

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем**

«До захисту допущено»

Завідувач кафедри

_____ Євгенія СУЛЕМА

«__» _____ 2024 р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інженерія програмного
забезпечення мультимедійних та інформаційно-пошукових систем»**

**спеціальності 121 Інженерія програмного забезпечення
на тему: «Вебзастосунок для пошуку та резервування послуг»**

Виконала:

студентка IV курсу, групи КП-03

Сіренко Вікторія Юріївна _____

Керівник:

Доцент кафедри ПЗКС, к.т.н., доцент,

Заболотня Тетяна Миколаївна _____

Консультант з нормоконтролю:

Доцент кафедри ПЗКС, к.т.н., доцент,

Онай Микола Володимирович _____

Рецензент:

Старший викладач кафедри ПМА ФПМ,

Мальчиков Володимир Вікторович _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студентка _____

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Євгенія СУЛЕМА

«__» _____ 2023 р.

ЗАВДАННЯ

на дипломний проєкт студенту

Сіренко Вікторії Юріївні

1. Тема проєкту «Вебзастосунок для пошуку та резервування послуг», керівник проєкту Заболотня Тетяна Миколаївна, к.т.н., доцент, затверджені наказом по університету від «30» травня 2024 р. №2205-с.
2. Термін подання студентом проєкту «13» червня 2024 р.
3. Вихідні дані до проєкту: див. Технічне завдання.
4. Зміст пояснювальної записки:
 - огляд існуючих програмних рішень;
 - обґрунтування вибору засобів розроблення;
 - структурно-алгоритмічна організація розробленого вебзастосунку;
 - особливості програмної реалізації вебзастосунку.

5. Перелік обов'язкового графічного матеріалу:

- структура бази даних (креслення);
- алгоритм пошуку вільних часових проміжків послуги для обраної дати (креслення);
- архітектура вебзастосунку (плакат);
- діаграма варіантів використання вебзастосунку (плакат).

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Онай М.В., доцент		

7. Дата видачі завдання «31» жовтня 2023 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	13.11.2023	
2.	Розроблення та узгодження технічного завдання	25.11.2023	
3.	Розроблення структури вебзастосунку	15.12.2023	
4.	Підготовка матеріалів першого розділу дипломного проєкту	30.12.2023	
5.	Розроблення дизайну сторінок та графічних елементів	03.02.2024	
6.	Підготовка матеріалів другого розділу дипломного проєкту	19.02.2024	
7.	Програмна реалізація вебзастосунку	10.03.2024	
8.	Тестування вебзастосунку	17.03.2024	
9.	Підготовка матеріалів третього розділу дипломного проєкту	30.03.2024	
10.	Підготовка матеріалів четвертого розділу дипломного проєкту	12.04.2024	
11.	Підготовка графічної частини дипломного проєкту	21.04.2024	
12.	Оформлення документації дипломного проєкту	26.05.2024	

Студент

Вікторія СІРЕНКО

Керівник проєкту

Тетяна ЗАБОЛОТНЯ

АНОТАЦІЯ

Даний дипломний проєкт присвячений створенню програмного забезпечення для пошуку та резервування послуг.

Програмне забезпечення реалізовано у вигляді вебсайту, який має забезпечити підтримку пошуку та резервування послуг зі зручним, інтуїтивно зрозумілим інтерфейсом користувача, з детальними фільтрами пошуку та легкою процедурою зміни бронювань, а також підтримкою кількох мов.

Основною метою розробки було надання користувачам змоги знаходити інформацію про необхідні послуги та здійснювати їх бронювання по всьому світу.

Користуватись додатком можуть як приватні особи, що шукають різноманітні послуги для особистих потреб, так і приватні підприємці та невеликі бізнеси, які прагнуть полегшити процес бронювання та управління запропонованими послугами.

Клієнтська частина цього вебзастосунку була розроблена за допомогою мови програмування JavaScript, а основою для розробки інтерфейсу послужила бібліотека React. Це вибір сприяв забезпеченню високої адаптивності та швидкодії користувацького інтерфейсу, що важливо для підтримки безперебійної взаємодії з користувачами. Серверна частина була реалізована з використанням мови Python, яка відома своєю потужністю і гнучкістю в обробці серверних запитів, що забезпечує ефективну взаємодію з клієнтською частиною. Для зберігання і керування даними була обрана база даних MongoDB. Цей вибір обумовлений високою швидкістю обробки даних та надійністю, що є критично важливим для забезпечення стабільності та продуктивності системи в цілому.

ABSTRACT

This diploma project is dedicated to creating software for searching and booking services.

The software is implemented as a website, which aims to provide support for service search and booking with a convenient, intuitive user interface, detailed search filters, an easy booking modification procedure, and multilingual support.

The primary goal of the development was to enable users to find information about necessary services and make reservations worldwide.

Both private individuals seeking various services for personal needs and private entrepreneurs and small businesses looking to streamline the booking and management of their offered services can use the application.

The client side of this web application was developed using the JavaScript programming language, with the React library serving as the basis for the interface development. This choice facilitated high adaptability and speed of the user interface, which is crucial for maintaining seamless interaction with users. The server side was implemented using Python, known for its power and flexibility in processing server requests, ensuring effective interaction with the client side. MongoDB was chosen for data storage and management. This choice is dictated by the high speed of data processing and reliability, which are critically important for ensuring the stability and productivity of the system as a whole.

Позначення	Найменування	Кіл-ть	Примітка
ДП.045440-07-99	Вебзастосунок для	1	
	пошуку та резервування		
	послуг. Алгоритм		
	пошуку вільних часових		
	проміжків послуги для		
	обраної дати. Блок-схема		
	алгоритму		
ДП.045440-08-98	Вебзастосунок для	1	
	пошуку та резервування		
	послуг. Компакт		
	диск		

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Євгенія СУЛЕМА

“ ___ ” _____ 2023 р.

**ВЕБЗАСТОСУНОК ДЛЯ ПОШУКУ ТА
РЕЗЕРВУВАННЯ ПОСЛУГ**

Технічне завдання

ДП.045440-02-91

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Тетяна ЗАБОЛОТНЯ

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Вікторія СІРЕНКО

2023

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ	3
2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ.....	3
3. ПРИЗНАЧЕННЯ РОЗРОБКИ	3
4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ	3
5. ВИМОГИ ДО ПРОЄКТНОЇ ДОКУМЕНТАЦІЇ.....	4
6. ЕТАПИ ПРОЄКТУВАННЯ	5
7. ПОРЯДОК ТЕСТУВАННЯ РОЗРОБКИ.....	5

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Вебзастосунок для пошуку та резервування послуг.

Галузь застосування: інформаційні технології, сфера послуг

2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ

Підставою для розроблення є завдання на дипломне проєктування, затверджене кафедрою програмного забезпечення комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім. Ігоря Сікорського).

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для користувачів, які мають на меті знайти та зарезервувати необхідні їм послуги (наприклад, послуги у сфері краси та здоров'я, ремонту автомобілів, юридичні послуги, побутові, транспортні, клінінгові та інші види послуг) в будь-якій частині світу. Також програма може використовуватись людьми, які надають певні послуги і хочуть мати зручну платформу для публікації інформації про свій сервіс та організації роботи за рахунок відслідковування даних про бронювання.

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Вебзастосунок повинен забезпечити реалізацію таких ключових можливостей як:

- 1) можливість створення облікового запису;
- 2) здатність створювати та видаляти резервування;

- 3) спроможність створення, редагування та видалення опису послуги;
- 4) можливість сортування інформації про послуги за різними критеріями;
- 5) можливість залишати коментарі про певний сервіс;
- 6) підтримка багатомовного інтерфейсу користувача;
- 7) можливість автоматичного оновлення вмісту вебсторінок в режимі реального часу.

5. ВИМОГИ ДО ПРОЄКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання проєкту повинна бути розроблена наступна документація:

- 1) пояснювальна записка;
- 2) програма та методика тестування;
- 3) керівництво користувача;
- 4) креслення:
 - «Пошук вільних часових проміжків послуги для обраної дати. Схеми алгоритму»;
 - «Структура бази даних. ERD-діаграма».

6. ЕТАПИ ПРОЄКТУВАННЯ

Вивчення літератури за тематикою роботи	16.11.2023
Розроблення та узгодження технічного завдання	27.11.2023
Розроблення структури вебзастосунку	15.12.2023
Розроблення дизайну сторінок та графічних елементів.....	03.02.2024
Програмна реалізація вебзастосунку	15.02.2024
Тестування вебзастосунку	07.04.2024
Підготовка матеріалів текстової частини проєкту	28.04.2024
Підготовка матеріалів графічної частини проєкту.....	12.05.2024
Оформлення технічної документації проєкту	25.05.2024

7. ПОРЯДОК ТЕСТУВАННЯ РОЗРОБКИ

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Євгенія СУЛЕМА

“ ____ ” _____ 2024 р.

ВЕБЗАСТОСУНОК ДЛЯ ПОШУКУ ТА
РЕЗЕРВУВАННЯ ПОСЛУГ
Пояснювальна записка

ДП.045440-03-81

“ПОГОДЖЕНО”

Керівник проекту:

_____ Тетяна ЗАБОЛОТНЯ

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Вікторія СІРЕНКО

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ	3
ВСТУП	5
1. ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ	6
1.1. Огляд проблеми, яка вирішується ПЗ	6
1.2. Аналіз існуючих рішень	8
1.3. Висновки до розділу 1	17
2. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБЛЕННЯ.....	19
2.1. Вибір мови програмування для реалізації серверної частини.....	19
2.2. Вибір технологій для реалізації клієнтської частини.....	26
2.3. Вибір СКБД.....	30
2.4. Висновки до розділу 2.....	35
3. СТРУКТУРНО-АЛГОРИТМІЧНА ОРГАНІЗАЦІЯ СИСТЕМИ	36
3.1. Опис структури застосунку.....	36
3.2. Структура бази даних вебзастосунку	38
3.3. Опис функціональних можливостей вебзастосунку.....	41
3.4. Алгоритм пошуку вільних часових проміжків для бронювання сервісу	46
3.5. Висновки до розділу 3	48
4. ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ВЕБЗАСТОСУНКУ	49
4.1. Додаткові інструменти, використані для програмної реалізації вебзастосунку	49
4.2. Опис реалізованого користувацького інтерфейсу	51
4.3. Особливості проведеного тестування	57
4.4. Рекомендації для подальшого вдосконалення	63
4.5. Висновки до розділу 4.....	64
ВИСНОВКИ.....	65
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	67
ДОДАТКИ.....	70

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

- Аутентифікація – процес перевірки особи користувача або системи;
- Авторизація – процес надання прав доступу до ресурсів після аутентифікації;
- Інтерфейс – засоби взаємодії між компонентами системи, такими як користувач та комп'ютер або дві програмні системи;
- ПЗ – програмне забезпечення;
- СКБД – Система Керування Базами Даних (програмне забезпечення для створення, управління та використання баз даних);
- СУБД – Система Управління Базами Даних (програмне забезпечення для управління базами даних);
- Шифрування – алгоритмічне (криптографічне) перетворення даних, яке виконується у посимвольній послідовності з метою одержання шифрованого тексту;
- API – Application Programming Interface (набір функцій та процедур для взаємодії між різними програмними компонентами);
- CORS – Cross-Origin Resource Sharing (механізм безпеки веббраузерів, що дозволяє контролювати доступ до ресурсів з різних доменів);
- CSS – Cascading Style Sheets (мова стилізації, що використовується для опису вигляду HTML-документів);
- DOM – Document Object Model (інтерфейс для роботи з HTML та XML-документами у вигляді об'єктів);
- GPS – Global Positioning System (система навігації, що використовує супутники для визначення місцезнаходження об'єктів на Землі);
- HTML – HyperText Markup Language (мова розмітки для створення вебсторінок);

HTTP – HyperText Transfer Protocol (протокол, що використовується для передачі даних в комп'ютерних мережах);

HTTPS – HyperText Transfer Protocol Secure (захищений варіант HTTP, що використовує SSL/TLS для шифрування даних);

JSON – JavaScript Object Notation (формат обміну даними, що базується на тексті та використовує структури об'єктів JavaScript);

JSX – JavaScript XML (розширення синтаксису JavaScript, що дозволяє використовувати XML-подібний синтаксис);

JVM – Java Virtual Machine (віртуальна машина, що виконує програми, написані на Java);

MVC – Model-View-Controller (архітектурний шаблон, який використовується під час проєктування та розроблення ПЗ);

NoSQL – Not Only SQL (тип баз даних, що не використовує реляційні моделі та SQL);

SQL – Structured Query Language (мова для управління та запитів до реляційних баз даних);

UI – User Interface (засоби взаємодії користувача з ПЗ);

WebSocket – протокол зв'язку, що забезпечує двосторонній інтерактивний зв'язок між веббраузером і сервером;

XML – eXtensible Markup Language (мова розмітки, що використовується для зберігання та передачі даних);

.NET – платформа для розроблення програмного забезпечення від Microsoft, що підтримує різні мови програмування.

ВСТУП

У сучасному світі, де темпи життя стрімко зростають, зручність і ефективність стають ключовими факторами вибору споживачів. Вебзастосунки, що дозволяють швидко знаходити, порівнювати та забронювати послуги, стають не тільки популярними, а й необхідними інструментами у повсякденному житті людей.

Важливим аспектом розвитку цифрових технологій є інтеграція технологічних рішень у повсякденне життя людей. Попит на онлайн-сервіси бронювання значно зростає у всьому світі, що зумовлено зручністю, персоналізацією та збільшенням доступу до Інтернету та мобільних пристроїв. Таким чином, створення вебзастосунку для пошуку та резервування послуг відповідає сучасним вимогам ринку, де користувачі прагнуть оптимізувати свій час та зусилля. Такі платформи не тільки спрощують вибір послуг, але й підвищують конкурентоспроможність постачальників, забезпечуючи їм можливість представити свої пропозиції.

Даний дипломний проєкт присвячений розробленню вебзастосунку, який дозволить користувачам шукати та резервувати послуги в різних сферах, таких як краса, здоров'я, освіта, відпочинок і багато інших.

Окрім цього проєкт спрямований на задоволення потреб широкого кола користувачів, надаючи їм можливість з легкістю управляти своїм часом і вибором послуг. Основна мета проєкту – створення інтуїтивно зрозумілого, зручного і функціонального вебзастосунку, який задовольнить всі потреби користувачів, які пов'язані з пошуком та резервуванням послуг.

Цей проєкт має велике практичне значення, адже результати розроблення можуть бути використані широким колом компаній та організацій, що надають послуги, забезпечуючи їм сучасний інструмент для залучення та обслуговування клієнтів.

1. ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ

У першому розділі даної кваліфікаційної роботи розглянемо проблеми, які виникають у користувачів при пошуку та бронюванні різноманітних послуг, а також проаналізуємо відповідні існуючі програмні рішення. Нижче наведено перелік завдань, які необхідно виконати в межах цього розділу:

1. Аналіз ринку та визначення проблем, з якими стикаються користувачі при пошуку та резервуванні послуг.
2. Дослідження найпопулярніших програмних застосунків, якими користуються люди для пошуку та резервування сервісів, визначення переваг та недоліків кожного з них.
3. Виконання порівняльного аналізу та побудова порівняльної таблиці результатів аналізу існуючих застосунків.
4. Формування розширеного переліку потрібної функціональності розроблюваного ПЗ.

1.1. Огляд проблеми, яка вирішується ПЗ

В повсякденному житті часто виникає ситуація, коли необхідно терміново скористатися послугами певного сервісу. В даному випадку, в першу чергу, користувачеві необхідно дізнатись про вільні місця та можливість їх бронювання. Окрім того, дуже важливо мати змогу фільтрувати дані про послуги за різними параметрами (наприклад, за категорією послуги, місцезнаходженням, вільною датою, тощо), щоб якнайшвидше знайти потрібний варіант. Також важливо розробити зручний та інтуїтивно зрозумілий інтерфейс, оскільки більшість існуючого програмного забезпечення у даній галузі має проблеми з елементами управління програмою.

Також до проблем, які виникають у користувачів різноманітних сервісів, треба віднести проблему із можливістю відміни чи перенесення

бронювання. В більшості випадків необхідно виконати декілька кроків, щоб мати змогу редагувати бронювання, внаслідок чого користувачі можуть зіткнутися з певними труднощами.

Додатково може виникнути потреба в підтримці вебзастосунком кількох мов для вільного ознайомлення з всіма доступними послугами навіть за кордоном.

Всі означені проблеми підкреслюють необхідність розроблення нового програмного забезпечення для підтримки пошуку та резервування послуг зі зручним, інтуїтивно зрозумілим інтерфейсом користувача, з детальними фільтрами пошуку та легкою процедурою зміни бронювань, а також підтримкою кількох мов інтерфейсу користувача.

Таким чином, метою даного дипломного проекту є розроблення програмного рішення для надання користувачам змоги знаходити інформацію про необхідні послуги та здійснювати їх бронювання по всьому світу.

Нижче наведено перелік завдань, які потрібно вирішити під час роботи над дипломним проектом, щоб досягти поставленої мети:

1. Аналіз ринку ПЗ для пошуку та бронювання послуг.
2. Порівняння додатків-конкурентів, виявлення їхніх сильних та слабких сторін.
3. Визначення функціональних можливостей програмного продукту, яких потребують потенційні користувачі.
4. Підготовка дизайну вебзастосунку.
5. Обґрунтування вибору стеку технологій для розроблення ПЗ.
6. Програмна реалізація вебзастосунку для пошуку та бронювання послуг.
7. Тестування створеного програмного забезпечення.

1.2. Аналіз існуючих рішень

Проведення аналізу існуючих рішень може допомогти визначити найкращий варіант дій, зменшити ризики та зекономити час і кошти, а також дає змогу розробити стратегію дій, засновану на сильних і слабких сторонах організацій [1].

Початковим кроком у процесі аналізу є підготовка до нього, яка включає в себе знаходження застосунків, що відповідають основним потребам потенційних користувачів. На даний момент програмних продуктів, які одночасно орієнтовані на пошук та бронювання послуг, не знайшлося, але є рішення дуже близькі за функціональністю. Серед таких рішень можемо виділити Booking. com [2], Google Maps [3] та OLX [4].

Визначимо головні вимоги для об'єктивного порівняння функціональності вищеписаних програмних рішень:

1. Можливість створення облікового запису.
2. Можливість сортувати послуги за різними критеріями.
3. Здатність створювати та видаляти бронювання за мінімальну кількість часу.
4. Спроможність створення, редагування та видалення опису послуги.
5. Можливість залишати коментарі про певний сервіс.
6. Захищеність даних від випадкового або навмисного видалення іншими користувачами.
7. Багатомовна підтримка.

Розглянемо вищезгадані програмні рішення більш детально відповідно до цього переліку вимог.

1.2.1. Booking. com

Booking.com [2] – це онлайн-сервіс для бронювання готелів, хостелів та інших помешкань для подорожей.

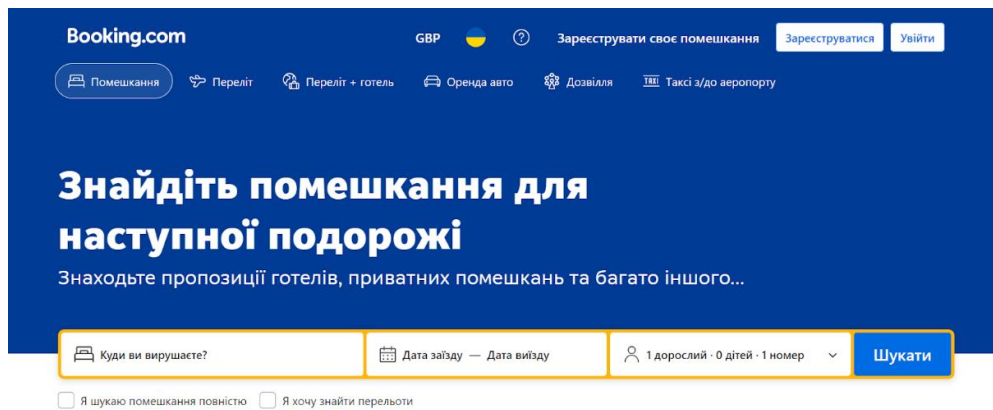


Рис. 1.1. Фрагмент головної сторінки Booking.com

На головній сторінці сайту є можливість обрати місто, в якому потрібно забронювати помешкання, дату перебування та кількість людей і номерів. Після введення необхідної інформації та натиснення кнопки “Шукати” сайт пропонує користувачеві варіанти помешкань, які задовольняють початковим критеріям. Надалі маємо можливість додати фільтри для знаходження бажаного варіанту.

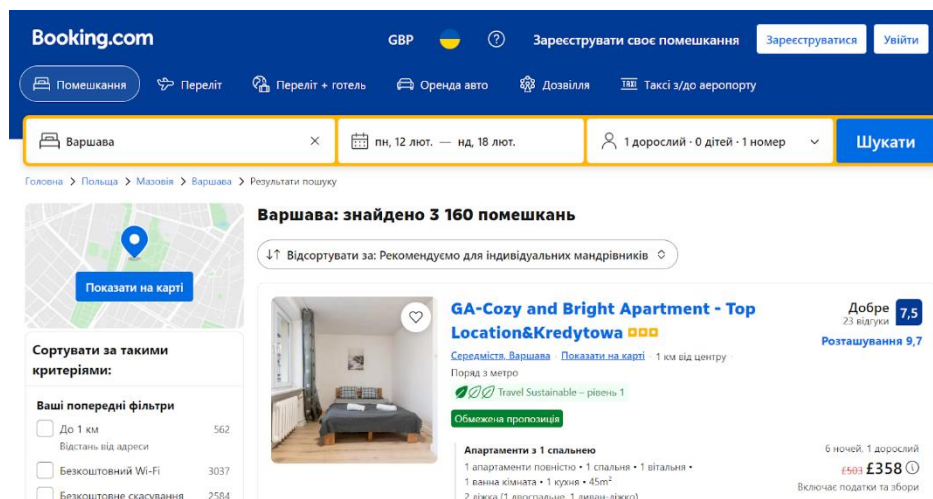


Рис. 1.2. Фрагмент сторінки Booking.com з фільтрами та варіантами бронювання

При натисненні назви помешкання можна переглянути детальну інформацію про нього, переглянути наявні місця і безпосередньо

забронювати апартаменти. Для бронювання необхідно надати персональну інформацію.

В цілому, Booking.com має багато корисних функцій, які допомагають користувачам знайти найкращі пропозиції для своєї подорожі. Ось деякі переваги [5]:

1. Пошук за категоріями: користувачі можуть шукати помешкання за різними категоріями, такими як тип помешкання, кількість зірок, відстань від центру міста, відгуки користувачів та багато іншого.
2. Перегляд помешкань на карті: користувачі можуть переглядати мапу, щоб знайти помешкання в певному районі.
3. Перегляд відгуків інших користувачів: користувачі можуть переглядати відгуки інших користувачів, щоб дізнатися більше про помешкання.
4. Гнучкі умови скасування: багато готелів на Booking.com пропонують гнучкі умови скасування бронювання, що може бути важливим для тих, хто планує подорож, але може зіткнутися з несподіваними змінами планів.

Але під час спроби бронювання були помічені такі недоліки:

1. Кожного разу при обранні одного фільтру сторінка оновлюється, і якщо користувач хоче обрати декілька фільтрів, йому доведеться кожного разу прокручувати сторінку вниз до того фільтру, де він зупинився, щоб далі додати наступні. Було б логічніше мати змогу обрати декілька фільтрів за раз та за допомогою кнопки застосувати всі фільтри, оновивши сторінку лише один раз.
2. При натисненні кнопки “Забронювати” немає можливості повернутися назад, навіть якщо не введено персональні дані і бронювання не завершено. Кожного разу при перегляді нової сторінки про апартаменти з’являється спливаюче вікно з

інформацією про незавершене бронювання, але відмінити це незавершене бронювання у користувача немає можливості.

Отже, Booking.com покриває більшу частину вимог, які необхідні для вдалого бронювання житла. Але, попри всі переваги, у даного застосунку наявні деякі недоліки, що пов'язані з елементами управління вебсайтом.

1.2.2. Google Maps

Google Maps – це онлайн-картографічний сервіс, розроблений компанією Google [3]. Сервіс надає можливість переглядати мапи, отримувати інформацію про місцевості, планувати маршрути, використовувати GPS-навігацію, дізнаватися про публічний транспорт, а також додавати відгуки та фотографії про різні локації.

Окрім всього вищезазначеного, за допомогою “Забронювати через Google” можна бронювати готелі, ресторани, авіаквитки, концертні квитки та інших послуги через різні партнерські платформи [6].

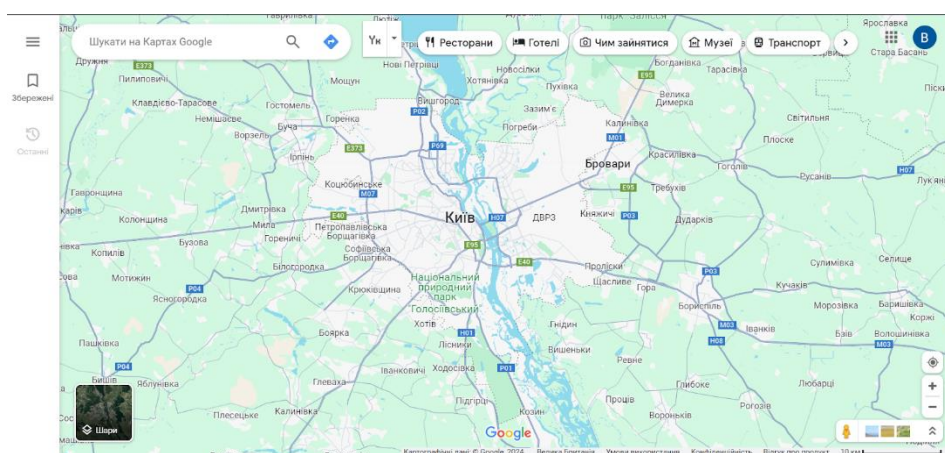


Рис. 1.3. Головна сторінка Google Maps

При пошуку на Google Maps, якщо партнери дозволяють бронювання через Google, то можна побачити опцію “Забронювати через Google”. Після вибору цієї опції відбувається перенаправлення на вебсайт партнера для завершення процесу бронювання.

Отже, головним недоліком при бронюванні через цей програмний продукт є те, що Google Maps не надає прямої можливості для бронювання помешкань чи подій в межах самої платформи. Ще одним недоліком є те, що доступність функції бронювання може залежати від регіону та партнерських угод, які укладено на момент користування.

Але варто зазначити і плюси використання Google Maps для створення бронювання.

