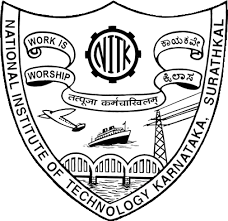
**National Institute of Technology Karnataka Surathkal,**

**Mangalore, India**

**Department of Computer Science and Engineering**



**Database Management System Project**

**Airline Reservation System**

**D. R. Vikas - 221CS215**

**Harsha Guandapalli - 221CS222**

**Manohar Rohit Vijay - 221CS230**

**Rayala Abhyuday - 221CS239**

**Regulagadda Rohan Chandra - 221CS241**

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| Sl.no | Topic | Page Number |
| 1 | Introduction | 3 |
| 2 | Literature Survey | 3 |
| 3 | Objectives | 4 |
| 4 | Features | 5 |
| 5 | Requirement Specification  Front End  Back End | 6 |
| 6 | System Design  Schema Diagram | 6 |
| 7 | Normalization Process | 7 |
| 8 | Tables | 8 |
| 9 | References | 11 |

**Introduction**

In the competitive aviation industry, effective airline management is crucial for operational efficiency, customer satisfaction, and profitability. Modern airlines utilize online booking systems to revolutionize the passenger experience, offering convenience and accessibility. These systems streamline reservations, minimize errors, and handle high transaction volumes efficiently. Online bookings also provide valuable data for tailoring services and marketing efforts. However, successful implementation requires careful planning and investment in user-friendly, secure technology infrastructure. Integration with other operational functions is essential for a seamless passenger experience. Airlines that embrace innovative technologies and prioritize customer-centric strategies can thrive by leveraging online booking systems to enhance satisfaction, drive revenue growth, and maintain competitiveness.

**Literature Survey**

The system enables users to register for an account, providing personal details such as name, address, contact information, gender, email, age, and creating a unique user ID and password. Registered users can directly book flights, while new users must register before booking. Users can check flight availability by inputting origin and destination cities along with the date of travel. Upon selecting a flight, the system prompts for confirmation before proceeding to reserve seats. For canceling or rescheduling tickets, users must provide ticket and flight details.

The system facilitates automated data entry as passengers input their details directly, minimizing manual entry errors. Real-time validation checks ensure accuracy and completeness of passenger information while automated processes prevent double bookings by updating seat availability instantly. Security measures include SSL encryption for data transmission and integration with secure payment gateways. Optional two-factor authentication adds an extra layer of security. Passengers can filter flights and compare fares across different airlines, aided by detailed flight listings and route maps. Flexible booking options cater to diverse travel needs, with features like seat selection and instant seat availability status. Automatic waitlist management notifies passengers of available seats, reducing manual work and streamlining the booking process.

The survey begins by outlining the fundamental objectives of ARSs, which include providing seamless ticket reservation, cancellation, and rescheduling functionalities, all while centralizing information to enable easy updates and minimize manual work. However, existing systems face notable challenges, such as the lack of real-time flight tracking, discrepancies in ticket pricing across different platforms, limitations in seat selection for connecting flights, and issues related to luggage tracking. To address these challenges, proposed solutions include integrated real-time flight tracking, centralized pricing information, advanced seat selection for connecting flights, and enhanced luggage tracking features. Additionally, future enhancements could encompass ticket editing capabilities, Aadhaar integration for ID verification, and health/Covid status verification for passengers, all aimed at further improving system functionality and user experience. The survey concludes by highlighting the significance of ongoing enhancements and technological advancements in meeting the evolving needs of customers and the airline industry, ultimately underlining the critical role of Java-based ARSs in modern air travel management.

**Objectives**

1. User Convenience: Provide a user-friendly interface for easy flight searches based on preferences like dates, destinations, and budget considerations.

1. Comprehensive Search: Offer a comprehensive search functionality enabling users to compare flights from different airlines based on factors like price, duration, and layovers.

1. Competitive Pricing: Ensure competitive pricing options, enabling users to find affordable and value-for-money flight options.

1. Canceling or Rescheduling: Provide a straightforward process for canceling flights, allowing users to easily navigate and manage their bookings.

1. Booking and Payment: Facilitate a seamless booking process with secure online payment options for quick and efficient flight reservations.

1. Real-time Updates: Deliver real-time updates on flight availability, schedule changes, and relevant information for users' travel plans.

1. Customer Support: Offer effective customer support services to assist users with any issues, changes, or queries related to their flight bookings.

1. Integration with Other Services: Integrate the flight booking system with other travel-related services like hotel reservations, bus bookings, and travel packages, providing users with a one-stop solution.

1. Catering Services: If applicable, allow users to customize in-flight meals by selecting specific dishes, dietary restrictions, or special requests during the booking process.

**Features**

**User Management**

1. User Profiles: Store personal information like name, contact details, and preferences.
2. Authentication and Authorization: Manage login, registration, and permissions.
3. User Preferences: Save preferred airlines, meal choices, seating preferences, etc.

**Flight Information**

1. Flights: Details include flight number, airline, departure and arrival times, duration, and available classes (economy, business, first class).
2. Airports: Information on airport names, locations (city, country), and facilities.

**Booking Management**

1. Flight Bookings: Records of user bookings, including flight details, class, seat selection, and fare.
2. Payment Information: Securely store payment transactions, including payment method, transaction ID, and status.
3. Booking Status: Track booking status (confirmed, canceled, pending).
4. Meals and Amenities: Details on available meal options and additional services (Wi-Fi, entertainment).

**Search and Filters**

1. Flight Search: Functionality to search for flights based on parameters like date, destination, and number of passengers.
2. Filters and Sorting: Options to filter and sort search results by price, duration, airline, departure time, etc.

