

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**Факультет кібербезпеки та програмної інженерії
Кафедра інженерії програмного забезпечення**

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
Катерина НЕСТЕРЕНКО
“ ” 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
“БАКАЛАВР”**

Тема: Сайт довідник розкладів польотів літаків

Виконавець: Свиридовський Микита Сергійович

Керівник: к.т.н. Кучеренко Володимир Миколайович

Нормоконтролер: Варнавський В'ячеслав Володимирович

Київ 2024

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки та програмної інженерії

Кафедра інженерії програмного забезпечення

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Програмне забезпечення систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Катерина НЕСТЕРЕНКО

" ____ " _____ 2024 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи студента
Свиридовського Микити Сергійовича

1. Тема проєкту: «Сайт довідник розкладів польотів літаків» затверджена наказом ректора від 8.12.2024 р. № 2483/ст
2. Термін виконання проєкту: з 3.01.2024 р. до 29.02.2024 р.
3. Вихідні дані до проєкту: програмний продукт розробити у вигляді web-застосунку довідник розкладів польотів літаків.
4. Зміст пояснювальної записки:
 1. Задача розробки веб застосунків який буде виступати, як довідник розкладів польотів літаків.
 2. Огляд вибору засобів реалізації програмного застосунку.
 3. Програмна реалізація застосунку.
 4. Результати роботи програми.
5. Перелік обов'язкових слайдів презентації:
 1. - Проблеми сучасних сайтів
 2. - Оцінка сайтів в сучасному світі
 3. - Інструменти для реалізації додатків в сучасному світі
 4. - Аналіз побудови сайтів довідників
 5. - Аналіз користувачів для додатку
 6. - Досвід користувача в додатках або що таке UX в сучасному світі
6. Календарний план-графік

№ пор	Завдання	Термін виконання	Відмітка про виконання
1.	Складання та затвердження графіку кваліфікаційної роботи Написання 1 розділу, представлення керівнику	03.01.24 – 09.01.24	
2.	Попередній друк 1 розділу та допоміжних сторінок (чернетка) — титульної, завдання, графіка, реферат, список скорочень, зміст, вступ, список джерел, 1-й нормоконтроль.	10.01.24 – 17.01.24	
3.	Написання 2 розділу, представлення керівнику	18.01.24 – 22.01.24	
4.	Написання 3 розділу, представлення керівнику	23.01.24 – 26.01.24	
5.	Написання 4 розділу, представлення керівнику	27.01.24 – 31.01.24	
6.	Загальне редагування та друк пояснювальної записки, графічного матеріалу	01.02.24 – 09.02.24	
7.	Проходження нормоконтролю, перевірка на антиплагіат, перепліт пояснювальної записки.	01.02.24 – 05.02.24	
8.	Розробка тексту доповіді. Оформлення графічного матеріалу для презентації	05.02.24 – 09.02.24	
9.	Отримання відгуку керівника, рецензії.	10.02.24 – 22.02.24	
10.	Підготовка матеріалів для передачі секретарю ДЕК (ПЗ, CD-R з електронними копіями ПЗ, презентації, відгук керівника, рецензія) в папці	23.02.24 – 28.02.24	

7. Дата видачі завдання 3.01.2024

Керівник:

Завдання прийняв до виконання:

Дата

к.т.н. Володимир Кучеренко

Микита СВИ

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Сайт довідник розкладів польотів літаків»: - с. 43, - рис. 26, - табл. 0, 5 - інформаційних джерел.

АВТОМАТИЗОВАНИЙ ПОШУК, РОЗКЛАДІВ ПОЛЬОТІВ ЛІТАКІВ,
ТЕХНОЛОГІЧНИЙ СТЕК VUE 3, WEB-ЗАСТОСУНОК

Об'єкт розробки – застосунок, що дозволяє зручно шукати розклади польотів літаків. Це буде відбуватися за допомогою пошуку рейсів та фільтрації і сортування.

Мета роботи – спростити пошук розкладів польотів літаків

ABSTRACT

Explanatory note to the qualification work "Website directory of aircraft flight schedules": - p., - fig., - table, - information sources.

AUTOMATED SEARCH, FLIGHT SCHEDULES, VUE 3 TECHNOLOGY
STACK, WEB APPLICATION

Property development – an application that allows you to conveniently search for airplane flight schedules. This will be done by searching for flights and filtering and sorting.

Purpose – simplify the search for flight schedule

ЗМІСТ

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. ІСНУЮЧИЙ СТАН, ПРОБЛЕМИ ТА ШЛЯХ ЇЇ ВИРІШЕННЯ.....	9
1.1 Існуючий стан, проблеми.....	9
1.1.1 Інформаційна неповнота.....	10
1.1.2 Неактуальність даних.....	10
1.1.3 Відсутність зручності та функціональності.....	11
1.1.4 Нестача інтеграції з іншими сервісами.....	11
1.2 Вирішення цих проблем.....	12
1.2.1 Проблеми оновлення даних у реальному часі.....	13
1.2.2 Удосконалення інтерфейсу та функціональності.....	14
1.2.3 Інтеграція з іншими сервісами.....	14
1.2.4 Забезпечення безпеки даних.....	15
Висновок.....	15
РОЗДІЛ 2. ВИМОГИ ДО ПРОГРАМНОГО ЗАСТОСУНКУ.....	17
2.1 Функціонал застосунку.....	17
2.1.1 Пошук рейсів.....	18

2.1.2 Відображення розкладів.....	18
2.1.3 Інформація про статус рейсу.....	19
2.1.4 Фільтрації та сортування.....	19
2.2 Інструменти для реалізації застосунку.....	20
2.2.1 Frontend.....	22
2.2.1.1 Vue.....	23
2.2.1.2 Quasar.....	24
2.2.2 Backend.....	25
2.2.2.1 Формат json.....	26
Висновок.....	27
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ.....	28
3.1 Створення схеми даних.....	28
3.2.1 Створення даних до проекту.....	30
3.3 Створення лицьової частини.....	31
3.3.1 Створення структурної схеми веб-сайту.....	33
3.3.2. Реалізація сторінок.....	34
Висновок.....	37
РОЗДІЛ 4. РЕЗУЛЬТАТИ РОБОТИ ПРОГРАМИ.....	39

4.1 Опис додатку.....	39
Висновок.....	41
ВИСНОВКИ	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	43

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ

Клієнт — це програма або пристрій, який звертається до іншого пристрою (сервера) для отримання певної інформації.

Сервер — це комп'ютер або система, яка надає відповіді клієнтам або ініціює ще якісь дії за запитом клієнта.

СУБД — система управління базами даних.

JSON (JavaScript Object Notation) - текстовий формат обміну даними, що базується на JavaScript.

VS Code (Visual Studio Code) - редактор коду.

npm (Node Package Manager) - пакетний менеджер.

ВСТУП

У наш час, коли авіація стає все більш доступною та затребуваною, важливість своєчасного та точного доступу до інформації про розклад польотів незаперечна. Однак багато існуючих сайтів, призначених для надання цієї ключової інформації, стикаються з серйозними обмеженнями, що впливають на зручність використання та достовірність даних, що надаються. Проблема в тому, що багато хто з них надає лише обмежений обсяг інформації про рейси, що робить пошук бажаних маршрутів складним і тимчасовим процесом для користувачів. Це часто призводить до незадоволеності та неповної впевненості у виборі оптимального рейсу. Більше того, відсутність можливості миттєвого оновлення даних призводить до того, що користувачі можуть отримувати неактуальну інформацію про статус рейсів, що, у свою чергу, позначається на їхньому плануванні та організації подорожей.

Наша мета – подолати ці проблеми, створюючи засіб, який не лише надає всю необхідну інформацію про рейси, але й робить це в режимі реального часу, забезпечуючи користувачам найактуальнішу та найдостовірнішу інформацію для їх подорожей. Я планую приділяти особливу увагу тому, щоб зробити процес пошуку та вибору рейсів максимально зручним та простим для всіх наших користувачів. Таке доповнення розкриває детальнішу інформацію про проблеми, з якими стикаються користувачі при використанні існуючих сайтів з розкладом польоті.

РОЗДІЛ 1. ІСНУЮЧИЙ СТАН, ПРОБЛЕМИ ТА ШЛЯХ ЇЇ ВИРІШЕННЯ

1.1. Існуючий стан, проблеми

У сучасному світі, де мобільність та швидкість стали ключовими чинниками життя, авіація стала не просто засобом пересування, але й символом зв'язку та глобальної інтеграції. Однак, серед усього цього прогресу, наявні проблеми, які ускладнюють доступ до інформації та планування подорожей через деякі обмеження та несправності.

У цьому розділі я розгляну основні проблеми, що стикаються користувачі, прагнучи отримати достовірну та актуальну інформацію про розклади польотів літаків. Також, я планую дослідити можливі шляхи вирішення цих проблем, спрямовані на поліпшення досвіду користувачів та забезпечення надійної та зручної інформаційної бази для планування подорожі.

Кафедра ІІЗ				НАУ 21.0410 ІІЗ			
<i>Розроб.</i>	Свиридовський М.С.			Сайт довідник розкладів польотів літаків	<i>Літ.</i>	<i>Лист</i>	<i>Листів</i>
<i>Керівник</i>	Кучеренко В.М.					9	43
					ІІ-501Бз		
<i>Н.-контр.</i>	Варнавський В.В.						

1.1.1. Інформаційна неповнота

- Обмежений перелік авіакомпаній: Деякі довідники містять інформацію лише про частину авіакомпаній або обмежуються лише національними перевізниками, що не враховує всіх доступних варіантів рейсів для пасажирів.
- Недостатня інформація про розклади: Сайти можуть не надавати повного списку рейсів, обмежуючись лише основними або найпопулярнішими маршрутами. Це ускладнює пошук рідкісних чи альтернативних рейсів для мандрівників.
- Відсутність додаткових даних про рейси: Деякі довідники можуть не включати важливі деталі про рейси, такі як тип літака, інформація про пересадки, обмеження багажу або особливості послуг на борту. Це важлива інформація для пасажирів під час вибору оптимального рейсу.
- Неповнота розкладів за часом: Сайти можуть обмежувати доступність розкладів лише на найближчі кілька днів, що ускладнює планування довгострокових поїздок.

1.1.2. Неактуальність даних

- Відсутність оновлень у реальному часі: Деякі сайти не оновлюють інформацію про рейси в режимі реального часу. Це означає, що пасажирів можуть отримати застарілу інформацію про час вильоту/прильоту, затримки або скасування рейсів.
- Недостовірність статусу рейсів: Користувачі можуть зіткнутися з ситуацією, коли інформація про статус рейсу на сайті не відповідає фактичній дійсності через нестачу своєчасного оновлення даних.
- Відсутність інформації про затримки або зміни: Інформація про непередбачені події, такі як затримки через погоду або технічні проблеми, може бути недоступною або неактуальною, що ускладнює планування подорожей для пасажирів.

