

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра прикладних інформаційних систем

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА
НА ТЕМУ:**

Веб-сервіс пошуку та бронювання авіабілетів

Галузь знань 12 «Інформаційні технології»

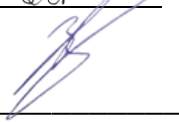
Спеціальність 122 «Комп'ютерні науки»

Освітня програма «Прикладне програмування»

Освітній рівень: бакалавр

Виконала: студентка 4 курсу, групи ПП-41


Богак Ю. М. 
(прізвище та ініціали)

Керівник Зосімов В.В. 
(прізвище та ініціали)

Д.Т.Н., доц.
(науковий ступінь, звання)

Унікальність тексту 91%

Випускна кваліфікаційна робота бакалавра допущена до захисту
рішенням кафедри *прикладних інформаційних систем*
Протокол №14 від 23 травня 2023 р.

зав. кафедри  проф. Плескач В.Л.

Київ – 2023

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

№з/п	Назва етапів кваліфікаційної роботи бакалавра	Термін виконання етапів кваліфікаційної роботи бакалавра	Відмітка про виконання
1.	Вибір теми та наукового керівника кваліфікаційної роботи бакалавра	14.10.2022	Виконано
2.	Видача завдання кваліфікаційної роботи бакалавра	24.10.2022	Виконано
3.	Настановча групова співбесіда з питань кваліфікаційної роботи бакалавра	31.10.2022	Виконано
4.	Затвердження плану кваліфікаційної роботи бакалавра	01.11.2022	Виконано
5.	Підбір та вивчення літературних та інших джерел з теми дослідження	08.11.2022	Виконано
6.	Підготовка і подання науковому керівнику першого варіанту I розділу роботи	21.12.2022	Виконано
7.	Підготовка і подання науковому керівнику першого варіанту II розділу роботи	31.01.2023	Виконано
8.	Підготовка і подання науковому керівнику першого варіанту III розділу роботи	30.03.2023	Виконано
9.	Подання роботи у першому варіанті	28.04.2023	Виконано
10.	Оформлення пояснювальної записки кваліфікаційної роботи бакалавра	03.05.2023	Виконано
11.	Подання кваліфікаційної роботи бакалавра на попередній захист	22.05.2023	Виконано
12.	Врахування зауважень керівника і подання роботи в остаточному варіанті (з відповідним висновком про допуск) на кафедрі	26.05.2023	Виконано
13.	Затвердження роботи в цілому (підготовка письмового відгуку керівника, письмова рецензія на бакалаврську роботу)	12.06.2023	Виконано
14.	Захист кваліфікаційної роботи бакалавра	26.06.2023	Виконано

Здобувач вищої освіти


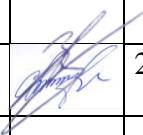

 (підпис)

Керівник


 (підпис)

ВІДОМІСТЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Складові частини кваліфікаційної роботи	Обсяг, арк.
Титульний аркуш	1
Календарний план кваліфікаційної роботи	1
Відомість кваліфікаційної роботи	1
Анотація	1
Анотація (іноземною мовою-англійською)	2
Зміст	2
Перелік скорочень, умовних позначень, термінів	1
Вступ	2
1	18
2	11
3	12
Висновки	1
Перелік використаних джерел	3
Додатки	4

				ДП ХХХХ 00.000.00		
	ПІБ	Підп.	Дата			
Розробн.	Богак Ю.М.			Відомість кваліфікаційної роботи	Лист	Листів
Керівн.	Зосімов В.В.					
Н/контр.	Макаренко С.А.		26.05.2023			
Зав.каф.	Плескач В.Л.					

АНОТАЦІЯ

Кваліфікаційна робота: 60с., 31 рис., 45 джерел, 3 дод.

Кваліфікаційна робота присвячена розробці веб-сервісу для пошуку та бронювання авіабілетів.

Метою кваліфікаційної роботи є ефективний пошук та бронювання авіабілетів на основі веб-сервісу.

Для досягнення поставленої мети треба вирішити такі **завдання**:

- Провести детальний аналіз потреб та вимог користувачів, враховуючи їхні очікування щодо функціональності та зручності використання.
- Вибрати відповідні технології для реалізації веб-сервісу, зокрема фреймворк, базу даних, мову програмування та інші необхідні компоненти.
- Створити архітектурний план веб-сервісу, включаючи модулі, маршрутизацію запитів, обробку даних та взаємодію з зовнішніми системами.
- Розробити логіку пошуку та бронювання авіабілетів, включаючи обробку запитів користувачів, валідацію даних, підключення до постачальників даних та відображення результатів.

Об'єкт дослідження.

Процеси ефективного пошуку та бронювання авіабілетів для користувачів на основі роботи веб-сервісу.

Предмет дослідження.

Програмно-технічні, організаційні засади, принципи, підходи щодо побудови веб-сервісу онлайн пошуку та бронювання авіабілетів.

Методи дослідження.

Аналіз та оцінка існуючих веб-сервісів пошуку та бронювання авіабілетів на основі реальних даних, включаючи рейтинги, відгуки та інші метрики та вивчення методів їх побудови. Проектування, розробки та побудова власного веб-сервісу, порівняння, що застосовано для аналізу наявних ресурсів.

Ключові слова: веб-сервіс, пошук, бронювання, авіабілет, технологія FastAPI, Angular, вебскрапінг

ABSTRACT

Thesis: 60 pages, 31 figures, 45 sources, 3 appendices.

The thesis is dedicated to the development of a web service for searching and booking airline tickets.

The purpose of the thesis is to create a convenient and efficient tool for users that allows them to easily find and book airline tickets at the best prices.

To achieve this goal, the following **tasks** need to be solved:

- Conduct a detailed analysis of user needs and requirements, taking into account their expectations regarding functionality, usability, and security.
- Choose appropriate technologies for implementing the web service, including framework, database, programming language, and other necessary components.
- Web service architecture development: Create an architectural plan for the web service, including a distributed system, modules, request routing, data processing, and interaction with external systems.
- Develop the logic for searching and booking airline tickets, including user request handling, data validation, integration with data providers, and result storage.

Object of study

The processes of the search and booking service for airline tickets for users.

Subject of study

Programmatic, technical, and organizational foundations, principles, and approaches to building a web service for online search and booking of airline tickets.

Research methods

Analysis and evaluation of existing web services for searching and booking airline tickets based on real data, including ratings, reviews, and other metrics, as well as studying the methods used to construct them. Design, development, and

construction of our own web service, with a comparison to the methods applied for analyzing existing resources.

Keywords: web service, search, booking, airline tickets, FastAPI, Angular, web scraping.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ	9
ВСТУП	10
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ТА ПРИНЦИПИ РОБОТИ ВЕБ-СЕРВІСІВ ПОШУКУ ТА БРОНЮВАННЯ АВІАБІЛЕТІВ	12
1.1	12
1.2	15
1.3	18
1.4	25
1.5	28
РОЗДІЛ 2 ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ ДЛЯ РОЗРОБКИ ВЕБ СЕРВІСІВ ПОШУКУ ТА БРОНЮВАННЯ АВІАБІЛЕТІВ	30
2.1	30
2.2	32
2.3	34
2.4	35
2.5	36
2.6	39
РОЗДІЛ 3 ПРОЕКТУВАННЯ, РОЗРОБЛЕННЯ, РЕАЛІЗАЦІЯ ВЕБ-СЕРВІСУ ПОШУКУ ТА БРОНЮВАННЯ АВІАБІЛЕТІВ	41
3.1	41
3.2	46
3.3	49
ВИСНОВОК	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	54
ДОДАТКИ	58
ДОДАТОК А	58
ДОДАТОК Б	59
ДОДАТОК В	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

КСБ – Комп'ютерна система бронювання;

API – Application Programming Interface;

HTTP – Hypertext Transfer Protocol;

SOAP – Simple Object Access Protocol;

REST – Representational State Transfer;

XML – eXtensible Markup Language;

API – Application Programming Interface;

JSON – JavaScript Object Notation;

ВСТУП

Сучасний розвиток технологій та Інтернету має значний вплив на різні сфери життя, включаючи туризм і подорожі. Віртуальний світ надає нам можливість отримувати інформацію, здійснювати покупки та резервування онлайн. Одним з найпопулярніших видів послуг, які надаються в Інтернеті, є пошук та бронювання авіабілетів.

Актуальність цієї теми зумовлена тим, що е-комерційні веб-сервіси пошуку та бронювання авіабілетів використовуються мільйонами людей по всьому світу для забезпечення зручності та ефективності у процесі планування та бронювання подорожей. Вони надають доступ до розкладів рейсів, цін, умов бронювання та інших важливих даних, що допомагають користувачам прийняти інформовані рішення щодо покупки авіабілетів. Ці сервіси сприяють розвитку глобального ринку авіап перевезень та стимулюють конкуренцію між авіакомпаніями, що призводить до покращення якості обслуговування та доступності для користувачів.

Метою кваліфікаційної роботи є ефективний пошук та бронювання авіабілетів на основі веб-сервісу.

Завдання дослідження:

- Провести детальний аналіз потреб та вимог користувачів, враховуючи їхні очікування щодо функціональності та зручності використання.
- Вибрати відповідні технології для реалізації веб-сервісу, зокрема фреймворк, базу даних, мову програмування та інші необхідні компоненти.
- Створити архітектурний план веб-сервісу, включаючи модулі, маршрутизацію запитів, обробку даних та взаємодію з зовнішніми системами.
- Розробити логіку пошуку та бронювання авіабілетів, включаючи обробку запитів користувачів, валідацію даних, підключення до постачальників даних та відображення результатів.

Об'єктом дослідження кваліфікаційної роботи бакалавра є процеси ефективного пошуку та бронювання авіабілетів для користувачів на основі

роботи веб-сервісу.

Предметом дослідження кваліфікаційної роботи бакалавра є програмно-технічні, організаційні засади, принципи, підходи щодо побудови веб-сервісу онлайн пошуку та бронювання авіабілетів.

Методи дослідження: Аналіз та оцінка існуючих веб-сервісів пошуку та бронювання авіабілетів на основі реальних даних, включаючи рейтинги, відгуки та інші метрики та вивчення методів їх побудови. Проектування, розробки та побудови власної системи е-комерції, порівняння, що застосовано для аналізу наявних ресурсів.

Практичне значення одержаних результатів полягає у створенні функціонального веб-сервісу пошуку та бронювання авіабілетів. Розроблений веб-сервіс надасть користувачам можливість швидкого та зручного пошуку авіабілетів за найкращими цінами. Користувачі зможуть з легкістю знайти бажані авіарейси та забронювати квитки в онлайн-режимі. Веб-сервіс значно спростить процес пошуку та бронювання авіабілетів, що дозволить користувачам значно скоротити час, необхідний для цих операцій. Вони зможуть швидко отримати доступ до різних варіантів авіарейсів, порівняти ціни та здійснити бронювання в зручний для них спосіб.

Структура роботи:

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, розподілених на підрозділи та висновку.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ТА ПРИНЦИПИ РОБОТИ ВЕБ-СЕРВІСІВ ПОШУКУ ТА БРОНЮВАННЯ АВІАБІЛЕТІВ

Веб-сервіси пошуку та бронювання авіабілетів є складовою частиною сучасної туристичної індустрії. Вони надають можливість користувачам швидко та зручно знаходити доступні авіарейси, порівнювати ціни, вибирати найзручніший маршрут і здійснювати бронювання квитків онлайн.

Основна мета веб-сервісів пошуку та бронювання авіабілетів полягає у забезпеченні користувачам доступу до широкого спектру інформації про авіарейси та їх ціни з різних авіакомпаній. Це дозволяє клієнтам здійснювати інформовані рішення щодо вибору та бронювання квитків.

1.1 Визначення веб-сервісів пошуку та бронювання авіабілетів

За останній час спостерігається значне зростання туристичного руху, що має прямий вплив на сферу транспорту і комунікацій. Збільшений попит на подорожі спонукає ці сфери стати основними користувачами інноваційних продуктів та інформаційно-комунікаційних технологій. Особливо важливими стали комп'ютерні системи бронювання та електронні системи інформації і комунікацій. Ріст кількості авіакомпаній, транспортних засобів і обсягів авіап перевезень призвів до потреби використання комп'ютерних систем бронювання, які стали незамінним інструментом для резервування авіаквитків. Використання таких систем дозволяє значно покращити обслуговування клієнтів, скорочуючи час, необхідний для оформлення квитків і забезпечуючи їх бронювання. Крім того, це сприяє підвищенню якості роботи персоналу авіакомпаній та ефективності їх діяльності.

Веб-сервіси пошуку та бронювання авіабілетів є інтерактивними онлайн-платформами, що надають користувачам можливість швидкого та зручного знаходження, порівняння та бронювання авіабілетів. Ці сервіси дозволяють користувачам здійснювати пошук доступних рейсів за різними критеріями, такими як дата вильоту, місце вильоту та призначення, кількість пасажирів

тощо.

Бронювання може мати різні визначення залежно від контексту. У широкому розумінні, це попереднє резервування певних ресурсів або послуг з метою їх подальшого використання в спеціальних ситуаціях або для конкретних осіб. Наприклад, це може включати резервування готельних номерів або квитків на певні дати для туристів або пасажирів.

Деякі вчені зосереджуються на спеціалізованому визначенні систем бронювання, яке стосується довготривалого співробітництва з туроператорами, турагентами та корпоративними клієнтами щодо послуг у сфері готелів. Це означає укладання вигідних договорів і базується на поняттях, таких як "продаж по запити" і "вільний продаж".

Останнім часом поширеність комп'ютерних систем бронювання зростає завдяки прогресу в галузі комп'ютерної техніки, доступу до мереж та використанню сучасних інформаційних технологій в різних економічних секторах. Комп'ютерна система бронювання (КСБ) є автоматизованою системою, яка містить інформацію про розклади авіапереvezень, наявність місць, тарифи та додаткові послуги. Вона дозволяє туристичним агентам здійснювати бронювання та оформлення квитків, роблячи ці можливості доступними для їх клієнтів (рисунок 1.1).