**Required Specifications**

**Front end:**

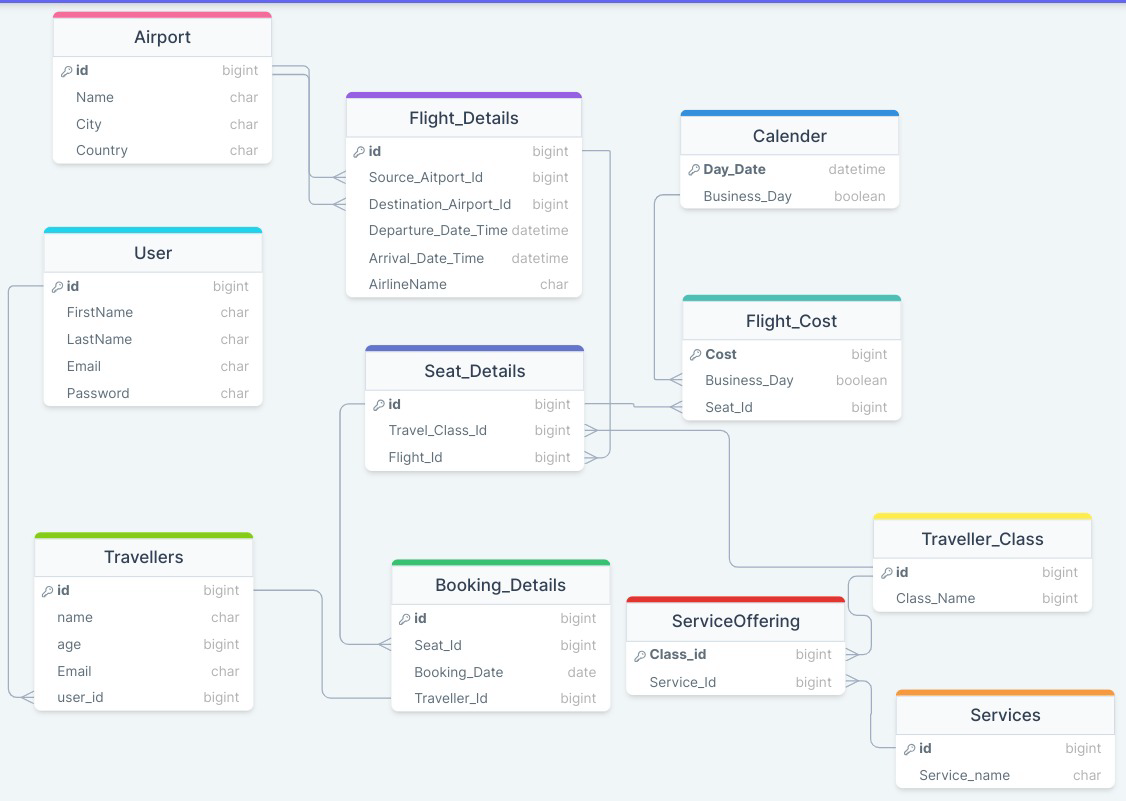
integrating React JS into the frontend framework enhances the UI/UX of the airline reservation system by enabling the creation of dynamic, responsive, and interactive user interfaces, ultimately improving the booking experience for users.

**Back end:**

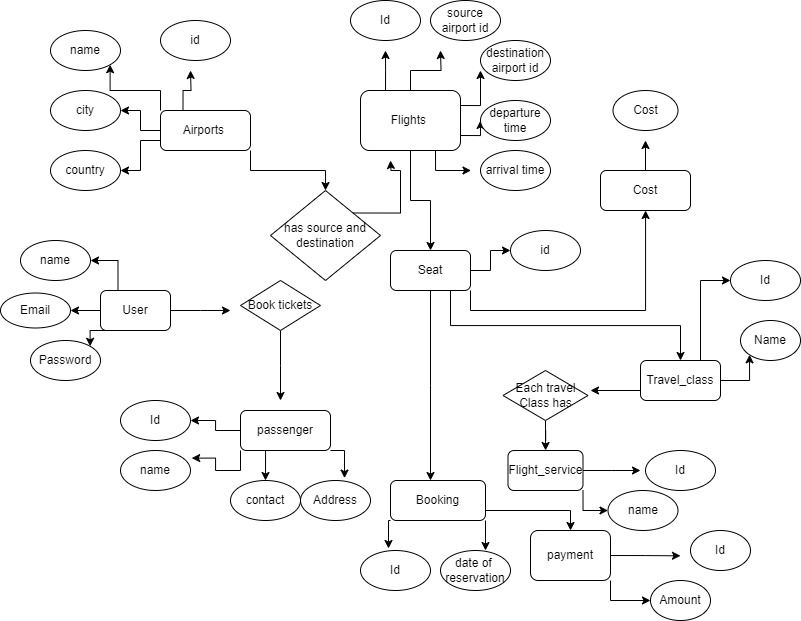
We will be using Node JS runtime to run our backend express server which enhances the communication between the frontend queries and database of our application. As informed, we will be using MySQL as our database.

**System Design**

**Database Schema Design:**



**ER diagram:**



**Normalization Process**

A relation is said to be in 1 normal form in DBMS (or 1NF) when it consists of an atomic value. In simpler words, 1NF states that a table's attribute would not be able to hold various values- it will only be able to hold an attribute of a single value. So, we normalized our tables in such a way that every attribute is single valued. For example, each traveler class has multiple services, so we created a separate table called service\_offering which maps each travel\_class with multiple services listed in the services table.

A relation is said to be in the 2nd Normal Form in DBMS (or 2NF) when it is in the First Normal Form but has no non-prime attribute functionally dependent on any candidate key's proper subset in a relation. In the same way, we have tables where each non-prime attribute is dependent only on primary key, but not other non-prime attributes.For example, In Airport Table, every attribute like city, airport name are all dependent only on Airport\_Id.

