність у дизайні та зручно керувати розміткою сторінок.

Нарешті, для додаткової інтерактивності та функціональності на сторінці можна використовувати JavaScript та бібліотеки, такі як jQuery або React, які дозволяють реалізувати асинхронні запити до сервера, валідацію форм, анімацію та інші ефективні функції.

На рисунку 3.8. зображено view відображення усього доступного житла. View містить розмітку, механізми частини коду C#, які дозволяє використовувати Razor page.

```

@{
    ViewData["Title"] = "Foreni";
}
<h2 class="text-center">@ViewData["Title"]</h2>
<hr />
<form method="get">
    <div class="row justify-content-center">
        <div class="form-group col-md-3 align-content-center">
            <label asp-for="City"></label>
            <select asp-for="City" class="form-control">
                <option value="">Усі</option>
                @foreach (var city in Model.Cities)
                {
                    <option value="@city">@city</option>
                }
            </select>
        </div>
        <div class="form-group col-md-3">
            <label asp-for="SearchTerm"></label>
            <input asp-for="SearchTerm" class="form-control" placeholder="Пошук">
        </div>
        <div class="form-group col-md-2">
            <label asp-for="Sorting"></label>
            <select asp-for="Sorting" class="form-control">
                <option value="0">Ціна за зростанням</option>
                <option value="1">Ціна за спаданням</option>
                <option value="2">Останні додані</option>
                <option value="3">Найстаріші</option>
            </select>
        </div>
        <div class="col-md-1 d-flex align-items-end">
            <input type="submit" value="Пошук" class="btn btn-success w-100" />
        </div>
    </div>
</form>

```

Рис. 3.8. Код головної сторінки для пошуку житла

3.3. Інтерфейс додатку

Користувач розпочинає взаємодію з сайтом з головної сторінки, де він може перейти до доступного житла та авіа рейсів. На рисунку 3.9 показано головну сторінку.

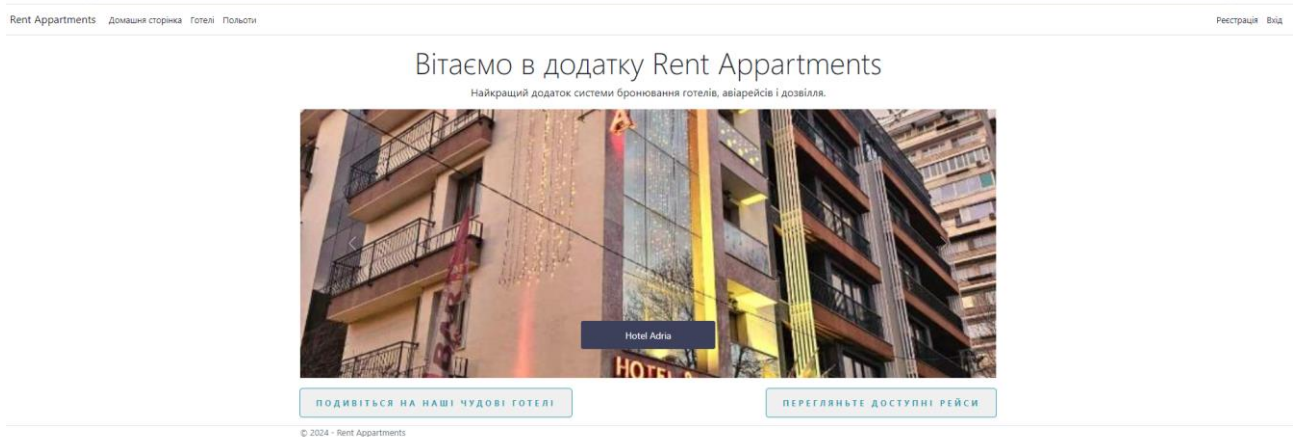


Рис 3.9. Головна сторінка сайту

Також на головній сторінці користувач може перейти до входу в систему, або ж зареєструватися.

