Flight Ticket Price Finder Web App

Group-19 Leafs members:

Pal Rohit Tilva

Kaiwen Tao

Navjot Singh

Karnpreet Cheema

GitHub repo:

https://github.com/CMPT-276-SPRING-2025/final-project-19-leafs

Overview

Finding affordable and convenient flight tickets can be time-consuming and overwhelming. Users often need to search multiple websites, compare prices, and check real-time flight statuses before making a booking decision. Our project aims to simplify this process by providing a user-friendly interface to search for flights, check prices, and get real-time flight updates, all in one place. Additionally, an AI-powered chatbot will assist users with their queries, making the experience more interactive and efficient.

Problems to solve

This project originated from the need for a streamlined flight search experience that minimizes manual effort and enhances user convenience. By integrating the Amadeus API and OpenAI API, we provide users with real-time flight information and intelligent chatbot support, ensuring a seamless booking process.

Potential Users:

- Casual Traveler
- Business Traveler
- Family Traveler

List of Apis – 4

Main APIs

- Amadeus (https://developers.amadeus.com/self service/category/flights)
- OpenAI (https://platform.openai.com/docs/overview)

Backup APIs

- AviationStack (https://aviationstack.com/documentation)
- Google Gemini (https://ai.google.dev/gemini-api/docs)

Description of features to implement for each API

Amadeus API Features

- **Flight Offers Search:** Search flights from over 400 airlines based on departure, destination, date, and budget.
- **Flight Offers Price:** Confirm final pricing (after taxes).
- **Flight Status API:** Provide real-time flight schedule updates, including departure/arrival times, terminal, and gate information.

Feature	**API Used**	**Endpoint**
Flight Search	Flight Offers Search	/v2/shopping/flight-offers
Final Pricing	Flight Offers Price	/v1/shopping/flight-offers/pricing
Live Flight Updates	Flight Status API	/v2/schedule/flights

OpenAI API Features

- **Chat Completion:** AI chatbot answers flight-related queries.
- **Context Memory:** Helps users by remembering previous responses.
- **Summarization:** Provides clear explanations of ticket policies and booking details.

Feature	**API Used**	**Endpoint**
Chatbot (Flight Queries)	Chat Completion API	/v1/chat/completions
Context Memory	Chat Completion API (with history)	/v1/chat/completions
Summarization	Chat Completion API	/v1/chat/completions

AviationStack API Features

- **Flight Offers Search:** Search for flights based on departure, destination, and date.
- **Flight Offers Price:** Confirm final flight pricing (taxes).
    - **NOTE:** AVIATIONSTACK **DOES NOT PROVIDE** PRICING DATA DIRECTLY.
    - **Workaround**: Retrieve flight details from AviationStack.
    - Redirect users to airline booking pages for final pricing.
- **Flight Status API:** Get live flight updates, including departure/arrival times, terminals, and gates.

Feature	**API Used**	**Endpoint**
Flight Offers Search	Flights API	/v1/flights
Flight Offers Price	Not available	Use airline sites
Flight Status API	Flights API	/v1/flights

Google Gemini API Features

- **Chat Completion:** AI chatbot answers flight-related queries.
- **Context Memory:** Helps users by remembering previous responses.
- **Summarization:** Provides clear explanations of ticket policies and booking details.

Feature	**API Used**	**Endpoint**
Chat Completion	Gemini Pro Model API	/v1/models/gemini-pro:generateContent
Context Memory	Gemini Pro Model API	/v1/models/gemini-pro:generateContent
Summarization	Gemini Pro Model API	/v1/models/gemini-pro:generateContent

Persona for each user group

Persona 1: Casual Traveler

Name: Emily Carter

Age: 29

Occupation: Marketing Specialist

Tech Proficiency: Moderate

Motivation: Looking for affordable and convenient flight options for vacations and work trips.

Pain Points: Struggles with comparing prices, tracking flights, and understanding ticket policies.

Persona 2: Business Traveler

Name: Daniel Lee

Age: 38

Occupation: Sales Manager

Tech Proficiency: High

Motivation: Needs quick, reliable, and efficient flight booking with minimal hassle.

Pain Points: Frequent last-minute travel changes, flight delays, and managing expenses.

Persona 3: Family Traveler

Name: Sarah Johnson

Age: 35

Occupation: Stay-at-home Parent

Tech Proficiency: Moderate

Motivation: Traveling with kids and family, prioritizing safety, convenience, and cost-effectiveness.

Pain Points: Managing multiple tickets, flight delays, and navigating airline policies for children.

