

FlyTinerary



Capstone Project
Isaac Heslop | 04.04.2023

Overview

Capstone Project

Introduction

Product Description

Architecture

User Stories & User Flow

Wireframe Design

Scope

In scope

Out of scope

Non-functional requirements

Project Planning

Trello Board

Testing Strategy

Implementation

Solution Review

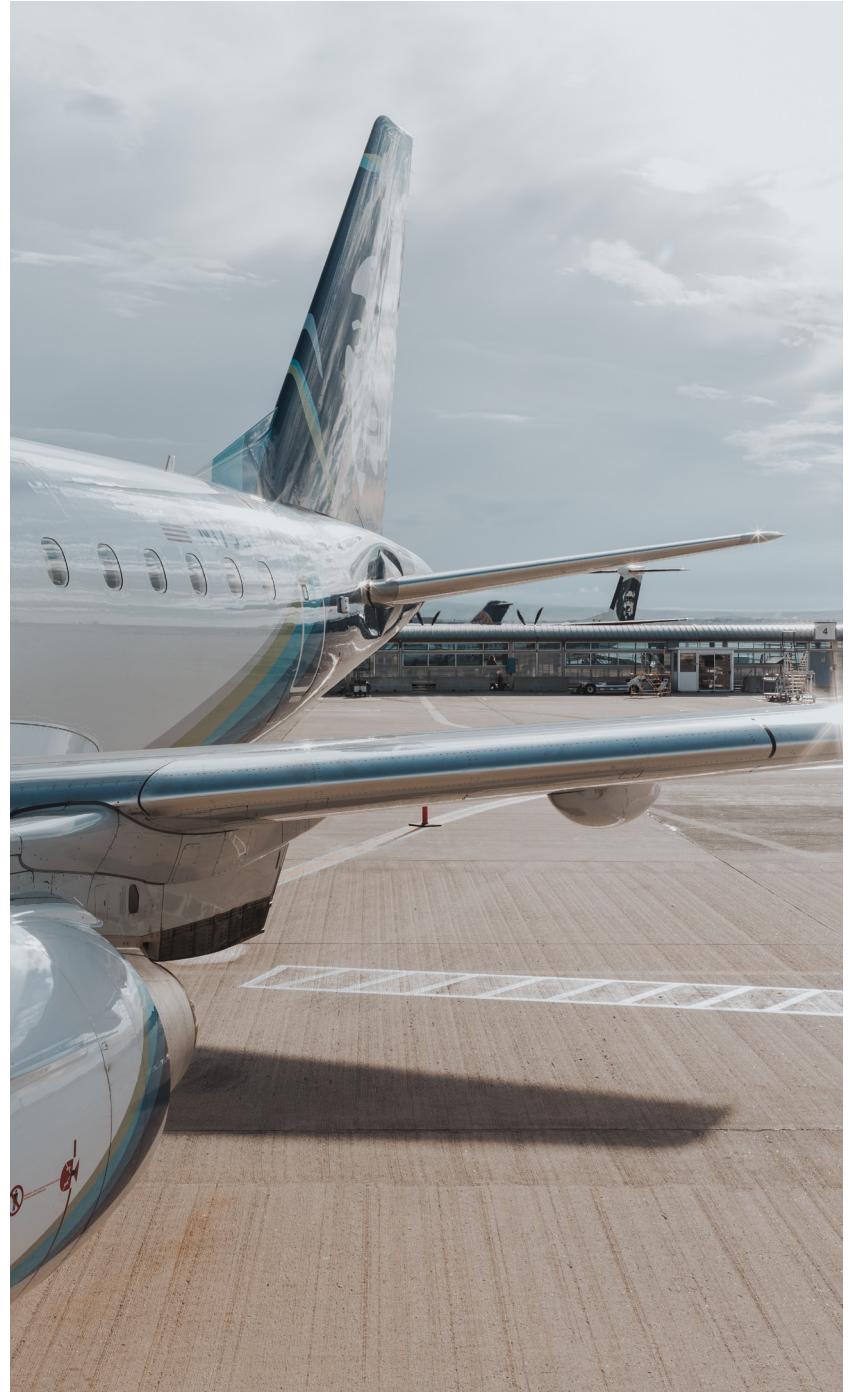


Introduction



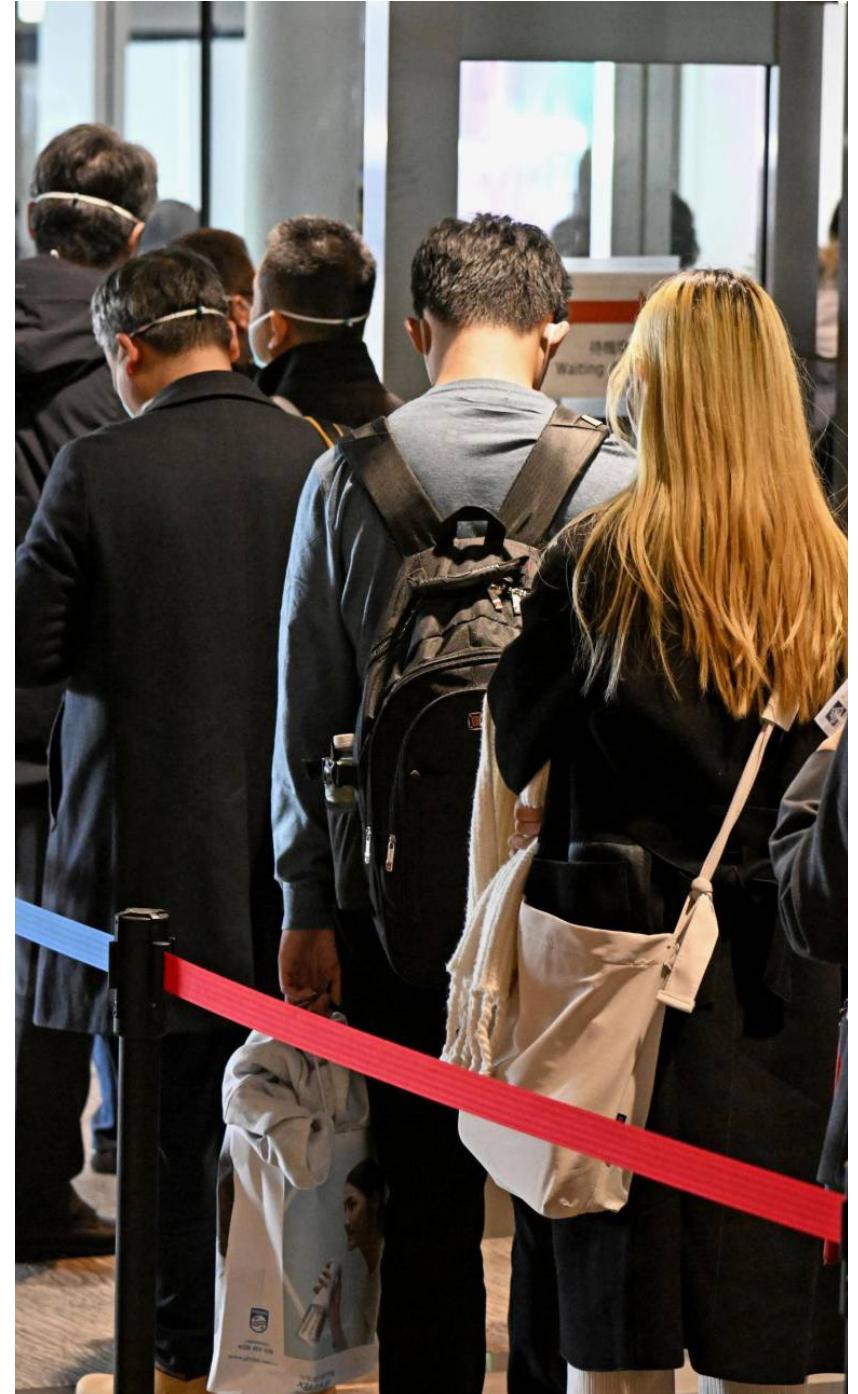
Purpose

FlyTinerary was born out of a desire to visualize and manage flights in a simple and easy-to-understand way. There are many flight tracking websites and applications out there, but none visualize air travel in a way that is simple and engaging.



