Airline Reservation System

A Mini Project Submitted to the

Maria Arts and Science College for Women

(Affiliated to the Manonmaniam Sundaranar University, Tirunelveli)

In partial fulfilment of the requirements for

The award of the degree of

BACHELOR OF DATA SCIENCE

Project work done by

S.ABI RAJA JOSHIBA

Reg.No-23081431804112001

Under the guidance of

N. HANNAH VINCY M.E

Assistant Professor, Department of Data science

Maria Arts and Science College for Women, Vallioor.

DEPARTMENT OF DATA SCIENCE

Maria College of Arts and Science for women

Vallioor

2023-2026

CERTIFICATE

Certified that this project AIRLINE RESERVATION SYSTEM Bonafide work done by **S.ABI RAJA JOSHIBA** during the academic year 2023 - 2026 who carried out the project under our supervision, certified further that to the best of our knowledge the work reported here in does not from part of any other project work on the basic of which a degree or award was conferred on an easier occasion on this or any other candidate.

Place: Vallioor

Date:

Head of the Department

External Examiner

1.

2.

Project Guide

Mrs. N. HANNAH VINCY M.E

DECLARATION

I hereby declare that the project work entitled "**Airline Reservation System**" Submitted to the **Department of Data Science, Maria Arts and Science College for Women, Vallioor** is a record of original work done by me.

Signature of the Candidate

PLACE:

DATE:

ACKNOWLEDGEMENT

At the Foremost, I thank God for all his blessings to complete my project successfully. Next, I thank my Parents for their financial and moral support. I express my sincere gratitude to the respected Principal (FAC) of our college **Dr. S. Sushma Jenifer pdrf.,** for their encouragement and moral support.

A special thanks to the Head of the Department **Dr. S. R. Siva Ambika M.Sc., Ph.D.**, for her encouragement to complete the project successfully.

My special thanks to the respected Guide **Mrs. N. HANNAH VINCY M.E** for her intensive support for the completion of the project.

We perceive as this opportunity as a big milestone in my career development. I will strive to use the gained skills and knowledge in the best possible way, and I will continue to work on the improvement of my knowledge.

CONTENT

1.Abstract:

The Airline Reservation System (ARS) is a comprehensive and automated software application designed to facilitate the booking and management of airline travel for passengers. At its core, the ARS integrates multiple components including flight scheduling, passenger information management, ticketing, payment processing, seat selection, and real-time inventory updates into a single coherent platform. The primary objective of such a system is to streamline the booking process for both customers and airline staff, while ensuring data accuracy, high availability, and system scalability.

Advanced ARS implementations are increasingly leveraging cloud infrastructure to support scalability, fault tolerance, and global accessibility. Furthermore, integration with Global Distribution Systems (GDS) such as Amadeus, Sabre, and Travelport allows airlines to expand their customer reach by making inventory accessible to third-party booking platforms.

Security and data integrity are paramount; thus, the system must comply with international standards, such as PCI DSS for payment data and GDPR for personal information. Modern systems also incorporate AI and ML algorithms for dynamic pricing, predictive maintenance, customer behaviour analysis, and demand forecasting.

The system typically includes modules for flight management, passenger information, booking and cancellation, fare calculation, and reporting. It is designed with a user-friendly interface and backed by a secure database to store and retrieve data quickly and accurately. Modern ARS solutions are integrated with Global Distribution Systems (GDS) and support features like e-ticketing, dynamic pricing, and mobile access.

2. Introduction:

An **Airline Reservation System (ARS)** is a software application used by airlines and travel agencies to manage flight bookings, ticketing, and scheduling. It allows customers to search for flights, check availability, reserve seats, and make payments online or through agents. For airlines, the system handles backend operations like managing flight schedules, passenger records, and fare tariffs.

Modern airline reservation systems are often integrated with global distribution systems (GDS) like Amadeus, Sabre, or Travelport, enabling real-time access to flight information across multiple airlines. These systems are essential for ensuring efficiency, accuracy, and customer satisfaction in the highly dynamic airline industry.

3. System Requirements:

3.1 Software Requirements:

Server Side:

- Operating System: Linux (Ubuntu/CentOS) or Windows Server
- Database: MySQL / PostgreSQL /MongoDB
- Backend Language: Java / PHP/ Node.js
- Web Server: Apache / Nginx
- Payment Gateway API: Stripe, Razorpay, PayPal, etc.
- GDS Integration (Optional): Sabre, Amadeus, Travelport

Client Side:

- Frontend Technologies: HTML5, CSS3, JavaScript
- Frameworks (optional): React / Angular / Vue.js
- Mobile App (Optional): Android (Kotlin/Java), iOS (Swift)

3.2. Hardware Requirements:

Server:

- Process: Intel Xeon or AMD EPYC (Quad Core or higher)

- RAM: 16 GB minimum
- Storage: 500 GB SSD or more
- Network: 1 Gbps Ethernet
- Backup system for disaster recovery

Client:

- Any device with browser (PC, laptop, mobile)
- RAM: 4 GB (for staff/admin)
- Modern web browser (Chrome, Firefox, Safari)

4.Objectives of the Front-End:

The front-end is the visible part of the Airline Reservation System — it's what users (passengers and admins) interact with. It includes the design, layout, forms, and user interface (UI) elements. It is built using:

- HTML – For page structure and content
- CSS – For design and layout
- JavaScript – For interactivity and logic (e.g., form validation or redirection)

Structure of the Front-End

The front-end is made up of multiple pages, each fulfilling a specific role:

File name	Description
search.html	Form to search for available flights.
flights.html	Displays the flight search results.
book.html	Booking form for passenger details.
admin.html	Admin login interface
style.css	Common styling used across all HTML pages.

Front-End Components and Pages

1. index.html – Homepage

Purpose:

- Acts as the landing page for users.
- Provides navigation to search for flights, book tickets, or access admin login.

Feature:

- Simple layout with welcome message and navigation links.
- Creates a professional first impression.

Why it's important:

- Helps users quickly understand where to go.
- Makes the app feel organized and intentional.

