

ASSIGNMENT-9

DEBASHRITA MANDAL

CONTENT:

1. Project Overview
2. Features Implemented
 3. Frontend Screenshots
 - 1> Landing Page (User)
 - 2> Landing Page (Admin)
 - 3> Add Flight
 - 3.1>Wireframe
 - 4> Search Flights
 - 4.1> As user
 - 4.2> As Admin
 - 5> Book Flight
 - 6> Booking History
 - 7> Cancel Booking
 - 8> Profile
 - 8.1> As user
 - 8.2> As Admin
 - 9> Change Password
 - 10> Register
 - 11> Login
 - 12> Logout
 4. Postman Screenshots
 5. MySQL Screenshots
 6. Email Notification
 7. Eureka Dashboard
 8. Message Broker – RabbitMQ
 - 9.SonarQube
 10. Jacoco
 11. Jmeter
 12. Docker (implemented in the last version of this project's backend; running locally as of now)

PROJECT OVERVIEW:

- Developed a **responsive Angular frontend** for a flight booking and ticket management platform.
- Implemented **JWT-based authentication** with login and registration, storing session data securely in LocalStorage.
- Built core features including **flight search, ticket booking, booking history, and ticket cancellation**.
- Designed **role-based navigation and route protection** to redirect users and admins to appropriate views.
- Integrated **HTTP interceptors** to automatically attach Bearer tokens to secured API requests.

IMPLEMENTED FEATURES:

- Implemented **Add Flight** functionality allowing admins to create and manage flight inventory.
- Enforced **Role-Based Access Control (RBAC)** to restrict flight management operations to admin users only.
 1. Admin has been seeded in the database
 2. Admin role cant be self assigned
- Configured **environment-specific properties** using separate configuration files for Local and Docker deployments.
- Optimized **Dockerfiles** to improve image size, build time, and container startup performance.
- Implemented **Change Password** functionality with validation and secure credential handling.
 1. Added new api endpoint in the backend
 2. Password can only be changed by a logging in user
- Additional data:
 1. Implemented dropdown for all source and destinations
 2. Blocked past dates from date picker
 3. Arranged current booking by status (confirmed first, then cancelled)
 4. Admin can access search flight, but cant book them (select flight button wont be visible)
 5. Separated the navbar into its own component to reduce code duplication

FRONTEND SCREENSHOTS:

1. Landing Page (User)

Login page when the user is logged out (Only search flight visible)

The screenshot shows a web browser window with the URL 'localhost:4200'. The page has a light blue gradient background. At the top, there's a navigation bar with links to 'Home', 'Search Flights', 'Login', and 'Register'. The main heading 'Your Journey Starts Here' is displayed in large, bold, dark blue text. Below it, a subtext reads 'Discover the world with ease. Book flights to your dream destinations at unbeatable prices.' Two buttons are present: a white button with a blue outline containing a magnifying glass icon and the text 'Search Flights', and a white button with a blue outline containing the text 'Learn More'.

Why Choose FlightBooker?

Login page when the user is logged in (View Booking and Profile visible)

The screenshot shows a web browser window with the URL 'localhost:4200'. The page has a light blue gradient background. At the top, there's a navigation bar with links to 'Home', 'Search Flights', 'View Bookings', 'Profile', and 'Logout'. The main heading 'Your Journey Starts Here' is displayed in large, bold, dark blue text. Below it, a subtext reads 'Discover the world with ease. Book flights to your dream destinations at unbeatable prices.' Two buttons are present: a white button with a blue outline containing a magnifying glass icon and the text 'Search Flights', and a white button with a blue outline containing the text 'Learn More'.

2. Landing Page (Admin)

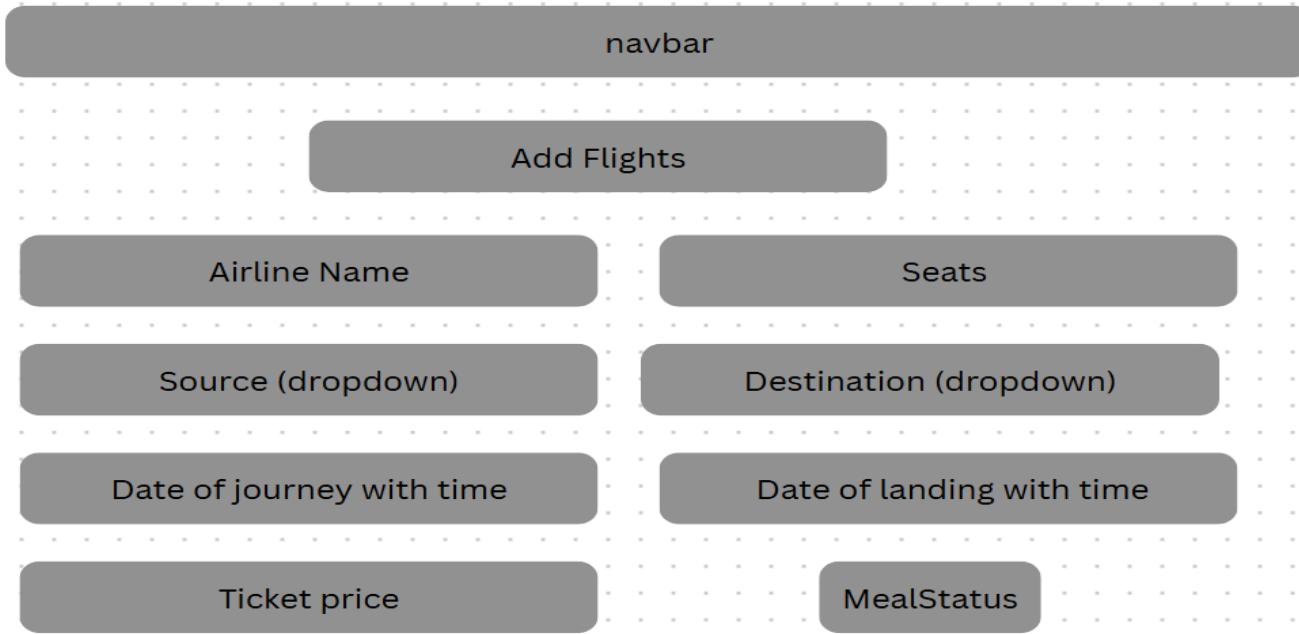
When the admin logs in: Add Flights visible

The screenshot shows a web browser window with the URL 'localhost:4200'. The page has a light blue header with the text 'Your Journey Starts Here' in large bold letters. Below it is a subtext: 'Discover the world with ease. Book flights to your dream destinations at unbeatable prices.' There are two buttons: 'Search Flights' and 'Learn More'. At the top of the page, there is a navigation bar with links like 'Home', 'Search Flights', 'Add Flight', 'Profile', and 'Logout'. The rest of the page is a blank white space.

Why Choose FlightBooker?

3. Add Flight Page

Wireframe for the add-flight page



Successful Booking

[Home](#) [Search Flights](#) [Add Flight](#)

[Profile](#) [Logout](#)



Add Flight to Inventory

Airline Name

Available Seats

Source

Destination

Departure Date & Time



Arrival Date & Time



Ticket Price

Meal Available

Add Flight

Flight added successfully

Unsuccessful booking: Input Validation with error messages

[Home](#) [Search Flights](#) [Add Flight](#)

[Profile](#) [Logout](#)

