A

PROJECT REPORT ON

**AIRLINE RESERVATION SYSTEM**

SUBMITTED BY

**Ms. Nausheen Mohammad Hanif Shaikh**

SUBMITTED TO

**SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE**

IN PARTIAL FULFILLMENT OF DEGREE

**MASTERS OF COMPUTER APPLICATION**

**(SEM- 1)**

UNDER THE GUIDANCE OF

**MS. SAMIKSHA YEOLA**

Through,



**Sadhu Vaswani Institute of Management Studies For Girls, Koregaon Park, Pune-411001**

2024-2025

**DECLARTION BY STUDENT**

To,

The Director,

SVIMS Koregaon Park,

I, undersigned hereby declare that this project titled, **“AIRLINE RESERVATION SYSTEM”** written and submitted by me to SPPU, Pune, in partial fulfilment of the requirement of the award of the degree of **MASTER OF COMPUTER APPLICATION (MCA-1)** under the guidance of **Ms. Samiksha Yeola**, is my original work.

I further declare that to the best of my knowledge and belief, this project has not been submitted to this or any other University or Institute for the award of any Degree

**Place: Pune**

**Date:**

**(Nausheen Mohammad Hanif Shaikh)**

**ACKNOWLEDGEMENT**

I extend my sincere gratitude to **Dr. B.H. Nanwani, Mrs. Neeta Raskar and Ms. Samiksha Yeola** for allowing me to carry out the study and for their constant and encouragement, valuable suggestions and guidance during the research work.

I extent my special thanks to professor **Ms. Samiksha Yeola** and **Dr. Shveti Chandan** for their kind co-operation and inspiration.

I extend my special gratitute to my dearest family members and friends who encouraged me and motivated me to complete the project report.

**Place: Pune**

**Date:**

**(Nausheen Mohammad Hanif Shaikh)**

**Index**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Contents** | **Pg. No** |
| **1.** | **Introduction** | **5** |
| **2.** | **Client/Organization Profile** | **6** |
| **3.** | **Need for System** | **7** |
| **4.** | **Scope & Feasibility of Work** | **8-9** |
| **5.** | **Operating Environment – H/w & S/w** | **9-10** |
| **6.** | **Architecture of system** | **11-13** |
| **7.** | **Detail Description of Technology Used** | **13-15** |
| **8.** | **Proposed System** | **16** |
| **9.** | **Objectives of System** | **16** |
| **10.** | **User Requirements** | **16-17** |
| **11.** | **DFD** | **18** |
| **12.** | **Table specifications (Database)** | **19-21** |
| **13.** | **ERD** | **22** |
| **14.** | **Class Diagram** | **23** |
| **15.** | **Use Case Diagrams** | **23** |
| **16.** | **Web Site Map Diagram (if Website )** | **24** |
| **17.** | **User Interface Design (Screens etc.)** | **24** |
| **18.** | **Limitations** | **28** |
| **19.** | **Future enhancement** | **29** |
| **20.** | **Bibliography** | **29-30** |
| **21.** | **Annexure** | **31** |

**Introduction**

An Airline Reservation System using PHP, HTML, and CSS is a web-based application designed to streamline flight booking and reservation processes. It enables users to search for flights, book tickets, and manage their reservations with ease.

The system is structured using HTML for creating the interface and web forms, ensuring a clear and user-friendly layout. CSS is used to enhance the visual appearance and provide a professional design for the application. PHP, as the server-side scripting language, handles backend processes such as validating user inputs, managing database interactions, and processing bookings dynamically.

**Core Features**

1. **Flight Search:** Users can search for flights by specifying source, destination, travel dates, and preferences.
2. **User Management:** Includes features for user registration, login, and profile management.
3. **Booking System:** Facilitates seat selection, booking confirmation, and e-ticket generation.
4. **Database Connectivity:** Uses MySQL to store flight details, user data, and booking records.
5. **Admin Panel:** Allows airline staff to manage flights, schedules, and seat availability.

**Technical Workflow**

* **Frontend:** Built with HTML for structure and CSS for styling, ensuring an intuitive user experience.
* **Backend:** PHP processes user actions, validates inputs, and interacts with the database.
* **Database:** MySQL stores and retrieves dynamic data such as flight schedules, bookings, and user details.

**Client/Organization Profile**

**Organization Name:**

SkyRide Tickets

**Overview:**

SkyRide Tickets Solutions specializes in developing cutting-edge digital solutions for the travel and transportation industry. With expertise in creating seamless, user-friendly platforms, SkyRide Tickets empowers airlines and passengers with robust tools for managing reservations, ensuring operational efficiency, and enhancing customer experiences.

**Project: Airline Reservation System**

**Description:**

The Airline Reservation System is a comprehensive, web-based application designed to streamline booking and management processes for airlines. Built using **PHP**, **CSS**, and **HTML**, this system offers a modern and responsive interface, ensuring ease of use for both customers and airline staff.

**Key Features:**

1. **User-Friendly Booking Portal:**
   * Search and filter flights by destination, date, and airline.
   * Real-time seat availability display.
   * Secure ticket booking with instant confirmation.
2. **Administrative Dashboard:**
   * Flight schedule management.
   * Real-time monitoring of bookings and cancellations.
   * Access to customer and revenue analytics.

3. **Customer Account Management:**

* Profile creation and personalized booking history.
* Manage tickets (reschedule/cancel).
* Notifications for schedule changes and promotions.

4. **Responsive Design:**

* Cross-device compatibility (desktop, tablet, and mobile).
* Optimized for modern web browsers.

**Need for System**

An **Airline Reservation System** addresses the critical needs of airlines and their customers by automating and optimizing the ticketing and booking process. Below are the key reasons such a system is essential:

**1. Efficient Flight Management**

Airlines operate across multiple routes and schedules, requiring a centralized system to manage:

* Flight schedules.
* Seat availability.
* Real-time updates for changes or cancellations.

**2. Improved Customer Experience**

Customers demand seamless booking experiences with features like:

* Online flight search and booking.
* Quick and secure payments.
* Self-service options for rescheduling or cancellations.

The system ensures convenience and enhances customer satisfaction.

**3. Cost-Effective Operations**

Manual booking processes are prone to errors and inefficiencies. An automated system reduces operational costs by:

* Minimizing human intervention.
* Streamlining ticket issuance and record management.
* Reducing the need for extensive customer service teams.

**Conclusion:**  
The Airline Reservation System is a vital tool for modern aviation businesses. It improves efficiency, customer satisfaction, and overall business performance, making it an indispensable asset for airlines.

**Scope & Feasibility of Work**

**Scope of Work**

The Airline Reservation System (ARS) project encompasses the design, development, testing, and deployment of a fully functional web-based platform. Below are the key components and deliverables:

**1. System Features**

1. **Customer Module**
   * Flight search with filters (destination, date, price, etc.).
   * Real-time seat availability and selection.
   * User registration, login, and profile management.
   * Booking history, ticket management, and cancellation.
2. **Payment Integration**
   * Secure payment gateway integration.
   * Multiple payment options (credit/debit cards, digital wallets).
3. **System Management**
   * Centralized database for passenger, booking, and flight information.
   * Role-based access control for system security.
   * Notifications via email or SMS for booking confirmations, changes, etc.

**Feasibility of Work**

**1. Technical Feasibility**

* **Technology Stack:**  
  The use of PHP, HTML, CSS, and MySQL ensures that the system is lightweight, scalable, and cost-effective. These technologies are widely used and have robust community support.
* **Integration:**  
  Feasible integration with payment gateways, third-party APIs (for real-time flight data), and notification services.

**2. Economic Feasibility**

* **Cost-Effective Development:**  
  Open-source technologies like PHP and MySQL reduce software licensing costs.
* **ROI:**  
  The system will significantly reduce operational costs while increasing efficiency, leading to a high return on investment.

**3. Operational Feasibility**

* **Ease of Use:**  
  The system will have an intuitive design, ensuring quick adoption by customers and staff.
* **Training:**  
  Minimal training will be required for administrative users due to the system's simplicity.

**Conclusion:**The project is technically and economically feasible, with a clear scope that addresses the needs of airlines and customers. Proper implementation will result in a system that is reliable, user-friendly, and aligned with the industry’s operational demands.

**Operating Environment – H/w & S/w**

The Airline Reservation System requires a robust and scalable operating environment to ensure efficient performance and seamless operation. Below are the recommended **hardware** and **software** specifications for deploying and running the system.

**1. Hardware Requirements**

**Server-Side Hardware**

* **Processor:**  
  Intel Xeon Processor or equivalent (Quad-core or higher).
* **RAM:**  
  8 GB or higher (16 GB recommended for high traffic).
* **Storage:**  
  500 GB SSD (scalable storage options for data growth).
* **Network:**  
  High-speed internet connection (minimum 1 Gbps bandwidth).
* **Backup:**  
  External storage for data backup (minimum 1 TB).
* **Redundancy:**  
  Load balancer and failover system for high availability.

**Client-Side Hardware**

* **Processor:**  
  Dual-core processor or higher.
* **RAM:**  
  4 GB or higher.
* **Storage:**  
  100 MB free space for cache and temporary files.
* **Display:**  
  Minimum resolution of 1280x720 (responsive design ensures compatibility).

