Use Cases

**1. View Public Info:**

The first page displays search functionality for anyone -> index.html

Functions

/search

query = "SELECT flight\_num, departure\_time, arrival\_time FROM flight"

if departure\_airport !='':

query+= " where departure\_airport = '%s'" %departure\_airport

if arrival\_airport !='' and departure\_airport =='':

query += " where arrival\_airport = '%s'" % arrival\_airport

elif arrival\_airport !='' :

query += " and arrival\_airport = '%s'" %arrival\_airport

if departure\_time != '' and (departure\_airport =='' and arrival\_airport ==''):

query += ' where departure\_time = "%s"' %departure\_time

elif departure\_time != '':

query += ' and departure\_time = "%s"' %departure\_time

**2. Register**

/registerAuth calls specific registration for account type

3 cases:

**/cus\_register**

ins = "INSERT INTO customer VALUES(\'{}\',\'{}\', md5(\'{}\'), \'{}\', \'{}\', \'{}\', \'{}\', \'{}\', \'{}\', \'{}\', \'{}\', \'{}\')"

cursor.execute(ins.format(username, name, password, building\_num, street, city, state, phone\_num, pass\_num,pass\_exp,pass\_country,dob))

**/staff\_register**

ins = "INSERT INTO airline\_staff VALUES(\'{}\', md5(\'{}\'), \'{}\',\'{}\', \'{}\', \'{}\')"

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

**/booking\_register**

ins = "INSERT INTO booking\_agent VALUES(\'{}\', md5(\'{}\'), \'{}\')"

cursor.execute(ins.format(username, password, booking\_ID))

**3. Login:**

**/ loginAuth**

Simple, checks db for existing user

query = "SELECT \* FROM user WHERE username = \'{}\' and password = md5(\'{}\')"

If existing user works call /home

/home processes login by user type

**/home**

3 cases:

**if account\_type == 'customer':**

query\_purchased\_flights = "SELECT \* FROM flight, purchases, ticket \

WHERE purchases.customer\_email = \'{}\' \

AND purchases.ticket\_id = ticket.ticket\_id \

AND ticket.flight\_num = flight.flight\_num and flight.status ='upcoming' "

query\_all\_flights= "SELECT \* FROM flight WHERE flight.status = 'upcoming'"

then renders 'user\_home\_page.html'

**if account\_type == 'booking\_agent':**

query2 = "SELECT \* FROM flight, purchases, booking\_agent, ticket WHERE booking\_agent.email = \'{}\' AND purchases.booking\_agent\_id = booking\_agent.booking\_agent\_id AND purchases.ticket\_id = ticket.ticket\_id and flight.flight\_num = ticket.flight\_num"

query\_all\_flights= "SELECT \* FROM flight WHERE flight.status = 'upcoming'"

query = "SELECT booking\_agent\_id FROM booking\_agent WHERE email = '%s'" % username

then renders 'booking\_home\_page.html'

**if account\_type == 'airline\_staff':**

query = "SELECT airline\_name FROM airline\_staff WHERE username = '%s'" % username

query2 = "SELECT \* from flight where airline\_name = \'{}\' and status ='upcoming' and departure\_time<= \'{}\'"

then renders 'staff\_home\_page.html'

**Customer use cases:**

**4. View My flights:**

**/home**

Account\_type = “customers”

query = "SELECT \* FROM flight where flight.status != '' "

if flag == 'my':

query= "SELECT \* FROM flight, purchases, ticket \

WHERE purchases.customer\_email = '%s' \

AND purchases.ticket\_id = ticket.ticket\_id \

AND ticket.flight\_num = flight.flight\_num " %session['username']

template = 'search\_purchased.html'

else:

template ='search\_results.html'

if departure\_airport !='':

query+= " and flight.departure\_airport = '%s'" %departure\_airport

if arrival\_airport !='' :

query += " and flight.arrival\_airport = '%s'" %arrival\_airport

if departure\_time != '':

query += ' and flight.departure\_time = "%s"' %departure\_time

renders either search\_purchased.html or search\_results.html

**5. Purchase tickets:**

After search customer has the option to purchase

**/purchase\_flight**

If account is customer:

This renders purchase\_flight.html

When the customer selects a flight to purchase, they are taken to a confirmation page

On confirmation **/insert\_purchase** is called

#first need to get ticket id from flight number

cursor = conn.cursor();

query = "SELECT \* FROM ticket WHERE ticket.flight\_num = \'{}\'"

ins = "INSERT INTO purchases VALUES(\'{}\',\'{}\',NULL,\'{}\')"

**6. Search for flights**

**/user\_search**

if flag == 'my':

query= "SELECT \* FROM flight, purchases, ticket \

WHERE purchases.customer\_email = '%s' \

AND purchases.ticket\_id = ticket.ticket\_id \

AND ticket.flight\_num = flight.flight\_num " %session['username']

template = 'search\_purchased.html'

else:

template ='search\_results.html'

if departure\_airport !='':

query+= " and flight.departure\_airport = '%s'" %departure\_airport

if arrival\_airport !='' :

query += " and flight.arrival\_airport = '%s'" %arrival\_airport

if departure\_time != '':

query += ' and flight.departure\_time = "%s"' %departure\_time

**7. Track My Spending**

**/track\_spending**

#query for total spent over last year

query1 = "SELECT SUM(f.price) FROM flight f JOIN ticket t ON f.flight\_num = t.flight\_num JOIN purchases p ON t.ticket\_id = p.ticket\_id WHERE p.customer\_email = \'{}\' AND purchase\_date >= \'{}\' AND purchase\_date <= \'{}\'"

cursor.execute(query1.format(username, year\_ago, todays\_date))

#query for amount spent per month (last 6 months)

query2 = "SELECT DATE\_FORMAT(p.purchase\_date, '%Y-%m-01'), SUM(f.price) FROM flight f JOIN ticket t ON f.flight\_num = t.flight\_num JOIN purchases p ON t.ticket\_id = p.ticket\_id WHERE p.customer\_email = \'{}\' AND purchase\_date >= \'{}\' AND purchase\_date <= \'{}\' GROUP BY DATE\_FORMAT(p.purchase\_date, '%Y-%m-01')"

cursor.execute(query2.format(username, six\_months\_ago, todays\_date))

Then renders spending.html

**8. Logout**

@app.route('/logout')

def logout():

session.pop('username')

return redirect('/')

**Booking agent use cases:**

**4. View My flights**

Goes to /Home first

If account\_type = booking\_agent

query2 = "SELECT \* FROM flight, purchases, booking\_agent, ticket WHERE booking\_agent.email = \'{}\' AND purchases.booking\_agent\_id = booking\_agent.booking\_agent\_id AND purchases.ticket\_id = ticket.ticket\_id and flight.flight\_num = ticket.flight\_num"

query\_all\_flights= "SELECT \* FROM flight WHERE flight.status = 'upcoming'"

query = "SELECT booking\_agent\_id FROM booking\_agent WHERE email = '%s'" % username

then logs in and renders booking agent home page

**5. Purchase tickets**

After search booking agent has the option to purchase, **/purchase\_flight** is called

If account\_type == ‘booking\_agent’

page\_to\_render = 'agent\_purchase\_flight.html'

Once at this page they must confirm purchase

**/agent\_confirm\_purchase**

Booking agent has the option to buy for existing customer or a new customer

Existing customer: page\_to\_render = 'agent\_purchase\_confirm.html'

New customer: page\_to\_render = 'agent\_create\_customer.html'

Both pages can call

**/agent\_insert\_purchase**

query1 = "SELECT ticket\_id FROM ticket WHERE ticket.flight\_num = '%s'" %int(flight\_num)

agent\_data = 'select booking\_agent\_id from booking\_agent where email = "%s"' %username

ins = "INSERT INTO purchases VALUES(\'{}\',\'{}\',\'{}\',\'{}\')"