1.1.3. Відсутність зручності та функціональності

- Сайти можуть мати заплутаний або не інтуїтивний інтерфейс, що робить складнішим пошук інформації про рейси і ускладнює користувачам знаходження необхідних даних.
- Обмежені можливості фільтрації та сортування: Недостатні опції для фільтрації результатів пошуку за параметрами, такими як ціна, час, авіакомпанія або тип літака можуть ускладнювати користувачам знаходження оптимальних варіантів рейсів.
- Відсутність додаткової інформації: Недолік детальної інформації про кожен рейс, такий як послуги на борту, інформація про багаж, карти терміналів та інші важливі деталі може зменшити зручність користування та призвести до невизначеності у пасажирів.
- Недостатня адаптація для мобільних пристроїв: Відсутність мобільної оптимізації або неадекватне відображення на мобільних пристроях може створювати незручності для користувачів, які воліють використовувати сайт на смартфонах або планшетах.

1.1.4. Нестача інтеграції з іншими сервісами

- Відсутність інтеграції з бронюванням готелів: Деякі сайти-довідники не надають можливості швидкої та зручної інтеграції з платформами для бронювання готелів, що дає змогу користувачам перемикатися між різними ресурсами для повного планування поїздки.
- Недолік зв'язку з послугами оренди автомобілів: Відсутність інтеграції з сервісами оренди автомобілів може бути перешкодою для користувачів, які хочуть одразу забронювати автомобіль після прильоту до нового міста.
- Неінтегровані пропозиції щодо туристичних послуг: Нестача зв'язку з туристичними агентствами або платформами для організації екскурсій

або інших додаткових послуг також обмежує вибір користувачів та потребує додаткових пошуків.

- Відсутність зв'язку з платформами для бронювання столиків у ресторанах: Нестача інтеграції із сервісами бронювання ресторанів може створити незручність для мандрівників, які хочуть спланувати своє харчування у новому місці заздалегідь.

1.2. Вирішення цих проблем

Історія прогресу намалювала картину змін, які пронизують кожен аспект нашого життя. У сфері авіації, швидкість та доступність інформації стали важливими складовими для пасажирів, що прагнуть планувати та здійснювати свої подорожі з легкістю. Приклад такого додатку є на рисунку 1.

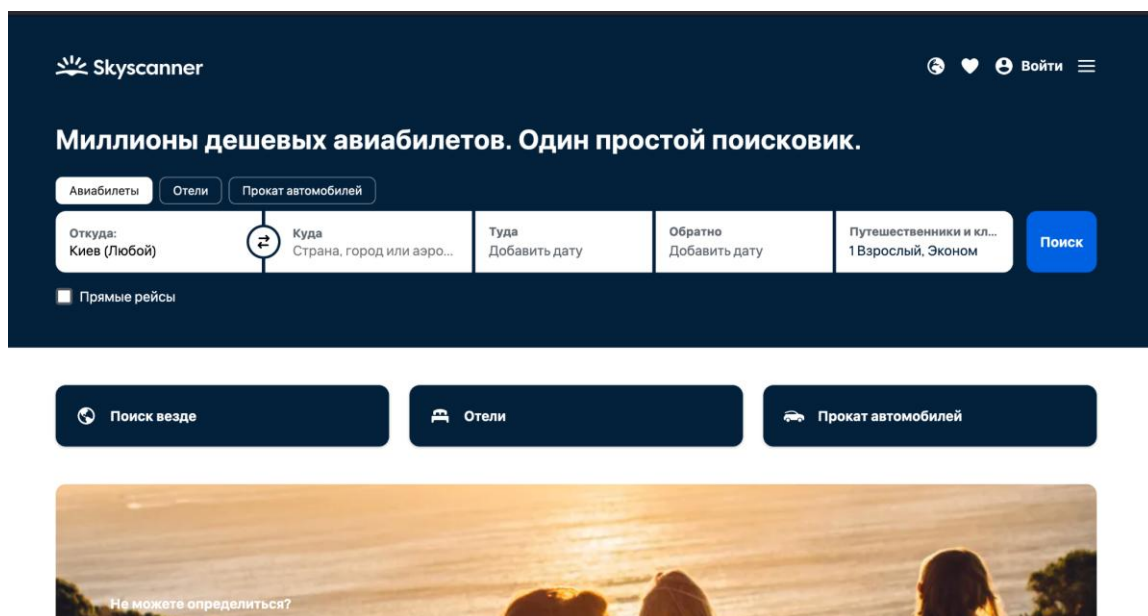


Рис. 1.1. Додаток skyscanner

Цей розділ відкриває нові перспективи та можливості, які стоять перед нами в контексті подолання та вирішення проблем, що супроводжують доступ до інформації про розклади польотів літаків. Я опишу глибину цих проблем та висвітлю інноваційні підходи, які можуть допомогти у створенні надійних та ефективних рішень.

В цьому розділі я розгляну високотехнологічні методи, стратегії та практики, які спрямовані на усунення обмежень та покращення якості послуг у сфері розкладів польотів. Різноманітність інновацій та перспективних рішень відкриває нові можливості для комфортного та ефективного планування подорожей.

В цьому розділі я опишу найсучасніші підходи до вирішення проблем, що виникають під час отримання та використання інформації про розклади польотів літаків, та зробимо крок до майбутнього, де доступ до цієї інформації стане більш зручним та доступним для кожного пасажера.

1.2.1. Проблеми оновлення даних у реальному часі

- Оновлення даних у реальному часі: Використання API та інтеграція з провайдерами даних: Розробка або інтеграція з API провайдерів даних про польоти та розклад авіакомпаній, які надають оновлену інформацію в реальному часі.
- Автоматизація оновлень: Створення системи, яка регулярно перевіряє та оновлює інформацію про рейси без необхідності ручного втручання, гарантуючи своєчасні оновлення для користувачів.
- Повідомлення та моніторинг статусів рейсів: Реалізація функціоналу для надсилання повідомлень користувачам про зміни у статусі рейсів, таких як затримки, скасування або зміни часу, щоб забезпечити актуальну інформацію в режимі реального часу.
- Тестування та оптимізація процесу оновлення: Регулярне тестування системи оновлення даних для усунення можливих збоїв або затримок, забезпечуючи надійність та точність інформації.

1.2.2. Удосконалення інтерфейсу та функціональності

- Оновлений інтерфейс користувача: Перегляд і модернізація дизайну та інтерфейсу, роблення його більш зрозумілим, привабливим і легким у користуванні.
- Розширені опції пошуку: Додавання додаткових параметрів для пошуку, таких як час вильоту/прильоту, кількість пересадок, типи літаків чи авіакомпаній, що дозволить користувачам знайти оптимальні рейси.
- Покращення фільтрації результатів: Впровадження потужних фільтрів для легкого відсіювання непотрібних варіантів, що робить процес пошуку більш ефективним.
- Мобільна оптимізація: Вдосконалення сайту для зручного користування на мобільних пристроях, щоб користувачі могли з легкістю шукати інформацію на будь-якому пристрої.

1.2.3. Інтеграція з іншими сервісами

- Бронювання готелів та проживання: Інтеграція з платформами для бронювання готелів, дозволяючи користувачам забронювати житло безпосередньо після вибору польоту.
- Сполучення з сервісами оренди автомобілів: Можливість забронювати автомобіль на прокат в аеропорту чи у місці призначення через інтеграцію з послугами оренди автомобілів.
- Підключення до сервісів туроператорів: Інтеграція з платформами, що пропонують туристичні послуги, такі як екскурсії, агенти місцевих турів тощо.
- Резервація столиків в ресторанах: Можливість бронювання столиків у ресторанах поблизу аеропорту чи у місці призначення для зручності користувачів.

- Інформація про транспорт та громадський транспорт: Представлення додаткової інформації про громадський транспорт або трансфери для переїзду від аеропорту до місця проживання.

1.2.4. Забезпечення безпеки даних

- Шифрування даних: Застосування сучасних методів шифрування для захисту особистих даних користувачів, таких як інформація про платіжні картки, паролі та особисті дані.
- Політика конфіденційності: Ретельне формулювання та дотримання політики конфіденційності, де чітко визначені правила зберігання та використання особистої інформації користувачів.
- Заходи безпеки для транзакцій: Застосування заходів безпеки під час обробки платежів, таких як зашифрованість і безпечні платіжні шлюзи, для запобігання крадіжкам інформації про платіжні дані.
- Актуалізація програмного забезпечення: Регулярне оновлення програмного забезпечення та застосунків для виправлення вразливостей та запобігання можливим атакам.
- Методи анонімізації: Використання методів анонімізації та псевдонімізації для обробки даних і зменшення ризику їх втрати чи зловживання.
- Моніторинг та захист від кібератак: Постійний моніторинг систем безпеки та застосування заходів для виявлення та запобігання потенційним кібератакам

Висновок

В цьому розділі ми роздивилися проблеми та варіанти вирішення цих проблем. Однак існуючий стан таких сайтів не завжди ідеальний. Часто зустрічаються проблеми з оновленням інформації, не достатньою точністю розкладів, а також нестабільною роботою інтерфейсу. Ці проблеми можуть

призвести до незручностей для користувачів і негативно вплинути на їх досвід використання сервісу.

РОЗДІЛ 2. ВИМОГИ ДО ПРОГРАМНОГО ЗАСТОСУНКУ

В даному розділі є завдання вивчення та розгляду ключових аспектів мінімального програмного застосунку для сайту-довідника розкладів політів літаків. Я планую зробити функціональності, що є основою для забезпечення базових, але важливих можливостей для користувачів у пошуку інформації про рейси.

Програмний застосунок для такого сайту виступає як важливий інструмент, що спрощує пошук рейсів та надає користувачам основну інформацію для планування подорожей. Мета цього проекту - розглянути та проаналізувати ключові функції, які є невід'ємною частиною цього програмного забезпечення, спираючись на потреби сучасних мандрівників та користувачів авіаперевезень.

Кафедра ІІЗ

НАУ 21.0410 ІІЗ

<i>Розроб.</i>	Свиридовський М.С.			Сайт довідник розкладів польотів літаків	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
<i>Керівник</i>	Кучеренко В.М.					17	43
					ІІ-501Бз		

2.1. Функціонал застосунку

В цьому пункті я опишу функціонал, який я буду реалізовувати для нашого додатку. Цей функціонал відкриє можливість для користувача досить зручно виконувати дії для пошуку розкладів польотів літаків.

2.1.1. Пошук рейсів

Функціонал пошуку рейсів на сайті розроблений для максимально зручного та ефективного пошуку оптимальних варіантів для подорожі. Ви можете вказати точні параметри, такі як відправний та прибутковий пункт (аеропорт або місто), дату вильоту та прильоту, кількість пасажирів та класи обслуговування, щоб отримати список рейсів, що відповідають вашим вимогам.

Гнучкий пошук дає можливість враховувати декілька дат, зокрема період +/- кілька днів від обраної дати, для пошуку найкращих варіантів. Фільтрація результатів дозволяє точно обрати рейси за такими критеріями, як ціна, час вильоту, кількість пересадок, авіакомпанія та інші параметри.

Наш сервіс показує деталізовану інформацію про кожен рейс: час вильоту та прильоту, тривалість подорожі, номер рейсу, авіакомпанію, тип літака та наявність вільних місць. Крім того, ви можете порівняти кілька рейсів одночасно, вибравши їх для докладного аналізу.