**2. Software Requirements**

**Server-Side Software**

* **Operating System:**
  + Linux (Ubuntu 20.04 LTS or CentOS 7 preferred).
  + Windows Server (optional, if needed for compatibility).
* **Web Server:**
  + Apache HTTP Server 2.4 or higher.
  + Alternatively, NGINX for lightweight and faster performance.
* **Database Server:**  
  MySQL 8.0 or higher for data management.
* **Programming Environment:**  
  PHP 8.0 or higher with necessary extensions (e.g., PDO, cURL).
* **Security Tools:**
  + SSL/TLS for secure data transfer.
  + Firewall tools such as UFW or iptables.
* **Backup Tools:**  
  Rsync or similar for periodic data backups.

**Client-Side Software**

* **Browser Compatibility:**
  + Google Chrome (latest version).
  + Mozilla Firefox (latest version).
  + Microsoft Edge (latest version).
  + Safari (latest version).
* **Other Tools:**  
  JavaScript enabled and updated browser for interactive features.

**Conclusion**

The system's operating environment should be robust, scalable, and secure to handle airline operations effectively. By meeting these hardware and software requirements, the system will ensure smooth performance, high availability, and future growth potential.

**Architecture of system**

The Airline Reservation System is built using a **3-tier architecture** for modularity, scalability, and maintainability. Below is a detailed breakdown of its architecture:

**1. Overview of 3-Tier Architecture**

1. **Presentation Layer (Frontend)**
   * The user interface for customers and administrators.
   * Handles interaction between users and the system.
2. **Business Logic Layer (Backend)**
   * Implements core functionalities such as booking, cancellations, and data processing.
   * Ensures security, validation, and business rule enforcement.
3. **Data Layer (Database)**
   * Manages data storage, retrieval, and updates for flights, users, and bookings.
   * Ensures data integrity and consistency.

**2. Detailed System Architecture**

**A. Presentation Layer**

* **Technologies Used:**  
  HTML, CSS, JavaScript (with frameworks like Bootstrap or Tailwind CSS for responsive design).
* **Components:**
  1. **Customer Interface:**
     + Flight search and booking form.
     + User registration/login interface.
     + Booking history and ticket management pages.
  2. **Admin Interface:**
     + Flight schedule management dashboard.
     + Booking and cancellation overview.
     + Analytics and reporting tools.
  3. **Responsive Design:**
     + Cross-device compatibility (desktop, tablet, and mobile).

**B. Business Logic Layer**

* **Technologies Used:**  
  PHP for server-side scripting, integrated with REST APIs for external services (e.g., payment gateway).
* **Key Functionalities:**
  1. **Flight Management:**
     + CRUD (Create, Read, Update, Delete) operations for flight schedules.
  2. **Booking Management:**
     + Handle seat reservations, ticket issuance, and cancellations.
  3. **Payment Processing:**
     + Secure integration with payment gateways.
  4. **User Management:**
     + Authentication, authorization, and profile management.
  5. **Notifications:**
     + Email/SMS notifications for booking confirmation and updates.
  6. **Business Rules:**
     + Enforce rules such as overbooking prevention and refund policies.

**C. Data Layer**

* **Technologies Used:**  
  MySQL (or other relational databases).
* **Key Data Structures:**
  1. **Flights Table:**
     + Flight ID, origin, destination, schedule, aircraft details, seat capacity.
  2. **Bookings Table:**
     + Booking ID, flight ID, customer ID, seat details, status (confirmed/canceled).
  3. **Users Table:**
     + Customer ID, personal details, login credentials.
  4. **Payments Table:**
     + Payment ID, amount, payment status, booking reference.
* **Data Flow:**
  1. Data is fetched or stored using SQL queries from the business logic layer.

**3. System Workflow**

**Step 1: Flight Search and Booking**

1. User searches for flights via the frontend.
2. Business logic retrieves flight options from the database and displays results.
3. User selects a flight and provides details for booking.
4. Booking data is validated and saved in the database.

**Step 2: Administration**

1. Admin users log in via the admin dashboard.
2. Admins can create/update flight schedules, monitor bookings, and generate reports.

**Detail Description of Technology Used**

The Airline Reservation System employs a stack of technologies designed for functionality, efficiency, and scalability. Below is a detailed description of each technology used in the system:

**1. Frontend Technologies**

The **frontend** is the user interface layer of the system, responsible for the user experience and interaction.

**a. HTML (HyperText Markup Language)**

* **Purpose:**
  + Serves as the backbone of the web application, structuring the content and layout of the website.
  + Defines elements such as forms, buttons, tables, and navigation bars.
* **Features Used:**
  + Semantic tags (e.g., <header>, <footer>, <article>, <section>).
  + Forms for flight search and booking input.

**b. CSS (Cascading Style Sheets)**

* **Purpose:**
  + Provides styling for the web pages to ensure they are visually appealing and user-friendly.
  + Handles layout, color schemes, typography, and responsiveness.
* **Features Used:**
  + Flexbox and Grid for responsive layouts.
  + Media queries for cross-device compatibility.
  + Frameworks like **Bootstrap** (optional) for consistent design patterns.

**c. JavaScript**

* **Purpose:**
  + Enables interactivity and dynamic content updates without refreshing the page.
  + Enhances user experience with real-time data validation and updates.
* **Features Used:**
  + AJAX for asynchronous server requests (e.g., live seat availability checks).
  + Form validation to ensure correct data input before submission.
  + Libraries or frameworks like **jQuery** (optional) for simplified DOM manipulation.

**2. Backend Technologies**

The **backend** handles the business logic, database operations, and server-side functionalities.

**a. PHP (Hypertext Preprocessor)**

* **Purpose:**
  + Serves as the server-side scripting language to handle requests, process data, and interact with the database.
  + Executes core business logic like booking, payment handling, and flight management.
* **Features Used:**
  + Database interaction using PDO (PHP Data Objects).
  + Form handling for user inputs (e.g., search, booking, cancellations).
  + Session management for secure user authentication.
* **Advantages:**
  + Open-source and widely supported.
  + Highly compatible with HTML and MySQL.

**3. Database Technologies**

The **database** stores and retrieves all system data, ensuring integrity and availability.

**a. MySQL**

* **Purpose:**
  + Relational database management system for structured data storage and retrieval.
  + Handles user details, flight schedules, bookings, payments, and analytics.
* **Features Used:**
  + Primary and foreign keys to maintain data relationships.
  + SQL queries for CRUD (Create, Read, Update, Delete) operations.
  + Indexing for optimized search performance.
* **Advantages:**
  + Scalable and reliable for large datasets.
  + Open-source with robust community support.

**Conclusion**

The combination of these technologies ensures the system is:

* **User-Friendly:** Modern frontend ensures an excellent user experience.
* **Robust:** PHP and MySQL provide reliable backend and data management.
* **Secure:** Encrypted communications and secure coding practices protect sensitive data.
* **Scalable:** The architecture allows for future growth and feature integration.

**Proposed System**

The proposed **Airline Reservation System (ARS)** is a web-based solution designed to manage airline bookings, streamline operations, and enhance customer experience. It automates the ticket booking process, provides real-time information on flight availability, and integrates features for secure payment, user management, and administrative controls.

**Objectives of the System**

The primary objectives of the proposed system are:

**1. Automate the Reservation Process**

* Eliminate manual booking processes and ensure quick, error-free ticketing.
* Provide real-time availability of flights and seats.

**2. Enhance Customer Experience**

* Offer an intuitive and user-friendly interface for customers to search, book, cancel, or modify reservations.
* Ensure 24/7 access to booking services on multiple devices.

**3. Improve Operational Efficiency**

* Simplify administrative tasks such as managing flight schedules, monitoring bookings, and generating reports.
* Minimize overbooking and double-booking errors.

**4. Secure Payment and Data Handling**

* Provide secure payment options with encryption for customer transactions.
* Safeguard sensitive customer and airline data with robust security measures.

**User Requirements**

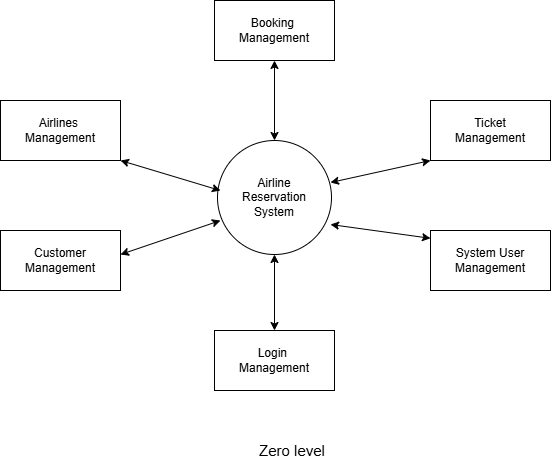
The proposed system serves two primary user groups: **Customers** and **Administrators**. Below are their requirements:

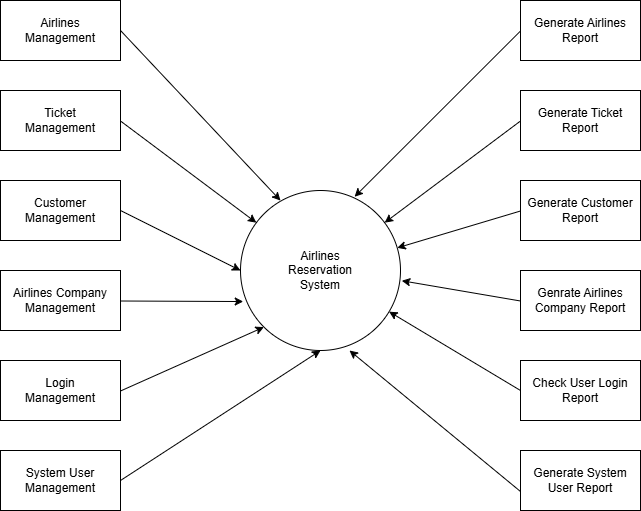
**1. Customer Requirements**

* **Flight Search and Booking**:  
  Ability to search flights by date, destination, and class, and book tickets seamlessly.
* **Seat Selection**:  
  Interactive seat maps for customers to select preferred seats.
* **Account Management**:  
  Options for registration, login, and profile management.
* **Booking History**:  
  View, modify, or cancel previous bookings.
* **Payment Options**:  
  Secure payment gateway supporting multiple payment methods.
* **Notifications**:  
  Receive booking confirmations, flight updates, and reminders via email or SMS.

