Airline Reservation System

1. Public home page
    a. View all flights in future 30 days
        i. Frontend: on our home page, we utilized the table class we defined to format the data sent from the backend to be displayed to the user. When the home page is loaded, data will be requested from the backend.
        ii. Backend: the backend uses a query to filter flights only in the future 30 days to be displayed.
    b. Search for flights
        i. Frontend: Both on the home page and client home page, there is a form that asks the user to fill out source city, destination city, departure date, arrival information. After they click the search flight button, their flight information will be displayed.
        ii. Backend: Use sql query to filter out Flight table's information that matches the form information user put in in the form from front end.
2. Login page
    a. Client login
        i. Frontend: clients will need to enter their email and password in order to login. If not all the fields are filled, an alert will be shown. The user-entered data will be sent to the backend to be verified. If their username or password is invalid, an alert will be shown. They will be automatically directed to the client homepage once they successfully logged in. An authentication is also updated so the users' status can be changed to logged in and will stay as logged in.
        ii. Backend: the user-entered data will be send to the backend, where the data will be verified. The user-entered password will first be hashed with md5 since the password stored in our database is also hashed with md5. The user's email will then be found in our database with a query so we can locate this user and fetch their stored password. The backend will either return true for the user to successfully login or send a message notifying the user about invalid username or password.
    b. Staff login
        i. Frontend: staffs will need to enter their username and password in order to login. If not all the fields are filled, an alert will be shown. The user-entered data will be sent to the backend to be verified. If their username or password is invalid, an alert will be shown. They will be automatically directed to the staff homepage once they successfully logged in. An authentication is also updated so the users' status can be changed to logged in and will stay as logged in.
        ii. Backend: the user-entered data will be send to the backend, where the data will be verified. The user-entered password will first be hashed with md5 since the password stored in our database is also hashed with md5. The user's username will then be found in our database with a query so

we can locate this user and fetch their stored password. The backend will either return true for the user to successfully login or send a message notifying the user about invalid username or password.

3. Logout
    a. Frontend: both client and staff share the same logout functionality. Once the button “logout” is clicked, the authentication status will be changed to false so the user can no longer access the client or staff homepage.
    b. Backend: there is no backend processing for this functionality. The backend will log “logout” in the console for reference.
4. Register page
    a. Client registration
        i. Frontend: in order to register for an account, the client will be required to fill in all the fields, if not all the fields are filled, an alert will be shown. Once the user has filled in all the required fields, if the email already exist, an alert will be displayed. If the email does not exist, they will be greeted and their data will be send to the backend.
        ii. Backend: once the data is sent to the backend, the data will be inserted to the client table with a query. Note that the password stored in our table is the md5 hashed password, but not the original password.
    b. Staff registration
        i. Frontend: in order to register for an account, the staff will be required to fill in all the fields, if not all the fields are filled, an alert will be shown. Once the user has filled in all the required fields, if the username already exist, an alert will be displayed. If the username does not exist, they will be greeted and their data will be send to the backend.
        ii. Backend: once the data is sent to the backend, the data will be inserted to the staff table with a query. Note that the password stored in our table is the md5 hashed password, but not the original password.
5. Client functionalities
    a. Homepage
        i. Frontend: the user must be logged in to view the client homepage. The authentication status will be checked when they attempt to access the client homepage. If they are not logged in, an alert will be shown and they will be automatically redirected to the public homepage. If they are logged in, all client functionalities will be displayed as buttons, so the user can be directed to appropriate pages.
        ii. Backend: there is no backend data processing for this functionality.
    b. View my flights
        i. Frontend: When the user clicks view my flights, the page will automatically display the flights of the user. We get the user by getting the user's email ID from where they logged in. After displaying all of their flights, the users have the option to enter the airline name. As they enter the airline name, the page will display all the flights’ ames that match what the user enters.

        ii. Backend: The backend checks if the user has inputted an airline name, and if not, the page will just display all the flights from that user. If there is an airline name input, the back end will further filter out the flights for the user.

    c. Purchase ticket
        i. Frontend: Purchase ticket is built in through search for flights page. After the user searches for the flight with the features that they desire to have, there is a purchase button for each flight that is found through the form they put in. Once they press the button, the specific flight will be purchased for them and added to the database.
        ii. Backend:Once the user decides to purchase a ticket, that ticket's information will be stored into the Ticket table and Purchase table.
    d. Cancel Trip
        i. Frontend: The customer will click the display your ticket IDs to get the IDs of tickets they have purchased. After displaying the ID, they will enter the ticket ID and other information of the flight that they wish to be canceled. After they fill out the form, they need to click the button for the selected data to be deleted from the Purchase Table and Ticket Table.
        ii. Backend:When the button your ticket IDs is clicked, the backend will find out all the ticket ids of this user by their email. After they got their flight id and inputted the form and clicked the confirm cancel button, their flight information will be deleted from purchase and ticket table.
    e. Review
        i. Frontend: Review function is built to view my flights. After the user clicks review flights, they will be transferred to the review page. They will need to fill out a form with the airline name, departure date (YYYY-MM-DD), departure time(hour:minute:second), flight number, and their rating and comment for that flight.
        ii. Backend:The backend will get the form date they user puts in, and store them into the feedback table.
    f. Track spending
        i. Frontend: When the user clicks the Track Spending button, by default, the system will display their spending for the last 6 months from the current date. If they fill out the Start date and End date, instead of displaying the last 6 months, it will display the spending of the user in their specified start date and end date.
        ii. Backend: Gives out the ticket number, and in the front end it will calculate the amount of money a user spent each month.