Серед переваг даного рішення виділено:

1. Зручний пошук і відображення інформації: Google Maps надає зручний інтерфейс для пошуку готелів, ресторанів, туристичних об'єктів та інших місць. Є можливість пошуку місця за ключовими словами, переглядання відгуків та рейтингів.
2. Інтеграцію з партнерами: сервіс "Забронювати через Google" співпрацює з десятками компаній в Інтернеті та дозволяє резервувати місця, записуватись на послуги та вирішувати інші завдання.
3. Доступність візуальної інформації: користувачі можуть переглядати фотографії та відео різних місць, щоб отримати візуальне уявлення перед бронюванням.
4. Наявну інформацію про місцезнаходження: Google Maps надає докладні дані про розташування, години роботи, відгуки та іншу корисну інформацію, яка може допомогти при виборі та бронюванні місця.

Отже, Google Maps, незважаючи на те, що не є безпосередньою платформою для бронювання, пропонує користувачам зручний інтерфейс для пошуку та отримання інформації про різноманітні місця. Інтеграція з партнерами дозволяє переходити до сторінок бронювання за допомогою опції "Забронювати через Google". Переваги полягають у простоті користування та можливості швидко знаходити потрібну інформацію, але кінцеве бронювання відбувається поза межами самої платформи. Це робить

Google Maps зручним інструментом для планування подорожей та отримання детальної інформації про місця навколо, але за умови виконання додаткових кроків для завершення бронювання.

1.2.3. OLX

OLX [4] – це онлайн-платформа для оголошень, де користувачі можуть розміщувати оголошення про продаж товарів та послуг. Платформа пропонує можливість обміну та купівлі-продажу різноманітних товарів, таких як електроніка, нерухомість, транспортні засоби, предмети господарювання, одяг та багато іншого. OLX діє в різних країнах та надає можливість користувачам швидко та легко розміщувати свої оголошення та знаходити потрібні товари чи послуги.

Незважаючи на те, що цей програмний продукт не пов'язаний з бронюванням сервісів, ідея продукту дуже близька до того результату, який прагнемо отримати в результаті виконання даного дипломного проекту.

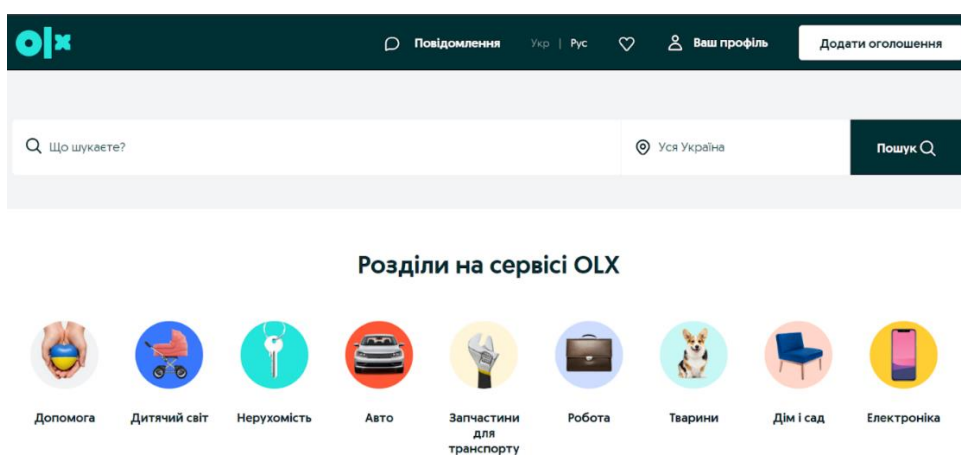


Рис. 1.4. Головна сторінка OLX

На головній сторінці програмного застосунку маємо можливість обрати місцевість та розділ товарів для полегшення пошуку. Крім того можемо використовувати вікно пошуку, щоб знайти товар за ключовими словами. Водночас можемо створити своє оголошення про продаж або обмін.

При виборі певної категорії та місцевості надалі присутні розширені фільтри для сортування товарів.

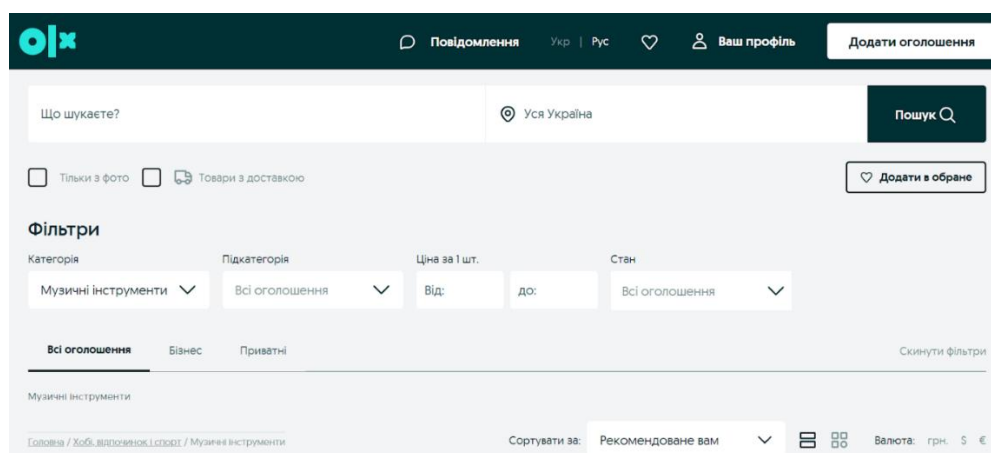


Рис. 1.5. Фрагмент сторінки OLX з фільтрами товарів

Серед переваг OLX можна зазначити :

1. Зручний інтерфейс: застосунок OLX надає зручний інтерфейс для користувачів, що робить процес розміщення та перегляду оголошень більш зручним.
2. Мобільний доступ: завдяки застосунку, користувачі можуть легко користуватися OLX на мобільних пристроях, що є зручним для тих, хто шукає або розміщує оголошення в дорозі.
3. Широкий асортимент категорій: OLX охоплює різні категорії товарів і послуг, що дозволяє користувачам знаходити або розміщувати оголошення на різний асортимент продукції.
4. Локалізацію оголошень: застосунок дозволяє користувачам легко знаходити товари чи послуги в своєму регіоні, сприяючи укладанню локальних угод.
5. Взаємодію між користувачами: OLX дозволяє продавцю та покупцям взаємодіяти, обговорювати деталі угод та домовлятися про умови покупки безпосередньо через платформу.

- б. Безкоштовне розміщення оголошень: один із ключових принципів OLX – це можливість безкоштовно розміщувати оголошення, що забезпечує доступність платформи для широкого кола користувачів.

Серед недоліків виділяється, що для кожної країни присутній окремий вебсайт з відсутністю можливості зміни мови. Було б краще розробити один сайт з можливістю обрання місцевості та функцією мовного вибору. Також деякі оголошення можуть бути застарілими чи неактуальними, що може призвести до вимушеної втрати часу для користувачів, які шукають конкретні товари чи послуги.

Отже, OLX надає зручну функціональність для користувачів, які шукають можливості покупки чи продажу різноманітних товарів. Застосунок дозволяє користувачам безкоштовно розміщувати оголошення, використовувати мобільний інтерфейс та взаємодіяти між собою для укладання угод. Однак, слід зазначити, що на OLX існує певний ризик зловживання та шахрайства, а також можливість зустріти застарілі чи неактуальні оголошення. Безпека та конфіденційність також можуть бути важливими аспектами під час здійснення угод на платформі.

1.2.4. Результати проведеного аналізу

Після аналізу програмних продуктів, які близькі за функціональністю до розроблюваного в рамках даного дипломного проєкту програмного продукту, можна сформулювати декілька висновків, що наведено нижче.

Все розглянуте програмне забезпечення має певні недоліки.

1. Booking.com має проблеми з користувацьким інтерфейсом.
2. Google Maps вимагає додаткових кроків для завершення бронювання, оскільки забезпечує інтеграцію з партнерами;
3. OLX має незалежні вебсайти для кожної країни з відсутністю функції мовного вибору.

В підрозділі 1.2 було сформовано перелік функціональних вимог, яким має відповідати розроблювальна система з пошуку та резервування послуг. Наступна таблиця містить результати порівняння сервісів за обраними критеріями оцінювання.

Таблиця 1.1

Порівняння існуючих програмних рішень

Функціональна вимога	Сервіс		
	Booking.com	Google Maps	OLX
Можливість створення облікового запису	+	+	+
Можливість сортувати послуги за різними критеріями	+	+/-	+/-
Здатність створювати та видаляти бронювання за мінімальну кількість часу	+/-	+/-	-
Спроможність створення, редагування та видалення опису послуги	+	+	+
Можливість залишати коментарі про певний сервіс	+	+	+
Захищеність даних від випадкового або навмисного видалення іншими користувачами	+	+	+
Багатомовна підтримка	+	+	-

Жодне з проаналізованих рішень не відповідає всім поставленим функціональним вимогам, які має виконувати вебзастосунок для пошуку та бронювання послуг. Тому сформуємо перелік можливостей, які має забезпечувати ПЗ:

1. Можливість сортування всіх послуг за певними категоріями. Враховуючи додавання спеціальної кнопки, щоб була можливість обрати декілька фільтрів для сортування послуг при цьому не оновлюючи сторінку кожного разу при обранні лише одного фільтру.
2. Здатність створювати та видаляти бронювання послуги за мінімальну кількість часу.
3. Спроможність створення, редагування та видалення опису послуги при наявності коректного доступу.
4. Можливість залишати коментарі про певний сервіс при наявності облікового запису.
5. Захищеність даних від випадкового або навмисного видалення іншими користувачами.
6. Підтримка функції мовного вибору.

1.3. Висновки до розділу 1

В першому розділі дипломного проекту успішно виконано наступні завдання:

1. Проаналізовано ринок ПЗ для пошуку та бронювання послуг.
2. Визначено головні перешкоди, з якими стикаються користувачі подібних програмних продуктів.
3. Розглянуто популярні ПЗ для пошуку та бронювання послуг.
4. Складено список критеріїв для порівняння систем у формі функціональних вимог.

5. Здійснено порівняльний аналіз існуючих програмних рішень, розглянуто переваги та недоліки кожного проаналізованого програмного продукту.
6. Зображено порівняльну таблицю результатів аналізу ПЗ.
7. Створено перелік функціональних можливостей, які має задовольняти програмна система, розроблювана в рамках виконання дипломного проєкту.

2. ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБЛЕННЯ

В даному розділі розглянуто основні існуючі засоби реалізації програмного забезпечення, їх характеристики, переваги та недоліки (складено порівняльну таблицю відповідно до обраних критеріїв оцінювання), а також проаналізовано доцільність їх використання для вирішення задач дипломного проєкту.

Вебзастосунок для пошуку та бронювання послуг розділений на дві частини: серверну та клієнтську. Таке розділення забезпечує більшу безпеку використання вебзастосунку, оскільки клієнтська частина не має прямого доступу до бази даних, що полегшує забезпечення безпеки системи. Крім того, це збільшує швидкість системи та зменшує навантаження на сервер.

Серверна частина має обробляти запити від клієнта, взаємодіяти з базою даних, містити бізнес-логіку та повертати результат. Натомість клієнтська частина відповідає за взаємодію з користувачем та передавання запитів від клієнта до серверної частини за допомогою протоколу HTTP.

База даних має забезпечувати ефективне зберігання, оновлення та вибірку інформації.

2.1. Вибір мови програмування для реалізації серверної частини

Визначимо, яким вимогам має задовольняти мова програмування при реалізації серверної частини для програмного продукту з пошуку та бронювання послуг:

1. Підтримка вебтехнологій , таких як HTTP-запити, WebSocket, і т.д. .
2. Великий вибір доступних бібліотек та фреймворків.
3. Наявність фреймворку для побудови вебсерверу.
4. Наявність модульності.
5. Здатність ефективно масштабуватися для оброблення зростаючого обсягу даних та запитів.

6. Наявність засобів для ефективного управління безпекою.
7. Підтримка асинхронності для ефективної роботи з багатьма одночасними запитами.
8. Можливість легко інтегруватися з різними системами управління базами даних.
9. Наявність великої кількості документації та ресурсів для вивчення.

2.1.1. Python

Python [7] – це інтерпретована об'єктно-орієнтована мова програмування високого рівня із суворою динамічною типізацією. Ця мова підтримує кілька парадигм програмування, зокрема: об'єктно-орієнтовану, процедурну, аспектно-орієнтовану та функціональну. Python має ефективні структури даних високого рівня та реалізує простий, але ефективний підхід до об'єктно-орієнтованого програмування. Також він є багатоплатформовою мовою програмування, що означає, що вона може працювати на різних операційних системах, таких як Windows, macOS, Linux та інші.

Спочатку розглянемо переваги Python [8]:

1. Простота: Python має простий синтаксис, що дозволяє швидко розробляти програми.
2. Багатофункціональність: Python підтримує кілька парадигм програмування, що дозволяє розробникам використовувати різні підходи до розв'язання проблем.
3. Великий вибір бібліотек та фреймворків: Python має велику кількість бібліотек та фреймворків, що дозволяє розробникам швидко створювати програми.
4. Високорівневі структури даних: Python має високорівневі структури даних, такі як списки, словники та кортежі, що дозволяє розробникам легко працювати з даними.

Незважаючи на те, що Python є потужною мовою програмування з багатьма перевагами, вона також має свої недоліки.

Розглянемо основні недоліки Python [8]:

1. Повільність: Python є інтерпретованою мовою програмування, що може призводити до повільної роботи програм.
2. Обмежена підтримка мультипоточності: Python має обмежену підтримку мультипоточності, що може призводити до проблем з ефективністю програм.
3. Суворі динамічна типізація: Python має сувору динамічну типізацію, що може призводити до проблем з безпекою програм.
4. Низька продуктивність: Python може мати низьку продуктивність для деяких видів завдань, таких як обробка великих обсягів даних.

2.1.2. C#

C# [9] – це об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, переваження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML.

C# широко використовується для створення програмного забезпечення різного призначення, такого як додатки для Windows, вебдодатки, ігри, мобільні додатки і багато іншого. Ця мова стала популярною серед розробників завдяки своїй ефективності і високому рівню підтримки від Microsoft.

Серед переваг C# можна виділити такі характеристики як [10]:

1. Безпека типів: C# має вбудовану систему типів, яка виявляє помилки під час компіляції, зменшуючи кількість помилок, які можуть виникнути в цілому під час розроблення.

2. Підтримка об'єктно-орієнтованої парадигми програмування: C# є повністю об'єктно-орієнтованою мовою програмування, що дозволяє розробникам створювати більш складні програми з різноманітною функціональністю.
3. Підтримка .NET: C# є основною мовою програмування для платформи .NET, що дозволяє розробникам створювати програмне забезпечення для різних платформ, таких як Windows, Linux та MacOS.
4. Швидка компіляція та виконання: C# має гарну швидкість компіляції та виконання, що робить його ефективною мовою для розроблення програмного забезпечення.
5. Автоматичне збирання сміття: у C# передбачено автоматичне збирання сміття, що означає, що розробники не повинні вручну керувати виділенням та звільненням пам'яті.

Але, не дивлячись на це, C# має свої недоліки, наприклад [10]:

1. Висока складність: C# може бути складною мовою програмування через велику кількість функцій та можливостей.
2. Залежність від .NET: розроблення програмного забезпечення на C# передбачає встановлення платформи .NET, що може бути проблемою для деяких користувачів.
3. Обмежена підтримка бібліотек: у C# обмежена підтримка бібліотек від третіх сторін порівняно з іншими мовами програмування, такими як Python та JavaScript.
4. Високі вимоги до ресурсів: деякі проекти, написані на C#, можуть вимагати більше ресурсів (наприклад, пам'яті та обчислювальної потужності), порівняно з іншими мовами програмування.

2.1.3. Java

Java [11] – це високорівнева об'єктно-орієнтована мова програмування та платформа для розроблення та виконання програмного забезпечення. Java – це універсальна мова програмування, спрямована на те, щоб бути простою, переносною та безпечною. Вона використовується для розроблення широкого спектру додатків, включаючи веб-додатки, мобільні додатки та корпоративне програмне забезпечення.

Серед переваг Java можна виділити наступні [12]:

1. Кросплатформеність: Java є високопортативною, що означає, що вона може працювати на різних платформах, включаючи Windows, Linux та MacOS.
2. Об'єктно-орієнтований підхід: Java – це повністю об'єктно-орієнтована мова програмування, що дозволяє розробникам створювати складні програми з більшою функціональністю.
3. Безпека: у Java вбудована модель безпеки, яка допомагає захищати від вірусів та іншого шкідливого програмного забезпечення.
4. Швидкість виконання: додатки на Java можуть працювати ефективно завдяки використанню віртуальної машини Java (JVM).

Однак, як і всі мови програмування, Java має свої недоліки [12], наприклад:

1. Необхідність залучення додаткових ресурсів: деякі додатки, написані на Java, можуть вимагати більше ресурсів, особливо в порівнянні з мовами, компільованими в нативний код.
2. Швидкість старту додатків: запуск додатків на Java може вимагати деякого часу, що може бути недоцільним для деяких додатків.
3. Не підходить для вузькоспеціалізованих додатків: для деяких областей розроблення, де важлива максимальна продуктивність

або низькі ресурсні витрати, Java може бути менш підходящою мовою.

4. Використання пам'яті: Java використовує більше пам'яті, ніж інші мови програмування, що може бути проблемою для додатків великого масштабу.
5. Обмежений доступ до баз даних: у Java обмежений доступ до баз даних порівняно з іншими мовами програмування, такими як Python і C#.

2.1.4. Результати проведеного аналізу

В підрозділі 2.1 визначено вимоги для розроблення серверної частини вебзастосунку для пошуку та бронювання послуг. Далі зображено порівняльну таблицю інструментів реалізації (Python, C#, Java) за попередньо сформульованими критеріями.

Таблиця 2.1

Порівняння мов програмування для реалізації серверної частини застосунку

Характеристика мови	Мова програмування		
	Python	C#	Java
Підтримка вебтехнологій, таких як HTTP-запити, WebSocket і т. д.	+	+	+
Великий вибір доступних бібліотек та фреймворків	+	+	+/-
Наявність фреймворку для побудови вебсерверу	+	+	+
Наявність модульності	+	+	+

Характеристика мови	Мова програмування		
	Python	C#	Java
Здатність ефективно масштабуватися для обробки зростаючого обсягу даних та запитів;	+	+	+
Наявність засобів для ефективного управління безпекою	+	+	+
Підтримка асинхронності для ефективної роботи з багатьма одночасними запитами	+	+/-	+/-
Можливість легко інтегруватися з різними системами управління базами даних	+	+	+/-
Наявність великої кількості документації та ресурсів для вивчення	+	+	+/-

В результаті порівняльного аналізу мов програмування виявлено, що Python задовольняє всім поставленим умовам для серверної частини програми. Завдяки великому вибору фреймворків та бібліотек, Python виявляється дуже гнучкою та зручною мовою для вебпрограмування. Ця мова ефективно використовується для побудови вебсерверів та має велику кількість інструментів для роботи з вебтехнологіями. Крім цього, має великий вибір інструментів для роботи з базами даних та забезпечує високий рівень безпеки.

Розглянувши мову програмування C#, можемо сказати, що вона також має високий рівень підтримки вебтехнологій та доступність фреймворків,

зокрема ASP.NET. А також вона здатна до ефективної інтеграції з базами даних та забезпечує добре вбудовані інструменти для управління безпекою. Але підтримка асинхронності може бути трошки меншою порівняно з іншими мовами.

Якщо говорити про Java, то ця мова є потужною мовою програмування для веброзроблення, зокрема завдяки фреймворкам, таким як Spring та Apache Tomcat. На додачу має великий вибір інструментів, які сприяють ефективному масштабуванню та модульності, але має недостатню кількість бібліотек та фреймворків для зручного використання. Ця мова підтримує асинхронні операції, але може бути менш гнучкою в цьому плані порівняно з іншими мовами.

Отже, враховуючи простоту, розширюваність та активну спільноту, Python є найбільш доцільним варіантом для написання серверної частини в порівнянні з іншими розглянутими мовами програмування.

2.2. Вибір технологій для реалізації клієнтської частини

Розглянемо найпопулярніші рішення для написання клієнтської частини застосунків, такі як React.js, Angular та Vue.js. В подальшому аналізі маємо на меті визначити оптимальну технологію, яка буде задовольняти наступні вимоги:

1. Підтримка крос-платформності.
2. Забезпечення швидкого розгортання та реалізації.
3. Висока продуктивність.
4. Наявність великого вибору бібліотек.
5. Підтримує відповідні заходи безпеки.

2.2.1. React.js

React.js [13] – це відкрита бібліотека фронтенду JavaScript для розроблення користувацьких інтерфейсів (UI) вебдодатків, вона широко використовується для створення динамічних та ефективних вебзастосунків.

Серед переваг React.js можна виділити [14]:

1. Наявність простого синтаксису та архітектури.
2. React дозволяє розбити вебсторінку на невеликі, самостійні компоненти, що робить розроблення, тестування та обслуговування коду більш простими та модульними.
3. Підвищення продуктивності за рахунок віртуального DOM React, який ефективно оновлює лише ті частини фактичного DOM, які потребують змін, мінімізуючи навантаження на продуктивність.
4. Наявність багатой екосистеми сторонніх інструментів та бібліотек.
5. Наявність крос-платформної функціональності.

До недоліків React.js можемо віднести [14]:

1. Швидкий розвиток React.js може ускладнювати вивчення технології та підтримку вже існуючих проєктів.
2. Відсутність детальної документації.

2.2.2. Angular

Angular [15] – це фреймворк для дизайну додатків та платформа для розроблення, створена для ефективного та складного створення односторінкових та багатосторінкових додатків.

Розглянемо переваги Angular [16]:

1. Наявна добре структурована архітектура з компонентами, сервісами та модулями.
2. Angular побудований на TypeScript, який додає статичну типізацію, інтерфейси та кращі інструменти. TypeScript виявляє помилки під час розроблення, що призводить до надійного коду.
3. Система DI Angular спрощує управління залежностями та сприяє модульному дизайну. Це покращує тестовість та повторне використання коду.

4. Забезпечує крос-платформне розроблення.

Тепер опишемо недоліки технології Angular [16]:

1. Синтаксис Angular може бути об'ємним, особливо порівняно з легкими бібліотеками, такими як React.
2. Час завантаження може бути довшим, особливо для невеликих проєктів.
3. Міграція між версіями Angular може виникати труднощі.

2.2.3. *Vue.js*

Vue.js [17] – це фреймворк JavaScript для створення користувацьких інтерфейсів. Він базується на стандартних мовах HTML, CSS та JavaScript і надає декларативну та компонентно-орієнтовану модель програмування, яка допомагає вам ефективно розробляти інтерфейси користувача, чи то прості, чи складні.

Визначимо сильні сторони Vue.js [18]:

1. Простий і зрозумілий синтаксис з можливістю використання HTML-подібних шаблонів.
2. Гнучка система для роботи з компонентами і директивами.
3. Добре підходить для розроблення невеликих і середніх проєктів, а також для поступового впровадження в існуючі вебсторінки.

Визначимо недоліки слабкі сторони Vue.js [18]:

1. Має меншу кількість готових рішень і бібліотек порівняно з Angular та React.
2. Має можливість виникнення проблем масштабованості великих проєктів зі зростанням складності.

2.2.4. *Результати проведеного аналізу*

Зобразимо таблицю для порівняння проаналізованих технологій для реалізації клієнтської частини за визначеними критеріями:

Порівняння технологій для реалізації клієнтської частини застосунку

Критерії порівняння	Технологія		
	React.js	Angular	Vue.js
Підтримка крос-платформності	+	+	+
Забезпечення швидкого розгортання та реалізації	+	+/-	+/-
Висока продуктивність	+	+/-	+
Наявність великого вибору бібліотек	+	+	+/-
Підтримка відповідних заходів безпеки	+	+	+

Проаналізувавши представлені технології, можемо зазначити, що React.js відповідає всім поставленим вимогам. Ця технологія відзначається простотою навчання та використання, завдяки компонентній архітектурі та використанню віртуального DOM для ефективного оновлення інтерфейсу. React сприяє розробленню застосунків із зручним управлінням станом, використовуючи єдиний об'єкт стану. Велика спільнота розробників та розширена екосистема, разом із підтримкою JSX, роблять React гарним вибором для створення вебдодатків.

Angular та Vue.js не задовольнили всім поставленим вимогам і тому не були обрані в якості технології для даного проєкту. Розглянувши основні недоліки, можемо сказати, що Angular-застосунки можуть бути більш об'ємними за розміром і це може впливати на час завантаження. Щодо Vue.js: його екосистема, хоча і активно розвивається, але все ще менша

порівняно з React, що може обмежити вибір сторонніх бібліотек та інструментів.

Отже, для створення інтерфейсу для пошуку та бронювання сервісів було обрано React.js.

2.3. Вибір СКБД

СКБД (система керування базами даних) [19] – це комплекс програмного забезпечення, який надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних.

Для порівняння було обрано три системи керування базами даних: MongoDB, MySQL та PostgreSQL.

Виділимо основні вимоги, яким має задовольняти СКБД, яку будемо використовувати для створення застосунку для пошуку та бронювання сервісів:

1. Наявність високої швидкодії та ефективності обробки при великих обсягах даних та високій навантаженості запитів.
2. Забезпечення потужними механізмами аутентифікації, авторизації та шифрування для захисту конфіденційності і цілісності даних.
3. Наявність автоматичної реплікації даних.
4. Наявність механізмів резервного копіювання та відновлення даних для запобігання втратам і забезпечення надійності.
5. Підтримка можливості створення та оптимізації індексів для прискорення виконання запитів на дані.

2.3.1. MongoDB

MongoDB [20] – це потужна система керування базами даних (СКБД), яка відноситься до класу NoSQL баз даних.

До основних переваг MongoDB можемо віднести [21]:

1. Можливість роботи як з структурованими, так і з неструктурованими даними.
2. Міцна підтримка різноманітних запитів.
3. Функціональність MongoDB дозволяє працювати з великими обсягами даних, легко розділяти та розподіляти бази даних для ефективного керування даними.
4. Можливість індексувати будь-яке поле в документі, покращуючи загальну продуктивність пошуку.
5. Підтримка динамічної схеми.

До основних недоліків MongoDB [21] можемо віднести те, що дана СУБД:

1. Не підтримує з'єднання як реляційні бази даних.
2. Не повністю відповідає принципам ACID [22].

2.3.2. MySQL

MySQL [23] – це відкрита реляційна система управління базами даних. Вона використовується для зберігання і керування даними у вебдодатках, програмному забезпеченні та інших сценаріях розвитку програмного забезпечення. MySQL дозволяє створювати, зчитувати, оновлювати та видаляти дані за допомогою мови структурованих запитів SQL.