Add Flight to Inventory

Airline Name

Available Seats

Source

Destination



Departure Date & Time



Arrival Date & Time



Ticket Price

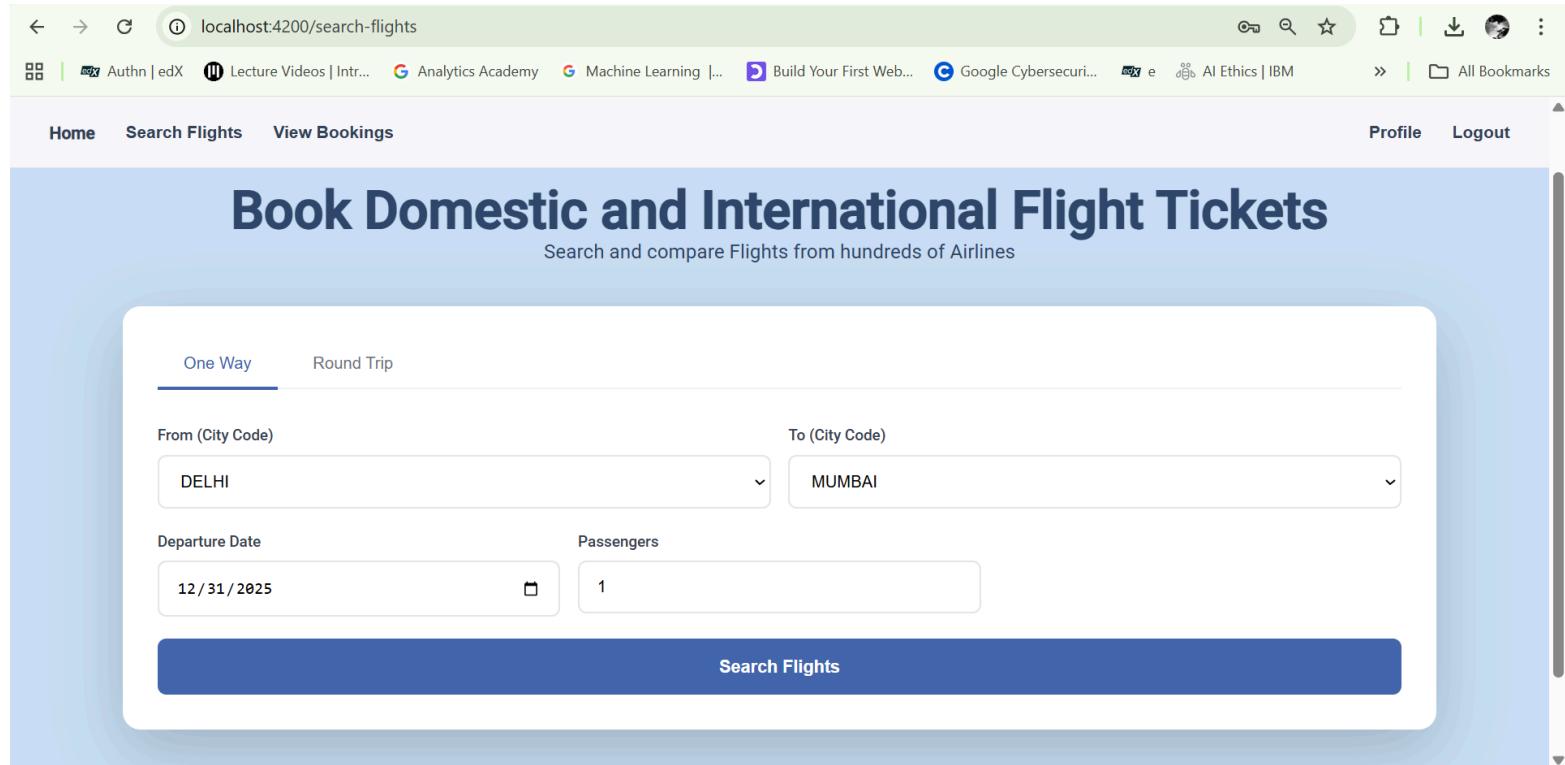
Meal Available

Add Flight

Please fill all required fields correctly

4. Search Flights

Search Flight for user: Select flight button enabled



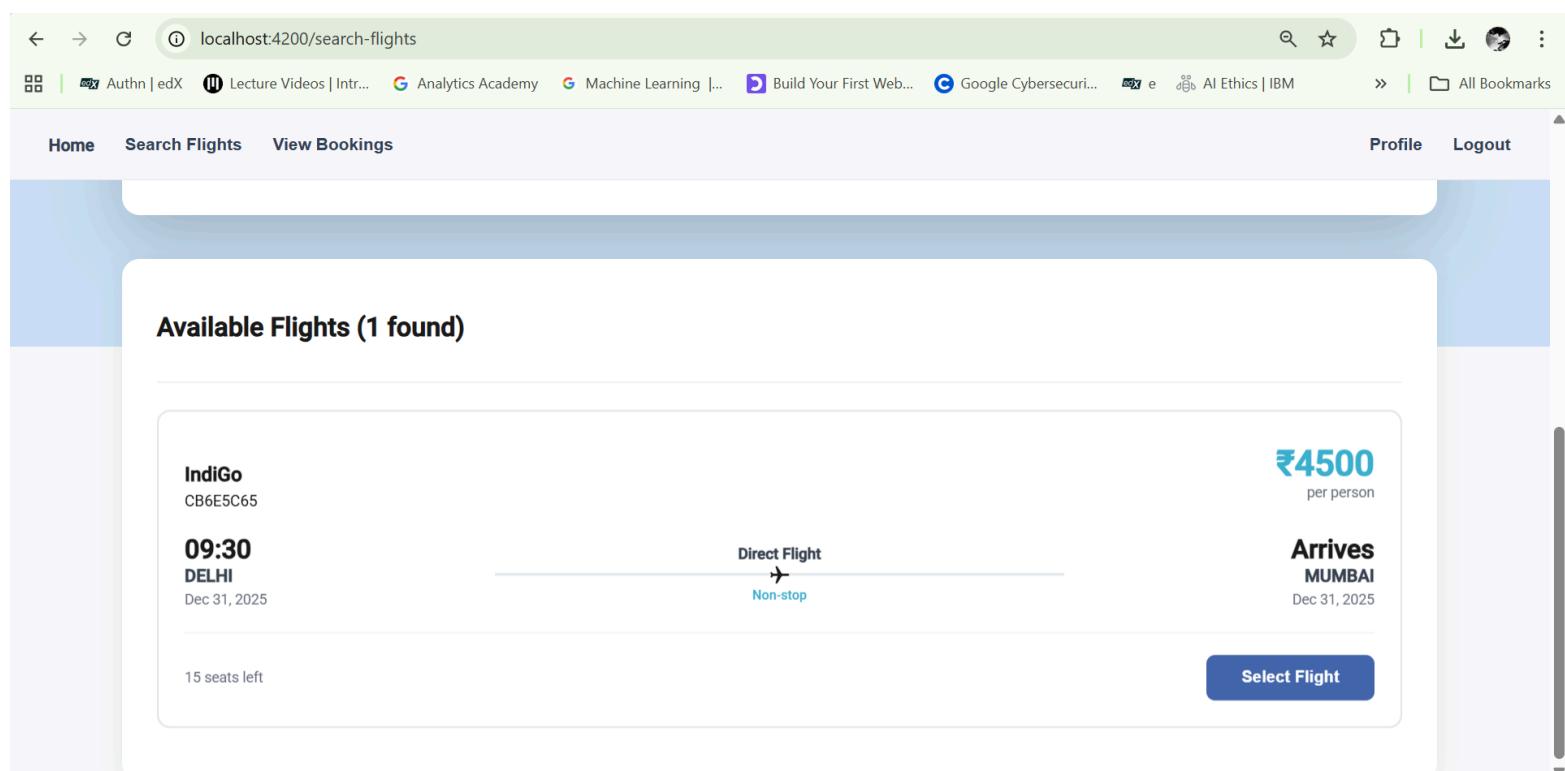
A screenshot of a web browser showing a flight search form. The URL in the address bar is "localhost:4200/search-flights". The page title is "Book Domestic and International Flight Tickets". The search criteria are set to "One Way" from "DELHI" to "MUMBAI" on "12/31/2025" for "1" passenger. A large blue "Search Flights" button is centered at the bottom of the form.

One Way Round Trip

From (City Code) To (City Code)

Departure Date Passengers

Search Flights



A screenshot of the same web browser showing the results of the flight search. The page title is "Available Flights (1 found)". One flight result is displayed for IndiGo, flight CB6E5C65, departing DELHI at 09:30 on Dec 31, 2025, arriving MUMBAI on the same day. The flight is described as a "Direct Flight Non-stop". The price is ₹4500 per person. A "Select Flight" button is located at the bottom right of the flight card.

Available Flights (1 found)

IndiGo
CB6E5C65

09:30
DELHI
Dec 31, 2025

Direct Flight
Non-stop

₹4500
per person

Arrives
MUMBAI
Dec 31, 2025

15 seats left

Select Flight

Search flight for admin: Select flight button not visible because only users should be able to book flights

The screenshot shows a web browser window with the URL localhost:4200/search-flights. The page title is "Book Domestic and International Flight Tickets". The search form includes fields for "From (City Code)" (set to DELHI), "To (City Code)" (set to MUMBAI), "Departure Date" (set to 12/31/2025), and "Passengers" (set to 1). A large blue "Search Flights" button is centered below the form.

The screenshot shows the same web browser window after the search has been performed. The page title is "Available Flights (1 found)". The result is a card for a flight from IndiGo. The flight details are: Departure at 09:30 from DELHI on Dec 31, 2025; Arrival at MUMBAI on Dec 31, 2025; Type: Direct Flight Non-stop; Price: ₹4500 per person; Seats left: 15. The "Search Flights" button is no longer visible, indicating the search has completed.

Logged out user can also access search flight page, but on clicking the select booking button they will be routed to the Login page

A screenshot of a web browser showing a flight booking form. The title bar says "localhost:4200/search-flights". The main heading is "Book Domestic and International Flight Tickets". Below it is a sub-heading "Search and compare Flights from hundreds of Airlines". There are two tabs at the top left: "One Way" (which is selected) and "Round Trip". The "From (City Code)" field contains "DELHI" and the "To (City Code)" field contains "MUMBAI". Below these fields are "Departure Date" (set to "12/31/2025") and "Passengers" (set to "1"). A large blue "Search Flights" button is centered below the passenger count.

5. Book Flights

Getting the username and email from the localstorage, they are marked as readonly

A screenshot of a web browser showing a "Confirm Booking" dialog. The title bar says "localhost:4200/book-flight/CB6E5C65". The dialog has sections for "Username" (readonly value "indu"), "Email" (readonly value "debashritamandal852@gmail.com"), "Passengers" (set to "1"), and "Meal Type" (set to "Vegetarian"). Below this is a "Passenger 1" section with fields for "Name" (readonly value "Debashrita"), "Age" (set to "22"), "Gender" (radio buttons for "Female" (selected) and "Male"), and "Seat Number" (set to "3B"). A large blue "Book Flight" button is at the bottom.