Додаткова інформація про кожен рейс, включаючи дані про пересадки, послуги на борту, правила багажу та розміщення місць, допомагає зробити обдуманий вибір для вашої майбутньої подорожі. Наш функціонал пошуку рейсів робить пошук та планування подорожей максимально зручним та інформативним для наших користувачів.

2.1.2. Відображення розкладів

На сайті, при перегляді кожного рейсу, ви отримуєте повний доступ до інформації, що стосується цього конкретного польоту. Ця інформація включає в себе точний час вильоту та прильоту, вказання дати, тривалість подорожі,

номер рейсу, авіакомпанію, тип літака, а також інформацію про наявність вільних місць на цьому рейсі.

Одна з ключових особливостей - це можливість фільтрації рейсів за різними критеріями. Ви можете обирати рейси в залежності від ціни, часу вильоту, кількості пересадок та інших параметрів, що дозволяє знайти оптимальний варіант для вашої подорожі. Інформація на сторінках рейсів оновлюється в режимі реального часу, щоб ви завжди мали доступ до актуальної інформації. Це означає, що будь-які зміни, пов'язані з розкладом чи пересадками, будуть відображені відразу. Крім базових даних про рейс, детальні сторінки також містять додаткову інформацію, таку як подробиці про можливі пересадки, типи послуг на борту, вимоги до багажу та розміщення місць у літаку. Це дає вам можливість зробити осмислений вибір для вашої майбутньої подорожі, маючи повний обсяг інформації під рукою.

2.1.3. Інформація про статус рейсу

Цей функціонал нашого сайту надає можливість користувачам в реальному часі отримувати актуальну інформацію про конкретний політ. Користувачі мають змогу вводити номер рейсу або вибирати його зі списку, щоб отримати оновлену інформацію про його статус. Ця інформація включає в себе дані про час вильоту та прильоту, можливі затримки, оновлення щодо розкладу, перенесення чи скасування польоту. Крім того, функція "Перегляд статусу рейсу" є важливою для тих, хто очікує пасажирів. Вона надає можливість стежити за рухом польоту, оновлюючи дані про прибуття і відправлення, щоб вчасно прибути на аеропорт або зробити інші відповідні заходи в залежності від статусу польоту. Ця функція спрощує процес відстеження польотів, надаючи точні та своєчасні дані про статус рейсів, що допомагає пасажирам і їхнім близьким забезпечити комфорт та безпеку під час подорожей.

2.1.4. Фільтрації та сортування

Функція фільтрації на сайті надає вам можливість точно налаштувати параметри пошуку рейсів. Ви можете застосовувати фільтри за часом вильоту та прильоту, тривалістю перельоту, кількістю пересадок, авіакомпанією, типом літака та іншими критеріями, щоб обрати лише ті варіанти, які відповідають вашим потребам. Крім фільтрації, функція сортування дозволяє вам організувати результати пошуку відповідно до ваших уподобань. Ви можете сортувати рейси за ціною, часом вильоту, часом прильоту або кількістю пересадок, щоб швидко знайти оптимальний варіант для вашої подорожі.

Ці дві функції працюють разом, щоб допомогти користувачу знайти найбільш підходящі рейси за вашими критеріями безпосередньо та ефективно. Вони роблять процес пошуку рейсів зручним та швидким, дозволяючи вам сконцентруватися на виборі найкращого варіанту для майбутньої подорожі.

2.2. Інструменти для реалізації застосунку

В цьому проекті я планую використовувати Vue 3 та Quasar. На сьогодні створення додатку з цими інструментами є сучасним рішенням. Vue 3, з його новими можливостями та швидкодією, разом із Quasar, надає гнучкість та потужність для створення інноваційних інтерфейсів. Цей проект використовуватиме найсучасніші технології Vue 3 для побудови веб-інтерфейсу, що буде відповідати вимогам сучасних користувачів. Quasar, зі своєю надійністю та розширюваністю, допоможе вам створити інтуїтивний та функціональний дизайн, оптимізований під різні платформи та пристрої.

Цей проект відкриє безліч можливостей для вас як розробника, дозволяючи вам ефективно використовувати переваги Vue 3 та Quasar для швидкої реалізації ідей та створення вражаючих, високофункціональних додатків. Використання цих технологій підкреслить вашу здатність реалізовувати проекти на передовому рівні розвитку веб-розробки.

Провідність Vue 3 та екосистеми Quasar відкривають перед розробниками безмежні можливості у створенні веб-додатків, що поєднують сучасність, швидкість та надійність. Vue 3, який відзначається своєю прогресивністю та збільшеною продуктивністю, спільно з розширювальним та універсальним фреймворком Quasar, створює оптимальне середовище для реалізації проєктів будь-якої складності.

Цей проєкт буде базуватися на основах Vue 3 та Quasar, щоб створити веб-додаток, який відповідає найсучаснішим стандартам розробки. Ви матимете можливість використовувати нові можливості Vue 3, такі як Composition API, для ефективно організації коду та покращення його читабельності та розширюваності.

Quasar додасть значну вартість, забезпечуючи не лише інтерфейс високої якості, а й можливість розгортання на різних платформах з одним кодом. Завдяки вбудованій підтримці різних типів пристроїв, ви матимете змогу розширити аудиторію та забезпечити відмінний досвід користувача на будь-яких пристроях. Для прикладу на рисунку 2 можна побачити додаток, який був створений за допомогою цього стеку.

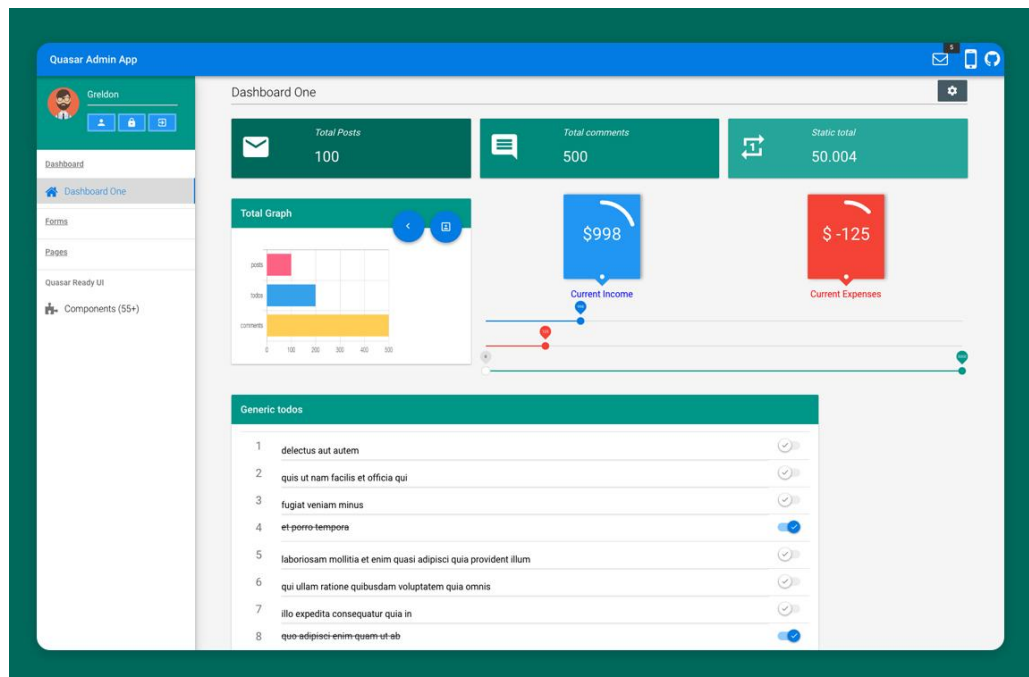


Рис. 2.1. Додаток на Vue 3 та Quasar

Для того щоб побудувати сайт потрібен backend і frontend, наочну схему можна побачити на рисунку 3.

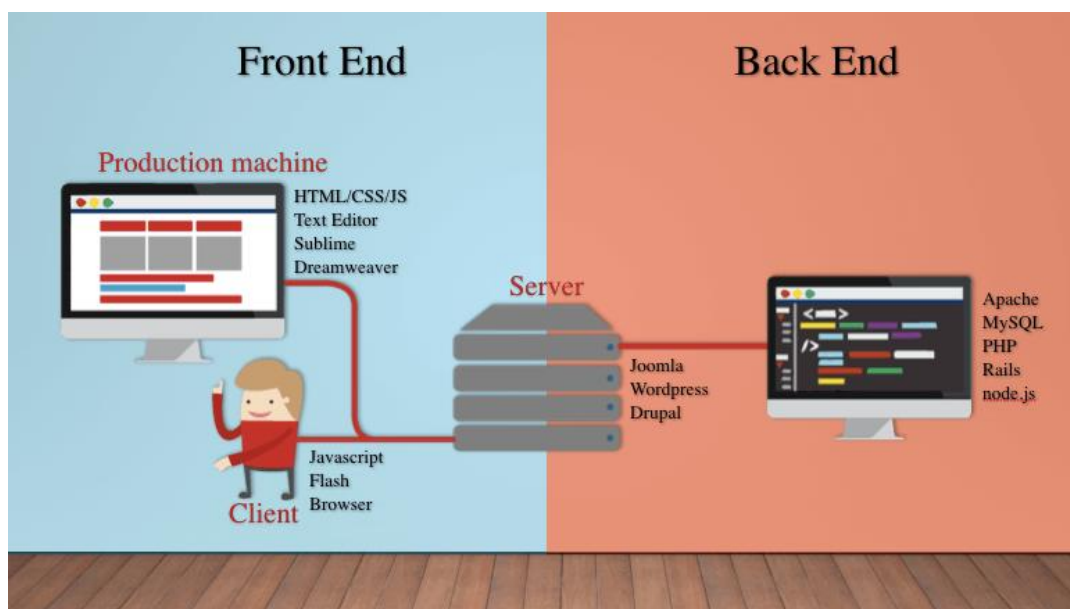


Рис. 2.2. Додаток на Vue 3 та Quasar

2.2.1. Frontend

Frontend (або клієнтська частина) веб-розробки - це сукупність технологій та інструментів, що відповідають за те, як користувачі бачать та взаємодіють з веб-сайтами та додатками. Це те, що ви бачите у своєму веб-браузері та як ви взаємодієте з цим контентом. Основні компоненти frontend включають у себе HTML, CSS та JavaScript. HTML визначає структуру контенту на сторінці, CSS відповідає за стиль та вигляд цього контенту, а JavaScript надає можливості для додавання інтерактивності та функціональності. За допомогою цих інструментів розробники створюють естетичні, функціональні та динамічні інтерфейси для користувачів. Frontend також включає в себе використання фреймворків та бібліотек, таких як React, Vue.js, або Angular, які полегшують та прискорюють процес розробки, а також дозволяють створювати більш складні та потужні додатки.

Основна мета frontend розробки - забезпечити користувачам зручну та приємну взаємодію з веб-сайтами та додатками, зробити їх доступними,

естетичними та легкими у використанні. На рисунку 3 можна побачити приклад, як працює frontend.