Розглянемо переваги MySQL [24]:

1. Забезпечує надійний рівень захисту даних.
2. Забезпечує вищу продуктивність за допомогою використання відображень, тригерів та збережених процедур.
3. Має унікальну структуру зберігання даних.
4. Має високу масштабованість для обробки великого обсягу даних.
5. Ефективно управляє зв'язками між таблицями.

Далі розглянемо недоліки MySQL [24]:

1. Є менш ефективним у роботі з дуже великими базами даних порівняно з деякими іншими базами даних.
2. Має менше потужних інструментів для розроблення та налагодження.
3. Може бути схильним до втрати даних через неефективне керування транзакціями.
4. Не підтримує перевірку обмежень SQL, що обмежує деякі опції перевірки даних.

2.3.3. PostgreSQL

PostgreSQL [25] – це потужна об'єктно-реляційна система управління базами даних, яка відома своєю надійністю, розширюваністю та відкритістю. Вона відноситься до категорії реляційних баз даних і забезпечує зберігання та обробку даних у вигляді таблиць, які взаємодіють за допомогою SQL.

Опишемо переваги PostgreSQL [25]:

1. Є системою баз даних з відкритим кодом, що означає, що вона доступна для використання безкоштовно.
2. Є надійною та стабільною системою керування базами даних, може працювати з додатками великого масштабу та критичними навантаженнями.
3. PostgreSQL дотримується властивостей ACID забезпечуючи цілісність даних та консистентність навіть у складних транзакціях.
4. Дозволяє створювати власні типи даних, оператори та функції, таким чином забезпечуючи розширення функціональності.
5. Має можливість налаштовувати параметри конфігурації для оптимізації продуктивності.

Далі опишемо недоліки PostgreSQL:

1. Необхідно виділити додатковий час для ознайомлення та розуміння всіх функцій та технологій, які використовує PostgreSQL.
2. PostgreSQL може вимагати багато пам'яті, особливо при роботі з великими наборами даних.
3. Відсутні вбудовані інструменти реплікації.
4. У деяких випадках продуктивність запису PostgreSQL може бути повільнішою порівняно з іншими базами даних.
5. Обмежені інструменти графічного інтерфейсу користувача.

2.3.4. Результати проведеного аналізу

Проаналізувавши розглянуті СКБД, можемо сказати, що вони відповідають майже всім поставленим вимогам. Але було вирішено, що MongoDB найбільш підходить до даного проєкту.

MongoDB відповідає всім вимогам в повному обсязі. Основним недоліком є те, що ця система керування базами даних не повністю відповідає вимогам ACID та може бути менш ефективною у випадку, коли потрібно виконувати складні операції з'єднання даних.

Що стосується MySQL, то вона може мати обмеження у використанні великих обсягів даних та в управлінні складними операціями з'єднання даних, а також не забезпечує деякі передові функції, які доступні в інших системах керування базами даних.

Хоча PostgreSQL відома своєю надійністю та розширюваністю, вона може потребувати часу для освоєння її високорівневих функцій та адміністрування. Також серед недоліків можемо виділити високе споживання пам'яті та можливість повільної продуктивності запису у певних випадках.

Зобразимо порівняльну таблицю проаналізованих СКБД за визначеними критеріями.

Порівняння СКБД

Критерії порівняння	СКБД		
	MongoDB	MySQL	PostgreSQL
Наявність високої швидкодії та ефективності обробки при великих обсягах даних та високій навантаженості запитів	+	+/-	+/-
Забезпечення потужними механізмами аутентифікації, авторизації та шифрування для захисту конфіденційності і цілісності даних	+	+	+
Наявність можливості автоматичної реплікації даних	+	+	+/-
Наявність механізмів резервного копіювання та відновлення даних для запобігання втратам і забезпечення надійності	+	+	+
Підтримка можливості створення та оптимізації індексів для прискорення виконання запитів на дані	+	+	+

Отже, в результаті аналізу та порівняння найпопулярніших систем керування базами даних, для вебзастосунку з пошуку та бронювання сервісів прийнято рішення використовувати MongoDB.

2.4. Висновки до розділу 2

Під час виконання другого розділу пояснювальної записки було виконано ряд завдань:

1. Визначено основні вимоги до інструментів реалізації програмного продукту.
 - мову програмування Python вибрано для створення серверної частини;
 - технологію React.js вибрано для розроблення клієнтської частини застосунку;
 - MongoDB вибрано як систему керування базами даних.
2. Виокремлено найбільш популярні безкоштовні інструменти реалізації.
3. Проведено аналіз вибраних інструментів, розглянуті переваги та недоліки кожного з них.
4. Створено порівняльні таблиці за визначеними критеріями.
5. Описано висновки порівняння та вибрано один інструмент для реалізації серверної, клієнтської частини застосунку та СКБД.

3. СТРУКТУРНО-АЛГОРИТМІЧНА ОРГАНІЗАЦІЯ СИСТЕМИ

3.1. Опис структури застосунку

Вебзастосунок для пошуку та резервування послуг розроблено за допомогою клієнт-серверної архітектури, тобто взаємодія між комп'ютерними програмами розділена на два окремі процеси: клієнтський та серверний. Серверна частина зазвичай відповідає за оброблення даних, управління ресурсами, зберігання інформації та виконання складних обчислень. Клієнтська частина виконує запити до сервера, очікуючи на відповідь, окрім цього ця частина зосереджена на інтерфейсах користувача та взаємодії з користувачем. Взаємодія між сервером та клієнтом відбувається через мережу за допомогою протоколу HTTP.

Архітектура розроблювального вебзастосунку ґрунтується на класичному архітектурному шаблоні Model-View-Controller (MVC) [26], що допомагає організувати взаємодію між різними частинами системи через чітке розмежування відповідальностей. Ця архітектура включає три основні компоненти: модель, представлення та контролер.

1. Модель виконує критично важливу роль, оскільки є центральним вузлом для управління даними і втілення бізнес-логіки застосунку. Вона забезпечує механізми для доступу до бази даних, займається обробленням та зберіганням інформації. Компонент моделі включає класи для роботи з базами даних, об'єкти, які представляють доменні моделі.
2. Компонент представлення відіграє ключову роль у формуванні того, як користувач сприймає застосунок. Цей компонент відповідає за відображення даних у вигляді HTML, який потім відображається в браузері користувача. Представлення складається з різних шаблонів, компонентів для відображення даних та елементів інтерфейсу, які надають можливість

користувачам взаємодіяти з функціональністю вебзастосунку, забезпечуючи таким чином інтуїтивно зрозумілу взаємодію.

3. Контролер служить мостом між користувачем, моделлю та представленням. Він обробляє вхідні запити від користувачів, визначає, які дії потрібно виконати, і координує дані між моделлю та представленням. Завдяки контролеру, вебзастосунок може відповідати на запити користувача, виконувати обчислення та забезпечувати потрібні дані для відображення результатів.

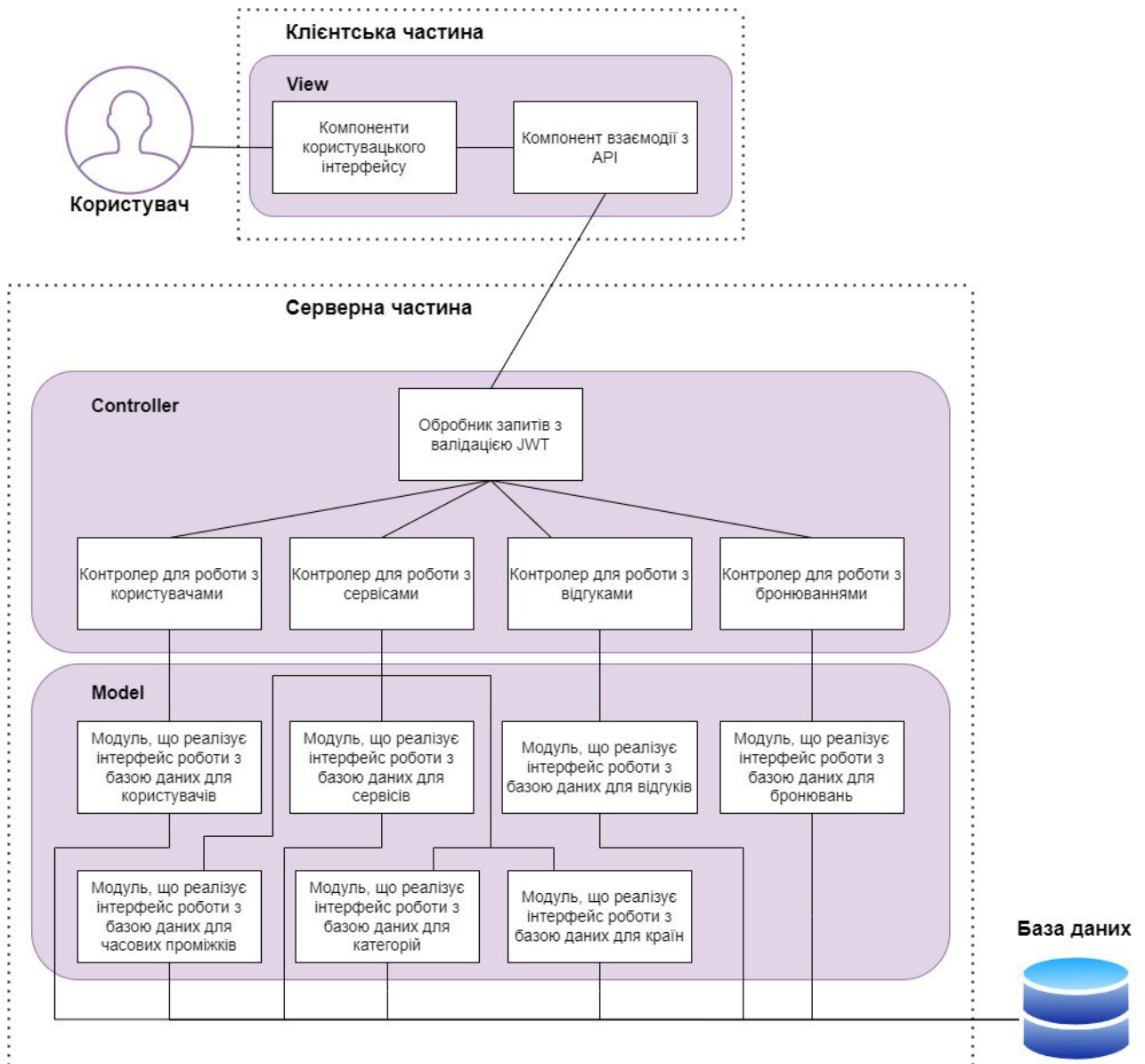


Рис. 3.1. Узагальнена архітектура вебзастосунку

Використання архітектури MVC допомагає встановити чіткий поділ обов'язків між компонентами системи, що, у свою чергу, робить код більш зрозумілим, спрощує його підтримку та розвиток. Цей метод також сприяє підвищенню адаптивності та можливості масштабування системи, дозволяючи їй ефективно реагувати на змінні потреби та вимоги користувачів, що є критично важливим в швидко змінюваному середовищі вебтехнологій.

3.2. Структура бази даних вебзастосунку

Для зберігання та відображення структурованих даних у програмному забезпеченні було вирішено використати нереляційну базу даних MongoDB, яка є оптимальним вибором, завдяки своїй гнучкості та масштабованості. MongoDB організує дані в колекції та документи, що дозволяє легко зберігати об'єкти JSON, забезпечуючи ефективну роботу зі структурами даних, які часто змінюються або не мають жорсткої схеми.

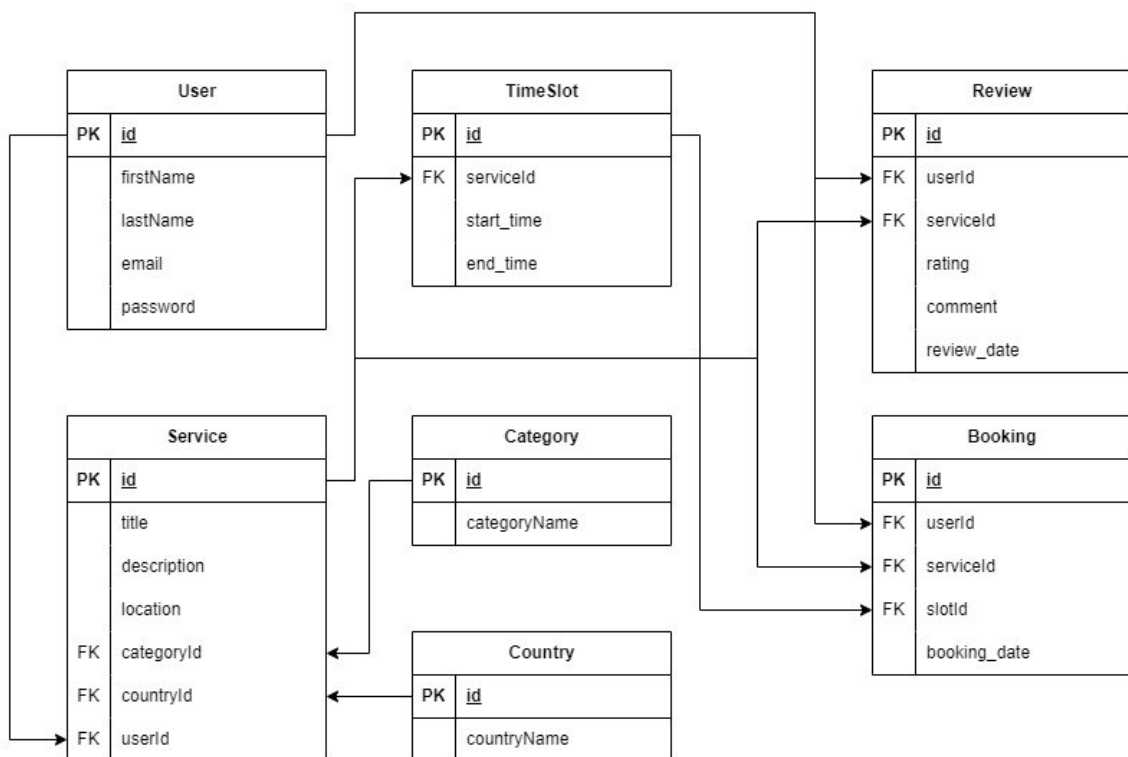


Рис. 3.2. Структурна схема бази даних застосунку

Сутність User (Користувач) використовується для зберігання та використання даних, які пов'язані з обліковим записом користувача:

- `_id`: унікальний ідентифікатор користувача;
- `firstName`: ім'я користувача;
- `lastName`: прізвище користувача;
- `email`: адреса електронної пошти користувача;
- `password`: пароль користувача, збережений в захешованому вигляді.

Сутність Service (Послуга) відображає дані послуги, яку створив користувач:

- `_id`: унікальний ідентифікатор послуги;
- `title`: заголовок послуги;
- `description`: опис послуги;
- `location`: місцезнаходження послуги;
- `categoryId`: дані про категорію до якої відноситься послуга, визначається об'єктом `Category`;
- `countryId`: дані про країну в якій базується послуга, визначається об'єктом `Country`;
- `userId`: дані про користувача, який створив послугу, визначається об'єктом `User`.

Сутність TimeSlot (Часовий проміжок) описує відрізок часу, протягом якого буде надана певна послуга:

- `_id`: унікальний ідентифікатор часового проміжку;
- `serviceId`: дані про сервіс до якого буде доданий часовий проміжок, визначається об'єктом `Service`;
- `start_time`: початок часового проміжку;
- `end_time`: кінець часового проміжку.

Сутність Category (Категорія) використовується для групування послуг за категоріями:

- `_id`: унікальний ідентифікатор категорії;
- `categoryName`: назва категорії.

Сутність Country (Країна) використовується для групування послуг за певною країною:

- `_id`: унікальний ідентифікатор країни;
- `countryName`: назва країни.

Сутність Review (Відгук) використовується для збереження та відображення відгуків про певну категорію:

- `_id`: унікальний ідентифікатор відгуку;
- `userId`: дані про користувача, який залишив відгук, визначається об'єктом User;
- `serviceId`: дані про послугу про яку залишають відгук, визначається об'єктом Service;
- `rating`: рейтинг послуги, який використовує шкалу від 1 до 5;
- `comment`: текст відгуку, який залишає користувач;
- `review_date`: дата створення відгуку.

Сутність Booking (Бронювання) використовується для зберігання даних про бронювання послуги:

- `_id`: унікальний ідентифікатор бронювання;
- `userId`: дані про користувача, який бронює послугу, визначається об'єктом User;
- `serviceId`: дані про послугу, яка бронюється, визначається об'єктом Service;
- `slotId`: дані про часовий проміжок, на який відбувається бронювання; визначається об'єктом TimeSlot;
- `booking_date`: дата, на яку бронюється послуга.

3.3. Опис функціональних вимог вебзастосунок

Створюваний вебзастосунок для пошуку та бронювання послуг володіє широким спектром функціональних вимог, а також надає користувачам інтуїтивно зрозумілий та зручний інтерфейс. Основні вимоги, яким задовольняє програма, є такими:

Таблиця 3.1

Функціональні вимоги вебзастосунок

Код вимоги	Опис вимоги
F-1	<p>Користувач має змогу зареєструватися в вебзастосунку. Для цього користувач має перейти на сторінку реєстрації та заповнити форму реєстрації:</p> <ul style="list-style-type: none">- ім'я;- прізвище;- електронну адресу;- пароль;- підтверджений пароль. <p>Якщо користувач не заповнить всі поля та / або введе некоректні дані, то має з'явитись відповідне повідомлення, яке вкаже користувачу, що необхідно змінити для успішної реєстрації.</p>
F-2	<p>Користувач повинен мати змогу увійти до свого облікового запису, ввівши у відповідну форму електронну пошту та пароль.</p> <p>Якщо дані, які користувач ввів є некоректними, то повинно з'явитись відповідне повідомлення для користувача.</p>
F-3	<p>Користувач повинен мати змогу здійснювати пошук послуги за її назвою.</p>

Код вимоги	Опис вимоги
F-4	<p>Користувач повинен мати змогу використовувати різні фільтри для вузькоспеціалізованого пошуку, такі як категорія, країна та дата.</p>
F-5	<p>Користувач повинен мати змогу створити сторінку власної послуги. Для цього необхідно перейти на відповідну сторінку та заповнити форму:</p> <ul style="list-style-type: none"> - назва послуги; - опис послуги; - адреса послуги; - категорія до якої належить послуга; - країна до якої належить послуга. <p>Якщо користувач не заповнить всі поля та / або введе некоректні дані, то має з'явитись відповідне повідомлення, яке вкаже користувачу, що необхідно змінити для успішного створення сторінки послуги.</p>
F-6	<p>Після успішного створення сторінки послуги, користувач повинен мати змогу додати часові проміжки роботи сервісу. Для цього необхідно заповнити відповідну форму, де необхідно вказати:</p> <ul style="list-style-type: none"> - час початку часового проміжку; - час закінчення часового проміжку. <p>Якщо час початку часового проміжку буде пізніше ніж час закінчення часового проміжку, то користувач має отримати повідомлення про цю помилку.</p>
F-7	<p>Користувач повинен мати змогу переглядати список всіх наявних послуг.</p>

Код вимоги	Опис вимоги
F-8	Користувач повинен мати можливість перейти на сторінку обраної послуги та переглянути детальну інформацію.
F-9	<p>Користувач повинен мати можливість забронювати обрану послугу. Для цього необхідно перейти на відповідну сторінку та обрати:</p> <ul style="list-style-type: none"> - дату резервації послуги; - часовий проміжок резервації послуги. <p>Якщо вільних часових проміжків для обраної дати немає, то користувач має отримати повідомлення про це.</p>
F-10	Користувач повинен мати змогу переглянути всі коментарі до послуги.
F-11	<p>Користувач повинен мати можливість залишати відгук про послугу. Для цього необхідно обрати послугу та заповнити форму, де вказати:</p> <ul style="list-style-type: none"> - рейтинг (від 1 до 5); - текст коментаря. <p>Якщо користувач не заповнить всі поля та / або введе некоректні дані, то має з'явитись відповідне повідомлення, яке вкаже користувачу, що необхідно змінити для успішного створення відгуку послуги.</p>
F-12	Користувач, який залишив коментар про послугу повинен мати змогу відредагувати свій відгук.
F-13	Користувач, який залишив коментар про послугу, а також власник послуги повинні мати змогу видалити коментар.
F-14	Користувач повинен мати змогу переглядати сторінку свого облікового запису.

Код вимоги	Опис вимоги
F-15	Користувач повинен мати змогу редагувати дані вказані при реєстрації, проте після зміни електронної пошти та / або паролю користувач повинен знову увійти до системи.
F-16	Користувачі повинні мати змогу видаляти свої облікові записи.
F-17	Користувачі повинні мати можливість переглядати історію своїх бронювань.
F-18	Користувачі повинні мати змогу видаляти створені бронювання.
F-19	Користувачі повинні мати змогу переглянути послуги, власниками яких вони є.
F-20	Власники послуг повинні мати змогу відслідковувати бронювання послуги, щоб мати загальне уявлення про роботу сервісу.
F-21	Власники послуги повинні мати змогу редагувати дані сервісу вказані при її створенні.
F-22	Власники послуги повинні мати змогу видаляти свій сервіс.
F-23	Власники послуги повинні мати змогу змінювати часові проміжки роботи сервісу.
F-24	Користувачі повинні мати змогу змінити мову користувацького інтерфейсу.
F-25	Користувач повинен мати змогу вийти зі свого облікового запису.

Функціональні можливості розробленого програмного забезпечення повністю відповідають описаним вимогам.

Далі представлена діаграма варіантів використання застосунку (рис. 3.3), яка допомагає візуалізувати всі основні функції програмного продукту та забезпечує розуміння, як різні типи користувачів можуть взаємодіяти з програмним забезпеченням для досягнення своїх цілей.

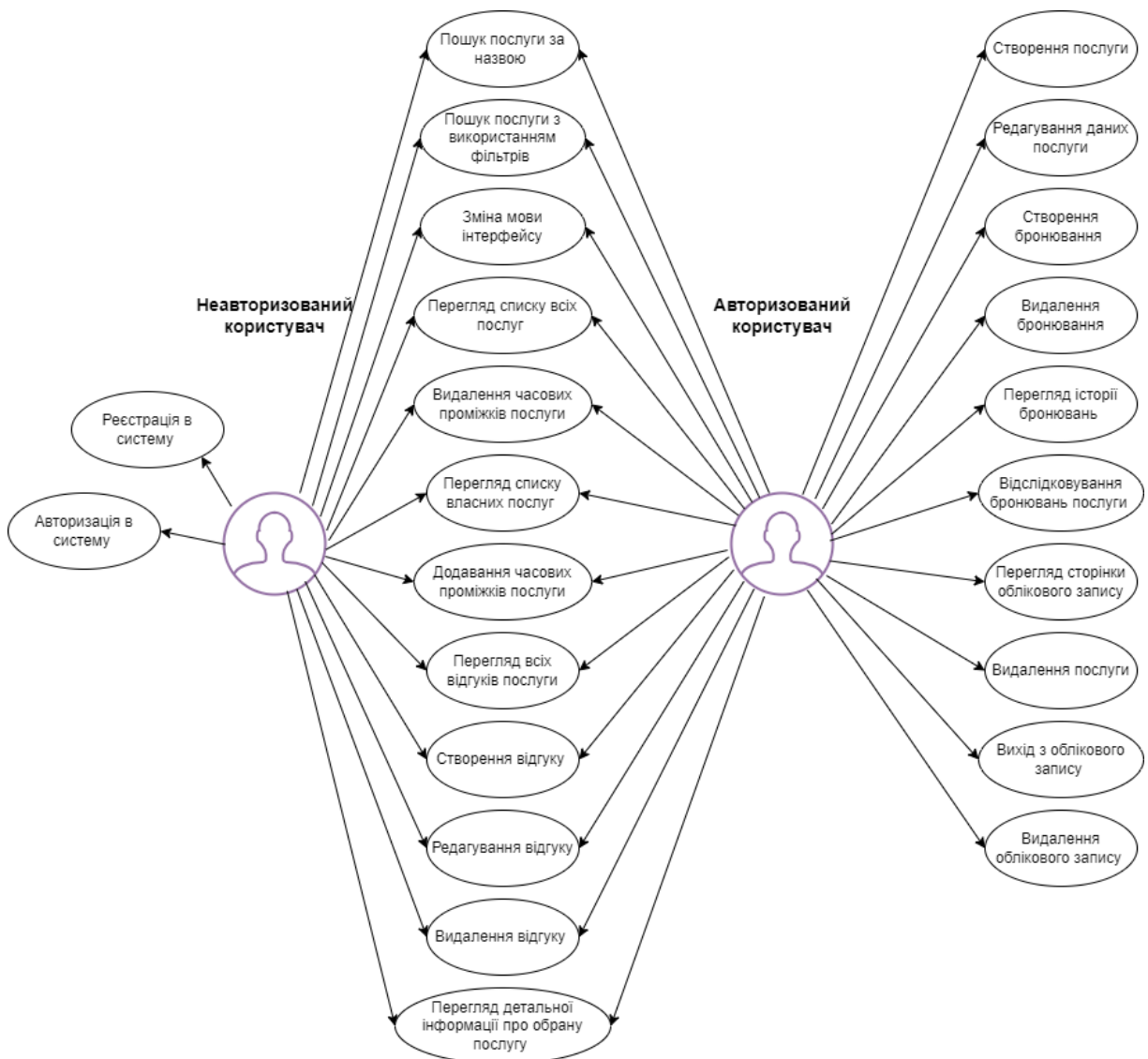


Рис. 3.3. Діаграма варіантів використання вебзастосунку

3.4. Алгоритм пошуку вільних часових проміжків для бронювання сервісу

Одним з найбільш важливих алгоритмів роботи розроблюваного застосунку є алгоритм пошуку вільних часових проміжків для бронювання сервісу.

Цей алгоритм безпосередньо використовується, коли користувач хоче забронювати вподобаний сервіс та знаходиться на сторінці перегляду детальної інформації про послугу. Користувач має можливість обрати дату, переглянути вільні часові проміжки та потім обрати час, який буде задовольняти його персональний розклад.

На рис. 3.4 наведено алгоритм пошуку вільних часових проміжків сервісу для обраної дати у вигляді блок-схеми.

Для початку роботи алгоритму необхідно обрати сервіс та дату бронювання в календарі. Потім система знаходить всі бронювання сервісу на обрану дату та виділяє окремий список ідентифікаторів часових проміжків для подальшого використання. Отримавши список ідентифікаторів часових проміжків, які вже заброньовані, алгоритмом передбачене знаходження списку всіх ідентифікаторів часових проміжків для обраного сервісу. Порівнюючи список заброньованих ідентифікаторів часових проміжків зі списком всіх ідентифікаторів часових проміжків, алгоритм виділяє окремий список ідентифікаторів доступних часових проміжків.