Booking details on successful booking

localhost:4200/booking-details/DFA55F50

Authn | edX Lecture Videos | Intr... Analytics Academy Machine Learning |... Build Your First Web... Google Cybersecuri... e AI Ethics | IBM All Bookmarks

Home Search Flights View Bookings Profile Logout

BOOKING DETAILS

PNR:	DFA55F50
Flight ID:	CB6E5C65
Status:	CONFIRMED
Seat Numbers:	3B
Total Price:	₹4500

PASSENGERS

*	NAME	GENDER	AGE
1	Debashrita	F	22

All inputs required validation

localhost:4200/book-flight/CB6E5C65

Authn | edX Lecture Videos | Intr... Analytics Academy Machine Learning |... Build Your First Web... Google Cybersecuri... e AI Ethics | IBM All Bookmarks

Home Search Flights View Bookings Profile Logout

Confirm Booking

Username	Email	Passengers	Meal Type
indu	debashritamandal852@gmail.com	1	Vegetarian

Passenger 1

Name	Age	Gender	Seat Number
Full name	Age	<input type="radio"/> Female <input type="radio"/> Male	e.g., 12A

Book Flight

All inputs required

6. Booking History

Filtered the booking by status

localhost:4200/booking-history

Home Search Flights View Bookings Profile Logout

My Booking History

<p>PNR D838F369 Passengers 1 Journey Date Dec 31, 2025 Total Price ₹4500 Status CONFIRMED</p> <p>Cancel Ticket</p>	<p>PNR 0C1DEDDB4 Passengers 1 Journey Date Dec 31, 2025 Total Price ₹4500 Status CONFIRMED</p> <p>Cancel Ticket</p>	<p>PNR 84F11FB4 Passengers 1 Journey Date Dec 31, 2025 Total Price ₹4500 Status CONFIRMED</p> <p>Cancel Ticket</p>
<p>PNR E5D05DC4 Passengers 1 Journey Date Dec 31, 2025 Total Price ₹4500 Status CONFIRMED</p> <p>Cancel Ticket</p>	<p>PNR DFA55F50 Passengers 1 Journey Date Dec 31, 2025 Total Price ₹4500 Status CONFIRMED</p> <p>Cancel Ticket</p>	<p>PNR 59E2B463 Passengers 1 Journey Date Dec 31, 2025 Total Price ₹4500 Status CONFIRMED</p> <p>Cancel Ticket</p>

localhost:4200/booking-history

Home Search Flights View Bookings Profile Logout

My Booking History

<p>Cancel Ticket</p>	<p>Cancel Ticket</p>	<p>Cancel Ticket</p>
<p>PNR 99C5D6E9 Passengers 2 Journey Date Dec 10, 2025 Total Price ₹9000 Status CONFIRMED</p> <p>Cancel Ticket</p>	<p>PNR E8739E21 Passengers 2 Journey Date Dec 31, 2025 Total Price ₹9000 Status CANCELLED</p> <p>Cancel Ticket</p>	<p>PNR A4CE9DC8 Passengers 1 Journey Date Dec 31, 2025 Total Price ₹4500 Status CANCELLED</p> <p>Cancel Ticket</p>
<p>PNR D4F30643 Passengers 2 Journey Date Dec 10, 2025 Total Price ₹9000 Status CANCELLED</p> <p>Cancel Ticket</p>	<p>PNR 97A2B1B1 Passengers 2 Journey Date Dec 10, 2025 Total Price ₹9000 Status CANCELLED</p> <p>Cancel Ticket</p>	

7. Cancel Booking

Ask for confirmation before cancelling

The screenshot shows a web application interface for managing bookings. At the top, there's a navigation bar with links like 'Home', 'Search Flights', 'View Bookings', 'Profile', and 'Logout'. Below this, a section titled 'My Booking History' displays several flight bookings in cards. One card for PNR D838F369 has a 'Cancel Ticket' button. A modal window titled 'Cancel Booking' is open over this card, asking 'Are you sure you want to cancel booking D838F369?' with 'No' and 'Yes, Cancel' buttons.

PNR	Passengers	Journey Date	Total Price	Status
D838F369	1	Dec 31, 2025	₹4500	CONFIRMED

PNR	Passengers	Journey Date	Total Price	Status
0C1DFDR4	1	Dec 31, 2025	₹4500	CONFIRMED

PNR	Passengers	Journey Date	Total Price	Status
84F11FB4	1	Dec 31, 2025	₹4500	CONFIRMED

PNR	Passengers	Journey Date	Total Price	Status
E5D05DC4	1	Dec 31, 2025	₹4500	CONFIRMED

PNR	Passengers	Journey Date	Total Price	Status
DFA55F50	1	Dec 31, 2025	₹4500	CONFIRMED

PNR	Passengers	Journey Date	Total Price	Status
59E2B463	1	Dec 31, 2025	₹4500	CONFIRMED

State updates immediately on successful cancellation

This screenshot shows the same booking history page after the cancellation. The modal now displays a 'Success' message: 'Ticket cancelled successfully' with a 'Close' button. The first booking card for PNR D838F369 now shows 'CANCELLED' in its status field instead of 'CONFIRMED'.

PNR	Passengers	Journey Date	Total Price	Status
D838F369	1	Dec 31, 2025	₹4500	CANCELLED

PNR	Passengers	Journey Date	Total Price	Status
0C1DFDR4	1	Dec 31, 2025	₹4500	CONFIRMED

PNR	Passengers	Journey Date	Total Price	Status
84F11FB4	1	Dec 31, 2025	₹4500	CONFIRMED

PNR	Passengers	Journey Date	Total Price	Status
E5D05DC4	1	Dec 31, 2025	₹4500	CONFIRMED

PNR	Passengers	Journey Date	Total Price	Status
DFA55F50	1	Dec 31, 2025	₹4500	CONFIRMED

PNR	Passengers	Journey Date	Total Price	Status
59E2B463	1	Dec 31, 2025	₹4500	CONFIRMED

Implemented the 24 hours rule and can't cancel a ticket when its past the journey date rule

The screenshot shows a grid of flight booking cards. The first three cards have their 'Cancel Ticket' buttons disabled, displaying a red button with a white 'Cancel Ticket' label. A central modal window titled 'Action Not Allowed' contains the message 'Journey has already started. Cancellation not allowed.' with a 'Close' button. The other four cards in the grid have functional 'Cancel Ticket' buttons.

PNR	Passengers	Journey Date	Total Price	Status
BEA70C00	2	Dec 10, 2025	₹9000	CONFIRMED
6AF5A39E	2	Dec 10, 2025	₹9000	CONFIRMED
E434EF57	2	Dec 10, 2025	₹9000	CONFIRMED
069A0BD5	2	Dec 10, 2025	₹9000	CONFIRMED
99C5D6E9	2	Dec 10, 2025		
E8739E21	2	Dec 31, 2025		
A4CE9DC8	1	Dec 31, 2025		

8. Profile

User profile (Displaying the saved data)

The screenshot shows a user profile page with a header navigation bar and a main content area. The main content area features a title 'My Profile' and a form containing fields for 'Username' (indu), 'Email' (debashritamandal852@gmail.com), and 'Role' (USER). A blue 'Change Password' button is at the bottom of the form.

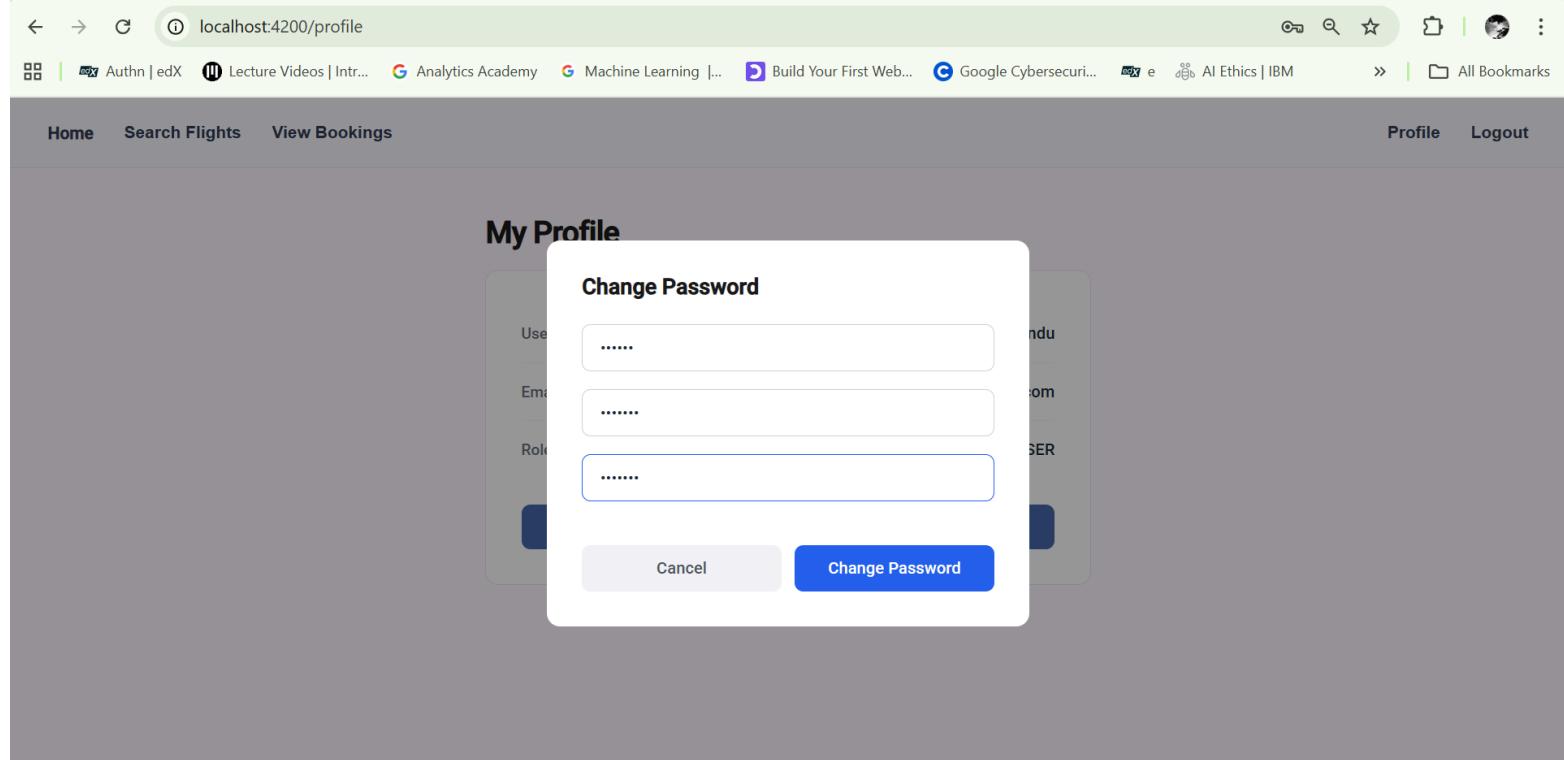
Username	indu
Email	debashritamandal852@gmail.com
Role	USER