Industry

Often booking holidays can become mundane and stressful, as your flight information appears as words on a screen, and columns or rows in an excel spreadsheet. FlyTinerary provides a way to visualize travel in three dimensions.



Stakeholders

This application is designed for anyone with a heart for travel, especially those looking to book extensive and complicated holidays.

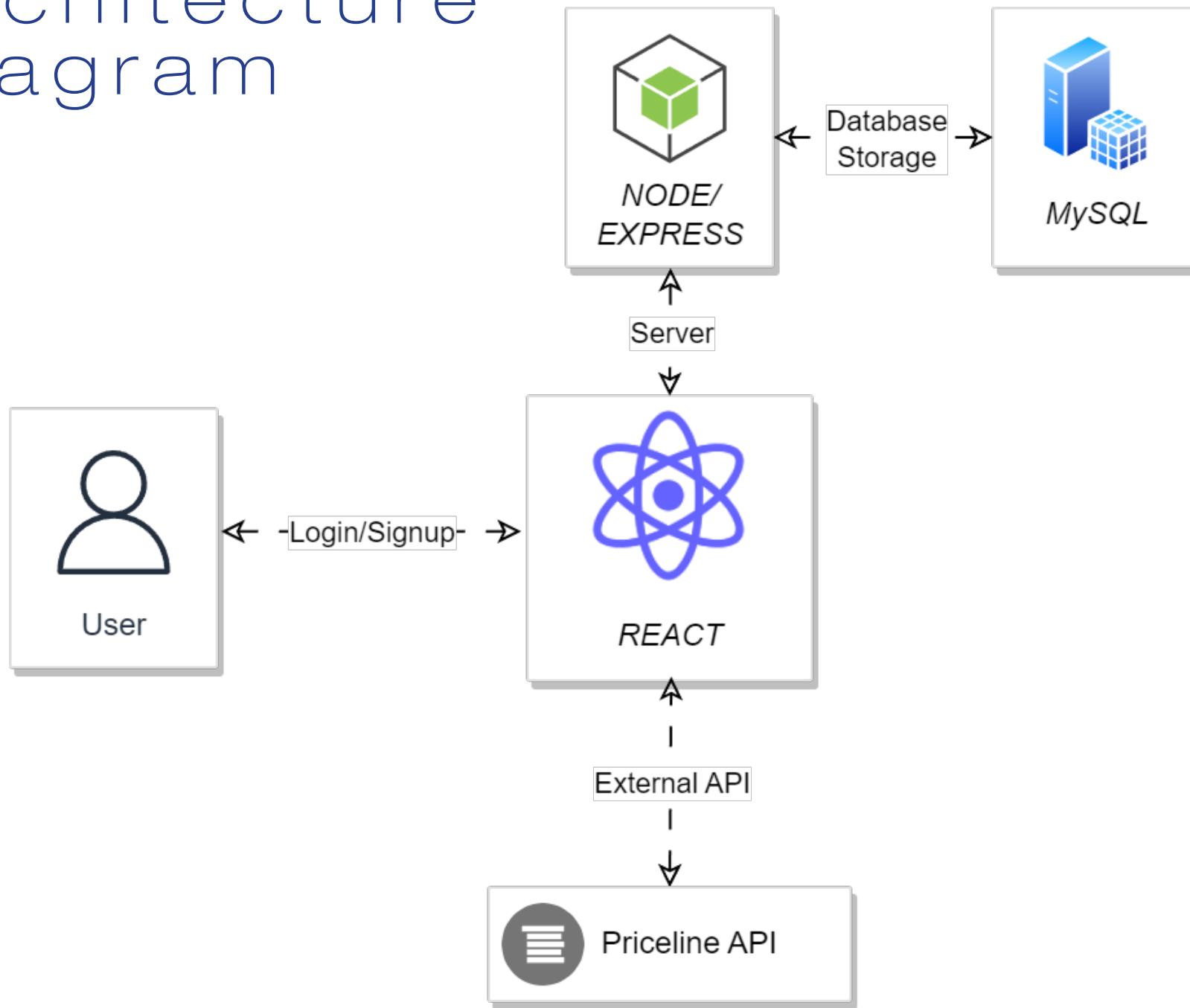
Users can expect a fast and responsive UI, which allows for real time flight data to be fetched and displayed, with the ability to create custom, personal itineraries containing flight details.