Таким чином, результат алгоритму є список ідентифікаторів доступних часових проміжків сервісу для обраної дати. Дана версія алгоритму є остаточною з огляду на поточні умови та вимоги, проте з розвитком технологій він може бути вдосконалений за рахунок використання нових методів і підходів.

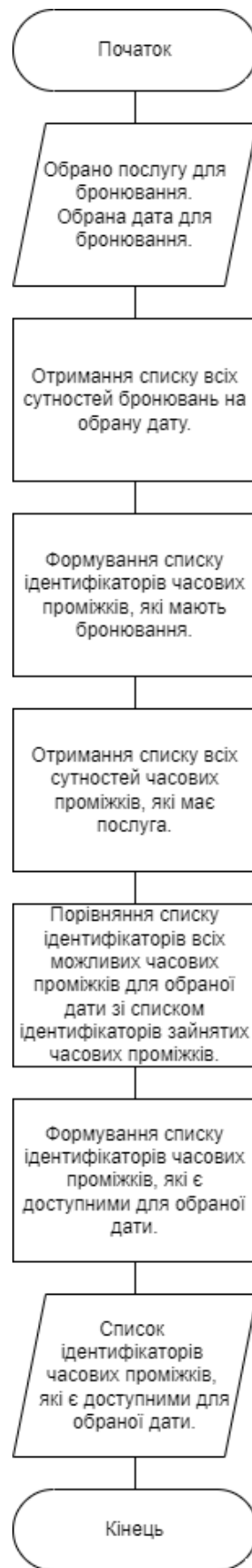


Рис. 3.4. Блок-схема алгоритму пошуку вільних часових проміжків послуги для обраної дати

3.5. Висновки до розділу 3

Під час роботи над третім розділом пояснювальної записки було завершено виконання наступних задач:

1. Описано архітектуру розробленого вебзастосунку та побудовано узагальнену схему архітектури програмного забезпечення.
2. Детально описано розроблену структуру даних та створено діаграму, яка відображає взаємозв'язки між її сутностями.
3. Описано функціональні вимоги до розробленого програмного продукту.
4. Побудовано блок-схему та описано роботу розробленого алгоритму пошуку вільних часових проміжків сервісу для обраної дати.

4. ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ВЕБЗАСТОСУНКУ

4.1. Додаткові інструменти, використані для програмної реалізації вебзастосунок

У процесі розроблення вебзастосунок для пошуку та резервування послуг використовувались різні основні компоненти бібліотеки React для створення естетично привабливих інтерфейсів. Використання цих компонентів не лише спростило процес розроблення, але й значно покращило візуальну привабливість вебдодатку, роблячи його більш зручним для користувачів. При цьому було застосовано:

1. React Router: React Router – це популярна бібліотека для маршрутизації у React додатках. Ця бібліотека дозволила організувати навігацію між різними частинами вебзастосунок без перезавантаження сторінки. Ця бібліотека допомогла забезпечити плавну та ефективну навігацію в застосунку для пошуку та резервування послуг, роблячи його інтуїтивно зрозумілим та зручним для користувачів.
2. React DatePicker: Ця бібліотека використовується при створенні користувачем часових проміжків. React DatePicker забезпечує зручний графічний інтерфейс для вибору дат та часу. Здатність цієї бібліотеки працювати з часовими даними забезпечується за рахунок додаткових налаштувань в компоненті “DatePicker”, що робить її дуже зручним рішенням для додатків, які потребують детальної інформації про дату та час, як вебзастосунок для пошуку та бронювання послуг.
3. React Calendar: React Calendar – це компонент календаря для React, який дозволяє легко інтегрувати календарні функції у вебдодатки. Цей компонент забезпечує гнучке і зручне управління датами, здатне відповідати різноманітним вимогам розробників і користувачів.

Окрім цих бібліотек React було використано:

1. Axios: Axios – це популярний HTTP клієнт, написаний на JavaScript, який використовується для виконання асинхронних запитів до серверів. Ця бібліотека дозволяє легко перетворювати дані для запитів і відповідей, щоб вони відповідали очікуванням сервера або клієнта.
2. Font Awesome: Font Awesome – візуальний інструментарій, який надає велику колекцію векторних іконок і інструментів, які можуть бути легко інтегровані в будь-який вебдизайн.

На стороні сервера функціонує серверний додаток, побудований на базі мікрофреймворку Flask. Цей фреймворк є основою для створення та обслуговування серверної логіки, забезпечуючи оброблення запитів, реалізацію бізнес-процесів, а також управління базами даних та транзакціями.

Flask за допомогою гнучкості та здатності інтегрувати різноманітні бібліотеки та утиліти Python значно розширив можливості серверної частини застосунку і сприяв ефективній розробці, окрім цього це дозволило мати повний контроль над компонентами розроблювального вебзастосунку.

Flask було використано для створення RESTful API, що є сучасним стандартом для проєктування мережових застосунків. RESTful API дозволив серверу взаємодіяти з клієнтськими застосунками через прості HTTP запити, обмінюючись даними у форматі JSON.

Також при розробленні серверної частини було використано інтеграцію з CORS (Cross-Origin Resource Sharing). CORS дозволяє вебдодаткам безпечно виконувати запити до ресурсів, розташованих на різних доменах, розширюючи можливості міждоменної взаємодії та покращуючи користувацький досвід. Цей механізм підвищує безпеку, забезпечуючи контроль над тим, які домени можуть доступатися до вебресурсів, і дозволяє розробникам гнучко масштабувати та інтегрувати різноманітні сервіси та додатки.

4.2. Опис реалізованого користувацького інтерфейсу

Вебзастосунок для пошуку та резервування послуг має односторінковий вебінтерфейс, який включає в себе такі елементи:

1. Сторінка авторизації та реєстрації.
2. Головна сторінка з пошуковою панеллю.
3. Сторінка для створення послуги.
4. Сторінка для створення робочих годин послуги (часових проміжків).
5. Сторінка для перегляду списку всіх послуг з елементами фільтрації.
6. Сторінка для перегляду детальної інформації про послугу зі списком відгуків.
7. Сторінка для редагування відгуку.
8. Сторінка для резервування послуги.
9. Сторінка профілю авторизованого користувача.
10. Сторінка редагування профілю авторизованого користувача.
11. Сторінка перегляду власних послуг.
12. Сторінка для редагування послуги та часових проміжків.
13. Сторінка перегляду зайнятості власної послуги.
14. Сторінка перегляду власних резервувань.

Коли користувач використовує вебзастосунок, то одразу потрапляє на сторінку авторизації. Залежно від наявності облікового запису, користувач обирає, що йому потрібно зробити: зареєструватися або увійти в систему.

Також користувач має варіант продовжити використання додатку без створення облікового запису, але більшість можливостей додатку в такому випадку для користувача будуть недоступні. Користувач, який не увійшов до системи, має можливість переглядати головну сторінку та використовувати інструменти пошуку та фільтрації для зручного відбору потрібної інформації згідно зі своїми потребами та вимогами.

Головна сторінка з пошуковою панеллю зображена на рис. 4.1. Неавторизований користувач також може переглядати повний перелік послуг разом з детальними описами кожної з них. На рис. 4.2 та 4.3 зображено повний перелік послуг з панеллю фільтрації.

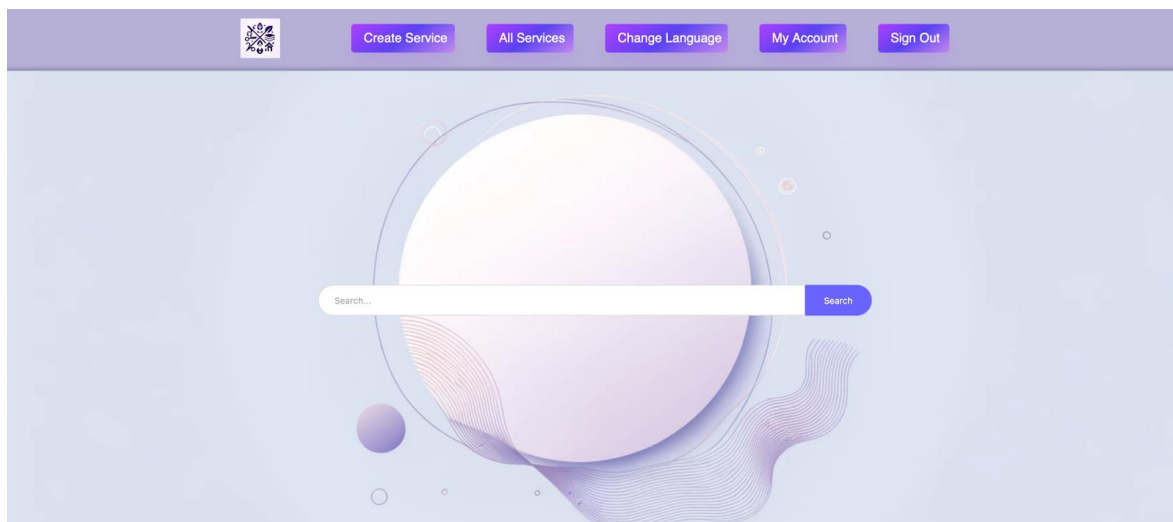


Рис. 4.1. Головна сторінка вебзастосунку

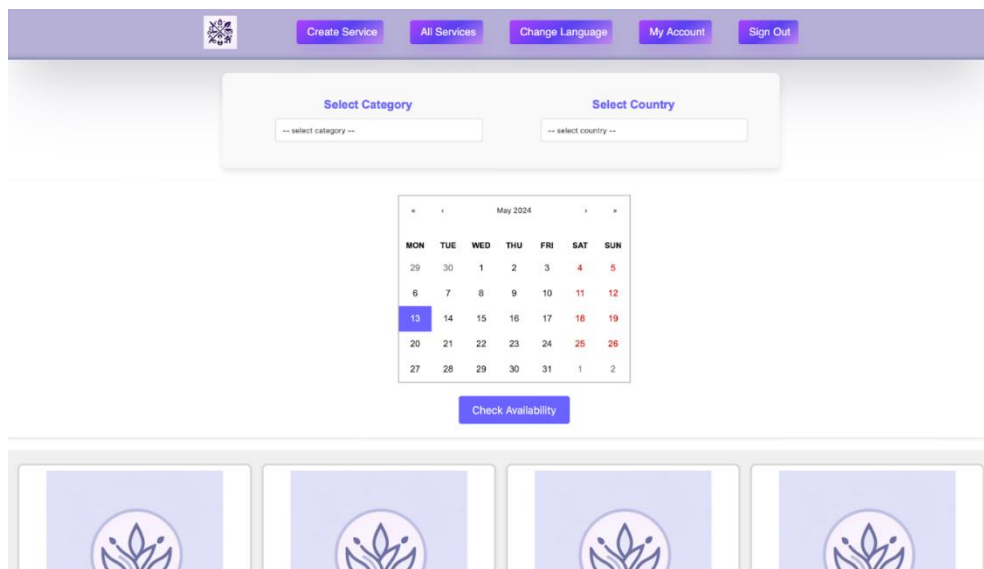


Рис. 4.2. Сторінка перегляду переліку послуг. Частина 1

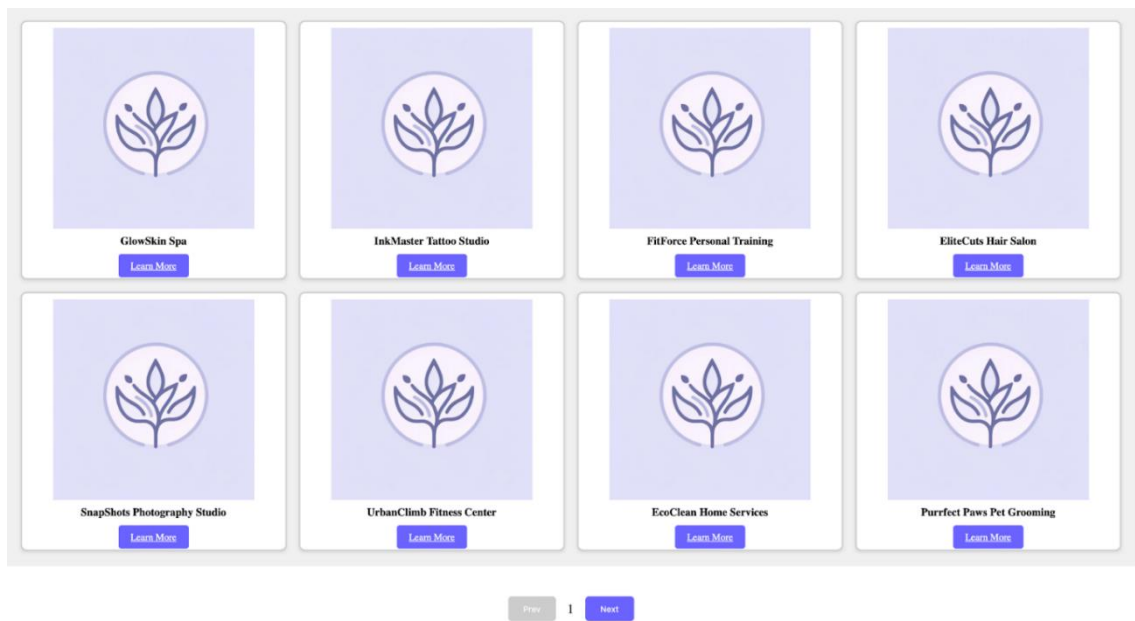


Рис. 4.3. Сторінка перегляду переліку послуг. Частина 2

Натомість авторизований користувач може створити послугу (рис. 4.4) та додавати часові проміжки після створення послуги, резервувати послугу (рис. 4.5), переглядати інформацію про власний обліковий запис, переглядати інформацію про власні бронювання (рис. 4.6) та власні послуги.

Рис. 4.4. Сторінка створення послуги

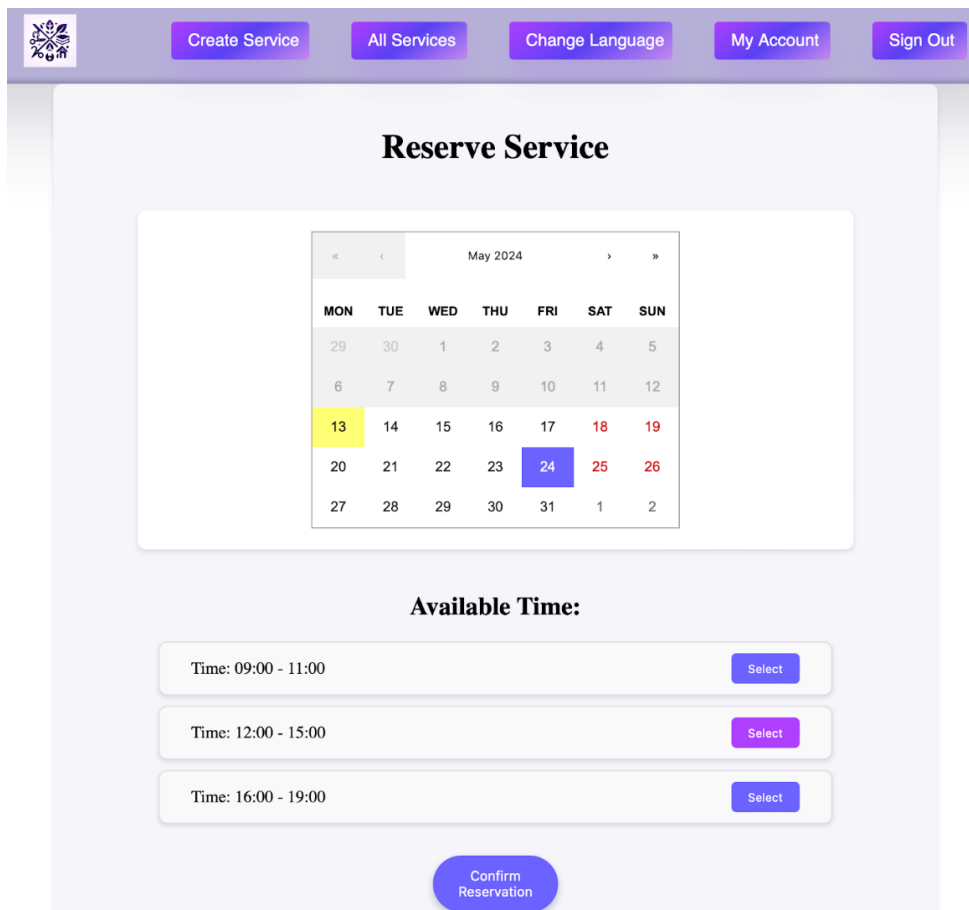


Рис. 4.5. Сторінка резервування послуги

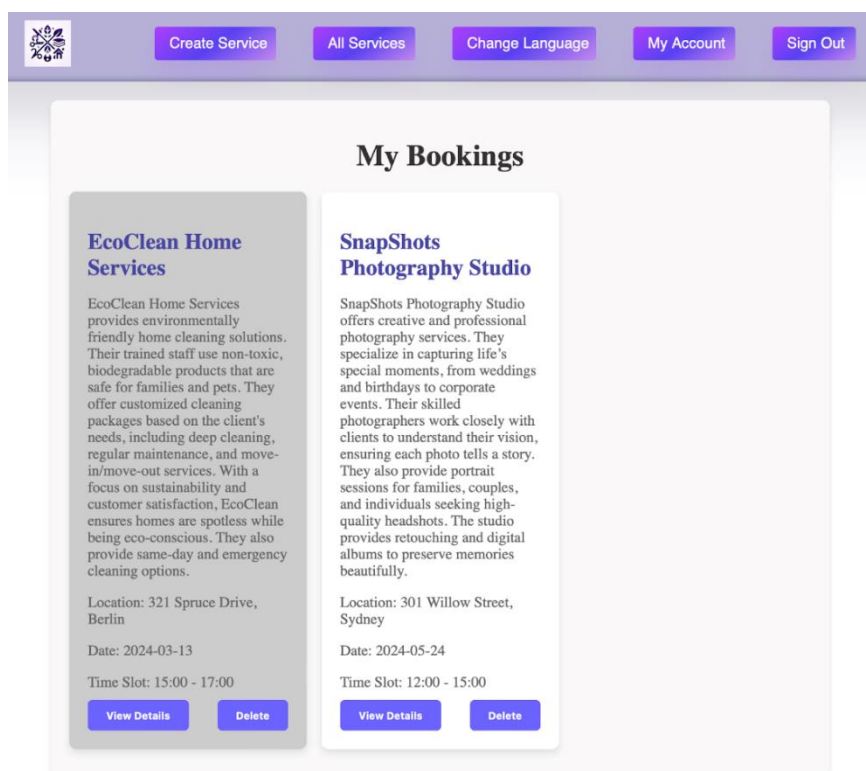


Рис. 4.6. Сторінка перегляду резервувань користувача

Також застосунок підтримує дві мови інтерфейсу користувача: англійську (рис. 4.7) та українську (рис. 4.8). Для зміни мови користувач має натиснути на кнопку “Change Language”.

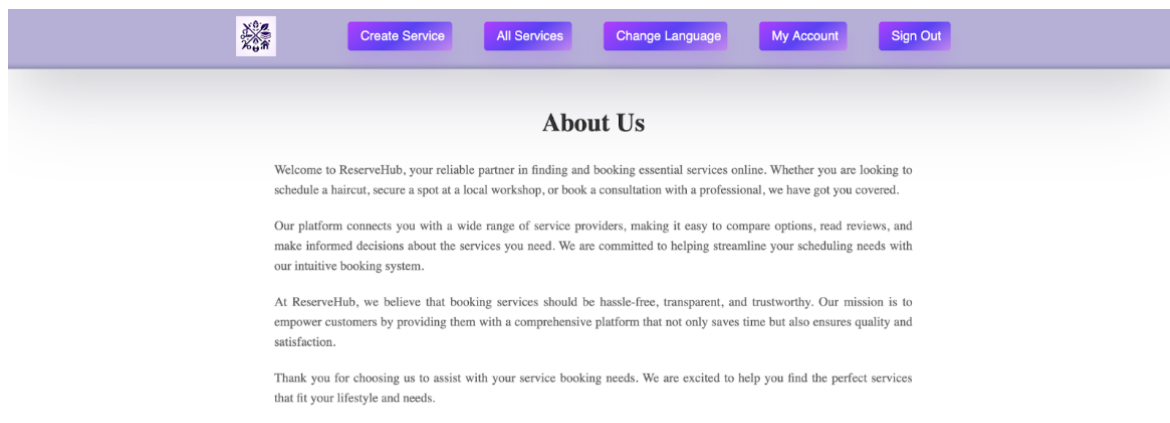


Рис. 4.7. Сторінка перегляду інформації про вебзастосунок. Англійська версія

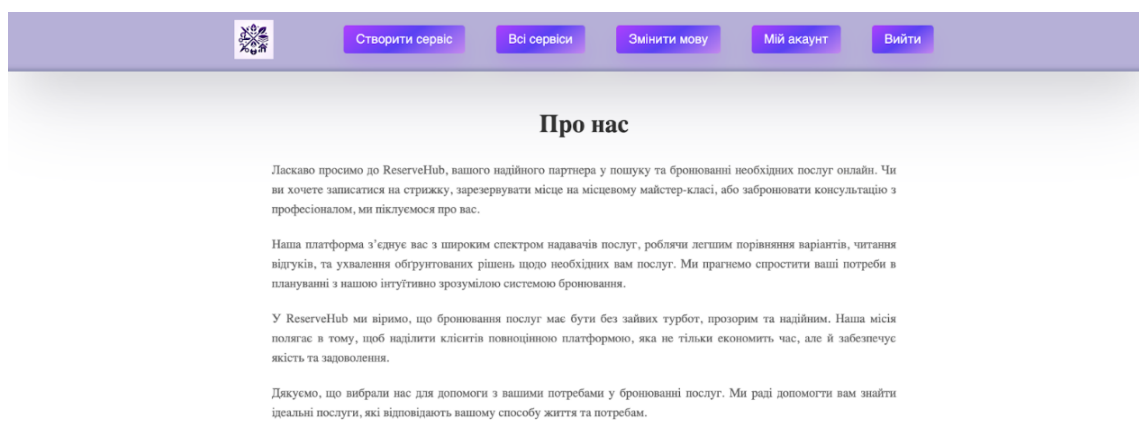


Рис. 4.8. Сторінка перегляду інформації про вебзастосунок. Українська версія

Інтерфейс застосунку розроблено таким чином, що він автоматично адаптується до розміру екрану пристрою, забезпечуючи оптимальне відображення та зручність користування на різних пристроях. Використання адаптивних макетів та медіа-запитів дозволяє контенту динамічно реорганізовуватися, щоб відповідати ширині екрану, від смартфонів (рис. 4.9, 4.10, 4.11) до настільних комп’ютерів. Такий підхід

забезпечує високу якість взаємодії для користувачів, незалежно від того, з якого пристрою вони входять у застосунок.

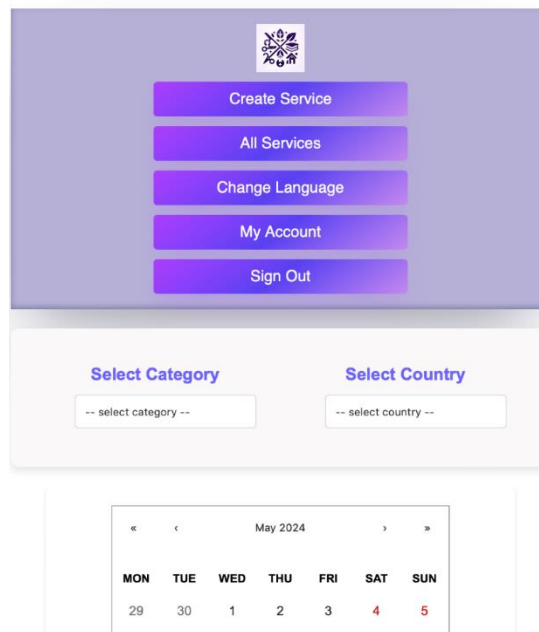


Рис. 4.9. Хедер вебзастосунку пристосований до розмірів екрану смартфона

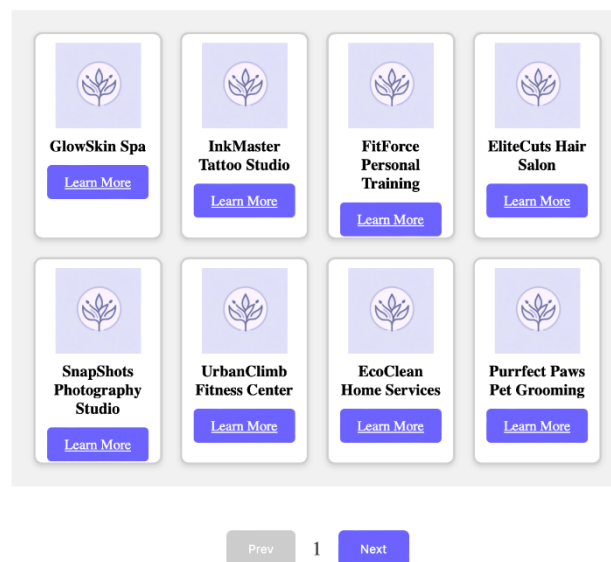


Рис. 4.10. Сторінка перегляду переліку послуг пристосована до розмірів екрану смартфона

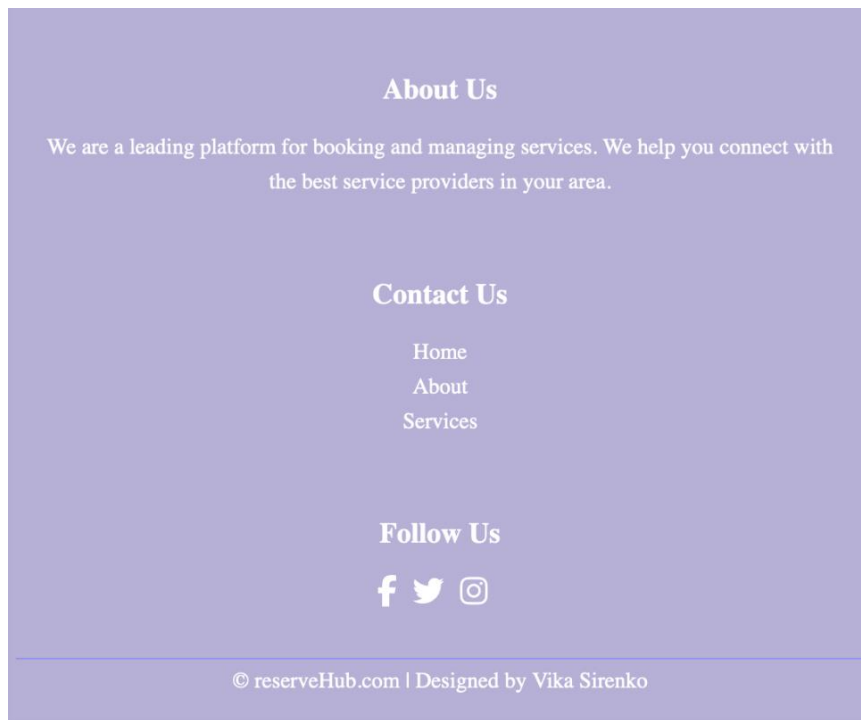


Рис. 4.11. Футер вебзастосунку, пристосований до розмірів екрану смартфона

4.3. Особливості проведеного тестування

Для перевірки стабільності основних функцій розробленого програмного забезпечення було проведено димовий тест. Було перевірено основні функції, такі як вхід в систему, пошук послуг, створення послуги та процес резервування сервісу, щоб упевнитися, що вони працюють без помилок і зручні для користувача. Такі тести дозволяють виявити будь-які критичні проблеми на ранніх стадіях розробки. Завдяки регулярному проведенню димових тестів, вебзастосунок забезпечує стабільність і надійність, гарантуючи, що він залишається функціональним та готовим до впровадження нових оновлень або функцій, не наражаючи на небезпеку основну діяльність користувачів.