**Tables:**

CREATE TABLE `Traveller\_Class`(

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`Class\_Name` BIGINT NOT NULL

);

CREATE TABLE `Flight\_Cost`(

`Cost` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`Business\_Day` TINYINT(1) NOT NULL,

`Seat\_Id` BIGINT NOT NULL

);

CREATE TABLE `Calender`(

`Day\_Date` DATETIME NOT NULL,

`Business\_Day` TINYINT(1) NOT NULL,

PRIMARY KEY(`Day\_Date`)

);

CREATE TABLE `ServiceOffering`(

`Class\_id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`Service\_Id` BIGINT NOT NULL

);

CREATE TABLE `Services`(

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`Service\_name` CHAR(255) NOT NULL

);

CREATE TABLE `Airport`(

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`Name` CHAR(255) NOT NULL,

`City` CHAR(255) NOT NULL,

`Country` CHAR(255) NOT NULL

);

CREATE TABLE `Flight\_Details`(

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`Source\_Aitport\_Id` BIGINT NOT NULL,

`Destination\_Airport\_Id` BIGINT NOT NULL,

`Departure\_Date\_Time` DATETIME NOT NULL,

`Arrival\_Date\_Time` DATETIME NOT NULL,

`AirlineName` CHAR(255) NOT NULL

);

CREATE TABLE `Seat\_Details`(

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`Travel\_Class\_Id` BIGINT NOT NULL,

`Flight\_Id` BIGINT NOT NULL

);

CREATE TABLE `Booking\_Details`(

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`Seat\_Id` BIGINT NOT NULL,

`Booking\_Date` DATE NOT NULL,

`Traveller\_Id` BIGINT NOT NULL

);

CREATE TABLE `Travellers`(

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`name` CHAR(255) NOT NULL,

`age` BIGINT NOT NULL,

`Email` CHAR(255) NOT NULL,

`user\_id` BIGINT NOT NULL

);

CREATE TABLE `User`(

`id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`FirstName` CHAR(255) NOT NULL,

`LastName` CHAR(255) NOT NULL,

`Email` CHAR(255) NOT NULL,

`Password` CHAR(255) NOT NULL

);

ALTER TABLE

`Seat\_Details` ADD CONSTRAINT `seat\_details\_flight\_id\_foreign` FOREIGN KEY(`Flight\_Id`) REFERENCES `Flight\_Details`(`id`);

ALTER TABLE

`Flight\_Details` ADD CONSTRAINT `flight\_details\_destination\_airport\_id\_foreign` FOREIGN KEY(`Destination\_Airport\_Id`) REFERENCES `Airport`(`id`);

ALTER TABLE

`Booking\_Details` ADD CONSTRAINT `booking\_details\_seat\_id\_foreign` FOREIGN KEY(`Seat\_Id`) REFERENCES `Seat\_Details`(`id`);

ALTER TABLE

`Travellers` ADD CONSTRAINT `travellers\_user\_id\_foreign` FOREIGN KEY(`user\_id`) REFERENCES `User`(`id`);

ALTER TABLE

`Booking\_Details` ADD CONSTRAINT `booking\_details\_traveller\_id\_foreign` FOREIGN KEY(`Traveller\_Id`) REFERENCES `Travellers`(`id`);

ALTER TABLE

`Flight\_Cost` ADD CONSTRAINT `flight\_cost\_seat\_id\_foreign` FOREIGN KEY(`Seat\_Id`) REFERENCES `Seat\_Details`(`id`);

ALTER TABLE

`Seat\_Details` ADD CONSTRAINT `seat\_details\_travel\_class\_id\_foreign` FOREIGN KEY(`Travel\_Class\_Id`) REFERENCES `Traveller\_Class`(`id`);

ALTER TABLE

`ServiceOffering` ADD CONSTRAINT `serviceoffering\_service\_id\_foreign` FOREIGN KEY(`Service\_Id`) REFERENCES `Services`(`id`);

ALTER TABLE

`Flight\_Details` ADD CONSTRAINT `flight\_details\_source\_aitport\_id\_foreign` FOREIGN KEY(`Source\_Aitport\_Id`) REFERENCES `Airport`(`id`);

ALTER TABLE

`ServiceOffering` ADD CONSTRAINT `serviceoffering\_class\_id\_foreign` FOREIGN KEY(`Class\_id`) REFERENCES `Traveller\_Class`(`id`);

ALTER TABLE

`Calender` ADD CONSTRAINT `calender\_business\_day\_foreign` FOREIGN KEY(`Business\_Day`) REFERENCES `Flight\_Cost`(`Business\_Day`);

