

Databases Project Report

Air Ticket Reservation System

Yueyao Chen (yc7009@nyu.edu)

Yufei Wu (yw6945@nyu.edu)

December 8, 2025

Contents

1	SQL Query Explanation for <code>new_app.py</code>	3
1.1	User Authentication and Registration	3
1.1.1	Login (<code>/loginAuth</code>)	3
1.1.2	Registration (<code>/registerAuth</code>)	3
1.2	Public User Features	4
1.2.1	Flight Search (<code>/search_flights</code>)	4
1.2.2	Flight Status Lookup (<code>/check_flight_status</code>)	4
1.3	Customer Features	4
1.3.1	Purchase Authorization (<code>/purchase_authorization_customer</code>)	4
1.3.2	Purchase Ticket (<code>/purchase_ticket</code>)	5
1.3.3	View Purchased Flights (<code>/view_customer_flights</code>)	5
1.3.4	Spending Analytics (<code>/customer_spending</code>)	5
1.4	Booking Agent Features	5
1.4.1	Purchase Authorization (<code>/purchase_authorization_agent</code>)	5
1.4.2	Purchase Ticket (<code>/purchase_ticket_agent</code>)	5
1.4.3	View Agent Flights (<code>/view_agent_flights</code>)	5
1.4.4	Commission Report (<code>/agent_commission</code>)	6
1.4.5	Top Customers (<code>/view_top_customers</code>)	6
1.5	Airline Staff Features	6
1.5.1	View Flights (<code>/staff_view_flights</code>)	6
1.5.2	View Customers on a Flight (<code>/staff_view_customers</code>)	6
1.5.3	Admin Tools	6
1.5.4	Operator Tools	6
1.5.5	Analytics Tools	6
2	HTML Template Explanation	6
2.1	Backend Code	7
2.1.1	<code>new_app.py</code>	7
2.2	Base Layout	7

2.2.1	<code>base.html</code>	7
2.3	Public Pages	7
2.3.1	<code>index.html</code>	7
2.3.2	<code>login.html</code>	7
2.3.3	<code>register.html</code>	7
2.3.4	<code>search_flights.html</code>	7
2.3.5	<code>check_flight_status.html</code>	8
2.4	Customer Pages	8
2.4.1	<code>customer_home.html</code>	8
2.4.2	<code>view_customer_flights.html</code>	8
2.4.3	<code>customer_spending.html</code>	8
2.4.4	<code>purchase_authorization_customer.html</code>	8
2.5	Booking Agent Pages	9
2.5.1	<code>agent_home.html</code>	9
2.5.2	<code>view_agent_flights.html</code>	9
2.5.3	<code>agent_commission.html</code>	9
2.5.4	<code>agent_top_customer.html</code>	9
2.5.5	<code>purchase_authorization_agent.html</code>	9
2.6	Airline Staff Pages	10
2.6.1	<code>staff_home.html</code>	10
2.6.2	<code>staff_view_flights.html</code>	10
2.6.3	<code>staff_view_customers.html</code>	10
2.6.4	<code>staff_view_agents.html</code>	10
2.6.5	<code>staff_add_agent.html</code>	10
2.6.6	<code>staff_add_airplane.html</code>	11
2.6.7	<code>staff_add_airport.html</code>	11
2.6.8	<code>staff_create_flight.html</code>	11
2.6.9	<code>staff_change_status.html</code>	11
2.6.10	<code>staff_frequent_customers.html</code>	11
2.6.11	<code>staff_reports.html</code>	12
2.6.12	<code>staff_delay_vs_on_time.html</code>	12
2.6.13	<code>staff_top_destinations.html</code>	12
2.6.14	<code>staff_grant_permission.html</code>	12
3	Mapping of Templates to Backend Functions	13

1 SQL Query Explanation for new_app.py

This section documents every user-facing feature and the SQL statements executed by the backend. Each explanation describes the purpose of the query and how it contributes to the system's functionality.

