Microservices Flight Application – Architectural Flow

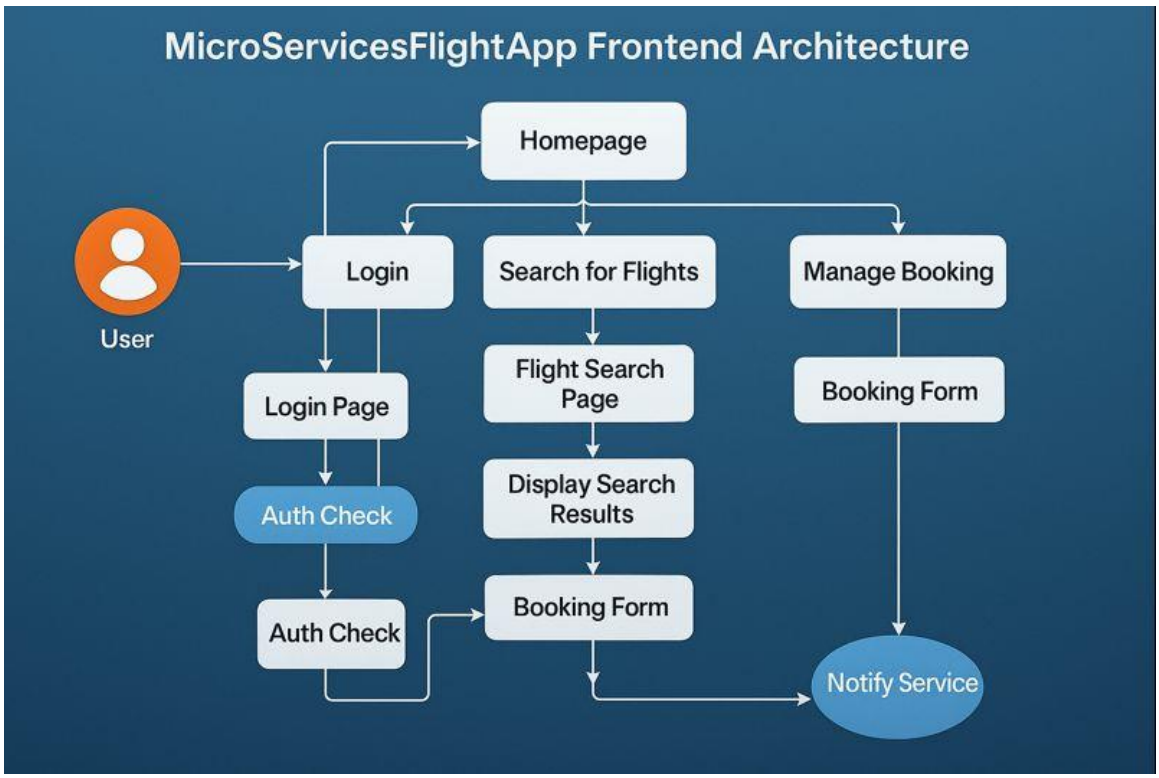
This document presents the complete architectural flow of the Microservices-based Flight Booking Application. The explanation is primarily visual and diagram-driven to ensure clarity during project submission, review, and viva evaluation.

1. Frontend Architecture Overview (Angular)

The frontend is developed using Angular and follows a modular, component-based architecture. It handles user interaction, routing, authentication checks, and API communication.

Frontend Flow Diagram

The diagram below shows how a user interacts with the frontend modules.