**Implementation:**

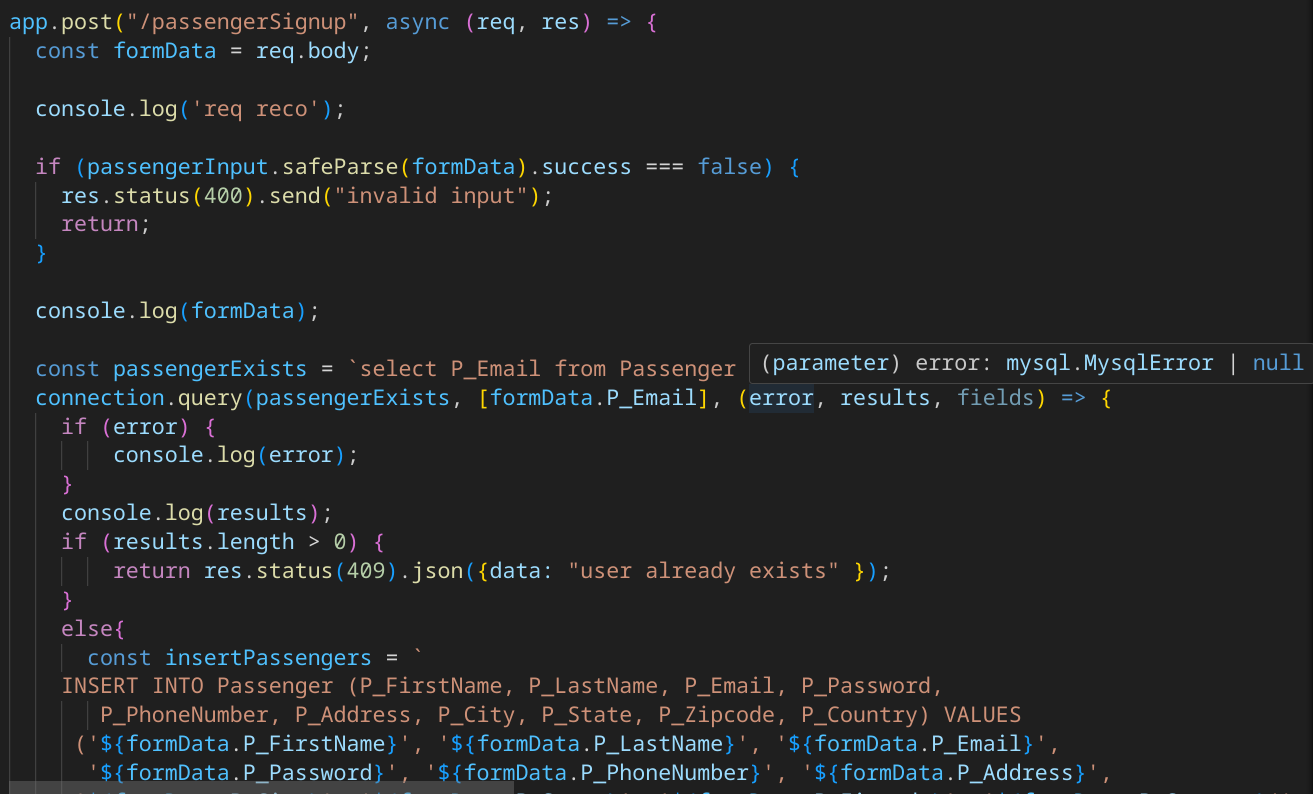
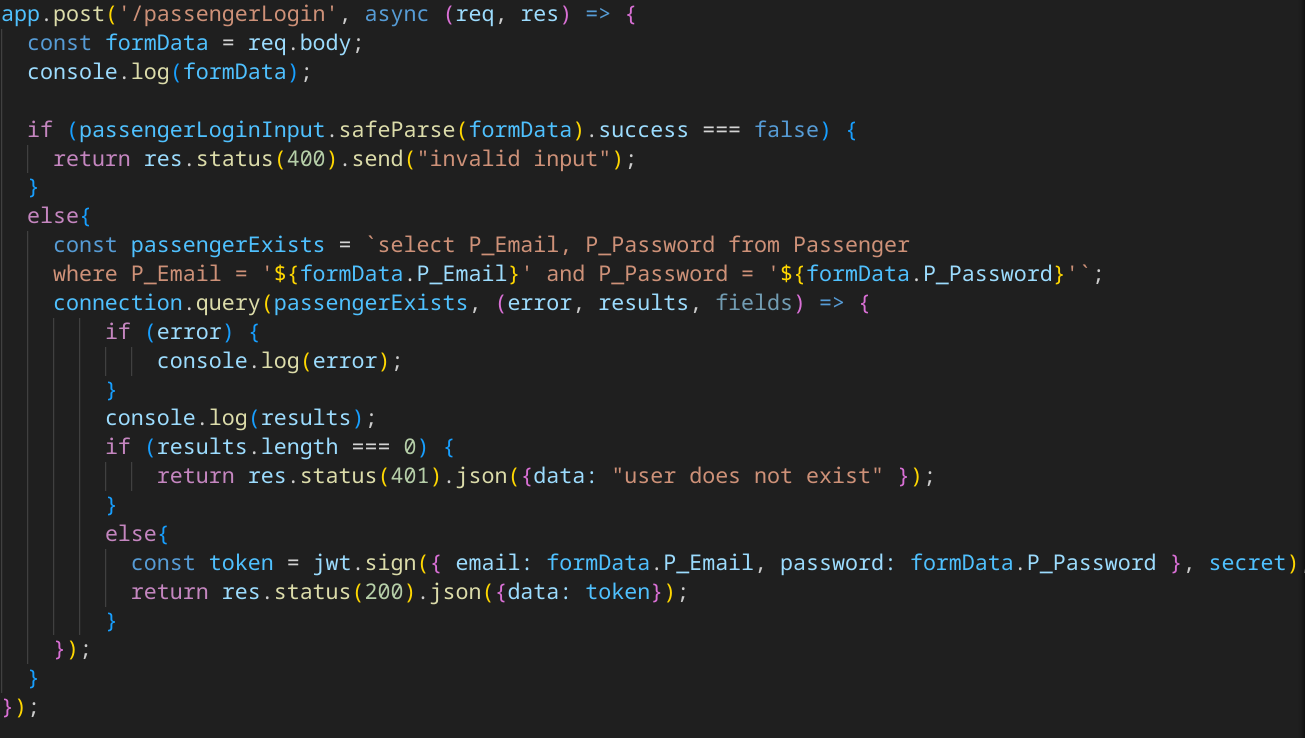
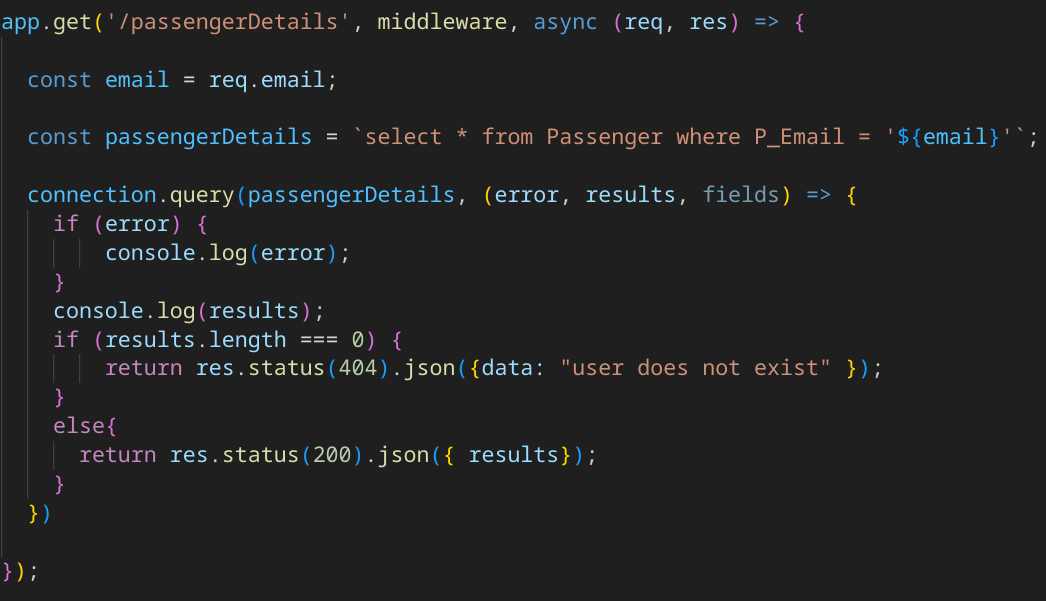