Далі описано детальну послідовність кроків виконаного тестування.

Тестові випадки димового тестування

№	Назва	Передумова	Сценарій дій	Очікуваний результат
1.	Можливість реєстрації в системі	Користувач не має облікового запису	<p>1. Користувач натискає на кнопку «Registration».</p> <p>2. Користувач заповнює поля.</p> <p>3. Користувач натискає на кнопку «Register».</p>	Користувач успішно зареєстрований в системі.
2.	Можливість авторизації в системі	Користувач зареєстрований в системі	<p>1. Користувач заповнює поля електронної адреси та паролю.</p> <p>2. Користувач натискає на кнопку «Sign In».</p>	Користувач успішно авторизувався.
3.	Можливість входу в систему без реєстрації або авторизації	Користувач не зареєстрований в системі	Користувач натискає на посилання «Go to Main Page».	Користувач може здійснювати пошук та фільтрацію сервісів, а також переглядати детальну інформацію про послугу.

№	Назва	Передумова	Сценарій дій	Очікуваний результат
4.	Можливість створення послуги	Користувач є авторизованим в системі	<ol style="list-style-type: none"> 1. Користувач переходить до вкладки «Create Service». 2. Користувач заповнює всі необхідні поля. 3. Користувач натискає на кнопку «Create Service» та переходить на сторінку створення часових проміжків. 4. Користувач заповнює поля «Start Time» та «End Time». 5. Користувач натискає на кнопку «Create Time Slot» для додавання часового проміжку в список. 6. Користувач натискає на кнопку «Complete Time Slots» для завершення додавання часових проміжків. 	Створена послуга відображається у вкладці «All Services» та у списку власних послуг користувача.

№	Назва	Передумова	Сценарій дій	Очікуваний результат
5.	Можливість пошуку послуги	Користувач є авторизованим або неавторизованим	1. Користувач вводить пошуковий запит в поле пошуку. 2. Користувач натискає на кнопку «Search».	Система відображає список послуг, які відповідають пошуковому запиту або повідомлення про те, що жодна послуга не відповідає запиту користувача.
6.	Можливість фільтрації послуг за заданими параметрами	Користувач є авторизованим або неавторизованим	1. Користувач переходить до вкладки «All Services». 2. Користувач обирає категорію та/або обирає країну та/або обирає дату для фільтрації.	Система відображає послуги, які задовольняють параметри фільтрації, або повідомлення про те, що жодна послуга не відповідає параметрам фільтрації.
7.	Можливість перегляду детальної інформації про обрану послугу	Користувач є авторизованим або неавторизованим	Користувач натискає на кнопку «Learn More».	Система відображає детальну інформацію про послугу.
8.	Можливість резервування послуги	Користувач є авторизованим.	1. Користувач натискає на кнопку «Reserve». 2. Користувач обирає дату.	Створене резервування відображається у списку власних резервувань користувача.

№	Назва	Передумова	Сценарій дій	Очікуваний результат
			<p>3. Користувач обирає часовий проміжок.</p> <p>4. Користувач натискає на кнопку «Confirm Reservation».</p>	
9.	Можливість створення відгуку про послугу	Користувач є авторизованим.	<p>1. Користувач надає оцінює послугу від 1 до 5.</p> <p>2. Користувач залишає коментар про послугу.</p> <p>3. Користувач натискає на кнопку «Submit Review».</p>	Система відображає створений коментар під інформацією про послугу.
10.	Можливість редагування відгуку про послугу	Користувач є авторизованим та є власником відгуку	<p>1. Користувач натискає на кнопку «Edit».</p> <p>2. Користувач редагує необхідні поля.</p> <p>3. Користувач натискає на кнопку «Update Review».</p>	Система зберігає та відображає всі внесені зміни до відгуку.

№	Назва	Передумова	Сценарій дій	Очікуваний результат
11.	Можливість видалення відгуку	Користувач є авторизованим та є власником послуги або власником відгуку	1. Користувач натискає на кнопку «Delete». 2. Користувач підтверджує, що хоче видалити послугу, натискаючи на «ОК».	Система видаляє відгук про послугу.
12.	Можливість зміни мови інтерфейсу користувача	Користувач є авторизованим або неавторизованим	Користувач натискає на кнопку «Change Language».	Система змінює мову інтерфейсу користувача на українську або на англійську.
13.	Можливість перегляду акаунту користувача	Користувач є авторизованим	Користувач переходить до вкладки «My account».	Система відображає інформацію про авторизованого користувача.
14.	Можливість перегляду списку власних послуг	Користувач є авторизованим	Користувач натискає на кнопку «View My Services».	Система відображає послуги створені авторизованим користувачем.
15.	Можливість перегляду списку власних бронювань	Користувач є авторизованим	Користувач натискає на кнопку «View My Reservations».	Система відображає резервування авторизованого користувача.

4.4. Рекомендації для подальшого вдосконалення

Розглянемо способи подальшого розширення та покращення вебзастосунку для пошуку та резервування послуг:

1. Покращення пошукової функціональності: інтеграція інтелектуальних алгоритмів пошуку, які можуть пропонувати користувачам варіанти на основі їхньої попередньої поведінки або популярних тенденцій. Це може включати використання технологій оброблення природної мови, машинного навчання, семантичного пошуку, тощо.
2. Мобільна оптимізація: розроблення нативних мобільних додатків для iOS та Android для безперешкодного доступу користувачів до вебзастосунку зі своїх смартфонів або планшетів.
3. Мовна підтримка: розширення списку мов користувацького інтерфейсу вебзастосунку для підвищення доступності та зручності для міжнародної аудиторії.
4. Інтеграція із зовнішніми API: інтеграція з Google Maps або іншими сервісами карт для надання користувачам інформації про місцезнаходження та напрямки до місць надання послуг. Інтеграція календаря вебзастосунку з іншими календарями, такими як Google Calendar або Microsoft Outlook для синхронізування події між різними платформами, гарантуючи, що всі зустрічі, заходи та нагадування будуть оновлюватись в реальному часі в усіх їхніх календарях. Це усуне потребу вручну додавати ті ж події в різні календарі, зменшуючи ризик пропуску важливих зобов'язань.
5. Забезпечення безпеки та конфіденційності: впровадження більш сучасних стандартів безпеки для захисту особистих даних користувачів та використання HTTPS для забезпечення безпеки даних під час передачі.

- б. Інтеграція чат-платформи: впровадження чату для спілкування між користувачами та постачальниками послуг у вебзастосунку істотно полегшить взаємодію між користувачем та постачальником, а також полегшить процес консультацій та вирішення проблем, пов'язаних з послугою.

Реалізація всіх вищезазначених стратегій масштабування та вдосконалення вебзастосунку може забезпечити стабільну роботу при зростанні запитів від користувачів, а також це дозволить підвищити рівень обслуговування та взаємодії з користувачами.

4.5. Висновки до розділу 4

У цьому розділі представлено детальний опис реалізації клієнтської та серверної частин розроблюваного веб-застосунку для пошуку та резервування послуг. Також розглянуто інтерфейс, який було розроблено для забезпечення зручної взаємодії користувача з програмним забезпеченням. Окрім цього, розглянуто стратегії та рекомендації для подальшого розвитку і оптимізації функціональності вебзастосунку.

ВИСНОВКИ

Метою даного дипломного проєкту було розроблення програмного рішення для надання користувачам змоги знаходити інформацію про необхідні послуги та здійснювати їх бронювання по всьому світу.

На початковому етапі було проведено аналіз існуючих рішень на ринку, що дозволило виявити основні тенденції та потреби користувачів, а також ідентифікувати недоліки. Для аналізу функціональності та визначення функціональних вимог майбутнього програмного забезпечення було розглянуто три програмні продукти: Booking.com, Google Maps та OLX.

Вибір технологій для реалізації проєкту ґрунтувався на поточних стандартах розроблення вебдодатків, включаючи використання сучасних фреймворків і мов програмування. Порівнявши популярні мови програмування та детально розглянувши переваги та недоліки кожної, було вирішено для клієнтської частини використовувати React.js, а для серверної частини – мову програмування Python.

Архітектура розробленого вебзастосунку була розроблена згідно шаблону Model-View-Controller (MVC), що дозволило відокремити бізнес-логіку (Model) від користувацького інтерфейсу (View) з контролем (Controller), що управляє взаємодією між ними. Такий підхід сприяв легшій модифікації та розширенню окремих компонентів без впливу на інші, що забезпечило гнучкість у розвитку проєкту.

В архітектурі також було передбачено використання бази даних, розділеної на сім таблиць, що оптимізувало зберігання та виклик даних. Таке рішення забезпечило ефективність обробки запитів та високий рівень безпеки інформації.

На відміну від розглянутих програмних продуктів розроблений вебзастосунок для пошуку та резервування послуг дозволяє зареєструватись в системі, створити послугу та налаштувати часовий розклад для

резервувань, шукати та фільтрувати послуги за різними категоріями, переглядати детальну інформацію про кожну послугу, резервувати послугу та скасовувати резервування, залишати коментарі, крім цього розроблений вебзастосунок може автоматично адаптуватися до розміру екрану пристрою, а також наявна підтримка декількох мов інтерфейсу.

Додатково було здійснено аналіз потенційних вдосконалень для наступних версій вебзастосунку, що може включати: покращення пошукової функціональності, мобільну оптимізацію, розширення мовної підтримки інтерфейсу, інтеграцію з зовнішнім API, інтеграція чат-платформи та подальше виправлення помилок та недоліків.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Як виконати конкурентний аналіз - класичні та сучасні методики [Електронний ресурс]. — Режим доступу: <https://wezom.com.ua/ua/blog/kak-delat-konkurentnyj-analiz> — (16.01.2024)
2. Booking.com: Про Booking.com™ [Електронний ресурс]. — Режим доступу: <https://www.booking.com/content/about.uk.html> — (17.01.2024)
3. Google Maps [Електронний ресурс]. — Режим доступу: https://en.wikipedia.org/wiki/Google_Maps — (16.01.2024)
4. OLX [Електронний ресурс]. — Режим доступу: <https://en.wikipedia.org/wiki/OLX> — (16.01.2024)
5. Booking.com: Як ми працюємо [Електронний ресурс]. — Режим доступу: https://www.booking.com/content/how_we_work.uk.html?aid=357028&label=bin859jc-1DCBQoggJCC2hvd193ZV93b3JrSDNYA2hQiAEBmAEpuAEXyAEM2AED6AEBiAIBqAIDuALFhqCtBsACAdICJGVjZjI1OGQ2LWI0OTktNGQ4Ni05ZjEzLTQ4OTkwNGQxM2IwMNgcCBOACAQ&sid=bc5529271a5a01f6e85cea7af6ab77c3&keep_landing=1& — (17.01.2024)
6. Reserve with Google [Електронний ресурс]. — Режим доступу: <https://www.google.com/maps/reserve/> — (29.01.2024)
7. Python (programming language) [Електронний ресурс]. — Режим доступу: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)) — (31.01.2024)
8. Advantages of Python | Disadvantages of Python [Електронний ресурс]. — Режим доступу: <https://pythongeeks.org/advantages-disadvantages-of-python/> — (01.02.2024)
9. C Sharp (programming language) [Електронний ресурс]. — Режим доступу:

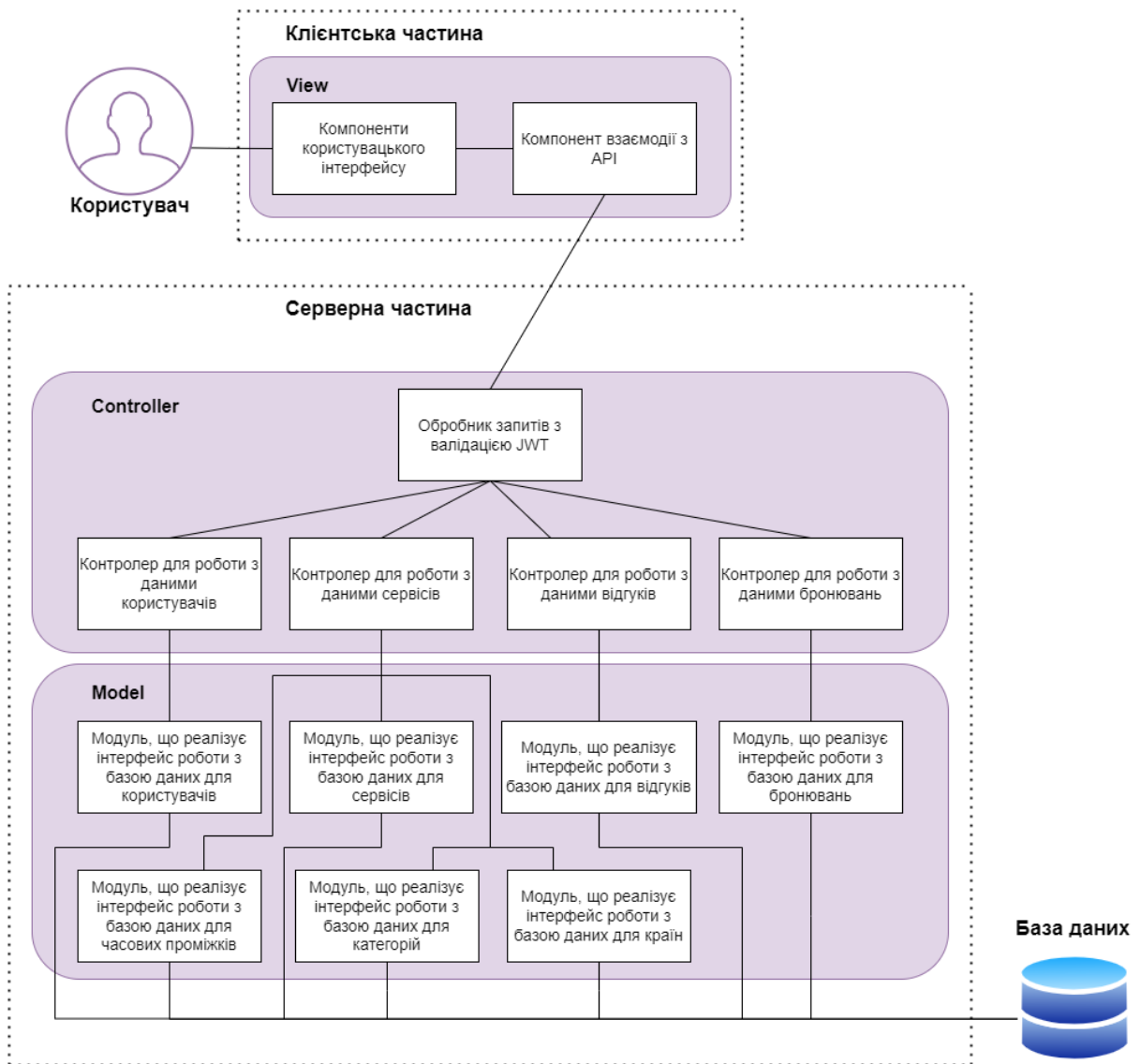
- [https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)) — (02.02.2024)
10. The Good and the Bad of C# Programming [Электронный ресурс]. — Режим доступа: <https://www.altexsoft.com/blog/c-sharp-pros-and-cons/> — (02.02.2024)
 11. Java (programming language) [Электронный ресурс]. — Режим доступа: https://en.wikipedia.org/wiki/Java_%28programming_language%29 — (02.02.2024)
 12. Advantages and disadvantages of Java [Электронный ресурс]. — Режим доступа: <https://www.javatpoint.com/advantages-and-disadvantages-of-java> — (02.02.2024)
 13. React (software) [Электронный ресурс]. — Режим доступа: [https://en.wikipedia.org/wiki/React_\(software\)](https://en.wikipedia.org/wiki/React_(software)) — (05.02.2024)
 14. Advantages And Disadvantages Of ReactJS [Электронный ресурс]. — Режим доступа: <https://www.w3schools.blog/advantages-disadvantages-reactjs> — (05.02.2024)
 15. Introduction to the Angular docs [Электронный ресурс]. — Режим доступа: <https://angular.io/docs> — (05.02.2024)
 16. Advantages and Disadvantages of Angular [Электронный ресурс]. — Режим доступа: <https://www.knowledgehut.com/blog/web-development/advantages-and-disadvantages-of-angular#advantages-of-angular%C2%A0> — (05.02.2024)
 17. Introduction| Vue.js [Электронный ресурс]. — Режим доступа: <https://vuejs.org/guide/introduction.html> — (05.02.2024)
 18. The Good and the Bad of Vue.js Framework Programming [Электронный ресурс]. — Режим доступа: <https://www.altexsoft.com/blog/pros-and-cons-of-vue-js/> — (05.02.2024)
 19. Система керування базами даних [Электронный ресурс]. — Режим доступа: <https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%>

D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85 — (06.02.2024)

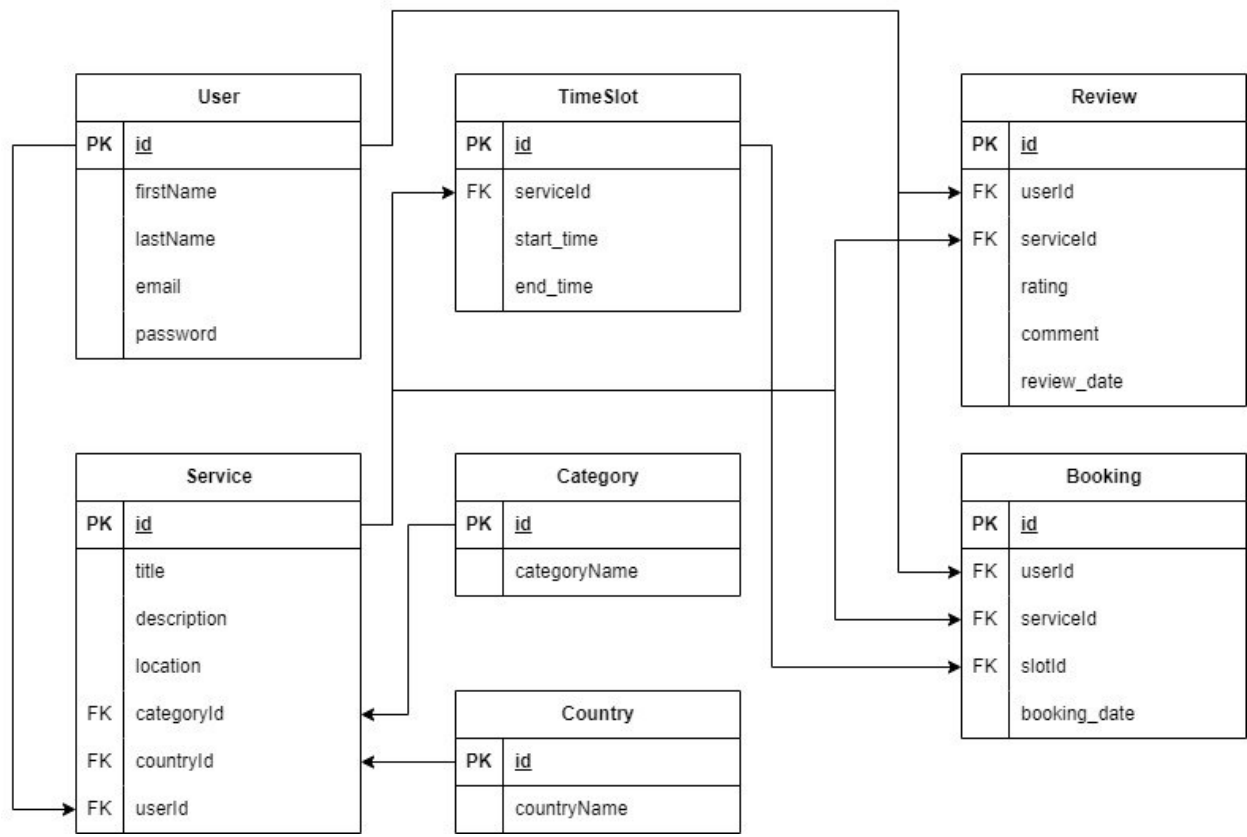
20. What Is MongoDB? [Электронный ресурс]. — Режим доступа: <https://www.mongodb.com/what-is-mongodb> — (06.02.2024)
21. MongoDB Advantages & Disadvantages [Электронный ресурс]. — Режим доступа: <https://www.geeksforgeeks.org/mongodb-advantages-disadvantages/> — (06.02.2024)
22. ACID [Электронный ресурс]. — Режим доступа: <https://en.wikipedia.org/wiki/ACID#:~:text=In%20computer%20science%2C%20ACID%20> — (06.02.2024)
23. MySQL [Электронный ресурс]. — Режим доступа: <https://en.wikipedia.org/wiki/MySQL> — (06.02.2024)
24. MySQL Advantages And Disadvantages [Электронный ресурс]. — Режим доступа: <https://www.w3schools.blog/mysql-advantages-disadvantages> — (06.02.2024)
25. PostgreSQL: About [Электронный ресурс]. — Режим доступа: <https://www.postgresql.org/about/> — (06.02.2024)
26. Model–view–controller [Электронный ресурс]. — Режим доступа: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller> — (04.05.2024)

ДОДАТКИ

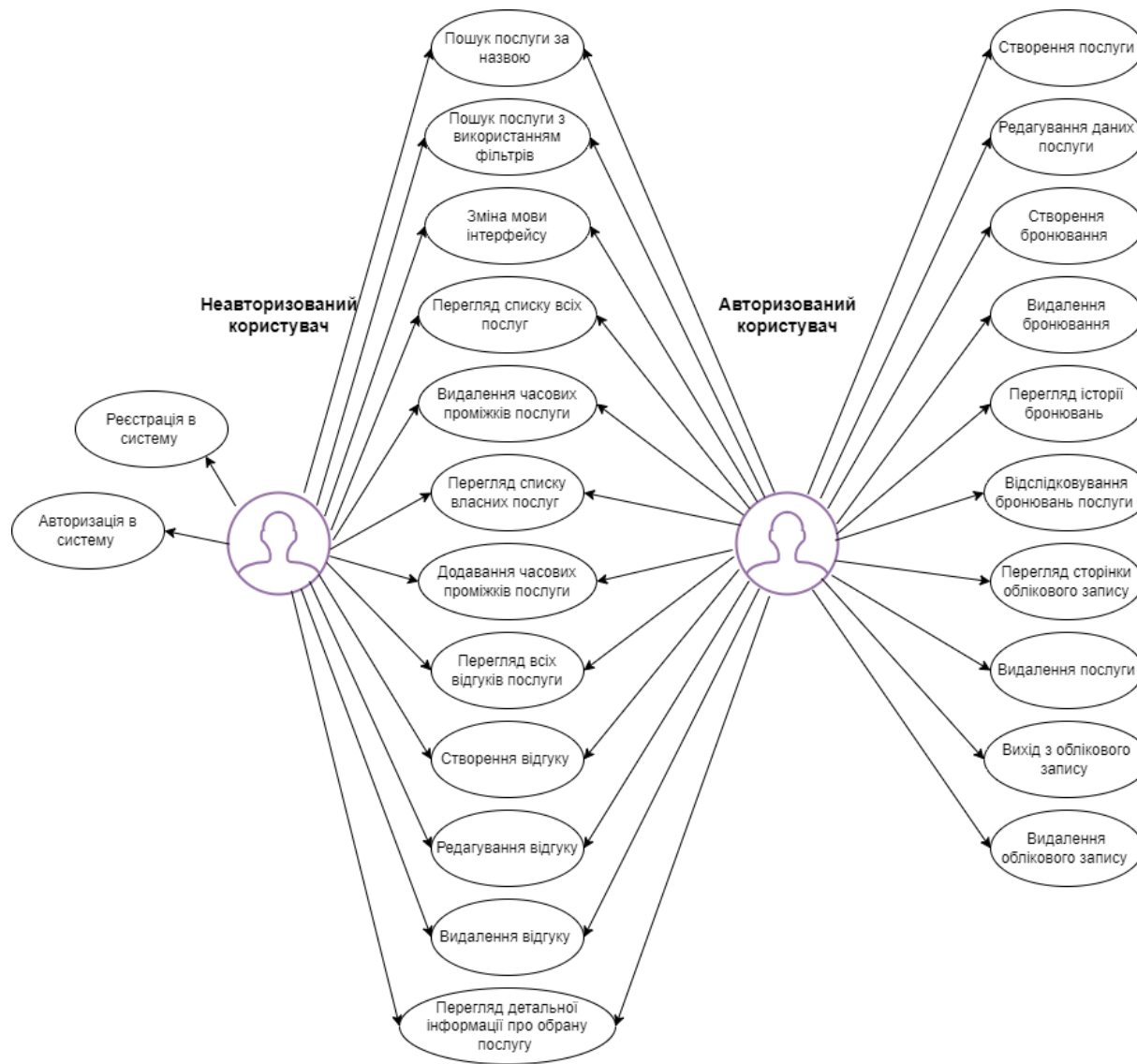
Додаток 1
Копії графічних матеріалів



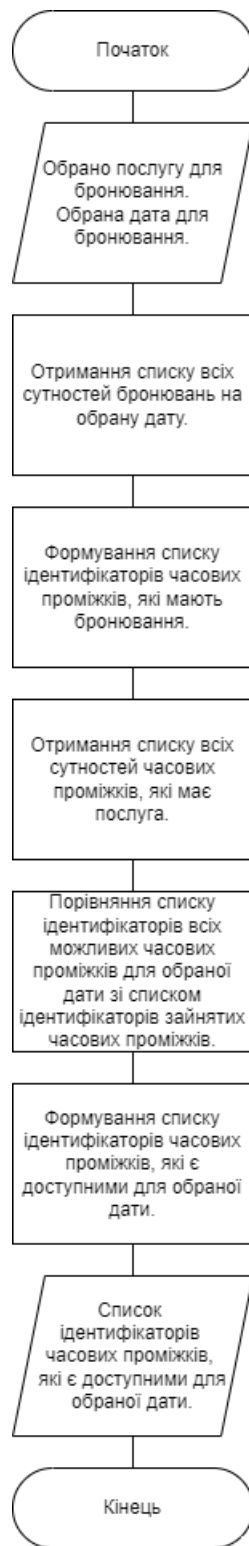
Узагальнена архітектура вебзастосунку
 Сіренко В.Ю., група КП-03