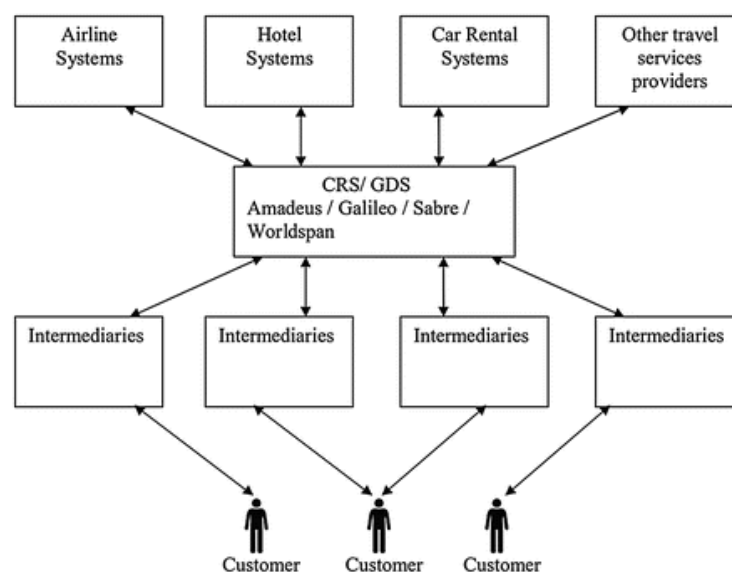


Рисунок 1.1 — Схема комп'ютерної системи бронювання (Computer Reservation System)

Веб-сервіси пошуку та бронювання авіабілетів взаємодіють з базами даних авіакомпаній та інших постачальників даних, щоб забезпечити користувачам актуальну та достовірну інформацію про рейси, ціни, наявність місць тощо. Крім того, ці сервіси надають можливість бронювання авіабілетів в режимі онлайн, забезпечуючи зручну форму для введення необхідних даних пасажира та обробку платежів.

Впровадження систем бронювання в електронній комерції дало багато переваг і поліпшень. Одна з основних переваг полягає у значному скороченні часу, необхідного для обслуговування клієнтів. Системи бронювання дозволяють здійснювати резервування онлайн, що дозволяє клієнтам швидко і зручно бронювати послуги без необхідності відвідувати фізичні офіси або здійснювати довгі телефонні розмови.

Крім того, впровадження систем бронювання сприяє зниженню собівартості послуг. Електронний процес бронювання дозволяє оптимізувати формування маршруту туристів за ціною, часом польоту та іншими параметрами. Клієнти можуть зручно порівнювати різні варіанти та вибирати оптимальні за вартістю та відповідністю своїм потребам.

Комп'ютерні системи резервування, такі як CRS (Computer Reservation System), мають широке поширення в галузі електронної комерції. Це пов'язано зі зростанням обсягів послуг та розширенням їх асортименту. Також використання електронних систем дозволяє знизити вартість бронювання і резервування послуг, що робить їх більш доступними для широкого кола клієнтів.

В цілому, впровадження систем бронювання в електронній комерції сприяє покращенню ефективності і зручності процесу бронювання для клієнтів, а також допомагає підприємствам оптимізувати свою роботу і забезпечити більш широкий доступ до своїх послуг.

1.2 Роль та значення веб-сервісів пошуку та бронювання авіабілетів у сучасному туристичному секторі

Швидкість передачі та обміну інформацією, її актуальність, своєчасність, адекватність і повнота стали вирішальними факторами успішної діяльності в багатьох галузях економіки сьогодні. Туристичний бізнес також все більше використовує сучасні комп'ютерні технології, які є необхідним елементом для підвищення конкурентоспроможності будь-якого туристичного підприємства. Індустрія туризму має потенціал використовувати різноманітні комп'ютерні технології, починаючи від спеціалізованих програмних продуктів для управління окремими туристичними фірмами до використання глобальних комп'ютерних мереж.

Веб-сервіси пошуку та бронювання авіабілетів відіграють важливу роль у сучасному туристичному секторі, надаючи широкі можливості туристам і подорожнім.

Веб-сервіси пошуку та бронювання авіабілетів в сучасному туристичному секторі відіграють надзвичайно важливу роль і мають значення як для туристів, так і для авіакомпаній та турагентств. Ці сервіси спрощують та оптимізують процес пошуку, порівняння та бронювання авіабілетів, надаючи безліч переваг.

Для туристів веб-сервіси пошуку та бронювання авіабілетів забезпечують зручність та ефективність у виборі та придбанні квитків. Туристи можуть використовувати фільтри та критерії, такі як дата, час, клас обслуговування та бюджет, щоб знайти оптимальні варіанти перельоту. Вони можуть здійснювати пошук і бронювання квитків з будь-якого місця за допомогою комп'ютера або мобільного пристрою, що дозволяє їм зекономити час і зусилля. Такі сервіси також надають детальну інформацію про рейси, авіакомпанії, правила багажу та інші важливі аспекти, що допомагає туристам зробити обізнаний вибір.

Авіакомпанії також отримують численні переваги від використання веб-сервісів пошуку та бронювання авіабілетів. Вони мають можливість збільшити

обсяг продажів, привернувши більше клієнтів через доступність та зручність онлайн бронювання. Впровадження таких систем дозволяє авіакомпаніям ефективно управляти запитами клієнтів та зменшувати витрати на обслуговування. Крім того, вони отримують доступ до аналітичної інформації про попит на авіабілет, що допомагає їм виробляти стратегічні рішення та адаптувати свою пропозицію до потреб ринку.

Турагентства та посередники також користуються вигодами від веб-сервісів пошуку та бронювання авіабілетів. Вони можуть пропонувати своїм клієнтам широкий вибір авіабілетів, рейсів та пакетів, що підвищує їх конкурентоспроможність. Використання таких систем дозволяє турагентствам ефективно керувати бронюваннями, спілкуватися з клієнтами та надавати професійну підтримку протягом усього процесу подорожі. Крім того, вони можуть зосередитися на стратегічних аспектах своєї діяльності та покращити продуктивність завдяки автоматизації рутинних завдань.

Нижче наведено важливість та роль цих сервісів у сучасному туризмі:

- Веб-сервіси пошуку та бронювання авіабілетів забезпечують зручний та простий спосіб пошуку, порівняння та бронювання авіаквитків. Вони доступні 24/7 з будь-якого місця з Інтернет-підключенням.

- Ці сервіси надають можливість вибору з великої кількості авіакомпаній, маршрутів, рейсів і цін. Туристи можуть знайти оптимальний варіант, враховуючи свої вимоги до ціни, розкладу, пересадок та інших параметрів.

- Завдяки веб-сервісам пошуку та бронювання авіабілетів туристи можуть значно скоротити час, який потрібен для пошуку найкращих варіантів. Всі необхідні дані доступні в одному місці, що дозволяє швидко порівнювати та вибирати.

- Веб-сервіси надають доступ до спеціальних пропозицій, акційних цін та знижок на авіабілет. Це дозволяє зекономити гроші на поїздки та знайти найвигідніші пропозиції.

- Туристи можуть бронювати авіабілету до будь-якого місця у світі, незалежно від свого місця проживання. Веб-сервіси дозволяють швидко знайти найкращі маршрути та з'єднання для будь-якої міжнародної або внутрішньонаціональної подорожі.

- Веб-сервіси пошуку та бронювання авіабілетів дозволяють туристам зручно планувати свої поїздки. Вони можуть переглядати розклади рейсів, вибирати зручний час вильоту та прибуття, а також планувати маршрути з врахуванням бажаної тривалості перебування у кожному місці.

- Веб-сервіси надають туристам велику кількість інформації про авіакомпанії, літаки, послуги та правила перевезення. Туристи можуть ознайомитись з усіма необхідними деталями, щоб зробити інформований вибір.

- Веб-сервіси пошуку та бронювання авіабілетів дозволяють туристам порівнювати різні варіанти рейсів та цін. Це допомагає знайти оптимальний баланс між ціною, комфортом та іншими факторами.

- Багато веб-сервісів мають систему відгуків і рейтингів, де туристи можуть ділитися своїми враженнями про авіакомпанії, рейси та обслуговування. Це допомагає іншим користувачам зробити об'єктивний вибір і оцінити якість послуг.

- Веб-сервіси пошуку та бронювання авіабілетів можуть інтегруватись з іншими сервісами, такими як готелі, оренда автомобілів, страхові послуги тощо. Це дозволяє туристам забезпечити повний комплекс послуг для своєї поїздки в одному місці.

- Веб-сервіси надають підтримку клієнтів у вигляді онлайн-чатів, телефонного зв'язку або електронної пошти. Користувачі можуть отримати відповіді на свої питання, отримати допомогу в разі змін чи відміни бронювання та розв'язати будь-які проблеми, пов'язані з авіаподорожами.

- Онлайн системи пошуку та бронювання привертають більше

клієнтів, забезпечуючи зростання обсягу продажів авіакомпаній.

- Системи автоматизують процес обробки запитів, зменшуючи час і зусилля, необхідні для відповіді на запити клієнтів.

- Онлайн системи надають доступ до даних про бронювання, які можуть бути використані для аналізу попиту, виявлення тенденцій та прийняття стратегічних рішень.

- Автоматизація процесу бронювання дозволяє авіакомпаніям знизити витрати на обслуговування клієнтів та адміністративні витрати.

Тож у сучасному туристичному секторі веб-сервіси пошуку та бронювання авіабілетів сприяють розвитку туризму, сприяючи зростанню числа подорожей та стимулюючи конкуренцію між авіакомпаніями, що сприяє покращенню якості послуг та зниженню цін для подорожуючих.

1.3 Огляд існуючих веб-сервісів пошуку та бронювання авіабілетів

Популярні веб-сервіси для пошуку та бронювання авіабілетів надають користувачам зручний та швидкий спосіб знайти найкращі пропозиції на авіаквитки. Вони допомагають зменшити час та зусилля, які потрібно витратити на пошук оптимального варіанту перельоту.

Ці веб-сервіси пропонують широкий вибір авіакомпаній та маршрутів, що дозволяє користувачам знайти найбільш зручні та економічно вигідні варіанти. Вони забезпечують зручну функціональність для фільтрації результатів пошуку за ціною, часом вильоту, трансферами та іншими параметрами, що допомагає точніше налаштувати пошук.

Крім пошуку квитків, ці веб-сервіси часто надають можливість забронювати також готелі, автомобілі та інші послуги, пов'язані з подорожами. Вони дозволяють користувачам отримати повний пакет послуг для організації подорожі в одному місці.

Існує кілька типів сервісів для пошуку та резервування авіаквитків, кожен

з яких має свої особливості і можливості.

Давайте розглянемо детальніше кожен з цих типів:

- Агрегатори авіаквитків є популярним типом сервісу, який збирає дані з різних авіакомпаній і онлайн-тревел-агентств, щоб надати користувачам комплексну інформацію про доступні рейси та ціни. Вони дозволяють шукати авіабілету за різними критеріями, такими як дата, місце відправлення та прибуття, кількість пересадок і багато іншого. Деякі з найпопулярніших агрегаторів авіаквитків включають Skyscanner, Momondo.

- Онлайн-агентства є сервісами, які допомагають користувачам знайти та забронювати авіабілету, а також надають додаткові послуги, такі як бронювання готелів, оренда автомобілів та туристичні пакети. Ці сервіси зазвичай працюють з кількома авіакомпаніями та постачальниками послуг, що дозволяє користувачам отримати більший вибір та зручність. Деякі популярні онлайн-агентства включають Expedia, CheapOair і Orbitz.

- Багато авіакомпаній мають свої власні веб-сайти, на яких користувачі можуть знайти і придбати квитки безпосередньо від авіакомпанії. Ці веб-сайти надають інформацію про розклад рейсів, доступність місць, ціни та спеціальні пропозиції. Резервація квитків через офіційний веб-сайт авіакомпанії може мати певні переваги, такі як пряма підтримка від компанії та можливість зареєструватися для програми лояльності.

- Метапошуковики є спеціальним типом сервісу, який сканує інформацію з інших веб-сайтів пошуку авіабілетів і надає користувачам комплексні результати. Вони використовують алгоритми для зіставлення та порівняння рейсів та цін з різних джерел. Метапошуковики дозволяють швидко знайти найкращі пропозиції з різних сервісів пошуку авіабілетів без необхідності відвідувати кожен сайт окремо. Деякі відомі метапошуковики авіаквитків включають Google Flights, Kayak.

Ці різні типи сервісів для пошуку та резервування авіаквитків мають свої переваги та особливості, і вони допомагають максимально спростити процес

пошуку та бронювання квитків, забезпечуючи швидкий доступ до інформації, порівняння цін та вибір оптимального варіанту для кожного користувача.

Популярні веб-сервіси для пошуку та бронювання авіабілетів мають інтуїтивний та зручний інтерфейс, який дозволяє легко орієнтуватися по сайту та здійснювати бронювання без зайвих труднощів. Вони також можуть надавати додаткові корисні функції, такі як сповіщення про знижки, зміни цін або можливості вигідних пропозицій.

Всі ці функції та особливості популярних веб-сервісів для пошуку та бронювання авіабілетів спрямовані на полегшення та зручність користувача під час пошуку, порівняння та бронювання квитків, допомагаючи забезпечити незабутню та безпроблемну подорож.

Найпопулярніші сервіси з пошук та бронювання авіаквитків:

1. Expedia (www.expedia.com):

Expedia Flights є окремою частиною веб-сервісу Expedia, присвяченою пошуку та бронюванню авіаквитків (рисунок 1.2). Він має зручний інтерфейс та широкі можливості фільтрації. Expedia Flights також пропонує пакетні пропозиції, які включають квитки на літак та готель.

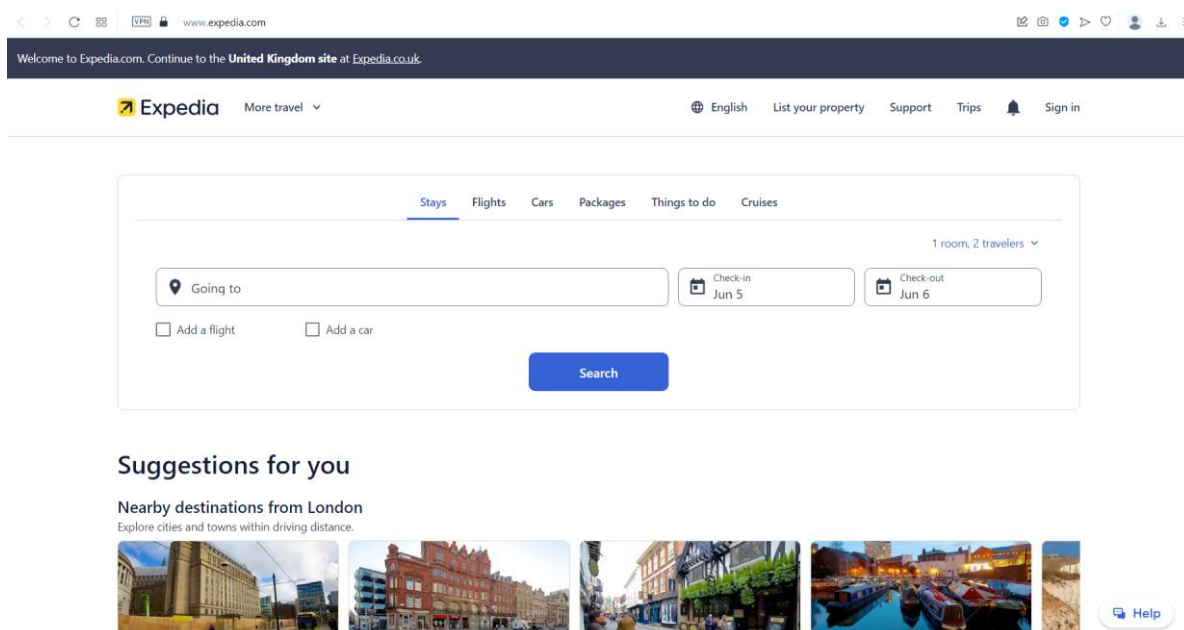


Рисунок 1.2 — Головна сторінка сервісу Expedia

2. Kayak (www.kayak.com):

Кауак є популярним сервісом, який дозволяє швидко порівнювати ціни на авіаквитки з різних джерел. Він пропонує зручні фільтри та інструменти для пошуку найкращих пропозицій (рисунок 1.3). Крім того, Кауак надає інформацію про багажні правила, рейтинги авіакомпаній та відгуки пасажирів.