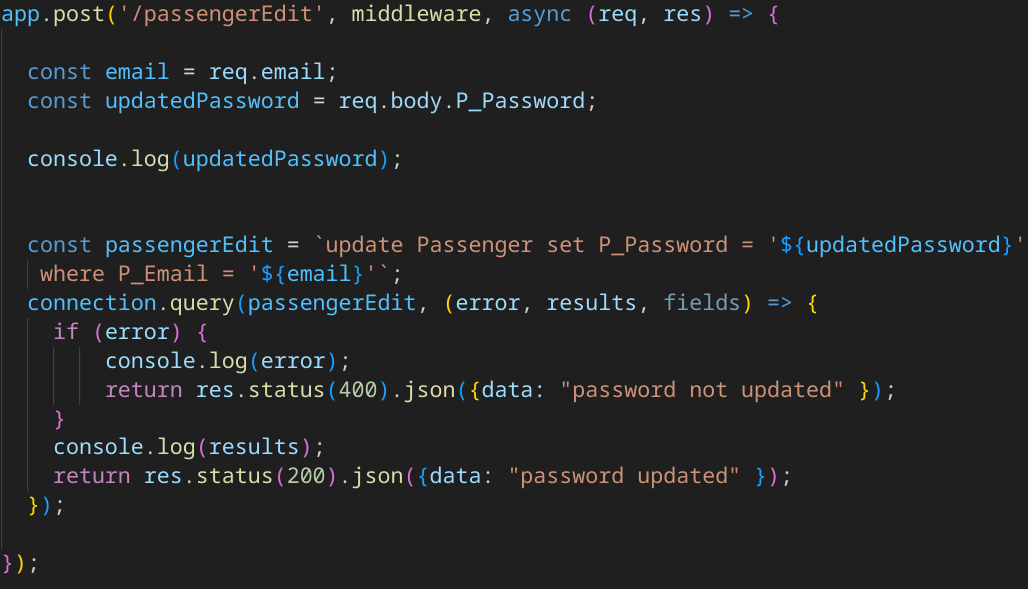
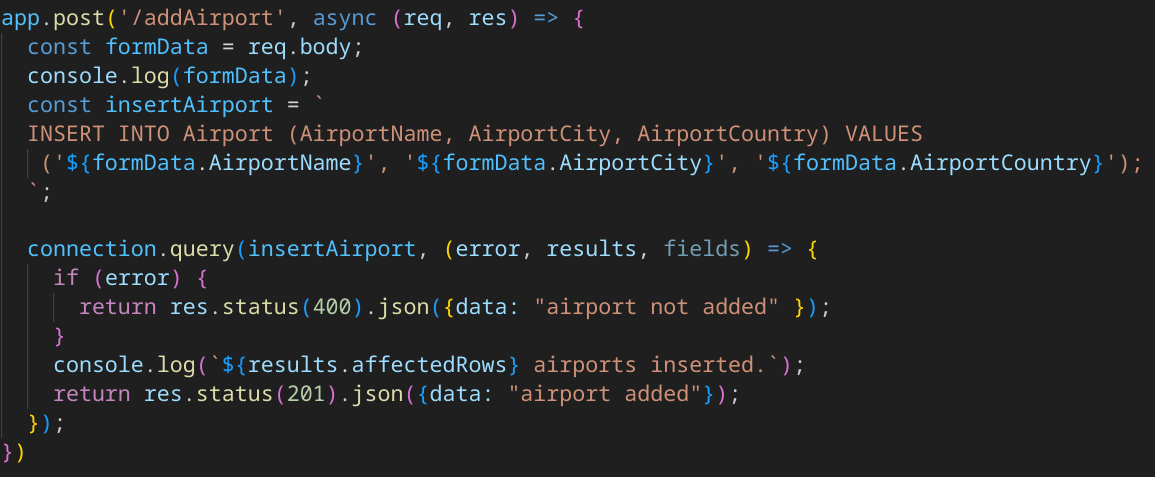
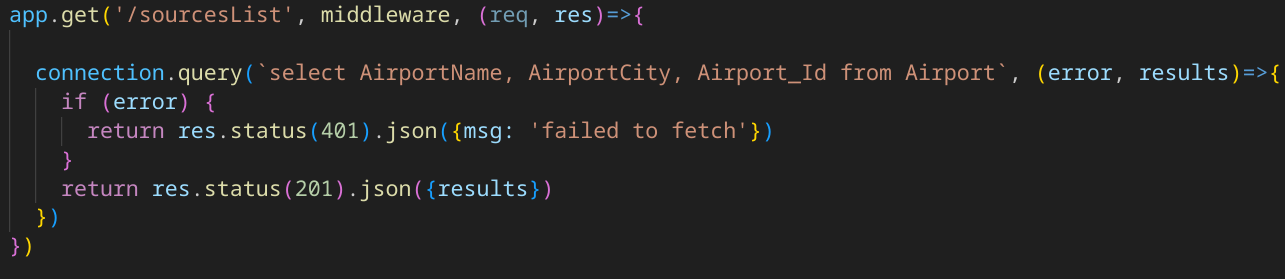