**DFD**

**Zero level- DFD**



**First Level- DFD**

**Table specifications (Database)**

**admindatabase**

| **Column** | **Type** | **Null** |
| --- | --- | --- |
| UserID | int(10) | No |
| Name | varchar(50) | No |
| Email | varchar(70) | No |
| Gender | varchar(10) | No |
| password | varchar(50) | No |
| DoB | date | No |
| Joined\_on | timestamp | No |

**bustransactions**

| **Column** | **Type** | **Null** |
| --- | --- | --- |
| T\_No. *(Primary)* | int(10) | No |
| email | varchar(50) | No |
| source | varchar(50) | No |
| dest | varchar(50) | No |
| Name | varchar(50) | No |
| Bus\_No | int(11) | No |
| NoOfpass | int(11) | No |
| card\_no | int(25) | No |
| expmonth | int(2) | No |
| expyear | int(2) | No |
| cvv | int(3) | No |
| pin | int(4) | No |
| Date | datetime | No |
| Amt | int(10) | No |

**Indexes**

| **Keyname** | **Type** | **Unique** | **Packed** |
| --- | --- | --- | --- |
| PRIMARY | BTREE | Yes | No |

**price**

| **Column** | **Type** | **Null** | **Default** |
| --- | --- | --- | --- |
| source | varchar(50) | No |  |
| dest | varchar(50) | No |  |
| class | varchar(25) | No |  |
| type | varchar(25) | No |  |
| Price | int(50) | No |  |
| Route | varchar(25) | Yes | *NULL* |

**pricebus**

| **Column** | **Type** | **Null** | **Default** |
| --- | --- | --- | --- |
| No. | int(15) | No |  |
| Bus\_No. | int(50) | No |  |
| source | varchar(50) | No |  |
| dest | varchar(50) | No |  |
| Price | int(50) | No |  |

**traintt**

| **Column** | **Type** | **Null** | **Default** |
| --- | --- | --- | --- |
| Mumbai | varchar(5) | Yes | *NULL* |
| Pune | varchar(5) | Yes | *NULL* |
| Chennai | varchar(5) | Yes | *NULL* |
| Kolkata | varchar(5) | Yes | *NULL* |
| J&K | varchar(5) | Yes | *NULL* |
| Goa | varchar(5) | Yes | *NULL* |
| Surat | varchar(5) | Yes | *NULL* |
| Karnataka | varchar(5) | Yes | *NULL* |
| Kerala | varchar(5) | Yes | *NULL* |
| Assam | varchar(5) | Yes | *NULL* |
| Hyderabad | varchar(5) | Yes | *NULL* |
| Delhi | varchar(5) | Yes | *NULL* |
| Bangalore | varchar(5) | Yes | *NULL* |
| Shimla | varchar(5) | Yes | *NULL* |

**transactions**

| **Column** | **Type** | **Null** |
| --- | --- | --- |
| T\_No. *(Primary)* | int(10) | No |
| email | varchar(50) | No |
| source | varchar(50) | No |
| dest | varchar(50) | No |
| Name | varchar(50) | No |
| Class | varchar(25) | No |
| Type | varchar(25) | No |
| NoOfpass | int(11) | No |
| card\_no | int(25) | No |
| expmonth | int(2) | No |
| expyear | int(2) | No |
| cvv | int(3) | No |
| pin | int(4) | No |
| Date | datetime | No |
| Amt | int(10) | No |
| Route | varchar(25) | No |

**Indexes**

| **Keyname** | **Type** | **Unique** | **Packed** |
| --- | --- | --- | --- |
| PRIMARY | BTREE | Yes | No |

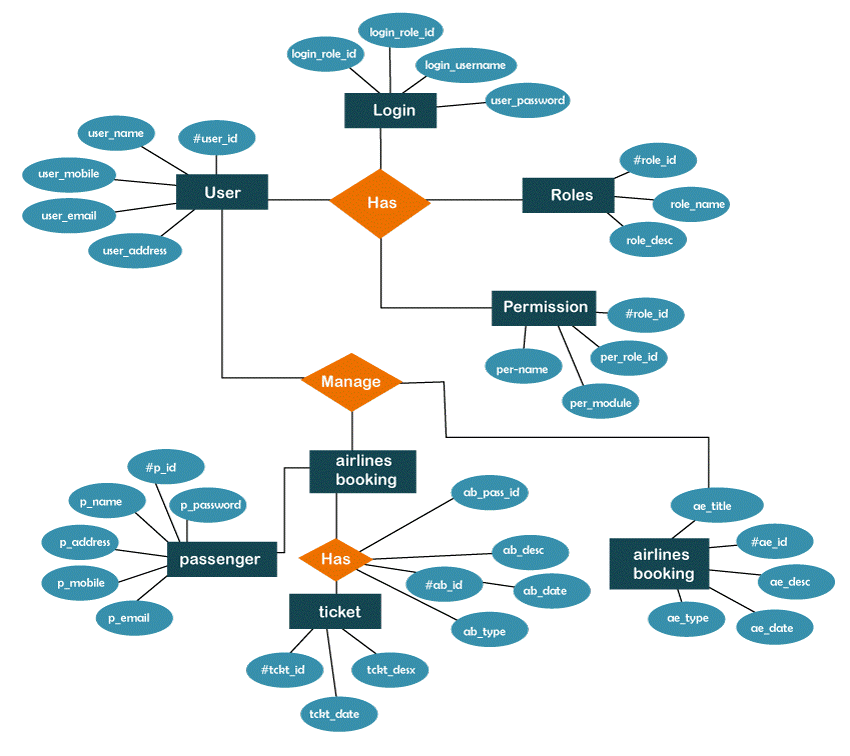
**userdatabase**

| **Column** | **Type** | **Null** |
| --- | --- | --- |
| UserID *(Primary)* | int(10) | No |
| Name | varchar(50) | No |
| Email | varchar(70) | No |
| Gender | varchar(10) | No |
| password | varchar(50) | No |
| DoB | date | No |
| Phone | varchar(10) | No |
| Joined\_on | timestamp | No |

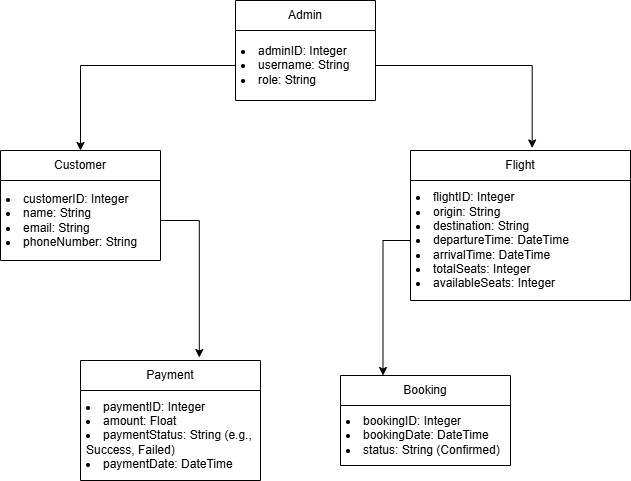
**Indexes**

| **Keyname** | **Type** | **Unique** | **Packed** | **Column** |
| --- | --- | --- | --- | --- |
| PRIMARY | BTREE | Yes | No | UserID |
| Email | BTREE | Yes | No | Email |