ДП.045440-06-99. Вебзастосунок для пошуку та резервування послуг. Структура бази даних. ERD-діаграма



Діаграма варіантів використання вебзастосунку
Сіренко В.Ю., КП-03



ДП.045440-07-99. Вебзастосунок для пошуку та резервування послуг. Алгоритм пошуку вільних часових проміжків послуги для обраної дати. Блок-схема алгоритму

Додаток 2
Копія презентації

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО"



ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

ВЕБЗАСТОСУНОК ДЛЯ ПОШУКУ ТА РЕЗЕРВУВАННЯ ПОСЛУГ

Виконала: Сіренко Вікторія Юріївна

Керівник: доц. кафедри ПЗКЗ, к.т.н., доц. Заболотня Тетяна Миколаївна

Київ-2024

ПОСТАНОВКА ЗАДАЧІ

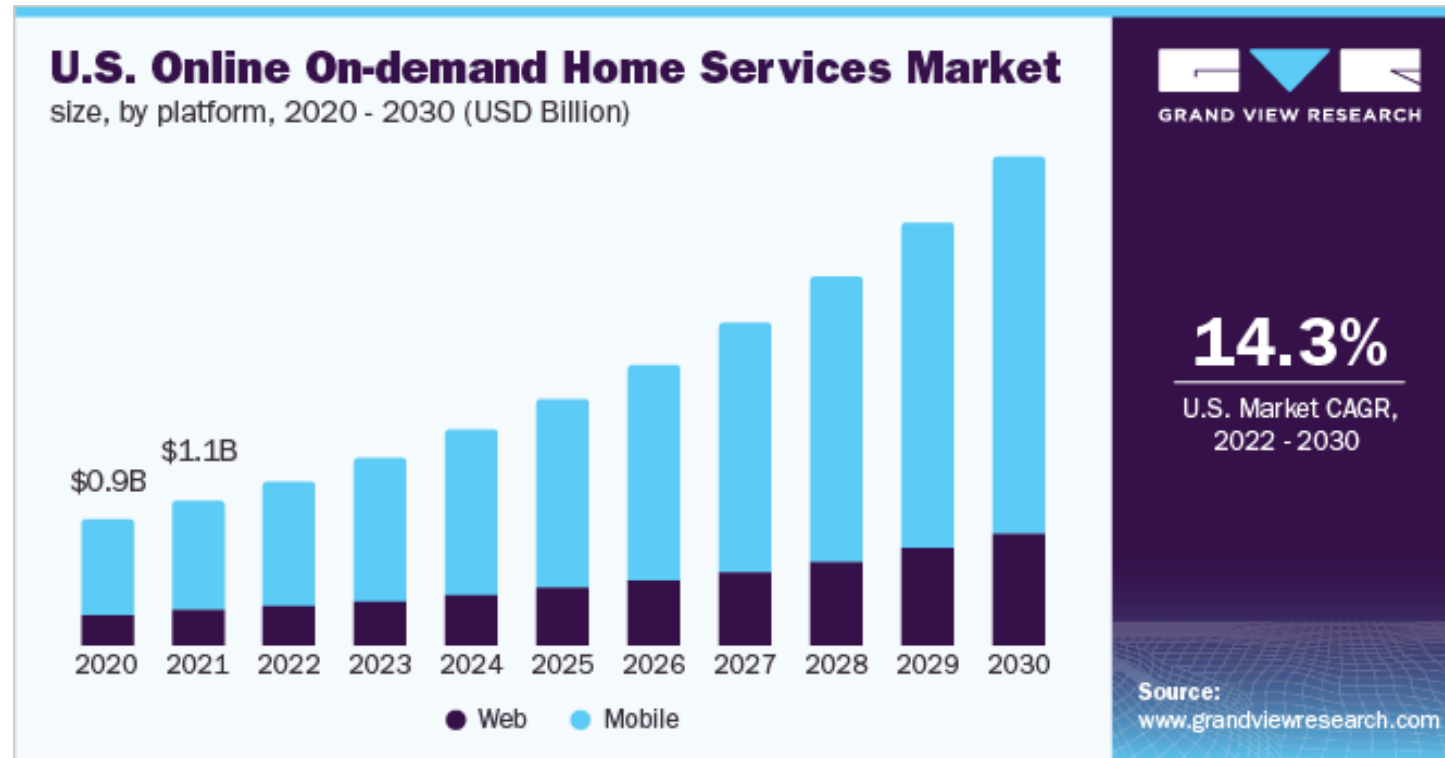


Мета проєкту: розробити програмне рішення для надання користувачам змоги знаходити інформацію про необхідні послуги та здійснювати їх бронювання по всьому світу.

Завдання:

1. Проаналізувати ринок ПЗ для пошуку та бронювання послуг.
2. Порівняти додатки-конкурентів, виявити їхніх сильні та слабкі сторони.
3. Визначити функціональні можливості програмного продукту, яких потребують потенційні користувачі.
4. Обґрунтувати вибір стеку технологій для розроблення ПЗ.
5. Підготувати дизайн вебзастосунку.
6. Розробити та програмно реалізувати вебзастосунок для пошуку та резервування послуг.
7. Протестувати створене програмне забезпечення.
8. Визначити напрями для подальшого розвитку.

АКТУАЛЬНІСТЬ



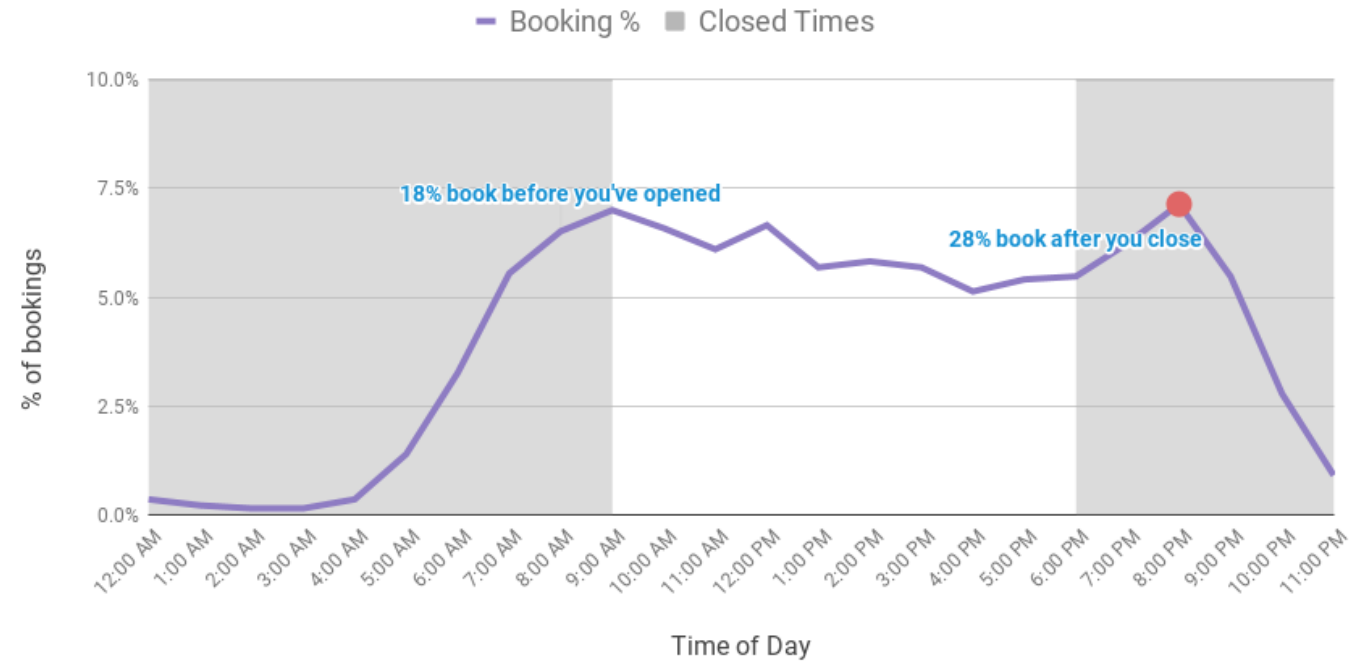
2022 рік – 4,3 мільярди доларів
2030 рік – 14,7 мільярдів доларів
Середньорічний темп зростання (CAGR) – 16,7%

АКТУАЛЬНІСТЬ



24 hour booking trend

How clients book when given the choice



54% записів відбувається в робочі години.
46% записів втрачаються, якщо відсутня можливість записатися
онлайн або через додаток.



ІСНУЮЧІ РІШЕННЯ

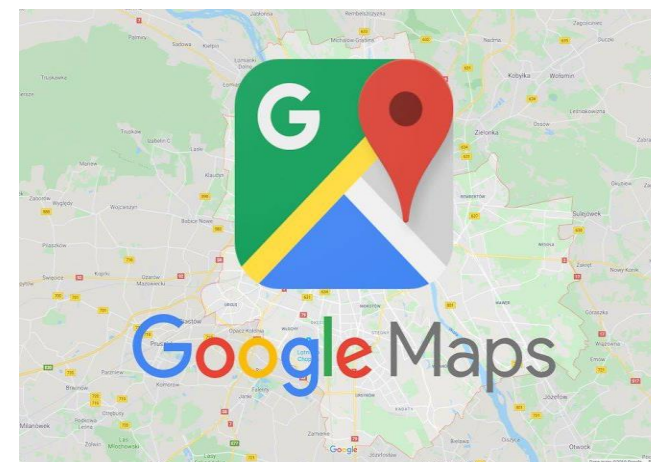
Booking.com



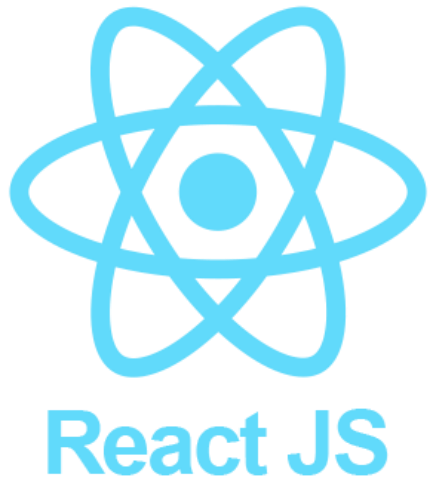
OLX



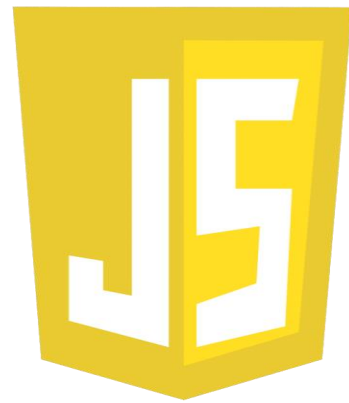
Google Maps



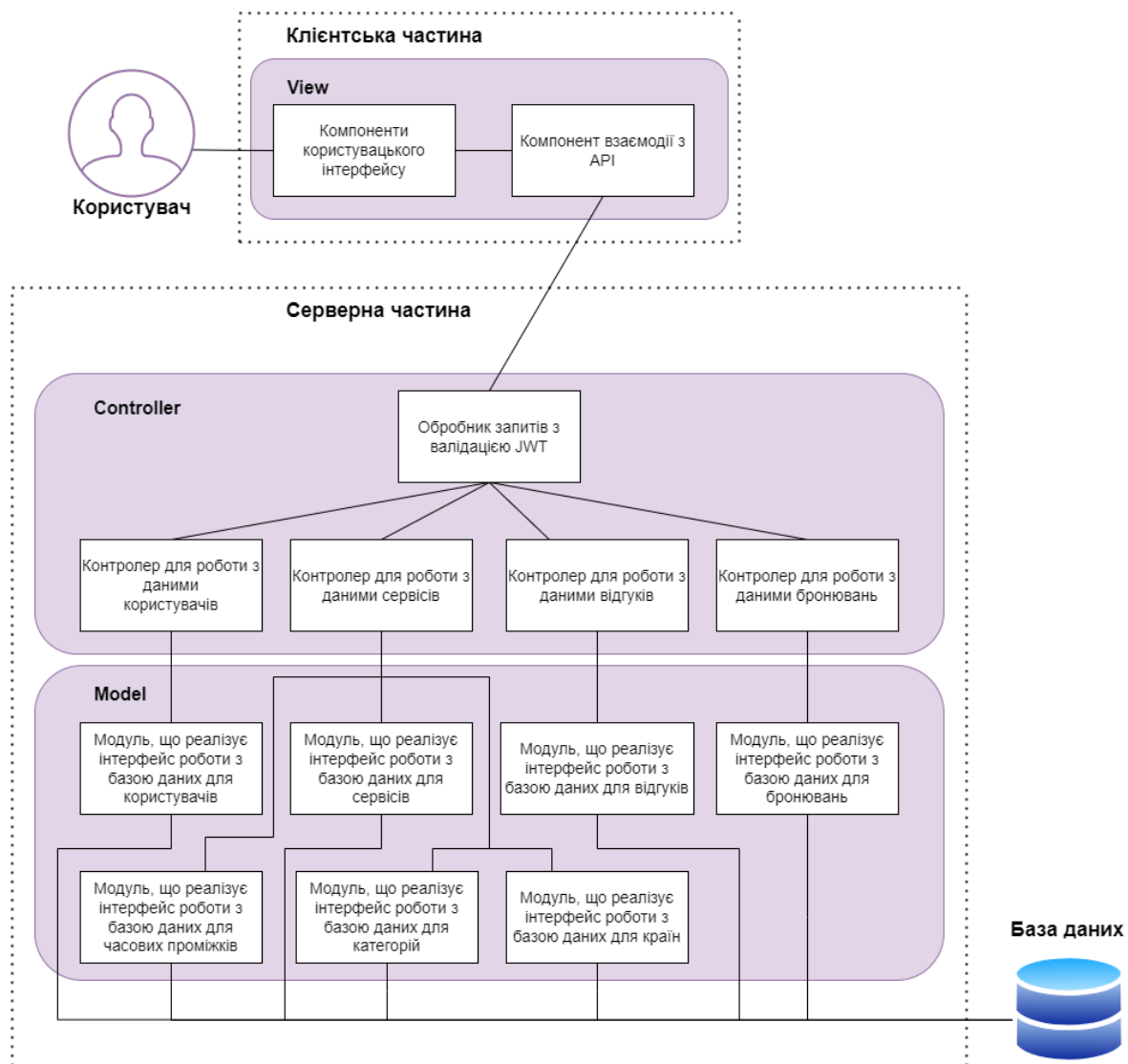
ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ



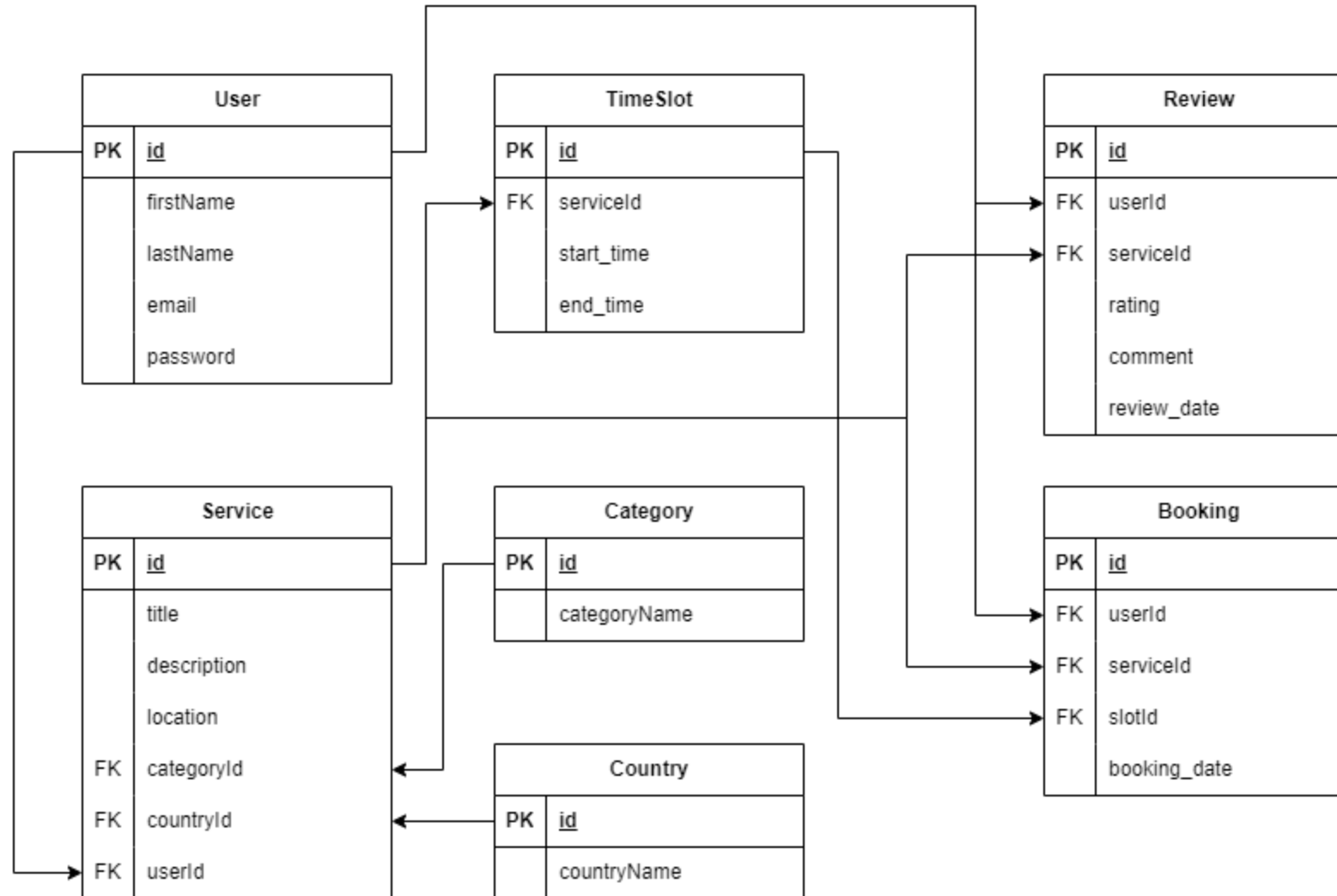
JavaScript



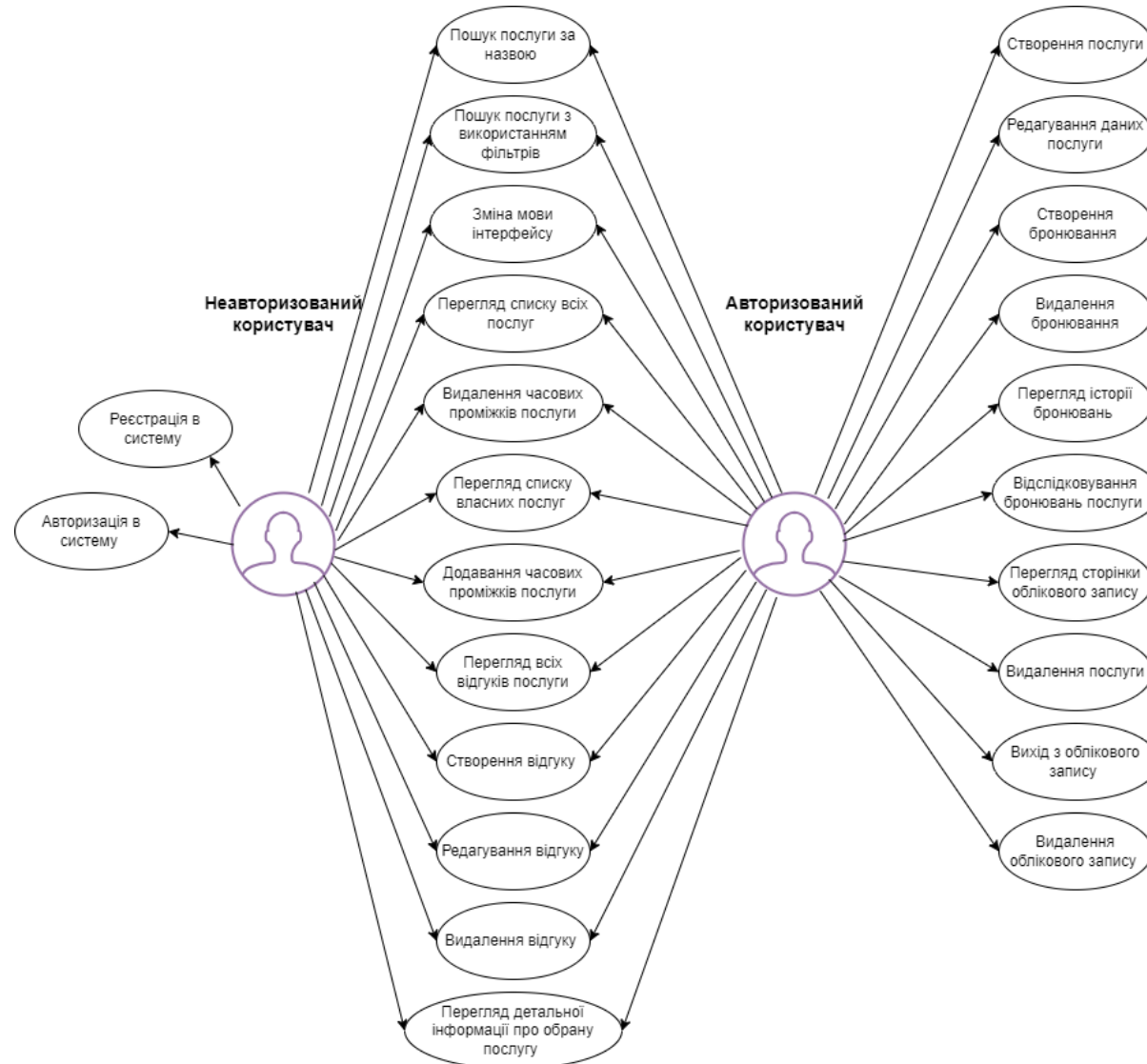
УЗАГАЛЬНЕНА АРХІТЕКТУРА ВЕБЗАСТОСУНКУ



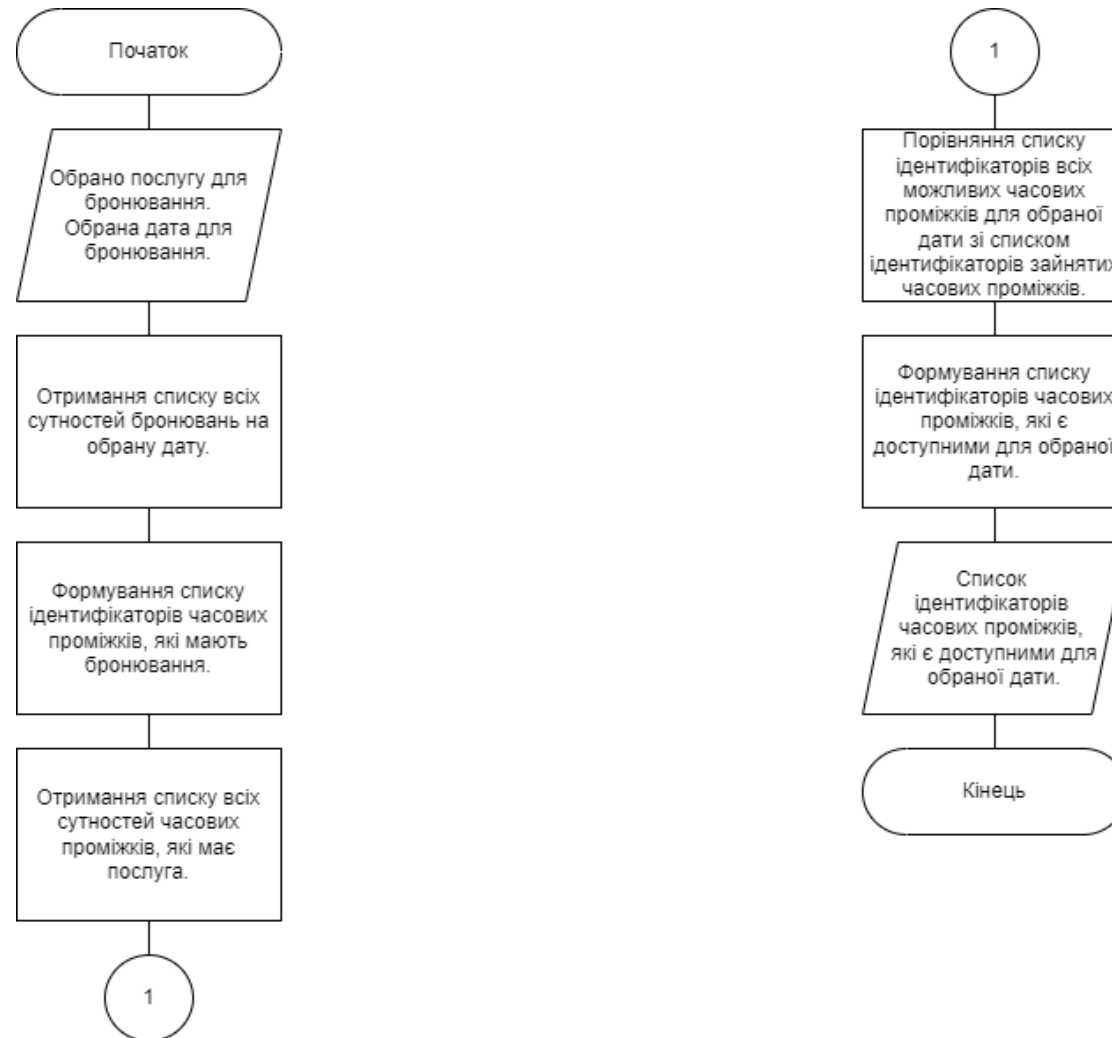
СТРУКТУРА БАЗИ ДАНИХ



ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ




АЛГОРИТМ ПОШУКУ ВІЛЬНИХ ЧАСОВИХ ПРОМІЖКІВ ПОСЛУГИ ДЛЯ ОБРАНОЇ ДАТИ



ПРИКЛАД РОБОТИ ВЕБЗАСТОСУНКУ




Registration Form



[Go to Main Page](#)

Create Service All Services Change Language My Account Sign Out



FitForce Personal Training

Description: FitForce Personal Training provides personalized fitness coaching to help clients reach their health goals. Their certified trainers create tailored workout plans and provide nutrition advice. Whether clients are looking to lose weight, build muscle, or improve their general fitness, FitForce designs programs suitable for every level. The training sessions can be one-on-one or in small groups for a more affordable option. With a focus on motivation and support, FitForce empowers clients to achieve lasting results.