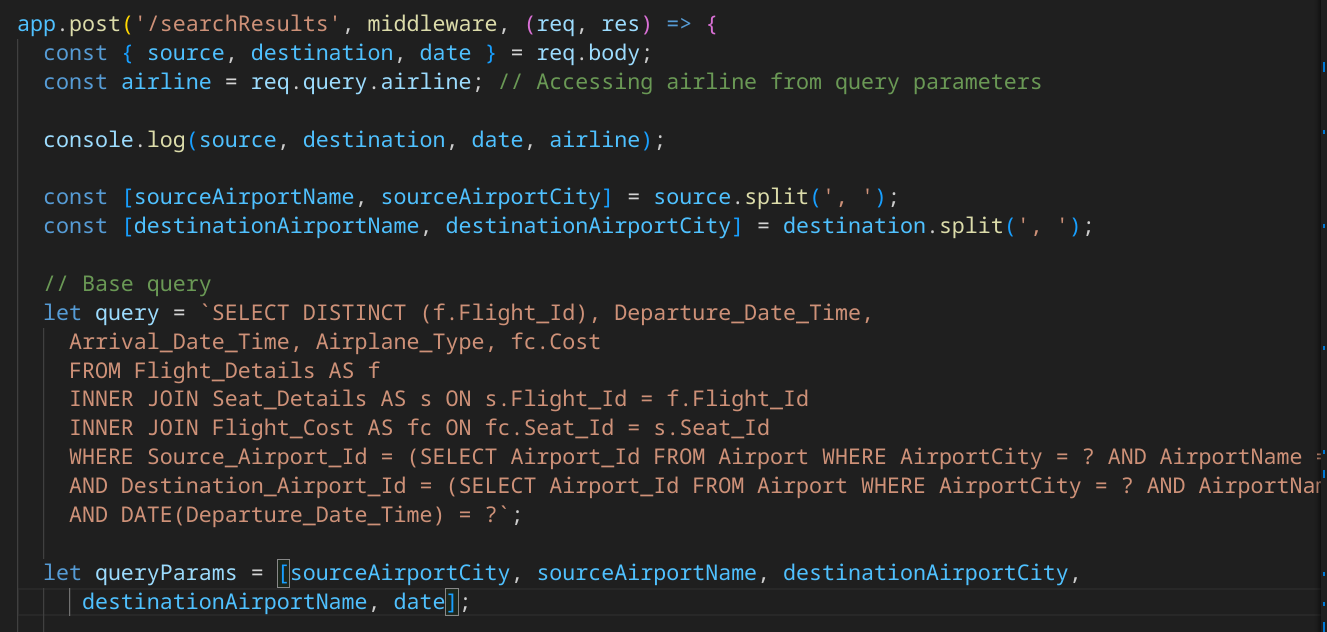
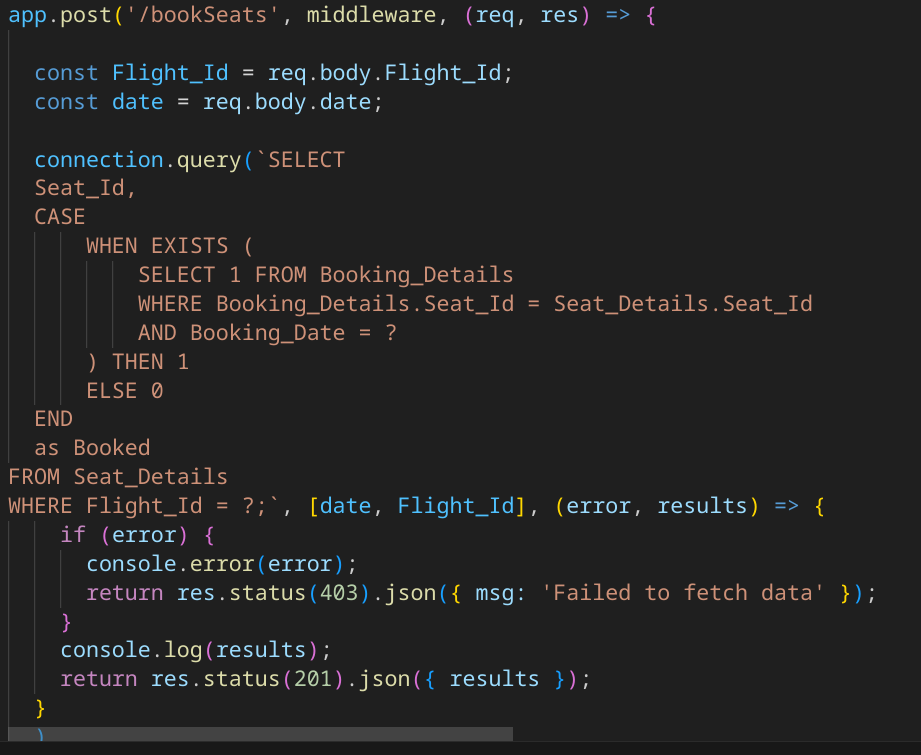
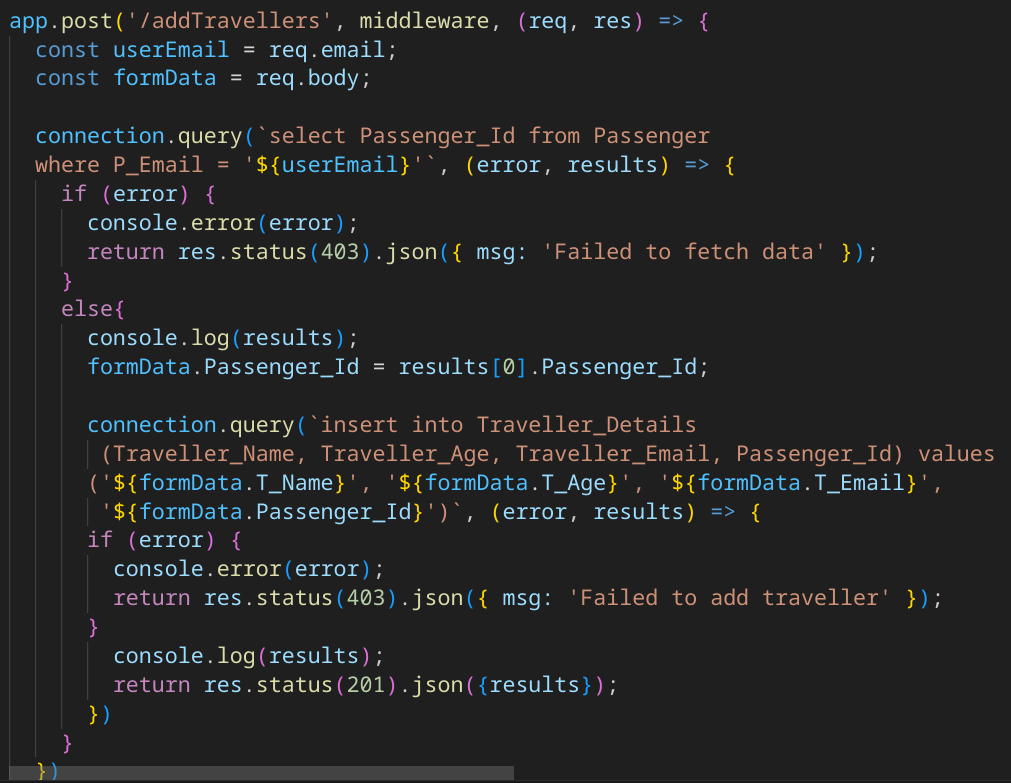
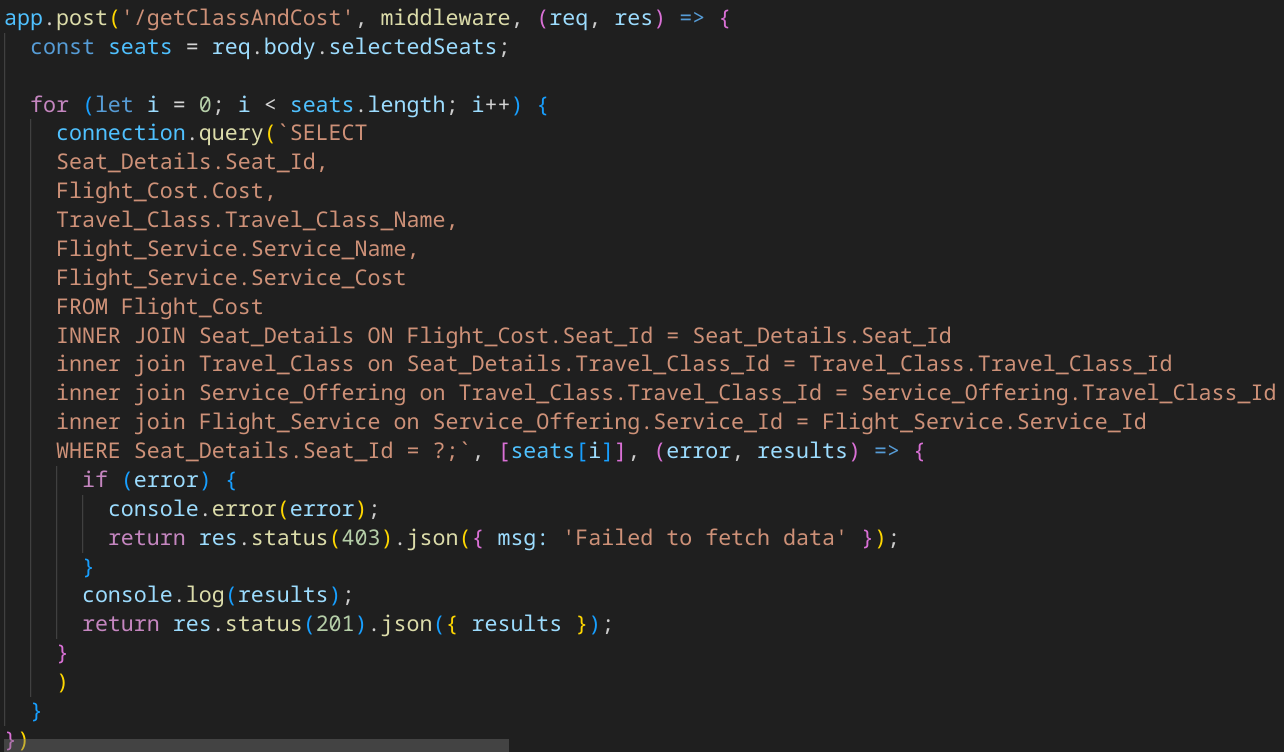