2. search.html – Flight Search Form

Purpose:

- Allows users to input travel details such as:
  - From and to destinations
  - Departure date and optional return date
  - Travel class (Economy, Business, First)
  - Number of passengers

Features:

- Responsive form layout
- Required fields to avoid empty searches
- JavaScript is used to redirect the user to a flight results page (flights.html) with all input data passed in the URL as query parameters.

Why it's important:

- This is the core interaction for users who want to travel.
- It filters flights based on real needs.
-

3. flights.html – Flight Results Display

Purpose:

- Displays search results after form submission.
- Shows matching flights with options like flight number, time, price, and "Book Now" button.

Features:

- Dynamically reads URL parameters (using JavaScript).
- Can display static or dynamic flight cards.

Why it's important:

- Helps users choose a flight from available options.
- Acts as a bridge between searching and booking.

4. book.html – Flight Booking Form

Purpose:

- Allows users to enter personal details and confirm booking.
- Fields include name, age, gender, and contact information.

Features:

- Uses HTML5 validation (required, type="number", etc.)
- Can be extended to include seat selection or payment integration.

Why it's important:

- This is the final step in the user booking process.
- Ensures correct passenger data is collected for ticket generation.

5. admin.html – Admin Login Page

Purpose:

- Provides a secure entry point for administrators to manage the system.

Features:

- Username and password fields
- Ready to connect with backend authentication logic
- Redirects to admin dashboard (not included in front-end)

Why it's important:

- Separates normal users from admin roles.
- Prepares the system for more complex management features like flight addition, cancellations, and report generation.

Core Functionalities of the Front-End:

The front-end of the ARS typically includes the following functionalities:

Flight Search:

- Users can search for flights using parameters like departure and arrival cities, dates, and number of passengers.
- Search results are dynamically updated based on filters (flight duration, price, airline).

Flight Booking:

- Selection of flights and seat preferences.
- Input of passenger details (name, age, passport info).
- User-driven booking confirmation with price breakdown.

Real-Time Updates:

- The front-end integrates with back-end systems to display live flight statuses (delays, cancellations) and updates to booking details.

Payment Integration:

- Secure payment forms for users to pay for their flights.
- Options to apply promo codes or discounts during payment processing.
- Display of invoices and booking receipts post-payment.

Booking Management:

- Users can view their bookings, check-in, and receive e-tickets.
- Users can modify or cancel bookings and get refund status.

Notifications & Alerts:

- Users receive notifications for upcoming flights, cancellations, gate changes, or delays.
- Email or SMS integration for reminders.

Challenges in Front-End Development:

- **Real-Time Data Integration**: Displaying live data (flight availability, pricing, payment status) in a timely manner can be challenging, especially in high-demand periods.
- **Cross-Browser Compatibility**: Ensuring that the front-end works seamlessly across different browsers (Chrome, Firefox, Safari) and devices requires careful testing.
- **Security:** Handling sensitive data, such as personal details and payment information, demands strict security protocols (e.g., HTTPS, tokenization) to protect users' privacy.
- **Scalability:** The front-end should be scalable to handle high traffic volumes, especially during peak travel seasons.

5. Objective of Back-End :

The back-end of an Airline Reservation System (ARS) is the engine that powers the entire booking and management process. It is responsible for handling all the business logic, data storage, transaction processing, and communication with external systems like payment gateways, flight information services, and user authentication systems. This report aims to provide an in-depth overview of the back-end architecture, technologies, components, and key functionalities that drive an ARS.

Key Components of the Back-End:

The back-end of an Airline Reservation System consists of multiple components that work together to support front-end functionalities. These include:

Database Management System (DBMS):

The central repository where all critical data (flight information, user details, bookings, transactions, etc.) is stored.

Server-Side Logic:

Handles the core business logic, including flight search algorithms, booking processing, payment verification, and more.

APIs (Application Programming Interfaces):

Interfaces for communication between the front-end, third-party services (e.g., payment processors, flight information providers), and internal services.

Authentication and Authorization:

Manages user authentication (login/logout) and authorization (permissions for accessing certain parts of the system, like admin vs. customer).

Payment Gateway Integration:

Handles the secure processing of financial transactions between customers and airlines.

Booking and Reservation System:

Manages the booking lifecycle, including availability checks, seat reservations, and confirmation of ticket purchases.

Notifications and Alerts:

Sends notifications to users (e.g., booking confirmations, flight status updates, cancellations).

Core Functionalities of the Back-End:

The back-end of an Airline Reservation System provides various essential services, including:

Flight Search Engine:

Flight Data Integration: The system queries databases or third-party APIs to gather real-time flight data (e.g., flight availability, pricing, duration, etc.).

Search Algorithm:

The back-end processes search queries based on user input (departure/arrival cities, dates, number of passengers) and fetches relevant results.

Filters:

The system applies filters to refine search results, such as sorting by price, duration, or airline.

Booking and Reservation Management:

Seat Availability:

When a user selects a flight, the back-end checks seat availability in real-time. It ensures that a seat is reserved only after payment confirmation.

Booking Creation:

Once a seat is selected, the system generates a booking reference number, stores the passenger's information, and reserves the seat.

Booking Updates:

Allows users to modify or cancel their bookings. The system must handle seat availability and update both the user's booking and the airline's inventory.

Booking Confirmation:

Once a booking is made, the system generates a ticket and sends confirmation details to the user (via email, SMS, etc.).

Payment Processing:

The back-end integrates with payment gateways (e.g., Stripe, PayPal, Razorpay) to securely process transactions. The payment details are validated, and once successful, a transaction record is created.

The system also handles failure cases, such as insufficient funds or timeouts, and alerts the user accordingly.

User Authentication & Authorization:

The system handles user registration, login, and profile management. It uses JWT or OAuth 2.0 tokens for secure user sessions.

Role-based Access Control (RBAC): The back-end distinguishes between different user roles (e.g., customers, admin, airline staff) and grants or restricts access based on roles.

Booking History & User Management:

The system maintains a user profile and booking history. Users can view their past bookings, make changes, or request refunds.

Admin Dashboards: Airline staff can use back-end interfaces to manage flight inventory, prices, cancellations, and customer requests.

Real-Time Flight Status Updates:

The back-end continuously fetches real-time flight information such as delays, cancellations, and gate changes, often from third-party APIs like FlightAware.

Notifications are sent to users when there are significant changes to their flight status.