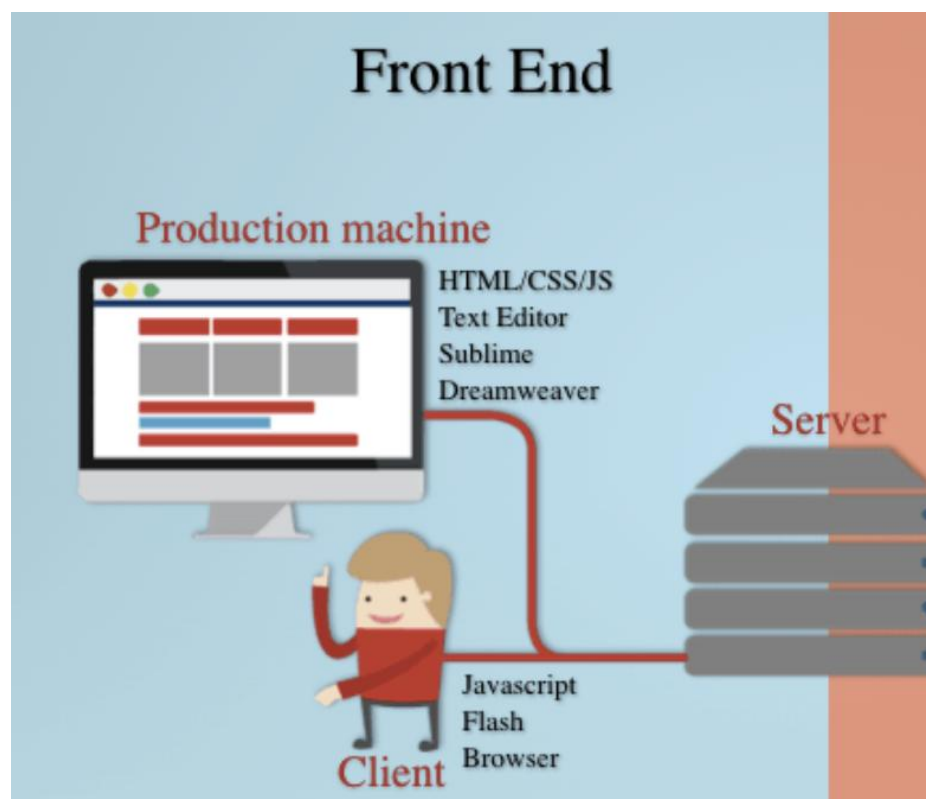


Рис. 2.3. Приклад роботи frontend

2.2.1.1. Vue

Vue.js - це прогресивний JavaScript фреймворк для створення веб-інтерфейсів та односторінкових додатків. Його основною перевагою є легкість в освоєнні та використанні, а також гнучкість та швидкодія. Vue надає інструменти для розробки користувацьких інтерфейсів, де ви можете організувати вашу програму за допомогою компонентів, кожен з яких має свій власний функціонал та структуру.

Цей фреймворк спрощує розробку веб-додатків шляхом забезпечення чіткої структури, розподілу відповідальності між компонентами та можливості зручної інтеракції між ними. Vue також забезпечує двостороннє зв'язування даних, що означає, що зміни в даних автоматично відображаються на користувацькому інтерфейсі та навпаки. Однією з ключових особливостей Vue

є його простота. Це дозволяє швидше створювати прототипи, а потім розширювати їх до більш складних додатків. Також Vue надає можливості для реактивного програмування, анімацій, обробки подій та інших функціональних можливостей для покращення користувацького досвіду.

Загалом, Vue.js - це потужний, легкий у використанні та швидкий фреймворк, який знаходить широке застосування для створення сучасних та ефективних веб-додатків.

В цьому проекті я планую використовувати Vue 3, тому що в нього більше швидкість, ефективну систему Composition API для зручної організації коду, менший обсяг, що сприяє швидшому завантаженню та покращеному реактивному програмуванню, що полегшує роботу з даними та їх відстеженням. Ці переваги роблять Vue 3 привабливим вибором для розробки цього проекту.

2.2.1.2. Quasar

Quasar Framework - це рішення для розробки веб-додатків, яке об'єднує в собі широкий набір готових компонентів та інструментів для зручної та швидкої розробки. Його велика колекція компонентів дозволяє створювати різноманітні елементи інтерфейсу, від кнопок та форм до складних таблиць і модальних вікон, спрощуючи роботу для розробників. Основна перевага Quasar - це можливість розгортання на різних платформах, включаючи веб, Android, iOS та Electron, що робить його універсальним інструментом для створення додатків, а також спрощує процес підтримки та розвитку. Завдяки гнучкості налаштування тем, стилів та можливостям кастомізації, Quasar стає потужним рішенням для швидкої і якісної розробки різноманітних проектів.

Основні особливості Quasar:

- Кросплатформений підхід: Quasar дозволяє розробляти одну кодову базу та згорнути додатки на різних платформах: веб, Android, iOS, Electron і т.д., що дозволяє заощаджувати час та ресурси.

- Компоненти та макети: Quasar постачається з широким набором готових компонентів, які включають в себе кнопки, форми, таблиці, модальні вікна та багато інших, що полегшує розробку та забезпечує високий рівень інтерфейсу.
- Гнучкість: Інструменти Quasar дозволяють налаштовувати стилі, теми, адаптувати компоненти та макети під власні потреби проекту.
- Спільнота та підтримка: Quasar має активну спільноту розробників, що дозволяє отримувати підтримку, допомогу та знайти рішення для різноманітних завдань.
- Інтеграція з Vue.js: Quasar базується на Vue.js, що робить його ідеальним вибором для розробки на основі цього популярного фреймворку.

2.2.2. Backend

Backend - це частина програмного забезпечення, яка відповідає за обробку даних, логіку бізнес-процесів та взаємодію з базою даних. У веб-розробці термін "backend" використовується для позначення серверної сторони веб-додатка, тобто тієї частини, яка виконується на сервері та не є доступною для користувача напряму через веб-браузер.

Backend включає в себе різноманітні елементи, такі як сервери, бази даних, застосунки для обробки бізнес-логіки, обробники запитів і так далі. Основна задача backend - це забезпечення правильної роботи веб-додатка та здійснення взаємодії з фронтендом (задній інтерфейс для користувача, який він бачить в браузері чи іншому клієнтському додатку).

Фронтенд (з переднього кінця) і бекенд (заднього кінця) співпрацюють разом, утворюючи повноцінний веб-додаток. Фронтенд відповідає за те, як інтерфейс виглядає та як користувач взаємодіє з додатком, тоді як бекенд забезпечує необхідну функціональність та обробку даних. На рисунку 4 можна побачити приклад, як працює backend.

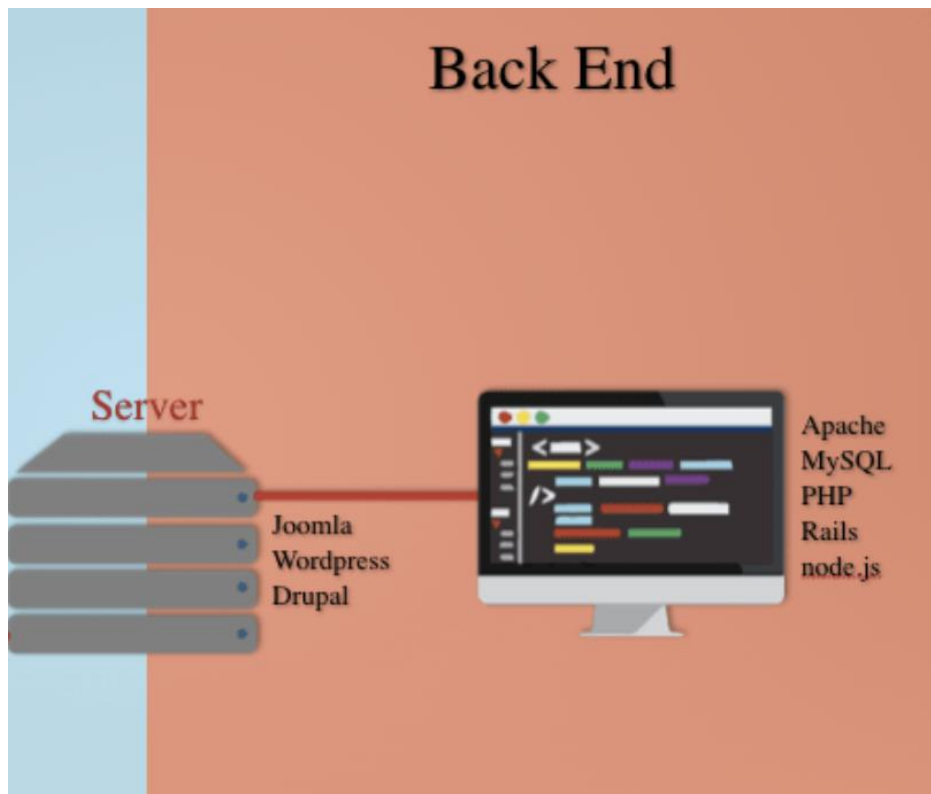


Рис. 2.4. Приклад роботи backend

2.2.2.1. Формат json

JSON (JavaScript Object Notation) - це легкий, текстовий формат обміну даними, який забезпечує простий та зрозумілий спосіб представлення структурованих даних. В основі JSON лежать об'єкти, які представляють собою набір пар "ключ-значення". Ці об'єкти можуть вкладатися один в одного, формуючи складні структури.

Використовуючи JSON, інформацію можна представити у вигляді текстового рядка, що легко читається як люди, так і програми. Крім того, JSON є незалежним від конкретної мови програмування, тобто дані у форматі JSON можуть бути оброблені різними мовами.

У веб-розробці JSON часто використовується для передачі даних між веб-сервером та клієнтською частиною додатків. Наприклад, відповідь від сервера часто буде у форматі JSON, щоб передати структуровані дані клієнту для подальшого використання чи відображення на веб-сторінці.

Висновок

В цьому розділі я роздивився функціональні вимоги додатку та інструменти для реалізації функціоналу. Ми оглянули ключові функції, які має надавати додаток, а також визначили необхідні інструменти і технології для їхньої реалізації. Це допомагає зрозуміти обсяг і складність проекту, а також визначити кращі підходи до розробки та вибрати інструменти, які найбільш підходять для вирішення поставлених завдань. Розглянуті інструменти vue 3 та quasar, які є популярними серед розробників програмного забезпечення і забезпечують широкий спектр можливостей для зручного та ефективного розвитку додатку.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ

Для реалізації додатку я буду використовувати файл json в якому лежать дані згідно з створеною схемою бекенда. Лицьову частину сайту буде виконана за допомогою quasar.

3.1. Створення схеми даних

Схема даних для сайту довідника розкладів польотів літаків включає кілька ключових таблиць. Перша таблиця - "Airlines", містить інформацію про авіакомпанії, зокрема, унікальний ідентифікатор, назву та країну. Друга таблиця - "Airports", містить дані про аеропорти, включаючи їхній унікальний ідентифікатор, назву, місто, країну та код. Третя таблиця - "Flights", містить інформацію про конкретні польоти, включаючи ідентифікатор польоту, ідентифікатор авіакомпанії, ідентифікатори відправного та прибуткового аеропортів, час вильоту та прибуття, тривалість польоту та ціну квитка. Четверта таблиця Users потрібна буде для зберігання користувачів.