The image shows a login window titled 'Увійти' (Login). The main heading is 'Використайте обліковий запис для входу' (Use your account to log in). There are two input fields: 'Електронна пошта' (Email) with the value 'admin@mail.com' and 'Пароль' (Password) with masked characters. Below the password field is a checkbox labeled 'Запам'ятати мене?' (Remember me?). A blue button labeled 'Увійти' (Login) is positioned below the form. At the bottom left, there is a blue link labeled 'Реєстрація' (Registration).

Рис. 3.10. Вікно входу в систему

У випадку якщо у користувача ще немає акаунту, він може його зареєструвати перейшовши на сторінку реєстрації, натиснувши кнопку Реєстрація. Вікно реєстрації зображено на рисунку 3.11.

Реєстрація

Створити новий акаунт.

Рис. 3.11. Вікно Реєстрації


Після вибору локації, та сортування, нам буде відображено доступне житло по обраним параметрам. Також додатково присутній короткий опис. Можемо обрати житло, яке найбільше підходить (рис. 3.12).

Готелі

Місто

Пошук за описом

Сортувати за




**Ibis Paris Gare
Montparnasse Catalogne**

★★★★

[Париж](#)

Кімнати: Single, Double, Triple,
Apartment




Hôtel Saint Marcel

★★★★

[Париж](#)

Кімнати: Single, Double



Kraft Hotel

★★★★

[Париж](#)

Кімнати: Single, Double

Сторінка 1 з 1

Рис. 3.12. Вибір житла

На сторінці бронювання житла відображається вся необхідна інформація для здійснення бронювання та перегляду деталей про житло.


У верхній частині сторінки вказана назва готелю ("Ibis Paris Gare Montparnasse Catalogne") з зірковим рейтингом і кнопкою "Зарезервувати" для переходу до оформлення бронювання. Адреса готелю вказана під назвою, щоб

користувачі могли легко знайти його розташування. Ліворуч знаходиться велика фотографія готелю, яка дає уявлення про інтер'єр та зручності. Під фотографією розміщений детальний опис готелю, включаючи відстань до відомих пам'яток, наявність Wi-Fi, кондиціонерів, парковки.

Праворуч від фотографії знаходиться розділ з таблицею "Доступно", в якій перераховані типи кімнат, максимальна кількість гостей і ціна за ніч. Весь описаний функціонал зображено на рисунку 3.13. Також на сторінці житла зображено доступне дозвілля в даній локації (рис. 3.14).

Ibis Paris Gare Montparnasse Catalogne Зарезервувати

11 Rue du Texel, 14-й округ, 75014, Париж, Франція, Париж, Франція



Ibis Париж Гарє Монпарнас Каталонія зручно розташований в 14-му окрузі Парижа, відстань до Роденового музею, Люксембурзького саду становить 1,6 миль, а до Музею Орсе - 2 миль. У готелі наявний бар, а номери з кондиціонером обладнані безкоштовним WiFi і власною ванною кімнатою. Приватна парковка доступна на території готелю. У номерах готелю є шафа. Також вони мають письмовий стіл та плоскі телевізори, а деякі номери мають балкон. Гості можуть снідати за буфетом, континентальним або американським сніданком кожен ранок. Багатомовний персонал готелю, який володіє англійською, іспанською, французькою та українською мовами, готовий допомогти цілодобово на рецепції. Поблизу є такі визначні місця, як Свята Шапель (на відстані 2 миль) та Парижська Експо - Порт Версаль (2,1 миль). Найближчий аеропорт - аеропорт Париж-Орлі, віддалений на 8,1 миль від Ibis Париж Гарє Монпарнас Каталонія.

← Попередній готель
Наступний готель →

Повернутися до усіх готелів

Доступно


Тип кімнати	Кількість гостей	Ціна за ніч
Single		\$ 165,00
Only 5 rooms left on our site		
Double		\$ 302,00
Only 5 rooms left on our site		
Triple		\$ 578,00
Only 5 rooms left on our site		
Apartment		\$ 630,00
Only 5 rooms left on our site		

Правила готелю


Ibis Paris Gare Montparnasse Catalogne приймає особливі побажання! Додайте їх на наступному кроці!