Notifications & Alerts

The system sends booking confirmations, flight reminders, cancellations, and delay notifications via email, SMS, or push notifications.

Users can opt into alerts for various events like flight status updates or price drops.

Reporting and Analytics

The back-end generates reports for airlines, covering areas such as sales performance, flight demand, and customer behavior.

Business Intelligence (BI) Tools: Tools like Tableau or Power BI can be integrated for visual data analysis.

5. Challenges in Back-End Development:

Scalability: The ARS must handle peak traffic periods, such as holiday seasons or special promotions, without performance degradation.

Real-Time Data Handling: Integrating real-time flight data from multiple external APIs can be complex, especially when managing delays, cancellations, or pricing changes.

High Availability: The system should ensure high uptime and fault tolerance, as any downtime could result in lost sales and customer dissatisfaction.

Data Consistency: Managing the consistency of flight and booking data is crucial, especially when multiple users may be attempting to book the same seat simultaneously.

6. Objectives of an Airline Reservation System:

The following are the primary objectives of an Airline Reservation System, explained in depth:

Streamline Flight Booking Process:

One of the core objectives of an ARS is to provide an efficient, user-friendly platform for customers to book flights. This is achieved through:

Online Flight Search: The system allows users to search for available flights based on parameters like departure/arrival cities, dates, and passenger details. It provides real-time availability and pricing.

Seat Selection: Customers can choose specific seats on the flight, including special requests such as window or aisle seats.

Simple Booking Flow: The system streamlines the booking process, minimizing the number of steps needed to complete a reservation, thus reducing friction for the customer.

Multi-Channel Booking: Airlines often use multiple platforms for booking, including websites, mobile apps, and call centers. The system integrates these platforms into a unified booking experience.

Objective: To simplify and expedite the booking process, making it as fast, easy, and transparent as possible for customers.

Increase Operational Efficiency for Airlines:

An effective ARS helps airlines enhance operational efficiency by:

Real-Time Inventory Management: The system allows the airline to manage seat availability, update flight schedules, and track inventory (such as seats, meal preferences, and luggage) in real time. This ensures accurate availability information is displayed to customers.

Reservation Control: Airlines can control booking quotas, overbooking, and cancellations, which helps manage seat allocation better.

Automation of Routine Tasks: The ARS automates administrative tasks such as generating e-tickets, sending booking confirmations, handling cancellations, and processing refunds.

Flight Schedule Management: The system helps airlines optimize flight schedules by analyzing demand patterns and adjusting routes, timings, and capacity accordingly.

Objective: To automate administrative tasks, optimize seat inventory management, and ensure seamless integration with operational workflows, allowing airlines to manage bookings effectively and efficiently.

Provide Personalized Customer Experience:

Customer satisfaction is at the core of any business, and an ARS aims to enhance the customer experience by offering personalized services:

Personalized Recommendations: The system can track a customer's past bookings and offer personalized flight suggestions, tailored promotions, and special offers (e.g., discounted upgrades).

User Profiles: ARS allows customers to create profiles with their personal details, preferences, loyalty program statuses, and historical booking data. This helps deliver a more customized and engaging experience.

Special Requests: The system accommodates special requests, such as vegetarian meals, extra legroom, or assistance for physically challenged passengers.

Loyalty Programs: Many ARS platforms integrate with the airline's frequent flyer or loyalty program, allowing customers to earn and redeem miles, access exclusive offers, or check their membership status directly.

Objective: To enhance the customer journey by providing a personalized, seamless, and engaging experience from search to booking and beyond.

Improve Revenue Generation:

The ARS plays a crucial role in optimizing an airline's revenue streams by:

Dynamic Pricing: The system can implement dynamic pricing models, adjusting ticket prices based on factors such as demand, seasonality, competitor pricing, and flight occupancy.

Ancillary Services: ARS often offers additional services such as extra baggage, in-flight meals, priority boarding, and seat upgrades, generating more revenue for airlines.

Cross-Selling and Up-Selling: By analyzing customer preferences and booking behavior, the system can recommend relevant add-ons, like hotel bookings, car rentals, and travel insurance, which boosts overall revenue.

Revenue Management: The ARS integrates with sophisticated revenue management systems, allowing airlines to forecast demand and optimize seat prices based on different market conditions (e.g., high vs. low season, special events).

Objective: To maximize profitability by leveraging dynamic pricing, offering value-added services, and capitalizing on customer spending potential through effective cross-selling and up-selling.

Facilitate Seamless Integration with Third-Party Systems:

An ARS must interact with various third-party systems for smooth operation:

Global Distribution Systems (GDS): The ARS integrates with GDS platforms like Amadeus, Sabre, or Travelport to access a global inventory of flight data, including partner airlines' schedules and availability.

Payment Gateways: It integrates with secure payment gateways (e.g., PayPal, Stripe, or credit card processors) to handle financial transactions and ensure safe, encrypted payment processes.

Partner Airlines: ARS systems allow for interline agreements with partner airlines, making it easier for customers to book connecting flights or transfer their bookings across different airlines.

External APIs: The system integrates with external services like flight status providers, weather services, and travel insurance companies to provide real-time data and comprehensive services.

Objective: To ensure the system can integrate with external services, partner networks, and third-party providers to offer a wide range of services and maintain operational cohesion.

Provide Real-Time Flight and Booking Updates:

The ARS ensures that users are kept up to date with real-time information related to their bookings and flight statuses:

Flight Status Updates: The system provides live updates on flight statuses, such as delays, cancellations, and gate changes, reducing customer frustration.

Booking Confirmation & Modifications: Customers receive timely notifications about their booking status, changes, or cancellations, along with e-tickets and boarding passes.

Alerts and Notifications: The system can send automatic reminders about flight timings, check-in deadlines, and special offers via email, SMS, or push notifications.

Objective: To ensure customers are always informed with the latest updates, enabling them to make decisions based on accurate, up-to-date information.

Enhance Data Security and Compliance:

Data security and regulatory compliance are critical to an ARS, especially in handling sensitive information such as customer details and payment data:

GDPR Compliance: For customers in the EU, the system ensures compliance with the General Data Protection Regulation (GDPR), guaranteeing that customer data is stored and processed securely.