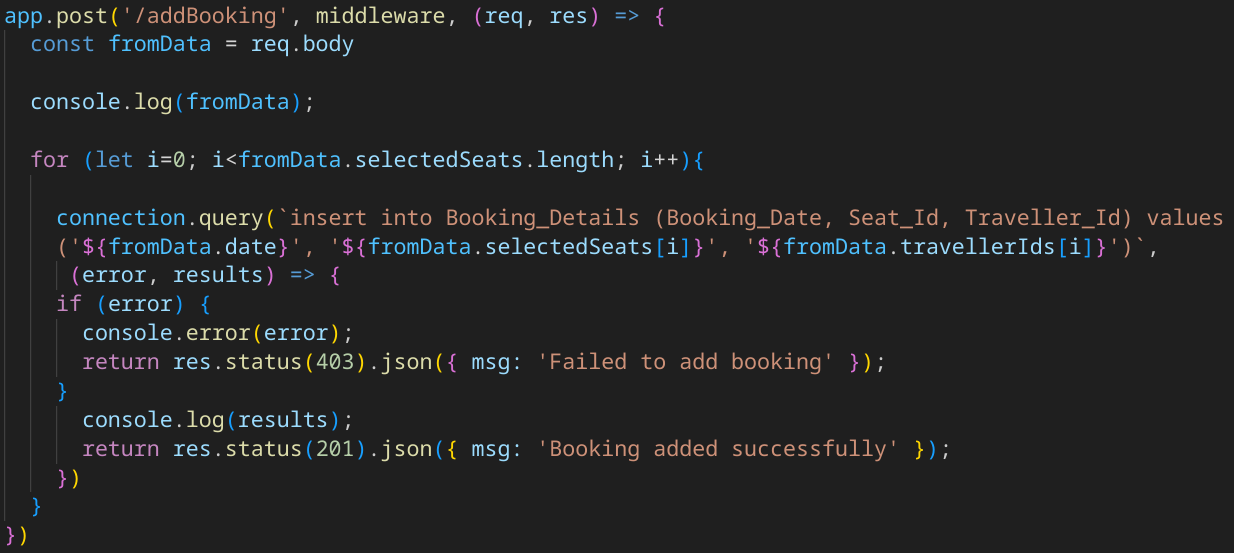
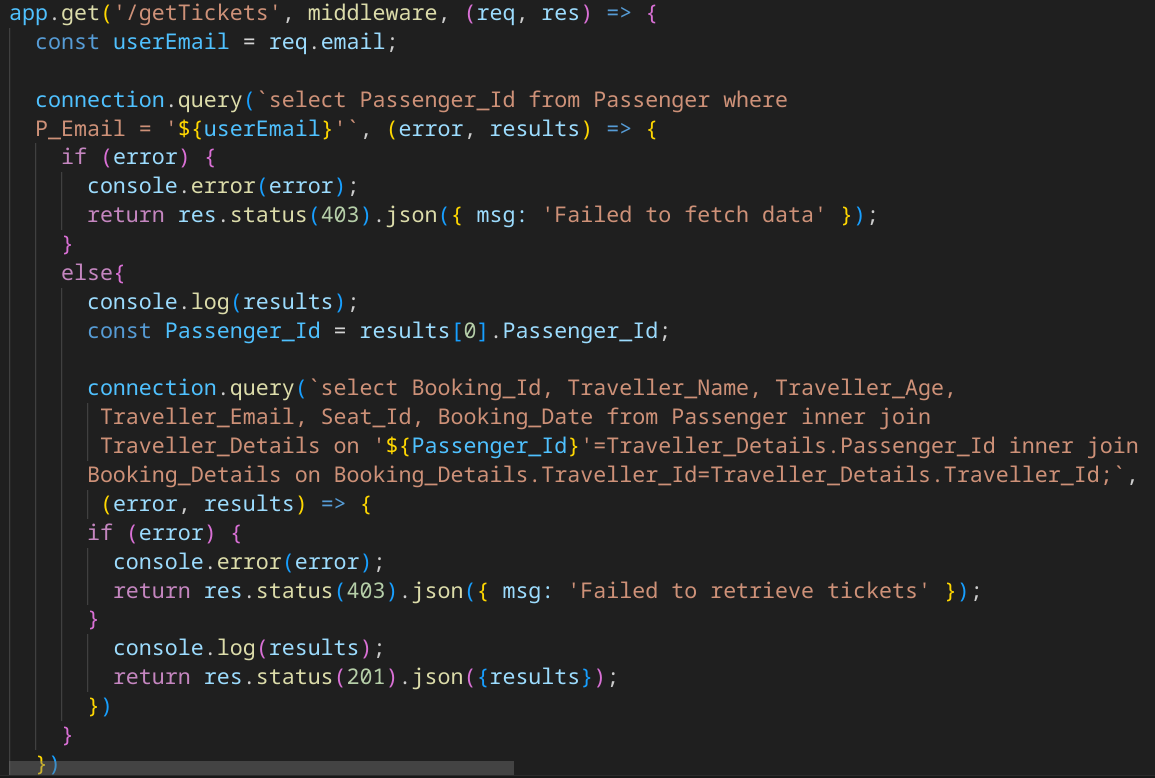
**● Description of the tools and technologies used.**