Admin profile (Displaying the saved data)

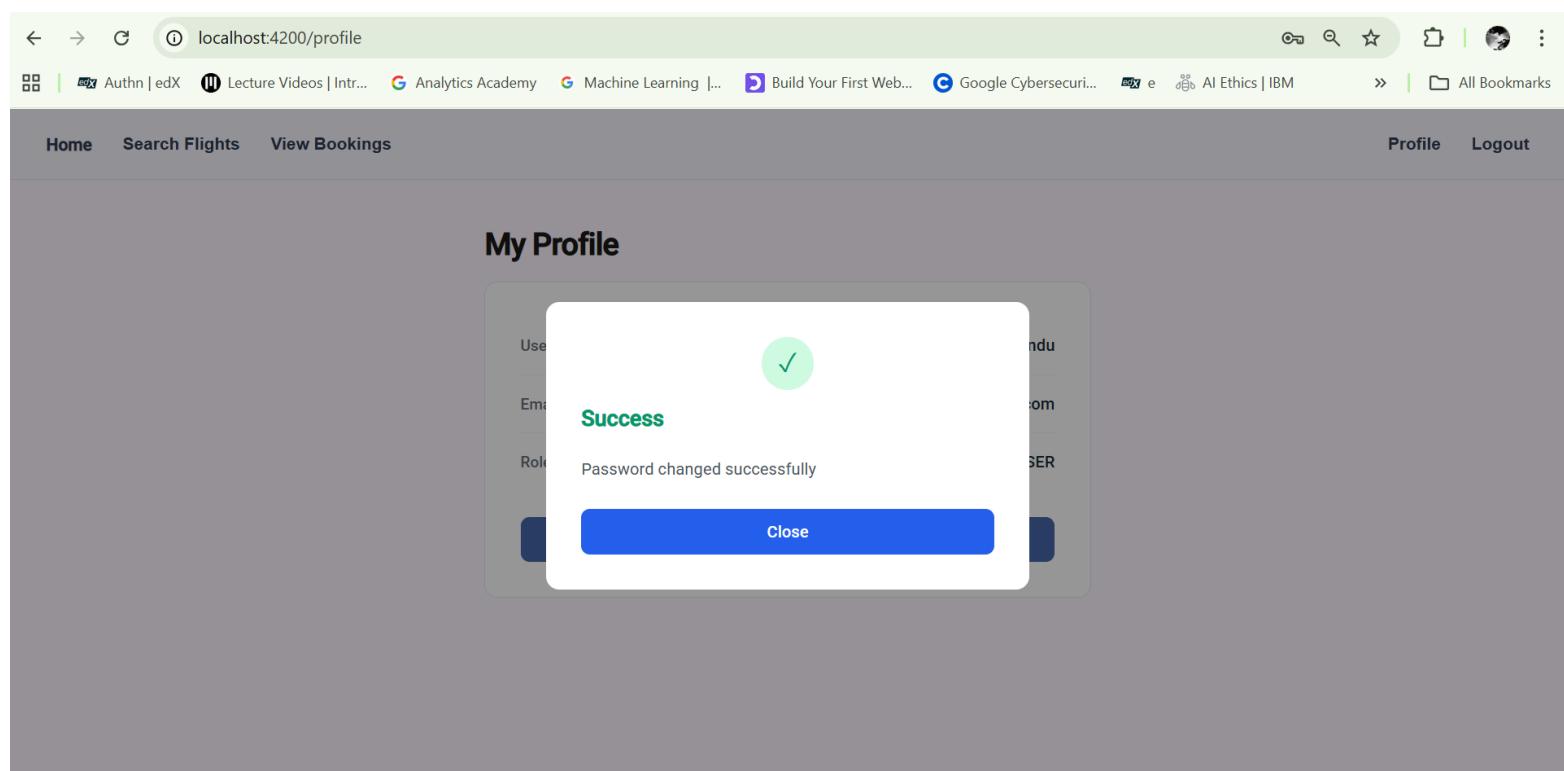
The screenshot shows a web browser window with the URL `localhost:4200/profile`. The page title is "My Profile". It displays three fields: "Username" (admin), "Email" (admin@gmail.com), and "Role" (ADMIN). Below these fields is a blue button labeled "Change Password". The browser's address bar and various tabs are visible at the top.

9. Change Password

The screenshot shows a web browser window with the URL `localhost:4200/profile`. The page title is "My Profile". A modal dialog box titled "Change Password" is open in the center. It contains three input fields: "Old Password", "New Password", and "Confirm New Password". At the bottom of the dialog are two buttons: "Cancel" and "Change Password". The background of the page is dimmed, indicating the modal is active. The browser's address bar and various tabs are visible at the top.



Successful password change



Unsuccessful password change: Old password was incorrect

A screenshot of a web browser window titled "localhost:4200/profile". The main content area shows a "My Profile" section with a "Change Password" button. A modal dialog box is displayed, containing a red "X" icon at the top right, the word "Error" in red, and the message "Old password is incorrect" below it. A blue "Close" button is at the bottom of the modal.

Unsuccessful password change:

A screenshot of a web browser window titled "localhost:4200/profile". The main content area shows a "My Profile" section with a "Change Password" button. A modal dialog box is displayed, containing a red "X" icon at the top right, the word "Error" in red, and the message "Passwords do not match" below it. A blue "Close" button is at the bottom of the modal.

10. Register

Successful registration

A screenshot of a web browser showing a "Create Account" form. The URL in the address bar is "localhost:4200/register". The page title is "Create Account" with a sub-instruction "Sign up to get started". The form fields are: "Username" (filled with "Arya"), "Email" (filled with "arya@gmail.com"), "Password" (filled with "*****"), and "Confirm Password" (filled with "*****"). A checkbox labeled "I agree to the Terms and Conditions" is checked. Below the form is a blue "Create Account" button. At the bottom, a link says "Already have an account? [Sign in](#)".

Unsuccessful registration: All inputs required

A screenshot of a web browser showing a "Create Account" form. The URL in the address bar is "localhost:4200/register". The page title is "Create Account" with a sub-instruction "Sign up to get started". The form fields are: "Username" (filled with "arya"), "Email" (empty), "Password" (empty), and "Confirm Password" (empty). An unchecked checkbox labeled "I agree to the Terms and Conditions" is present. Below the form is a red callout box containing the text "All fields are required". At the bottom is a blue "Create Account" button.

Unsuccessful registration: Username taken

A screenshot of a web browser window titled "Create Account". The URL in the address bar is "localhost:4200/register". The page content includes fields for "Username" (containing "arya"), "Email" (containing "arya@gmail.com"), "Password" (containing "*****"), and "Confirm Password" (containing "*****"). A checkbox labeled "I agree to the Terms and Conditions" is checked. Below the form, a red error message box contains the text "Username already taken". At the bottom is a blue "Create Account" button.

Unsuccessful registration: Email already taken

A screenshot of a web browser window titled "Create Account". The URL in the address bar is "localhost:4200/register". The page content includes fields for "Username" (containing "arya1"), "Email" (containing "arya@gmail.com"), "Password" (containing "*****"), and "Confirm Password" (containing "*****"). A checkbox labeled "I agree to the Terms and Conditions" is checked. Below the form, a red error message box contains the text "Email already registered". At the bottom is a blue "Create Account" button.

Unsuccessful registration: Passwords do not match

The screenshot shows a web browser window with the URL `localhost:4200/register`. The page title is "Create Account". The form fields are filled as follows: Username: "arya1", Email: "arya1@gmail.com", Password: "*****", Confirm Password: "*****". A checkbox labeled "I agree to the Terms and Conditions" is checked. A red error message "Passwords do not match" is displayed below the confirm password field. A blue "Create Account" button is at the bottom.

Create Account

Sign up to get started

Username

arya1

Email

arya1@gmail.com

Password

Confirm Password

I agree to the [Terms and Conditions](#)

Passwords do not match

Create Account

Unsuccessful registration: Please agree to terms and conditions (dummy validation)

The screenshot shows a web browser window with the URL `localhost:4200/register`. The page title is "Create Account". The form fields are filled as follows: Username: "arya1", Email: "arya1@gmail.com", Password: "*****", Confirm Password: "*****". An unchecked checkbox labeled "I agree to the Terms and Conditions" is present. A red error message "Please agree to the terms and conditions" is displayed below the checkbox. A blue "Create Account" button is at the bottom.