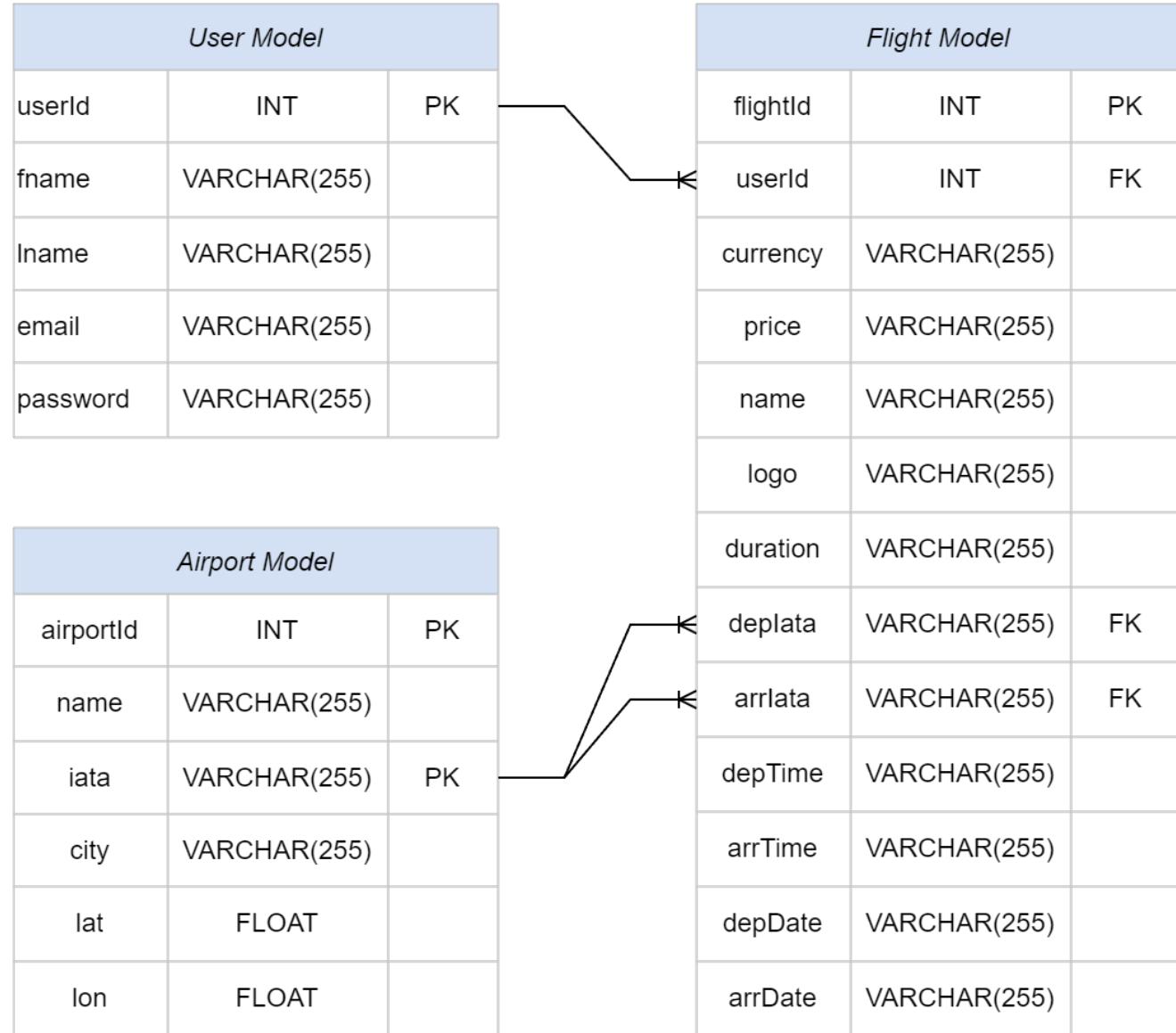
Product Description



Architecture Diagram



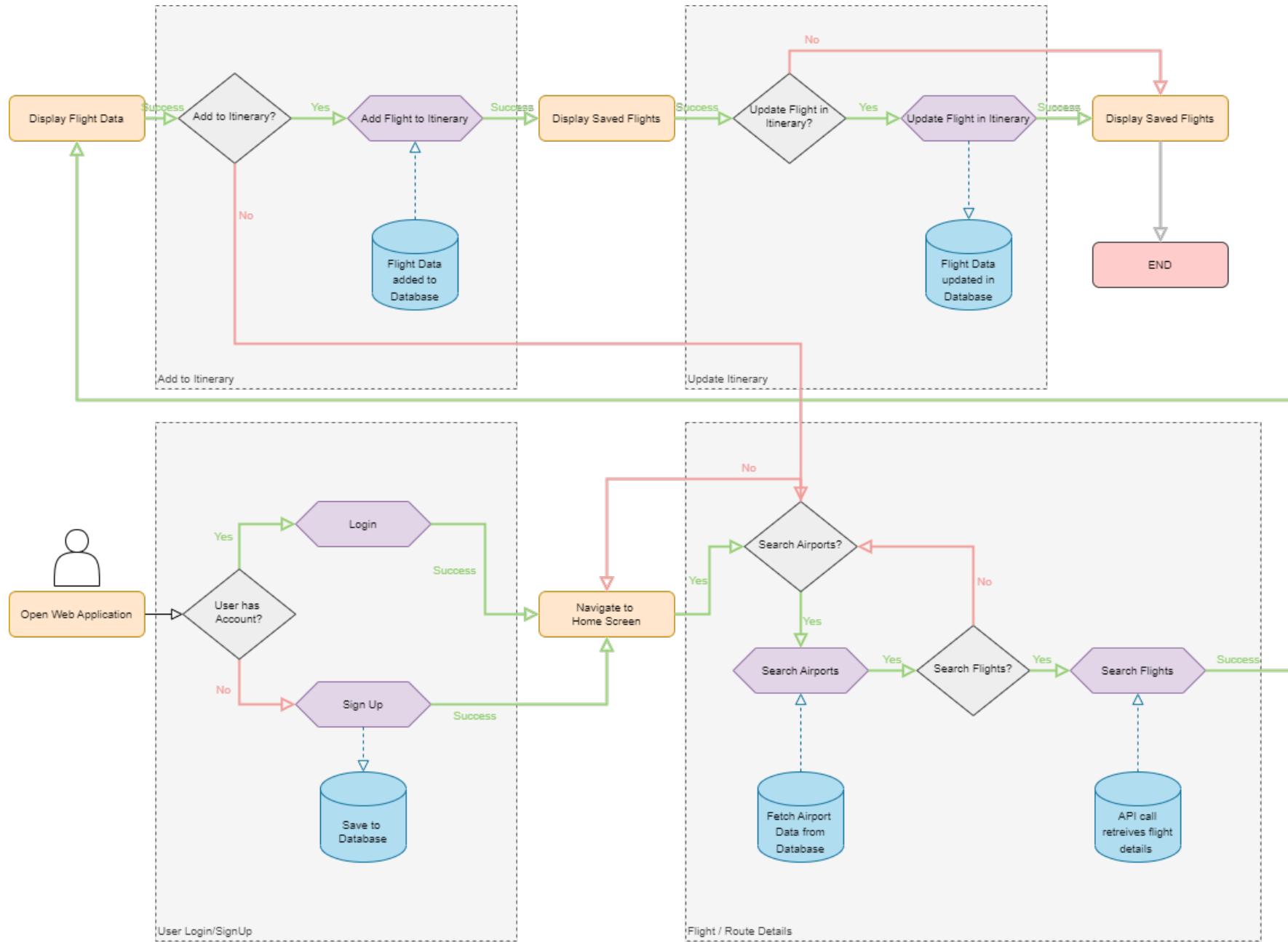
Database Diagram



User Stories

Title	Description	Priority (1-5)	Notes
Login/Sign Up Page	I want the login/signup page to be visually dynamic to set the tone for the rest of the application	2	Accomplished
User Login/Signup	I want the user to be able to login or signup simply and efficiently	1	Accomplished
Search Airports	I want the user to be able to search through a database for any airport worldwide	1	Accomplished
Route Visualisation	I want the chosen route to be visually represented on a globe	1	Accomplished
Search Flights	I want the user to be able to search for available flights between two airports	1	Accomplished
Flight Details	I want the details of the flight to be displayed and shown efficiently and in a minimal design	2	Accomplished. <i>Possible future feature / out of scope.</i> View multiples legs of journey not added
Add Flight to Itinerary	I want the user to be able to add the flight details to their own personal itinerary	1	Accomplished
View Itinerary Flight Details	I want the users flight information to be displayed and shown efficiently design-wise	2	Accomplished
Visualize Custom Itinerary	I want the users chosen flight routes to be displayed visually on a globe	1	Accomplished
Update Flight Price	I want the user to be able to update the price of the flights	3	Accomplished
Remove Flight from Itinerary	I want the user to be able to remove the flight from their itinerary	2	Accomplished
Smooth Transitions	I want the application to render smoothly, with minimal clunkiness and render issues.	3	<i>Possible future feature / out of scope.</i> Globe component can sometimes be slow to render.