- Заїзд** З 16:00
- Візд** По 12:00
- Тварини** Розміщення домашніх тварин допускається за запитом. Може стягуватися плата.
- Паркінг** До послуг безкоштовна автостоянка.

Визначні місця в Париж ...



[Ейфелева Вежа](#)
Париж
від \$40,00 з дорослого



[Лувр](#)
Париж
від \$46,46 з дорослого

Рис. 3.13. Робота на сторінці бронювання житла

Визначні місця в Париж ...

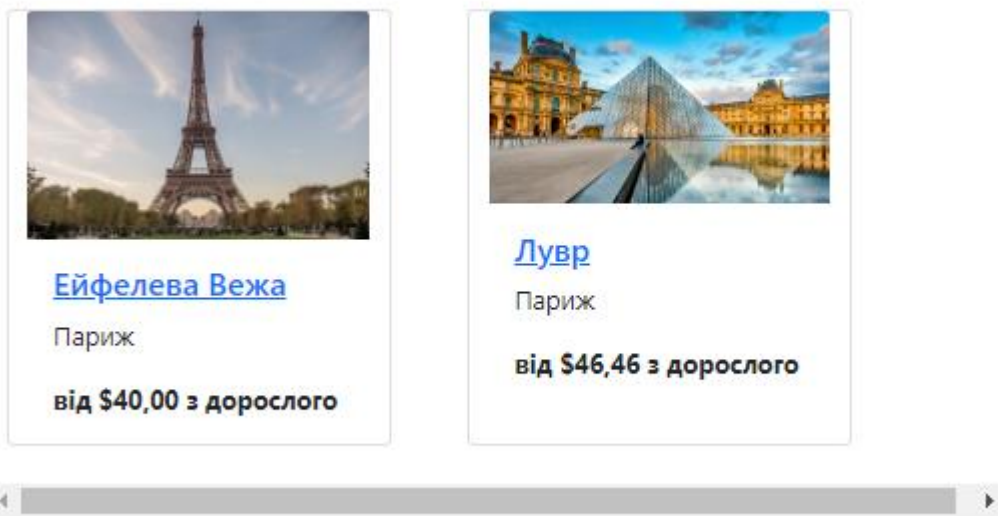


Рис. 3.14. Доступне дозвілля

На сторінці підтвердження бронювання відображено деталі бронювання, дати та ціна. Доступний функціонал підтвердження та оплати онлайн (рис. 3.15, 3.16).

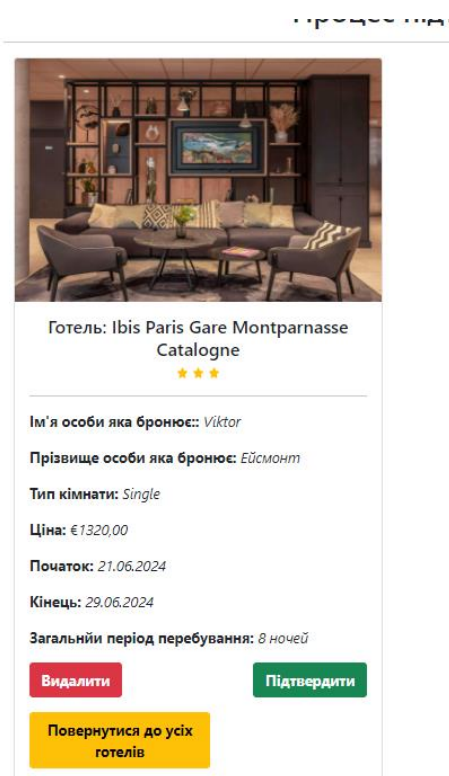


Рис. 3.15. Сторінка підтвердження бронювання

The screenshot shows a 'Payment Details' form with a 'Remember' checkbox. The 'CARD NUMBER' field contains 'Valid Card Number' and has a lock icon. Below it are 'EXPIRY DATE' and 'CV CODE' fields. The 'EXPIRY DATE' has two input boxes, both containing '0'. The 'CV CODE' has an input box containing 'CV'. Below the form is a blue button labeled 'Final Payment USD 1320,00' and a large green button labeled 'Pay'.