Кафедра ІІЗ

НАУ 21.0410 ІІЗ

<i>Розроб.</i>	Свиридовський М.С.			Сайт довідник розкладів польотів літаків	<i>Лім.</i>			<i>Лист</i>	<i>Листів</i>
<i>Керівник</i>	Кучеренко В.М.							28	43
					ПІ-501Бз				

Створення схеми даних я буду робити в додатку dbdesigner, який дозволяє зручно створити схему для додатку.

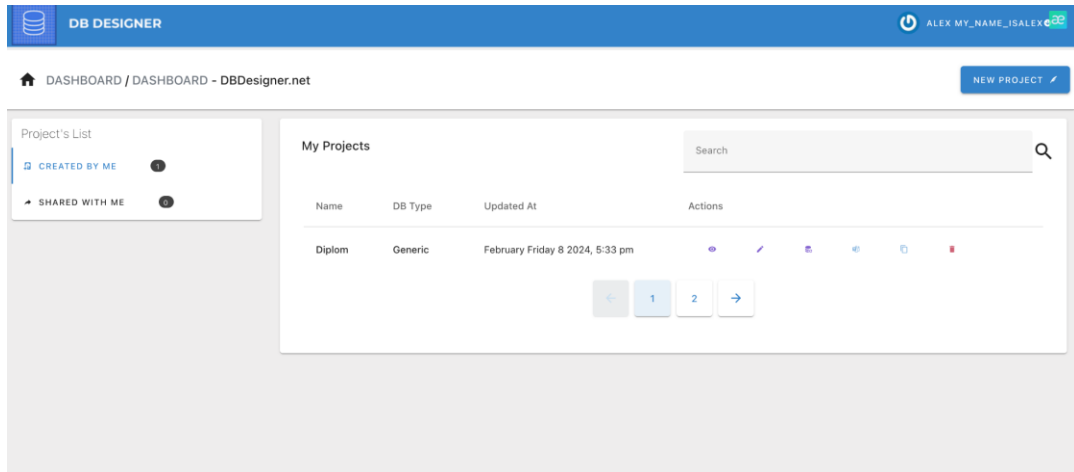


Рис. 3.1. dbdesigner

Для початку потрібно описати таблиці, які будуть в проєкті:

- Таблиця "Airlines" (Авіакомпанії):
 - airline id (Ідентифікатор авіакомпанії, унікальний ключ)
 - name (Назва авіакомпанії)
 - country (Країна, в якій базується авіакомпанія)
- Таблиця "Airports" (Аеропорти):
 - airport id (Ідентифікатор аеропорту, унікальний ключ)
 - name (Назва аеропорту)
 - city (Місто, де розташований аеропорт)
 - country (Країна, де розташований аеропорт)
 - code (Код аеропорту)
- Таблиця "Flights" (Польоти):
 - flight id (Ідентифікатор польоту, унікальний ключ)
 - airline_id (Зовнішній ключ до таблиці "Airlines")
 - departure airport id (Зовнішній ключ до таблиці "Airports")
 - arrival airport id (Зовнішній ключ до таблиці "Airports")
 - departure_time (Час вильоту)
 - arrival time (Час прибуття)

- duration (Тривалість польоту)
- price (Ціна квитка)
- Таблиця "Users" (Користувачі):
 - user_id (Ідентифікатор користувача, унікальний ключ)
 - username (Ім'я користувача)
 - email (Електронна пошта)
 - password_hash (Хеш пароля)

Далі потрібно буде створити схему у додатку dbdesigner:

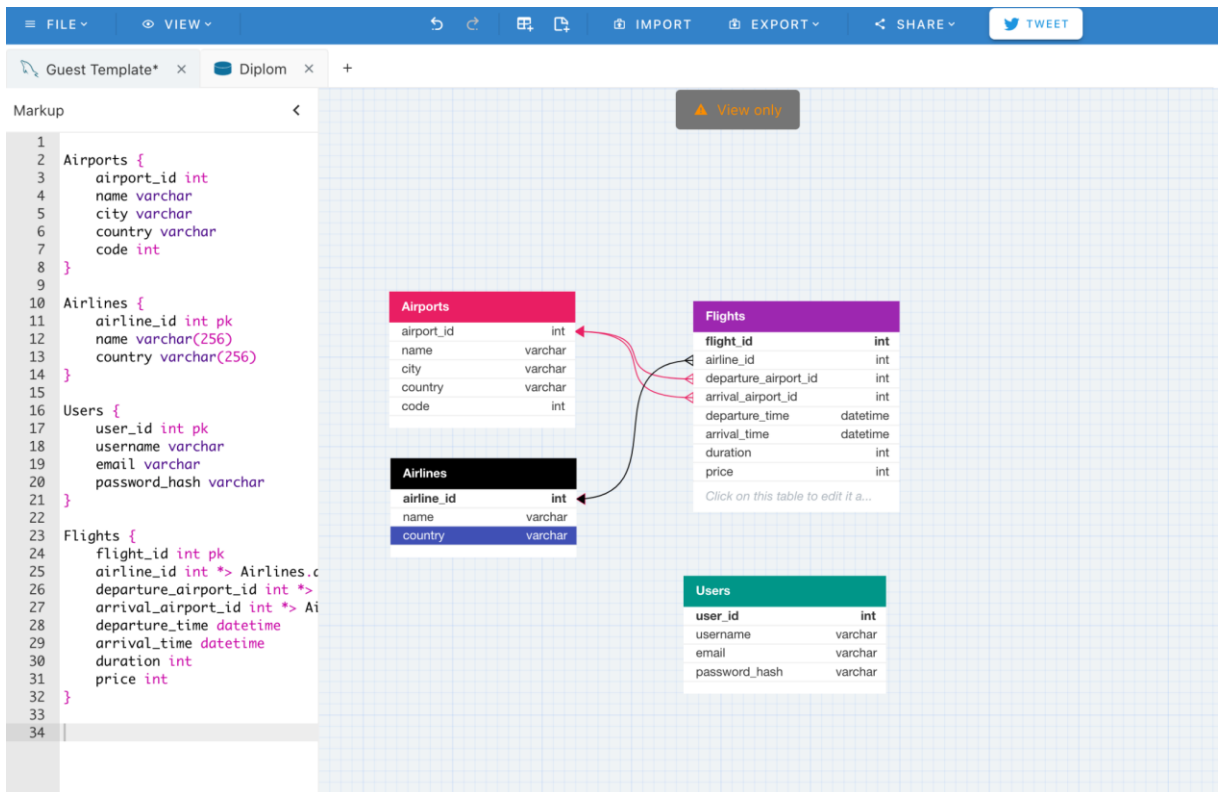


Рис. 3.2. Схема даних додатку

3.2.1. Створення даних до проекту

Згідно схеми якої ми побудували потрібно створити дані:

```

"airlines": [
  { "airline_id": 1, "name": "Emirates", "country": "United Arab Emirates" },
  { "airline_id": 2, "name": "Singapore Airlines", "country": "Singapore" },
  { "airline_id": 3, "name": "Qatar Airways", "country": "Qatar" },
  { "airline_id": 4, "name": "British Airways", "country": "United Kingdom" },
  { "airline_id": 5, "name": "Lufthansa", "country": "Germany" },
  { "airline_id": 6, "name": "Delta Air Lines", "country": "United States" },
  { "airline_id": 7, "name": "American Airlines", "country": "United States" },
  { "airline_id": 8, "name": "Air France", "country": "France" },
  { "airline_id": 9, "name": "Cathay Pacific", "country": "Hong Kong" }
],

```

Рис. 3.3. Дані airlines

```

"airports": [
  { "airport_id": 1, "name": "Dubai International Airport", "city": "Dubai", "country": "United Arab Emirates", "code": "DXB" },
  { "airport_id": 2, "name": "Changi Airport", "city": "Singapore", "country": "Singapore", "code": "SIN" },
  { "airport_id": 3, "name": "Hamad International Airport", "city": "Doha", "country": "Qatar", "code": "DOH" },
  { "airport_id": 4, "name": "Heathrow Airport", "city": "London", "country": "United Kingdom", "code": "LHR" },
  { "airport_id": 5, "name": "Frankfurt Airport", "city": "Frankfurt", "country": "Germany", "code": "FRA" },
  { "airport_id": 6, "name": "Los Angeles International Airport", "city": "Los Angeles", "country": "United States", "code": "LAX" },
  { "airport_id": 7, "name": "John F. Kennedy International Airport", "city": "New York City", "country": "United States", "code": "JFK" },
  { "airport_id": 8, "name": "Charles de Gaulle Airport", "city": "Paris", "country": "France", "code": "CDG" },
  { "airport_id": 9, "name": "Hong Kong International Airport", "city": "Hong Kong", "country": "Hong Kong", "code": "HKG" }
],

```

Рис. 3.4. Дані airports

```

"flights": [
  { "flight_id": 1, "airline_id": 1, "departure_airport_id": 1, "arrival_airport_id": 3, "departure_time": "2024-02-09T08:00:00Z", "arrival_time": "2024-02-09T10:00:00Z" },
  { "flight_id": 2, "airline_id": 2, "departure_airport_id": 2, "arrival_airport_id": 1, "departure_time": "2024-02-10T10:00:00Z", "arrival_time": "2024-02-10T12:00:00Z" },
  { "flight_id": 3, "airline_id": 3, "departure_airport_id": 3, "arrival_airport_id": 2, "departure_time": "2024-02-11T12:00:00Z", "arrival_time": "2024-02-11T14:00:00Z" },
  { "flight_id": 4, "airline_id": 1, "departure_airport_id": 4, "arrival_airport_id": 5, "departure_time": "2024-02-12T14:00:00Z", "arrival_time": "2024-02-12T16:00:00Z" },
  { "flight_id": 5, "airline_id": 2, "departure_airport_id": 5, "arrival_airport_id": 6, "departure_time": "2024-02-13T09:00:00Z", "arrival_time": "2024-02-13T11:00:00Z" },
  { "flight_id": 6, "airline_id": 3, "departure_airport_id": 6, "arrival_airport_id": 4, "departure_time": "2024-02-14T11:00:00Z", "arrival_time": "2024-02-14T13:00:00Z" },
  { "flight_id": 7, "airline_id": 4, "departure_airport_id": 7, "arrival_airport_id": 8, "departure_time": "2024-02-15T16:00:00Z", "arrival_time": "2024-02-15T18:00:00Z" },
  { "flight_id": 8, "airline_id": 5, "departure_airport_id": 8, "arrival_airport_id": 9, "departure_time": "2024-02-16T11:30:00Z", "arrival_time": "2024-02-16T13:30:00Z" },
  { "flight_id": 9, "airline_id": 6, "departure_airport_id": 9, "arrival_airport_id": 7, "departure_time": "2024-02-17T08:00:00Z", "arrival_time": "2024-02-17T10:00:00Z" }
],

```

Рис. 3.5. Дані flights

```

"users": [
  { "user_id": 1, "username": "john_doe", "email": "john.doe@example.com", "password_hash": "hashed_password_1" },
  { "user_id": 2, "username": "jane_smith", "email": "jane.smith@example.com", "password_hash": "hashed_password_2" },
  { "user_id": 3, "username": "alex_wong", "email": "alex.wong@example.com", "password_hash": "hashed_password_3" },
  { "user_id": 4, "username": "emily_jones", "email": "emily.jones@example.com", "password_hash": "hashed_password_4" },
  { "user_id": 5, "username": "michael_brown", "email": "michael.brown@example.com", "password_hash": "hashed_password_5" },
  { "user_id": 6, "username": "sarah_wilson", "email": "sarah.wilson@example.com", "password_hash": "hashed_password_6" }
],

```

Рис. 3.6. Дані users

3.3. Створення лицьової частини сайту

Для початку потрібно створити проект для цього треба створити папку та відкрити її в редакторі коду vs code, після цього прописати команду **npm init quasar**, за допомогою менеджера пакетів створити проект.

