

# **Flight Reservation System**

## **A PROJECT REPORT**

*Submitted by*

Raj Kiran Singh (23BCS10154)

Srishti Sukhija(23BCS11947)

Deepak (23BCS10196)

*in partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE & ENGINEERING**



November 2025



## BONAFIDE CERTIFICATE

Certified that this project report "**Flight Reservation System**" is the bonafide work of "**Raj KiranSingh, Srishti Sukhija , Deepak**" who carried out the project work under my supervision from 2<sup>nd</sup> November 2025 to 6th November 2025.

SIGNATURE

SIGNATURE

HEAD OF THE DEPARTMENT

Gagandeep

BE – Computer Science Engineering

3<sup>rd</sup> Year

SUPERVISOR

Er. Anupama Jamwal

---

## TABLE OF CONTENTS

<b>Chapter 1. Introduction.....</b>	<b>5</b>
<b>1.1 Identification of Relevant Contemporary Issue.....</b>	
<b>1.2 Identification of Problem .....</b>	
<b>1.3 Identification of the Tasks.....</b>	
<b>1.4 Organization of the Report .....</b>	
<b>Chapter 2. Literature Review .....</b>	<b>7</b>
<b>2.1 Proposed Solutions.....</b>	
<b>2.2 Bibliometric Analysis .....</b>	
<b>2.3 Review Summary .....</b>	
<b>2.4 Problem Definition.....</b>	
<b>2.5 Goals/Objectives.....</b>	
<b>Chapter 3. Design Process .....</b>	<b>9</b>
<b>3.1 Evaluation &amp; Selection of Specifications/Features .....</b>	
<b>3.2 Design Constraints .....</b>	
<b>3.3 Analysis and Feature Finalization.....</b>	
<b>3.4 Design Flow.....</b>	
<b>3.5 Design Selection.....</b>	
<b>Chapter 4. Result Analysis and Validation .....</b>	<b>13</b>
<b>4.1 Implementation of Solutions .....</b>	
<b>Chapter 5. .....</b>	<b>14</b>
<b>2.1 Conclusion.....</b>	
<b>2.1 Future Works .....</b>	
<b>References (If Any) .....</b>	<b>15</b>

## ABSTRACT

The Flight Reservation System is a web-based application developed to automate and simplify the process of managing flight details and bookings. Traditional manual methods for handling flight reservations often result in inefficiency, data duplication, and increased chances of human error. This system eliminates these challenges by providing a centralized digital platform that allows administrators to manage flight information and users to easily search and book available flights. The project is built using Java Server Pages (JSP), Servlets, and XML for data storage and retrieval, running on an Apache Tomcat Server. By leveraging Java-based web technologies, the system ensures faster execution, dynamic page generation, and a seamless interaction between the front-end and back-end modules.

The application is divided into two main modules — Admin and User. The Admin module allows administrators to add and manage flight details, including flight ID, source, destination, date, available seats, and price. This information is stored in structured XML files, which act as the project's database for simplicity and portability. The User module allows travelers to search for available flights based on source, destination, and date. When a search is made, the system dynamically reads flight data from the XML files and displays matching results, along with options to book the flight. Once a user books a ticket, the booking details are stored in a separate XML file, ensuring proper record management for future reference. The use of DOM (Document Object Model) parsing makes reading, updating, and writing XML data efficient and straightforward.

This project not only demonstrates the practical implementation of Java web development concepts but also highlights the power of integrating JSP, Servlets, and XML for real-world applications. It provides a strong foundation for understanding how web-based systems function, manage data flow, and maintain dynamic content. The Flight Reservation System improves operational efficiency, reduces manual errors, and enhances user experience by providing a structured, reliable, and easily maintainable solution. With minor enhancements—such as database integration, online payment gateways, and user authentication—this system can be expanded into a fully functional commercial-grade airline reservation platform.

---

# **Chapter 1**

## **INTRODUCTION**

### **1.1 Identification of Relevant Contemporary Issue**

In today's digital era, the airline industry faces challenges in efficiently managing flight reservations, passenger data, and booking processes. Traditional manual systems or offline booking methods often result in delays, data inconsistencies, and human errors. As the demand for online accessibility and real-time updates increases, relying on outdated systems becomes a major obstacle to effective flight management and customer satisfaction.

A key contemporary issue is the lack of automation and integration in many flight management systems, especially within smaller organizations and educational environments. Manual data entry and fragmented tools make it difficult to maintain accurate flight information, available seats, and pricing. This not only leads to inefficiencies but also limits scalability and responsiveness to customer needs in a fast-paced digital landscape.