After insertion into purchases return home

**6. Search for flights**

**/agent\_search**

if flag == "my":

query = "SELECT \* from flight, purchases, booking\_agent, ticket WHERE booking\_agent.email = '%s' and booking\_agent.booking\_agent\_id = purchases.booking\_agent\_id and flight.flight\_num = ticket.flight\_num and purchases.ticket\_id = ticket.ticket\_id" %session['username']

page\_to\_render = 'agent\_search\_results.html'

else:

query = "SELECT \* FROM flight"

if departure\_airport =='' and arrival\_airport =='' and departure\_time =='' and max\_date=='' and min\_date=='':

query +=' where flight.status = "upcoming"'

else:

query +=' where flight.status != ""'

page\_to\_render = 'agent\_search\_results\_all.html'

if departure\_airport !='':

query+= " and flight.departure\_airport = '%s'" %departure\_airport

if arrival\_airport !='' :

query += " and flight.arrival\_airport = '%s'" %arrival\_airport

if departure\_time != '':

query += ' and flight.departure\_time = "%s"' %departure\_time

if max\_date !='' and min\_date!='':

query += ' and (flight.departure\_time >= "%s"' %min\_date

query += ' and flight.departure\_time <= "%s)"'%max\_date

elif max\_date !='' and min\_date == '':

query += ' and flight.departure\_time <= "%s"'%max\_date

elif min\_date !=''and max\_date == '':

**7. View my commission**

**/earnings**

query1 = "SELECT Count(price) FROM flight f JOIN ticket t ON f.flight\_num = t.flight\_num JOIN purchases p ON t.ticket\_id = p.ticket\_id WHERE booking\_agent\_id = \'{}\' AND p.purchase\_date >= \'{}\' AND p.purchase\_date <= \'{}\'"

query2 = "SELECT sum(price) FROM flight f JOIN ticket t ON f.flight\_num = t.flight\_num JOIN purchases p ON t.ticket\_id = p.ticket\_id WHERE booking\_agent\_id = \'{}\' AND p.purchase\_date >= \'{}\' AND p.purchase\_date <= \'{}\' "

query3 = "SELECT avg(price) FROM flight f JOIN ticket t ON f.flight\_num = t.flight\_num JOIN purchases p ON t.ticket\_id = p.ticket\_id WHERE booking\_agent\_id = \'{}\' AND p.purchase\_date >= \'{}\' AND p.purchase\_date <= \'{}\'"

Then renders agent\_commission.html

**8. View Top Customers**

This case is done in two parts

**/top\_customers\_number**

query = "SELECT customer\_email, COUNT(p.ticket\_id) FROM flight f JOIN ticket t ON f.flight\_num = t.flight\_num JOIN purchases p ON t.ticket\_id = p.ticket\_id WHERE booking\_agent\_id = \'{}\' AND p.purchase\_date >= \'{}\' GROUP BY customer\_email ORDER BY COUNT(t.ticket\_id) desc LIMIT 5"

renders top\_customers2.html

**/top\_customers\_money**

query2 = "SELECT customer\_email, sum(price) \*0.1 FROM flight f JOIN ticket t ON f.flight\_num = t.flight\_num JOIN purchases p ON t.ticket\_id = p.ticket\_id WHERE booking\_agent\_id = \'{}\' AND p.purchase\_date >= \'{}\' GROUP BY customer\_email ORDER BY sum(price) desc LIMIT 5"

renders top\_customers\_money.html

**Airline Staff use cases:**

**4. View My flights:**

**/staff\_search'**

query = "SELECT \* FROM flight where airline\_name='%s'" %session['airline\_name']

if departure\_airport !='':

query+= " and departure\_airport = '%s'" %departure\_airport

if arrival\_airport !='' :

query += " and arrival\_airport = '%s'" %arrival\_airport

if departure\_time != '':

query += ' and departure\_time = "%s"' %departure\_time

if max\_date !='' and min\_date!='':

query += ' and (departure\_time >= "%s"' %min\_date

query += ' and departure\_time <= "%s)"'%max\_date

elif max\_date !='' and min\_date == '':

query += ' and departure\_time <= "%s"'%max\_date

elif min\_date !=''and max\_date == '':

query += ' and departure\_time >= "%s"' %min\_date

**5. Create new flights:**

From staff homepage they call

**/create\_new\_flight**

Checks their airline name

query1 = "SELECT a.airline\_name FROM airline\_staff a WHERE a.username = \'{}\'"

renders create\_new\_flight.html for form input

Once details about the flight are filled

Call **/insert\_new\_flight**

This checks the staffs airline name with the flight airline name for authentication

ins = "INSERT INTO flight VALUES(\'{}\', \'{}\',\'{}\', \'{}\', \'{}\', \'{}\', \'{}\', \'{}\', \'{}\')"