PCI-DSS Compliance: For payment processing, the system complies with Payment Card Industry Data Security Standards (PCI-DSS), ensuring that financial transactions are securely handled.

User Authentication: The system uses advanced security protocols like OAuth 2.0, JWT, and two-factor authentication (2FA) to protect user accounts and prevent unauthorized access.

Objective: To provide a secure platform for customers to book flights while adhering to global data privacy and security standards.

Support Multiple Payment Methods:

In an increasingly globalized market, the ARS must offer multiple, secure payment options to cater to a diverse customer base:

Credit/Debit Card Payments: Standard credit and debit card transactions are supported, ensuring customers can pay with ease.

E-Wallets: The system integrates with digital wallets such as PayPal, Apple Pay, or Google Pay for quicker payment processing.

Cryptocurrency: Some modern ARS platforms are exploring cryptocurrency payments to appeal to tech-savvy travelers.

Objective: To provide a flexible and inclusive payment ecosystem that accommodates various payment preferences, enhancing the customer experience.

Enhance Reporting and Analytics:

Effective reporting is essential for airlines to track their performance and make data-driven decisions:

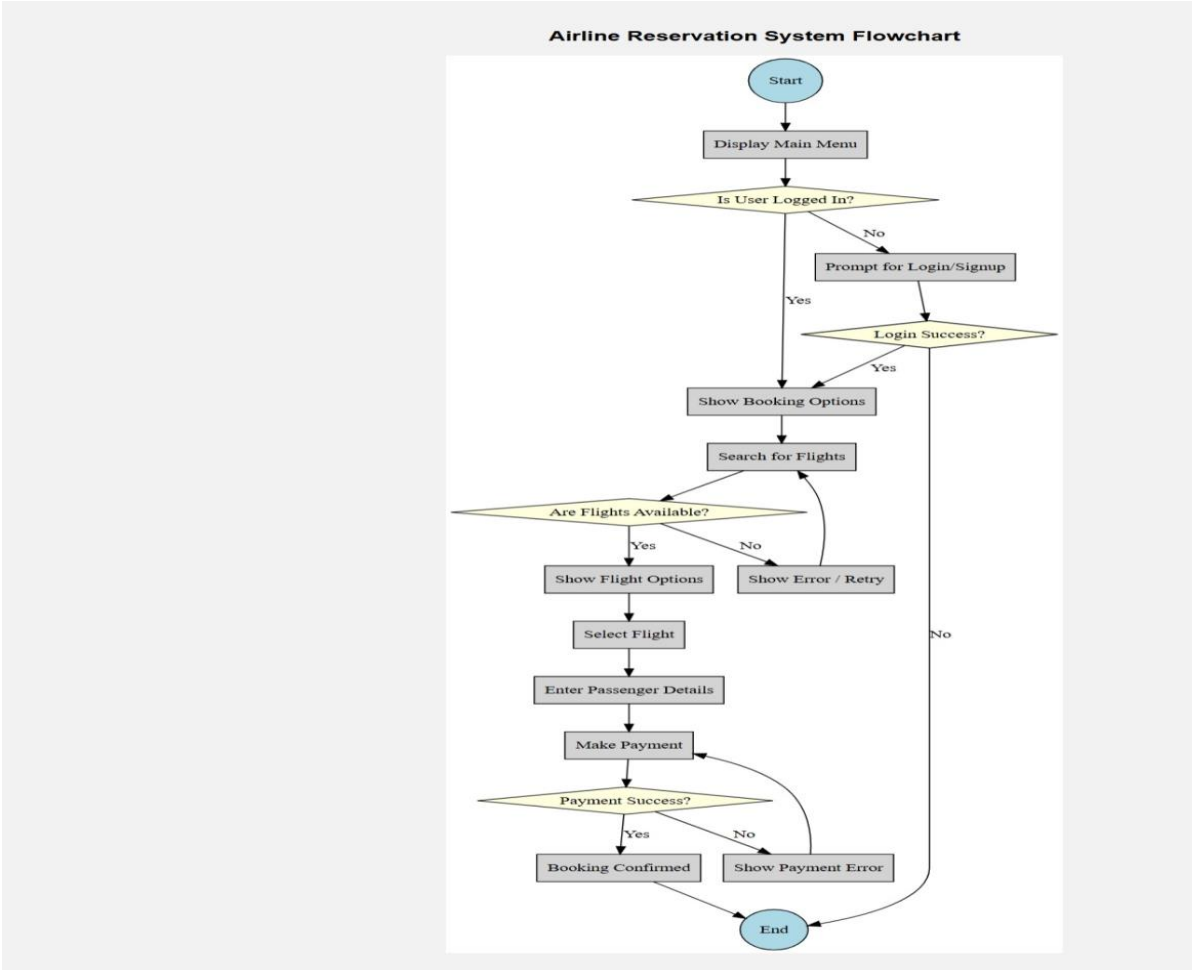
Revenue Reports: Airlines can access detailed reports about bookings, revenue per flight, seat occupancy rates, and ancillary sales.

Customer Insights: The ARS generates reports on customer behavior, preferences, and booking patterns, which can help airlines better understand their market and refine their offerings.

Operational Efficiency Metrics: Airlines can track the efficiency of their booking process, including the time taken to process reservations and cancellations.

Objective: To provide the airline with actionable insights that can be used for strategic decision-making, operational improvement, and marketing initiatives.