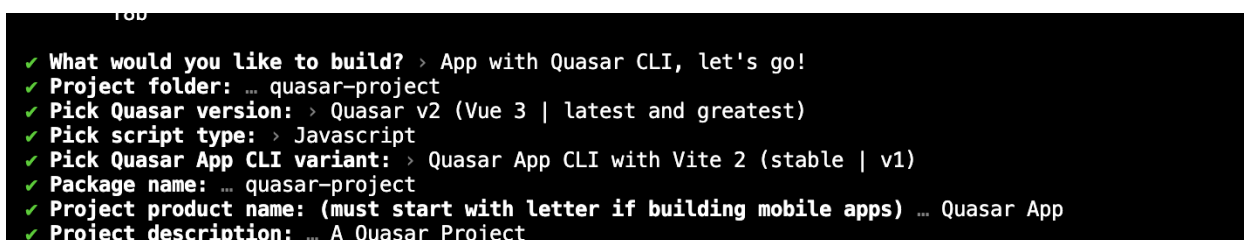
Visual Studio Code (VS Code) - є безкоштовним і відкритим редактором тексту, розробленим компанією Microsoft. Він призначений для створення програмного забезпечення та роботи з різноманітними мовами програмування, включаючи JavaScript, Python, C++, Java, HTML, CSS та багато інших. VS Code

має багато функцій, таких як підсвічування синтаксису, автодоповнення, вбудований дебагер, система контролю версій і багато інших, що робить його популярним серед розробників програмного забезпечення. Він є платформо незалежним і може працювати на різних операційних системах, таких як Windows, macOS та Linux.

npm (Node Package Manager) - це стандартний пакетний менеджер для середовища виконання JavaScript Node.js. Він використовується для управління пакетами, які використовуються у проектах JavaScript. npm дозволяє розробникам швидко і легко встановлювати, оновлювати та видаляти залежності проекту, такі як бібліотеки, фреймворки або інші зовнішні модулі, а також розповсюджувати свої власні пакети.

За допомогою npm ви можете встановлювати пакети з веб-репозиторію npm, який є найбільшим у світі репозиторієм програмних пакетів. Також ви можете використовувати npm для управління версіями пакетів, виконання скриптів та багато іншого. npm постачається з Node.js і автоматично встановлюється разом з ним, що робить його широко використовуваним в середовищі розробки JavaScript.

Після того, як я прописав цю команду потрібно вибрати налаштування проекту:



```
✓ What would you like to build? > App with Quasar CLI, let's go!  
✓ Project folder: ... quasar-project  
✓ Pick Quasar version: > Quasar v2 (Vue 3 | latest and greatest)  
✓ Pick script type: > Javascript  
✓ Pick Quasar App CLI variant: > Quasar App CLI with Vite 2 (stable | v1)  
✓ Package name: ... quasar-project  
✓ Project product name: (must start with letter if building mobile apps) ... Quasar App  
✓ Project description: ... A Quasar Project
```

Рис. 3.7. Налаштування проекту

Після цього в нас буде така структура проекту:



Рис. 3.8. Структура проекту

Далі ми маємо можливість запустити проект за допомогою команди `npm run dev`:

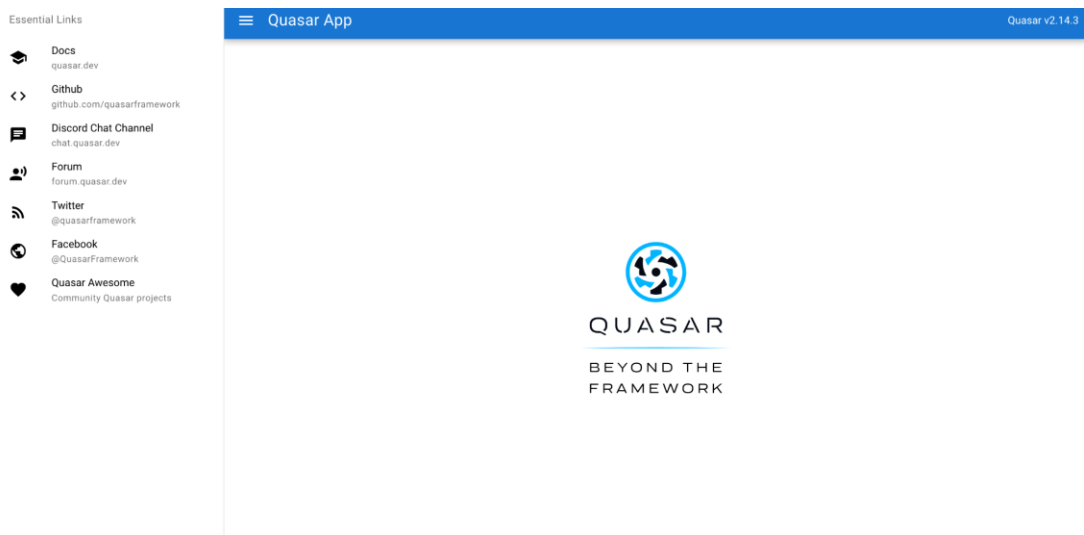


Рис. 3.9. Запущений проект

3.3.1. Створення структурної схеми веб-сайту

Структурна схема веб-сайту - це візуальне зображення основних сторінок та їх взаємозв'язків на сайті. Вона допомагає зрозуміти, як користувачі будуть

навігувати по сайту та які функції доступні на кожній сторінці. Зазвичай це схема, яка включає в себе головні розділи, підрозділи, можливі дії користувачів та зв'язки між сторінками.

Далі потрібно побудувати цю схему для додатку, а на основі нього зробити роутинг сайту.

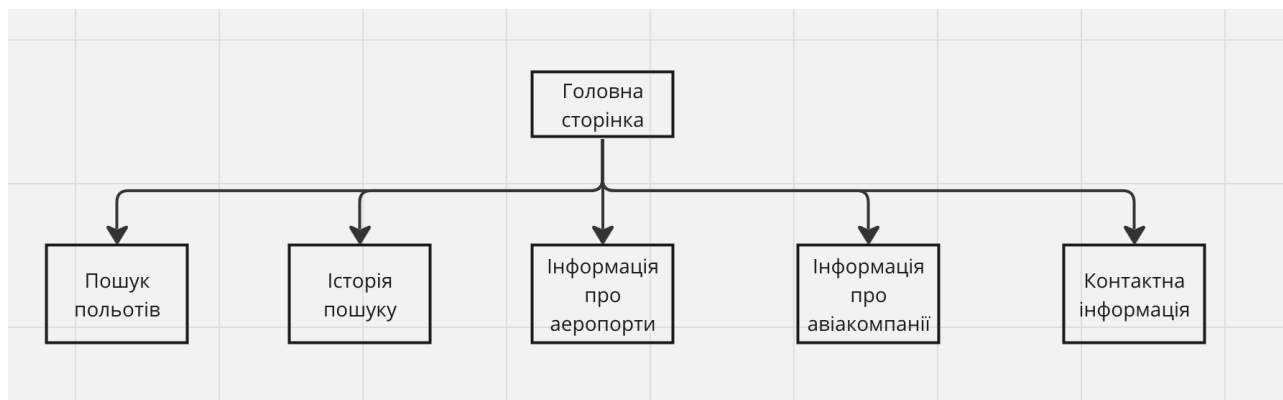


Рис. 3.10. Структурна схема веб-сайту

- Пошук польотів - користувачі матимуть можливість ввести критерії пошуку, такі як місце відправлення, дату вильоту, тощо. Форма може містити різні поля та варіанти вибору, такі як випадаючі списки для аеропортів або календар для вибору дати.
- Історія пошуку - Сторінка, на якій відображається історія польотів, дозволяється переходити на сторінку із заданими налаштуваннями по натисканню на історію.
- Інформація про аеропорти - Сторінка на якій буде інформація про аеропорти
- Інформація про авіакомпанії - Сторінка на якій буде інформація про авіакомпанії
- Контактна інформація - Сторінка для зберігання контактів

3.3.2. Реалізація сторінок

Для початку треба прописати роутинг для сторінок, після цього потрібно написати код для них.

```
const linksList = [
  {
    title: 'Польоти',
    caption: 'Пошук польотів',
    icon: '✈',
    link: '/flying'
  },
  {
    title: 'Історія',
    caption: 'Історія пошуку',
    icon: '📄',
    link: '/history'
  },
  {
    title: 'Аеропорти',
    caption: 'Інформація про аеропорти',
    icon: '✈',
    link: '/airport-info'
  },
  {
    title: 'Авіакомпанії',
    caption: 'Інформація про авіакомпанії',
    icon: '🏠',
    link: '/company-info'
  },
  {
    title: 'Контакти',
    caption: 'Контактна інформація',
    icon: '📞',
    link: '/contacts'
  }
]
```

Рис. 3.11. Роутинг

Після цього можемо починати реалізацію сторінок в проєкті, для початку почнемо з меню.