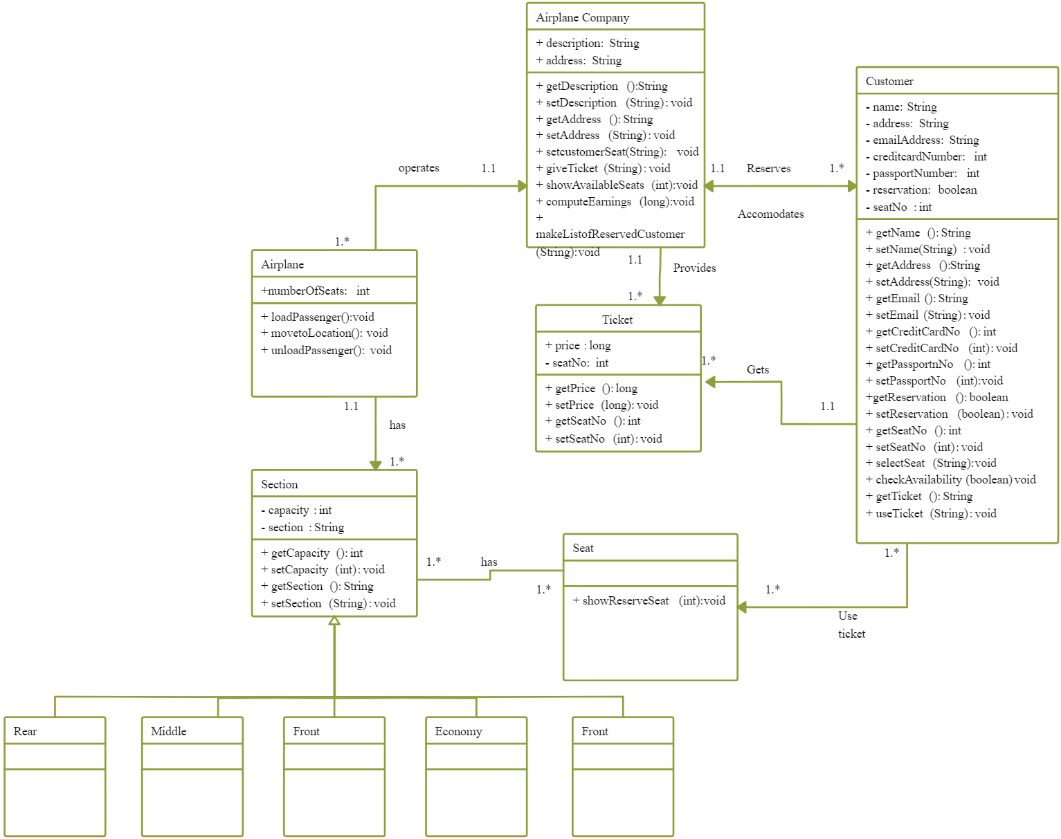
**ERD**



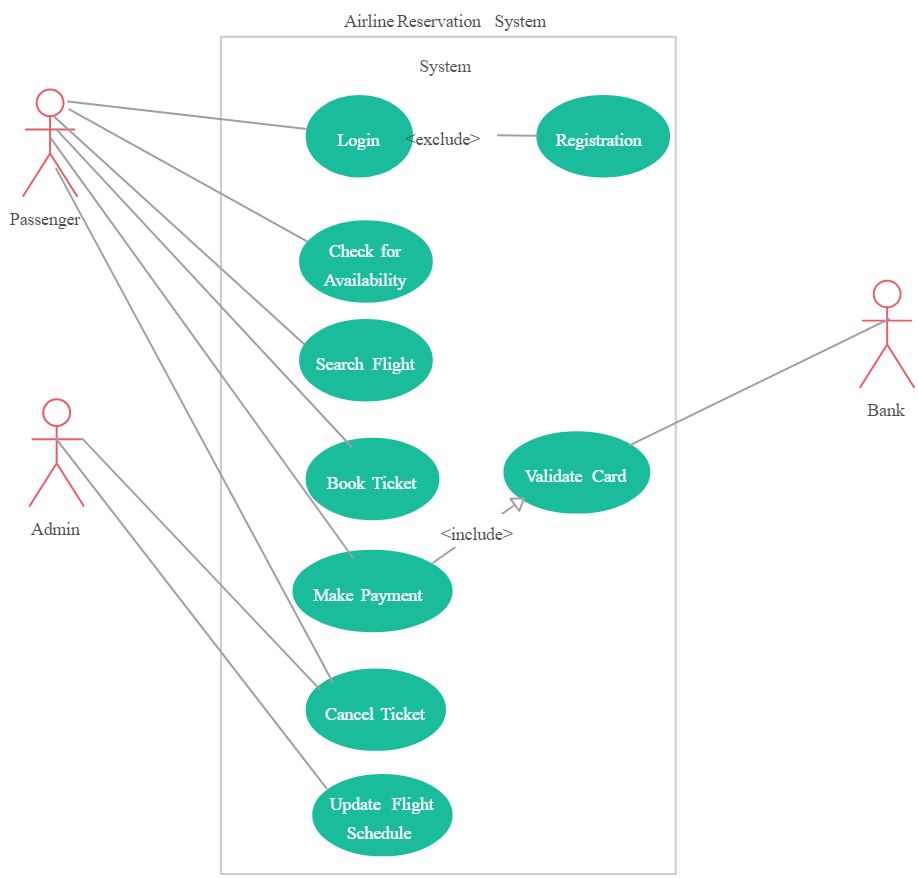
**Object Diagram**

****

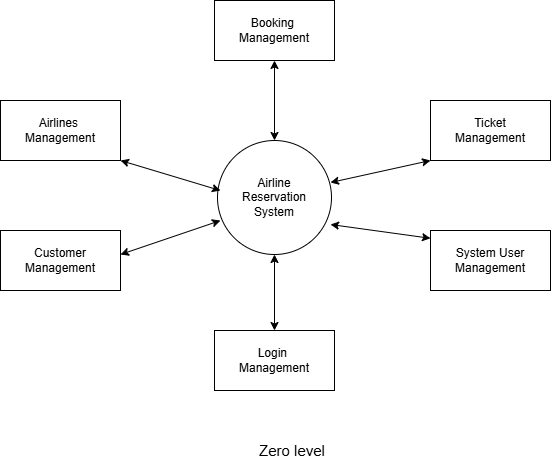
**Class Diagram**



**Use Case Diagrams**

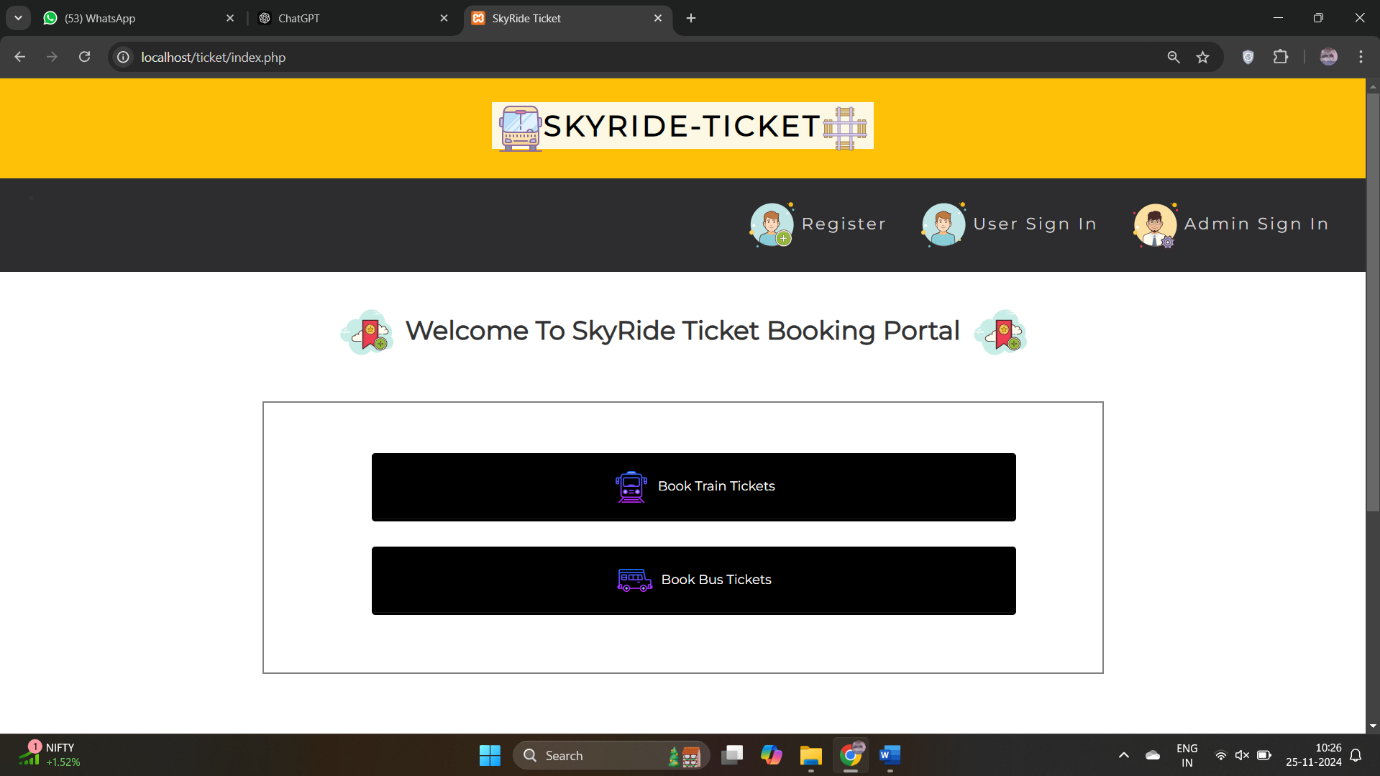


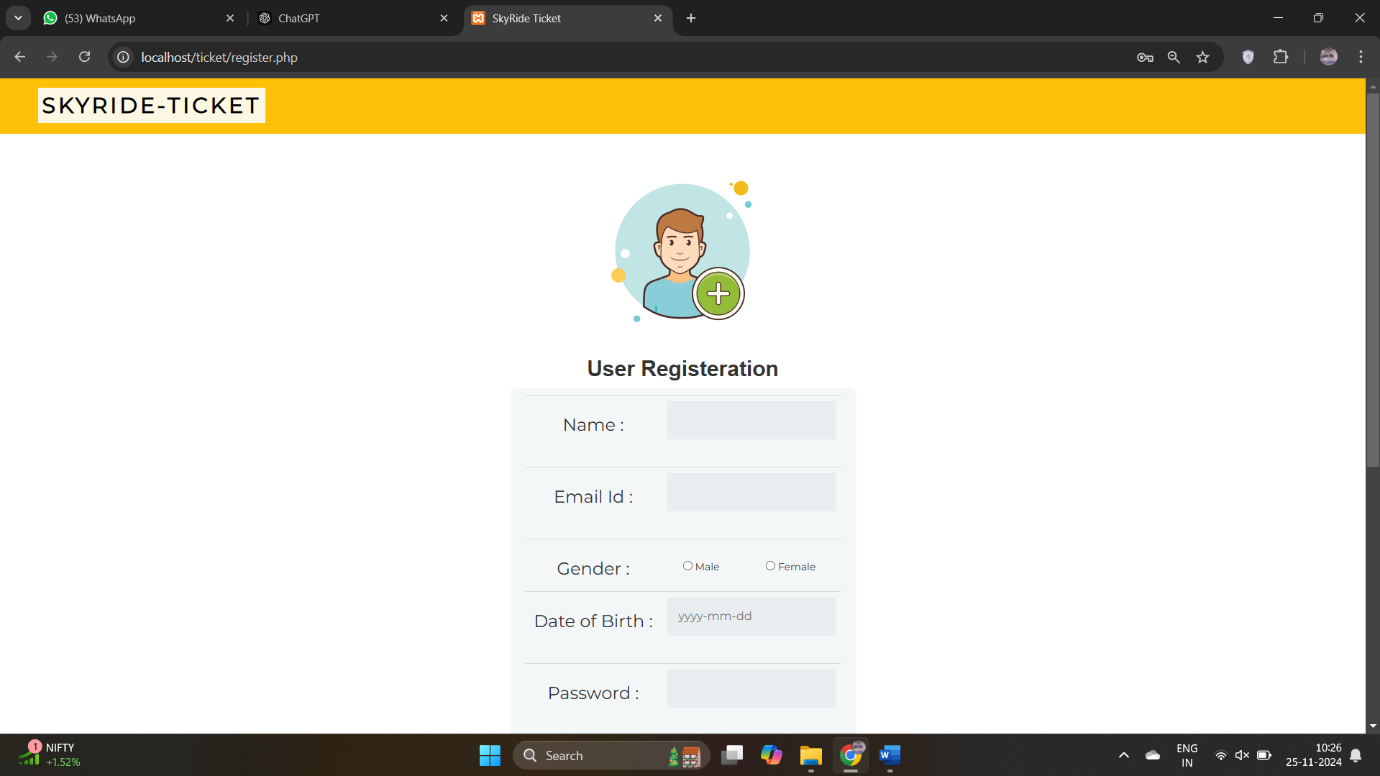
**Web Site Map Diagram (if Website )**

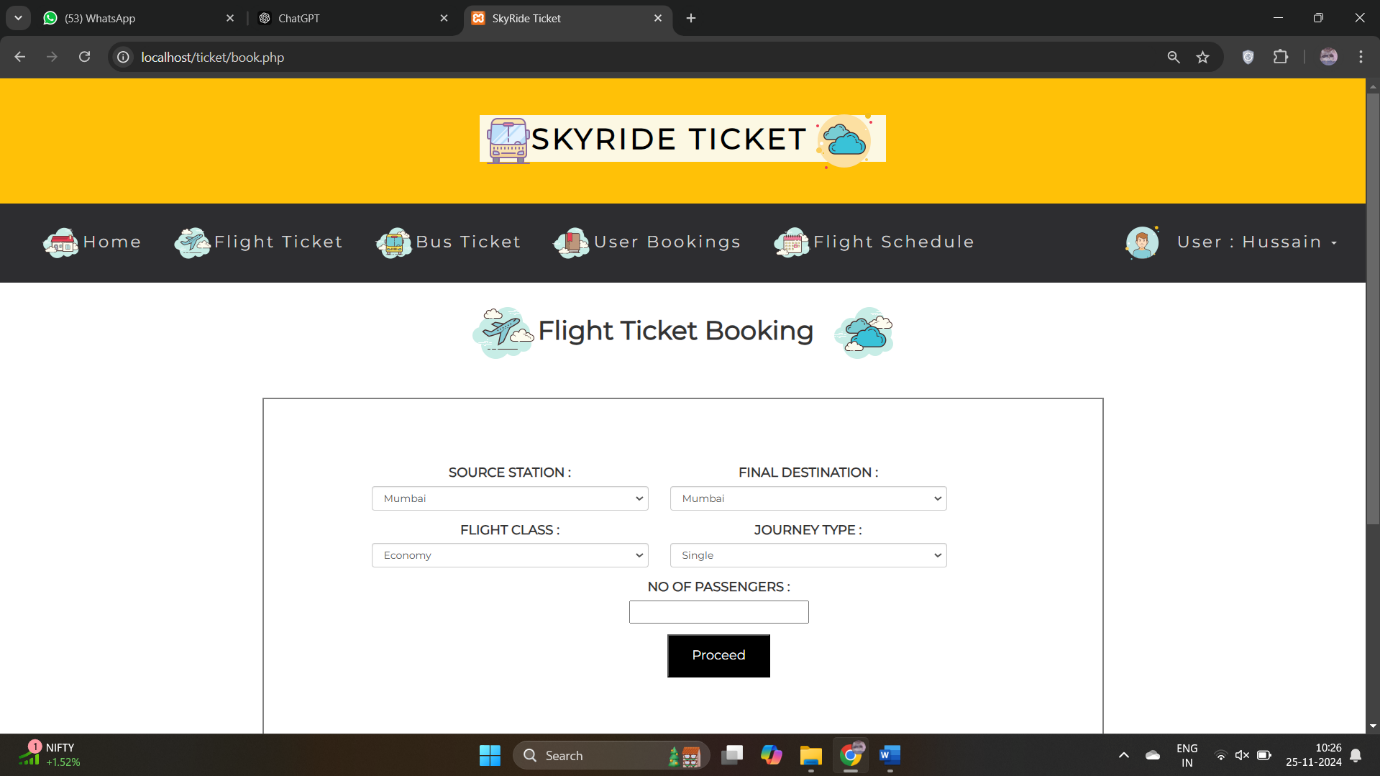


**User Interface Design (Screens etc.)**

**Index Page:**

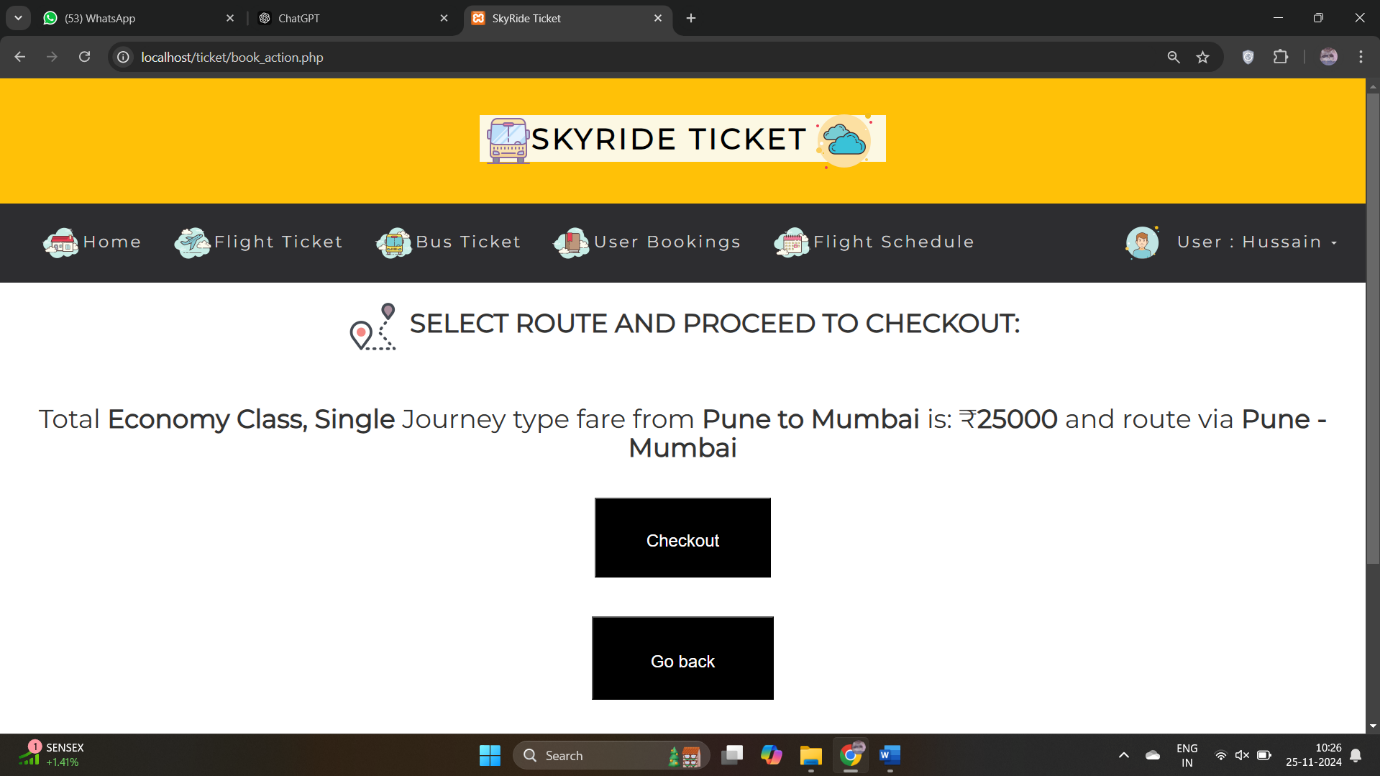
****

**Registration:**

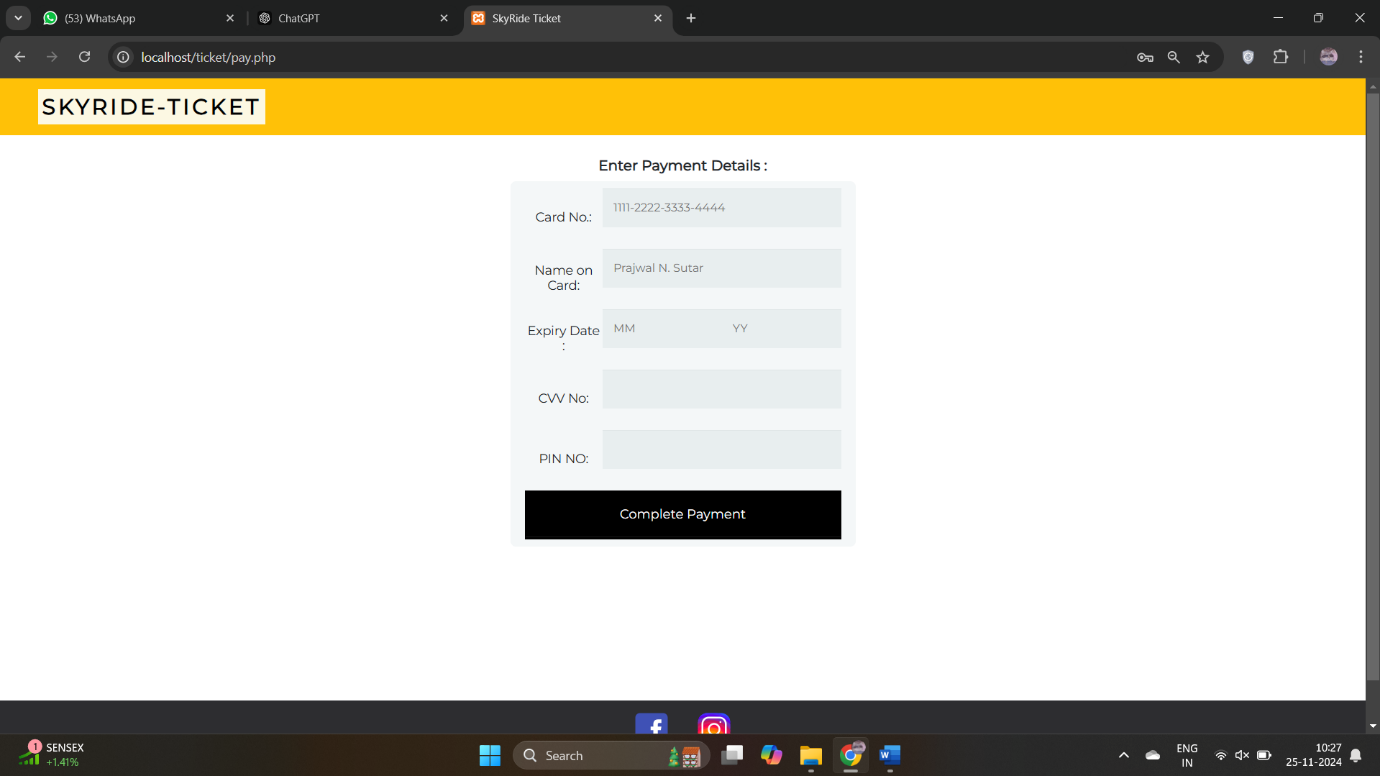


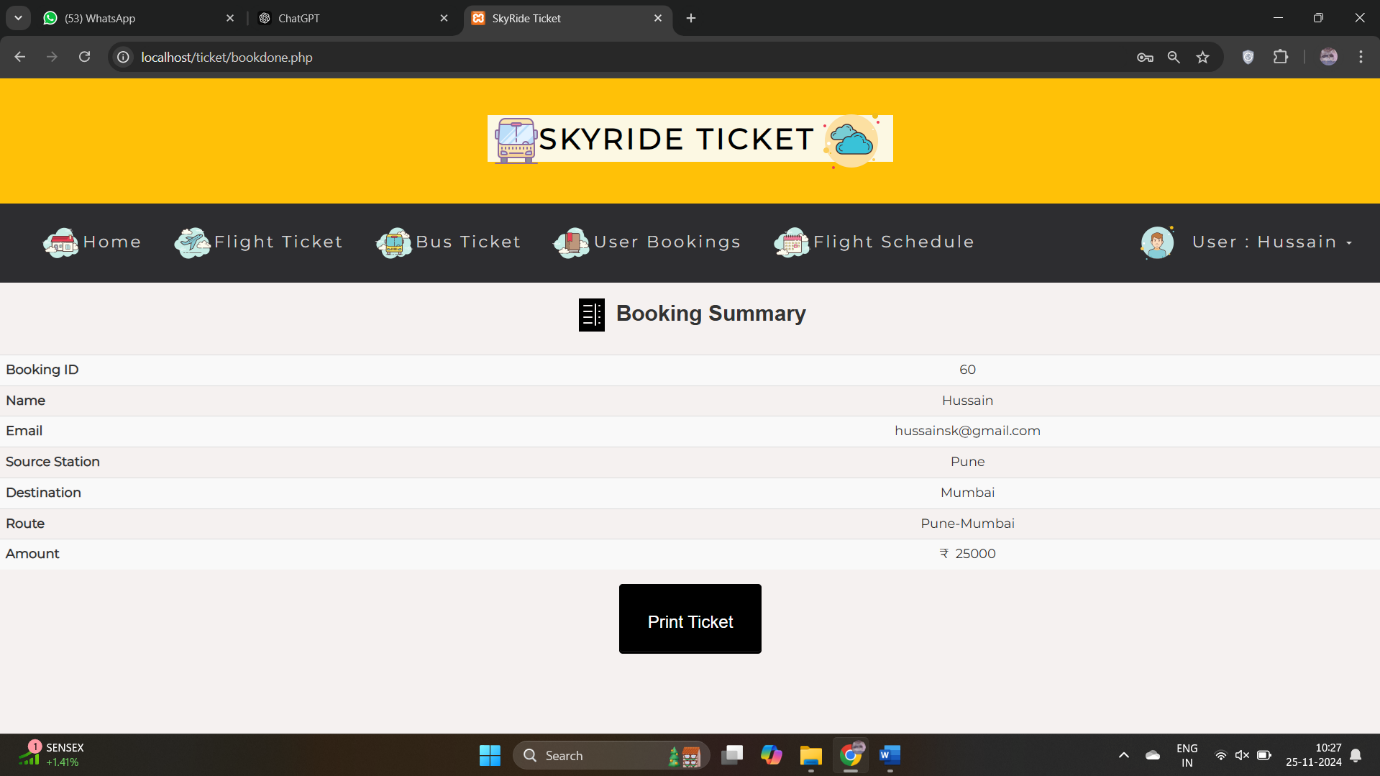
**Booking Site:**

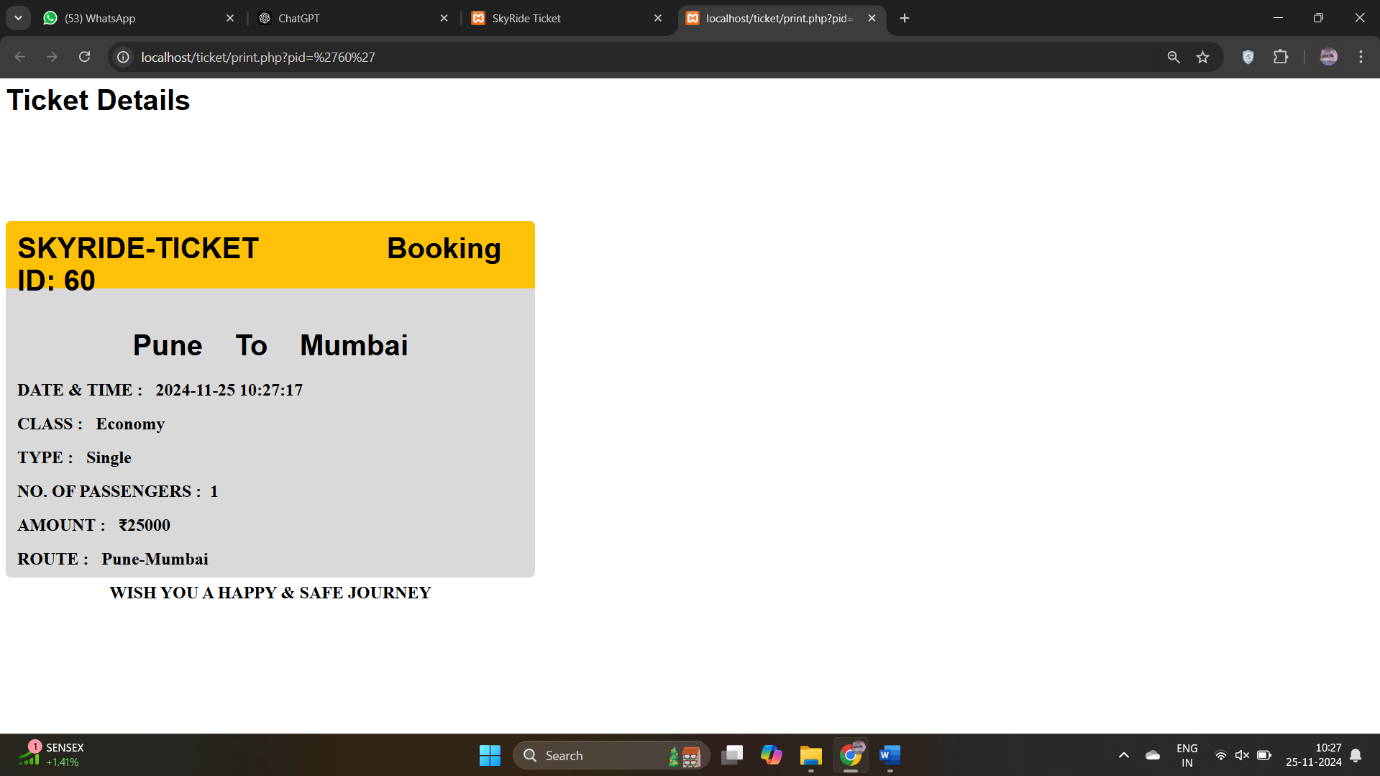
**Checkout:**

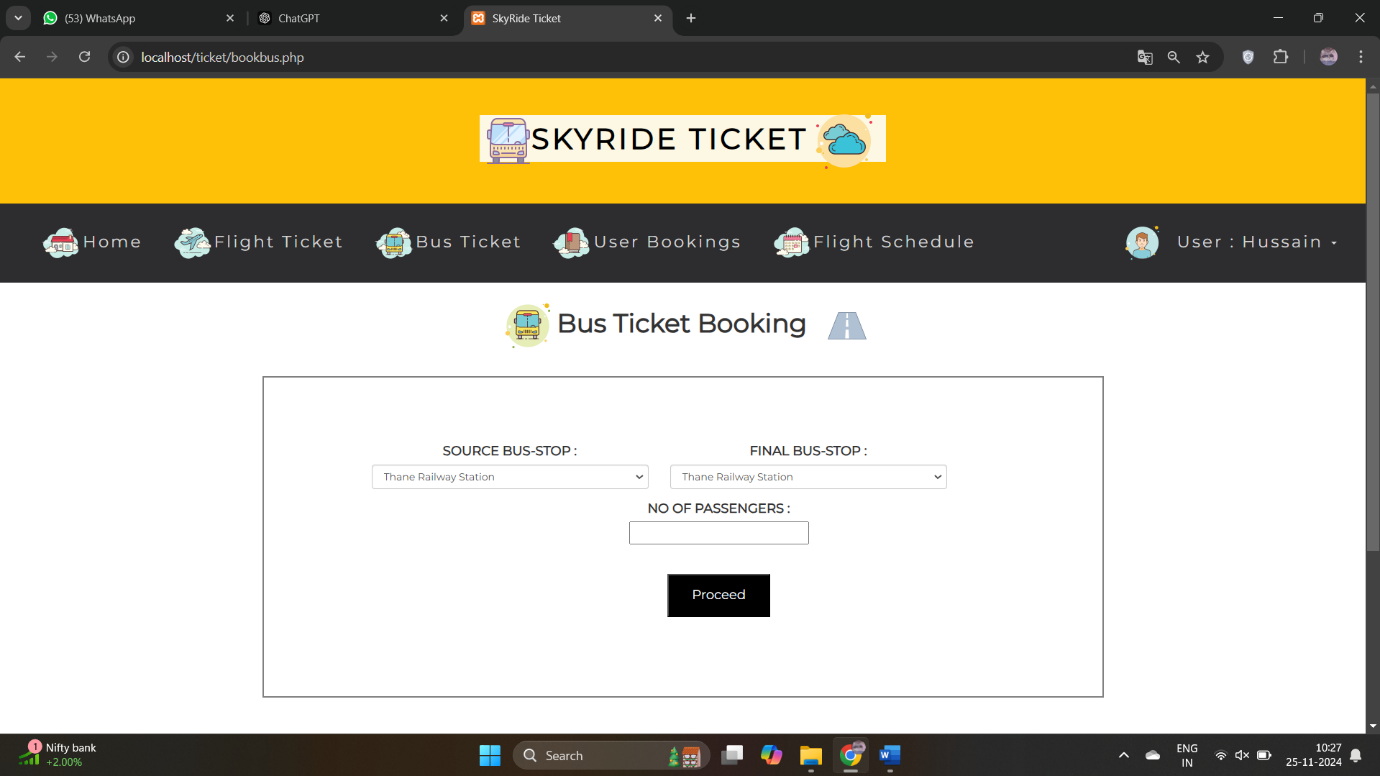