7.DATA FLOW DAIGRAM:



8. APPENDIX:

8.1. SOURCE CODE:

Intex.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Airline Reservation - Search</title>
  <style>
    body { font-family: Arial; background: #f0f6fc; }
   header { background: #003366; color: white; text-align: center; padding: 15px; }
    form {
     background: white; max-width: 500px; margin: 30px auto; padding: 20px;
      border-radius: 10px; box-shadow: 0 4px 8px rgba(0,0,0,0.1);
    }
    label { display: block; font-weight: bold; margin-top: 12px; }
    input, select, button {
     width: 100%; padding: 10px; margin-top: 6px; border: 1px solid #ccc; border-radius: 6px;
    }
   button { background: #003366; color: white; font-size: 16px; cursor: pointer; }
    button:hover { background: #00509e; }
  </style>
</head>
<body>

  <header>
```

```
  <h1>Airline Reservation System</h1>
</header>

<form id="searchForm">
  <label>From:</label>
  <input type="text" id="from" required>

  <label>To:</label>
  <input type="text" id="to" required>

  <label>Departure Date:</label>
  <input type="date" id="date" required>

  <label>Return Date:</label>
  <input type="date" id="return">

  <label>Class:</label>
  <select id="class">
    <option value="Economy">Economy</option>
    <option value="Business">Business</option>
    <option value="First">First</option>
  </select>

  <label>Passengers:</label>
  <input type="number" id="passengers" value="1" min="1" max="10">

  <button type="submit">Search Flights</button>
</form>

<script>
```

```
   document.getElementById("searchForm").addEventListener("submit",
function(e) {
      e.preventDefault();
      const from = document.getElementById("from").value;
      const to = document.getElementById("to").value;
      const date = document.getElementById("date").value;
      const returnDate = document.getElementById("return").value;
      const travelClass = document.getElementById("class").value;
      const passengers = document.getElementById("passengers").value;

      // Redirect to flights.html with query parameters
     window.location.href                                                    =
`flights.html?from=${from}&to=${to}&date=${date}&return=${returnDate}
&class=${travelClass}&passengers=${passengers}`;
    });
  </script>

</body>
</html>
```

Flight.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Available Flights</title>
  <style>
    body { font-family: Arial; background: #f0f6fc; }
   header { background: #003366; color: white; text-align: center; padding:
15px; }
    .flight-card {
      background: white; max-width: 500px; margin: 20px auto; padding: 20px;
      border-radius: 10px; box-shadow: 0 4px 8px rgba(0,0,0,0.1);
    }
   button { background: #003366; color: white; border: none; padding: 10px;
width: 100%; cursor: pointer; border-radius: 6px; }
    button:hover { background: #00509e; }
```

```
  </style>
</head>
<body>

  <header>
    <h1>Available Flights</h1>
  </header>

  <div id="results"></div>

  <script>
    function getParams() {
      const params = {};
     new URLSearchParams(window.location.search).forEach((val, key) => {
params[key] = val; });
      return params;
    }

    const params = getParams();

    const flights = [
      { airline: "Air India", time: "08:30 AM", price: "₹5000" },
      { airline: "IndiGo", time: "12:45 PM", price: "₹4500" },
      { airline: "SpiceJet", time: "06:15 PM", price: "₹4800" }
    ];

    let resultDiv = document.getElementById("results");
   resultDiv.innerHTML = `<h2 style="text-align:center;">Flights from
${params.from} to ${params.to} on ${params.date}</h2>`;

    flights.forEach((flight, i) => {
      resultDiv.innerHTML += `
        <div class="flight-card">
          <h3>${flight.airline}</h3>
          <p><strong>Time:</strong> ${flight.time}</p>
          <p><strong>Class:</strong> ${params.class}</p>
          <p><strong>Passengers:</strong> ${params.passengers}</p>
          <p><strong>Price:</strong> ${flight.price}</p>
         <button
onclick="bookFlight('${flight.airline}','${flight.time}','${flight.price}')">Book
Now</button>
        </div>
      `;
    });
```

```
    function bookFlight(airline, time, price) {
     window.location.href =
`booking.html?from=${params.from}&to=${params.to}&date=${params.dat
e}&class=${params.class}&passengers=${params.passengers}&airline=${ai
rline}&time=${time}&price=${price}`;
    }
  </script>

</body>
</html>
```

Booking.html

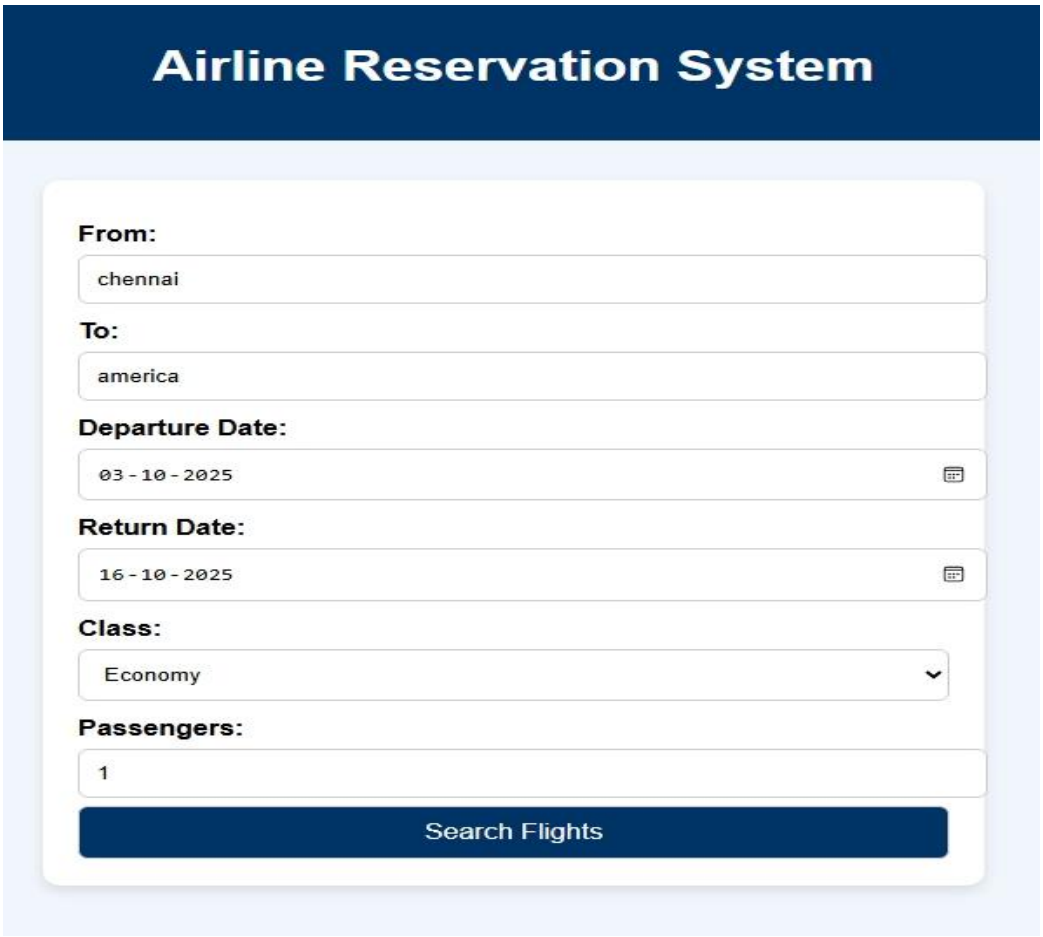
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Booking Form</title>
  <style>
    body { font-family: Arial; background: #f0f6fc; }
   header { background: #003366; color: white; text-align: center; padding:
15px; }
    form, .confirmation {
     background: white; max-width: 500px; margin: 30px auto; padding: 20px;
      border-radius: 10px; box-shadow: 0 4px 8px rgba(0,0,0,0.1);
    }
    label { display: block; margin-top: 12px; font-weight: bold; }
    input, button {
     width: 100%; padding: 10px; margin-top: 6px; border: 1px solid #ccc;
border-radius: 6px;
    }
   button { background: #003366; color: white; font-size: 16px; cursor:
pointer; }
    button:hover { background: #00509e; }
   .confirmation { background: #d4edda; color: #155724; border: 1px solid
#c3e6cb; }
```