Responsiveness / Mobile Version	I want the application to be optimized for both browser and mobile based use	4	<i>Possible future feature / out of scope.</i>
Airport City Information / Attractions	I want to display a list of attractions for each airport/city	5	<i>Possible future feature / out of scope.</i>

User Flow



Wireframe Design

The wireframe design for FlyTinerary consists of six pages arranged in a 2x3 grid:

- Login Page:** Features a "FlyTinerary" header, "Login" and "Sign Up" buttons, input fields for "Email Address" and "Password", a "Forgot password?" link, and a large blue "Login" button.
- Globe Page (Home):** Shows a green globe with several red dots representing plotted locations. Includes a sidebar with "FlyTinerary" and "MyTinerary" sections, and buttons for "From...", "To...", "Find Flights", and "Clear Selections".
- My Itinerary:** Displays a green globe with red lines connecting red dots, representing flight paths. A sidebar lists "SAVED FLIGHT" entries.
- Signup Page:** Similar to the Login page, featuring "FlyTinerary" branding, input fields for "Name", "Email Address", and "Password", and a large green "Sign Up" button.
- Globe Page (Attractions):** Shows a green globe with one red dot labeled "(Location Selected)". Includes a sidebar with "FlyTinerary" and "MyTinerary" sections, and a "Search" button.
- Globe Page (Flights):** Shows a green globe with a single red line representing a flight path. Includes a sidebar with "FlyTinerary" and "MyTinerary" sections, and a "Flights List: A -> B" section displaying five flight options with logos, departure times, arrival times, and prices.

FIGMA Link:

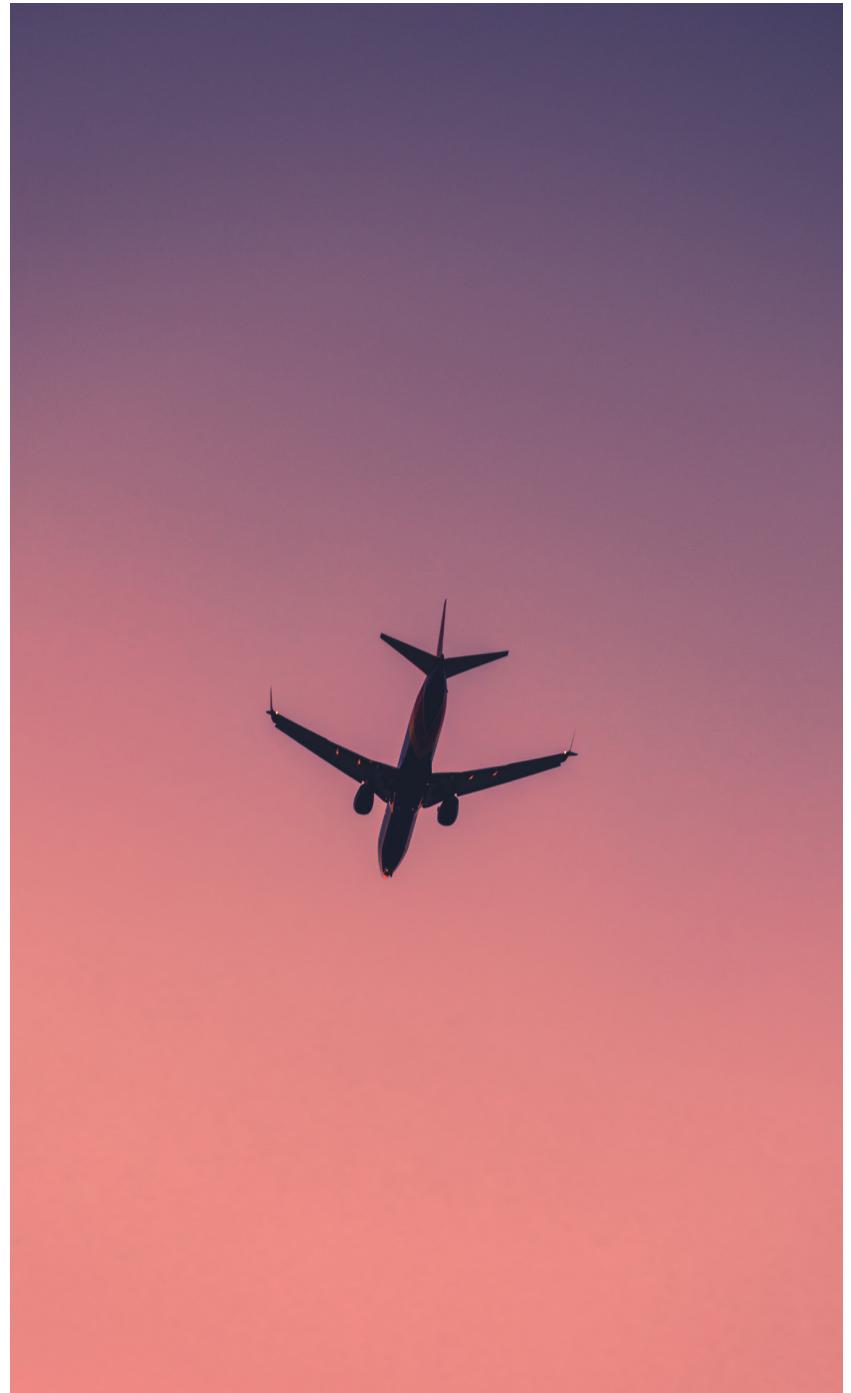
<https://www.figma.com/file/NDjjZbTeQyFxF8UTrwIYTf/FlyTinerary?node-id=0%3A1&t=H5xHNUEl8NfWOYco-1>

A photograph taken from an airplane window during sunset or sunrise. The sky is a gradient of dark blue at the top, transitioning through purple and pink to a bright yellow and orange near the horizon. Below the horizon, a vast, arid landscape stretches towards the viewer, featuring rolling hills and mountains. In the bottom left corner, the edge of the airplane's wing and engine are visible, pointing towards the horizon.