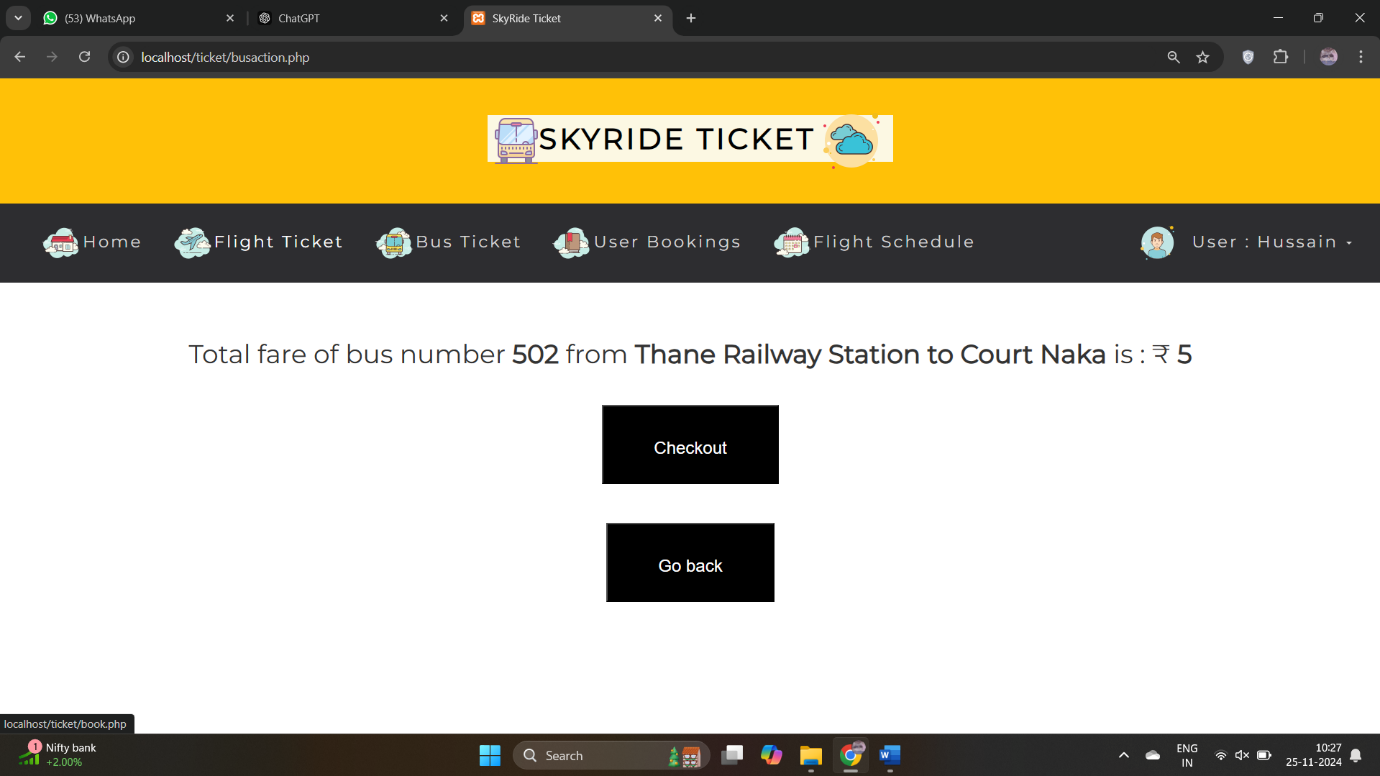
**Payment page:**

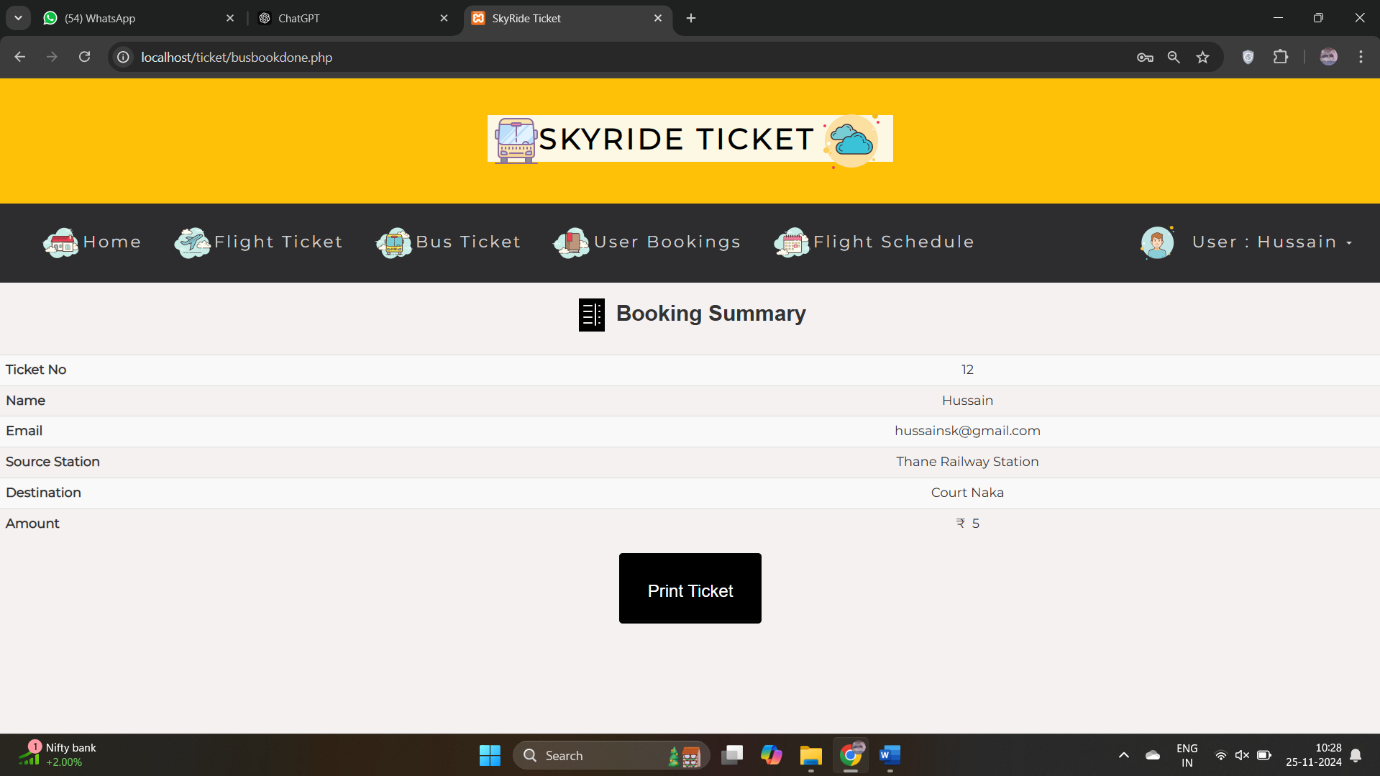


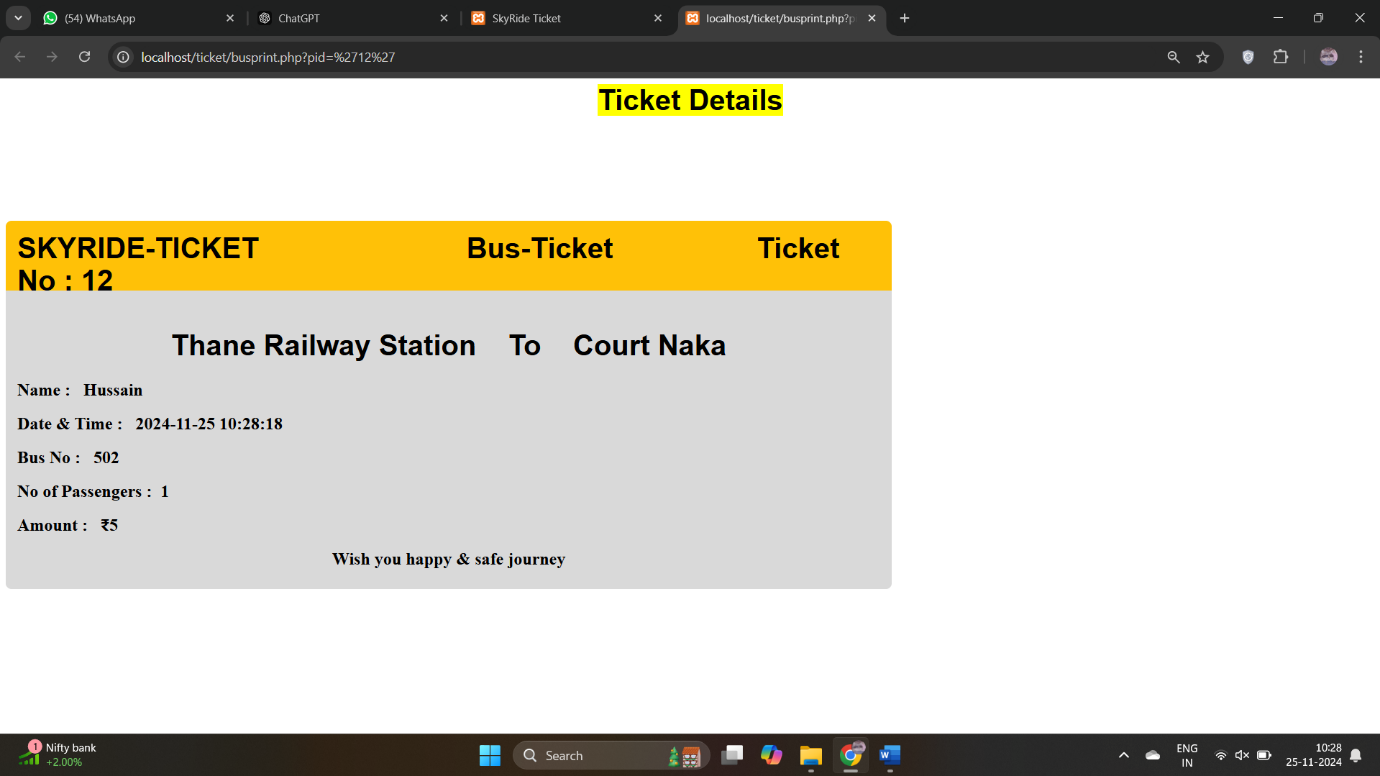
**Summary:**

**Ticket Details:**





**Bus Ticket Booking:**

**Bus Ticket:**

**Limitations**

**Scalability Challenges**

* The system may face difficulties in handling a high number of concurrent users, particularly during peak seasons or promotional events. This can lead to slow response times or system crashes without adequate infrastructure or optimization.

**Security Vulnerabilities**

* As the system deals with sensitive user data (e.g., payment details, personal information), it is prone to risks such as data breaches or hacking if robust security measures like encryption, firewalls, and regular audits are not implemented.