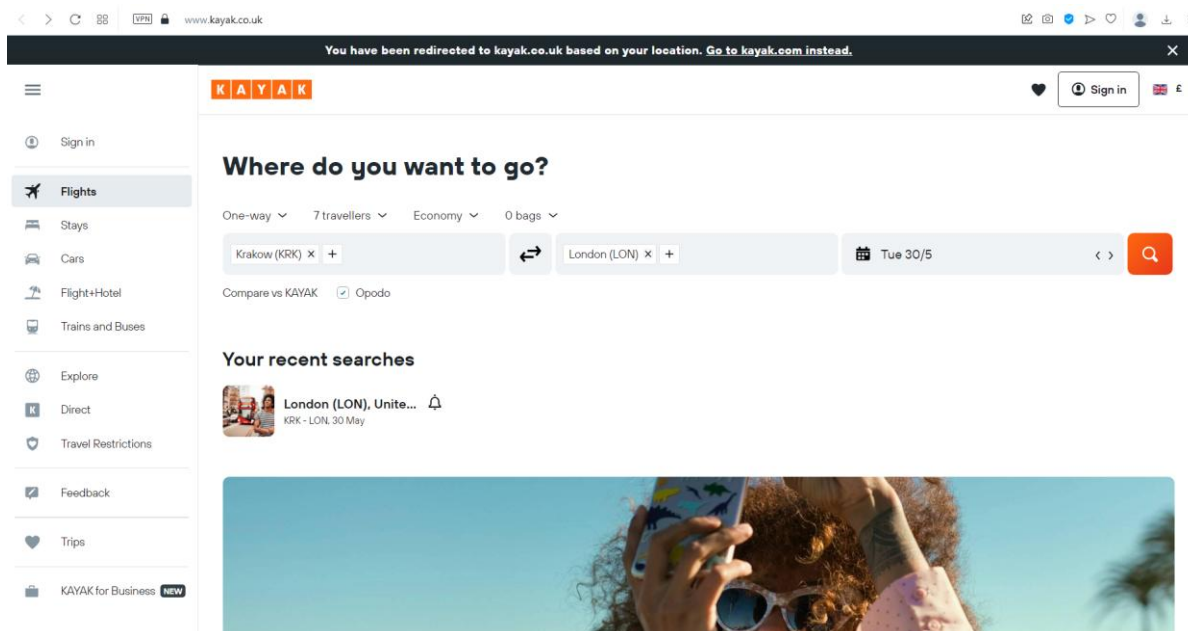


Рисунок 1.3 — Головна сторінка сервісу Kayak

3. Skyscanner (www.skyscanner.com):

Skyscanner є глобальним веб-сервісом для пошуку авіаквитків. Він допомагає знайти найкращі ціни та розклади рейсів. Skyscanner пропонує можливість порівняння цін з різних авіакомпаній та агентств (рисунок 1.4). Крім того, він має корисні функції, такі як оповіщення про зміни цін та календар цін на певний маршрут.

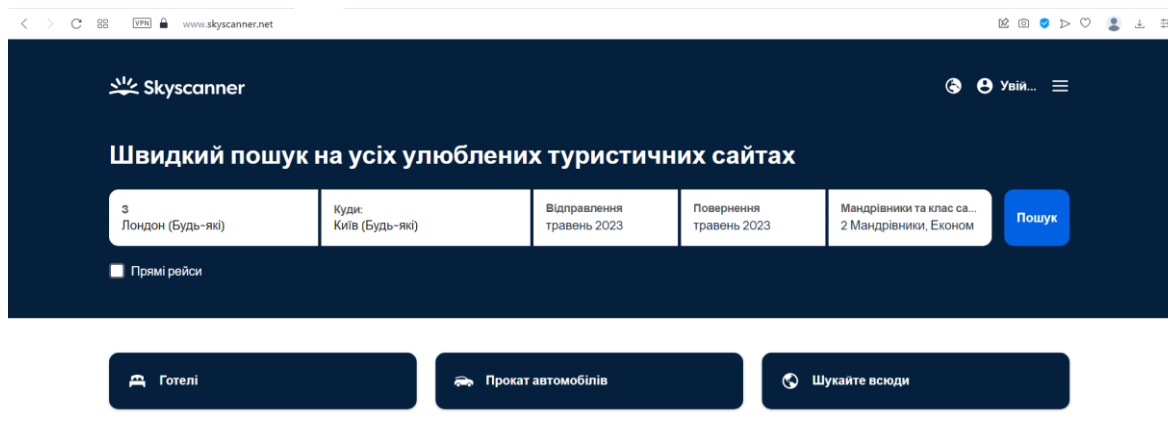


Рисунок 1.4 — Головна сторінка сервісу Skyscanner

4. Google Flights (www.google.com/flights):

Google Flights є потужним інструментом для пошуку авіабілетів. Він пропонує розширені функції фільтрації, інтерактивну карту маршрутів та можливість отримати сповіщення про зміни цін (рисунок 1.5). Google Flights також надає корисні поради щодо вибору оптимального часу та днів для польотів.

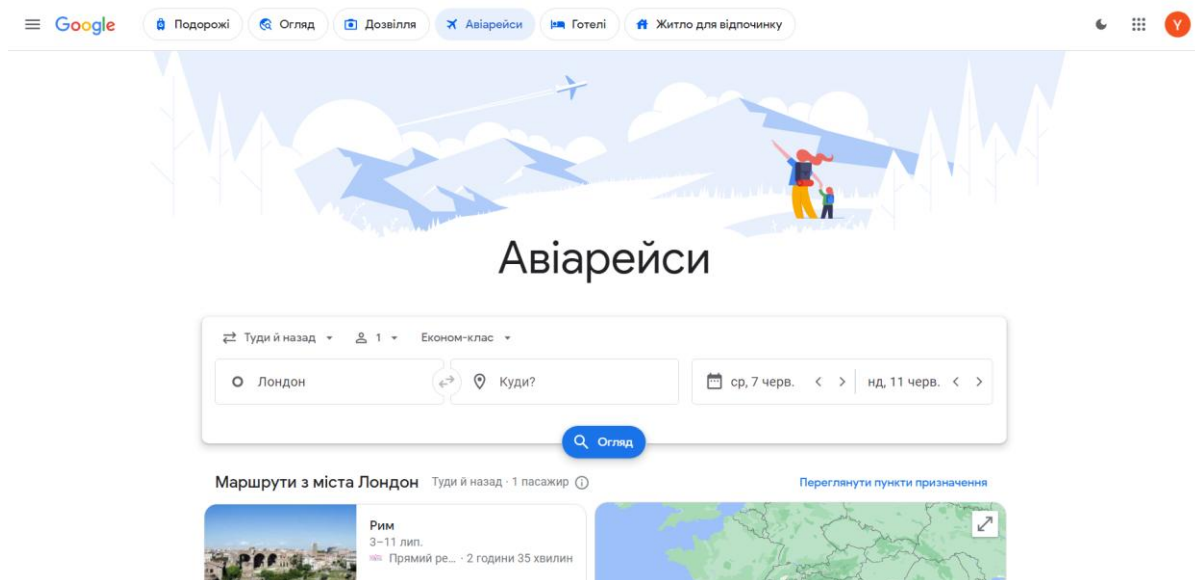


Рисунок 1.5 — Головна сторінка сервісу Google Flights

5. Momondo (www.momondo.com):

Momondo є веб-сервісом, спрямованим на пошук найкращих пропозицій на авіаквитки. Він використовує різні джерела даних та алгоритми для знаходження оптимальних варіантів (рисунок 1.6). Momondo також включає функцію "Гнучкі дати", яка дозволяє знайти найбільш вигідні ціни на різні дні.

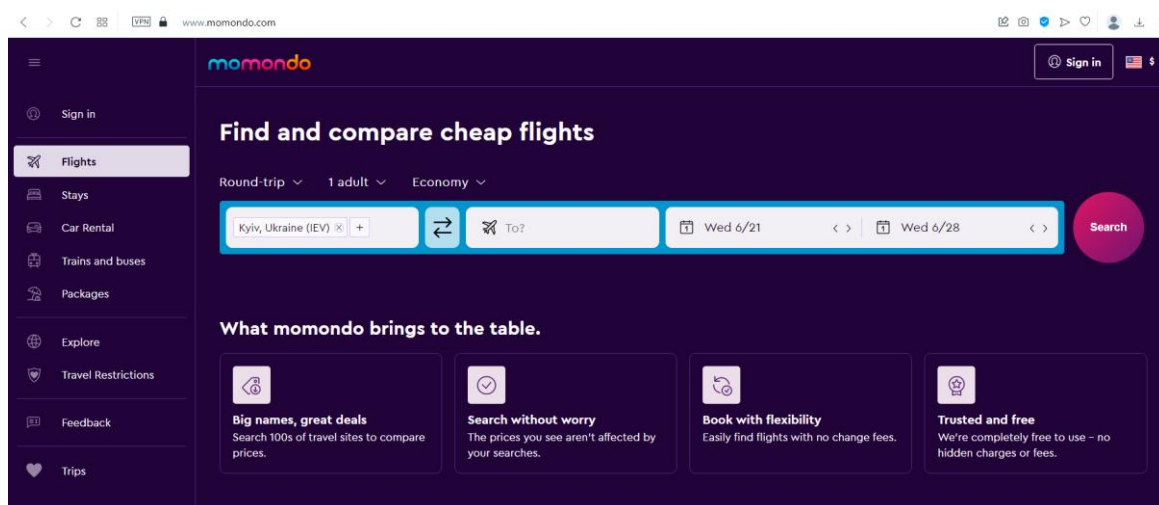


Рисунок 1.6 — Головна сторінка сервісу Momondo

6. CheapOair (www.cheapoair.com):

CheapOair є веб-сервісом, спеціалізованим на пошуку дешевих авіаквитків. Він пропонує широкий вибір пропозицій та спеціальних акцій для різних маршрутів (рисунок 1.7). CheapOair також надає можливість забронювати готелі та оренду автомобіля.

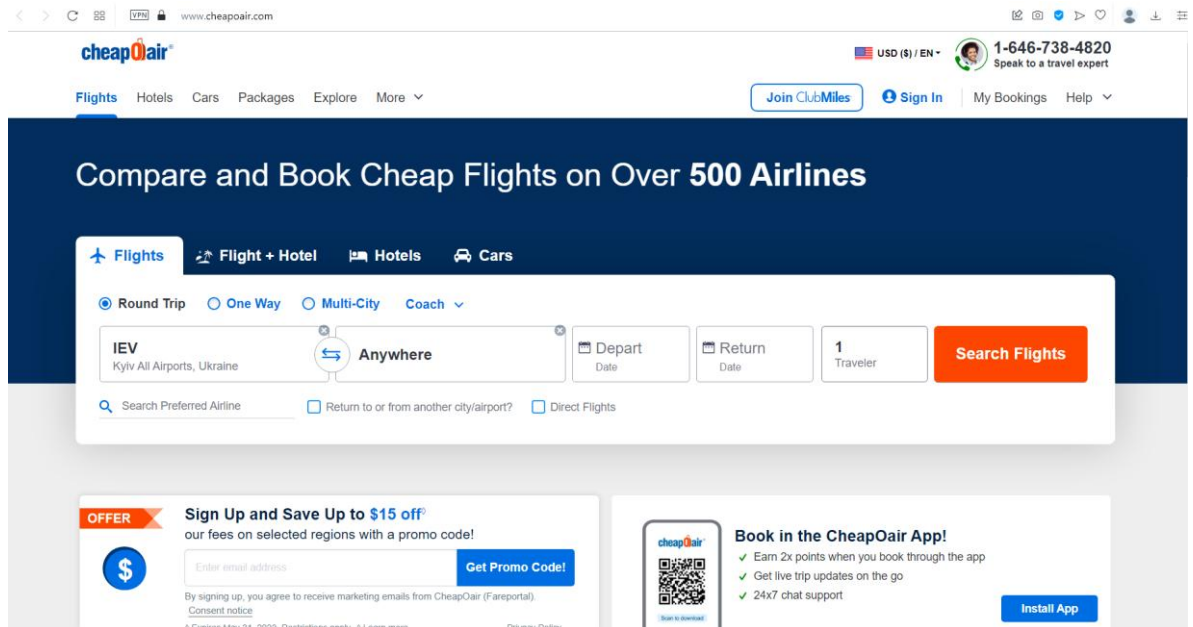


Рисунок 1.7 — Головна сторінка сервісу CheapOair

7. Orbitz (www.orbitz.com):

Orbitz є веб-сервісом, який допомагає знайти квитки на літак, готелі та оренду автомобіля. Він має широкий вибір пропозицій та можливості фільтрації за різними критеріями (рисунок 1.8). Orbitz також пропонує програму лояльності, яка дозволяє збирати бонусні бали та отримувати знижки.