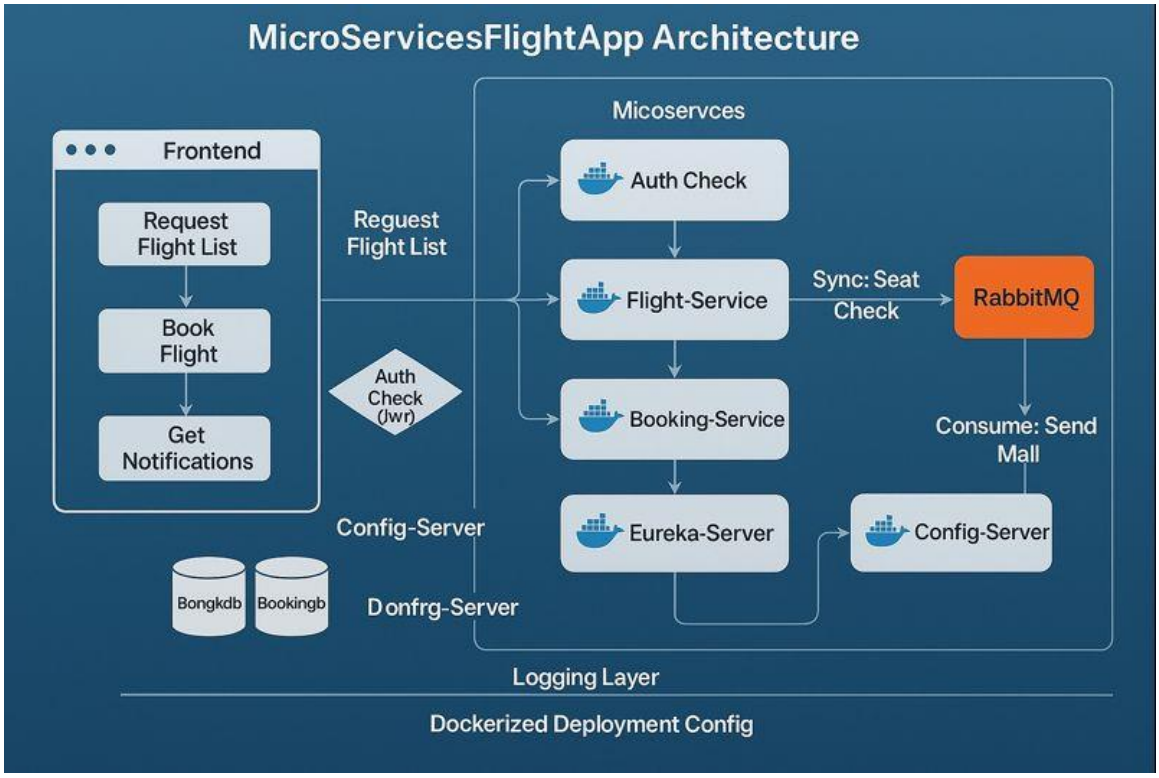


2. Backend Microservices Architecture Overview

The backend is built using Spring Boot microservices. Each service is independently deployable and communicates through REST APIs and asynchronous messaging.

Backend Architecture Diagram

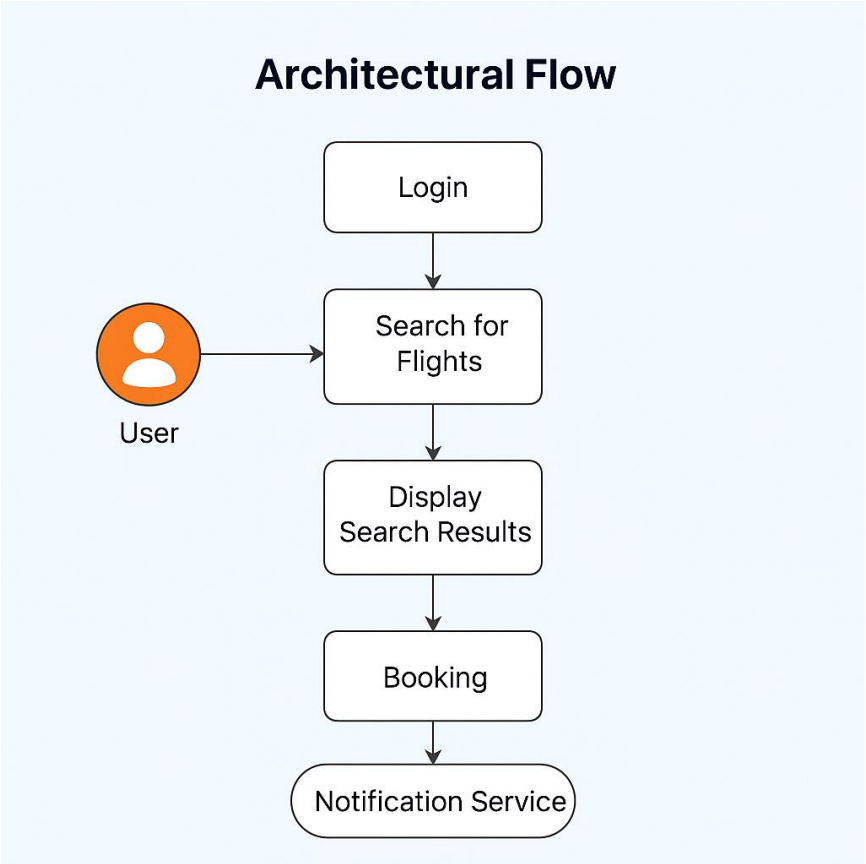
The diagram below illustrates the interaction between backend services and infrastructure.



3. End-to-End Working Flow (Visual Sequence)

Instead of textual steps, the complete system working is represented visually using the sequence flow diagram below.

System Flow Diagram



4. Key Architectural Highlights

- JWT-based authentication with route protection
- API Gateway as a single entry point
- Service discovery using Eureka
- Asynchronous communication via RabbitMQ
- Independent databases for Flight and Booking services
- Fully Dockerized deployment

Conclusion

This architecture emphasizes scalability, fault tolerance, and clean separation of concerns. The visual-first approach ensures easy understanding and professional project presentation.