1.1 User Authentication and Registration

1.1.1 Login (/loginAuth)

To authenticate a user, the system performs role-specific credential checks and loads permissions when necessary.

Customer Login

```
SELECT * FROM customer WHERE email = %s AND password = %s;
```

Explanation: Validates customer credentials with hashed passwords.

Booking Agent Login

```
SELECT * FROM booking_agent WHERE email = %s AND password = %s;
```

Explanation: Identifies the booking agent and initializes the session.

Airline Staff Login

```
SELECT * FROM airline_staff WHERE username = %s AND password = %s;
```

Explanation: Authenticates airline staff using a username-based login.

Fetch Staff Permissions

```
SELECT permission_type  
FROM permission  
WHERE username = %s;
```

Explanation: Retrieves admin/operator roles for staff control-panel authorization.

1.1.2 Registration (/registerAuth)

Checks uniqueness of accounts and inserts new user entries.

Check for Existing User

```
SELECT * FROM customer WHERE email = %s;  
SELECT * FROM booking_agent WHERE email = %s;  
SELECT * FROM airline_staff WHERE username = %s;
```

Explanation: Prevents duplicate registrations.

Check Airline Existence (Staff)

```
SELECT * FROM airline WHERE airline_name = %s;
```

Explanation: Ensures staff register only under valid airlines.

Insert New User

```
INSERT INTO customer (...) VALUES (...);  
INSERT INTO booking_agent (...) VALUES (...);  
INSERT INTO airline_staff (...) VALUES (...);
```

Explanation: Inserts new users into the correct tables.

1.2 Public User Features

1.2.1 Flight Search (/search_flights)

Supports flexible searches with airport codes, cities, or aliases.

Resolve Aliases

```
SELECT airport_name FROM city_airport_map WHERE LOWER(city_alias)=LOWER(%s);  
SELECT airport_name FROM airport WHERE LOWER(airport_name)=LOWER(%s);  
SELECT airport_name FROM airport WHERE LOWER(airport_city)=LOWER(%s);
```

Search Flights

```
SELECT f.*  
FROM flight f  
WHERE FIND_IN_SET(f.departure_airport, %s)  
    AND FIND_IN_SET(f.arrival_airport, %s)  
    AND DATE(f.departure_time) BETWEEN %s AND %s;
```

1.2.2 Flight Status Lookup (/check_flight_status)

```
SELECT airline_name, flight_num, departure_airport, arrival_airport,  
      departure_time, arrival_time, status  
FROM flight  
WHERE airline_name=%s AND flight_num=%s;
```

1.3 Customer Features

1.3.1 Purchase Authorization (/purchase_authorization_customer)

Fetch Flight

```
SELECT ...  
FROM flight  
WHERE airline_name=%s AND flight_num=%s;
```

Check Duplicate Purchase

```
SELECT COUNT(*)
FROM ticket t JOIN purchases p ON t.ticket_id=p.ticket_id
WHERE t.airline_name=%s AND t.flight_num=%s
AND p.customer_email=%s;
```

1.3.2 Purchase Ticket (/purchase_ticket)

Includes password revalidation, capacity check, incrementing ticket IDs, and inserting purchases.

Password Validation

```
SELECT password FROM customer WHERE email=%s;
```

Seat Capacity

```
SELECT a.seats
FROM flight f JOIN airplane a ...
WHERE f.airline_name=%s AND f.flight_num=%s;
```

Insert Ticket + Purchase (omitted here for brevity; identical to earlier version)

1.3.3 View Purchased Flights (/view_customer_flights)

```
SELECT f.*
FROM flight f JOIN ticket t ... JOIN purchases p ...
WHERE p.customer_email=%s
ORDER BY f.departure_time ASC;
```

1.3.4 Spending Analytics (/customer_spending)

Queries compute total spending and month-by-month aggregates.