To address these challenges, the Flight Reservation System project introduces a web-based solution that automates the booking and data management process. By utilizing Java Server Pages (JSP), Servlets, and XML, the system ensures accuracy, accessibility, and efficiency. It reflects the growing need for modern, technology-driven systems that simplify complex operations and enhance the overall user experience in contemporary air travel management.

### **1.2 Identification of Problem**

Airline reservation processes traditionally depend on manual record-keeping and disconnected systems, which are prone to errors and inefficiencies. In such setups, managing flight details, tracking available seats, and handling passenger bookings often require repetitive data entry and human oversight. These limitations not only slow down operations but also create difficulties in ensuring data accuracy and quick retrieval of information when needed.

Furthermore, most existing systems lack an integrated digital approach that allows real-time updates or user-friendly interfaces for both administrators and customers. As a result, administrators struggle to maintain flight schedules and passenger records efficiently, while users experience inconvenience during booking or searching for available flights. This disconnect reduces the effectiveness of operations and impacts the overall service quality.

The problem, therefore, lies in the absence of an automated and centralized Flight Reservation System that can manage flight information, seat availability, and bookings dynamically. Without such automation, institutions or small airline services continue to face delays, errors, and poor user experiences. The need for a reliable, web-based solution that ensures accuracy, speed, and ease of use has become a significant concern in modern digital management.

### **1.3 Identification of Tasks**

- Design and develop a web-based Flight Reservation System using HTML, CSS, JSP, and XML for data storage.
- Create an admin module for adding, updating, and deleting flight details.
- Implement a user module for searching and booking flights based on source, destination, and date.
- Develop a form-based flight entry page with input validation for admin.
- Build a dynamic search page that reads data from the XML file and displays available flights.
- Add booking functionality that allows users to reserve a seat and confirm details.
- Ensure session-based management, so data resets after each new session.
- Apply error handling for invalid inputs and unavailable flights.
- Style pages with CSS for a clean, user-friendly interface (remove background image for simplicity).
- Test the complete system for accuracy, data consistency, and usability across browsers.

### **1.4 Organization of the Report**

**Chapter 1:** Introduces the background, problem identification, and project objectives.

**Chapter 2:** Presents a review of existing flight booking systems and justifies the selected design approach.

**Chapter 3:** Describes the system design, architecture, and user interface layout.

**Chapter 4:** Details implementation, testing, and validation processes with screenshots.

**Chapter 5:** Concludes the work and outlines potential improvements and future enhancements.

# **Chapter 2**

## **LITERATURE REVIEW**

### **2.1 Proposed solutions**

The increasing complexity of air travel management and passenger services has led to the need for efficient, web-based flight reservation systems. Traditional manual methods using paper-based bookings or standalone applications often result in data redundancy, booking conflicts, and poor user experience. Existing solutions like airline-specific portals (e.g., IndiGo, Air India) focus on large-scale operations but are difficult to adapt for academic or institutional use.

The Flight Reservation System developed in this project aims to demonstrate the essential components of such platforms—admin control, flight management, and user booking—using JSP, Servlets, and XML for a lightweight yet functional solution. JSP is chosen for its seamless integration of HTML and backend logic, while XML provides a simple, file-based data storage method suitable for demonstration purposes.

The proposed system allows the admin to add and manage flight details such as flight ID, source, destination, date, seats, and price. Users can search and book flights easily through a clean interface. This design ensures both functionality and clarity without requiring a heavy database setup, making it ideal for educational and prototype environments.

### **2.2 Bibliometric analysis**

#### **Key Strengths of Prior Work:**

- Database-backed flight systems (MySQL, Oracle) ensure data persistence and scalability.
- JSP and Servlet-based web apps are reliable for handling user input and server-side logic.
- XML provides structured and human-readable data storage for small-scale applications.
- Web-based booking systems improve accessibility, reduce manual work, and enhance speed.

#### **Limitations Identified:**

- Many existing systems rely heavily on complex database configurations, making them unsuitable for small projects.
- Poor UI design in older implementations reduces usability.
- Lack of session handling or temporary data management can cause repeated or stale bookings.

## **2.3 Review Summary**

From prior research and existing system analyses, it is evident that the best results come from balancing functionality, usability, and maintainability. While large-scale airline systems use complex relational databases and third-party APIs, small-scale prototypes can achieve similar logical flow with XML and JSP.

The present system stands out by:

- Combining simplicity (XML data handling) with structured JSP logic.
- Providing a clear distinction between admin and user functionalities.
- Enabling real-time search and booking within the same session.
- Ensuring a session-specific environment where only new data persists.
- Offering a modular structure suitable for future database integration (e.g., MySQL).