Create Account

Sign up to get started

Username

arya1

Email

arya1@gmail.com

Password

Confirm Password

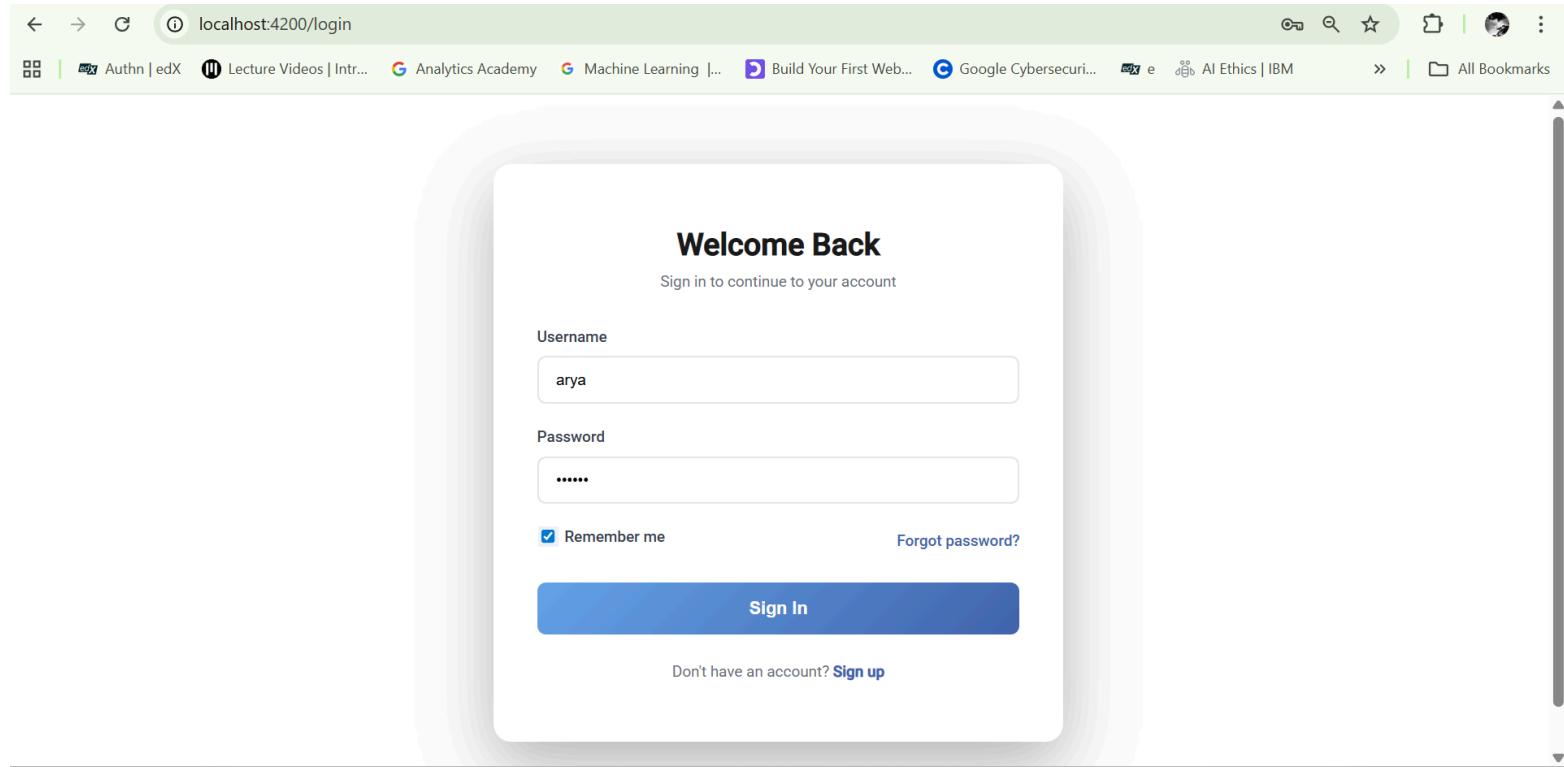
I agree to the [Terms and Conditions](#)

Please agree to the terms and conditions

Create Account

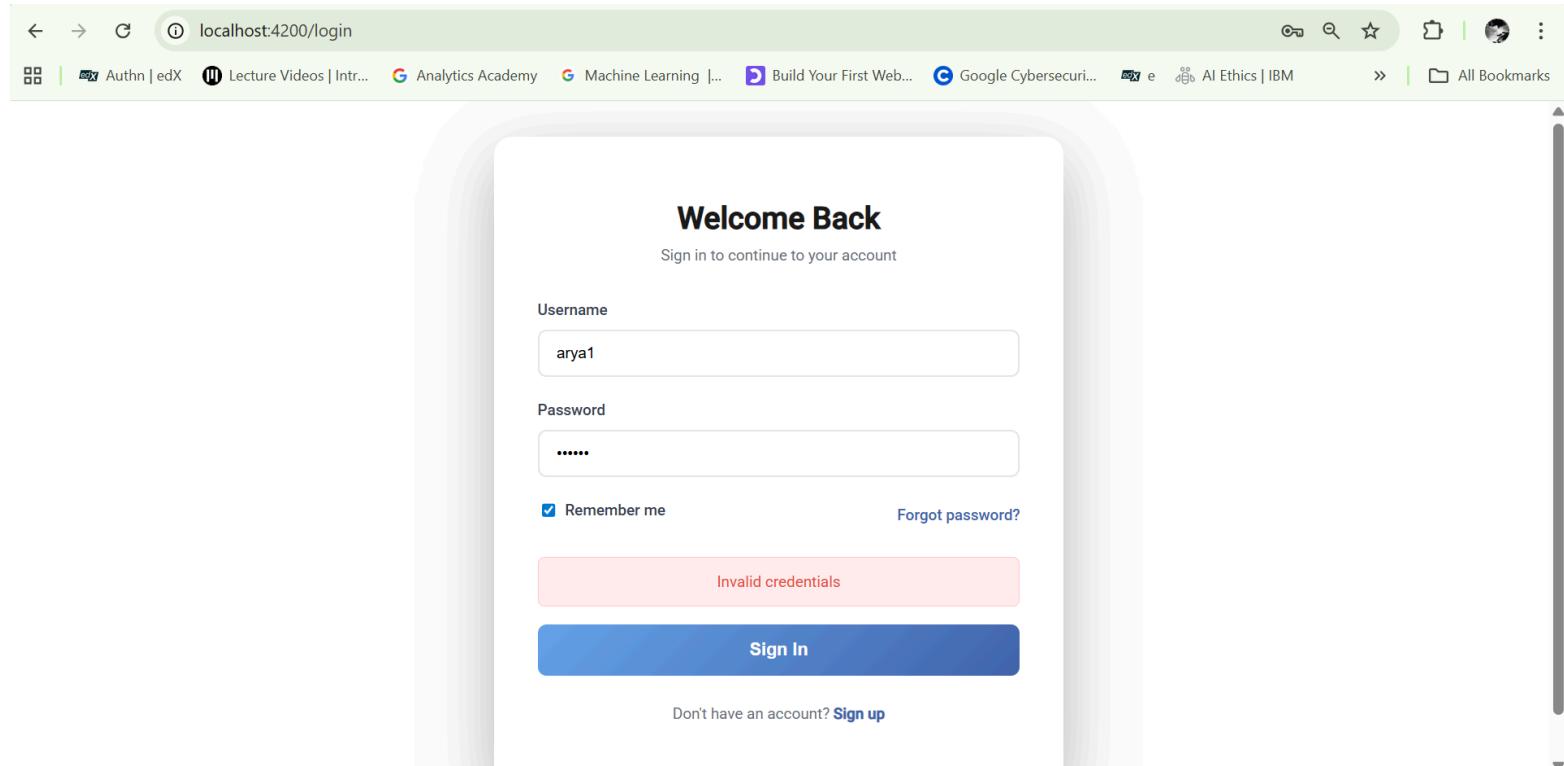
11. Login

Successful login



A screenshot of a web browser showing a successful login page. The title bar says "localhost:4200/login". The main content is a "Welcome Back" sign-in form. The "Username" field contains "arya" and the "Password" field contains "*****". A checked "Remember me" checkbox is next to a "Sign In" button. Below the form, a link says "Don't have an account? [Sign up](#)". The browser's address bar also shows "localhost:4200/login".

Unsuccessful login: Invalid credentials



A screenshot of a web browser showing an unsuccessful login attempt due to invalid credentials. The title bar says "localhost:4200/login". The main content is a "Welcome Back" sign-in form. The "Username" field contains "arya1" and the "Password" field contains "*****". A checked "Remember me" checkbox is next to a "Sign In" button. Below the form, a pink error message box says "Invalid credentials". A blue "Sign In" button is below it. Below the button, a link says "Don't have an account? [Sign up](#)". The browser's address bar also shows "localhost:4200/login".

Unsuccessful login: All fields required

The screenshot shows a browser window with the URL `localhost:4200/login`. The page title is "Welcome Back" with the subtitle "Sign in to continue to your account". There are two input fields: "Username" and "Password", both of which are empty. Below the password field is a checked checkbox labeled "Remember me" and a link "Forgot password?". A large red error message box contains the text "Please enter username and password". At the bottom is a blue "Sign In" button. Below the button is a link "Don't have an account? [Sign up](#)". The browser's address bar and various bookmarks are visible at the top.

11.Logout

Clicking on logout takes you back to the sign in page

The screenshot shows a browser window with the URL `localhost:4200/login`. The page title is "Welcome Back" with the subtitle "Sign in to continue to your account". The "Username" and "Password" fields are now populated with placeholder text ("Enter your username" and "Enter your password"). Below the password field is an unchecked checkbox labeled "Remember me" and a link "Forgot password?". At the bottom is a blue "Sign In" button. Below the button is a link "Don't have an account? [Sign up](#)". The browser's address bar and various bookmarks are visible at the top.

4. POSTMAN SCREENSHOTS

Change password

The screenshot shows the Postman interface with a successful API call. The collection 'FMS Microservice' is selected. A PUT request to 'http://localhost:8765/auth/change-password' was made at 21 Dec 2025, 10:05 PM. The request body contained:

```
1 {
2   "oldPassword": "password123",
3   "newPassword": "newpassword456"
4 }
```

The response was 200 OK, with a duration of 519 ms and a size of 403 B. The response body was:

```
{ } JSON ▾ ▶ Preview ▶ Visualize ▾
```

```
1 {
2   "message": "Password changed successfully"
3 }
```

Old password incorrect error

The screenshot shows the Postman interface with an error response. The collection 'FMS Microservice' is selected. A PUT request to 'http://localhost:8765/auth/change-password' was made at 2:19 AM. The request body contained:

```
1 {
2   "oldPassword": "password13",
3   "newPassword": "newpassword456"
4 }
```

The response was 403 Forbidden, with a duration of 547 ms and a size of 404 B. The response body was:

```
{ } JSON ▾ ▶ Preview ▶ Try with different auth credentials ▾
```

```
1 {
2   "error": "Old password is incorrect"
3 }
```

No jwt, unauthorized error

The screenshot shows the Postman interface with a 'PUT' request to `http://localhost:8765/auth/change-password`. The 'Authorization' tab is selected, showing 'Bearer Token' as the type with a 'Token' input field. The response status is '401 Unauthorized'. The request body is empty, containing only the number '1'.