1.4 Booking Agent Features

1.4.1 Purchase Authorization (/purchase_authorization_agent)

Checks airline affiliation, retrieves flight, and checks duplicate purchase.

1.4.2 Purchase Ticket (/purchase_ticket_agent)

Validates password, ensures airline authorization, checks capacity, generates ticket, and inserts into purchases.

1.4.3 View Agent Flights (/view_agent_flights)

Joins `flight`, `ticket`, `purchases`, and `booking_agent` to show all sales made by the agent.

1.4.4 Commission Report (/agent_commission)

Computes total commission, average commission, and total ticket count.

1.4.5 Top Customers (/view_top_customers)

Two ranking queries: top 5 by tickets and top 5 by commission.

1.5 Airline Staff Features

1.5.1 View Flights (/staff_view_flights)

Shows flights filtered by date range, origin, destination, or time window.

1.5.2 View Customers on a Flight (/staff_view_customers)

Displays passenger list for a selected flight.

1.5.3 Admin Tools

Includes:

- Creating flights
- Adding airplanes
- Adding airports
- Granting permissions
- Adding booking agents to the airline

1.5.4 Operator Tools

- Updating flight status
- Delay vs. on-time analytics

1.5.5 Analytics Tools

- Monthly sales reports
- Frequent customers
- Top destinations
- Top booking agents

2 HTML Template Explanation

This section describes the purpose and functionality of each HTML template used in the Air Ticket Reservation System. All text below is included exactly as written in the project documentation.

2.1 Backend Code

2.1.1 new_app.py

This is the main backend server script for the entire system, built with Flask and MySQL. It handles user sessions, checks permissions (Admin/Operator), executes SQL queries safely, and renders the HTML templates described above. It is the core “controller” that connects frontend pages with the database and business logic.

2.2 Base Layout

2.2.1 base.html

This is the common layout template for all pages. It defines the site title “Air Ticket Reservation System”, shows any flashed messages (success or error alerts), and provides a `{% block content %}` area where each specific page inserts its own content. All other templates extend this file so they have a consistent look and message area.

2.3 Public Pages

2.3.1 index.html

This is the public landing page. It welcomes the user to the Air Ticket Reservation System and provides simple navigation links/buttons to Login, Register, Search Flights (public), and Check Flight Status (public). If a user is not logged in, they start here.

2.3.2 login.html

This page lets any user log in. The form asks the user to choose a role (Customer, Booking Agent, Airline Staff), and then enter Email/Username and Password. If the backend detects invalid credentials, it passes an error string and the template displays it. It also has a button to go back to the home (index) page.

2.3.3 register.html

This page lets new users register for an account. It first asks the user to choose a role (Customer, Booking Agent, Airline Staff). Then, depending on the role, it collects different information: for customers it asks for personal, contact, and passport details; for staff it asks for name, date of birth, and airline name. Flash messages from the backend show if registration was successful or if there were errors (duplicate email, invalid airline, etc.). There is a “Back to Home” link.

2.3.4 search_flights.html

This page is used for public flight search. The user enters Source Airport/City, Destination Airport/City, and a Start/End date. The backend runs a query and passes a list of flights. If flights exist, the page shows a results table with Flight Number, Airline, Departure Time, Arrival Time, and Price. The “Action” column changes by user role: logged in customers see a “Purchase” button; agents see a field for customer email and a “Purchase for Customer” button; others are asked to log in as customer or agent. There is a link back to the home page.

2.3.5 check_flight_status.html

This is the public page to check the status of a specific flight. The user types the Airline Name and Flight Number, then clicks “Check Status”. If there is an error (missing input or flight not found), the backend passes an error message and the template shows it. If a matching flight is found, the page displays a “Flight Status Result” section with flight number, airline, route (departure → arrival), departure time, arrival time, and current status. There is also a “Back to Home” link.

2.4 Customer Pages

2.4.1 customer_home.html

This is the home dashboard for a logged-in customer. It greets the customer using session[‘email’] and shows simple navigation options: Search Flights, View My Flights, Track My Spending, and Logout. Each button links to the corresponding customer function in new_app.py.

