

# FlightHub Documentation

This website is designed to facilitate the management and usage of an airline's services for both staff and customers. The platform offers tools for staff to manage flights, passengers, and revenue while giving customers the ability to search for flights, purchase tickets, and view their flight history.

## Core Features Include:

1. Flight search and booking
2. Customer and staff registration
3. Dashboards for managing flight and customer information
4. Customer and staff profile management
5. Flight statistics and feedback views

## CSS Files

- index.css  
Purpose: Styles the index.html page.  
Context of Use: Applied to the homepage (index.html), ensuring visual consistency and layout.
- page.css  
Purpose: Styles the customer\_page.html and staff\_page.html pages.  
Context of Use: Applied to both customer and staff login/register pages, providing shared visual elements across user types.
- staffdashboard.css  
Purpose: Styles the staffdashboard.html page.  
Context of Use: Used in the staff dashboard to ensure consistent layout and design for staff-specific features and tables.

---

## HTML Files

- `airlineFleet.html`

Purpose: Displays a table of all airplanes belonging to an airline.

Context of Use: Accessed via the “Fleet” button on `staffdashboard.html`, showing the airline’s fleet.

- `buyTicket.html`

Purpose: Provides a form for customers to complete their flight ticket purchase.

Context of Use: Accessed through the `index.html` page after a customer searches for flights and clicks “Purchase.”

- `clientdashboard.html`

Purpose: Displays a table of purchases made by the customer, including options to cancel or rate flights. Also shows a table of customer spending for the last year and the last six months.

Context of Use: Customer dashboard for viewing and managing purchases and spending. Button redirects to `customerProfile.html` to add a phone number.

- `customer_page.html`

Purpose: Provides login and registration forms for customers.

Context of Use: Accessed by customers for logging in or registering for an account.

- `customerProfile.html`

Purpose: Allows customers to add their phone number to their profile.

Context of Use: Accessed from the client dashboard to update the customer's phone number information.

- `customerSearch.html`

Purpose: Allows staff to search for a specific customer and view their history with the airline.

Context of Use: Contains a table showing all customers of an airline, with a search form for retrieving a customer’s flight history.

- index.html  
Purpose: Displays flight search fields and a navigation bar that changes based on login status.  
Context of Use: Accessible to both logged-in and non-logged-in users, with options to search for flights and choose a user role for login or registration.
- passengerList.html  
Purpose: Displays a table of all passengers on a particular flight.  
Context of Use: Accessed through staffdashboard.html by clicking the “Passenger Lists” button.
- staffdashboard.html  
Purpose: Staff dashboard providing tools to manage flights, passengers, airplanes, airports, maintenance schedules, and revenues.  
Context of Use: Contains various forms and tables for flight and revenue management, including buttons that redirect to other pages (e.g., airlineFleet.html, passengerList.html, customerSearch.html, viewstats.html, and staffProfile.html).
- staff\_page.html  
Purpose: Provides login and registration forms for staff members.  
Context of Use: Accessed by staff for logging in or registering for an account.
- staffProfile.html  
Purpose: Allows staff members to add email addresses and phone numbers to their account.  
Context of Use: Accessed from the staff dashboard to update staff member contact information.
- viewstats.html  
Purpose: Displays ratings and comments for flights, along with the average rating.  
Context of Use: Accessed through the “View” button on the staffdashboard.html page to see customer feedback on flights.

# Use Cases

## Use Case 1: Search for Flights (One-way and Round-trip)

ID: UC-001

Actor(s): Customer

Description: A customer can search for future flights based on source city/airport name, destination city/airport name, and departure or return dates (for round-trip).

Preconditions:

- The customer does not need to be logged in.

Postconditions:

- Available flights are displayed based on the search criteria.

Trigger:

- Customer initiates a search by providing flight details.

Basic Flow:

- Customer enters the source city/airport, destination city/airport, and departure date (for one-way) or return date (for round-trip).
- The system converts the airport names to airport codes to process flight searches
- The system displays the matching flight options to the customer.

Alternative Flows:

- Invalid Search Criteria: The system informs the customer that no flights match the search criteria.

Special Requirements: None

Priority: High

Frequency of Use: Frequent

---

## Use Case 2: User Registration (Customer and Staff)

ID: UC-002

Actor(s): Customer, Staff

Description: A user (either customer or staff) can register for an account with the system.

Preconditions:

- The user has not already registered.

Postconditions:

- A new user account is created.

Trigger:

- The user submits the registration form.

Basic Flow:

1. The user provides registration details (e.g., email, password, username, etc.).
2. For customers, the system checks if the email already exists
3. For staff, the system checks if the username already exists for the airline
4. If the check is successful, the system creates a new account.

Alternative Flows:

- Existing User: The system informs the user that the email or username already exists.

Special Requirements: MD5 hashing is used for password storage via `hash_password()` using `hashlib.md5()`.

Priority: High

Frequency of Use: Occasional

---

## Use Case 3: User Login (Customer and Staff)

ID: UC-003

Actor(s): Customer, Staff

Description: A registered user can log into the system.

Preconditions:

- The user must be registered.

Postconditions:

- The user is logged in, and session data is created.

Trigger:

- The user submits login credentials.

Basic Flow:

1. The customer provides email and password.
2. The system verifies the credentials with the database
3. If successful, the customer is logged in.
4. For staff, the system verifies airline, username, and password
5. If successful, the staff member is logged in.

Alternative Flows:

- Invalid Credentials: The system displays an error message.

Special Requirements: None

Priority: High

Frequency of Use: Frequent

---

## Use Case 4: View My Flights (Customer)

ID: UC-004

Actor(s): Customer

Description: A customer can view their past, current, and future flights.

Preconditions:

- The customer must be logged in.

Postconditions:

- The system displays the customer's flight history.

Trigger:

- The customer selects the option to view their flights.

Basic Flow:

1. The customer chooses to view future flights, past flights, or all flights.
2. The system displays the relevant flights.

Alternative Flows:

- No Flights Found: The system informs the customer that no flights match the criteria.

Special Requirements: None

Priority: High

Frequency of Use: Frequent

---

## Use Case 5: Purchase Tickets

ID: UC-005

Actor(s): Customer

Description: A customer can purchase tickets for a flight.

Preconditions:

- The customer must be logged in.

Postconditions:

- The ticket is purchased and recorded in the database.

Trigger:

- The customer selects a flight and proceeds to purchase a ticket.

Basic Flow:

1. The customer selects a flight to purchase a ticket.
2. The system checks if the ticket is available.
3. If available, the system inserts the purchase into the purchases table
4. The system confirms the purchase with the customer.

Alternative Flows:

- Ticket Unavailable: The system informs the customer that the ticket has already been purchased.

Special Requirements: None

Priority: High

Frequency of Use: Frequent