Register user

The screenshot shows the Postman interface with a 'POST' request to `http://localhost:8765/auth/register`. The 'Body' tab is selected, showing the raw JSON payload:

```
1 {
2   "username": "anika",
3   "email": "anika@gmail.com",
4   "password": "password123"
5 }
```

The response status is '200 OK'. The response body contains the registered user's details, including a new 'role' field:

```
1 {
2   "username": "anika",
3   "email": "anika@gmail.com",
4   "role": "USER"
5 }
```

Admin role cant be self assigned

The screenshot shows the Postman interface with the following details:

- Workspace:** FMS Microservice
- Request:** POST /register
- Body:** Raw JSON (selected)

```
1 {  
2   "username": "anika",  
3   "email": "anika@gmail.com",  
4   "password": "password123",  
5   "role": "ADMIN"  
6 }  
7 }
```
- Response Status:** 403 Forbidden
- Response Body:**

```
1 {  
2   "error": "ADMIN role cannot be self-assigned"  
3 }
```

Login

The screenshot shows the Postman interface with the following details:

- Workspace:** FMS Microservice
- Request:** POST /signin
- Body:** Raw JSON (selected)

```
1 {  
2   "username": "debashrita",  
3   "password": "password123"  
4 }  
5 
```
- Response Status:** 200 OK
- Response Body:**

```
1 {  
2   "token": "eyJhbGciOiJIUzI1NiJ9.  
eyJzdWIiOiJKZWJhc2hyaxRrhIiwicm9sZSI6Ii1VTRVIiLCJpYXQiOjE3NjUxNDc3MzYsImV4cCI6MTc2NTIzNDEzNn0.  
zKbfZCLMV8IkY_MgzUNzGapL3nVimPNqhU8SO-1DWmA"  
3 }
```

Adding inventory (Only by Admin)

The screenshot shows the Postman interface. On the left, the 'Workspaces' sidebar lists various projects, with 'FMS Microservice' selected. Under 'FMS Microservice', the 'POST Add inventory' collection is highlighted. The main workspace displays a POST request to 'http://localhost:8765/FLIGHT-MICROSERVICE/api/flight'. The 'Authorization' tab is active, showing a 'Bearer Token' dropdown with a token value: eyJhbGciOiJIUzI1NiJ9.eyJzdWl0JhY1...'. Below this, a note states: 'The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.' The 'Body' tab shows a raw JSON payload: { "Access": "Denied" }. The response section shows a 403 Forbidden status with a 108 ms duration and 285 B size. The bottom navigation bar includes 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Vault', and 'Trash'.

Searching for a flight (open for any role)

The screenshot shows the Postman interface. The 'Workspaces' sidebar lists various projects, with 'FMS Microservice' selected. Under 'FMS Microservice', the 'POST Search Flights' collection is highlighted. The main workspace displays a POST request to 'http://localhost:8765/FLIGHT-MICROSERVICE/api/flight/search'. The 'Headers' tab is active, showing a 'Content-Type' header set to 'application/json'. The 'Body' tab shows a raw JSON payload: { "onwardFlight": [{ "availableSeatNumbers": ["1A", "1B", "2D", "1E", "1F"], "date": "2025-12-10T09:30", "flightId": "79448598", "airline": "IndiGo", "availableSeats": 5 }] }. The response section shows a 200 OK status with a 2.37 s duration and 1020 B size. The bottom navigation bar includes 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Vault', and 'Trash'.

Booking a flight (only for User)

The screenshot shows the Postman interface with the following details:

- Collection:** FMS Microservice
- Request Type:** POST
- URL:** http://localhost:8765/BOOKING-MICROSERVICE/api/flight/booking/AFA9CCF7
- Authorization:** Bearer Token (Token field is redacted)
- Body:** JSON response (201 Created)

```
{ "pnr": "E434EF57", "message": "Booking successful", "totalPrice": 9000.0 }
```

Search by PNR (only by User)

The screenshot shows the Postman interface with the following details:

- Collection:** FMS Microservice
- Request Type:** GET
- URL:** http://localhost:8765/BOOKING-MICROSERVICE/api/flight/ticket/6AF5A39E
- Authorization:** Bearer Token (Token field is redacted)
- Body:** JSON response (200 OK)

```
{ "pnr": "6AF5A39E", "flightId": "AFA9CCF7", "status": "CONFIRMED", "seatNumbers": [ "1A", "1B" ], "passengers": [ ] }
```

Search by mail (for User)

The screenshot shows the Postman interface with the following details:

- Workspace:** FMS Microservice
- Request URL:** http://localhost:8765/BOOKING-MICROSERVICE/api/flight/booking/history/debashritamandal1852@gmail.com
- Method:** GET
- Authorization:** Bearer Token (Token field contains a redacted token)
- Response Status:** 200 OK
- Response Body:**

```
1 {  
2   "email": "debashritamandal1852@gmail.com",  
3   "history": [  
4     {  
5       "pnr": "D4F30643",  
6       "status": "CANCELLED",  
7       "flightId": "79448598"  
8     },  
9   ]  
}
```

Delete the ticket (only by User)

The screenshot shows the Postman interface with the following details:

- Workspace:** FMS Microservice
- Request URL:** http://localhost:8765/BOOKING-MICROSERVICE/api/flight/booking/cancel/97A2B1B1
- Method:** DELETE
- Authorization:** Bearer Token (Token field contains a redacted token)
- Response Status:** 200 OK
- Response Body:**

```
1 {  
2   "pnr": "97A2B1B1",  
3   "message": "Ticket cancelled successfully"  
4 }
```


Flightdbnew: Airlines table

Navigator ::::::::::::::::::::: SQL File 1 booking users users users

SCHEMAS Filter objects

- flightdb
- flightdb_reactive
- ▼ flightdbnew
 - Tables
 - airline
 - ▼ flight
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - flight_available_seats
 - Views
 - Stored Procedures
 - Functions
- gatewaydb

Administration Schemas

Information :::::::::::::::::::::

SQL File 1 • SELECT * FROM flightdbnew.airline;

Result Grid | Filter Rows: Edit: 

	id	name
▶	1	IndiGo
	2	Air India
	3	Spice Jet
	4	SpiceJet
*	NULL	NULL

airline 1 ×

Output :::::::::::::::::::::

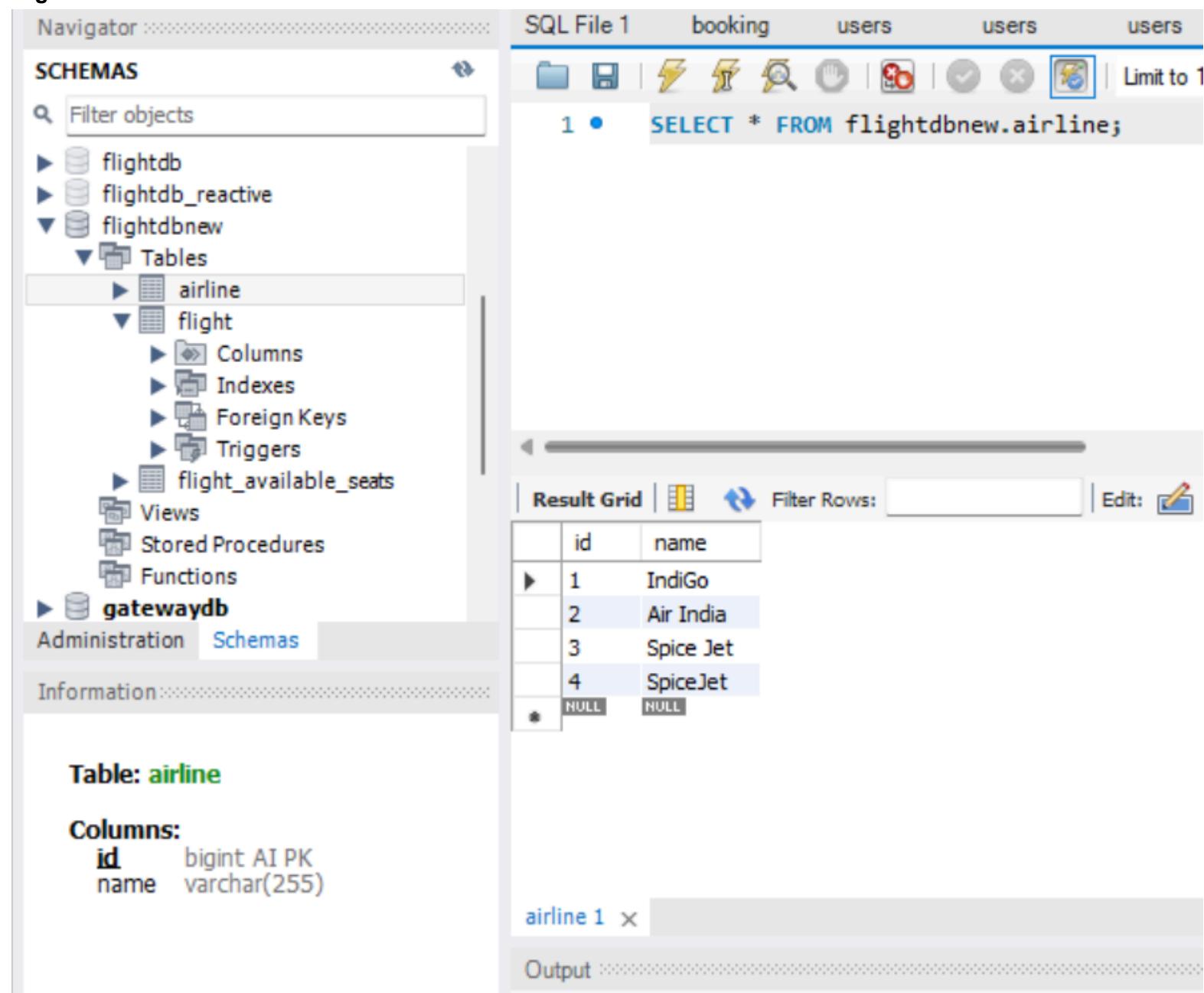


Table: airline

Columns:

id	bigint AI PK
name	varchar(255)