2.4.2 view_customer_flights.html

This page lets customers see their own tickets and flights. At the top, there are filters: Show (Upcoming, All, Past), Start Date, End Date, Origin, and Destination, plus Apply/Reset buttons. The backend provides a flights list based on these filters. If there are no flights, it shows “No flights found.” If there are flights, it shows a “My Flights List” table with Flight Number, Airline, formatted Departure Time and Arrival Time (or “N/A” if missing), and Status. There is a “Back to Home” link for navigation.

2.4.3 customer_spending.html

This page shows a customer’s spending report. It has a small form with Start Date, End Date, and an “Update Report” button. The backend calculates total_spent and monthly_spending (month + total amount). The page displays the total spending (formatted to two decimals) and, if there is data, a table with Month and Amount Spent. If there is no spending in that range, it shows a message. If the backend generates a chart image (base64), the template shows a “Spending Chart” section where the bar chart can be displayed. A “Back to Home” link returns to the customer dashboard.

2.4.4 purchase_authorization_customer.html

This page is a confirmation step when a customer buys a ticket. The backend first fetches the flight details and checks if the customer has already purchased this flight. If yes, the page shows a warning message with a (**Warning Icon**). The main part of the page shows a “Flight Summary” with airline, flight number, from/to airports, departure and arrival times, price, and status. The customer must re-enter their account password to confirm the purchase for security. Buttons allow them to “Cancel and Back to Search” or “Confirm Purchase”.

2.5 Booking Agent Pages

2.5.1 agent_home.html

This is the home dashboard for a logged-in booking agent. It greets the agent (Booking Agent + email) and shows buttons to Search Flights, View Customer Flights (flights they have sold), View My Commission, View Top Customers, and Logout. Each button links to the corresponding agent routes in new_app.py.

2.5.2 view_agent_flights.html

This page shows all flights sold by the current booking agent. There is a filter form with Start Date, End Date, Source Airport/City, and Destination Airport/City, plus a “Filter Flights” button. The backend returns flights that match these filters. If there are flights, the page shows a “Results” table with Flight Number, Airline, Departure Time, Arrival Time, Source, Destination, and Status. If no flights match, it displays “No flights found for your criteria.” A “Back to Home” link returns to the agent dashboard.

2.5.3 agent_commission.html

This page shows a commission report for the current agent. It includes fields for Start Date and End Date and a “Filter” button. The backend calculates the total commission earned, average commission per ticket, and total tickets sold within the selected period, and passes them to the page along with the chosen date range. The template displays a summary sentence “From start_date to end_date” and then the three numbers. There is also a “Back to Home” link.

2.5.4 agent_top_customer.html

This page displays two “Top 5 customers” lists for the agent. The first section shows the “Top 5 Customers by Tickets Purchased (Last 6 Months)” with a table of email and ticket count, or a message if no data. The second section shows the “Top 5 Customers by Commission Earned (Last Year)” with a table of email and total commission (CNY), or a message if there is no data. The backend can also generate bar charts and send them to this page for visual analytics. A “Back to Home” link takes the agent back to their home dashboard.

2.5.5 purchase_authorization_agent.html

This is the agent version of the purchase confirmation page. The backend loads the selected flight and checks whether the target customer has already bought that flight; if yes, the page shows a warning line. The template then shows a “Flight Summary” (airline, flight number, from/to airports, times, price, status) and a “Customer” section with the customer’s email. The agent must confirm by entering their own agent account password before completing the purchase. There are buttons to cancel and go back to search or to confirm the purchase.