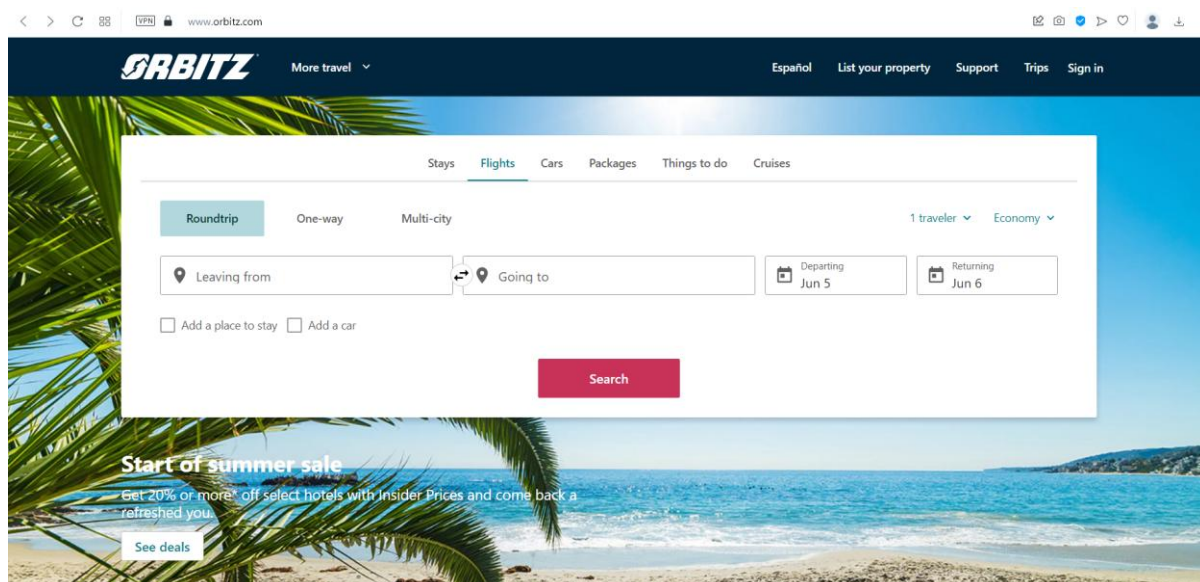


Рисунок 1.8 — Головна сторінка сервісу Orbitz

8. Priceline (www.priceline.com):

Priceline є популярним веб-сервісом для пошуку та бронювання авіабілетів. Він пропонує гнучкі опції пошуку, включаючи можливість знайти "Назвіть свою ціну" та отримати унікальні пропозиції (рисунок 1.9). Priceline також надає інформацію про готелі та оренду автомобіля.

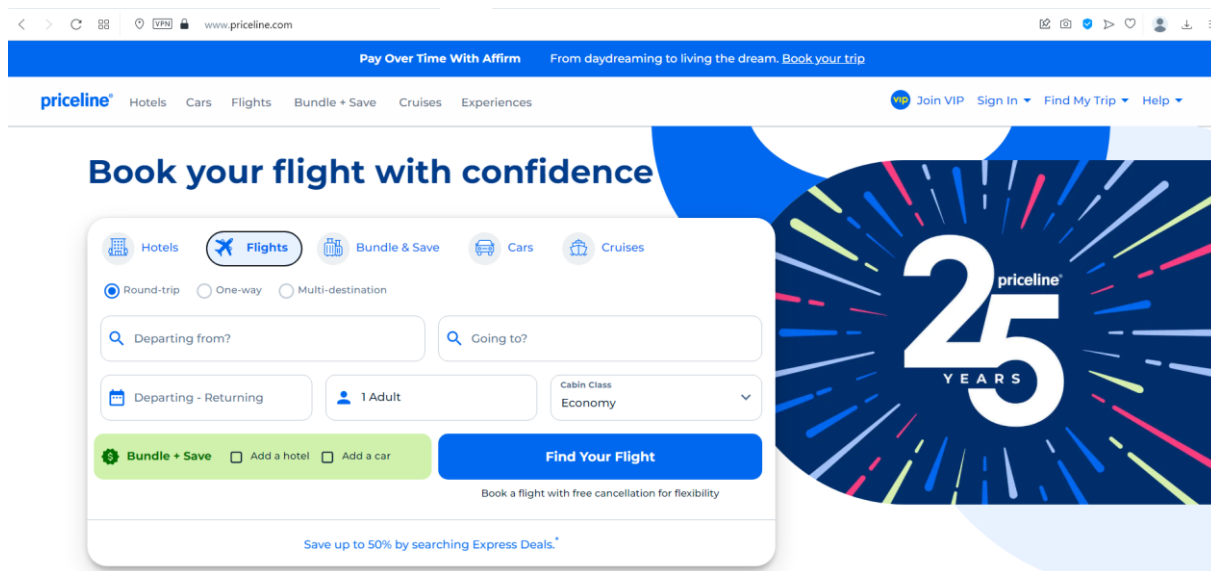


Рисунок 1.9 — Головна сторінка сервісу Priceline

9. Travelocity (www.travelocity.com):

Travelocity є веб-сервісом, спрямованим на пошук та бронювання авіаквитків, готелів та оренди автомобіля (рисунок 1.10). Він пропонує широкий вибір варіантів та можливість забронювати пакетні пропозиції.

Travelocity також надає додаткові послуги, такі як екскурсії та круїзи.

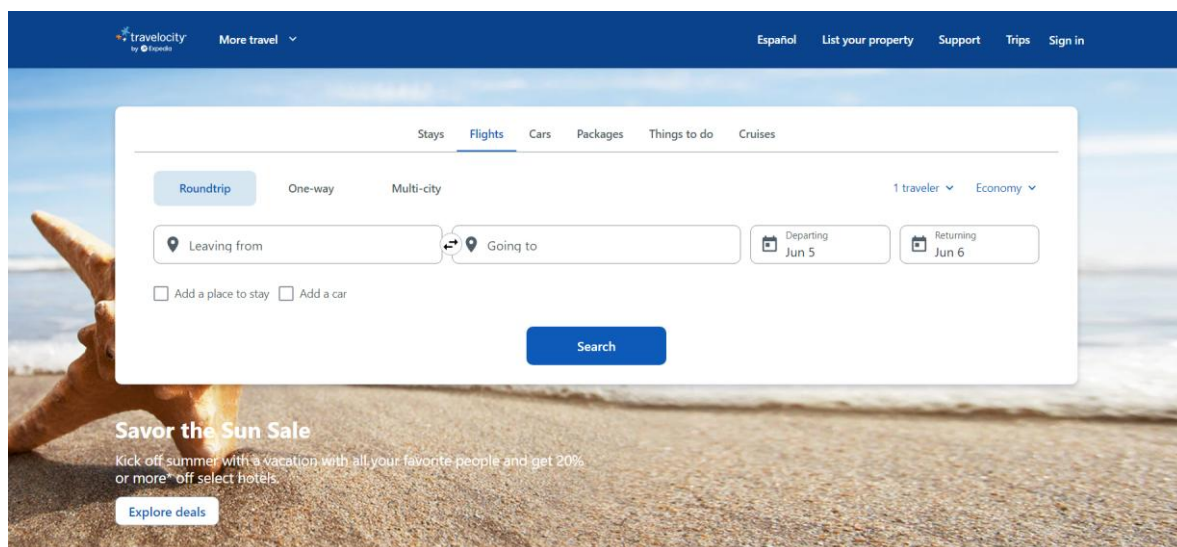


Рисунок 1.10 — Головна сторінка сервісу Travelocity

1.4 Завдання та функціональні можливості веб-сервісів пошуку та бронювання авіабілетів

Завдання веб-сервісів пошуку та бронювання авіабілетів полягає у наданні користувачам зручного та ефективного інструменту для пошуку, порівняння та бронювання авіабілетів.

Веб-сервіси пошуку та бронювання авіабілетів надають користувачам широкий спектр функціональних можливостей. Основні з них включають:

1. Пошук доступних авіабілетів на різні напрямки та дати відправлення.
2. Фільтрація результатів пошуку за різними параметрами, такими як ціна, авіакомпанія, час вильоту та прильоту, кількість пересадок тощо.
3. Порівняння цін та умов перельоту між різними авіакомпаніями.
4. Відображення детальної інформації про кожен рейс, включаючи маршрут, час вильоту та прильоту, тривалість перельоту, тип літака тощо.
5. Вибір бажаного місця у салоні літака.
6. Можливість додаткового бронювання послуг, таких як додатковий багаж, страхування, пересадки, готелі тощо.
7. Відображення графіка рейсів та їх доступності на різних датах.

8. Можливість вибору гнучких дат відправлення для пошуку найкращих пропозицій.
9. Відображення рейтингів та відгуків про авіакомпанії та конкретні рейси.
10. Калькулятор вартості авіабілетів з урахуванням додаткових зборів та податків.
11. Можливість збереження результатів пошуку та налаштування повідомлень про знижки та акції.
12. Розрахунок тривалості та оптимальних маршрутів для подорожей з пересадками.
13. Можливість збереження особистої інформації та персональних налаштувань для швидшого бронювання.
14. Підтримка різних мов та валют для зручності користувачів з різних країн.
15. Опції пошуку за гнучкими критеріями, такими як бажаний час перельоту, проміжна зупинка тощо.
16. Відображення актуальної інформації про правила та обмеження авіакомпаній, такі як багаж, пересадки, вік пасажирів тощо.
17. Опція підписки на новини та спеціальні пропозиції авіакомпаній та сервісу.
18. Відстеження статусу рейсу та отримання сповіщень про зміни в графіку.
19. Можливість вибору різних класів обслуговування та сидінь у літаку.
20. Інтеграція з платіжними системами для онлайн-оплати авіабілетів.
21. Можливість збереження профілю користувача з історією бронювань та збором бонусних балів.
22. Відображення додаткової інформації про аеропорти, таку як послуги, трансфери, розташування тощо.
23. Можливість обрати спеціальні пропозиції, такі як гнучкі тарифи або

пакетні пропозиції з авіабілетами та готелями.

24. Опції сортування результатів пошуку за різними параметрами, такими як ціна, тривалість, рейтинг тощо.

25. Можливість перегляду історії пошуку та бронювань для відстеження здійснених операцій.

26. Відображення інформації про вимоги до документів та візовий режим для міжнародних перельотів.

27. Можливість звернутися до служби підтримки клієнтів для вирішення питань та отримання допомоги.

28. Розрахунок прогнозованої вартості перельоту з урахуванням змін у цінах та тарифах.

29. Можливість вибору різних способів доставки авіабілетів, таких як електронний квиток, доставка кур'єром або видача у авіакасі.

30. Відображення інформації про правила та обмеження щодо повернення та зміни бронювань.

31. Можливість збереження попередніх пошукових запитів та налаштувань для зручного повторного використання.

32. Інтеграція з картами та навігаційними сервісами для відображення маршрутів та вказівок до аеропорту.

33. Можливість перегляду акційних пропозицій та знижок на авіабілети.

34. Відображення інформації про правила безпеки та процедури належного перевірки на аеропорту.

35. Можливість обміну або повернення авіабілетів за певних умов.

36. Відображення інформації про правила ввезення та вивезення товарів на різних міжнародних маршрутах.

37. Можливість збереження та редагування профілю пасажирів для зручного заповнення бронювання.

38. Інтеграція з соціальними мережами для швидкої авторизації та спільного пошуку з друзями.

39. Можливість отримання спеціальних привілеїв та знижок для постійних користувачів сервісу.

40. Можливість перегляду графіка роботи та контактних даних авіакомпаній.

41. Збереження історії платежів та звітів для ведення фінансового обліку.

42. Можливість обміну повідомленнями з авіакомпаніями та представниками сервісу щодо бронювань та змін.

43. Відображення інформації про перевізників та їх флот літаків для вибору певного типу апарату.

44. Можливість вибору різних опцій харчування та послуги на борту літака.

45. Інтеграція з календарем та планувальником для автоматичного внесення даних про польоти та подорожі.

46. Можливість збереження улюблених маршрутів та налаштувань для швидкого доступу та пошуку.

Отже, веб-сервіси пошуку та бронювання авіабілетів спрощують процес пошуку та придбання квитків, забезпечують широкий вибір, порівняння цін та зручності для користувачів. Вони стали невід'ємною частиною туристичної індустрії та дозволяють людям швидко здійснювати подорожі без зайвих зусиль та витрат часу.

1.5 Взаємодія з базами даних авіакомпаній та інших постачальників даних

Веб-сервіси пошуку та бронювання авіабілетів взаємодіють з базами даних авіакомпаній та інших постачальників даних для забезпечення актуальної та повної інформації про наявність рейсів, ціни, розклади та інші важливі деталі. Ця взаємодія відбувається через різні механізми і протоколи, зокрема:

- API (Application Programming Interface). Багато авіакомпаній та

постачальників даних надають API, що дозволяє веб-сервісам звертатися до їх баз даних для отримання необхідної інформації. Через API можна отримати доступ до даних про рейси, наявність місць, ціни, додаткові послуги тощо.

- XML або JSON. Дані, що надсилаються з баз даних авіакомпаній та інших постачальників, часто представляються у форматі XML або JSON. Ці формати дозволяють структурувати дані і передавати їх між системами для подальшої обробки.

- Передача даних в реальному часі. Деякі веб-сервіси використовують протоколи, що дозволяють отримувати дані в реальному часі. Це дозволяє користувачам отримувати найсвіжішу інформацію про наявність рейсів та їхні зміни.

- Синхронізація баз даних. Деякі веб-сервіси використовують механізми синхронізації баз даних для забезпечення консистентності та актуальності інформації. Це дозволяє підтримувати узгодженість даних між веб-сервісом та базами даних авіакомпаній та постачальників.

Взаємодія з базами даних авіакомпаній та інших постачальників даних є критично важливою для забезпечення коректної та актуальної інформації про авіабілеті. Це дозволяє веб-сервісам забезпечувати користувачів найкращими варіантами перельотів, цінами та іншими деталями, що сприяє зручності та ефективності процесу пошуку та бронювання авіабілетів.

РОЗДІЛ 2 ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ ДЛЯ РОЗРОБКИ ВЕБ СЕРВІСІВ ПОШУКУ ТА БРОНЮВАННЯ АВІАБІЛЕТІВ

Розробка веб-сервісів пошуку та бронювання авіабілетів вимагає використання різноманітних технологій та інструментів. Вони включають веб-фреймворки, мови програмування, бази даних та інші компоненти для створення потужних та ефективних додатків. Веб-фреймворки, такі як Django, Ruby on Rails або Laravel, надають зручність розробки та прискорюють процес створення веб-додатків. Мови програмування, такі як JavaScript, Python або PHP, використовуються для реалізації функціональності та взаємодії з користувачами. Системи керування базами даних, наприклад MySQL, PostgreSQL або MongoDB, забезпечують зберігання та ефективний доступ до даних про авіарейси та ціни. Крім того, для забезпечення безпеки та захисту особистих даних користувачів використовуються шифрування даних за допомогою протоколу HTTPS та інструменти для перевірки платежів та запобігання шахрайству. Використання цих технологій та інструментів дозволяє розробникам створювати потужні та надійні веб-сервіси для пошуку та бронювання авіабілетів, що забезпечують зручну та задоволену користувачів функціональність.

