Directory Guide

Work division:

Daniela: Front end code. All front end related code was coded by Daniela.

Colton: Back end code. All back end flask + sql code was coded by Colton.

This project is built upon Flask and Vue.js, where flask serves the HTML page which is populated and dynamically altered by Vue. Using Vue, the project directory structure has many extra components and files, so this document will only summarize the important files that direct the business logic of the program.

```
/frontend/src/
├── components
│   ├── Bar.vue
│   ├── Bookings.vue
│   ├── Forms.vue
│   ├── Home.vue
│   ├── NotFound.vue
│   ├── Pie.vue
│   ├── Spending.vue
│   └── Statistics.vue
├── router
│   ├── index.js
│   └── App.vue
├── main.js
```

router/App.vue
The main Vue page that gets rendered globally. Here global variables and login and logout function are added. The other components get rendered through this page too. Menu button added here too.

router/index.js
This page is where I handle routing from one page to another

Main.js
Where the View is rendered. Module imports to the app are done here

components/Bar.vue

The component where the bar graphs are created. This component is imported in Statistics.vue and Spending.vue

components/Bookings.vue
The page where users can see their upcoming flights. All user cases are included here.
Customer user case: view my flights
Booking agent user case: view my flights
Airline Staff user case: view my flights

components/Forms.vue
Page where airline staff can add/update new information to the DB
Airline Staff user case: create new flights, change status of flight, add airplane, add airport

components/Home.vue
Main page where users can search for flights, buy tickets and check flight status
Customer user case: purchase ticket, search for flights,
Booking agent user case: purchase ticket (on behalf of customer), search for flights,

components/NotFound.vue
A simple page where 404 not found is shown if wrong routing happens ex: going to a http address that does not exist

components/Pie.vue
The component where the pie graphs are created. This component is imported in Statistics.vue for Airline Staff use case (comparison commission)

components/Spending.vue
Where customers will be able to see their spending
Customer user case: Track my spending

components/Statistics.vue
Page for booking agents and airline staff. Many statistics can be seen here
Booking agent user case: view my commission, view top customers
Airline staff user case: view all booking agents, view frequent customer, view reports, comparison of revenue earned, top destinations

backend/requirements.txt
Import requirements in main.py

/main.py:
The main server file of the program. Runs the flask application and handles all API calls to the server.

/static/python/helpers.py:
Contains a few helper functions for the server, including a function for hashing passwords, checking passwords, for making labels and values for the barchart.

/dbfiles/airplanes.sql
Exported SQL database

/dbfiles/build_fake_data.sql
The test inserts that we started with for airports, airplanes, et

/dbfiles/sql_queries.sql
A summary file of the sql queries that interact with our database

/dbfiles/final_schema_build_tables.sql
Builds the database. Unlike airplanes.sql, this only contains the schema information and none of the extra inserts.