2.6 Airline Staff Pages

2.6.1 staff_home.html

This is the main dashboard for airline staff. It greets the staff with their email and prints out their permissions list. If the permissions include “Admin”, the page shows extra admin actions: Add Booking Agent, Add New Airplane, Add New Airport, Create New Flight, Grant Permission, View All Booking Agents. If the permissions include “Operator”, it also shows “Change Flight Status”. All staff can see links to Search Flights, View Frequent Customers, View My Airline Flights, View Reports, View Delay vs On-time Comparison, View Top 3 Destinations, and Logout.

2.6.2 staff_view_flights.html

This page allows staff to search and view flights operated by their own airline. The form includes Start Date, End Date, Source Airport or City, Destination Airport or City, and a flight time filter (All, Future, Past). The backend decides whether it is the default view (next 30 days) or a custom search and passes `is_default_view`. The page shows either “Flights in next 30 days” or “Flights found”, plus “Total flights: N”. If a flight is not empty, it renders a table with Flight Number, Departure Airport, Departure Time, Arrival Airport, Arrival Time, Price, Status, Airplane ID, and an “Action” link “View Customers” for each flight. If no flights match, it shows a message. There is a “Back to Home” link.

2.6.3 staff_view_customers.html

This page shows the list of customers on a specific flight. It displays a title “Customers for Flight `flight_num`”. If the customer is not empty, it shows a simple table with Name and Email for each customer who purchased a ticket on that flight. If there are no customers, it displays “No customers found for this flight.” A “Back to Flights” link returns the staff to the flight list page.

2.6.4 staff_view_agents.html

This page shows the top booking agents for the staff’s airline. It prints a header “Top Booking Agents”, and, if the airline name is available, says “Showing booking agents performance for airline `airline_name`.” It then has four sections: (1) Top 5 by Tickets Sold (Past Month), (2) Top 5 by Tickets Sold (Past Year), (3) Top 5 by Commission Earned (Past Month), and (4) Top 5 by Commission Earned (Past Year). Each section displays a list of agent emails with ticket or commission values if there is data, or a “No data...” style message otherwise. There is a “Back to Home” link.

2.6.5 staff_add_agent.html

This page allows admin staff to connect an existing booking agent to their airline. It shows a simple form with “Agent Email” and a “Confirm” button. The backend checks whether the agent exists and whether they already work for this airline, then inserts a record if allowed and shows flash messages. A “Back to Home” link returns to the staff dashboard.

2.6.6 staff_add_airplane.html

This page lets admin staff add new airplanes for their airline and view all existing airplanes. When is_admin is true, a note explains the user is an Admin and can add airplanes for a certain airline (name shown if available). The form asks for Airplane ID and Number of Seats and has a “Create Airplane” button. Under “All Airplanes”, it displays a table listing Airplane ID and Seats for this airline; if no airplanes are found, it shows a message. A “Back to Home” link is provided.

2.6.7 staff_add_airport.html

This page allows admin staff to add new airports and see all airports in the system. At the top, if is_admin is true, it shows a line saying the user is logged in as an Admin and can add new airports for their airline (with the airline name). There is a small form with “Airport Name” and “Airport City” and a “Create Airport” button. Below, it lists “All Airports” with a table of City and Airport Name. If the airports list is empty, it shows “No airports found.” There is a “Back to Home” link.

2.6.8 staff_create_flight.html

This is the page where admin staff create new flights for their airline. The form fields include Flight Number, Airplane ID, Departure Airport, Departure Time (with a specified datetime format), Arrival Airport, Arrival Time, Price, and Status (options: upcoming, in progress, delayed, cancelled). When the form is submitted, the backend inserts a new flight record. Below the form, the page shows “Upcoming Flights (Next 30 Days)” with a table if flights exist: flight number, departure airport/time, arrival airport/time, a combined time range, status, and price. If there are no upcoming flights, it shows a message. There is a “Back to Home” link.