```
  </style>
</head>
<body>

  <header>
    <h1>Booking Form</h1>
  </header>

  <div id="bookingSection"></div>

  <script>
    function getParams() {
      const params = {};
     new URLSearchParams(window.location.search).forEach((val, key) => {
params[key] = val; });
      return params;
    }

    const params = getParams();
    let bookingDiv = document.getElementById("bookingSection");

    bookingDiv.innerHTML = `
      <form id="bookingForm">
        <h3>Booking for ${params.airline} (${params.time})</h3>
       <p><strong>From:</strong> ${params.from} → <strong>To:</strong>
${params.to}</p>
        <p><strong>Date:</strong> ${params.date}</p>
        <p><strong>Class:</strong> ${params.class}</p>
        <p><strong>Price:</strong> ${params.price}</p>

        <label>Passenger Name:</label>
        <input type="text" id="name" required>

        <label>Age:</label>
        <input type="number" id="age" min="1" required>
```

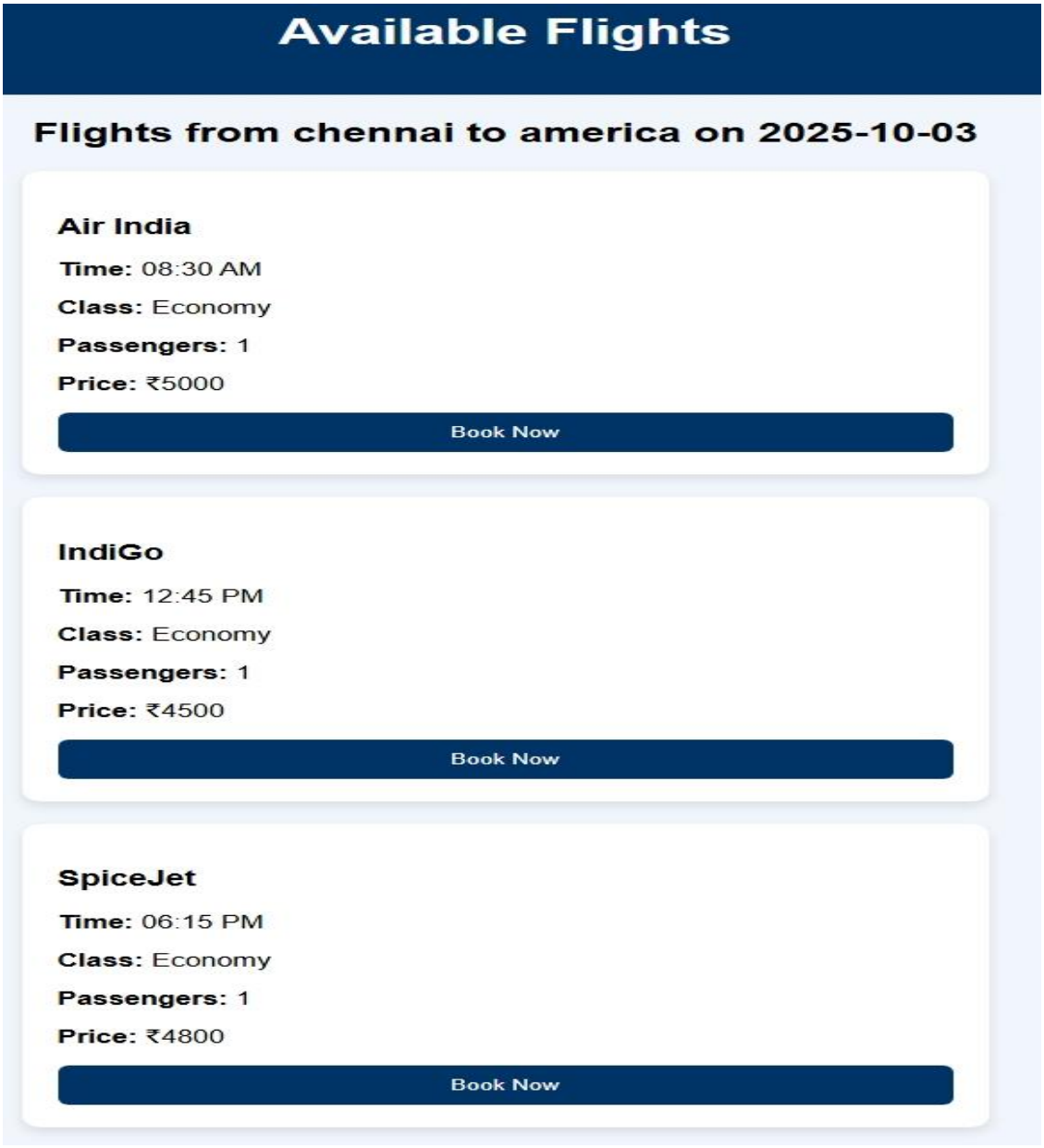
```
        <label>Email:</label>
        <input type="email" id="email" required>