2.1 Інформаційні технології побудови веб-сервісів пошуку та бронювання авіабілетів

Інформаційні технології, що використовуються в побудові веб-сервісів пошуку та бронювання авіабілетів, включають різноманітні технології, протоколи та інструменти. Нижче розглянуто деякі з них:

- API (Application Programming Interface). API використовується для взаємодії між різними системами. У випадку веб-сервісів пошуку та бронювання авіабілетів, API використовується для отримання даних від авіакомпаній та інших постачальників послуг. API надає доступ до різних

функцій та можливостей, таких як пошук рейсів, отримання цін та наявності квитків, бронювання тощо.

- Веб-скрапінг. Це техніка, за допомогою якої веб-сервіс може отримувати дані з веб-сторінок, необхідні для пошуку та бронювання авіабілетів. Веб-скрапінг дозволяє автоматично зчитувати та аналізувати структуровані дані з веб-сторінок, такі як ціни, розклади рейсів, наявність місць тощо. Це дає можливість веб-сервісу забезпечити актуальну інформацію користувачам.

- Бази даних. Для зберігання та управління великим обсягом даних, пов'язаних з авіабілетами, використовуються бази даних. Це можуть бути реляційні бази даних, які забезпечують структуроване зберігання даних, або нереляційні (NoSQL) бази даних, які дозволяють швидкий та гнучкий доступ до даних.

- Серверні технології. Для розробки веб-сервісу використовуються серверні технології, такі як Python (зокрема фреймворк FastAPI), Java, PHP тощо. Ці технології дозволяють створювати логіку роботи сервісу, обробляти запити користувачів, взаємодіяти з базою даних та іншими компонентами системи.

- Хмарні технології. Застосування хмарних технологій, таких як Amazon Web Services (AWS), Microsoft Azure, Google Cloud тощо, дозволяє забезпечити масштабованість та доступність веб-сервісу. Хмарні платформи надають можливість гнучко масштабувати обчислювальні та зберігання даних, забезпечуючи високу продуктивність та надійність сервісу.

- Інтерфейс користувача. Для зручного взаємодії з веб-сервісом розробляється інтерфейс користувача (UI). Це може бути веб-сторінка, мобільний застосунок або API для взаємодії з іншими системами. Інтерфейс користувача повинен бути зручним, інтуїтивно зрозумілим та естетично привабливим.

Ці технології допомагають побудувати ефективний, функціональний та безпечний веб-сервіс пошуку та бронювання авіабілетів, що задовольняє потреби користувачів і сприяє розвитку туристичного сектора.

2.2 Розробка архітектурного плану веб-сервісу

Розробка архітектурного плану веб-сервісу є надзвичайно важливим етапом у процесі розробки будь-якого веб-застосунку. Архітектурний план допомагає визначити структуру та організацію веб-сервісу. Він визначає, які компоненти будуть включені, як вони будуть взаємодіяти між собою та як буде організовано потік даних. Це дозволяє забезпечити ефективне та логічне функціонування системи.

Розробка архітектурного плану дозволяє забезпечити розширюваність веб-сервісу. Це допомагає визначити, як нові функції та можливості можуть бути додані до системи без необхідності значних переробок або змін усього коду. Це дозволяє забезпечити більш гнучку та масштабовану архітектуру.

Наявність архітектурного плану дозволяє розробити ефективну та продуктивну систему. Він визначає оптимальний спосіб використання ресурсів, оптимізує роботу з базами даних, забезпечує швидкий доступ до інформації та розробляє стратегію масштабування системи залежно від зростання обсягу даних та навантаження.

Архітектурний план визначає, як веб-сервіс може бути інтегрований з іншими системами та сервісами. Він враховує сумісність з різними протоколами та стандартами, що дозволяє легко обмінюватись даними з іншими системами та розширювати функціональні можливості шляхом інтеграції з іншими сервісами.

В цілому, розробка архітектурного плану веб-сервісу дозволяє забезпечити ефективну, гнучку, масштабовану та безпечну систему, яка задовольняє потреби користувачів і забезпечує конкурентну перевагу в туристичній галузі.

Архітектурний план веб-сервісу пошуку та бронювання авіабілетів:

- FastAPI сервер. Розробка та реалізація серверної частини з використанням FastAPI, який надає API для взаємодії з клієнтськими застосунками або інтерфейсом користувача. Цей сервер отримує запити від користувачів та обробляє їх. Користувачі формують посилання на сайт Kayak, передаючи необхідні параметри пошуку. Цей модуль отримує запити від FastAPI сервера та передає їх до Kayak.

- Користувацький інтерфейс. Розробка веб-сторінки, яка надає користувачам можливість вводити параметри пошуку, такі як місце відправлення, місце призначення, дату вильоту тощо. Цей інтерфейс взаємодіє з FastAPI сервером для передачі параметрів пошуку.

- Веб-скрапінг. Реалізація модуля для веб-скрапінгу, який отримує параметри запиту від користувача, формує посилання на сайт Kayak, виконує запити та отримує HTML сторінки з інформацією про наявні квитки. За допомогою парсингу HTML сторінок, цей модуль витягує необхідну інформацію про квитки, таку як ціни, рейси, часи вильоту тощо.

- База даних. Збереження отриманої інформації про квитки у базі даних для подальшого використання та відображення. База даних може містити таблиці зі збереженими даними про рейси, ціни, місця відправлення та призначення тощо.

- Представлення результатів. Формування та відображення результатів пошуку користувачам через інтерфейс або застосунок. Інформація про наявні квитки, отримана з веб-скрапінгу та збережена у базі даних, може бути відображена у зручному форматі для користувачів.

Всі компоненти взаємодіють між собою, дозволяючи користувачам формувати посилання на сайт kayak за допомогою FastAPI сервера, отримувати актуальну інформацію про квитки шляхом веб-скрапінгу та відображати результати пошуку через користувацький інтерфейс або застосунок.

2.3 Використання REST архітектури при створенні веб-сервісу пошуку авіабілетів

При створенні веб-сервісу пошуку авіабілетів можна використовувати REST (Representational State Transfer) архітектурний стиль, який забезпечує простоту, масштабованість та незалежність між клієнтом та сервером.

REST (Representational State Transfer) є архітектурним стилем, що використовується для побудови розподілених систем, основаних на веб-протоколах. Вперше представлений в дисертації Роя Філдінга в 2000 році, REST став широко використовуваним підходом до розробки веб-сервісів.

Основними принципами REST є:

- Клієнт-серверна архітектура. Система розділяється на дві основні складові - клієнтів, які виконують запити, та сервери, які надають ресурси або обробляють запити. Це дозволяє забезпечити відокремленість між користувацьким інтерфейсом та серверними функціями.

- Безстанність. Кожний запит клієнта до сервера містить усю необхідну інформацію для обробки запиту. Сервер не зберігає стан клієнта між запитами, що дозволяє покращити масштабованість і надійність системи.

- Кешування. Сервер може позначати відповіді, які можуть бути кешовані клієнтами для подальшого використання. Це зменшує навантаження на сервер і покращує швидкодію системи.

- Єдинообразність інтерфейсу. Система використовує стандартні HTTP-методи (GET, POST, PUT, DELETE) для взаємодії з ресурсами. Це спрощує розуміння та використання системи і забезпечує її доступність для різних клієнтів.

- Шарування. Система може бути розподілена на шари, де кожен шар виконує певну функціональність. Це дозволяє досягти більшої гнучкості та масштабованості системи.

REST є популярним підходом до розробки веб-сервісів через свою

простоту, масштабованість і здатність до інтеграції з різними платформами та пристроями. Він покладається на використання протоколу HTTP і надає можливість обміну даними між клієнтом і сервером за допомогою стандартних веб-запитів і відповідей.

У REST архітектурі, кожен елемент або функціональність системи розглядається як ресурс, представлений унікальним ідентифікатором URI. Наприклад, авіабілет може бути представлений як ресурс з URI '/flights'. Клієнт може використовувати HTTP методи, такі як GET, POST, PUT та DELETE, для взаємодії з цим ресурсом.

Даний архітектурний стиль використовує різні HTTP методи для виконання різних операцій над ресурсами. Наприклад, GET метод використовується для отримання даних про авіабілет, POST метод - для створення, PUT метод - для оновлення, DELETE метод - для видалення.

У REST використовуються HTTP статуси відповіді для передачі інформації про стан операції. Наприклад, статус 200 OK вказує на успішну операцію, статус 404 Not Found - на відсутність ресурсу, статус 201 Created - на успішне створення ресурсу.

REST може використовувати різні формати даних для обміну інформацією між клієнтом та сервером, такі як JSON або XML. JSON є популярним форматом для передачі структурованих даних у REST API.

Також REST може використовувати гіперпосилання для навігації між різними ресурсами. Наприклад, у відповіді на запит про авіабілет, можуть бути включені гіперпосилання на подальшу інформацію про кожен авіабілет.

2.4 Загальна інформація про засоби розробки веб-сервісу

Для розробки системи пошуку авіаквитків можна використовувати різні платформи та засоби розробки в залежності від вимог.

В даній кваліфікаційній роботі для розробки веб-сервісу було використано наступні платформи та засобів:

- Платформа FastAPI. Вона є швидким (за рахунок використання асинхронності) та простим у використанні веб-фреймворком для розробки серверної частини системи. Він підтримує створення API за допомогою Python і може бути використаний для прийому запитів користувачів та обробки їх даних. FastAPI надає можливості для створення швидкого та масштабованого веб-сервісу, забезпечуючи автоматичну документацію, валідацію даних та підтримку асинхронних запитів.

- Selenium. Він є популярним інструментом для автоматизації веб-браузера. Ви можете використовувати Selenium для взаємодії з веб-скрапінгом та отримання актуальної інформації про наявні квитки з веб-сайтів, таких як Kayak. Selenium дозволяє емулювати дії користувача, такі як натискання кнопок, заповнення форм, навігація по сторінкам та отримання результатів.

- Angular. Angular є потужним фреймворком для розробки клієнтської частини веб-застосунків. Ви можете використовувати Angular для створення користувацького інтерфейсу системи пошуку авіаквитків, який буде забезпечувати зручну навігацію та взаємодію з користувачем.

Ці три засоби - FastAPI, Selenium та Angular - можна поєднувати разом для розробки повноцінного веб-сервісу пошуку авіаквитків. FastAPI використовується для створення серверної частини, яка обробляє запити користувачів та взаємодіє зі скрапером, побудованим з використанням Selenium. Angular використовується для розробки клієнтської частини, яка відображає результати пошуку та забезпечує зручний інтерфейс для користувача.

2.5 Веб-скрапінг

Веб-скрапінг – це техніка, за допомогою якої веб-сервіс може отримувати дані з веб-сторінок, необхідні для пошуку та бронювання авіабілетів. Веб-

скрапінг дозволяє автоматично зчитувати та аналізувати структуровані дані з веб-сторінок, такі як ціни, розклади рейсів, наявність місць тощо (рисунок 2.1).

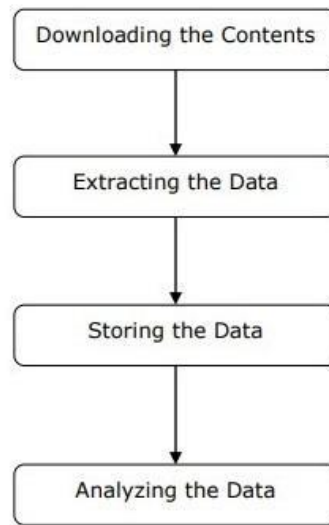


Рисунок 2.1 — Робота веб-скрапінгу

Selenium - це інструмент для автоматизації веб-браузерів, який широко використовується для створення веб-сервісів з веб-скрапінгом. Основна перевага використання Selenium полягає у можливості інтерактивно взаємодіяти з веб-сторінками, що містять динамічний контент, який генерується за допомогою JavaScript (рисунок 2.2).

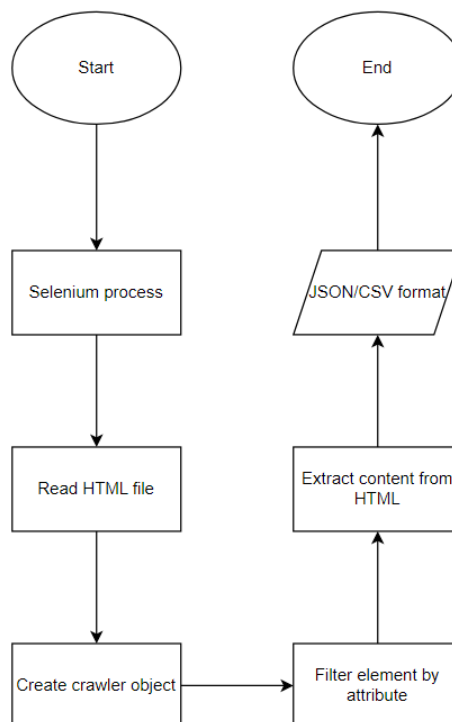


Рисунок 2.2 — Блок-схема збирання даних за допомогою Selenium через API

Основні компоненти Selenium включають наступне:

- **WebDriver.** Він є основним компонентом Selenium і використовується для керування браузером. Він надає різні методи для навігації по сторінках, виконання дій (натискання кнопок, заповнення форм тощо) та отримання даних з веб-сторінок. WebDriver підтримує різні браузери, такі як Chrome, Firefox, Safari та інші.
- **Locator Strategies.** Selenium надає різні стратегії для знаходження елементів на веб-сторінці. Це може бути за допомогою CSS-селекторів, XPath або інших методів ідентифікації елементів. За допомогою цих стратегій можна локалізувати потрібні елементи для взаємодії з ними.
- **Actions.** Selenium дозволяє виконувати різні дії на веб-сторінці, такі як натискання кнопок, заповнення форм, перетягування елементів тощо. Це забезпечує можливість взаємодіяти з веб-сторінкою так само, як це робить користувач.
- **Waits.** Час очікування є важливою частиною веб-скрапінгу, оскільки сторінки можуть завантажуватись динамічно або вимагати деякого часу для відповіді. Selenium надає можливість встановлювати різні часи очікування, щоб забезпечити правильну взаємодію зі сторінкою.
- **Capturing Screenshots.** Selenium також може захоплювати знімки екрана веб-сторінок. Це корисний інструмент для відладки та візуалізації даних, які отримуються зі сторінок.

Selenium підтримує мови програмування, такі як Python, Java, C#, Ruby та інші, що дозволяє розробникам використовувати їх у своїх проектах. Використовуючи Selenium разом з іншими інструментами, такими як FastAPI та Angular, ви можете створити потужні веб-сервіси для резервування авіаквитків з можливістю взаємодії з веб-сторінками.

2.6 Протоколи передачі даних, та їх застосування у веб-сервісах для пошуку та бронювання авіабілетів

Протоколи передачі даних, такі як JSON (JavaScript Object Notation) та XML (Extensible Markup Language), є двома широко використовуваними форматами для обміну даними у веб-сервісах.

