

## **MAIN:**

`about.html`: the about page of our website

`layout.html`: the layout setting for the home page, before login

`index.html[home()]`: the homepage before users logging in. Show a google map and public search functions. Provide link to login, register, and about page.

`preregister.html[preregister()]`: for user to choose a type of registration(customer/airline staff/booking agent)

`prelogin.html[prelogin()]`: for user to choose a type of login(customer/airline staff/booking agent)

`search.html[search()]`: for users not logged in to search for upcoming flights based on a range of dates and destination/city or source airport/city

`status.html[status()]`: for users not logged in to check flight status based on a range of dates and destination/city or source airport/city

`app.py`: generate the entire project

`main.css`: file for generating font, word size, color, title, content, heading, image settings

## **CUSTOMER:**

`registercus.html[customerreg()]`: for users to register for a customer account, need to type in some information. report error message if information is not in the correct format.

`logincus.html[customerlogin()]`: the login page for customers, will check with the database to see if the information required is correct. If correct, then show the page after login. If not, then show the 'wrong information' message.

`layoutcus.html`: after customer login, the layout setting for all subsequent web pages

`homecus.html[cushome()]`: check if customer is logged in. If yes, direct to the staff home page. If not, direct back to the prelogin page.

`profilecus.html`: after customer login, for the customer to see his/her profile information

`accountcus.html[accountcus()]`: after customer login, for the customer to track spending — see his/her total amount of money spent in the past year and a bar chart showing month wise money spent for last 6 months. Additionally, the customer can specify a range of dates to view total amount of money spent and a bar chart showing month wise spending.

`bookcus.html[bookcus()]`: after customer login, for the customer to search flights and book flights based on his/her specification of a range of dates and destination/city or source airport/city

`bookcusconfirm.html[bookcusconfirm()]`: after customer login and book a flight, for the database to check if there are still seats available. If yes, load in the information and generate a ticket\_id and display it to the customer, telling the customer successful purchase.

`statuscus.html[statuscus()]`: after customer login, for the customer to search for flights by specifying a range of dates and destination/city or source airport/city

`tripscus.html[tripscus()]`: after customer login, for the customer to see his/her upcoming flights and can specify range of dates, destination/city and source airport/city to search for flights.

## **BOOKING AGENT:**

`registerbook.html[bookreg()]`: for users to register for a booking agent account, need to type in some information. Report error message if information is not in the correct format.

`loginbook.html[booklogin()]`: the login page for booking agents, will check with the database to see if the information required is correct. If correct, then show the page after login. If not, then show the 'wrong information' message.

`layoutbook.html`: after booking agent login, the layout setting for all subsequent web pages

`homebook.html[bookhome()]`: check if booking agent is logged in. If yes, direct to the staff home page. If not, direct back to the prelogin page.

`profileagent.html`: after booking agent login, for the agent to see his/her profile information

`bookagent.html[bookagent()]`: after booking agent login, for the agent to search flights and book flights for certain customer based on his/her specification of a range of dates and destination/city or source airport/city

`bookagentconfirm.html[bookagentconfirm()]`: after booking agent login and book a flight for a customer, for the database to check if there are still seats available. If yes, load in the information and generate a ticket\_id and display it to the customer, telling the customer successful purchase.

`commissionagent.html[commissionagent()]`: after booking agent login, for the agent to see the total commission received and the avg commission per ticket in

the past 30 days. Can specify a range of dates to see the total amount of commission received and number of tickets sold.

`statusagent.html[statusagent()]`: after booking agent login, for the agent to search for flights by specifying a range of dates and destination/city or source airport/city

`topcus.html[topcus()]`: after booking agent login, for the agent to see the top 5 customers who bought the most tickets in the past 6 months and the top 5 who gave the most commission last year. A bar chart showing such information is included.

`tripsagent.html[tripsagent()]`: after booking agent login, for the agent to see the upcoming flights he/she has bought for his/her customers. Can specify a range of dates and destination/city or source airport/city to see the flights he/she has purchased for customers.

## **AIRLINE STAFF:**

`registerstaff.html`: for users to register for an airline staff account, need to type in some information. Report error message if information is not in the correct format.

`loginstaff.html[stafflogin()]`: the login page for airline staffs, will check with the database to see if the information required is correct. If correct, then show the page after login. If not, then show the 'wrong information' message.

`layoutstaff.html`: after airline staff login, the layout setting for all subsequent web pages

`homestaff.html[staffhome()]`: check if staff is logged in. If yes, direct to the staff home page. If not, direct back to the prelogin page.

`profilestaff.html`: after airline staff login, for the staff to see his/her profile information

`addairport.html[addairport()]`: after airline staff login, for the staff to add a new airport by entering its name and the city name

`addplane.html[addplane()]`: after airline staff login, for the staff to add a new plane for his/her airline by entering the airplane id and seats capacity

`createflights.html[createflights()]`: after airline staff login, for the staff to create flights for his/her airline

`revenue.html[revenue()]`: after airline staff login, for the staff to see the indirect and direct sales in the last month & last year

`statuschange.html[statuschange()]`: after airline staff login, for the staff to change the status of the flights in his/her airline

`topdest.html[topdest()]`: after airline staff login, for the staff to see the 3 most popular destinations for last 3 months and last year

`tripsstaff.html[tripsstaff()]`: after airline staff login, for the staff to see the flights of the airline he/she works for

`viewagents.html[viewagents()]`: after airline staff login, for the staff to see the top 5 booking agents based on number of sales and the top 5 agents based on commission received for the last year

`viewcus.html[viewcus()]`: after airline staff login, for the staff to see the most frequent customer within the last year and the list of flights customers have taken for his/her airline

`viewreports.html[viewreports()]`: after airline staff login, for the staff to see the total amounts of ticket sold in a certain period of time in bar chart