Location: 567 Willow Street, Sydney

Category: Fitness Trainer

Country: Australia

ПРИКЛАД РОБОТИ ВЕБЗАСТОСУНКУ



Navigation bar with buttons: Create Service, All Services, Change Language, My Account, Sign Out

My Bookings

EcoClean Home Services

EcoClean Home Services provides environmentally friendly home cleaning solutions. Their trained staff use non-toxic, biodegradable products that are safe for families and pets. They offer customized cleaning packages based on the client's needs, including deep cleaning, regular maintenance, and move-in/move-out services. With a focus on sustainability and customer satisfaction, EcoClean ensures homes are spotless while being eco-conscious. They also provide same-day and emergency cleaning options.

Location: 321 Spruce Drive, Berlin

Date: 2024-03-13

Time Slot: 15:00 - 17:00

[View Details](#)

[Delete](#)

SnapShots Photography Studio

SnapShots Photography Studio offers creative and professional photography services. They specialize in capturing life's special moments, from weddings and birthdays to corporate events. Their skilled photographers work closely with clients to understand their vision, ensuring each photo tells a story. They also provide portrait sessions for families, couples, and individuals seeking high-quality headshots. The studio provides retouching and digital albums to preserve memories beautifully.

Location: 301 Willow Street, Sydney

Date: 2024-05-31

Time Slot: 12:00 - 15:00

[View Details](#)

[Delete](#)

Service selection and booking interface. Includes dropdowns for 'Select Category' and 'Select Country', a calendar for May 2024, and a grid of service cards with 'Learn More' buttons.

MON	TUE	WED	THU	FRI	SAT	SUN
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

- GlowSkin Spa
- InkMaster Tattoo Studio
- FitForce Personal Training
- EliteCuts Hair Salon
- SnapShots Photography Studio
- UrbanClimb Fitness Center
- EcoClean Home Services
- Purrfect Paws Pet Grooming

Service reservation interface. Includes a calendar for May 2024, an 'Available Time' section with time slots, and a 'Confirm Reservation' button.

MON	TUE	WED	THU	FRI	SAT	SUN
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

Available Time:

- 11:00 - [Select](#)
- 15:00 - [Select](#)
- 19:00 - [Select](#)

[Confirm Reservation](#)

ПРИКЛАД РОБОТИ ВЕБЗАСТОСУНКУ



Navigation: [Create Service](#) [All Services](#) [Change Language](#) [My Account](#) [Sign Out](#)

My Services

Harmony Hub Studio

[Learn More](#) [Availability](#)

[Edit](#) [Delete](#)

Navigation: [Create Service](#) [All Services](#) [Change Language](#) [My Account](#) [Sign Out](#)

User Profile

First Name: Viktoriia
Last Name: Sirenko
Email: vika_sirenko@gmail.com

[Edit Profile](#) [Delete Profile](#)

[View My Services](#) [View My Reservations](#)

Add Review

Rating:

Comment:

★★★★★ 07/05/2024

coaches are experienced. You can ask as much questions about sport and diet equipment for all levels.

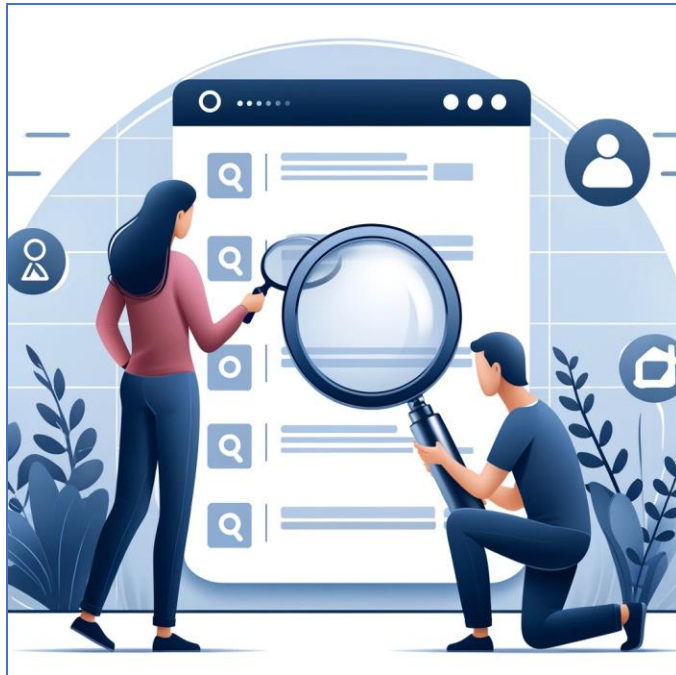
[Edit](#) [Delete](#)

ДИМОВЕ ТЕСТУВАННЯ ВЕБЗАСТОСУНКУ



1. Тестування процесу реєстрації / авторизації.
2. Тестування роботи фільтрів за параметрами.
3. Тестування процесу створення, редагування та видалення послуг.
4. Тестування можливості резервування послуги.
5. Тестування функціональності керування профілем користувача.
6. Тестування процесу створення, редагування та видалення відгуків.

НАПРЯМКИ ПОДАЛЬШОГО РОЗВИТКУ ПРОЄКТУ



Покращення пошукової
функціональності



Створення мобільної версії
застосунку



Розширення списку мов
користувацького
інтерфейсу

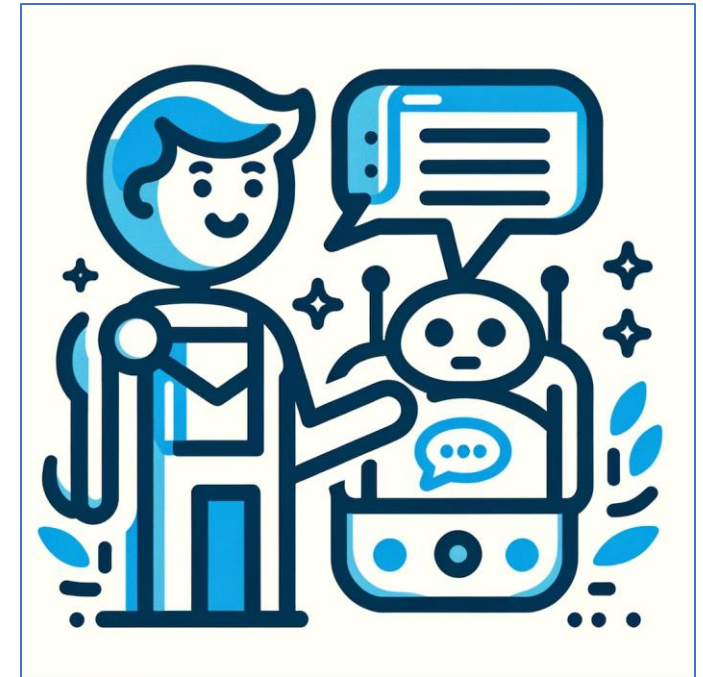
НАПРЯМКИ ПОДАЛЬШОГО РОЗВИТКУ ПРОЄКТУ



Інтеграція із зовнішніми
API



Вдосконалення системи
безпеки та
конфіденційності



Інтеграція чат-платформи



ВИСНОВКИ

1. Проведено аналіз існуючих рішень та ідентифіковано переваги та недоліки кожного з них.
2. Визначено та описано функціональні вимоги до програмного забезпечення.
3. Запропоновано узагальнену архітектуру програмного додатку та структуру бази даних.
4. Розроблено алгоритми функціонування застосунку, зокрема алгоритм пошуку вільних часових проміжків послуги для обраної дати.
5. Реалізовано програмне забезпечення для пошуку та резервування послуг.
6. Проведено тестування додатку.
7. Визначено можливості для подальшого розвитку.



Дякую за увагу!

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Євгенія СУЛЕМА

“ ___ ” _____ 2023 р.

ВЕБЗАСТОСУНОК ДЛЯ ПОШУКУ ТА
РЕЗЕРВУВАННЯ ПОСЛУГ

Програма та методика тестування

ДП.045440-04-51

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Тетяна ЗАБОЛОТНЯ

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Вікторія СІРЕНКО

2023

ЗМІСТ

1. ОБ'ЄКТ ВИПРОБУВАНЬ	3
2. МЕТА ТЕСТУВАННЯ.....	3
3. МЕТОДИ ТЕСТУВАННЯ.....	3
4. ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ.....	4

1. ОБ'ЄКТ ВИПРОБУВАНЬ

Вебзастосунок для пошуку та резервування послуг, представляє собою сайт, розроблений за допомогою мови програмування Python.

2. МЕТА ТЕСТУВАННЯ

Під час тестування необхідно переконатись у наступному:

1. Функціональність елементів інтерфейсу програмного забезпечення.
2. Можливість створення, редагування та відміни резервувань користувачами.
3. Можливість створення, модифікації та видалення створених користувачем послуг.
4. Можливість створення, редагування та видалення відгуків про послугу.
5. Можливість створення, редагування та видалення аккаунту користувача.
6. Відповідність користувацького інтерфейсу вимогам Технічного завдання.

3. МЕТОДИ ТЕСТУВАННЯ

В рамках тестування використовується підхід Gray Box Testing, при якому проводиться аналіз і програмного коду, і самого програмного забезпечення щодо їх відповідності функціональним вимогам. Цей процес відбувається на етапі системного тестування. Застосовуються різні типи тестів:

1. Функціональне тестування включає перевірку інтеграції компонентів програми та взаємодію користувача з інтерфейсом.
2. Оцінка продуктивності програмного забезпечення, включно з тестуванням стабільності та навантажувальним тестуванням.
3. Перевірка інтерфейсу для забезпечення оптимальної взаємодії користувача з програмою.

4. ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Тестування виконується засобами інструментарію SpecFlow.
Працездатність web-ресурсу перевіряється шляхом:

- 1) динамічного ручного тестування – введенням граничних та недопустимих значень в поля, які можна редагувати;
- 2) динамічного ручного тестування на відповідність функціональним вимогам;
- 3) статичного тестування коду;
- 4) тестування вебзастосунку в різних веббраузерах;
- 5) тестування при максимальному навантаженні;
- 6) тестування стабільності роботи при різних умовах;
- 7) тестування зручності використання;
- 8) тестування інтерфейсу.

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Євгенія СУЛЕМА

“ ___ ” _____ 2024 р.

ВЕБЗАСТОСУНОК ДЛЯ ПОШУКУ ТА
РЕЗЕРВУВАННЯ ПОСЛУГ
Керівництво користувача

ДП.045440-05-34

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Тетяна ЗАБОЛОТНЯ

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Вікторія СІРЕНКО

2024

ЗМІСТ

1. Опис структури вебзастосунку	3
2. Процедура авторизації та реєстрації користувача	4
3. Головна сторінка з пошуковою панеллю.....	5
4. Процес створення послуги	7
5. Процес створення робочих годин послуги	8
6. Сторінка для перегляду списку всіх послуг з елементами фільтрації.....	9
7. Сторінка для перегляду детальної інформації про послугу зі списком відгуків.....	9
8. Сторінка для редагування відгуку.....	11
9. Процес резервування послуги.....	12
10. Сторінка профілю авторизованого користувача.....	13
11. Сторінка редагування профілю авторизованого користувача.....	14
12. Сторінка перегляду власних послуг.....	15
13. Сторінка для редагування послуги та часових проміжків.....	16
14. Сторінка перегляду зайнятості власної послуги.....	17
15. Сторінка перегляду власних резервувань.....	17

1. Опис структури вебзастосунку

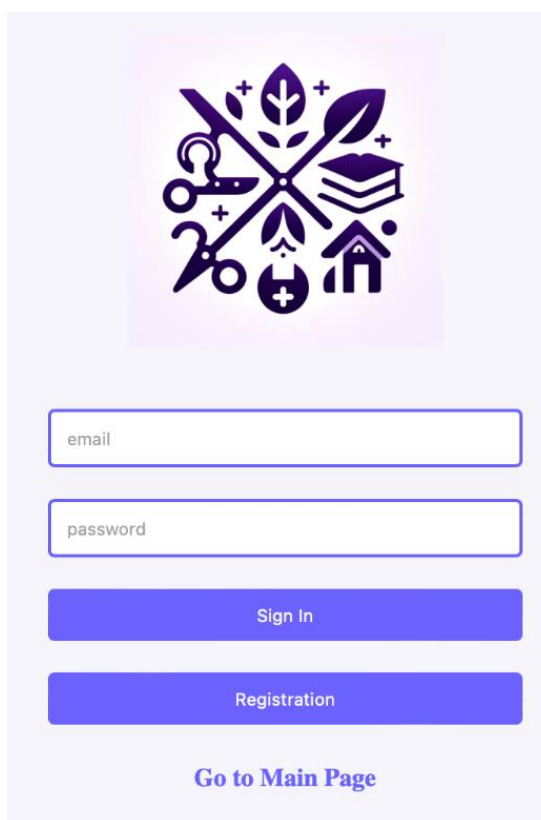
Програмне забезпечення для пошуку та резервування послуг розроблено у вигляді односторінкового вебзастосунку.

У системі наявні такі вебсторінки:

1. Сторінка авторизації та реєстрації.
2. Головна сторінка з пошуковою панеллю.
3. Сторінка для створення послуги.
4. Сторінка для створення робочих годин послуги (часових проміжків).
5. Сторінка для перегляду списку всіх послуг з елементами фільтрації.
6. Сторінка для перегляду детальної інформації про послугу зі списком відгуків.
7. Сторінка для редагування відгуку.
8. Сторінка для резервування послуги.
9. Сторінка профілю авторизованого користувача.
10. Сторінка редагування профілю авторизованого користувача.
11. Сторінка перегляду власних послуг.
12. Сторінка для редагування послуги та часових проміжків.
13. Сторінка перегляду зайнятості власної послуги.
14. Сторінка перегляду власних резервувань.

2. Процедура авторизації та реєстрації користувача

При використанні вебзастосунку користувач одразу потрапляє на сторінку авторизації (рис. 1). Залежно від наявності облікового запису, користувач обирає, що йому потрібно зробити: зареєструватися або увійти в систему.



The image shows a login interface. At the top center is a purple circular logo containing several white icons: a leaf, a book, a house, a person, and a pair of scissors. Below the logo are two white input fields with purple borders. The first field is labeled 'email' and the second is labeled 'password'. Underneath these fields are two purple buttons with white text: 'Sign In' and 'Registration'. At the bottom of the form is a purple link labeled 'Go to Main Page'.

Рис. 1. Сторінка авторизації

Якщо користувач має обліковий запис, то для входу в систему необхідно ввести електронну пошту та пароль та натиснути «Sign In» (рис. 2). Поле введення електронної пошти автоматично перевіряється на правильність даних, і у випадку помилкового вводу користувач отримує відповідне повідомлення (рис. 3).

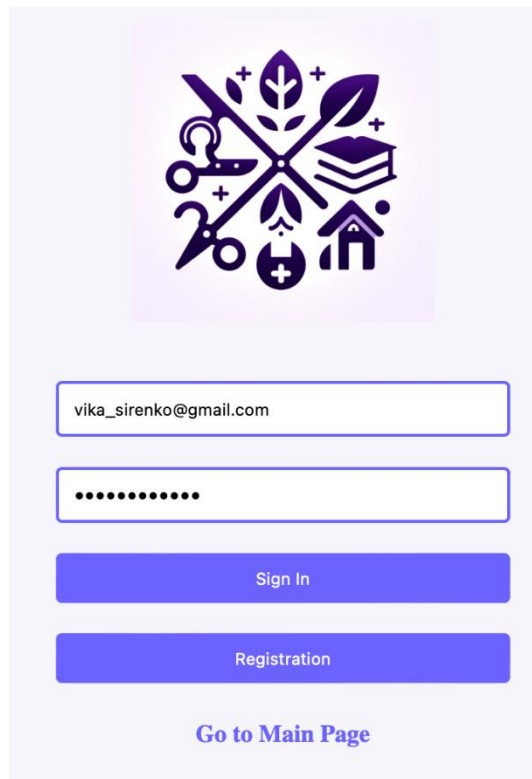


Рис. 2. Сторінка авторизації з заповненими полями

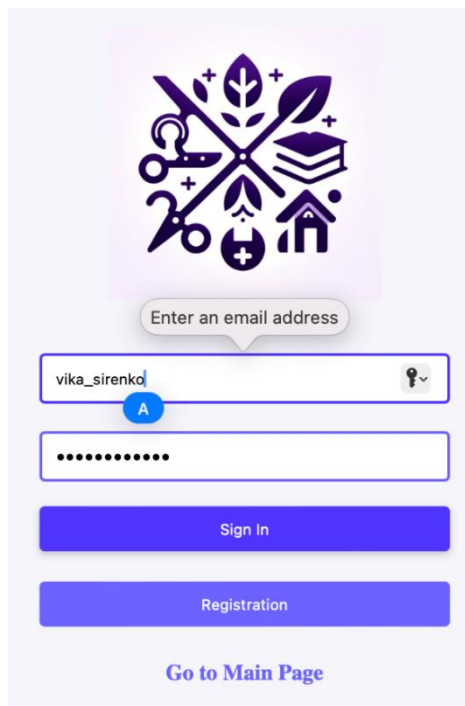
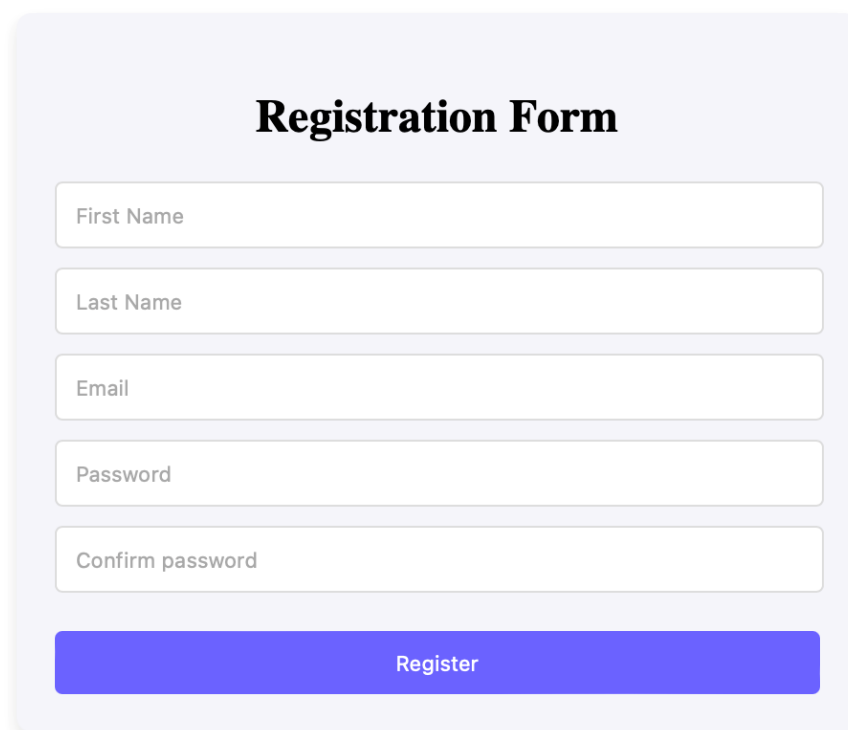


Рис. 3. Сторінка авторизації з прикладом повідомлення про помилку

Якщо користувач не має облікового запису, то для реєстрації необхідно натиснути «Registration», після чого система перенаправить користувача на сторінку реєстрації (рис. 4). Користувач має заповнити всі поля, вказавши: ім'я, прізвище, електронну адресу, пароль та підтвердження паролю.

Якщо користувач спробує зареєструватись, але не всі поля будуть заповнені, то користувач отримає повідомлення про це. Також поля введення автоматично перевіряються на правильність даних, і у випадку помилкового вводу або невідповідності паролю та підтвердження паролю користувач отримує відповідне повідомлення.



The image shows a registration form titled "Registration Form". It contains five input fields: "First Name", "Last Name", "Email", "Password", and "Confirm password". Below the fields is a blue button labeled "Register".

Рис. 4. Сторінка реєстрації

Після успішної реєстрації користувач отримує повідомлення про те, що реєстрація пройшла успішно.

Також користувач має варіант продовжити використання додатку без створення облікового запису, натиснувши «Go to Main Page», але більшість можливостей додатку в такому випадку для користувача будуть недоступні.

3. Головна сторінка з пошуковою панеллю

Після успішної авторизації або реєстрації або неавторизованого входу користувач переходить на головну сторінку з пошуковою панеллю (рис. 5).

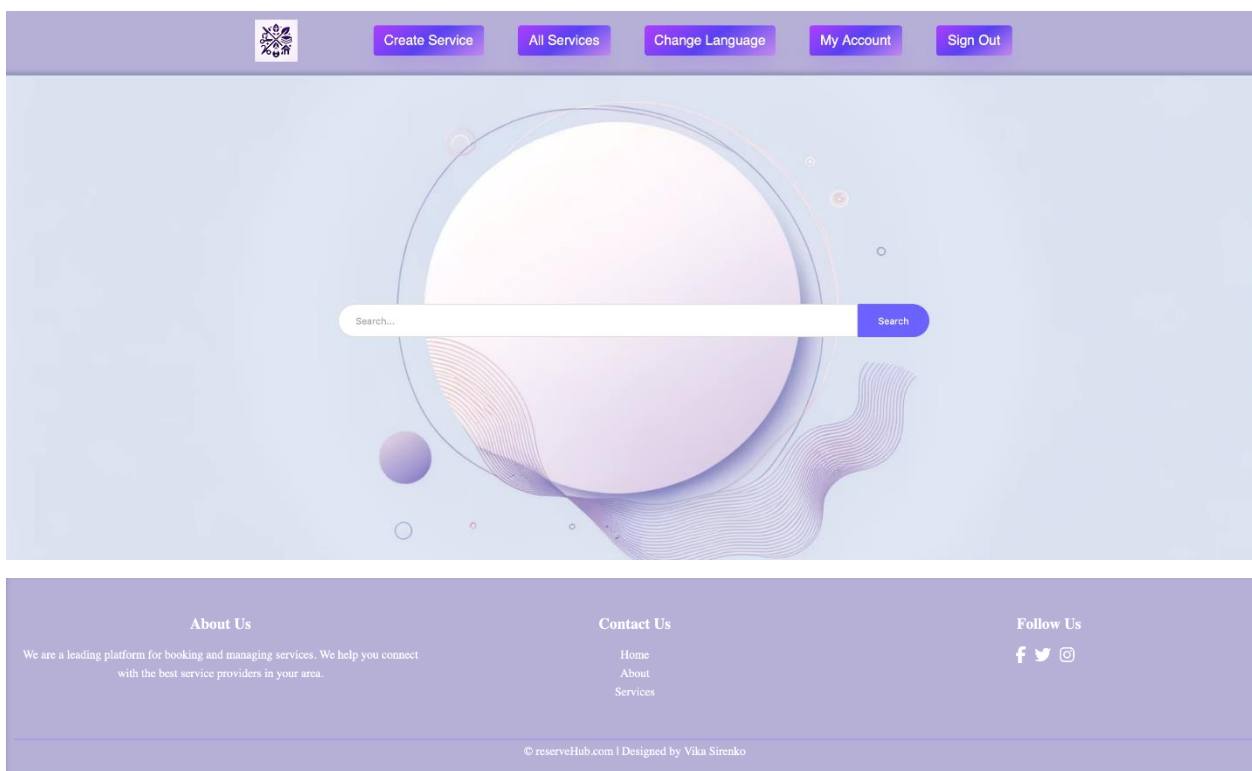
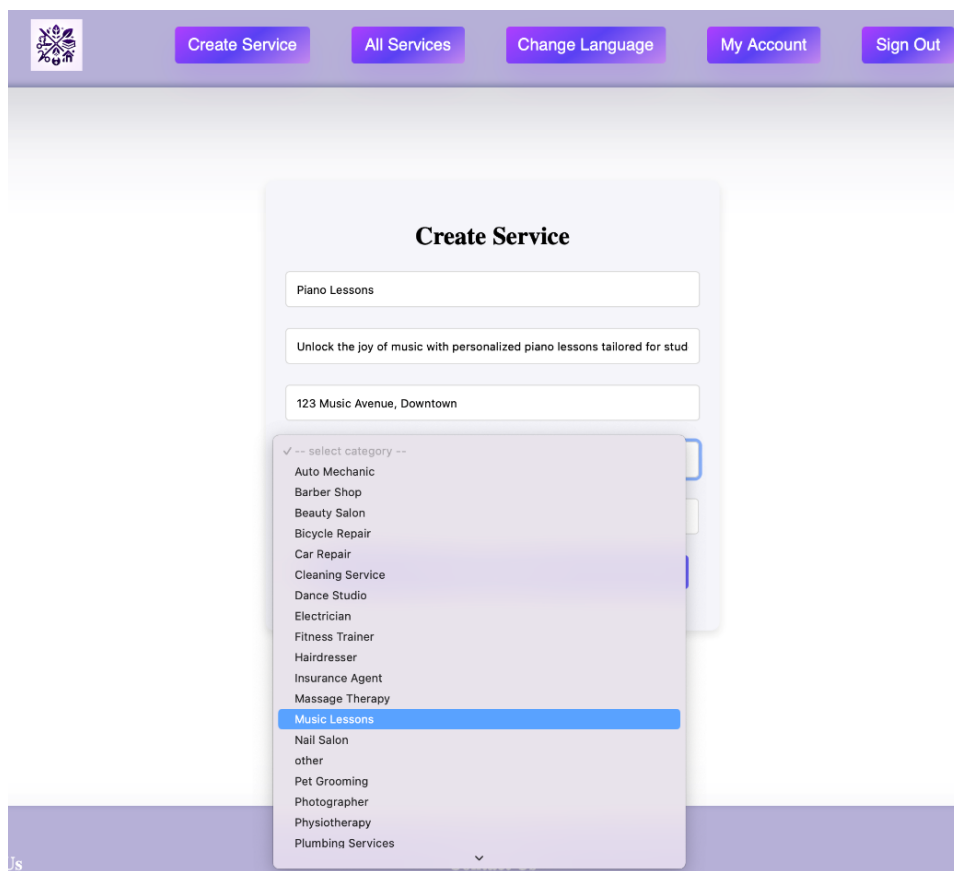


Рис. 5. Головна сторінка з пошуковою панеллю

На цій сторінці користувач може здійснювати пошук послуг за назвою ввівши пошуковий запит у пошукову панель.

