Project File Descriptions

Client Files (/client/)

- public/index.html - Root HTML file for all pages
- src/helpers/auth.js - creates an authentication context for user authentication when logged in
    - Used when displaying different webpages specific to users (customer or staff) who are logged in, and validating user sessions when logged in
- src/pages/
    - CustomerHome.js - Customer Use Cases 1, 4
    - CustomerReview.js - Customer Use Case 5
    - CustomerSearch.js - Customer Use Cases 2, 3
    - CustomerSpending.js - Customer Use Case 6
    - Home.js - View public information, Login, Register
    - LoginCustomer.js - Login as Customer
    - LoginStaff.js - Login as Staff
    - Logout.js - Customer Use Case 7, Staff Use Case 12
    - RegisterCustomer.js - Register as Customer
    - RegisterStaff.js - Register as Staff
    - StaffAdd.js - Staff Use Case 4, 5
    - StaffCustomerDetails.js - Staff Use Case 6, 7
    - StaffHome.js - Staff Use Case 1, 2, 3
    - StaffReport.js - Staff Use Case 8, 9, 10, 11
- styles/ - CSS styles
    - App.css
    - RegisterCustomesr.css
    - RegisterStaff.css
- App.js - navigation page/component; routes to all other pages
- index.js - root element for React, renders App component.

Server Files (/server/)

- middleware/auth.js - Handles authentication through access tokens
- models/
    - User_queries.js - all functions containing SQL queries called in the backend for general user (customer or staff) related tasks
    - Flight_queries.js – all functions containing SQL queries called in the backend for flight-related tasks
    - Ticket_queries.js – all functions containing SQL queries called in the backend for ticket/purchasing/canceling related tasks
- routes/

    - Customer.js - Customer router; Interfaces with MySQL through queries in models/
    - Flight.js - Flight router; interfaces with MySQL through queries in models/Flight_queries.js
    - Login.js - Login routers responsible for interacting with Staff and Customer database to verify users and create sessions through bcrypt and access tokens
    - Register.js - Register routers responsible for creating new users and hashing passwords with bcrypt
    - Staff.js - Staff routers responsible for interacting with MySQL by calling functions from ‘models/’ and sending the subsequent data to the front end for staff specific use cases
- server.js - creates the server and instantiates backend routes
- db.js - establishes the connection to MySQL server and database on our local machine