Рис. 3.16. Сторінка оплати онлайн

На сторінці з доступними рейсами, можна обрати потрібний пункт відправлення та прибуття, відсортувати за потрібними параметрами та обрати рейс (рис. 3.17).

The screenshot shows a flight search interface titled 'Усі польоти'. It has three search filters: 'Пункт відправлення' (Софія), 'Пункт прибуття' (Мадрид), and 'Сортувати за' (Ціна за зростанням). A green 'Пошук' button is next to the sorting filter. Below the filters is a flight card for Wizz Air. The card shows the flight details: Wizz Air logo, 06:40 departure from SOF, 235 хв duration, 09:35 arrival at MAD. Below the flight details, it says 'Загальна сума: \$91,00' and 'Включає податки та збори'. A blue button labeled 'Детальніше >' is next to the price. At the bottom of the card are navigation arrows '<<' and '>>'. Below the card is the text 'Сторінка 1 з 1'. At the very bottom, there is a copyright notice: '© 2024 - Rent Apartments'.

Рис. 3.17. Сторінка з усіма польотами

На сторінці деталей польоту можна переглянути деталі та забронювати рейс. (рис. 3.18).

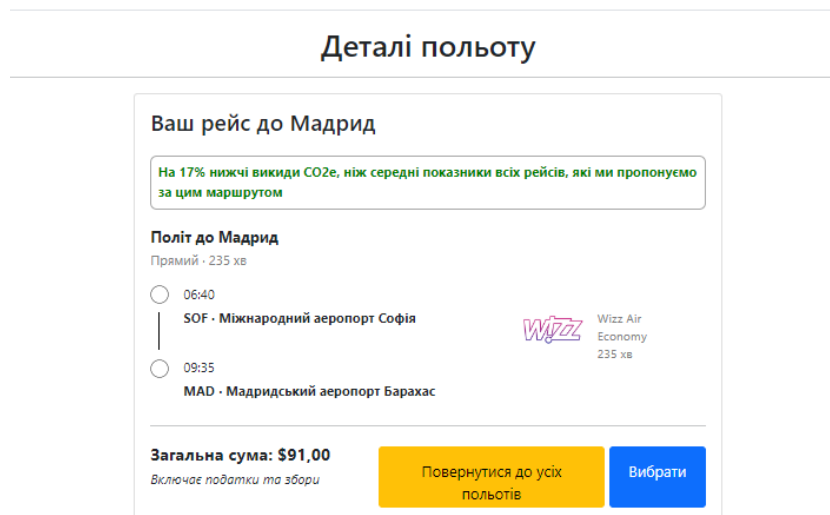


Рис. 3.18. Сторінка деталей польоту.

Підтвердження та оплата реалізована по аналогії з житлом.

Головна сторінка адміністратора виглядає так, як показано на рисунку 3.19. Доступний функціонал: додавання житла, польотів, дозвілля, та керування користувачів. Також є можливість змінювати деталі житла (рис. 3.20).

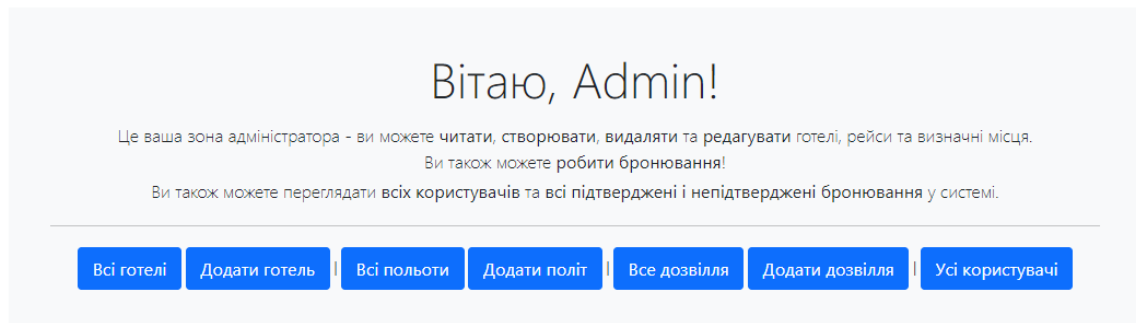


Рис 3.18. Головна сторінка адміністратора

Готелі

Місто

Пошук за описом

Сортувати за

Пошук

Sotetsu Fresa Inn
Yokohama Higashiguchi

★★★★

[Токіо](#)