User stories: 1 each for each feature

Main API 1: Amadeus API

1. **Flight Offers Search**

   As a traveler, I want to search for flights from over 400 airlines based on my departure, destination, date, and budget, so that I can quickly find options that fit my travel plans.

2. **Flight Offers Price**

   As a traveler, I want to confirm the final pricing of flights—including all taxes and fees—so that I can accurately plan my budget and avoid unexpected costs.

3. **Flight Status API**

   As a traveler, I want to receive real-time flight schedule updates, including departure/arrival times, terminal, and gate information, so that I can manage my travel schedule effectively.

Main API 2: OpenAI API

4. **User Story 4: Chat Completion for Flight Queries**

   As a traveler, I want to interact with an AI chatbot that leverages chat completion, so that I can ask flight-related queries (e.g., baggage policies, seat selection, in-flight services) and receive immediate, accurate responses.

5. **User Story 5: Context Memory for Personalized Assistance**

   As a traveler, I want the AI chatbot to remember my previous interactions and preferences, so that my experience becomes more personalized, and I don't have to repeat my information.

6. **User Story 6: Summarization of Ticket Policies and Booking Details**

   As a traveler, I want the AI chatbot to provide a concise summary of ticket policies, cancellations, and booking details, so that I can quickly understand the key information without wading through lengthy text.

Backup API 1: AviationStack API

7. **User Story 7: Flight Offers Search**

   As a traveler, I want to search for flights based on departure, destination, and date using a backup API, so that I can still find available flights if the primary service is down.

8. **User Story 8: Flight Offers Price**

   As a traveler, I want to retrieve flight details for pricing via a backup API and be redirected to the airline booking page, so that I can confirm the final cost even when direct pricing data is unavailable.

9. **User Story 9: Flight Status (Backup)**

   As a traveler, I want to receive live flight status updates—including departure/arrival times, terminal, and gate information—from a backup API, so that I remain informed during my journey.

Backup API 2: Google Gemini API

10. **User Story 10: Chat Completion for Flight Queries**

    As a traveler, I want to use an AI chatbot powered by the backup API to ask flight-related queries, so that I can continue receiving assistance if the primary chatbot is unavailable.

11. **User Story 11: Context Memory for Ongoing Assistance**

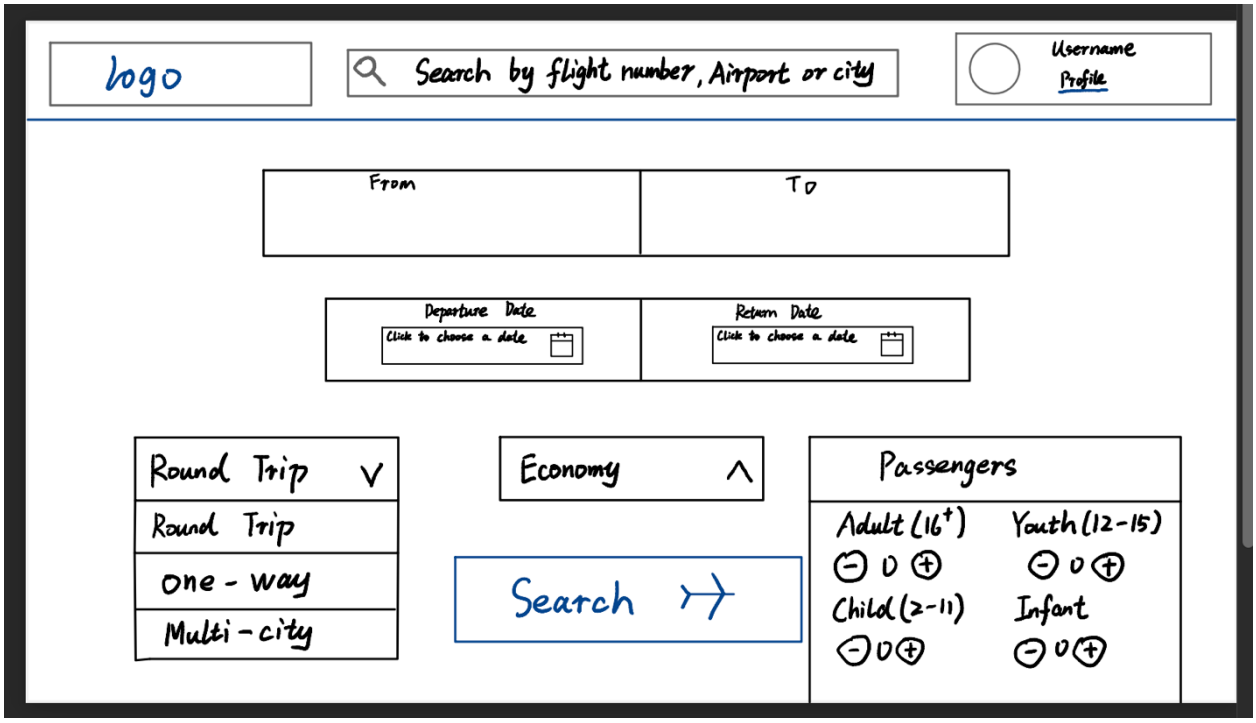
    As a traveler, I want the backup AI chatbot to remember my previous queries and preferences, so that I can benefit from a personalized interaction even when the main service is down.

