Project File List

​​

| File name | What is in the file? |
| --- | --- |
| add\_airplane.html | add airplane page |
| add\_airport.html | add airport page |
| change\_status.html | staff changes flight status page |
| create\_new\_flights.html | staff create new flight page |
| customer\_home.html | customer home page |
| error.html | error page |
| index.html | login in and register tabs for page |
| login\_customer.html | customer login page |
| login\_staff.html | staff login page |
| login.html | login page with customer and staff tab |
| purchase.html | customer purchase ticket page |
| rate.html | customer rate flight page |
| register\_customer.html | customer register page |
| register\_staff.html | staff register page |
| register.html | register page with customer and staff tab |
| search\_flight\_staff.html | staff search flight page |
| search\_flight.html | customer search flight page |
| spending.html | track customer spending page |
| staff\_home.html | staff home page |
| table.html | template for flight information table |
| view\_avg\_flight\_ratings.html | staff view average flight ratings page |
| view\_customer\_in\_flight.html | staff view customer list in a flight page |
| view\_earned\_revenue.html | staff view earned revenue page |
| view\_flight\_ratings.html | staff view all flight ratings page |
| view\_frequent\_customers.html | staff view frequent customers page |
| view\_top\_destinations.html | staff view top 3 destinations page |
| ticket\_number.html | staff view past tickets sale page |
| app.py | python code for the back end and front end to run |
| airlines.sql | The final sql file that is used to build the database. It also contains test cases that team members used to test code. |

Queries and Use cases

Customer Log In

query = 'SELECT email, cust\_password,name FROM customer WHERE email = %s and cust\_password = md5(%s)'

cursor.execute(query, (email, password))

* The query returns the email name and customer password in the database. This is used to check if the user exists in the database. Password is hashed using md5 hashing algorithm.

Staff Log In

query = 'SELECT user\_name, password, airline\_name FROM airline\_staff WHERE user\_name = %s and password = md5(%s)'

cursor.execute(query, (username, password))

* The user name and password of staff in a particular airline. This is used to check if the staff details exists in the database.

Customer Register

ins = 'INSERT INTO customer (name, email, cust\_password, building\_num, street, city, state, cust\_phone\_num, passport\_num, passport\_expiration, passport\_country,cust\_dob ) VALUES(%s, %s,md5(%s),%s,%s,%s,%s,%s,%s,%s,%s,%s)'

* The query is used to insert new customer information into the database

Staff Register

ins = 'INSERT INTO airline\_staff VALUES(%s, %s, %s, %s, md5(%s), %s)'

phone\_que = 'INSERT INTO staff\_phone VALUES(%s, %s)'

cursor.execute(ins, (airline, username,first\_name, last\_name, password, dob))

conn.commit()

cursor.execute(phone\_que, (username, phone))

* The query is used to insert staff information, including staff phone number into the database.

Customer View Flight

query = 'SELECT ticket.ticket\_ID, flight.airplane\_ID, flight.airline\_name, flight.flight\_num, ticket.sold\_price, flight.status, flight.departure\_date\_time, flight.arrival\_date\_time, flight.departure\_airport\_code, flight.arrival\_airport\_code FROM flight, purchase, customer, ticket WHERE purchase.email = %s AND customer.email = purchase.email AND ticket.flight\_num = flight.flight\_num AND ticket.ticket\_ID = purchase.ticket\_ID'

* Displays details of all the flights customer purchased

*else*: # default: view future flights

query = 'SELECT \* FROM flight WHERE departure\_date\_time > CURDATE()'

* Default view, display all future flights

Staff View Flight

query = 'SELECT flight.airplane\_ID, flight.airline\_name, flight.flight\_num, flight.base\_price, flight.status, flight.departure\_date\_time, flight.arrival\_date\_time, flight.departure\_airport\_code, flight.arrival\_airport\_code FROM flight, airline\_staff WHERE flight.airline\_name = airline\_staff.airline\_name AND airline\_staff.user\_name = %s AND (departure\_date\_time BETWEEN CURDATE() AND DATE\_ADD(CURDATE(),INTERVAL 30 DAY))'

* View flight details of the airline that the staff is working for. View from last 30 days.

Customer View Rating

query = 'SELECT flight\_num, rating, comment FROM rate WHERE email = %s'

* View a customer’s rating of flights.

Customer View Spending

query = "SELECT MONTHNAME(purchase\_date\_time) as month, SUM(sold\_price) as month\_total FROM ticket WHERE email=%s AND DAY(%s)<=DAY(purchase\_date\_time) AND DAY(%s)>=DAY(purchase\_date\_time) AND MONTH(%s)<=MONTH(purchase\_date\_time) AND MONTH(%s)>=MONTH(purchase\_date\_time) AND YEAR(%s)<=YEAR(purchase\_date\_time) AND YEAR(%s)>=YEAR(purchase\_date\_time) GROUP BY YEAR(purchase\_date\_time), MONTH(purchase\_date\_time);"

* View customer’s spending on tickets in a range of dates, group by month and year.