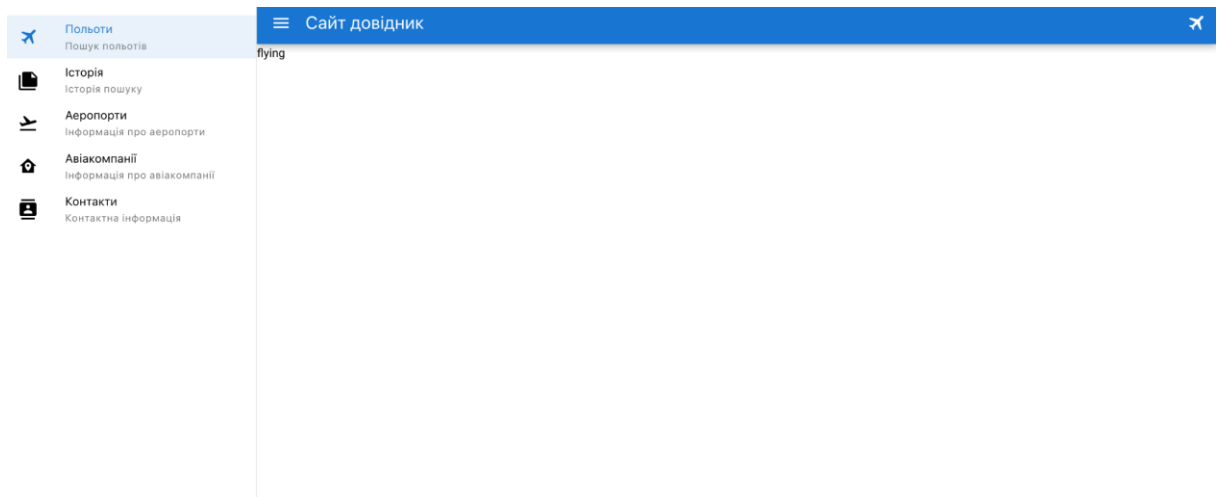


Рис. 3.12. Меню додатку

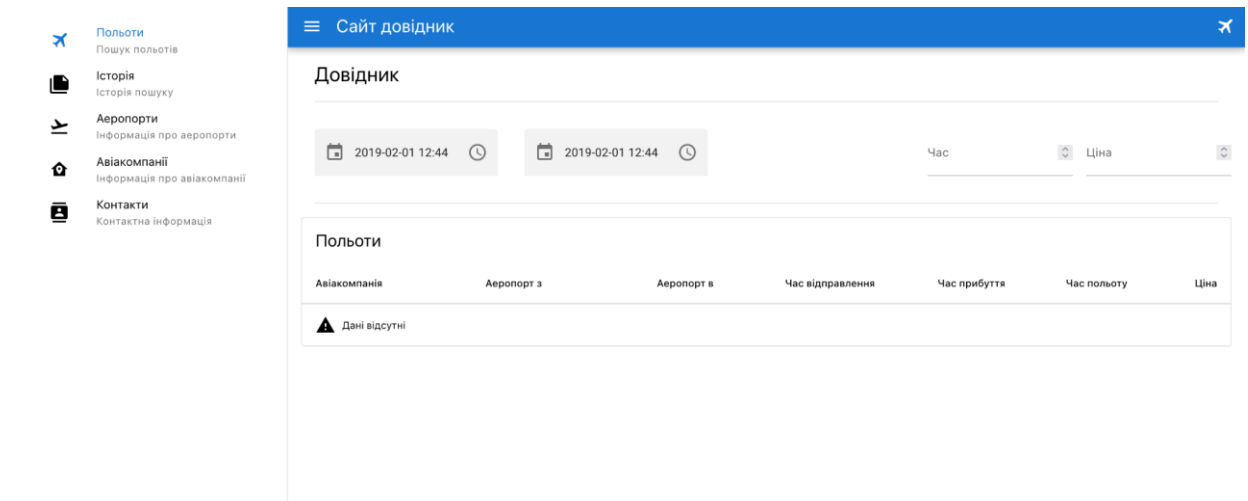


Рис. 3.13. Сторінка польоти

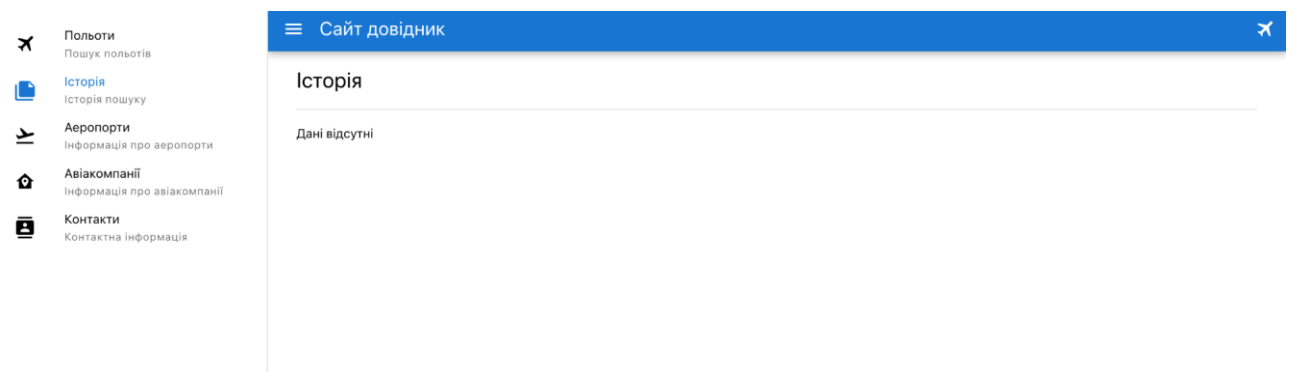


Рис. 3.14. Сторінка історія

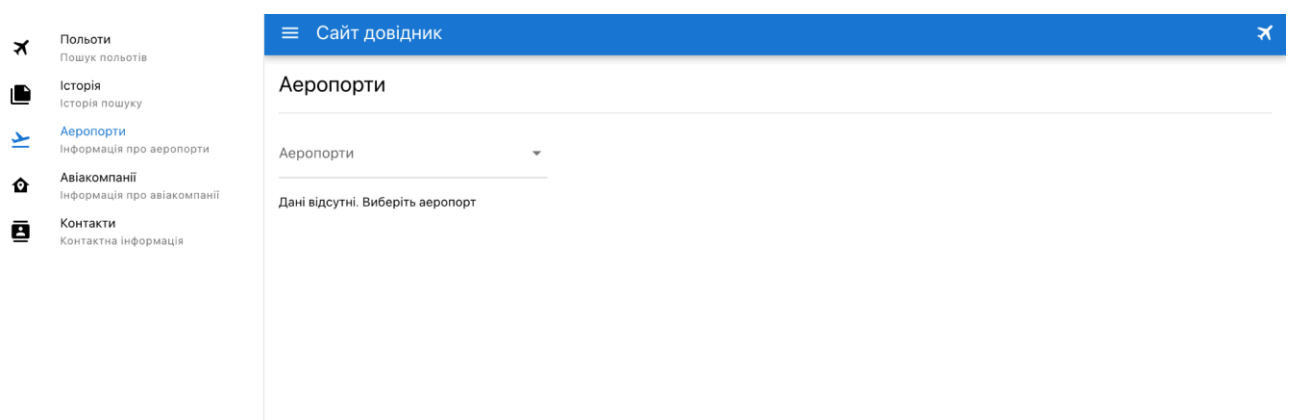


Рис. 3.15. Сторінка аеропорти

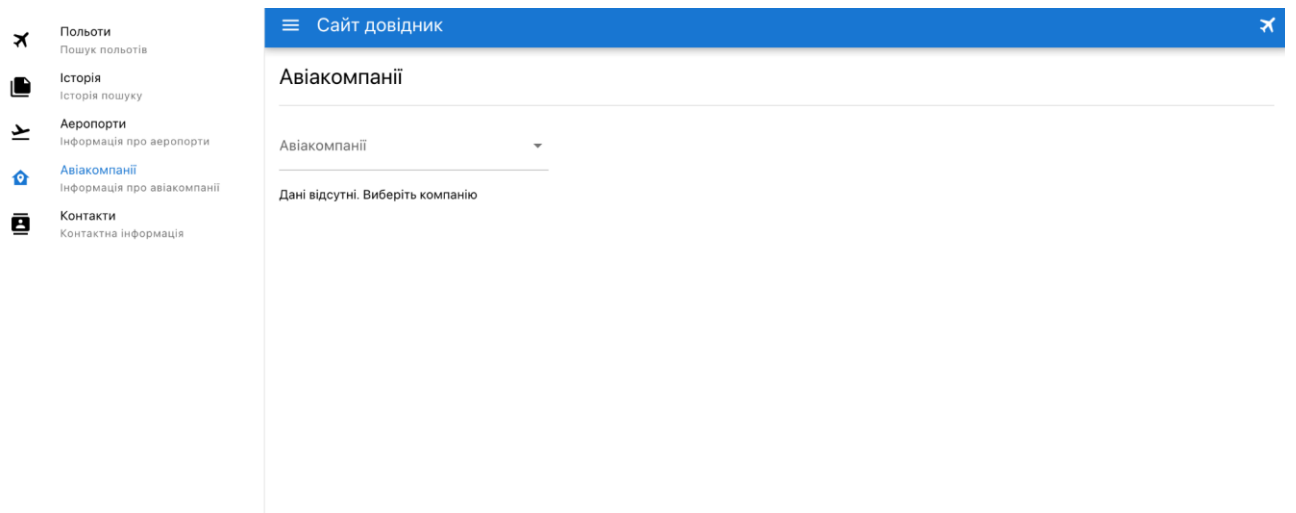


Рис. 3.16. Сторінка авіакомпанії

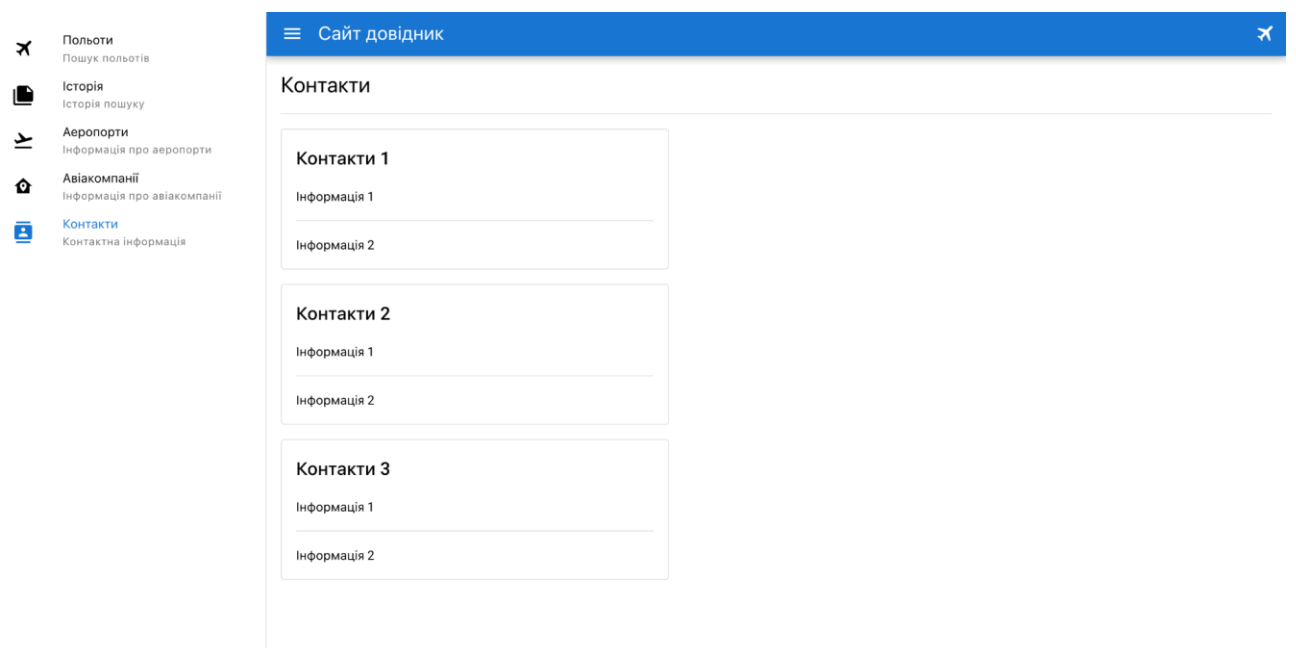


Рис. 3.17. Сторінка контакти

Висновок

В цьому розділі ми створили структуру даних для проекту та зробили самі дані, це дозволило нам мати архітектуру в проекті. Схема даних допомагає організувати дані у логічні групи та визначити взаємозв'язки між ними, що полегшує керування та роботу з даними.