4. Процес створення послуги

Авторизований користувач має можливість створювати послуги. Для цього необхідно перейти на вкладку «Create Service», після чого користувач перейде на сторінку створення послуги, де необхідно заповнити поля: заголовок, опис, місцезнаходження та обрати зі списку категорію послуги та країну (рис. 6).



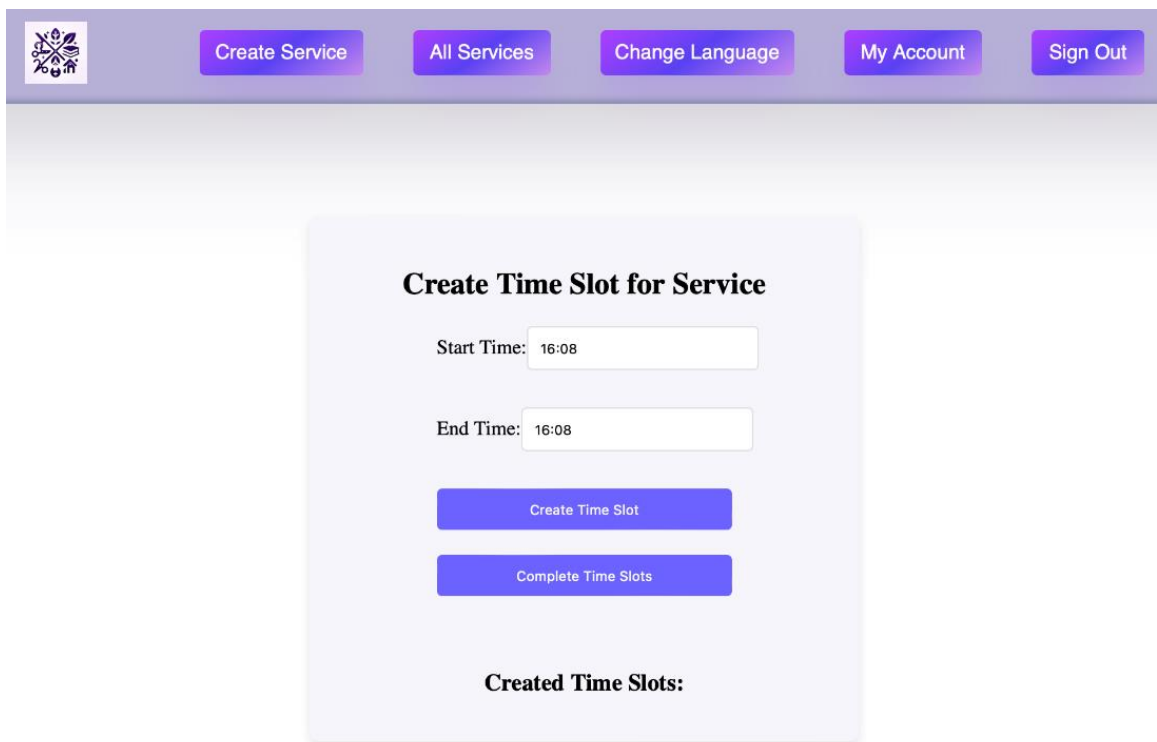
The screenshot shows a web interface for creating a service. At the top, there is a navigation bar with a logo on the left and five buttons: 'Create Service', 'All Services', 'Change Language', 'My Account', and 'Sign Out'. The main content area is titled 'Create Service' and contains four input fields: a text field with 'Piano Lessons', a text area with 'Unlock the joy of music with personalized piano lessons tailored for stud', and a text field with '123 Music Avenue, Downtown'. Below these fields is a dropdown menu with a checkmark and the text '-- select category --'. The menu lists various service categories, with 'Music Lessons' highlighted in blue. Other categories include Auto Mechanic, Barber Shop, Beauty Salon, Bicycle Repair, Car Repair, Cleaning Service, Dance Studio, Electrician, Fitness Trainer, Hairdresser, Insurance Agent, Massage Therapy, Nail Salon, other, Pet Grooming, Photographer, Physiotherapy, and Plumbing Services.

Рис. 6. Сторінка створення послуги з правильно заповненими полями

Заповнивши всі поля, необхідно натиснути «Create Service» після чого з’явиться повідомлення про успішне створення сервісу.

5. Процес створення робочих годин послуги

Після успішного створення послуги користувач переходить на сторінку додавання робочих годин (часових проміжків) послуги (рис. 7). Користувач має встановити початковий час та кінцевий час часового проміжку, далі натиснути «Create Time Slot». Після чого створений часовий проміжок має відобразитись в списку «Created Time Slots». Для завершення створення робочих годин послуги необхідно натиснути «Complete Time Slots».



Create Time Slot for Service

Start Time: 16:08

End Time: 16:08

Create Time Slot

Complete Time Slots

Created Time Slots:

Рис. 7. Сторінка створення часових проміжків

6. Сторінка для перегляду списку всіх послуг з елементами фільтрації

Перейшовши на вкладку «All Services», авторизовані та неавторизовані користувачі мають можливість переглядати список всіх послуг, а також

здійснювати фільтрування послуг за категорією, країною та бажаною датою резервування послуги (рис. 8).

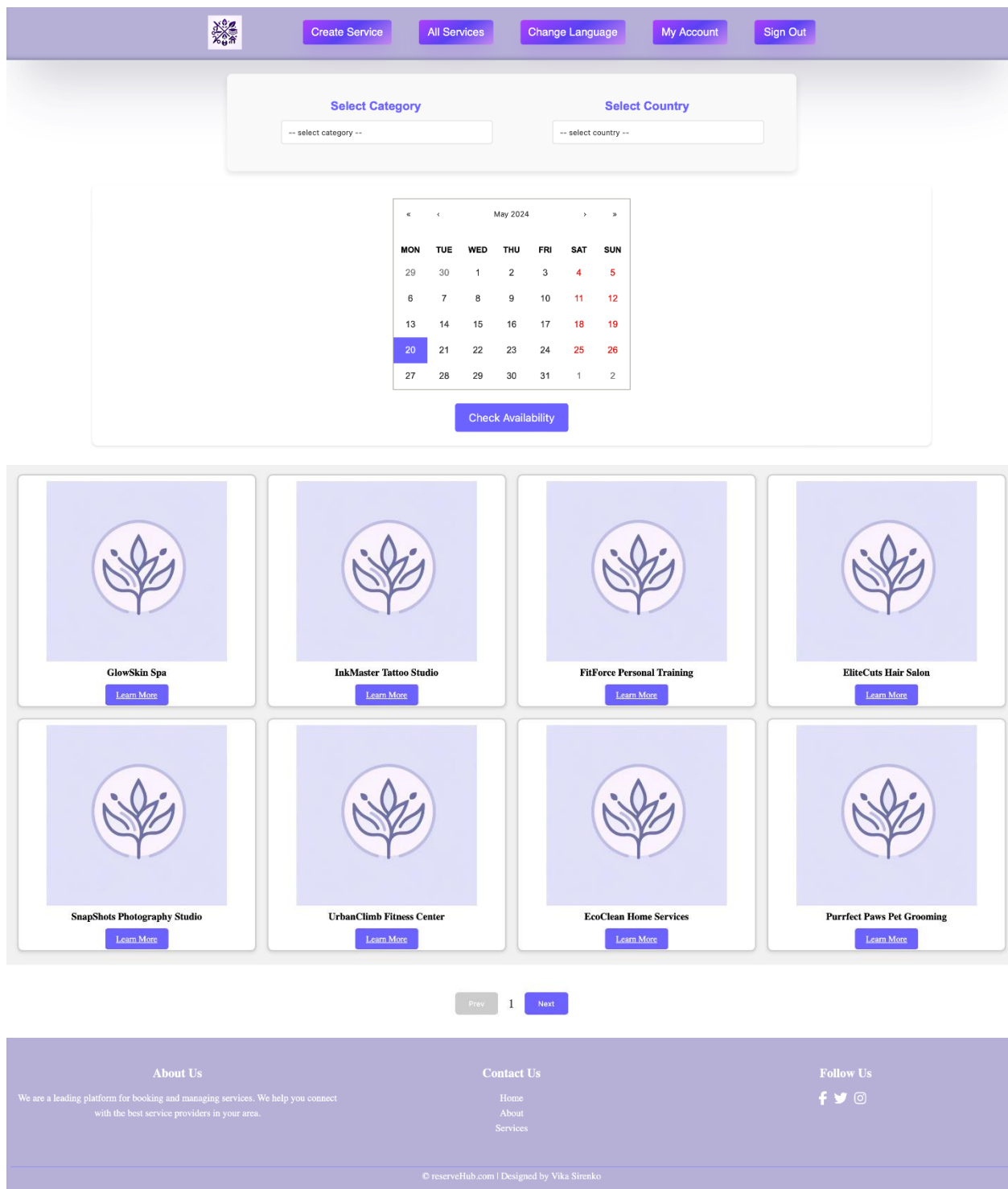


Рис. 8. Сторінка перегляду списку послуг з елементами фільтрації

7. Сторінка для перегляду детальної інформації про послугу зі списком відгуків

Авторизовані та неавторизовані користувачі можуть переглядати детальну інформацію про сервіс натиснувши «Learn More» на картці послуги. Система перейде на сторінку (рис. 9-10), де користувач зможе ознайомитись з назвою, описом, місцезнаходженням, категорією послуги, а також переглянути відгуки та залишити відгук. Для цього необхідно встановити рейтинг, написати коментар та натиснути «Submit Review». Система має оновити сторінку і коментар з'явиться в списку відгуків.

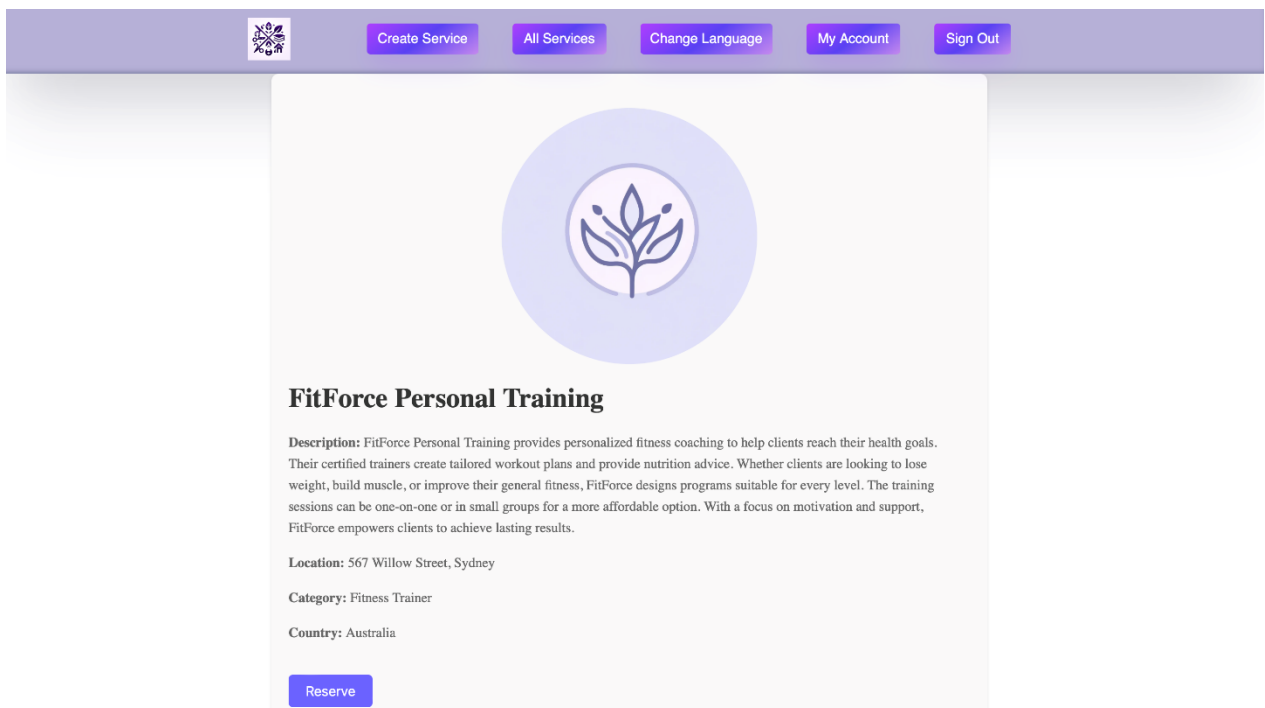


Рис. 9. Сторінка перегляду детальної інформації про послугу. Частина 1

The image shows two parts of a web interface. The top part is titled 'Add Review' and contains a 'Rating:' dropdown menu with the value '1' selected, a 'Comment:' text area, and a blue 'Submit Review' button. The bottom part is titled 'Reviews' and displays a review by 'David' with a 5-star rating and the date '07/05/2024'. The review text reads: 'The location is really good. And all coaches are experienced. You can ask as much questions about sport and diet as you want. There is a lot of different equipment for all levels.' Below the text are 'Edit' and 'Delete' buttons.

Рис. 10. Сторінка перегляду детальної інформації про послугу. Частина 2

8. Сторінка для редагування відгуку

Для редагування відгуку користувач має натиснути «Edit», далі система переходить на сторінку редагування відгуку (рис. 11). Після внесення змін необхідно натиснути «Update Review» і відгук буде оновлено.

The image shows the 'Edit Review' page. At the top, there is a navigation bar with a logo and buttons for 'Create Service', 'All Services', 'Change Language', 'My Account', and 'Sign Out'. The main content area is titled 'Edit Review' and contains a 'Rating:' dropdown menu with the value '4' selected, a 'Comment:' text area with the text 'Really good!', and a blue 'Update Review' button.

Рис. 11. Сторінка для редагування відгуку

9. Процес резервування послуги

Для резервування послуги спочатку користувач має натиснути «Reserve» на сторінці послуги, яку хоче зарезервувати. Потім система перейде до сторінки резервування послуги.

Користувач має обрати бажану дату та часовий проміжок, після чого з'явиться кнопка «Confirm Reservation» (рис. 12), натиснувши яку, має з'явитися повідомлення про успішне бронювання послуги.

The screenshot displays the 'Reserve Service' interface. At the top, there is a navigation bar with buttons for 'Create Service', 'All Services', 'Change Language', 'My Account', and 'Sign Out'. The main heading is 'Reserve Service'. Below it is a calendar for May 2024. The calendar shows the following dates: 29, 30, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 (highlighted in yellow), 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 (highlighted in blue), 1, 2. Below the calendar, there is a section titled 'Available Time:' with three time slots, each with a 'Select' button: 'Time: 09:00 - 11:00', 'Time: 12:00 - 15:00', and 'Time: 16:00 - 19:00'. At the bottom center, there is a large blue button labeled 'Confirm Reservation'.

Рис. 12. Сторінка резервування послуги з прикладом обрання дати та часового проміжку

10. Сторінка профілю авторизованого користувача

Авторизований користувач має можливість переглядати власний профіль (рис. 13). Для цього необхідно перейти на вкладку «My Account».

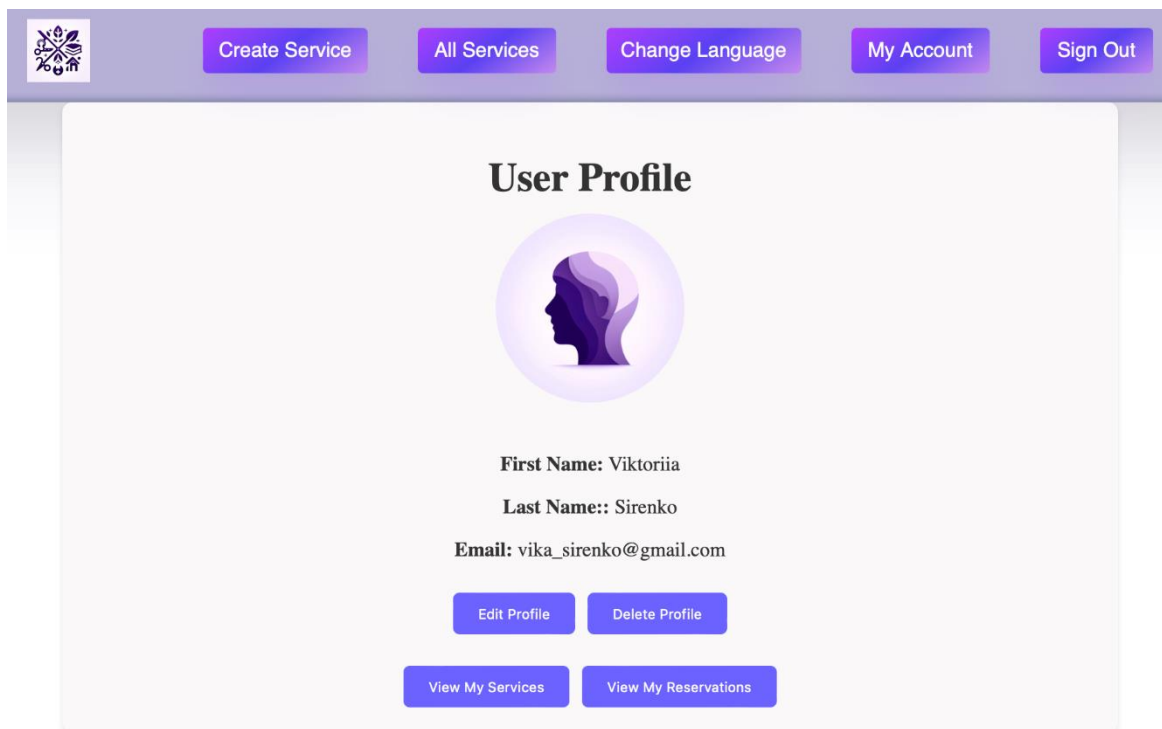


Рис. 13. Сторінка профілю авторизованого користувача

11. Сторінка редагування профілю авторизованого користувача

Також авторизований користувач може редагувати дані власного профілю. Для цього необхідно натиснути «Edit Profile». Система перейде на сторінку редагування даних (рис. 14), де користувач зможе змінити ім'я, прізвище, електронну адресу та пароль. Для внесення змін необхідно натиснути «Update Profile». Після внесення змін користувач має знову авторизуватися в системі.

Header: Create Service, All Services, Change Language, My Account, Sign Out

Edit Profile

First Name: Viktoria

Last Name: Sirenko

Email: vika_sirenko@gmail.com

Password (leave blank to keep the same): Enter new password

Confirm New Password: Confirm new password

Update Profile

Рис. 14. Сторінка редагування профілю авторизованого користувача

12. Сторінка перегляду власних послуг

Авторизований користувач має можливість переглядати список створених послуг (рис. 15). Для цього необхідно натиснути «View My Services».

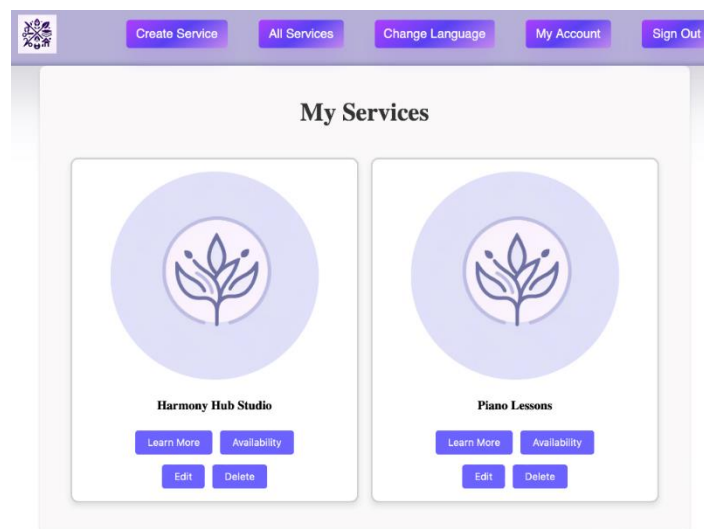


Рис. 15. Сторінка перегляду створених послуг авторизованого користувача

13. Сторінка для редагування послуги та часових проміжків

Користувач може редагувати інформацію про послугу та видаляти або додавати нові часові проміжки, натиснувши «Edit», ситсема перейде на сторінку внесення змін (рис. 16-17).

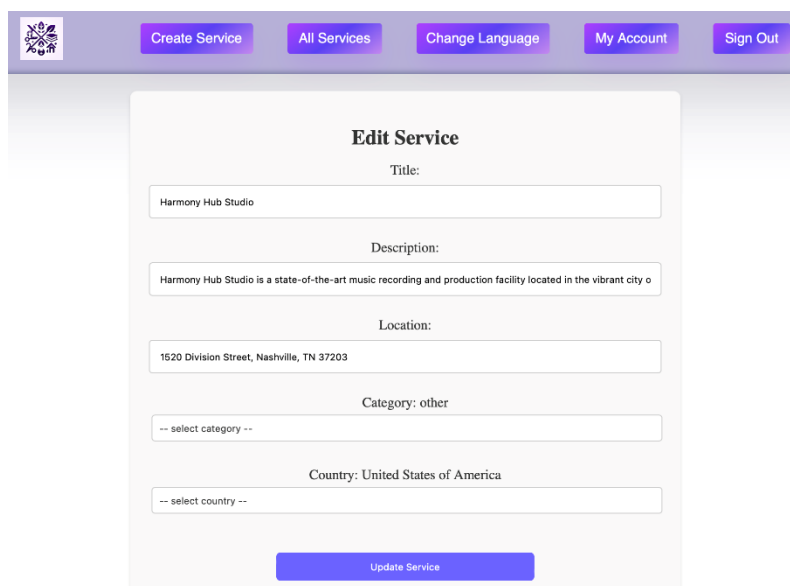


Рис. 16. Сторінка редагування даних послуги. Частина 1

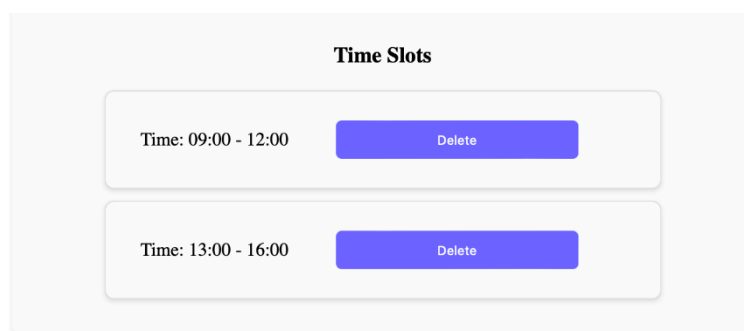


Рис. 17. Сторінка редагування даних послуги. Частина 2

Для видалення часових проміжків користувач має натиснути «Delete». Для збереження внесених змін користувач повинен натиснути «Update Service»,

після чого користувач отримає повідомлення з пропозицією додати нові часові проміжки. Якщо натиснути «Yes», то система перейде на сторінку додавання часових проміжків. При натисненні «No» система перейде на сторінку перегляду детальної інформації про послугу.

14. Сторінка перегляду зайнятості власної послуги

Авторизований користувач має можливість переглядати зайнятість створеної послуги (рис. 18). Для цього необхідно натиснути «Availability».

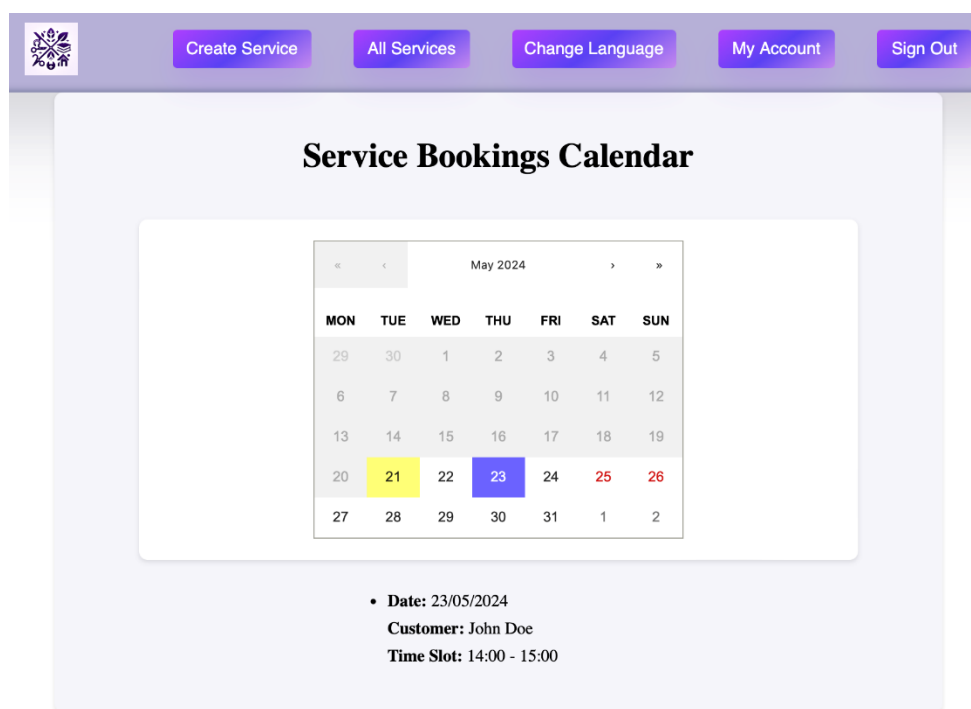


Рис. 18. Сторінка перегляду зайнятості послуги

15. Сторінка перегляду власних резервувань

Авторизований користувач має можливість перегляду та відміни власних резервувань. Для цього необхідно натиснути «View My Reservations» і система

перейде на сторінку резервувань (рис. 19). Для видалення резервування необхідно натиснути «Delete».

My Bookings

EcoClean Home Services

EcoClean Home Services provides environmentally friendly home cleaning solutions. Their trained staff use non-toxic, biodegradable products that are safe for families and pets. They offer customized cleaning packages based on the client's needs, including deep cleaning, regular maintenance, and move-in/move-out services. With a focus on sustainability and customer satisfaction, EcoClean ensures homes are spotless while being eco-conscious. They also provide same-day and emergency cleaning options.

Location: 321 Spruce Drive, Berlin

Date: 2024-03-13

Time Slot: 15:00 - 17:00

[View Details](#) [Delete](#)

SnapShots Photography Studio

SnapShots Photography Studio offers creative and professional photography services. They specialize in capturing life's special moments, from weddings and birthdays to corporate events. Their skilled photographers work closely with clients to understand their vision, ensuring each photo tells a story. They also provide portrait sessions for families, couples, and individuals seeking high-quality headshots. The studio provides retouching and digital albums to preserve memories beautifully.

Location: 301 Willow Street, Sydney

Date: 2024-05-31

Time Slot: 12:00 - 15:00

[View Details](#) [Delete](#)

Рис. 19. Сторінка перегляду резервувань авторизованого користувача