**Internet Dependency**

* Since the system is web-based, users and administrators must have a stable internet connection to access it. This limitation can be a significant drawback in areas with poor connectivity or during outages.

**Future enhancement**

1. **Mobile Application Development**
   * **Purpose:** Develop dedicated Android and iOS apps to provide a seamless booking experience on mobile devices.
   * **Benefits:** Improved accessibility, real-time notifications, and offline booking features.
2. **Integration with Global Distribution Systems (GDS)**
   * **Purpose:** Connect the system to GDS platforms like Amadeus or Sabre to offer broader flight options, including partner airlines.
   * **Benefits:** Enhanced flight availability, better pricing options, and access to international bookings.
3. **AI-Powered Recommendations**
   * **Purpose:** Implement AI to provide personalized suggestions for flights, destinations, and services based on user preferences and travel history.
   * **Benefits:** Increased customer satisfaction and better user engagement through tailored experiences.

**BIBLIOGRAPHY**

**Books and Publications**

* Robert C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship* (Prentice Hall, 2008).
* Larry Ullman, *PHP and MySQL for Dynamic Web Sites* (Peachpit Press, 2017).

**Web Resources**

* Official PHP Documentation: <https://www.php.net/docs.php>
* W3Schools for HTML, CSS, and JavaScript Tutorials: <https://www.w3schools.com>
* MySQL Documentation: <https://dev.mysql.com/doc>
* Apache HTTP Server Documentation: <https://httpd.apache.org/docs>

**Tools and Frameworks**

* Visual Studio Code (Code Editor): <https://code.visualstudio.com>
* Bootstrap CSS Framework: <https://getbootstrap.com>
* XAMPP for Local Development: <https://www.apachefriends.org>

**ANNEXURE:**

Database Connection (PHP):

<?php

$host = 'localhost';

$user = 'root';

$password = '';

$database = 'airline\_system';

$conn = new mysqli($host, $user, $password, $database);

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

?>