Bookingdbnew: Booking table

The screenshot shows the SQL Assistant interface with the following details:

- Schemas:** bookingdbnew
- Tables:** booking, booking_passengers, booking_seat_numbers, passenger
- Views, Procedures, Functions:** None listed.
- Flights:** flight, airline
- Flight Details:** flightdb, flightdb_reactive, flightdbnew
- Information:** Schemas
- Table:** airline
- Columns:**

id	name
bigint AI PK	varchar(255)
- Result Grid:** Shows 9 rows of booking data.
- Action Output:** Shows a log entry for a SELECT query.
- Object Info:** Session

Bookingdbnew: passenger table

The screenshot shows the SQL Assistant interface with the following details:

- Schemas:** bookingdbnew
- Tables:** booking, booking_passengers, booking_seat_numbers, passenger
- Views, Procedures, Functions:** None listed.
- Flights:** flight, airline
- Flight Details:** flightdb, flightdb_reactive, flightdbnew
- Information:** Schemas
- Table:** airline
- Columns:**

id	name
bigint AI PK	varchar(255)
- Result Grid:** Shows 10 rows of passenger data.
- Action Output:** Shows a log entry for a SELECT query.
- Object Info:** Session

6. Email Notification

The screenshot shows a Gmail inbox with 6,904 messages. An email from debashritamandal852@gmail.com is selected, with the subject "Booking Confirmation - 4EA5863B". The email body contains the following text:

Hello Debashrita Mandal,
Your booking is confirmed.
PNR: 4EA5863B
Flight ID: 79448598

At the bottom of the email view, there are buttons for Reply, Forward, and a smiley face icon.

7. Eureka server

The screenshot shows an Eureka server dashboard at localhost:8761. At the top, there are configuration settings:

Lease expiration enabled	false
Renews threshold	8
Renews (last min)	8

A red warning message is displayed: "EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE."

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - host.docker.internal:api-gateway:8765
BOOKING-MICROSERVICE	n/a (1)	(1)	UP (1) - host.docker.internal:BOOKING-MICROSERVICE:8082
CONFIG-SERVER-QUIZAPP	n/a (1)	(1)	UP (1) - host.docker.internal:config-server-quizapp:8888
FLIGHT-MICROSERVICE	n/a (1)	(1)	UP (1) - host.docker.internal:FLIGHT-MICROSERVICE

General Info

Name	Value
------	-------

CIRCUIT BREAKER (Resilience4j)

The screenshot shows the Resilience4j UI interface. On the left, there's a sidebar with a tree view of microservices: FMS, FMS Microservice, FMS_Reactive, FMS_Reactive_Mongo, New Collection, Quiz_microservices, Spring_Mongo_Reactive, Springboot_Mongodb, and WebfluxTutorial. Under FMS Microservice, 'Book Ticket' is selected. The main area is titled 'FMS Microservice / Book Ticket' and shows a 'POST' request to 'http://localhost:8082/api/flight/booking/79448598'. The 'Body' tab is active, displaying a JSON payload:

```
1 {
2   "email": "debashritamandal1852@gmail.com",
3   "name": "Debashrita Mandal",
4   "numberOfSeats": 2,
5   "passengers": [
6     { "name": "Debashrita Mandal", "gender": "F", "age": 21 },
7     { "name": "Aahana Banerjee", "gender": "F", "age": 22 }
]
```

The response status is '201 Created' with a time of '3.10 s' and a size of '253 B'. Below the response, there are tabs for Body, Cookies, Headers (5), Test Results, and a preview of the JSON response:

```
1 {
2   "message": "Cannot book right now: Flight service is unavailable",
3   "status": "FAILED"
4 }
```

8. RABBITMQ SERVER

The screenshot shows the Docker Desktop interface. The left sidebar has sections for Ask Gordon (BETA), Containers (selected), Images, Volumes, Kubernetes, Builds, Models, MCP Toolkit (BETA), Docker Hub, Docker Scout, and Extensions. The main area is titled 'Containers' and shows a summary of CPU and memory usage:

Container CPU usage: 0.48% / 800% (8 CPUs available)
Container memory usage: 133.3MB / 7.36GB

A table lists running containers:

	Name	Container ID	Image	Port(s)	CPU (%)	Actions
<input type="checkbox"/>	rabbitmq	-	-	-	0.71%	

Below the table, there's a 'Walkthroughs' section with two cards:

- Multi-container applications (8 mins)
- Containerize your application (\$ docker init) (3 mins)

At the bottom, there's a status bar with 'Engine running', system information (RAM: 1.40 GB, CPU: 0.13%, Disk: 1.73 GB used / limit 1006.85 GB), and a version 'v4.53.0'.

Registered RabbitMQ on Docker Desktop.

localhost:15672/#/

RabbitMQ TM RabbitMQ 3.11.28 Erlang 25.3.2.9

Refreshed 2025-12-01 11:58:42 Refresh every 5 seconds Virtual host All Cluster rabbit@5acd0416240f User fms_user Log out

Overview Connections Channels Exchanges Queues Admin

Overview

Totals

Queued messages last minute ? Currently idle

Message rates last minute ? Currently idle

Global counts ?

Connections: 0 Channels: 0 Exchanges: 7 Queues: 0 Consumers: 0

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@5acd0416240f	38 1048576 available	0 943629 available	404 1048576 available	145 MIB 3.0 GiB high watermark	954 GiB 48 MiB low watermark	3m 58s	basic disc 2 rss	This node All nodes	

Churn statistics
Ports and contexts
Export definitions

Logged into RabbitMQ.

localhost:15672/#/queues

RabbitMQ TM RabbitMQ 3.11.28 Erlang 25.3.2.9

Refreshed 2025-12-02 00:00:14 Refresh every 5 seconds Virtual host All Cluster rabbit@5acd0416240f User fms_user Log out

Overview Connections Channels Exchanges **Queues** Admin

Queues

All queues (1)

Pagination

Page 1 of 1 - Filter: Regex ? Displaying 1 item , page size up to: 100

Overview		Messages				Message rates			+/-	
Virtual host	Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack
/fms	booking.email.queue	classic	D	idle	0	0	0	0.00/s	0.00/s	0.00/s

Add a new queue

HTTP API Documentation Tutorials New releases Commercial edition Commercial support Discussions Discord Slack Plugins GitHub

Created a queue for Booking-Microservice.

9. JACOCO AND SONARQUBE REPORTS:

FLIGHT-MICROSERVICE:

1. JACOCO REPORT

Code Coverage: 93%

The screenshot shows a JaCoCo report for the 'flight-microservice' project. The main table provides a detailed breakdown of code coverage across various elements:

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cqty	Missed	Lines	Missed	Methods	Missed	Classes
com.flight.exception	82%	100%	1	6	2	14	1	5	0	1		
com.flight	0%	n/a	2	2	3	3	2	2	1	1		
com.flight.service	97%	85%	3	16	0	76	0	6	0	1		
com.flight.controller	91%	100%	1	7	1	11	1	5	0	1		
com.flight.model	100%	n/a	0	2	0	10	0	2	0	2		
Total	32 of 532	93%	3 of 26	88%	7	33	6	114	4	20	1	6

Created with JaCoCo 0.8.10 202304240956

2. SONARQUBE REPORT

Code Coverage: 93.6%

The screenshot shows the SonarQube summary report for the 'flight-microservice' project. The left sidebar provides navigation options like Overview, Main Branch, Pull Requests, and Branches. The main area displays the quality gate status and various metrics:

Debashrita Mandal > flight-microservice > master

Summary Issues Security Hotspots More

Define what should be considered new code for this project. After the next analysis the quality gate status will indicate if your new code is deployable or not.

Define New Code

Security: 0 Open issues (A) | Reliability: 0 Open issues (A) | Maintainability: 35 Open issues (A)

Accepted Issues: 0 | Coverage: 93.6% (No conditions set on 114 Lines to cover) | Duplications: 0.0% (No conditions set on 659 Lines)

Information and Administration links are also visible in the sidebar.

BOOKING-MICROSERVICE

1. JACOCO REPORT:

Code Coverage: 90%

The screenshot shows a detailed code coverage report for the 'booking-microservice' project. The report is presented in a table format with the following columns: Element, Missed Instructions, Cov., Missed Branches, Cov., Missed Cxty, Missed Lines, Missed Methods, and Missed Classes. The table includes a summary row for the entire project and a total row at the bottom.

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Cxty	Missed Lines	Missed Methods	Missed Classes				
com.booking.service	93%	72%	10	30	9	117	0	10	0	1		
com.booking.config	0%	n/a	4	4	6	6	4	4	1	1		
com.booking.publisher	0%	n/a	1	1	3	3	1	1	1	1		
com.booking	37%	n/a	1	2	2	3	1	2	0	1		
com.booking.model	100%	n/a	0	3	0	10	0	3	0	3		
com.booking.exception	100%	100%	0	7	0	16	0	6	0	1		
com.booking.controller	100%	n/a	0	4	0	5	0	4	0	1		
Total	74 of 769	90%	11 of 42	73%	16	51	20	160	6	30	2	9

2. SONARQUBE REPORT

Code Coverage: 84.7%

The screenshot shows the SonarQube project summary for 'booking-microservice'. The summary page displays several key metrics: Security (0 open issues), Reliability (0 open issues), Maintainability (38 open issues), Accepted Issues (0), Coverage (84.7%), and Duplications (0.0%). The left sidebar provides navigation links for Overview, Main Branch (selected), Pull Requests, and Branches.