Кімнати:

Змінити
Видалити

[Перевірити доступність >](#)

Hotel Adria

★★★★

[Софія](#)

Кімнати:

Змінити
Видалити

[Перевірити доступність >](#)

Hotel ZOO Sofia - Secured Paid Parking

★★★★

[Софія](#)

Кімнати: Single, Double, Triple

Змінити
Видалити

[Перевірити доступність >](#)

Eurostars Sofia City

★★★★

[Софія](#)

Кімнати: Single, Double, Triple, Apartment

Змінити
Видалити

[Перевірити доступність >](#)

Emin Palace

★★★★

[Стамбул](#)

Кімнати: Single, Double, Triple

Змінити
Видалити

[Перевірити доступність >](#)

Ad maiora suite

★★★★

[Рим](#)

Кімнати: Double, Triple, Apartment

Змінити
Видалити

[Перевірити доступність >](#)

<<
Сторінка 1 з 4
>>

© 2024 - Rent Apartments

Рис. 3.19. Головна сторінка пошуку житла зі сторони адміністратора

ВИСНОВКИ

В кваліфікаційній роботі було розглянуто проектування та реалізацію веб-застосунку для оренди житла, бронювання рейсів та організації дозвілля. Основною метою було створення зручного, функціонального та інтуїтивно зрозумілого інструменту, який забезпечує користувачам можливість легко знайти і забронювати необхідні послуги.

На початкових етапах роботи було виконано аналіз вимог до системи, що дозволило чітко визначити ключові функціональні можливості застосунку. Основні функції включали можливість реєстрації та аутентифікації користувачів, перегляд та бронювання житла, рейсів і дозвілля, а також управління бронюваннями.

Було проведено проектування архітектури веб-застосунку з використанням сучасних технологій ASP.NET MVC, MS SQL та Entity Framework. Це дозволило забезпечити високу продуктивність, масштабованість та надійність системи. Архітектура включала чітке розділення на фронтенд та бекенд частини, що сприяло зручності розробки та підтримки системи.

Реалізація застосунку включала налаштування служб та підключення до бази даних, створення моделей даних, контролерів та представлень. Важливу роль у проекті відігравала база даних, для якої було створено структуру з урахуванням усіх необхідних зв'язків між сутностями.

Особлива увага приділялась забезпеченню безпеки системи, включаючи захист даних користувачів та забезпечення коректного функціонування механізмів аутентифікації та авторизації.

Протягом реалізації проекту були використані різні методи тестування, що дозволило виявити та виправити помилки на ранніх етапах розробки. Застосунок пройшов всі необхідні тести і відповідає вимогам до якості програмного забезпечення.

Таким чином, розроблений веб-застосунок повністю відповідає поставленим вимогам і забезпечує користувачам можливість зручно та швидко орендувати житло, бронювати рейси та організувати дозвілля. Впроваджені технології та підходи дозволяють легко масштабувати систему, додавати нові функції та забезпечувати високу якість обслуговування користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Asp.Net MVC офіційна документація - <https://learn.microsoft.com/uk-ua/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-5.0&tabs=visual-studio>
2. Entity Framework офіційна документація - <https://learn.microsoft.com/en-us/ef/core/>
3. Ms Sql документація - <https://www.w3schools.com/sql/>
4. Freeman, A., & Sanderson, A. (2014). Pro ASP.NET MVC 5 (Expert's Voice in ASP.Net). Apress.
5. Ritcher, J. (2014). CLR via C#. Microsoft Press.
6. Dewson, R. (2012). Beginning SQL Server for Developers. Apress.
7. Delorme, G. (2016). Entity Framework Core Cookbook. Packt Publishing.