Scope

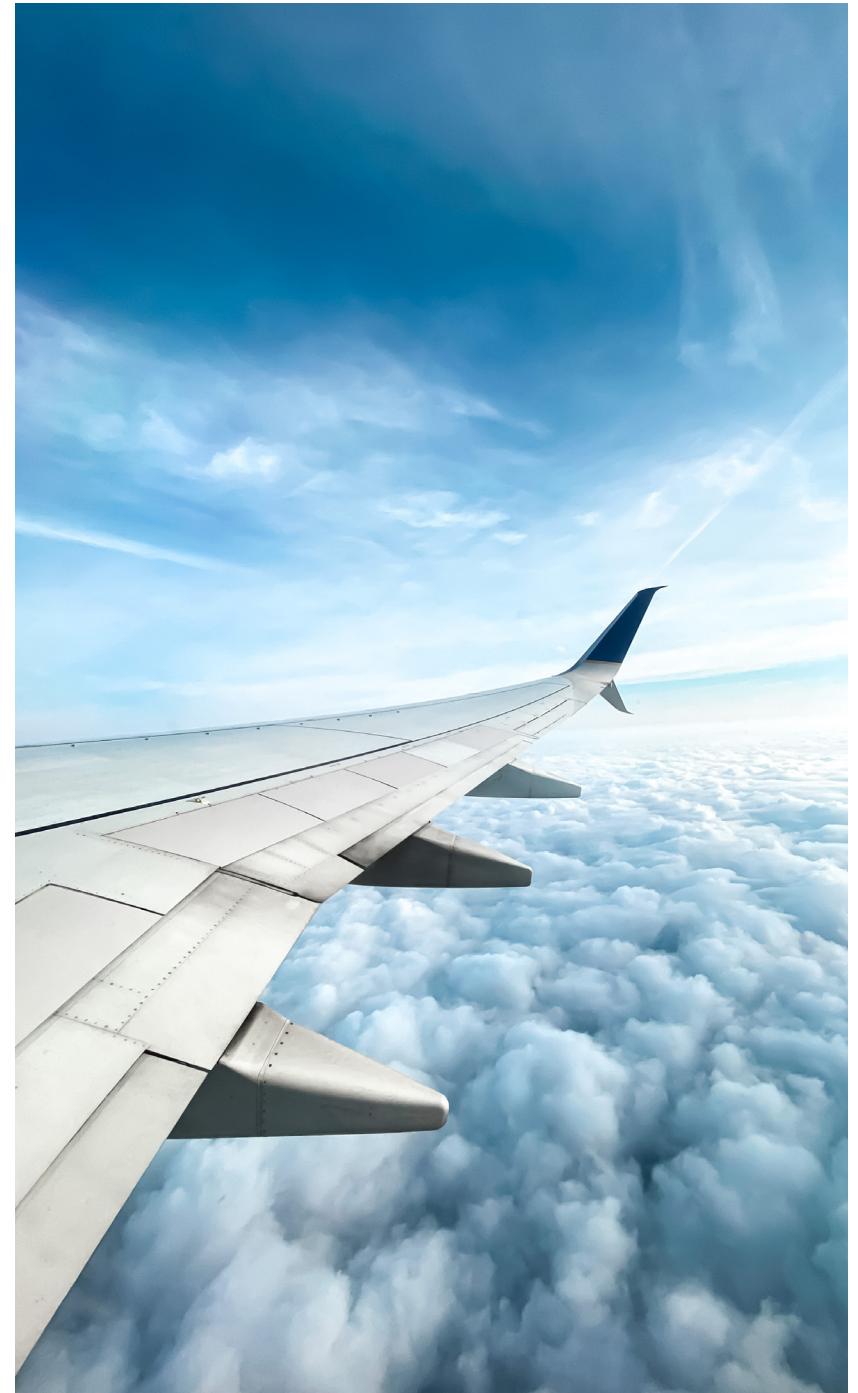
In Scope

- Users can login or signup and have their details saved for future app usage.
- Users can select a departure and arrival airport and visualize the route on the globe.
- Users can see real time flight data between two airports.
- Flights can be added to own personal itinerary.
- Itinerary graphically visualized on a globe, showing multiple flight paths.
- Flights can be removed from Itinerary.
- Flight Price can be updated within itinerary.
- Users can logout at any time.



Out of Scope

- View multiple legs of journey on globe. Currently only shows initial departure and final arrival airport.
- Globe component buggy, sometimes doesn't render properly.
- Add mobile version or update current application to be responsive.
- Add city markers, which allow users to see different attractions in each location.



Non-functional Requirements

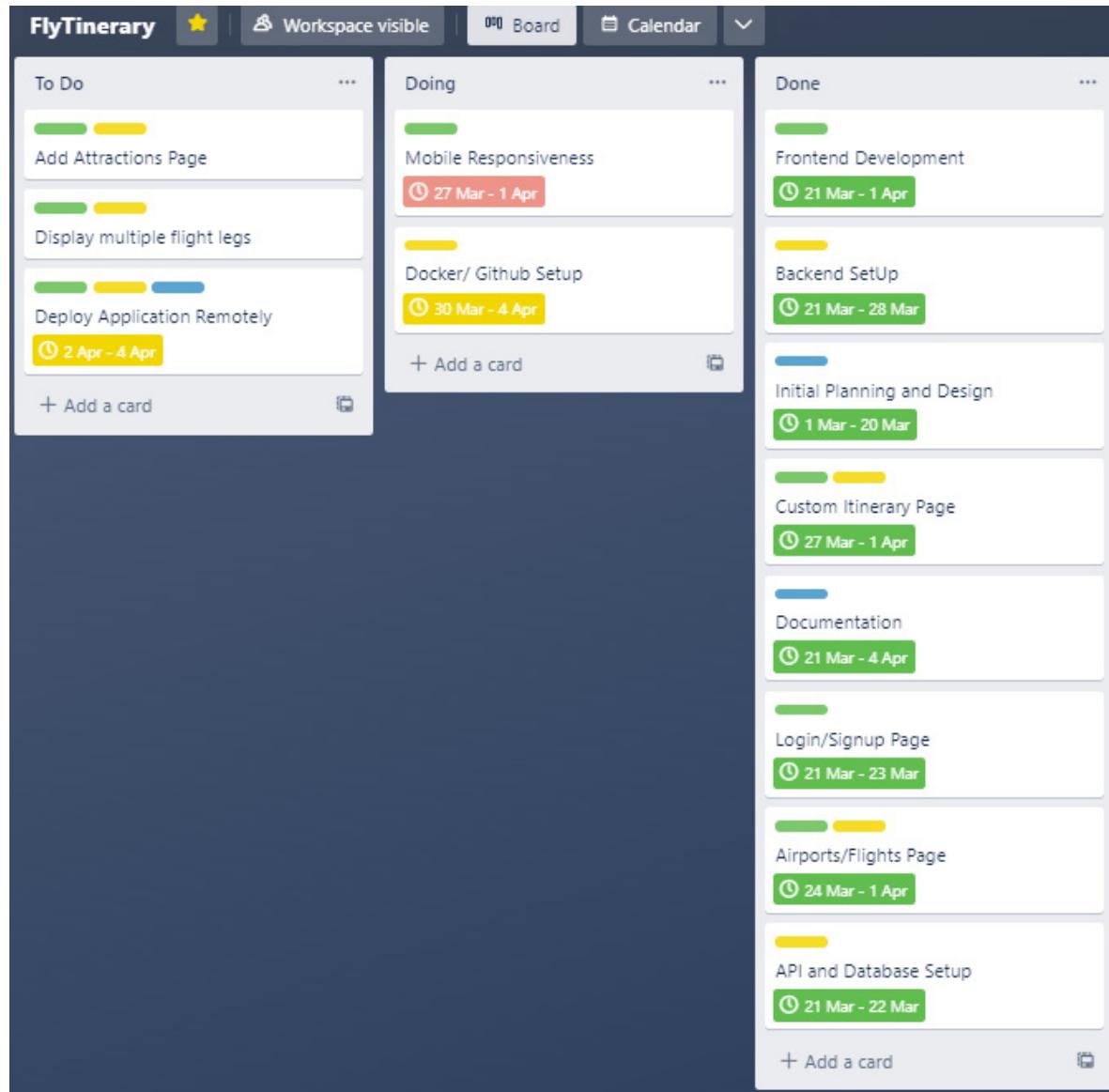
- Login and Signup: In the future add password encryption and a session timeout.
- Ease of use: Functionality and familiar patterns for UI.
- Responsiveness. Currently the API latency is approx. 5200 ms for flight data
- Security: Security measures will need to be added to provide security for sensitive user information.
- Deployment: Application hasn't been deployed yet and is currently ran locally.
- Maintainability: MVC Structure with separate front and backend applications. Github workflow to docker containers for frontend. Creating a remote accessible backend that is easily accessible would be a future design measure/feature.