JSON є легким та простим у використанні форматом, що базується на синтаксисі JavaScript. Він дозволяє представляти структуровані дані у вигляді об'єктів, масивів та примітивних типів даних. JSON є популярним у веб-розробці, оскільки його формат легко читається як людиною, а також його можна легко обробляти та створювати в багатьох мовах програмування.

XML є розширюваним мовним форматом, який використовується для представлення структурованих даних. Він використовує теги для визначення різних елементів та атрибутів, що дозволяє гнучкіше представлення даних. XML широко використовується у веб-сервісах, оскільки його можна легко обробляти за допомогою XML-схем, а також він підтримує багатомовний підхід із заданням власних елементів та правил структури даних.

У веб-сервісах для пошуку та бронювання авіаквитків ці протоколи використовуються для передачі даних між клієнтом (наприклад, веб-браузером або мобільним застосунком) та сервером, який забезпечує функціональність пошуку та бронювання квитків.

Наприклад, при взаємодії з веб-сервісом пошуку та бронювання авіаквитків, клієнт може надсилати запити на сервер у форматі JSON або XML, що містять параметри пошуку, наприклад, місце відправлення, місце призначення, дату вильоту тощо. Сервер обробляє ці запити та повертає відповіді у вигляді JSON або XML з результатами пошуку, включаючи інформацію про доступні рейси, ціни, розклади тощо.

Клієнт може далі обробляти ці дані, відображати їх у зручному для користувача форматі та надсилати запити на бронювання квитків з використанням обраного протоколу. Сервер з свого боку обробляє запити на бронювання та повертає відповіді, підтверджуючи або відхиляючи бронювання.

Використання JSON або XML у веб-сервісах для пошуку та бронювання авіаквитків дозволяє ефективно передавати та обробляти структуровані дані, що спрощує взаємодію між клієнтом та сервером та забезпечує зручну обробку результатів пошуку та бронювання.

РОЗДІЛ 3 ПРОЕКТУВАННЯ, РОЗРОБЛЕННЯ, РЕАЛІЗАЦІЯ ВЕБ-СЕРВІСУ ПОШУКУ ТА БРОНЮВАННЯ АВІАБІЛЕТІВ

Проектування, розроблення та реалізація веб-сервісу пошуку та бронювання авіабілетів - це комплексний процес, що включає в себе кроки від концепції до функціонального продукту. Перш за все, проводиться аналіз потреб та вимог користувачів, що дозволяє визначити ключові функціональність, архітектуру та інтерфейс веб-сервісу. Наступним етапом є розроблення детального плану проекту, включаючи визначення технологій, бази даних, системи аутентифікації та інших важливих аспектів. Після цього розробники приступають до реалізації веб-сервісу, створюючи необхідні функціональні модулі, інтегруючи зовнішні сервіси та розробляючи логіку пошуку та бронювання авіабілетів. Під час розробки важливим етапом є тестування та відлагодження системи, що дозволяє виявити та виправити помилки. Нарешті, після успішного завершення розробки веб-сервісу, він готовий для використання користувачами, забезпечуючи їм зручний та швидкий доступ до пошуку та бронювання авіабілетів.

3.1 Опис засобів програмування веб-сервісу

Опис API

Для розробки інформаційної системи було обрано фреймворк FastAPI.

FastAPI – дуже простий та досить швидкий фреймворк, призначений саме для організації REST API сервісів. Фреймворк написаний на Python, може працювати на різних операційних системах, має хорошу документацію. Головна зручність його використання полягає у простоті кодингу, а крім того – в автоматичному формуванні документації за API у форматі Open API. Крім цього, FastAPI є вбудована асинхронність, валідація параметрів запитів.

Принципи роботи веб-сервісів з використанням FastAPI включають:

1. Декларативний підхід.

FastAPI пропонує декларативний стиль програмування, де користувач описує структуру та функціональність веб-сервісу за допомогою анотацій та декораторів. Це полегшує створення та розуміння коду і дозволяє швидко створювати потужні API.

2. Автоматична валідація даних.

FastAPI використовує Pydantic, бібліотеку для валідації даних, що дозволяє визначати моделі з обмеженнями та правилами для полів. При обробці запиту, FastAPI автоматично валідує та серіалізує дані, що дозволяє забезпечити правильність та безпеку даних.

3. Асинхронна обробка запитів.

FastAPI підтримує асинхронне програмування, що дозволяє обробляти багатопотокові запити без блокування виконання інших запитів. Це забезпечує високу продуктивність та швидку відповідь сервера навіть при великому навантаженні.

4. Вбудована документація.

FastAPI автоматично генерує документацію API на основі анотацій та докстрінгів вашого коду. Це дозволяє швидко створювати зрозумілу та докладну документацію для вашого веб-сервісу, включаючи доступні маршрути, параметри та приклади використання.

5. Швидке розгортання.

FastAPI дозволяє легко розгорнути ваш веб-сервіс на різних платформах і середовищах, таких як сервери на базі Docker або хмарні платформи. Завдяки вбудованому веб-серверу ASGI, FastAPI забезпечує швидку та стабільну роботу навіть при великому навантаженні.

6. Автентифікація та авторизація.

FastAPI підтримує різні механізми автентифікації та авторизації, такі як токени авторизації або сесії. Ви можете легко налаштувати правила доступу до ресурсів вашого веб-сервісу та забезпечити безпеку даних та обмеження доступу до функціональності.

7. Обробка помилок.

FastAPI дозволяє визначати функції-обробники для обробки помилок, що виникають під час обробки запитів. Це дозволяє забезпечити гнучку обробку помилок та відповідність статусних кодів помилок.

Отож, FastAPI є одним з сучасних фреймворків для розробки веб-сервісів, який дозволяє швидко створювати ефективні та масштабовані API. Він підтримує стандарти RestfulAPI та надає багато утиліт для автоматичної документації API, валідації даних та автогенерації коду клієнта. Заснований на мові програмування Python, FastAPI пропонує декларативний підхід до опису маршрутів та операцій API, що спрощує розробку та підтримку веб-сервісів.

За валідацію відповідає Pydantic. Pydantic застосовує підказки типу під час виконання та видає помилки, коли дані недійсні. Це надає можливість валідувати дані на етапі ініціалізації екземпляру сутності.

Окрім того FastAPI поставляється разом із залежністю Starlette. Він виступає в ролі проміжного програмного забезпечення між FastAPI та асинхронним веб-сервером. Дана залежність дозволяє використовувати cookies, сесії.

FastAPI містить пакети для автоматичної генерації REST API. Це архітектурний зразок для створення веб-сервісів. REST – це набір правил, в яких викладено найкращі практики обміну даними між клієнтами та серверами. Вони використовують HTTP-запити для керування даними та зв'язку з веб-службами. API-інтерфейси REST не зберігають стан, кешуються та узгоджені. Вони чудово підходять для створення універсальних та масштабованих веб-застосунків.

У REST API здебільшого використовується формат JSON. Він походить від мови Java Script. Це спрощує зв'язок між інтерфейсом користувача та бізнес-логікою.

Для взаємодії із базою даних, підключено пакет sqlalchemy.

Компонент бізнес-логіки є сервісом, який описує певні REST-ресурси. Вони можуть реалізовувати HTTP-методи, до яких звертаються з використанням інтерфейсу.

Комунікація веб-клієнта та веб-сервера відбувається за допомогою протоколу HTTP. Цей протокол визначає правила та формат обміну даними між клієнтом і сервером.

У циклі запит-відповідь клієнт надсилає HTTP-запит до сервера, у якому вказується потрібна дія, наприклад, отримати певний ресурс або виконати певну операцію. Запит може містити додаткові параметри, які передаються серверу для визначення точних вимог клієнта.

Після отримання запиту сервер обробляє його і генерує відповідь. Відповідь містить код стану, який вказує, чи був запит успішним, а також необхідні дані або повідомлення для клієнта. Сервер надсилає цю відповідь клієнту за допомогою HTTP-протоколу (рисунок 3.1).

Цей цикл запит-відповідь дозволяє клієнту та серверу взаємодіяти і обмінюватися даними відповідно до визначених правил протоколу HTTP. Це основна технологія, що лежить в основі веб-застосунків та дозволяє взаємодіяти з веб-серверами для отримання ресурсів, виконання операцій та обміну інформацією.

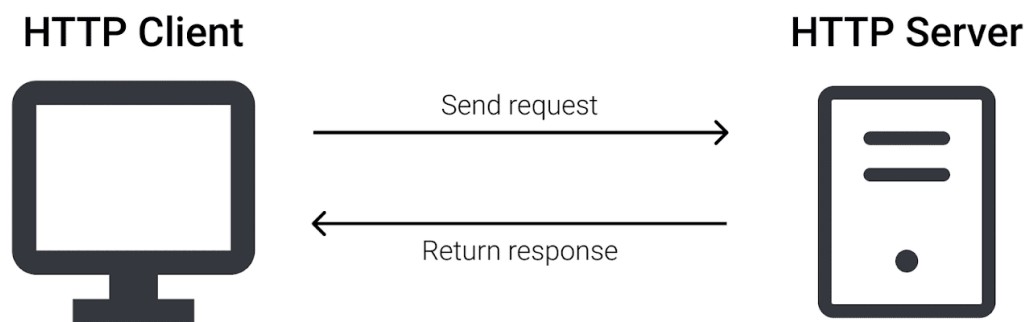


Рисунок 3.1 — Комунікація веб-клієнта та веб-сервера

Повідомлення HTTP — це спосіб обміну даними між сервером і клієнтом. Кожне HTTP-повідомлення відповідає строгому формату відповідно до специфікації. Повідомлення HTTP-запиту містить рядок запиту, поля заголовка

та додаткові дані повідомлення.

Метод GET — отримання даних, PUT та PATCH використовують для зміни атрибутів певних об'єктів, POST використовується для створення об'єкту, метод DELETE — для видалення об'єкту.

Опис розробки клієнтської частини веб-застосунків

Для розробки фронтенду було вибрано фреймворк Angular. Angular представляє фреймворк від компанії Google для створення клієнтських програм. Angular надає таку функціональність як двостороннє зв'язування, що дозволяє динамічно змінювати дані в одному місці інтерфейсу при зміні даних моделі в іншому, шаблони, маршрутизація і так далі.

Однією з ключових особливостей Angular є те, що він використовує мову програмування TypeScript.

Angular використовує HTML як мову шаблонів, що дозволяє розширювати її за допомогою директив. Ці директиви додають інформацію про поведінку елементів в коді. Використання директив дозволяє розробникам зосередитися на логіці та працювати більш продуктивно. Вони також можуть бути повторно використані, що поліпшує читабельність коду.

Правильний підхід до використання Angular дозволяє розробляти програми швидко, навіть якщо вони мають великий обсяг. Angular використовує схему Model-View-Controller (MVC), яка дозволяє розділити логіку, представлення та дані програми. Це допомагає підтримувати чистоту коду і полегшує розробку та розуміння програмного рішення.

Усі компоненти програми розміщуються в середині моделі Angular, що дає зручну можливість керувати ними. Розбиття програми на модулі дозволяє завантажувати лише необхідні сервіси, що сприяє ефективному виконанню автоматичного тестування.

Angular допомагає прив'язувати компоненти програми один до одного, передавати дані, тощо. Фреймворк дозволяє створювати не лише веб-програми. З його допомогою можна писати код, який може бути адаптований до іншого середовища. Наприклад, програма зможе працювати в мобільній або десктопній

операційній системі.

3.2 Реалізація веб-сервісу

Для початку розробки веб-застосунку за допомогою FastApi, необхідно виконати опис використовуваних сутностей.

Ці сутності можна поділити на два види: модель та схема.

Класи моделі наслідують пакет sqlalchemy та використовуються для об'єктно-реляційного відображення всіх даних. Даний клас описує таблиці та її відношення (рисунок 3.2).

```

1  from sqlalchemy import Boolean, Column, ForeignKey, Integer, String
2  from sqlalchemy.orm import relationship
3  from sqlalchemy.orm import declarative_base
4
5  from sqlalchemy.orm import Session
6
7  Base = declarative_base()
8
9  class Airports(Base):
10     __tablename__ = "airports_data"
11
12     iata_code = Column(String, primary_key=True, unique=True)
13     city_airport = Column(String, nullable=False)
14     country = Column(String, nullable=False)
15
16     def get_iata_by_city(db: Session, city: str):
17         citi_iata = db.query(Airports.iata_code).filter(Airports.city_airport == city).first()
18         if citi_iata == None:
19             return None
20         return citi_iata[0]
21
22     def get_all_airports(db: Session) -> list:
23         airports = db.query(Airports.city_airport).all()
24         if airports == None:
25             return None
26         return airports

```

Рисунок 3.2 — Структура таблиці 'airports_data'

Натомість клас схеми наслідує пакет pydantic для валідації даних. Кожен з атрибутів класів має володіти анотацією типу та може отримувати значення за замовчування.

Обробник запитів у FastApi являється у вигляді декорованої функції. Клас містить декоратори для кожного HTTP-методу (рисунок 3.3-3.6).

```

@main_router.post("/flight/search", response_model=List[schemas.FlightsResult])
def form_url(searchParams: schemas.SearchParams, db: Session = Depends(get_db)):
    get_url = GetUrl(searchParams, db)

    parsing = Scraping(get_url.url(), searchParams.date)
    parsing.close_cookies()

    return parsing.get_data()

@main_router.get("/flight/search", response_model=List[schemas.Airports])
def all_airports(db: Session = Depends(get_db)) -> list:
    return models.get_all_airports(db)

```

Рисунок 3.3 — Приклад обробки HTTP запиту методами POST та GET для отримання деталей про наявні авіабілету та інформацію про аеропорти, які може вибрати користувач.