2.6.9 staff_change_status.html

This page lets staff with Operator permission update flight statuses. If the user is an operator, it displays a line saying they can update flight statuses for their airline. The page then shows “Select a Flight” and, if there are future flights, it lists each flight as “Flight flight_num — departure → arrival — Departs: time — Status: status”. If no future flights exist, it tells the user “No future flights available.” Under that, there is a “New Status” dropdown or options (Upcoming, In Progress, Delayed, Cancelled, Landed) and an “Update Status” button. A “Back to Staff Home” link returns to the staff dashboard.

2.6.10 staff_frequent_customers.html

This page shows frequent customers for the staff’s airline. The first section is “Most Frequent Customers (Past Year)”, where the backend passes top_customers; if not empty, the page lists customer Email and Tickets Purchased; otherwise, it shows a message. The second part is a form to “View Flights for a Customer (of Your Airline)” by entering a Customer Email. If customer_flights is provided, the page shows a table “Flights for email” listing Flight Number, Departure, Arrival, Times (departure → arrival), Status, and Price. There is a “Back to Home” link.

2.6.11 staff_reports.html

This page displays monthly ticket sales reports for the staff's airline. The header shows "Monthly Report from start_date to end_date". There are quick links/buttons for "Last 30 Days" and "Last Year", plus fields for Start Date and End Date and a "Filter" button. It shows the total number of tickets sold in that period. If there is monthly data (months not empty), it labels a section "Monthly Ticket Sales" (with the actual bar chart rendered via an `img` tag in practice); otherwise, it shows "No data available for the selected period." A "Back to Home" link is included.

2.6.12 staff_delay_vs_ontime.html

This page shows delay vs on-time statistics for the staff's airline. If `airline_name` is present, it shows which airline the data is for. It has two sections: "Last 30 Days" and "Last 1 Year". For each section, the template calculates the counts of delayed and on-time flights from the `last_month_stats` or `last_year_stats`. If the total is greater than zero, it prints the number of Delayed Flights and On-time Flights and can display a bar chart image (using `lastmonth_chart` or `lastyear_chart`). If there is no data, it shows a message saying there are no statistics for that period. There is a "Back to Home" link.

2.6.13 staff_top_destinations.html

This page lists the most popular destination cities for the staff's airline. The first part is "Top 3 Most Popular Destinations for Last 3 Months", where it shows a table of City and Number of Flights using `top_destinations_last3months`, or a message if no data. The second part is "Top 3 Most Popular Destinations for Last Year", again showing City and Number of Flights from `top_destinations_lastyear` or a message if empty. There is a "Back to Home" link at the bottom.

2.6.14 staff_grant_permission.html

This page lets a staff member grant permission to another staff user. It has a form with "Staff Username" and a dropdown "Permission Type" with options "Admin" or "Operator". When the form is submitted, the backend checks that the target staff exists, belongs to the same airline, and doesn't already have a permission, then inserts a new permission record. There is a "Confirm" button to apply and a "Back to Home" link.

3 Mapping of Templates to Backend Functions

Template	Backend Route
index.html	index()
login.html	login()
register.html	register()
search_flights.html	search_flights()
check_flight_status.html	check_flight_status()
customer_home.html	customer_home()
view_customer_flights.html	view_customer_flights()
customer_spending.html	customer_spending()
purchase_authorization_customer.html	purchase_authorization_customer()
agent_home.html	agent_home()
view_agent_flights.html	view_agent_flights()
agent_commission.html	agent_commission()
agent_top_customers.html	view_top_customers()
purchase_authorization_agent.html	purchase_authorization_agent()
staff_home.html	staff_home()
staff_view_flights.html	staff_view_flights()
staff_view_customers.html	staff_view_customers()
staff_add_airplane.html	staff_add_airplane()
staff_add_airport.html	staff_add_airport()
staff_create_flight.html	staff_create_flight()
staff_change_status.html	staff_change_status()
staff_frequent_customers.html	staff_frequent_customers()
staff_view_agents.html	staff_view_agents()
staff_top_destinations.html	staff_top_destinations()
staff_grant_permission.html	staff_grant_permission()
staff_add_agent.html	staff_add_agent()