12. **User Story 12: Summarization of Ticket Policies and Booking Details**

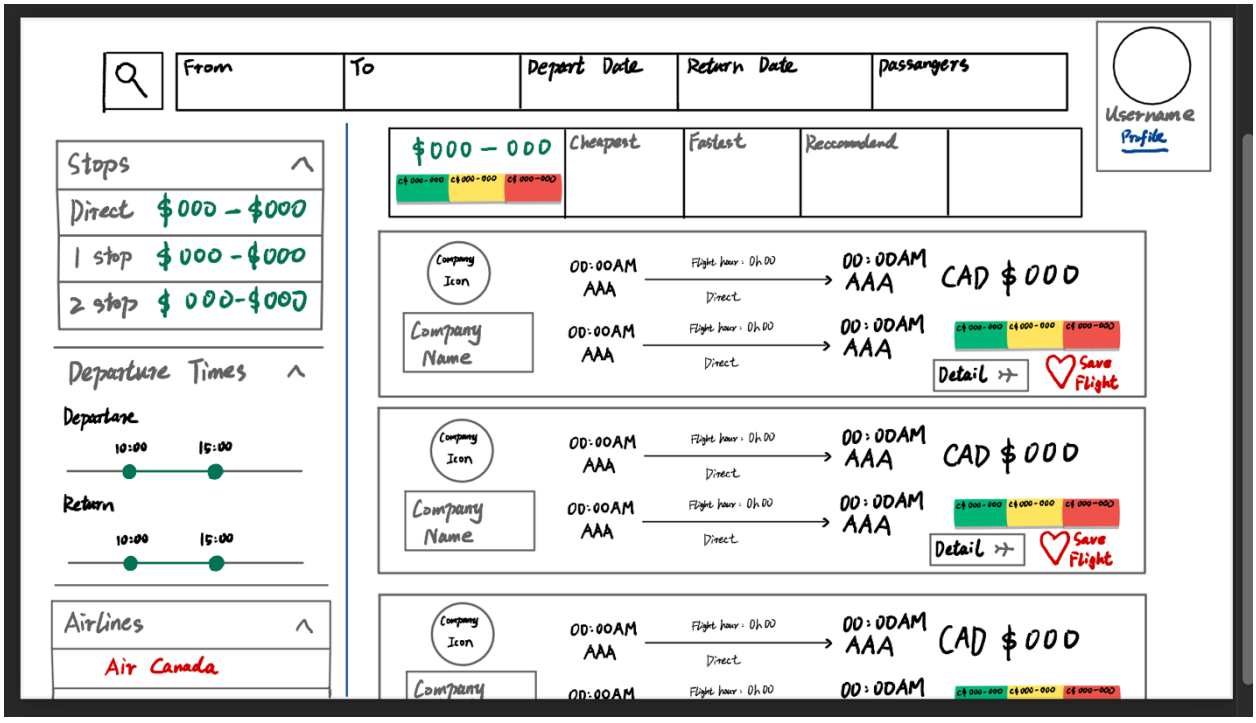
    As a traveler, I want the backup AI chatbot to provide a clear and concise summary of ticket policies and booking details, so that I can quickly understand the essential information without confusion.