        <button type="submit">Confirm Booking</button>
      </form>
    `;

    document.getElementById("bookingForm").addEventListener("submit",
  function(e) {
      e.preventDefault();

      const name = document.getElementById("name").value;
      const age = document.getElementById("age").value;
      const email = document.getElementById("email").value;

      bookingDiv.innerHTML = `
        <div class="confirmation">
          <h3>✅ Booking Confirmed!</h3>
          <p><strong>Name:</strong> ${name} (${age} years)</p>
          <p><strong>Email:</strong> ${email}</p>
          <p><strong>Airline:</strong> ${params.airline}</p>
          <p><strong>Time:</strong> ${params.time}</p>
          <p><strong>From:</strong> ${params.from} → ${params.to}</p>
          <p><strong>Date:</strong> ${params.date}</p>
          <p><strong>Class:</strong> ${params.class}</p>
          <p><strong>Passengers:</strong> ${params.passengers}</p>
          <p><strong>Price:</strong> ${params.price}</p>
        </div>
      `;
    });
  </script>
  </body>
</html>
```

8.2. SCREEN LAYOUT:

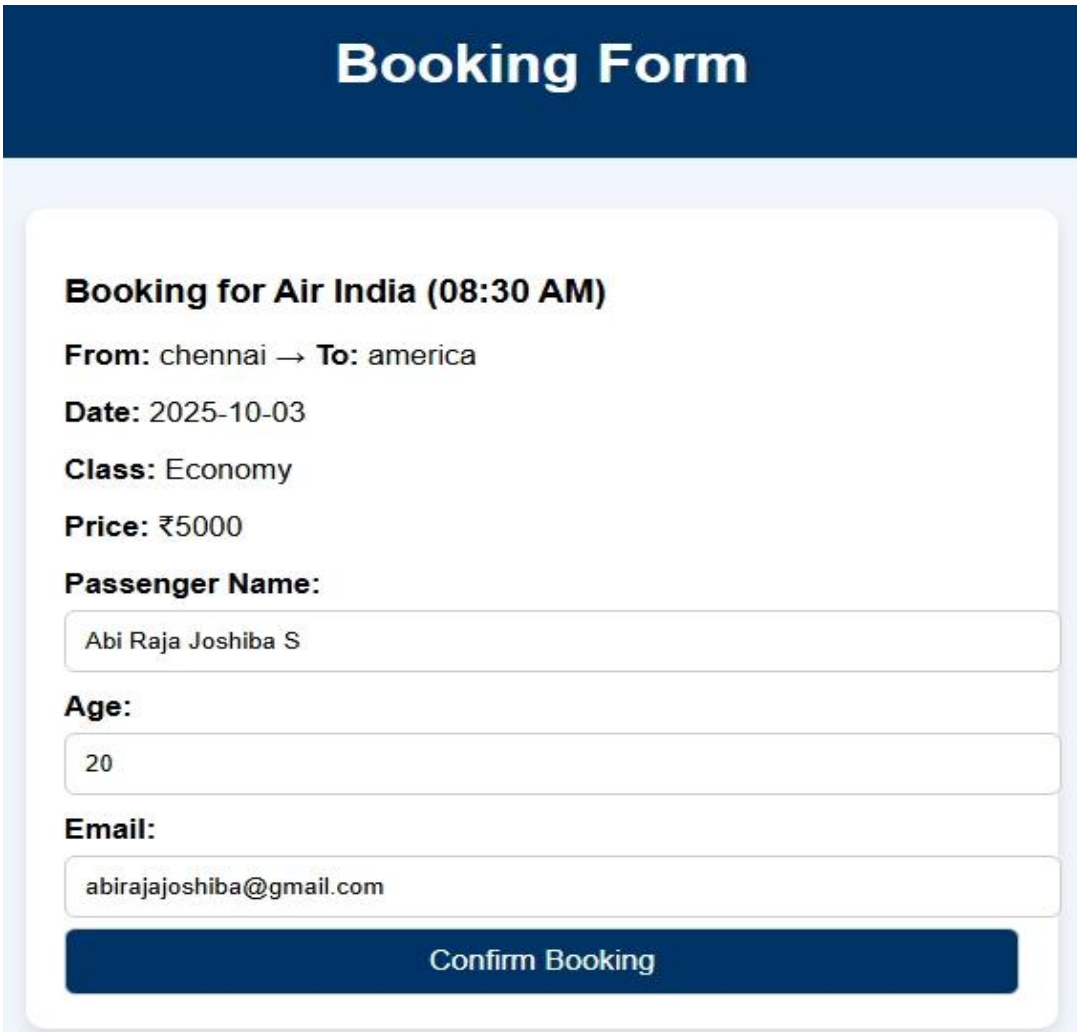
Index:

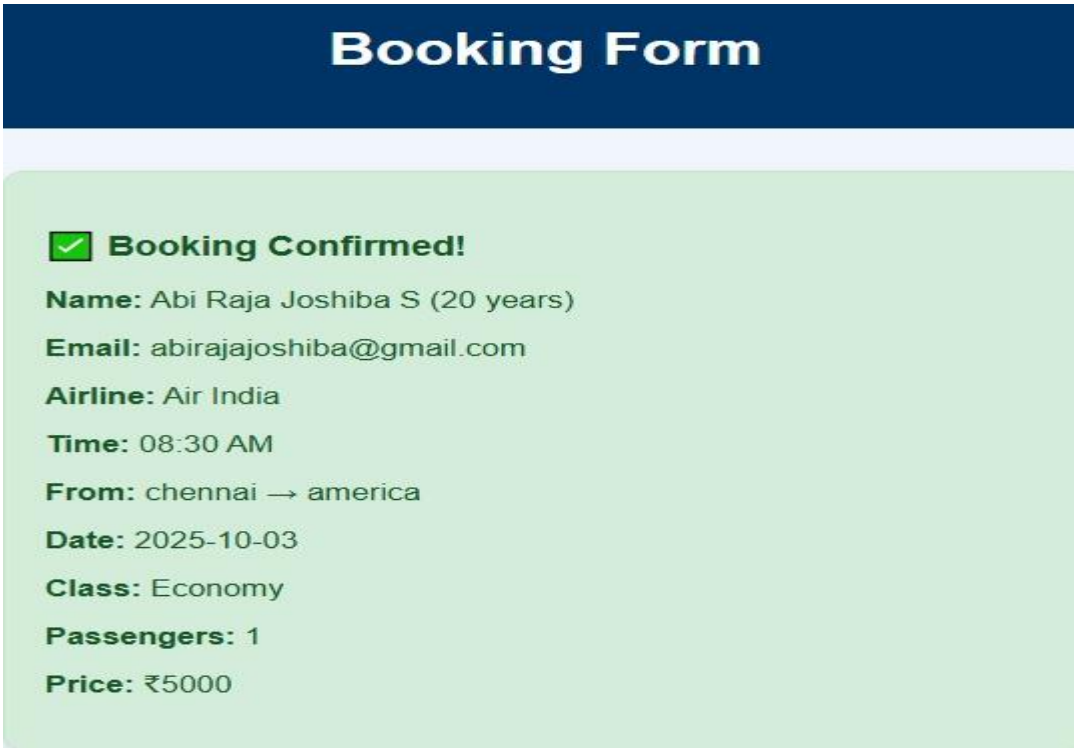


Flights:


Available Flights
Flights from chennai to america on 2025-10-03
Air India
Time: 08:30 AM
Class: Economy
Passengers: 1
Price: ₹5000
Book Now
IndiGo
Time: 12:45 PM
Class: Economy
Passengers: 1
Price: ₹4500
Book Now
SpiceJet
Time: 06:15 PM
Class: Economy
Passengers: 1
Price: ₹4800
Book Now

Booking:


Booking Form
Booking for Air India (08:30 AM)
From: chennai → To: america
Date: 2025-10-03
Class: Economy
Price: ₹5000
Passenger Name:
Abi Raja Joshiba S
Age:
20
Email:
abirajajoshiba@gmail.com
Confirm Booking


Booking Form
Booking Confirmed!
Name: Abi Raja Joshiba S (20 years)
Email: abirajajoshiba@gmail.com
Airline: Air India
Time: 08:30 AM
From: chennai → america
Date: 2025-10-03
Class: Economy
Passengers: 1
Price: ₹5000

9. Future enhancement:

1. Sustainability

- Electric/Hybrid Aircraft: Traditional jet engines produce a lot of CO_2. Electric or hybrid planes, powered partly or fully by batteries or electric motors, can drastically cut emissions. While still in development, these could revolutionize short- and medium-haul flights.
- Sustainable Aviation Fuels (SAF): Made from renewable sources like plant oils or waste, SAF burns cleaner than fossil fuels and can reduce carbon footprint without needing new engines. Airlines are increasingly investing in SAF production and use.
- Zero-Waste Initiatives: Airlines are working to cut single-use plastics, replace disposable items with compostable alternatives, and improve recycling onboard and at airports to reduce environmental impact.

2. Passenger Experience

- Personalized Travel: Using AI and big data, airlines can offer tailored services—like meal preferences, entertainment options, and even travel suggestions—based on individual passenger profiles and past behaviors.
- Biometric Boarding: Facial recognition technology speeds up check-in, security, and boarding by verifying identity without physical documents or boarding passes, reducing lines and contact points.
- VR/AR Entertainment: Instead of traditional screens, passengers might enjoy immersive VR environments or AR overlays that make flights more entertaining or even educational.