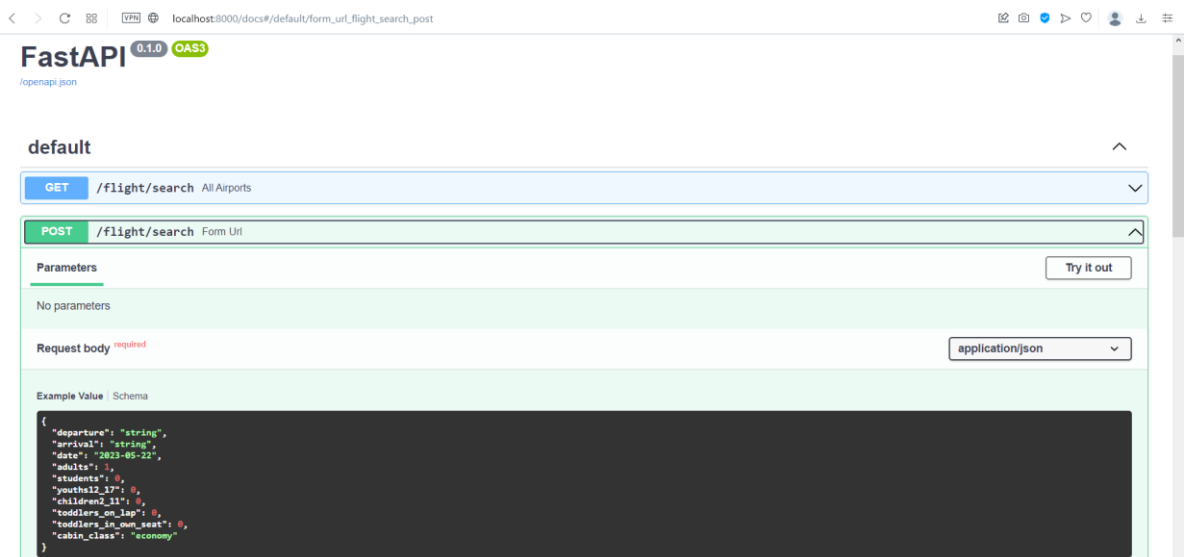


Рисунок 3.4 — Інтерактивна документація API (опис POST та GET методів).

```

FlightsResult v {
  date_time_from* string($date-time)
                  title: Date Time From
  date_time_in* string($date-time)
                 title: Date Time In
  total_price* string
                title: Total Price
  price_for_person string
                  title: Price For Person
                  default:
  airport_departure_IATA* string
                          title: Airport Departure Iata
  airport_departure_name* string
                          title: Airport Departure Name
  airport_arrival_IATA* string
                       title: Airport Arrival Iata
  airport_arrival_name* string
                       title: Airport Arrival Name
  route*
    Route v {
      stops* string
              title: Stops
      transfers Transfers v [
                title: Transfers
                default: List []
                Changes v {
                  airport_IATA* string
                              title: Airport Iata
                  airport_name* string
                              title: Airport Name
                  waiting_time* string
                              title: Waiting Time
                }
      ]
    }
  duration* string
            title: Duration
  cabin_class* string
               title: Cabin Class
  company* string
            title: Company
  url* string
        title: Url
}

```

Рисунок 3.5 — Інтерактивна документація API (опис схеми з даними про наявні авіабілет)

```

SearchParams v {
  departure* string
             title: Departure
  arrival* string
            title: Arrival
  date string($date)
        title: Date
        default: 2023-05-22
  adults integer
          title: Adults
          default: 1
  students integer
            title: Students
            default: 0
  youths12_17 integer
              title: Youths12 17
              default: 0
  children2_11 integer
                title: Children2 11
                default: 0
  toddlers_on_lap integer
                  title: Toddlers On Lap
                  default: 0
  toddlers_in_own_seat integer
                        title: Toddlers In Own Seat
                        default: 0
  cabin_class string
               title: Cabin Class
               default: economy
}

```

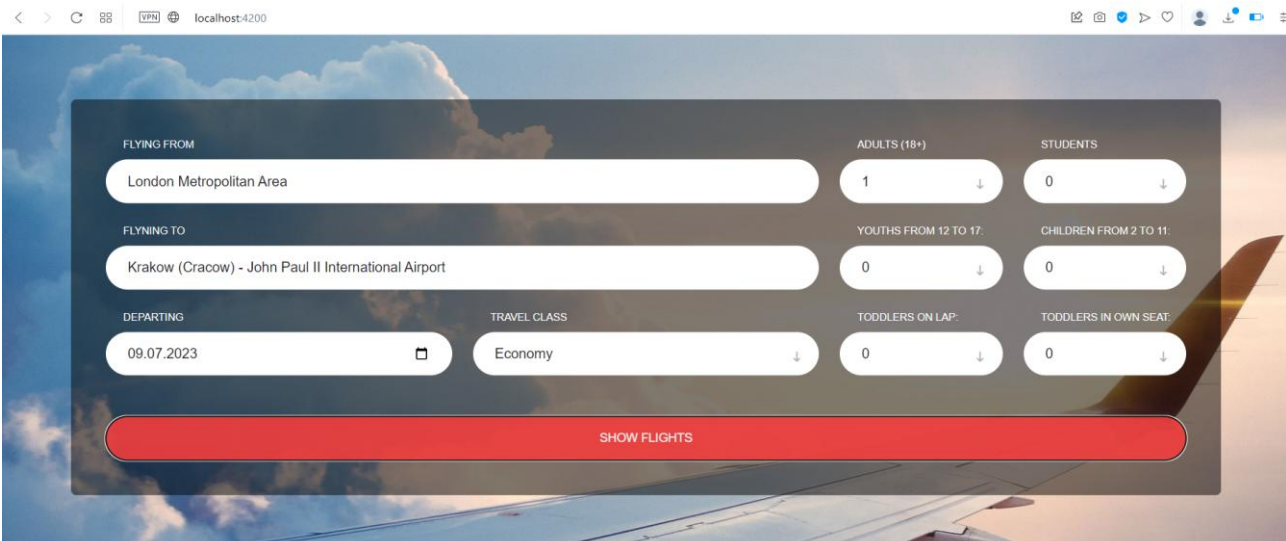
Рисунок 3.6 — Інтерактивна документація API (опис схеми з параметрами для пошуку авіабілетів).

3.3 Інструкція користувача

В розробленому веб-застосунку реалізовано основний функціонал для пошуку та бронювання авіабілетів.

При відкритті застосунку, користувачу пропонується ввести параметри для пошуку авіабілетів. Користувач повинен ввести аеропорт з якого він хоче летіти, а також куди він хоче летіти. Окрім того необхідно вказати дату вильоту та клас, яким пасажир хоче летіти (Economy, Premium economy, Business, First). Для отримання загальної ціни за переліт групи людей, користувач повинен ввести кількість пасажирів (рисунк 3.7). Авіакомпанії оцінюють послуги по різному, в залежності від віку пасажирів, тож виділено наступні вікові групи пасажирів:

- Дорослі (18+)
- Студенти
- Діти (12-17 років)
- Діти (2-11 років)
- Немовля (на колінах)
- Немовля (власне сидіння)



The screenshot displays a flight search form with the following fields and values:

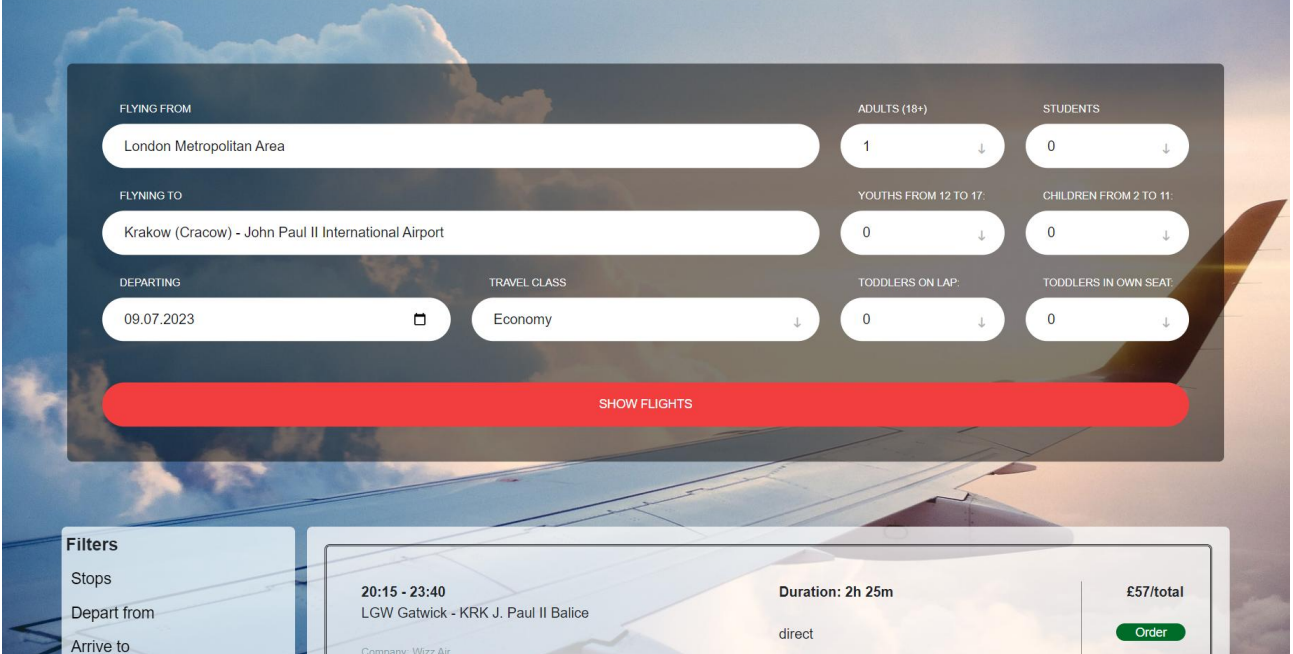
FLYING FROM	ADULTS (18+)	STUDENTS	
London Metropolitan Area	1	0	
FLYING TO	YOUTHS FROM 12 TO 17:	CHILDREN FROM 2 TO 11:	
Krakow (Cracow) - John Paul II International Airport	0	0	
DEPARTING	TRAVEL CLASS	TODDLERS ON LAP:	TODDLERS IN OWN SEAT:
09.07.2023	Economy	0	0

A red button labeled "SHOW FLIGHTS" is located at the bottom of the form.

Рисунок 3.7 — Форма з полями для вводу параметрів, за якими користувач шукає авіабілет

Після натиску кнопки “SHOW FLIGHTS”, система потребуватиме трохи часу щоб отримати знайти всі наявні авіабілеті(рисунок 3.8-3.9)

Після отримання даних, користувач зможе переглянути список з наявними варіантами подорожі:



The screenshot displays a flight search form with the following fields and values:

- FLYING FROM: London Metropolitan Area
- FLYING TO: Krakow (Cracow) - John Paul II International Airport
- DEPARTING: 09.07.2023
- TRAVEL CLASS: Economy
- ADULTS (18+): 1
- STUDENTS: 0
- YOUTHS FROM 12 TO 17: 0
- CHILDREN FROM 2 TO 11: 0
- TODDLERS ON LAP: 0
- TODDLERS IN OWN SEAT: 0

A red button labeled "SHOW FLIGHTS" is positioned below the search criteria.

Below the search form, a flight result is displayed with the following details:

- Filters:** Stops, Depart from, Arrive to
- Time:** 20:15 - 23:40
- Route:** LGW Gatwick - KRK J. Paul II Balice
- Company:** Wizz Air
- Duration:** 2h 25m
- Flight Type:** direct
- Price:** £57/total
- Action:** Order

Рисунок 3.8 — Список наявних авіабілетів у системі



Рисунок 3.9 — Список наявних авіабілетів у системі

Користувач має можливість фільтрувати отримані дані за певними фільтрами (до прикладу кількість зупинок: прямий рейс, 1 пересадка, 2+ пересадки) (рисунок 3.10)

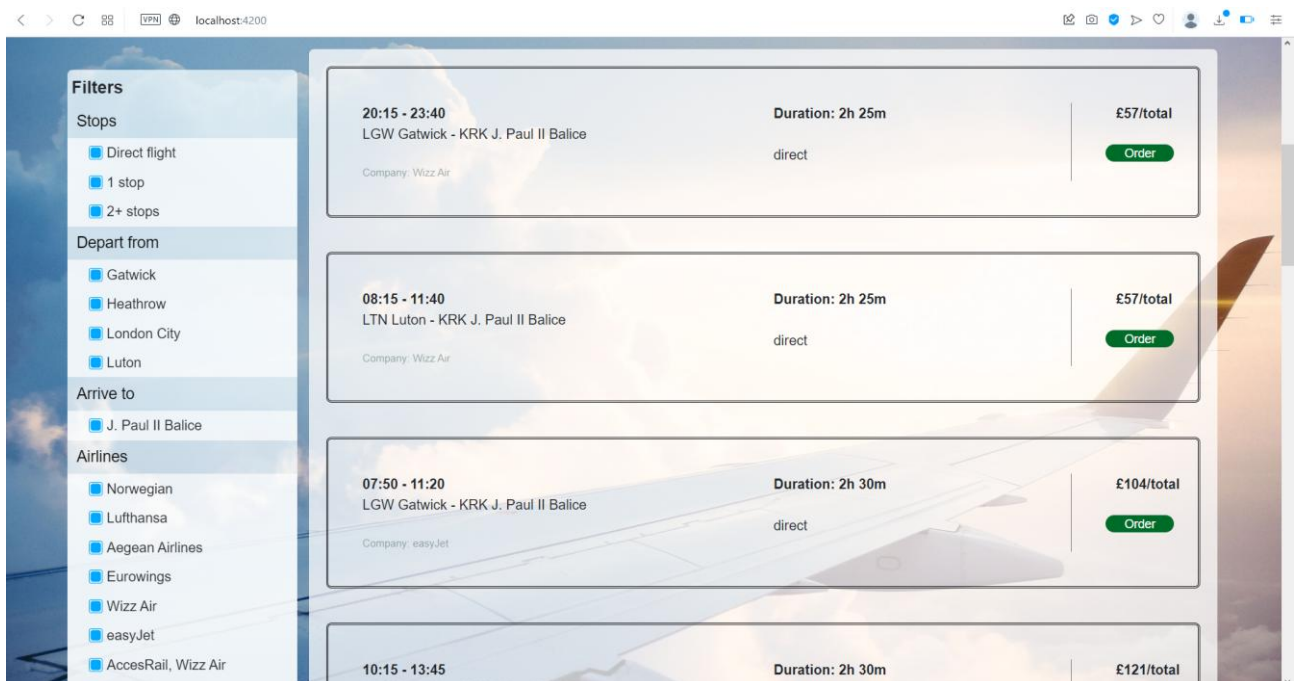


Рисунок 3.10 — Зображення фільтрів, наявних у даному веб-сервісі



Рисунок 3.11 — Результат застосування фільтру кількості зупинок

При наявності пересадки, користувач має змогу отримати інформацію про місце пересадки та час очікування (рисунок 3.12).

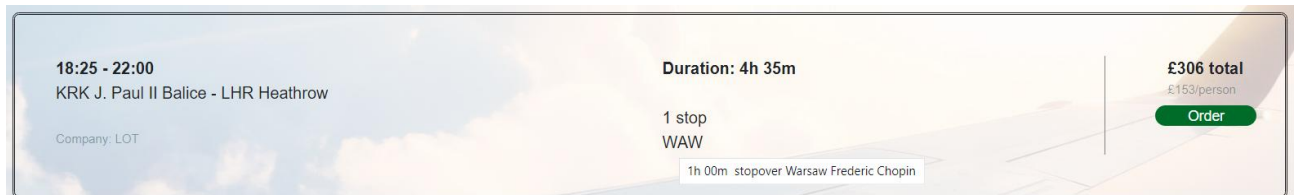


Рисунок 3.12 — Дані про зміну аеропорту та час очікування в аеропорту

Після того, як користувач натисне на кнопку “Order”, його перенаправить на сайт авіакомпанії, де він зможе оформити бронювання (рисунок 3.13).