Staff View tickets sold

query = 'SELECT MONTHNAME(purchase\_date\_time) as month, COUNT(ticket\_ID) as tick\_total FROM ticket WHERE airline\_name=%s AND purchase\_date\_time between %s and %s GROUP BY YEAR(purchase\_date\_time), MONTH(purchase\_date\_time);'

* View the number of tickets sold for each month, within a range of dates. Group by month

Staff Searches for Flight

query = 'SELECT \* FROM flight WHERE flight.airline\_name = %s AND flight.departure\_airport\_code = %s AND flight.arrival\_airport\_code = %s AND (flight.departure\_date\_time BETWEEN %s AND %s)'

* View flight details based on depart/arrive airport, and depart/arrive time and airline

Staff Create New Flights

query = 'SELECT flight.airplane\_ID, flight.airline\_name, flight.flight\_num, flight.base\_price, flight.status, flight.departure\_date\_time, flight.arrival\_date\_time, flight.departure\_airport\_code, flight.arrival\_airport\_code FROM flight, airline\_staff WHERE airline\_staff.airline\_name = flight.airline\_name AND airline\_staff.user\_name = %s AND (flight.departure\_date\_time BETWEEN CURDATE() AND DATE\_ADD(CURDATE(),INTERVAL 30 DAY))'

* Display the flights that already exist

query = 'INSERT INTO flight(airplane\_ID, airline\_name, flight\_num, base\_price, status, departure\_date\_time, arrival\_date\_time, departure\_airport\_code, arrival\_airport\_code) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)'

* Create a new flight by inserting values into flights table

Staff Add Airplane

query = 'SELECT airplane.airline\_name, airplane.airplane\_ID, airplane.num\_of\_seats FROM airline\_staff, airplane WHERE airline\_staff.user\_name = %s AND airline\_staff.airline\_name = airplane.airline\_name’

* View available airplanes

query = 'INSERT INTO airplane (airline\_name, airplane\_ID, num\_of\_seats) VALUES (%s, %s, %s)'

* Add airplanes into the database

Staff Add Airport

query = 'SELECT airport\_code, airport\_name, airport\_city FROM airport'

* Check if airport exist

query = 'INSERT INTO airport(airport\_code, airport\_name, airport\_city) VALUES (%s, %s, %s)'

* Add into airport database

Staff Change Flight Status

query = 'SELECT \* FROM flight, airline\_staff WHERE flight.flight\_num = %s AND flight.departure\_date\_time = %s AND airline\_staff.user\_name = %s AND airline\_staff.airline\_name = flight.airline\_name'

* Check if a flight exist

query1 = 'UPDATE flight SET status = %s WHERE flight\_num = %s AND departure\_date\_time = %s'

* Update the flight status of a flight

Staff View Flight Rating

query = 'SELECT rate.flight\_num, rate.email, rate.rating, rate.comment FROM airline\_staff, flight, rate WHERE airline\_staff.user\_name= %s AND flight.airline\_name = airline\_staff.airline\_name AND flight.flight\_num = rate.flight\_num order by rate.flight\_num'

* View all ratings from customers on a flight

Staff View Average Flight Ratings

query = 'SELECT rate.flight\_num, avg(rate.rating) FROM airline\_staff, flight, rate WHERE airline\_staff.user\_name= %s AND flight.airline\_name = airline\_staff.airline\_name AND flight.flight\_num = rate.flight\_num group by flight.flight\_num'

* View the average ratings of a flight

Staff View Frequent Customers

query = 'SELECT customer.name, customer.email FROM ticket, airline\_staff, customer WHERE airline\_staff.user\_name = %s AND ticket.airline\_name = airline\_staff.airline\_name AND ticket.email = customer.email AND ticket.departure\_date\_time BETWEEN DATE\_ADD(CURDATE(),INTERVAL -1 year) AND CURDATE() GROUP BY customer.email ORDER BY (count(customer.name)) DESC '

* View most frequent customers.

Staff View Earned Revenue

#for past year

query = 'SELECT sum(ticket.sold\_price) FROM ticket, airline\_staff WHERE airline\_staff.user\_name = %s AND airline\_staff.airline\_name = ticket.airline\_name AND ticket.purchase\_date\_time BETWEEN DATE\_ADD(CURDATE(),INTERVAL -1 year) AND CURDATE() group by airline\_staff.airline\_name'

* View last year’s earned revenue.