6. Staff functionalities
    a. Homepage
        i. Frontend: the user must be logged in to view the staff homepage. The authentication status will be checked when they attempt to access the client homepage. If they are not logged in, an alert will be shown and they will be automatically redirected to the public homepage. If they are logged

in, all staff functionalities will be displayed as buttons, so the user can be directed to appropriate pages.

  ii. Backend: there is no backend data processing for this functionality.

b. View flights
   i. Frontend: when clicked into view flights page, the page will automatically display the flights for the future thirty days that belong to the same airline company with the signed in staff and their information. The information of the flights for the future thirty days include the flight number, and airline company name, the departure date of the flight, and the departure time of the flight.
   ii. Backend: the backend checks the date of today, and calculates the date of the future 30 days. The flights and their information that departure within the future 30 days will be sent back to the frontend page.

c. Create new flight
   i. Frontend: to create a new flight, the staff is required to fill in all the fields required. If not all fields are filled, an alert will be shown. Once they have filled in all the required fields and click on the button, the data will be sent to the backend so it can be stored in our database.
   ii. Backend: once the data is received at backend, the new flight will be inserted to our database with a query.

d. Change status
   i. Frontend: in order to change the status of a flight, the staff should first search for the flight they want to change. The filled in data will be sent to the backend. Once the flight is found, the staff will be able to click on change, which changes the data from on-time to delayed or from delayed to on-time. The status can be changed as many times as possible.
   ii. Backend: once the filled in data is received at the backend, the backend will use a query to find the corresponding flight and report it to the frontend. Once the change request is received, a query is used to change the status in our database.

e. Create new airplane
   i. Frontend: to create a new airplane, the staff is required to fill in all the fields required. If not all fields are filled, an alert will be shown. Once they have filled in all the required fields and click on the button, the data will be sent to the backend so it can be stored in our database.
   ii. Backend: once the data is received at backend, the new airplane will be inserted to our database with a query.

f. Create new airport
   i. Frontend: to create a new airport, the staff is required to fill in all the fields required. If not all fields are filled, an alert will be shown. Once they have filled in all the required fields and click on the button, the data will be sent to the backend so it can be stored in our database.
   ii. Backend: once the data is received at backend, the new airport will be inserted to our database with a query.

g. View feedback
    i. Frontend: in order to view the feedback of any particular flight, the staff is required to fill in all the fields, including the flight number, departure date of the flight, departure time of the flight, and the airline name. If not all fields are filled, an alert will be shown. Once they have filled in all the required fields and click on the view button, the data will be sent to the backend so it can retrieve the feedback of the flight. Both the comment and the rating from all the customers who put down feedback of this flight will be displayed as a table.
    ii. Backend: when the required fields are filled, the feedback information of the particular flight will be retrieved and sent back to the frontend page.

h. View most frequent customer
    i. Frontend: when clicked into view most frequent customer page, the page will automatically display the information of the most frequent customer. The information includes the customer name, the date of birth of the customer, and the customer email address.
    ii. Backend: the backend retrieves the flight data, finds the most frequent customer. Then it will retrieve this customer information and send the information back to the frontend page.

i. View top destinations
    i. Frontend: when clicked into view top destinations page, the page will automatically display the top three most popular destination airports names.
    ii. Backend: the backend retrieves the tickets data, finds the top three most popular destination airports, and sends the names of these three airports back to the frontend page.

j. View report (within a year, within last month)
    i. Frontend: when clicked into view tickets sold page, the page will automatically display the information of the tickets. The information includes the ticket ID, sold price, card number that purchased the ticket, type of the card, expiration date, name of the card, purchase date, airline name, flight number, departure date, and the departure time.
    ii. Backend: the backend calculates the date of today, then retrieves the ticket data and finds the tickets that are sold a year before today's date, or a month before today's date. Then it will return the information of these tickets back to the frontend page.

k. View revenue of last year and last month
    i. Frontend: when clicked into show revenue of last year's or last month's page, the page will automatically display the revenue dollar amount. The total revenue is calculated on the frontend as well.
    ii. Backend: the backend calculates the date of today, then retrieves the sold values of all the tickets from a year or a month from today. The the sold values will be sent back to the frontend page to do the sum calculation.

l. View revenue of economy and business class

i. Frontend: when clicked into show revenue of economy or business page, the page will automatically display the revenue dollar amount. The total revenue is calculated on the frontend as well.
ii. Backend: the backend retrieves the tickets data, then finds all the tickets that are economy or business class. The sold values of the tickets will be sent back to the frontend to do the sum calculations.