Project Planning



Trello Board



TRELLO Link:

<https://trello.com/invite/b/YrzMJ3x3/ATTI02a3aadfd0bd065cf803b488e48a28bf06020350/flytinerary>

Testing Strategy

Login/Signup Page:

- Login:
 - o Click login and have user information stored in local storage.
 - o If email or password is invalid, provide an error message.
- Signup:
 - o Click signup and have user information added to database.



Select Airports Page:

- Select Airports:
 - o Airports in database sorted by user input.
 - o Search airports either by airport or city name
- Visualize Flight Path
 - o Display flight path from selected airports.
 - o Update flight path when a different flight or route is selected.

Select Flights:

- Display flight data which matches users chosen parameters
 - o Pagination allows user to cycle through results.
 - o Display correct flight data (first departure airport info and final arrival airport info)
- On click, provide option to add flight to itinerary.
 - o Flights added to itinerary with current userId as reference.

Itinerary:

- Display saved flight data:
 - o Allow multiple flights and their data to be displayed.
 - o If flight is removed from itinerary, flight data must disappear too.
- Visualize flight routes:
 - o Saved flight routes shown on globe.
 - o If the flight is removed from itinerary, then the flight route disappears too.
- On click provide option to remove from itinerary or update flight price
 - o User input updates flight price in database.
 - o Remove from itinerary removes flight from database

Implementation

Application Installation

Pre-requisites:

- Node.js/NPM: <https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>
- mySQL/Workbench:
 - o For Mac:
 - mySQL: <https://dev.mysql.com/doc/refman/5.7/en/macos-installation.html>
 - Workbench: <https://dev.mysql.com/doc/workbench/en/wb-mac.html>
 - o For Windows:
 - mySQL: <https://dev.mysql.com/doc/refman/5.7/en/windows-installation.html>
 - Workbench: <https://dev.mysql.com/doc/workbench/en/wb-windows.html>

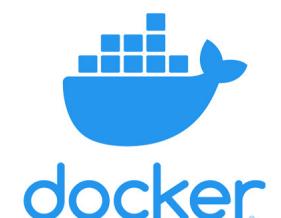
Application Set-Up:

Front-end and back-end Installation:

- Method 1: Github Clone
git clone <https://github.com/iHeslop/FlyTinerary.git>
cd Flytinerary

```
// Install the Backend:          //Install the frontend:  
cd back-end                      cd front-end  
npm install                         npm install
```

- Method 2: Docker Container (Frontend only)
https://hub.docker.com/r/iheslop/flytinerary_frontend



Implementation

Database Setup

Once installed, both methods will require a mySQL Database:

1. First create a .env file in the “flytinerary/backend” folder containing this information:

```
DB_NAME=flytinerary  
DB_USER=*your user here*  
DB_PASSWORD=*your password here*  
DB_HOST=localhost  
DB_PORT=3306
```

2. In mySQL workbench, create a new schema called flytinerary.

Remember to update the DB_USER and the DB_PASSWORD in the .env file with your local user and password

3. Execute the SQL query in the flytinerary-queries file in this repository to set up your initial database table.

4. FlyTinerary should be set up and ready to go! Just run npm start in both the back-end and front-end folders. If you can login/sign-up the database is working!