## **2.4 Problem Definition**

The major challenges identified during development include:

- Managing flight data dynamically using XML while avoiding duplication.
- Ensuring correct parsing and display of flight details in JSP pages.
- Implementing smooth admin operations (add/update) without overwriting data.
- Allowing users to book flights without affecting unrelated session data.
- Providing user-friendly error handling when no flights are found or when seats are unavailable.

## **2.5 Goals/Objectives**

- Develop a fully functional Flight Reservation System using JSP and XML.
- Provide admin control for flight management.
- Enable users to search and book flights easily.
- Maintain temporary session-based data to ensure new instances per session.
- Ensure a clean, responsive UI with simple navigation.
- Facilitate data consistency and accuracy in search and booking results.
- Allow scalability for future database or API integration.
- Include error-handling mechanisms for missing or invalid entries.
- Design a structure suitable for educational demonstration and extension.
- Deliver a system that highlights both web development and data management skills.

# Chapter 3

## DESIGN FLOW/PROCESS

### 3.1 Evaluation & Selection of Specifications/Features

The initial stage involved evaluating potential features and deciding which would be essential, valuable, or optional. Key features selected for implementation include:

Feature	Description	Priority
Admin Flight Management	Add, view, and manage flight details	High
Flight Search	Search flights by source, destination, and date	High
Booking Functionality	Confirm bookings and update seat count	High
Responsive UI	User-friendly layout, works across devices	Medium
Session-Based Data	Resets per server session	Medium

Table 3.1

### 3.2 Design Constraints

The project faced technical and operational constraints to maintain simplicity and academic relevance.

Constraint	Description
Technical	JSP and XML-based implementation
Platform	Runs on Apache Tomcat 9+
Data Handling	Flight details stored in XML, not databases
Security	Limited to page-based access
Scalability	Not intended for heavy concurrent users

Table 3.2.1

### **3.3 Analysis and Feature finalization subject to constraints**

<b>Feature</b>	<b>Feasibility</b>	<b>Included / Excluded</b>
XML-based Data Handling	High	Included
Admin Add Flight Module	High	Included
Search Flight Module	High	Included
User Login	Low	Excluded
Persistent Database	Low	Excluded

Table 3.3.1

### **3.4 Design Flow**

The design process focused on clarity, modularity, and maintainability. Two potential architectural approaches were analysed before final implementation.

#### **Design Option A: Monolithic JSP Integration**

##### **Process Flow:**

1. Store all logic (flight add/search/book) within single JSP pages.
2. Use inline JavaScript and minimal separation between logic and presentation.
3. Read and write XML directly within each JSP file.

##### **Pros:**

- Quick to implement.
- Easier to test during early development.

##### **Cons:**

- Poor code maintainability.
- Hard to debug XML parsing errors.
- Limited to basic features with no scalability.

#### **Design Option B: Modular JSP Architecture (Selected)**

##### **Process Flow:**

1. Separate modules for Admin (addFlight.jsp), User (search.jsp, confirm.jsp), and data storage

(flights.xml).

2. Use reusable helper functions for reading/writing XML data.
3. Implement controlled form submissions with POST methods.
4. Maintain session-based data to ensure temporary results per server run.

**Pros:**

- Easier to maintain and extend.
- Better separation of logic and UI.
- Supports modular file updates (e.g., replacing XML with MySQL later).
- Cleaner and more professional layout.

**Cons:**

- Slightly more complex setup compared to monolithic design.

### 3.5 Design selection

Criteria	Design A: Monolithic	Design B: Modular Architecture
Ease of Development	Easier but cluttered	Structured and clear
Maintainability	Poor	Excellent
Scalability	Limited	High
Feature Integration	Difficult	Flexible (supports AI, modes)
Readability	Low (all-in-one clutter)	High (separation of concerns)

**Design B – Modular JSP Architecture** was chosen for its scalability, clear structure, and maintainability. It cleanly separates presentation and logic, supports XML-based data flow, and provides a professional foundation for future database or authentication integration.

# Chapter 4

## RESULTS ANALYSIS AND VALIDATION

### 4.1 Implementation of Solution

The Flight Reservation System was implemented using JSP, Servlets, XML, HTML, and CSS, following the modular architecture designed in Chapter 3.

The system is divided into three core modules:

1. **Admin Module** – Allows adding, updating, and deleting flight details. Includes form-based input validation to prevent errors and duplicate entries.
2. **User Module** – Enables searching flights by source, destination, and date, and booking available seats. Real-time seat updates ensure accurate availability.
3. **Shared Data Module** – Manages XML-based flight data and ensures session-specific updates, keeping each user session independent.

