AIRLINES MANAGEMENT SYSTEM

SUBMITTED BY

DEVIREDDY GOUTHAMI- 22BFA37033

DEPURI SUCHIRITHA- 22BFA37032

DAMPETLA PRANATHI- 22BFA37027

K PAVITHRA- 22BFA37048

INTRODUCTION –

The Airlines Management System is developed to improve the overall efficiency and effectiveness of airline operations. It allows airlines to manage their fleet and schedule flights in a systematic and organized manner.

The system supports seamless ticket booking by providing real-time seat availability and automatic seat allocation, ensuring a smooth reservation process for passengers. It also facilitates passenger check-in and handles baggage efficiently, reducing delays and manual effort.

To enhance communication, the system generates automated notifications and alerts for passengers, airlines, and airports, ensuring timely updates and coordination. With a user-friendly interface, the system is easy to operate for airline staff. Its robust database management ensures secure storage and quick access to large volumes of data related to flights, bookings, and passengers. Overall, the system optimizes workflow, increases customer satisfaction, and contributes to the smooth and reliable operation of airline services.

OBJECTIVES:

- **To streamline the management of flights, reservations, and passenger information.**
- **To automate ticket booking and provide real-time seat availability updates.**
- **To enable efficient scheduling of flights and proper allocation of airline resources.**
- **To simplify passenger check-in and baggage handling processes.**
- **To enhance communication between airlines, airports, and using automated notifications and alerts**

- **To ensure smooth operations through a user-friendly interface and centralized data handling.**
- **To improve overall workflow efficiency and customer satisfaction in the airline industry.**
- **To securely store and manage large volumes of operational and passenger data using a robust database system.**

REQUIREMENTS USED –

Functional Requirements:

- Flight management (scheduling, rescheduling, cancellations)
- Ticket booking and reservation system
- Real-time seat availability check
- Passenger check-in functionality
- Baggage handling and tracking system
- Automated notification and alert system (SMS/Email)
- User account management (Admin, Staff, Passenger)
- Report generation (flight status, booking summary, etc.)

Category	Description
Front End	HTML,CSS,JavaScript
Back-End	Java,Python,PHP
Database	MySQl,PostgreSQL
Server	Apache Tomcat
Tools	IDE
Hosting	Cloud
Hardware	Minimum 4 GB RAM,i5 or higher CPU

TECHNICAL SPECIFICATIONS -

- **Frontend Tools**
- **React.js / Angular / Vue.js:** For building dynamic, responsive UIs
- **Bootstrap / Tailwind CSS:** For responsive design and styling

- **Axios / Fetch API:** For handling HTTP requests
- **Backend Tools**
- **Spring Boot (Java) / Django (Python):** For building scalable backend services
- **Express.js (Node.js):** For lightweight REST API development
- **Hibernate / SQLAlchemy / Entity Framework:** For ORM-based database interaction
- **JWT / OAuth2:** For secure authentication and session handling
- **Database Management**
- **MySQL / PostgreSQL / Oracle:** For relational data management
- **MongoDB:** If using a NoSQL approach for flexible data modeling
- **phpMyAdmin / pgAdmin:** For database GUI tools
- **DevOps & Deployment**
- **Git / GitHub / GitLab:** Version control
- **Docker / Kubernetes:** For containerization and orchestration

- **Jenkins / GitHub Actions:** For CI/CD pipelines
- **AWS / Azure / Heroku:** For cloud deployment and hosting

PROJECT IMPLEMENTATION –

Front-End

- **Technologies**: HTML, CSS, JavaScript, React.js (or Angular)
- **Purpose**: Provides a user interface for passengers, staff, and admins
- **Features**:
  - Login/Signup
  - Search & book flights
  - Seat selection
  - Check-in form
  - Notifications & alerts
- **Tools**: Axios (API calls), React Router, Bootstrap/Tailwind

Back-End

- **Technologies**: Java (Spring Boot) / Python (Django) / Node.js (Express)
- **Purpose**: Handles business logic, data processing, and APIs
- **Features**:
  - User authentication (JWT)
  - Flight & booking management
  - Check-in & baggage handling
  - Reports & notifications
- **Database**: MySQL / MongoDB
- **Tools**: Postman (testing), Hibernate/ORM, Swagger (API docs)

APPLICATION SCENARIOS -

A passenger visits the airline's website or app to **book a flight** from Delhi to Mumbai.

1. They **search** for available flights by entering departure and destination cities and dates.
2. The system displays **available flights** with timing, pricing, and seat options.

3. The passenger **selects a flight**, chooses a **preferred seat**, and **completes the booking** with payment.
4. Before travel, they **check-in online**, select baggage options, and receive a **boarding pass via email or app**.
5. At the airport, staff use the system to verify check-ins, update **real-time flight status**, and manage baggage.
6. If the flight is delayed or rescheduled, the system automatically **notifies the passenger** via SMS/email.
7. Admins and airline staff monitor all operations through the **dashboard** for reporting and decision-making.

RESULT

The Airlines Management System successfully streamlines airline operations by integrating flight scheduling, seat allocation, ticket booking, check-in, and baggage management into a single platform. It enhances **operational efficiency**, reduces manual errors, improves **passenger experience**, and ensures **real-time communication** between passengers, airline staff, and airport authorities. The system is scalable, secure, and user-friendly, making it suitable for real-world deployment in the airline industry.

Output Format

The Airlines Management System provides various user-specific outputs based on their roles—**passenger**, **staff**, or **admin**. The outputs are displayed in a user-friendly and structured format that

enhances user experience and system efficiency. The following are the main outputs generated by the system:

- When a **user logs in**, the system verifies credentials and redirects them to their respective dashboards based on role (passenger, staff, admin).
- Upon **searching for flights**, the system displays a list of available flights with details such as flight number, airline name, departure and arrival times, and fare.
- In the **booking module**, the system allows passengers to select a flight, view seat availability through a visual seat map, choose a seat, and confirm their booking. A booking ID is generated along with ticket details.
- During **check-in**, the system updates the passenger's status, assigns gate and time details, and generates a downloadable boarding pass.
- For **admins and staff**, the dashboard displays real-time statistics such as total flights for the day, number of bookings, completed check-ins, and system alerts or issues.
- The system also provides **alerts and notifications** to users via email or SMS in case of delays, cancellations, or changes in flight schedules.

THANK YOU