11. JMETER TESTING

1. 20 USERS

Summary Report1.jmx (C:\Users\KIIT\Downloads\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\Summary Report1.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

00:00:14 0/0 0/170

Test Plan

Thread Group (20 users)

- HTTP Request
- Summary Report
- View Results Tree

Thread Group (50 Users)

- HTTP Request
- Summary Report
- View Results Tree

Thread Group (100 Users)

- HTTP Request
- Summary Report
- View Results Tree

HTTP Request

Name: HTTP Request

Comments:

Basic Advanced

Web Server

Protocol [http]: http Server Name or IP: localhost Port Number: 8082

HTTP Request

Method: GET Path: /api/flight/ticket/D4F30643 Content encoding:

Redirect Automatically Follow Redirects Use KeepAlive Use multipart/form-data Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?

Detail Add Add from Clipboard Delete Up Down

2. 50 USERS

Summary Report1.jmx (C:\Users\KIIT\Downloads\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\Summary Report1.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help 00:00:14 0 0/170

Test Plan

- Thread Group (20 users)
 - HTTP Request
 - Summary Report
 - View Results Tree
- Thread Group (50 Users)
 - HTTP Request
 - Summary Report
 - View Results Tree
- Thread Group (100 Users)
 - HTTP Request
 - Summary Report
 - View Results Tree

HTTP Request

Name: HTTP Request
Comments:

Basic Advanced

Web Server

Protocol [http]: http Server Name or IP: localhost Port Number: 8082

HTTP Request

Method: GET Path: /api/flight/ticket/D4F30643 Content encoding:

Redirect Automatically Follow Redirects Use KeepAlive Use multipart/form-data Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value:	URL Encode?	Content-Type	Include Equals?
-------	--------	-------------	--------------	-----------------

Detail Add Add from Clipboard Delete Up Down

Summary Report1.jmx (C:\Users\KIIT\Downloads\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\Summary Report1.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help 00:00:14 0 0/170

Test Plan

- Thread Group (20 users)
 - HTTP Request
 - Summary Report
 - View Results Tree
- Thread Group (50 Users)
 - HTTP Request
 - Summary Report
 - View Results Tree
- Thread Group (100 Users)
 - HTTP Request
 - Summary Report
 - View Results Tree

Summary Report

Name: Summary Report
Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
HTTP Request	100	5	3	13	1.73	0.00%	4.6/min	0.03	0.01	389.0
TOTAL	100	5	3	13	1.73	0.00%	4.6/min	0.03	0.01	389.0

Include group name in label Save Table Data Save Table Header

Summary Report1.jmx (C:\Users\KIIT\Downloads\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\Summary Report1.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

00:00:14 0 0/170

Test Plan

- Thread Group (20 users)
 - HTTP Request
 - Summary Report
 - View Results Tree
- Thread Group (50 Users)
 - HTTP Request
 - Summary Report
 - View Results Tree
- Thread Group (100 Users)
 - HTTP Request
 - Summary Report
 - View Results Tree

View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename:

Search: ... Case sensitive Regular exp. Search Reset

Text Sampler result

HTTP Request

Raw Parsed

Scroll automatically?

This screenshot shows the Apache JMeter interface with the 'View Results Tree' configuration selected in the left sidebar. The main panel displays a list of 15 'HTTP Request' entries, each preceded by a green checkmark indicating success. The 'Sampler result' dropdown is set to 'Text'. At the bottom, there are tabs for 'Raw' and 'Parsed' data, and a checkbox for 'Scroll automatically?'. The status bar at the top right shows '00:00:14 0 0/170'.

3. 100 USERS

Summary Report1.jmx (C:\Users\KIIT\Downloads\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\Summary Report1.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

00:00:14 0 0/170

Test Plan

- Thread Group (20 users)
 - HTTP Request
 - Summary Report
 - View Results Tree
- Thread Group (50 Users)
 - HTTP Request
 - Summary Report
 - View Results Tree
- Thread Group (100 Users)
 - HTTP Request
 - Summary Report
 - View Results Tree

HTTP Request

Name: HTTP Request

Comments:

Basic Advanced

Web Server

Protocol [http]: http Server Name or IP: localhost Port Number: 8082

HTTP Request

Method: GET Path: /api/flight/ticket/D4F30643 Content encoding:

Redirect Automatically Follow Redirects Use KeepAlive Use multipart/form-data Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?

Detail Add Add from Clipboard Delete Up Down

This screenshot shows the Apache JMeter interface with the 'HTTP Request' configuration selected in the left sidebar. The main panel displays the 'HTTP Request' settings, including the protocol (http), server name (localhost), port number (8082), method (GET), and path (/api/flight/ticket/D4F30643). There are also checkboxes for 'Follow Redirects', 'Use KeepAlive', 'Use multipart/form-data', and 'Browser-compatible headers'. Below the settings, there are tabs for 'Parameters', 'Body Data', and 'Files Upload'. At the bottom, there is a table for sending parameters with the request, and a row of buttons for 'Detail', 'Add', 'Add from Clipboard', 'Delete', 'Up', and 'Down'.

Summary Report1.jmx (C:\Users\KIIT\Downloads\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\Summary Report1.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help 00:00:14 ⚠ 0 0/170

Test Plan

- Thread Group (20 users)
 - HTTP Request
 - Summary Report
 - View Results Tree
- Thread Group (50 users)
 - HTTP Request
 - Summary Report
 - View Results Tree
- Thread Group (100 users)
 - HTTP Request
 - Summary Report
 - View Results Tree

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB..	Sent KB/sec	Avg. Bytes
HTTP Request	300	6	3	16	1.52	0.00%	13.4/min	0.08	0.03	389.0
TOTAL	300	6	3	16	1.52	0.00%	13.4/min	0.08	0.03	389.0

Include group name in label? Save Table Data Save Table Header

Summary Report1.jmx (C:\Users\KIIT\Downloads\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\Summary Report1.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help 00:00:14 ⚠ 0 0/170

Test Plan

- Thread Group (20 users)
 - HTTP Request
 - Summary Report
 - View Results Tree
- Thread Group (50 users)
 - HTTP Request
 - Summary Report
 - View Results Tree
- Thread Group (100 users)
 - HTTP Request
 - Summary Report
 - View Results Tree

View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: Errors Successes Configure

Search: Case sensitive Regular exp. Search Reset

Text Sampler result

HTTP Request
HTTP Request

Scroll automatically? Raw Parsed

12.Docker Desktop

The screenshot shows the Docker Desktop interface. On the left, the sidebar includes options like Ask Gordon (BETA), Containers, Images, Volumes, Kubernetes, Builds, Models, MCP Toolkit (BETA), Docker Hub, Docker Scout, and Extensions. The main area displays a stack named 'fms_microservices' located at C:\Users\KIIT\OneDrive\Desktop\chubb_training\FMS_Microservices. The stack contains seven services: 'fms_microse' (8761:8761), 'rabbitmq' (15672:15672), 'mysql' (3306:3306), 'notifica...' (8083:8083), 'config-s...' (8888:8888), 'api-gate...' (9745:9745), and 'service-...' (pink dot). Each service has a status icon, a blue square, a three-dot menu, and a trash can icon. To the right of the services is a terminal window showing Spring Boot logs from 2025-12-08T16:36:12.585Z to 2025-12-08T16:36:23.554Z. The logs include INFO messages from ServiceRegistryQuizappApplication and o.s.cloud.context.scope.GenericScope. A vertical toolbar on the right provides quick access to search, refresh, and other Docker functions. At the bottom, a status bar indicates 'Engine running', RAM usage (6.31 GB), CPU usage (1.51%), and disk usage (11.63 GB used / limit 1006.85 GB).

Ask Gordon BETA

Containers

Images

Volumes

Kubernetes

Builds

Models

MCP Toolkit BETA

Docker Hub

Docker Scout

Extensions

Engine running

RAM 6.31 GB CPU 1.51% Disk: 11.63 GB used (limit 1006.85 GB)

fms_microservices
C:\Users\KIIT\OneDrive\Desktop\chubb_training\FMS_Microservices

View configurations

Ctrl+K

Search

?

4

Notifications

Settings

Grid

M

Minimize

Maximize

Close

service-... ●

fms_microse 8761:8761

rabbitmq ●

rabbitmq:3-n 15672:15672

Show all ports

mysql ●

mysql:8.0 3306:3306

notifica... ●

fms_microse 8083:8083

config-s... ●

fms_microse 8888:8888

api-gate... ●

fms_microse 9745:9745

View configurations

Ctrl+K

Search

?

4

Notifications

Settings

Grid

M

Minimize

Maximize

Close

Spring Boot (v4.0.0)

2025-12-08T16:36:12.585Z INFO 1 --- [service_registry_quizapp] [main] c.e.d.ServiceRegistryQuizappApplication : Starting ServiceRegistryQuizappApplication v3.1.0 using Java 17.0.17 with PID 1 (/app/app.jar started by root in /app)

2025-12-08T16:36:12.644Z INFO 1 --- [service_registry_quizapp] [main] c.e.d.ServiceRegistryQuizappApplication : No active profile set, falling back to 1 default profile: "default"

2025-12-08T16:36:23.554Z INFO 1 --- [service_registry_quizapp] [main] o.s.cloud.context.scope.GenericScope : RootFactory id=fbbbaedc-5771-3000-0fc1-

Update available