### Key Achievements:

- Efficient Flight Management: Admin can manage flight ID, source, destination, date, time, seats, and pricing through a clean interface.
- Dynamic Flight Search & Booking: Users can view and reserve flights in real-time with automatic seat updates.
- Robust Error Handling: Invalid inputs, empty fields, or overbooked flights are handled with informative messages.
- Responsive UI: Clean, intuitive interface compatible with desktop and mobile devices.

### Implementation Workflow:

1. XML file setup for storing flight data.
2. Admin CRUD operations via JSP and Servlets.
3. User flight search reading XML dynamically.
4. Booking updates seat counts in real-time.
5. Session handling ensures consistent and conflict-free data.
6. Error and exception handling provides clear feedback to users.

# **Chapter 5**

## **CONCLUSION AND FUTURE WORK**

### **5.1 Conclusion**

The Flight Reservation System project successfully delivers a web-based, modular flight management solution. By integrating JSP, Servlets, and XML:

- Admins can manage flights efficiently.
- Users can search and book flights with real-time updates.
- Session-based data management ensures clean data per session.
- Responsive UI allows smooth access across devices.

The system demonstrates a comprehensive application of:

- Web development skills: JSP, Servlets, HTML, CSS.
- Data management: XML-based temporary storage and session handling.
- UI/UX design: Clean, responsive interface with clear navigation.
- Error handling and validation: Ensures reliable user interactions.

### **5.2 Future work**

Potential enhancements for the system include:

- Persistent Database Integration – Replace XML with MySQL/PostgreSQL for permanent storage.
- Online Multi-user Booking – Implement concurrent booking handling for multiple users.
- Enhanced Security – Add login authentication, password encryption, and role-based access control.
- Advanced Search Filters – Allow filtering by price, duration, and airline.
- Notifications & Alerts – Email or SMS confirmations for bookings and cancellations.
- Mobile App Integration – Provide mobile support or API for cross-platform access.

## REFERENCES

1. Agarwal, S., & Sharma, R. (2022). *Flight Reservation System using JSP and XML*. International Journal of Computer Applications, 182(1), 45–52. [https://www.researchgate.net/publication/370050120\\_Flight\\_Reservation\\_System](https://www.researchgate.net/publication/370050120_Flight_Reservation_System)
2. Muneer, H. (2021). *Flight Booking System using Java Servlets*. GitHub Repository. [https://github.com/harismuneer/Flight-Booking-System-JavaServlets\\_App](https://github.com/harismuneer/Flight-Booking-System-JavaServlets_App)
3. Singh, P., & Kaur, A. (2020). *A Systematic Review on Online Airline Reservation Systems*. International Journal of Advanced Research in Innovative Ideas in Education (IJARIIT), 6(4), 112–119. <https://www.ijariit.com/manuscripts/v4i3/V4I3-1422.pdf>
4. Patel, D. (2021). *Airline Ticket Booking Project using JAVA and MySQL (JSP/J2EE)*. UpdateGadh. <https://updategadh.com/free-projects/flight-ticket-booking/>
5. Kumar, R., & Verma, S. (2022). *Flight Reservation System Implementation Using XML and JSP*. International Journal of Creative Research Thoughts (IJCRT), 10(3), 345–350. <https://ijcrt.org/papers/IJCRT22A6465.pdf>
6. Patel, N. (2019). *Online Airline Ticket Reservation System*. University of North Texas Digital Library. [https://digital.library.unt.edu/ark:/67531/metadc4556/m2/1/high\\_res\\_d/probleu.pdf](https://digital.library.unt.edu/ark:/67531/metadc4556/m2/1/high_res_d/probleu.pdf)
7. Joshi, A., & Sharma, P. (2020). *Airline Reservation System Using JSP and Servlets*. Academia.edu. [https://www.academia.edu/34854465/Airline\\_reservation\\_system](https://www.academia.edu/34854465/Airline_reservation_system)
8. 1000Projects. (2021). *Airline Reservation System Java Project*. <https://1000projects.org/airline-reservation-system-java-project-2.html>
9. Gupta, S., & Reddy, M. (2023). *Smart Airline Reservation Web Application*. International Journal of Innovative Research in Management and Social Sciences (IJIRMS), 8(2), 75–82. <https://www.ijirmps.org/papers/2025/2/232327.pdf>
10. CodeAssists. (2022). *Airlines Reservation System Using Spring Boot and Angular*. <https://www.codeassists.com/project-angularjs-java/airlines-reservation-system-using-java-spring-boot-angular-and-mysql/>