Design Doc:
Flights Viewer

For simplicity, both the backend and frontend will be contained in the same GitHub. If this were a larger project I would most definitely separate the Front and Backend.

Project Instructions:

Goal:

- Create a single-page application that allows users to upload a CSV containing their favorite airports. The CSV is then ingested, and transformed into a shareable public map, showcasing the user's submitted airports according to their latitude and longitude.

React Frontend

Axios for api requests
Google-maps-react
Uuidv4 to generate unique id for map

Base page: CSV upload

- create a unique id and send CSV in post request to API

Base page + id in the url:

- Checks to see if map with that id exists and if it does
    - Get airports locations and data from the backend and display on google map
- If the ID does not exist, redirect back to CSV upload with message saying map does not exist

Rails Backend

- Use Ruby on Rails built-in CSV library to process CSV and then store data in the database
- add each airport to map
- Create database relationship between locations and maps
- Possibly move to worker if files are too big

Database:

- There will be a join table for the map and locations because they have a many-many relationship
- Tables
    - Map
        - ID
        - Unique ID
    - Location
        - ID
        - Unique ID
        - Airport Name
        - City
        - Country
        - IATA/FA
        - ICAO
        - Latitude
        - Longitude
        - Altitude
        - Timezone

Backend API endpoints:

- upload CSV (Post)
    - create map with unique id passed from frontend
    - parse csv and add each airport to map
- get map (GET)
    - get request with map id
    - returns all airports data for that map