This application utilizes a React.js frontend for building the user interface. React excels at creating modular and dynamic UI components that efficiently update based on user interactions and data changes.

On the backend, Node.js serves as the runtime environment, executing the application's core logic. Its event-driven, asynchronous nature makes it ideal for real-time applications. Express.js sits on top of Node.js, acting as a web framework that simplifies server creation and offers functionalities for managing routes and middleware.

Data persistence is handled by a MySQL database hosted on Clever Cloud. The application interacts with this data through SQL queries specifically designed for MySQL, enabling functionalities like data retrieval, manipulation, and storage. This combination provides a robust and scalable architecture for the web application.

**● Code snippets**



**Conclusion:**

**● Summary of the project:**

The airline ticket booking mini project undertaken in the domain of database management aimed to develop a comprehensive system for managing ticket reservations, flight schedules, passenger information, and related transactions. Throughout the project, various database management concepts and techniques were applied to design and implement a robust database schema and accompanying functionalities.

The project involved the creation of an intricate database schema encompassing entities such as flights, passengers, bookings, payment details, airports, and aircraft. Implementation included defining relationships, establishing constraints, and ensuring data normalization to minimize redundancy and maintain data integrity.

Additionally, the project entailed the development of a user-friendly interface to interact with the database system. This interface facilitated functionalities such as flight search, ticket booking, reservation management, payment processing, and itinerary generation. The interface was designed to cater to the needs of both customers and airline staff, offering a seamless experience and enhancing operational efficiency.

**Achievement of Project Objectives:**

The project successfully met its primary objectives, which included:

Database Design: A well-structured and normalized database schema was developed to efficiently organize and manage airline-related data. The schema minimized redundancy, ensured data consistency, and facilitated ease of maintenance.

Functionality Implementation: All essential functionalities expected in an airline ticket booking system were effectively implemented. These functionalities included flight search, seat reservation, ticket booking, payment processing, cancellation handling, and itinerary generation.

User Interface Development: A user-friendly interface was designed and implemented to interact with the database system. The interface provided intuitive navigation, clear presentation of information, and seamless transaction processing, enhancing user satisfaction and experience.

Data Security and Integrity: Measures were taken to ensure the security and integrity of the database system. Techniques such as access control, encryption, and data validation were employed to safeguard sensitive information and prevent unauthorized access or data manipulation.

**Lessons Learned:**

Several valuable lessons were gleaned from the project experience:

Requirement Analysis: Thorough requirement analysis is crucial for understanding the needs of stakeholders and defining the scope of the project. Clear communication and collaboration with stakeholders help in identifying and prioritizing essential features and functionalities.

Database Normalization: Proper normalization of the database schema is essential for reducing redundancy, minimizing update anomalies, and maintaining data integrity. Adhering to normalization principles simplifies data management and ensures scalability.

User Experience Design: Attention to user experience design is paramount for creating an intuitive and user-friendly interface. Iterative prototyping and user testing aid in refining the interface to meet the needs and expectations of users effectively.

Data Security Measures: Implementing robust data security measures is critical for protecting sensitive information and maintaining compliance with privacy regulations. Regular security audits and updates help in mitigating potential vulnerabilities and threats.

In conclusion, the airline ticket booking mini project provided valuable insights into database management principles, software development methodologies, and user-centered design practices. The successful execution of the project demonstrated the importance of meticulous planning, effective collaboration, and continuous improvement in delivering a functional and reliable database system tailored to the requirements of the airline industry.

**References:**

1. <https://www.studocu.com/in/document/galgotias-university/major-project/project-report/38079179>
2. <https://www.geeksforgeeks.org/how-to-design-database-for-flight-reservation-system/>
3. <https://www.academia.edu/36303556/Database_Management_System_Project_on_Airlines_Reservation_Database_Design>
4. <https://www.americanairlines.in/intl/in/index.jsp>
5. <https://www.singaporeair.com/en_UK/in/home#/book/bookflight>
6. <https://www.javatpoint.com/er-diagram-for-the-airline-reservation-system>