The screenshot displays the LOT website's flight selection interface. At the top, there is a navigation menu with options: SEARCH, FLIGHTS, PASSENGERS, EXTRAS, PAYMENT, and CONFIRMATION. Below this, the user is prompted to 'Select your flights' and has chosen the route 'Departure flight Cracow (KRK) → London (LHR)'. A date selector shows 'Saturday, 10 June 2023' is selected, with other dates and prices visible: Wed 07.06 (136,80 GBP), Thu 08.06 (136,80 GBP), Fri 09.06 (136,80 GBP), Sat 10.06 (152,80 GBP), Sun 11.06 (152,80 GBP), Mon 12.06 (136,80 GBP), and Tue 13.06 (136,80 GBP). The flight options are presented in a table with three columns: ECONOMY, PREMIUM, and BUSINESS. Each row represents a different flight option with its departure time, duration, arrival time, and aircraft. The Economy class is available for all options at 152,80 GBP, while Premium and Business classes are either 'not available' or priced at 389,90 GBP. A 'Sort Flights' button is visible on the left.

Flight Details	ECONOMY	PREMIUM	BUSINESS
KRK → LHR 05:35 → 09:25 4:50 h WAW LO3910 / LO281	152,80 GBP	not available	389,90 GBP
KRK → LHR 05:35 → 17:20 12:45 h WAW LO3910 / LO279	152,80 GBP	not available	389,90 GBP
KRK → LHR 05:35 → 22:00 17:25 h WAW LO3910 / LO285	152,80 GBP	not available	389,90 GBP

Рисунок 3.13 — Процес оформлення обраного квитка безпосередньо на сайті авіакомпанії LOT

ВИСНОВОК

В результаті виконання кваліфікаційної роботи, були виконані наступні задачі:

По-перше, було описано термін веб-сервіси. Розглянуто роль та значення веб-сервісів пошуку та бронювання авіабілетів у сучасному туристичному секторі. Виконано огляд існуючих веб-сервісів пошуку та бронювання авіабілетів. Сформульовано принципи архітектури веб-сервісів пошуку та бронювання авіабілетів. Розглянуто та проаналізовано існуючі веб-сервіси для пошуку та бронювання авіабілетів. Розглянуто механізми взаємодії з базами

даних авіакомпаній та інших постачальників даних

По-друге, розглянуто технології побудови веб-сервісів пошуку та бронювання авіабілетів, розглянуто техніку вебскрапінгу, на основі виконаних досліджень обрано технології для розробки системи.

По-третє, проаналізовано та описано технології для розробки веб-сервісу. Описано використовувані сутності для веб-застосунку. Розроблено прототип програмної системи бронювання авіабілетів, який реалізовує основний функціонал, що очікується від подібних систем. Для програмної реалізації було застосовано фреймворки FastApi для розробки бекенду, та Angular-фреймворк для розробки фронтенду, а також підключена база даних PostgreSQL для роботи з даними програми.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація по FastApi: веб-сайт. URL: <https://fastapi.tiangolo.com>.
2. Плєскач В.Л. , Затонацька Т.Г. Інформаційні системи і технології на підприємствах: підручник К.:Знання, 2011. 718с.
3. Документація по Angular : веб-сайт. URL: <https://angular.io/docs>
4. Документація по PostgreSQL: веб-сайт. URL: <https://www.postgresql.org/docs/>
5. Skyscanner: веб-сайт. URL: <https://www.skyscanner.net>
6. Коваленко О. С., Добровська Л. М. Проектування інформаційних систем: Загальні питання теорії проектування ІС: підручник. Київ: 2020. 192с.
7. Електронний підручник по CSS: веб-сайт. URL: <https://www.w3schools.com/css/default.asp>
8. Kiwi: веб-сайт. URL: <https://www.kiwi.com/en/>
9. Bootstrap 4: веб-сайт. URL: <https://getbootstrap.com>
10. Плєскач В. Л., Затонацька Т. Г. Електронна комерція: підручник К.: Знання, 2007. 535 с.
11. Гаврилов В. П. Інформаційні системи і технології в туризмі: лабораторний практикум для студентів напряму підготовки "Туризм": підручник – Х. : ХНЕУ ім. С. Кузнеця, 2014. 174 с.
12. Онлайн-бронювання: веб-сайт. URL: <https://uk.wikipedia.org/wiki/Онлайн-бронювання>
13. Booking.com: веб-сайт. URL: <https://www.booking.com/>
14. Онлайн резервування та придбання квитків – Укрзалізниця: веб-сайт. URL: <https://booking.uz.gov.ua/>
15. Сценарій використання: веб-сайт. URL: https://uk.wikipedia.org/wiki/Сценарій_використання
16. Expedia: веб-сайт. URL: <https://www.expedia.co.uk>
17. Travelocity: веб-сайт. URL: www.travelocity.com
18. Priceline: веб-сайт. URL: www.priceline.com

19. Orbitz: веб-сайт. URL: www.orbitz.com
20. Cheapoair: веб-сайт. URL: www.cheapoair.com
21. Momondo: веб-сайт. URL: www.momondo.com
22. Google flights: веб-сайт. URL: www.google.com/flights
23. Електронний підручник по HTML: веб-сайт. URL: <https://www.w3schools.com/html/default.asp>
24. Електронний підручник по JavaScript: веб-сайт. URL: <https://www.w3schools.com/js/default.asp>
25. Документація по DBeaver: веб-сайт. URL: <https://dbeaver.com/docs/wiki/>
26. Документація по SQLAlchemy: веб-сайт. URL: <https://docs.sqlalchemy.org/en/20/dialects/postgresql.html>
27. Документація по Pydantic: веб-сайт. URL: <https://docs.pydantic.dev/latest/>
28. Subbu Allamaraju. RESTful Web Services Cookbook: підручник. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. URL: <http://web-algarve.com/books/web%20services%20&%20REST/RESTful%20Web%20Services%20Cookbook.pdf>
29. Франчук В.М., Галицький О.В. Вибір системи управління вмістом сайту: науковий часопис НПУ імені М. П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання. 2014. 20с. URL: <https://vfranchuk.fi.npu.edu.ua/images/files/statty/43.pdf>
30. Історія розвитку Web: веб-сайт. URL: <https://uk.wikipedia.org/wiki/Інтернет>
31. Глушаков С. В. Програмування WEB-сторінок: навч. посіб. Харків: Фоліо, 2004. 390 с.
32. Горнаков С. Г. Освоюємо популярні системи керування сайтом (CMS): підручник: ДМК Пресс, 2009. 336 с.

33. How To Become A Web Developer — Everything You Need To Know : веб-сайт. URL: <https://careerfoundry.com/en/blog/web-development/what-does-it-take-to-becomea-web-developer-everything-you-need-to-know-before-getting-started/>
34. Крокфорд Дуглас. JavaScript. Сильні сторони, 2016. — 176 с
35. Хавербеке М. Виразний JavaScript: навч. посібник: Вид. 2-ге, 2015. 745 с.
36. Ніксон Р. Створюємо динамічні веб-сайти за допомогою PHP, MySQL, JavaScript, CSS і HTML5: навч. посібник: 2016. - 510 с.
37. What is REST?: веб-сайт. URL: <http://www.restapitutorial.com/lessons/whatisrest.html>
38. Кейт Джонс. DOM Scripting: Web Design with JavaScript and the Document Object Model: підручник. 2005. 368 с.
39. Особливості веб-додатків: веб-сайт. URL: <http://sites.znu.edu.ua/webprog/lect/1191.ukr.html>
40. Різниця між веб-сайтом та веб-додатком: веб-сайт. URL: <https://dinarys.com/blog/website-vs-webapplication>
41. Основи HTML: веб-сайт. URL: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics
42. Основи JavaScript: веб-сайт. URL: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics
43. Документація по Visual Studio Code : веб-сайт. URL: <https://code.visualstudio.com/docs>
44. Eric Matthes. Python crash course. No starch press, 2016.
45. Client Server Architecture: веб-сайт. URL: https://cio-wiki.org/wiki/Client_Server_Architecture.

ДОДАТКИ

ДОДАТОК А

```
def get_route(self):

    routes = [self.pageSource[i].find_element(By.CLASS_NAME, 'JWE0').text for i in range(len(self.pageSource))]

    for i in range(len(self.pageSource)):

        if routes[i] == 'direct' or routes[i].split(' ')[0] == 0:
            self.data[i]['route'] = {'stops' : "direct", }

        else:

            stops = routes[i].split('\n')
            if int(stops[0][0]) > 1:
                stops_str = str(stops[0][0]) + " stops"
            else:
                stops_str = str(stops[0][0]) + " stop"
            self.data[i]['route'] = {'stops' : stops_str, }

            time = self.pageSource[i].find_elements(By.CSS_SELECTOR, 'div.c_cgF.c_cgF-mod-variant-full-airport > span')
            route = []

            for t in time:
                airport = t.text

                info = t.find_element(By.TAG_NAME, 'span').get_attribute('title')
                time = info.split('stopover')[0]
                change = re.search(r'<b>(.*?)</b>', info).group(1)
                route.append({'airport_IATA': airport, 'airport_name': change, 'waiting_time' : time})

            self.data[i]['route']['transfers'] = route
```

Рисунок А1 – Застосування веб-скрапінгу для отримання даних про маршрут рейсу

```

api > config.py > ...
1  from functools import lru_cache
2
3  from pydantic import BaseSettings
4  import toml
5
6  toml_config = toml.load("config.toml")
7  class Settings(BaseSettings):
8      SECRET_KEY = toml_config['SECRET_KEY']
9      SQLALCHEMY_ECHO = toml_config['SQLALCHEMY_ECHO']
10     ENVIRONMENT = toml_config['ENVIRONMENT']
11
12     class DevSettings(Settings):
13         DEBUG = True
14
15         DB_HOST = toml_config['DB_HOST']
16         DB_USER = toml_config['DB_USER']
17         DB_PASS = toml_config['DB_PASS']
18         DB_PORT = toml_config['DB_PORT']
19         DB_NAME = toml_config['DB_NAME']
20         SQLALCHEMY_DATABASE_URL = f"postgresql+psycopg2://{DB_USER}:{DB_PASS}@{DB_HOST}:{DB_PORT}/{DB_NAME}"
21
22
23     @lru_cache()
24     def get_config():
25         if Settings().ENVIRONMENT == "development":
26             return DevSettings()
27
28         else:
29             return DevSettings()

```

Рисунок 1Б – Параметри конфігурації для сервера бази даних

```

api > models.py > ...
1  from sqlalchemy import Boolean, Column, ForeignKey, Integer, String
2  from sqlalchemy.orm import relationship
3  from sqlalchemy.orm import declarative_base
4
5  from sqlalchemy.orm import Session
6
7  Base = declarative_base()
8
9  class Airports(Base):
10     __tablename__ = "airports_data"
11
12     iata_code = Column(String, primary_key=True, unique=True)
13     city_airport = Column(String, nullable=False)
14     country = Column(String, nullable=False)
15
16     def get_iata_by_city(db: Session, city: str):
17         citi_iata = db.query(Airports.iata_code).filter(Airports.city_airport == city).first()
18         if citi_iata == None:
19             return None
20         return citi_iata[0]
21
22     def get_all_airports(db: Session) -> list:
23         airports = db.query(Airports.city_airport).all()
24         if airports == None:
25             return None
26         return airports

```

Рисунок 2Б – Клас опису даних бази даних

```

<div *ngFor="let ticket of searchResults" class="ticket-block">
  <div class="row">
    <div class="col-md-6">
      <div class="row">
        <div><b>{{ticket.date_time_from | date:"HH:mm"}} - {{ticket.date_time_in | date:"HH:mm"}}</b></div>
      </div>
      <div class="row">
        <div>{{ticket.airport_departure_IATA}} {{ticket.airport_departure_name}} - {{ticket.airport_arrival_IATA}}
          {{ticket.airport_arrival_name}}</div>
      </div>
      <div class="row"><br></div>
      <div class="row">
        <div id="additional_info">Company: {{ticket.company}}</div>
      </div>
    </div>
    <div class="col-md-4">
      <div class="row">
        <div><b>Duration: {{ticket.duration}}</b></div>
      </div>
      <div class="row"><br></div>
      <div class="row">
        <div>{{ticket.route.stops}}</div>
      </div>
      <div class="row">
        <div *ngIf="ticket.route" class="ticket-route">
          <div *ngFor="let route of ticket.route.transfers">
            <div title="{{route.waiting_time}} stopover {{route.airport_name}}">{{route.airport_IATA}} </div>
          </div>
        </div>
      </div>
    </div>
    <div class="col-md-1">
      <div class="outer">
        <div class="inner"></div>
      </div>
    </div>
    <div class="col-md-1">
      <div class="row">
        <div><b>{{ticket.total_price}}</b></div><br>
        <div id="additional_info">{{ticket.price_for_person}}/person</div><br>
      </div>
      <div class="row">
        <a id="order-button" href={{ticket.url}}>Order</a>
      </div>
    </div>
  </div>

```

Рисунок 1В – HTML-код представлення списку авіабілетів

```

<div class="col-md-8">
  <div class="form-group">
    <span class="form-label">Flying from</span>

    <select class="form-control" id="departure" name="Service" formControlName="departure" (change)="departureChanged()">
      <option placeholder="City or airport" [value]="serv.city_airport" *ngFor="let serv of departureCities">
        {{serv.city_airport}}
      </option>
    </select>
  </div>
</div>

```

Рисунок 2В – HTML-код створення випадючого списку із назвами аеропортів

```
<button class="filters-sub-header" id="filters-stops"
  (click)="toggleSection(filterStops)">Stops</button>
<div #filterStops class="filters-sub-section">
  <div *ngFor="let filter of filterState.stops.values; let i = index" class="filters-item">
    <div style="display: flex; align-items: center;">
      <input type="checkbox" class="checkbox-input" id="checkbox-{{i}}"
        (change)="filtersChange($event, filter, 'stops')" checked>
      <label for="checkbox-{{i}}" class="checkbox-label">
        <span class="checkbox">
          </span>
        </label>
      <span class="filter-value">
        {{filter.title}}
      </span>
    </div>
    <div class="filter-value-info">
    </div>
  </div>
</div>
</div>
<div>
```

Рисунок 3В – HTML-код додавання фільтрів пошуку авіабілетів