After insert into flight query for all flights upcoming

query1 = "SELECT \* FROM flight WHERE flight.airline\_name = \'{}\'"

and display results on page staff\_airline\_flights.html

**6. Change Status of flights**

**7. Add airplane in the system:**

**/create\_new\_airplane**

Checks their airline name

query1 = "SELECT a.airline\_name FROM airline\_staff a WHERE a.username = \'{}\'"

renders create\_new\_airplane.html

Once here they fill out details. If airline name doesnt match error is displayed and they are unable to insert new airplane

If they have authentication then

ins = "INSERT INTO airplane VALUES(\'{}\', \'{}\', \'{}\')"

query to display results of airline airplanes

query1 = "SELECT \* FROM airplane WHERE airplane.airline\_name = \'{}\'"

render results at staff\_airline\_airplanes.html

**8. Add new airport in the system**

From staff home call /create\_new\_airport

Renders create\_new\_airport.html

Once input forms filled call /insert\_new\_airport

ins = "INSERT INTO airport VALUES(\'{}\', \'{}\')"

**9. View all the booking agents**

**10. View frequent customers**

**11. View reports**

From staff home page call / view\_report

query1 = "SELECT DATE\_FORMAT(purchase\_date, '%Y-%m-01'), Count(t.ticket\_id) FROM flight f JOIN ticket t ON f.flight\_num = t.flight\_num JOIN purchases p ON t.ticket\_id = p.ticket\_id GROUP BY month(purchase\_date)"

renders view\_report.html

From here they can specify a range of dates and then call **/search\_report**

query1 = "SELECT DATE\_FORMAT(purchase\_date, '%Y-%m-01'), Count(t.ticket\_id) FROM flight f JOIN ticket t ON f.flight\_num = t.flight\_num JOIN purchases p ON t.ticket\_id = p.ticket\_id WHERE p.purchase\_date >= \'{}\' AND p.purchase\_date <= \'{}\' GROUP BY month(purchase\_date)"

cursor.execute(query1.format(min\_datetime\_object,max\_datetime\_object))

renders results at search\_report\_results.html

**12. Comparison of Revenue earned**

From staff home page call **/** **comparison\_revenue**

Querys for total revenue with booking agent and without booking agent, feed into pie chart

query1 = "SELECT SUM(price) FROM flight f JOIN ticket t ON f.flight\_num = t.flight\_num JOIN purchases p ON t.ticket\_id = p.ticket\_id WHERE booking\_agent\_id IS NULL"

query2 = "SELECT SUM(price) FROM flight f JOIN ticket t ON f.flight\_num = t.flight\_num JOIN purchases p ON t.ticket\_id = p.ticket\_id WHERE booking\_agent\_id IS NOT NULL"

renders chart\_no\_booking\_agent.html

**13. View Top destinations**

From staff home page call **/view\_top\_destination**

#top destinations of last 3 months

query1 = "SELECT arrival\_airport, COUNT(t.ticket\_id) FROM flight f JOIN ticket t ON f.flight\_num = t.flight\_num JOIN purchases p ON t.ticket\_id = p.ticket\_id AND p.purchase\_date >= \'{}\' GROUP BY arrival\_airport ORDER BY COUNT(t.ticket\_id) desc LIMIT 3"

cursor.execute(query1.format(three\_month\_ago))

renders top\_destination.html

From here they can get a chart from the last year by following a link to **/top\_year\_des**

query1 = "SELECT arrival\_airport, COUNT(t.ticket\_id) FROM flight f JOIN ticket t ON f.flight\_num = t.flight\_num JOIN purchases p ON t.ticket\_id = p.ticket\_id AND p.purchase\_date >= \'{}\' GROUP BY arrival\_airport ORDER BY COUNT(t.ticket\_id) desc LIMIT 3"

cursor.execute(query1.format(year\_ago))

renders top\_year.html