- Smart Seats: Future seats could adjust firmness, temperature, or even monitor passenger health (heart rate, oxygen levels) to improve comfort and well-being. Some may have massage functions or connectivity to personal devices.
- High-Speed Wi-Fi: Reliable, fast internet on flights will allow streaming, video calls, and remote work without frustration, making flights more productive and enjoyable.

3. Operational Efficiency

- AI Predictive Maintenance: Sensors on aircraft monitor engine and component health in real-time. AI analyzes this data to predict failures before they happen, allowing timely repairs and avoiding delays or cancellations.
- Automated Baggage Tracking: Using IoT tags and RFID, baggage can be tracked accurately throughout the journey, reducing lost luggage and giving passengers real-time updates.
- Optimized Flight Routes: AI and big data help select the most fuel-efficient and least congested routes, saving fuel and time while reducing emissions.
- Digital Twins: Virtual replicas of aircraft and operations simulate different scenarios, helping airlines optimize maintenance schedules, fuel use, and operational decisions.

4. Safety & Health

- Enhanced Air Filtration: Advanced HEPA filters combined with UV sterilization kill viruses and bacteria, improving cabin air quality and reducing disease spread.
- Touchless Interfaces: To minimize germ transmission, airports and planes will have more touchless kiosks, doors, and controls activated by voice, gesture, or mobile devices.
- Health Monitoring: Wearables integrated with airline apps could monitor passenger health during flights and alert crew if medical assistance is needed.

5. New Business Models

- Subscription Services: Instead of buying individual tickets, frequent flyers might subscribe monthly for a set number of flights or unlimited travel within a network, similar to streaming services.
- Flexible Ticketing: Dynamic pricing models offer more customization — passengers can add or drop services (like extra luggage or priority boarding) easily, tailoring their journey and price.
- Integrated Travel: Airlines may partner with high-speed ground transport (like hyperloop or maglev trains) for seamless door-to-door travel, combining air and land transit in one booking.

10. Conclusion:

An Airline Reservation System (ARS) is a critical backbone of modern air travel, serving as the centralized platform where flights are scheduled, seats are allocated, bookings are made, and customer information is managed. Over the years, ARS has evolved from manual processes to highly sophisticated, automated digital systems that enhance efficiency, accuracy, and customer satisfaction.

In conclusion, the Airline Reservation System is the linchpin of airline operations, directly influencing revenue, customer experience, and operational efficiency. Its continuous evolution, driven by advancements in technology and changing customer expectations, ensures airlines remain competitive in a complex and dynamic industry. Airlines that invest in robust, secure, and flexible ARS platforms will be better equipped to innovate, adapt, and thrive in the future of air travel.

11.Reference:

1.GeeksforGeeks | Your All-in-One Learning Portal

2.ChatGPT

3.W3Schools Online Web Tutorials

4.Wikipedia, the free encyclopedia