#for past three months

query1 = 'SELECT sum(ticket.sold\_price) FROM ticket, airline\_staff WHERE airline\_staff.user\_name = %s AND airline\_staff.airline\_name = ticket.airline\_name AND ticket.purchase\_date\_time BETWEEN DATE\_ADD(CURDATE(),INTERVAL -3 month) AND CURDATE() group by airline\_staff.airline\_name'

* View last three month’s earned revenue.

Staff View Top Destinations

query = 'SELECT airport.airport\_city FROM airline\_staff, ticket, flight, airport WHERE airline\_staff.user\_name = %s AND airline\_staff.airline\_name = ticket.airline\_name AND flight.flight\_num = ticket.flight\_num AND flight.arrival\_airport\_code = airport.airport\_code AND (flight.arrival\_date\_time BETWEEN DATE\_ADD(CURDATE(),INTERVAL -3 month) AND CURDATE())GROUP BY airport.airport\_city ORDER BY (count(airport.airport\_city)) DESC LIMIT 3'

* View last three month’s top destination

query1 = 'SELECT airport.airport\_city FROM airline\_staff, ticket, flight, airport WHERE airline\_staff.user\_name = %s AND airline\_staff.airline\_name = ticket.airline\_name AND flight.flight\_num = ticket.flight\_num AND flight.arrival\_airport\_code = airport.airport\_code AND (flight.arrival\_date\_time BETWEEN DATE\_ADD(CURDATE(),INTERVAL -1 year) AND CURDATE())GROUP BY airport.airport\_city ORDER BY (count(airport.airport\_city)) DESC LIMIT 3'

* View last year’s top destination

Staff View Customer List in Particular Flight

query = 'SELECT distinct customer.name, customer.email FROM ticket,flight,customer WHERE flight.airline\_name = %s AND flight.flight\_num = %s AND flight.flight\_num = ticket.flight\_num AND flight.departure\_date\_time = ticket.departure\_date\_time AND flight.departure\_date\_time = %s AND ticket.email = customer.email '

* View last year’s top destination

Customer Search Flight

*if* way == "one way":

query = "SELECT flight.airplane\_ID, flight.airline\_name, flight.flight\_num,flight.base\_price,flight.status FROM flight WHERE %s = SUBSTRING(departure\_date\_time,1,10) AND %s = departure\_airport\_code AND %s = arrival\_airport\_code"

cursor.execute(query, (depart\_date, depart\_airport, arrive\_airport))

flights = cursor.fetchall()

print(flights)

*elif* way == "return":

query = "SELECT flight.airplane\_ID, flight.airline\_name, flight.flight\_num,flight.base\_price,flight.status FROM flight WHERE %s = SUBSTRING(departure\_date\_time,1,10) AND %s = departure\_airport\_code AND %s = arrival\_airport\_code"

query2 = "SELECT flight.airplane\_ID, flight.airline\_name, flight.flight\_num,flight.base\_price,flight.status FROM flight WHERE %s = SUBSTRING(departure\_date\_time,1,10) AND %s = departure\_airport\_code AND %s = arrival\_airport\_code"

* Customer searches for a particular flight. If one way is chosen, the return-date is ignored.

Customer Purchase Ticket

flight\_q = 'SELECT \* FROM flight WHERE %s = flight\_num AND %s = SUBSTRING(departure\_date\_time,1,16)' # used substring to filter seconds

total\_seat\_q = "SELECT num\_of\_seats FROM airplane WHERE %s = airplane\_ID"

count\_q = "SELECT COUNT(\*) as count FROM ticket WHERE %s = flight\_num"

ticket\_q = 'INSERT INTO ticket (email, airplane\_ID, airline\_name, flight\_num, departure\_date\_time, sold\_price, card\_type, card\_num, name\_on\_card, expiration\_date,purchase\_date\_time) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,NOW())'

id\_q = 'SELECT ticket\_ID FROM ticket ORDER BY purchase\_date\_time DESC LIMIT 0,1'

purchase\_q = 'INSERT INTO purchase VALUES(%s,%s)'

* Check if flight exist
* Get total number of seats on that flight
* Check how many seats occupied
* Insert into ticket
* Complete the purchasing process

Customer Rate Flight

query = 'SELECT \* FROM ticket WHERE %s = email AND %s = flight\_num AND %s = ticket\_ID AND purchase\_date\_time <= CURRENT\_TIMESTAMP()'

insert\_q = 'INSERT INTO rate VALUES (%s, %s, %s, %s)'

* Check if flight exist
* Insert ratings and comment

Responsibilities

Both Long Phi and Tiffany Yan worked on the backend and frontend of the project. Long Phi implemented the use cases on the customer end and the login/register portions for both staff and customer. Tiffany Yan implemented the use cases on the staff end. Overall it was a 50-50 effort from both sides.