Також була створена структурну схему для лицьової частини В структурній схемі веб-сайту важливо розглянути організацію та взаємозв'язки між його різними частинами. Описана структура надає чітке уявлення про те, як користувачі будуть навігувати по сайту та як вони можуть взаємодіяти з його контентом. Вона відображає основні розділи, сторінки та можливі дії користувачів. На сторінці "Пошук польотів", наприклад, користувачі матимуть зручну форму пошуку, яка дозволить їм швидко та ефективно знаходити польоти за різними критеріями. Після введення параметрів пошуку, користувачеві буде показано список польотів з детальною інформацією, що допоможе їм зробити вибір. Крім того, на цій сторінці можуть бути доступні фільтри та сортування, що зробить пошук ще більш зручним.

РОЗДІЛ 4. РЕЗУЛЬТАТИ РОБОТИ ПРОГРАМИ

4.1 Опис додатку

В даному проєкті ми реалізували цей додаток, далі ми розберемо функціонал додатку.

1. Пошук - Є можливість шукати дані за 4 критеріями дати(прибуття та відправка) та ціна і час. В даному приклади ми зробили пошук за ціною.

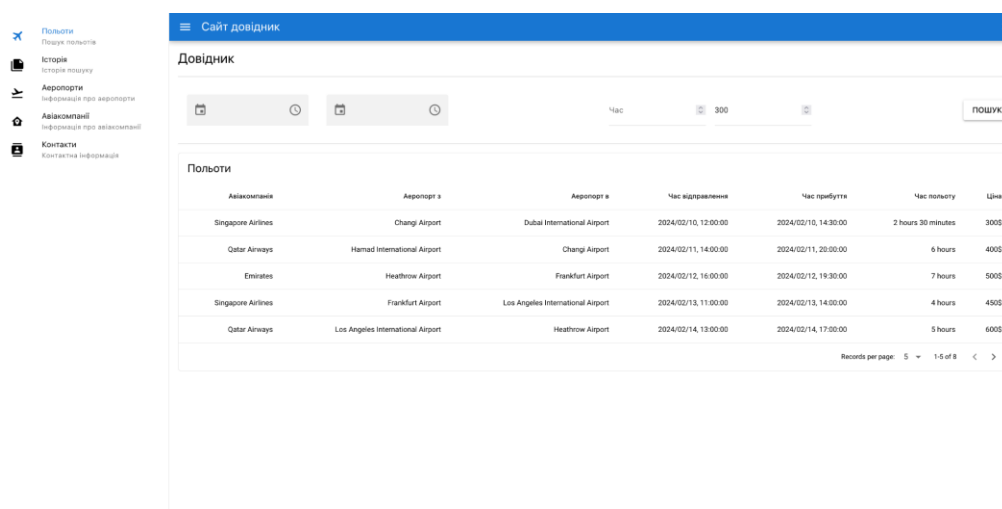


Рис. 4.1. Пошук

Кафедра ІІЗ

НАУ 21.0410 ІІЗ

<i>Розроб.</i>	Свиридовський М.С.			Сайт довідник розкладів польотів літаків	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
<i>Керівник</i>	Кучеренко В.М.					39	43
					ПІ-501 Бз		

2. Історія - Кожен раз, коли ми робимо пошук, ми маємо можливість подивитися в історію по яким критерієм ми шукали. Як бачимо на рисунку 4.2. Наш пошук додався в історію.

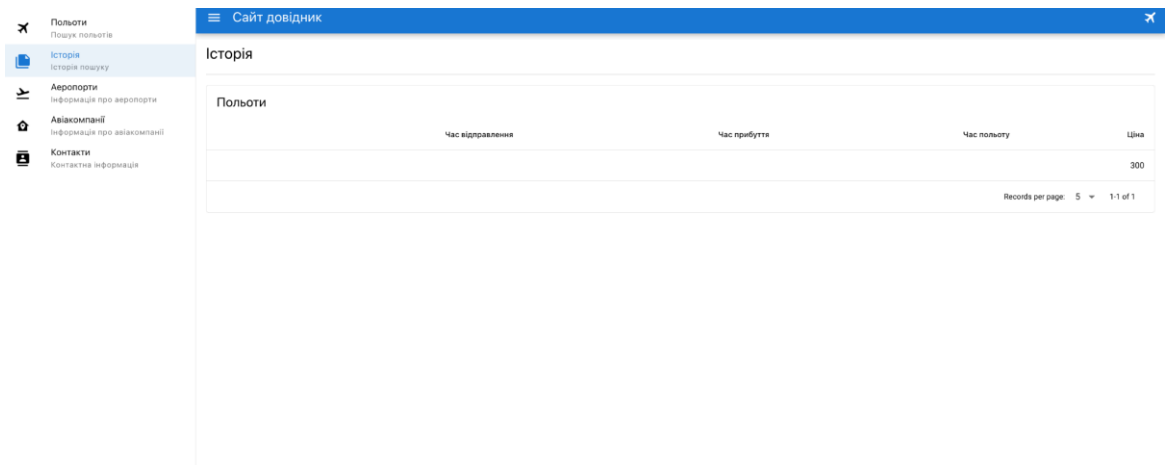


Рис. 4.2. Історія пошуку

3. Аеропорти - На цьому табі в нас є можливість подивитися всі можливі аеропорти додатку. Ми можемо дізнатися про них більш детально.

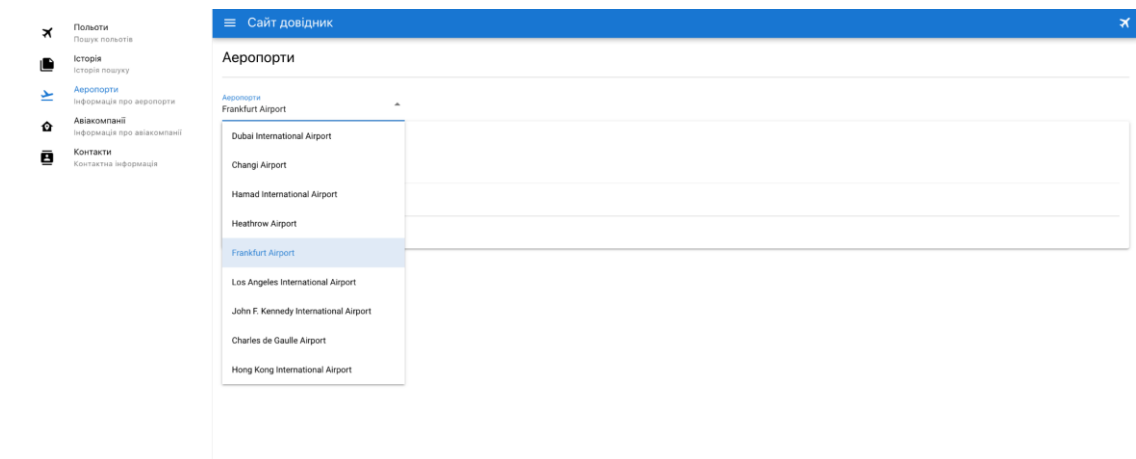


Рис. 4.3. Аеропорти

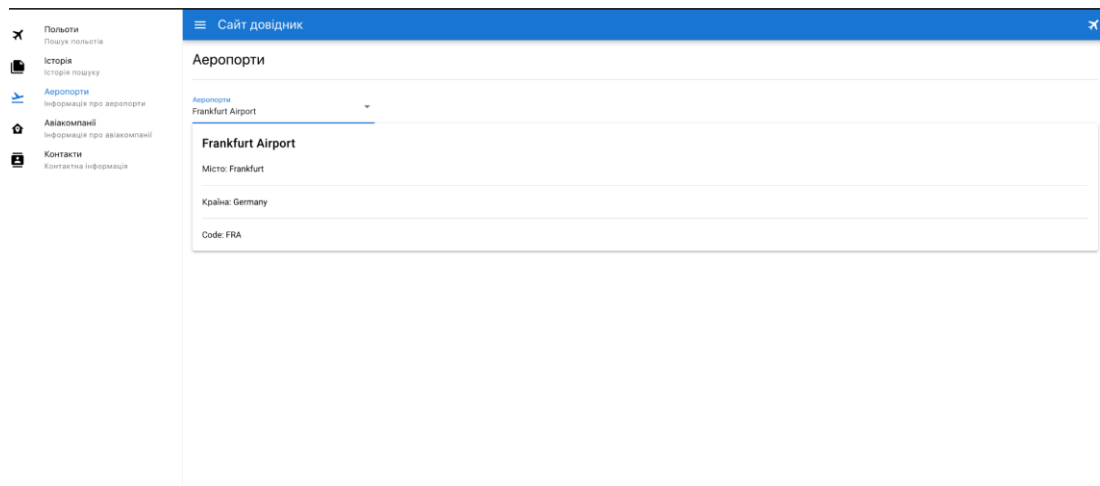


Рис. 4.4. Вибраний аеропорт

4. Авіакомпанії - Теж сама логіка як в табі аеропорти, але тільки для авіакомпаній.
5. Контакти - Це сторінка для додаткової інформації додатку, контакти які потрібні для технічної підтримки. Також є можливість додати туди банери або рекламну інформацію.

Висновок

Підсумковий висновок щодо цього додатку показує, що він є ефективним інструментом для отримання розкладів польотів та інформації про авіакомпанії та аеропорти. Використання сучасних технологій, таких як Vue 3 та Quasar, дозволяє створити зручний та динамічний інтерфейс, що сприяє комфортному користуванню.

ВИСНОВКИ

Цей проект є відмінним прикладом використання сучасних технологій для створення інтерактивних та функціональних веб-додатків. Використання Vue 3 надає високу швидкість та продуктивність, дозволяючи швидко реагувати на зміни та надавати зручний інтерфейс для користувачів. Quasar надає багато можливостей для швидкої розробки з використанням компонентної архітектури, що сприяє ефективному управлінню структурою проекту та забезпечує високу реактивність і динамічність веб-сторінок. Завдяки такій комбінації фреймворків і бібліотек, веб-додаток може бути швидко розгорнутий та легко підтримуваний.

Бекенд частина є JSON файл в якості бекенду спрощує процес розробки та розгортання, оскільки не потрібно налаштовувати та керувати базою даних. Такий підхід особливо корисний для невеликих проектів або демонстраційних версій, де не вимагається велика кількість даних або складні операції з ними.

У цілому, проект сайту довідника розкладів польотів літаків, побудований на Vue 3 з використанням Quasar та JSON файлу як бекенду, є вдалим прикладом використання сучасних технологій для створення зручного та ефективного веб-додатку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Project Development Manual: [Електронний ресурс] – Режим доступу до ресурсу: <https://www.dot.ny.gov/divisions/engineering/design/dqab/pdm>
2. Guide to Database Schema: [Електронний ресурс] – Режим доступу до ресурсу: <https://www.integrate.io/blog/complete-guide-to-database-schema-design-guide/>
3. Інформація про бібліотеку quasar [Електронний ресурс] – Режим доступу до ресурсу: <https://quasar.dev/>
4. Інформація про фреймворк vue 3: [Електронний ресурс] – Режим доступу до ресурсу: <https://vuejs.org/>