Solution Review



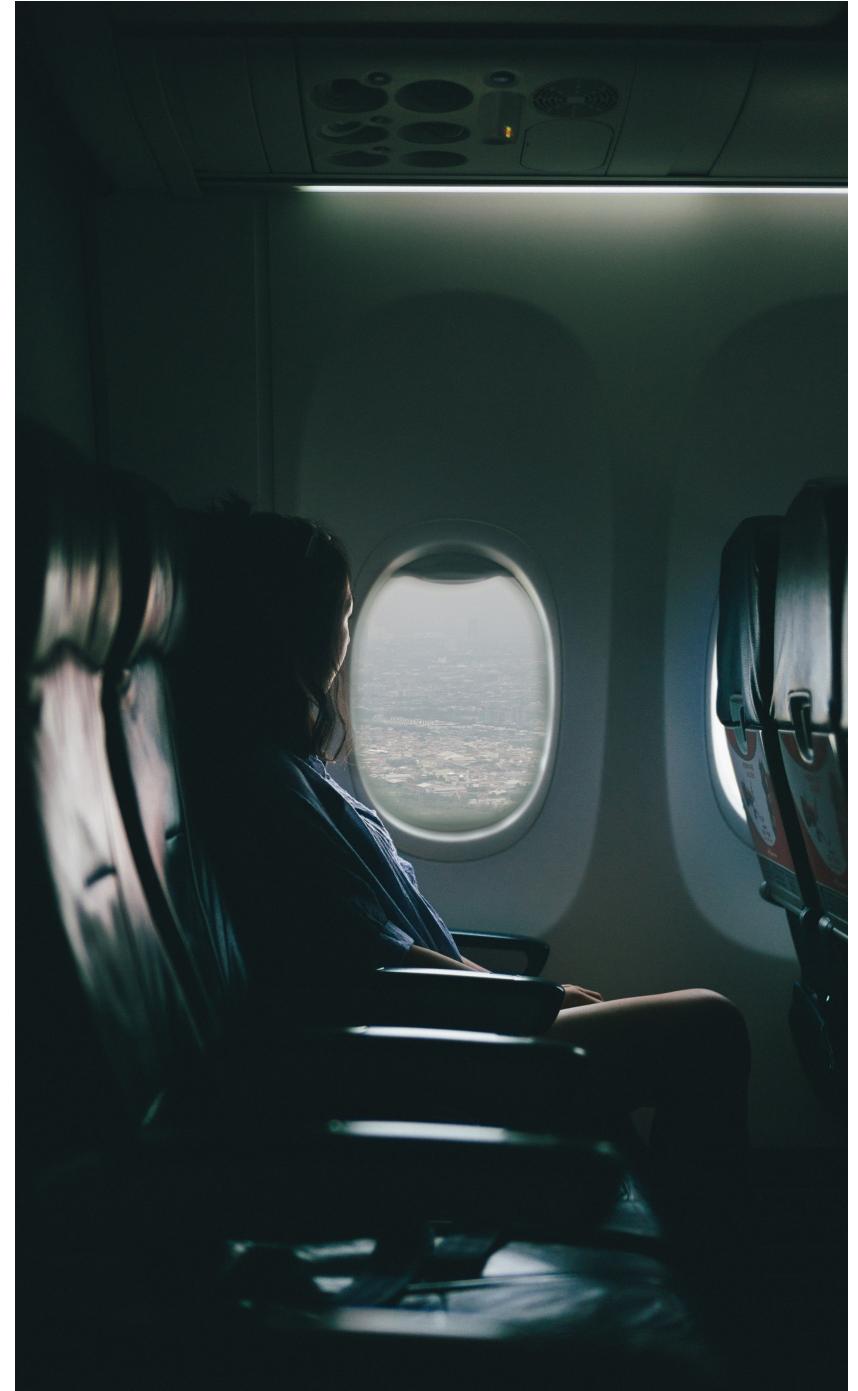
End-to-End Solution

Currently, FlyTinerary is a basic flight visualization application which allows for users to create custom itineraries and edit these said itineraries.

The application aims to provide a simple and easy way to visualize travel routes, using real time flight data across the globe in a fast and efficient way.

There are multiple ways to increase the scope of the application, which would provide extra benefits to the user experience, but due to time constraints these additions fell out of the scope.

In its current state, FlyTinerary is effective in achieving its initial goals, with room for expansion and enhanced experience still possible.



References:

- Github: Version Control & Storage

<https://github.com/iHeslop/FlyTinerary>

- Docker: Portable Container

https://hub.docker.com/r/iheslop/flytinerary_frontend

- mySQL: Database storage and management

<https://www.mysql.com/>

- Priceline com Provider API

<https://rapidapi.com/tipsters/api/priceline-com-provider>

- Material UI: UI components and design

<https://mui.com/>

- Postman: Database CRUD operations testing

<https://www.postman.com/>

- Node.js:

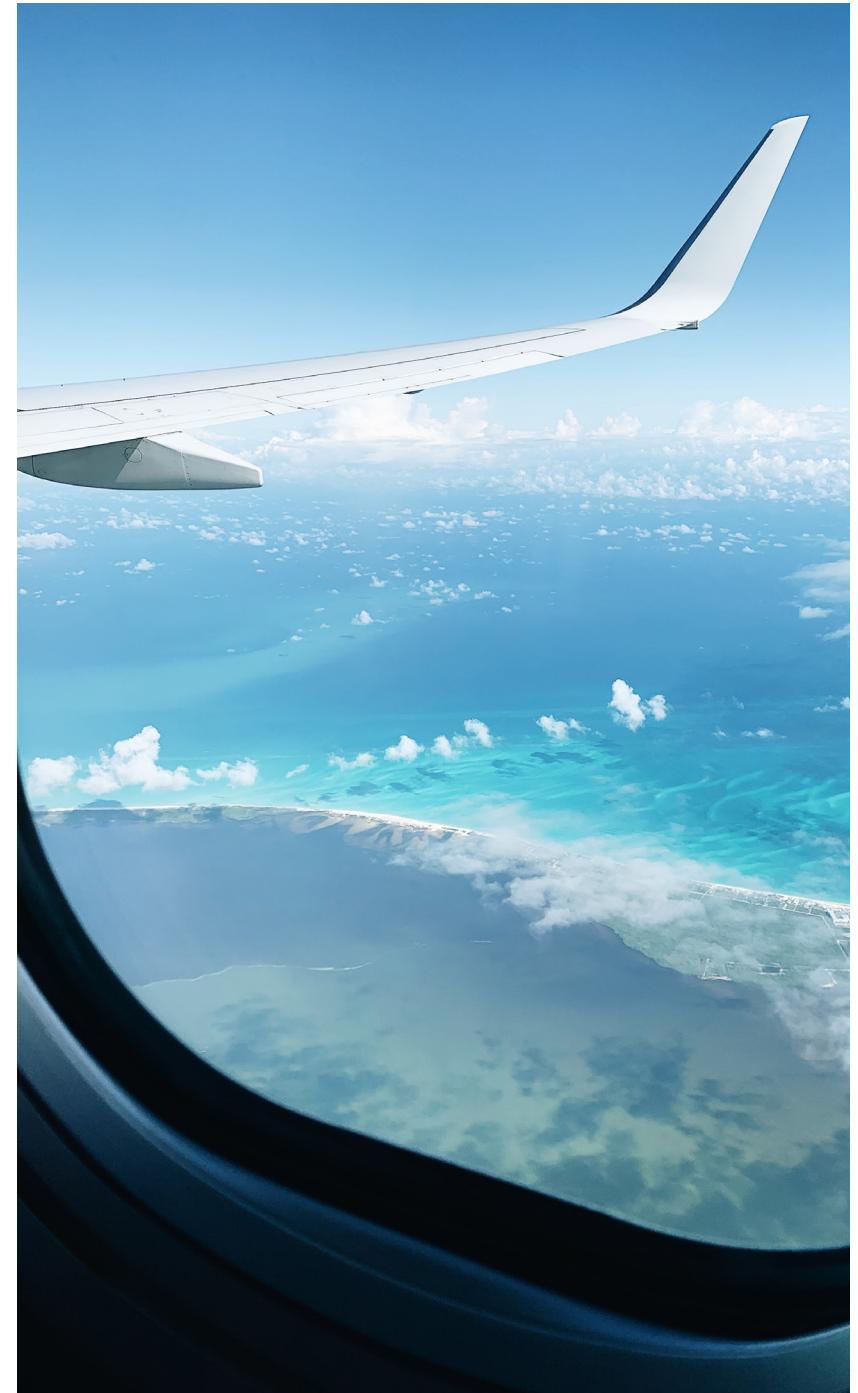
<https://nodejs.org/en>

- Express.js:

<https://expressjs.com/>

- React:

<https://react.dev/>





Isaac Heslop | 04.04.2023