Low fidelity interfaces

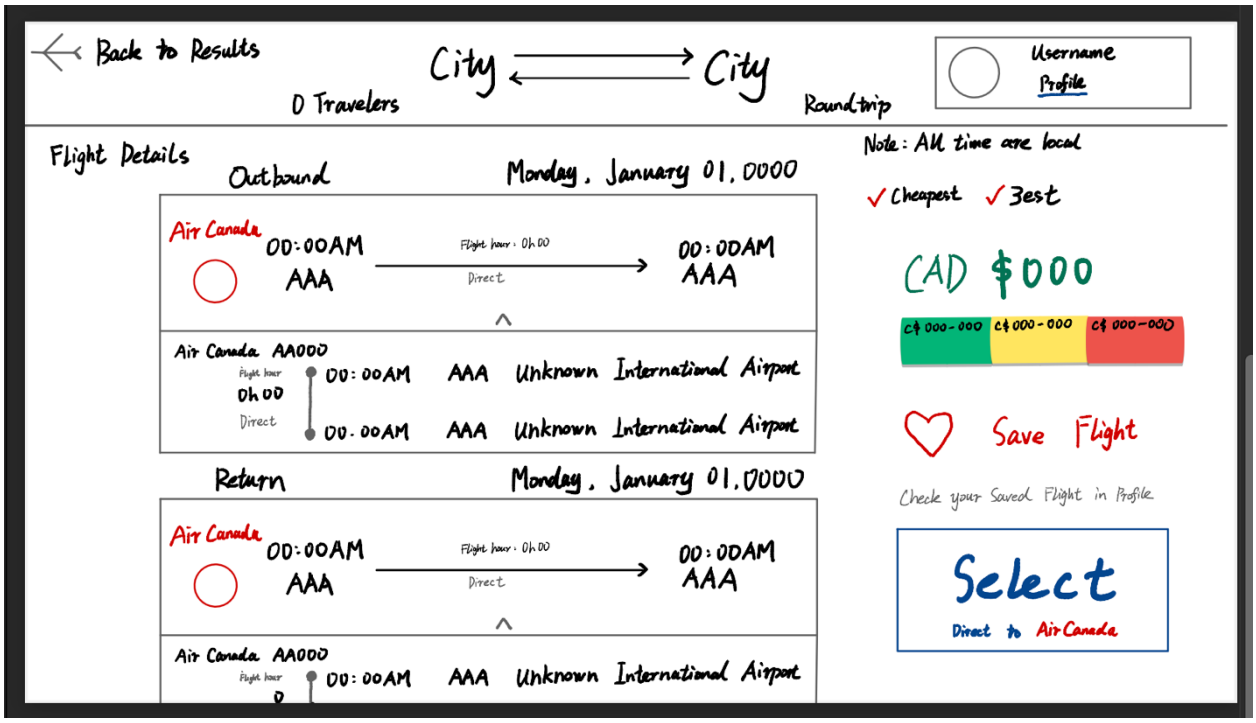
1. Home/Landing Page



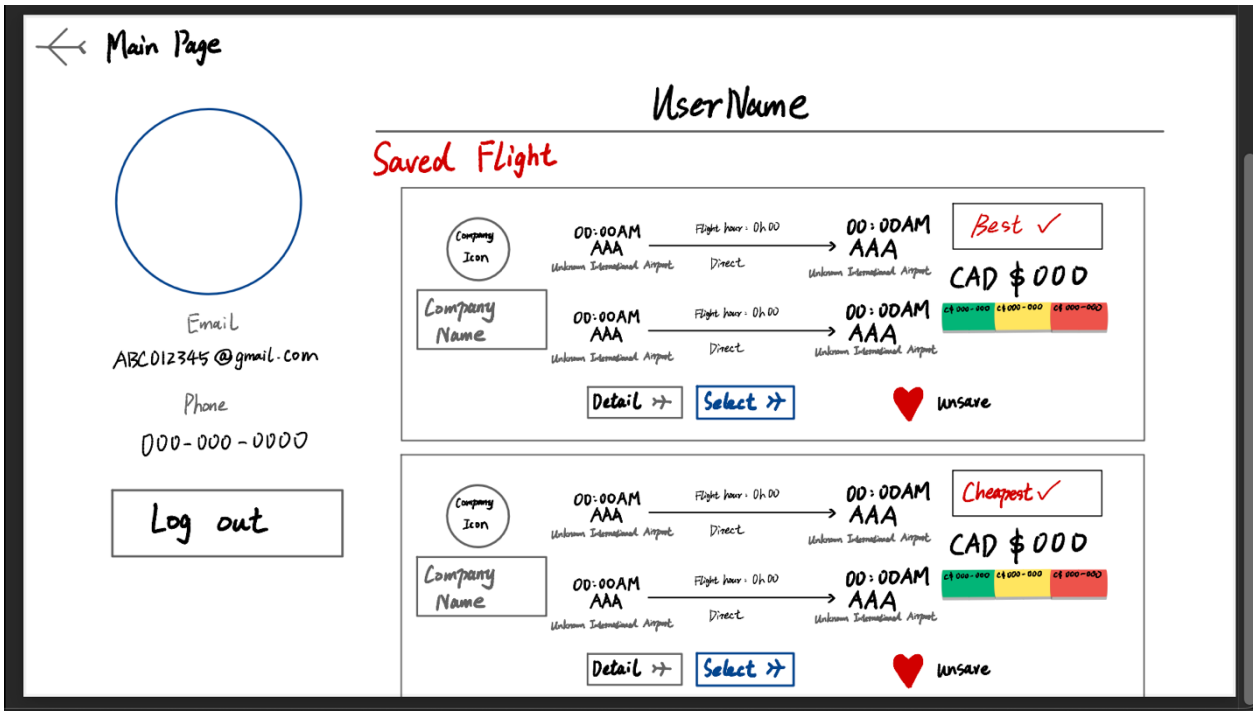
2. Flight Options Page



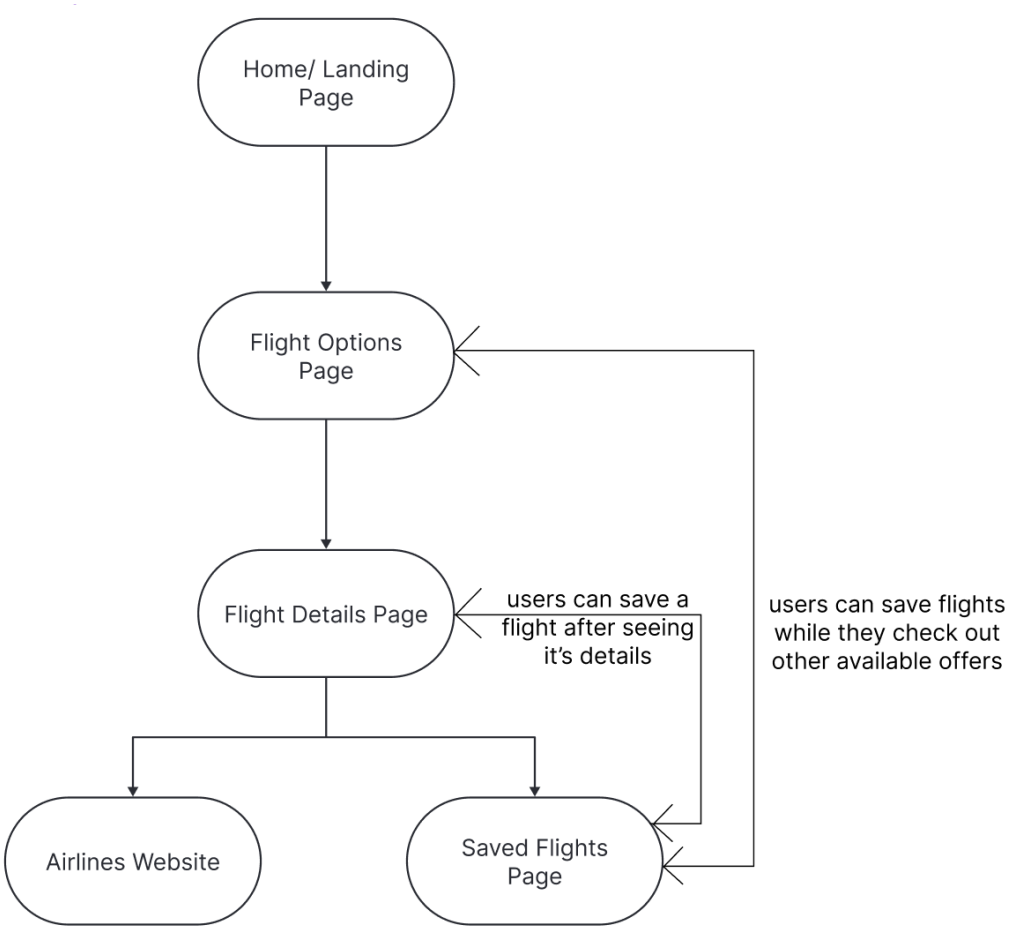
3. Flight Details Page



4. Saved Flights Page



User Flow



Also available on: https://www.figma.com/design/Y3XsMWMfqNgebyQjEJscRI/Untitled?node-id=0-1&t=yy94rCy9z3Ph8RQu-1

Tech stack:

- **Frontend Framework**
  - React.js
    Reason:
    - Component-based structure makes it easy to manage UI.
    - Efficient API calls using fetch or Axios.
    - React Query can handle caching API responses.
- **UI Library**
  - Tailwind CSS
    Reason: Lightweight and customizable for styling.
- **Testing**
  - Jest
    Reason: to test chatbot or API responses
- **Deployment**
  - Vercel
    Reason:
    - Simple, free hosting for frontend-only apps.
    - Automatic CI/CD from GitHub repo.