## Railway Ticket Management System

A

RDBMS Lab Project Report

Submitted in the partial fulfillment of the requirement for the award of Bachelor of Technology

in Computer and Communication Engineering

By:

## Ayush Kumar 229303417 Section- A



April, 2024

**Department of Computer and Communication Engineering School of Computing and Intelligent Systems**

**Manipal University Jaipur**

VPO. Dehmi Kalan, Jaipur, Rajasthan, India – 303007

**TABLE OF CONTENTS**

Contents Page Number

Student Declaration i

[Abstract ii](#_TOC_250002)

List of Figures 0

1. Introduction 1
2. Methodology 2
3. Entity-Relationship Model 3-4
4. Key Operations in Database (Screenshots of Codes & description)
   1. Creation of tables
   2. Insertion of Data

5-8

1. Key features in Database (Screenshots of Codes & description)
   1. Indexes
   2. Stored Procedures
   3. Triggers

9-10

1. Frontend(Code and Screenshot) 11-12
2. Backend Connectivity 13-14
3. Conclusion 15
4. Bibliography 16

# STUDENT DECLARATION

*I hereby declare that this project Railway Ticket Management System is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the University or other Institute, except where due acknowledgements has been made in the text.*

Place: Manipal University Jaipur Ayush Kumar

Date: 23/04/2024 Registration Number: 229303417

B.Tech. (CCE) 4th Semester Section:A

# ABSTRACT

The Web-Based Railway Ticket Booking System (RTBS) represents a pioneering effort to modernize and streamline the ticket booking process for railway passengers. This project focuses on the development of a user-friendly web interface that allows users to input essential personal information, including Name, Date of Birth (DOB), User ID, and Age. The collected data is then seamlessly integrated into the existing database infrastructure, facilitating efficient ticket reservation and management.

At its current stage of development, the RTBS offers a robust framework for capturing and updating user data within the railway system's database. Through a series of webpages, users can enter their personal details, which are validated and securely transmitted to the database backend. This functionality lays the foundation for future enhancements, including the implementation of online ticket booking, seat reservation, and itinerary management features.

Key features of the RTBS include:

* Intuitive webpage interface for user data input
* Data validation mechanisms to ensure accuracy and completeness
* Secure transmission protocols to protect sensitive information
* Seamless integration with the existing railway database architecture
* Scalable framework to accommodate future expansion and feature development

While the project is currently in a partially completed state, significant progress has been made in laying the groundwork for a comprehensive railway ticket booking solution. By prioritizing database integration and user data management, the RTBS sets the stage for future advancements in online ticketing functionality, ultimately aiming to enhance the overall passenger experience and efficiency of railway operations.

In summary, the Web-Based Railway Ticket Booking System represents a promising step towards modernizing ticketing infrastructure and embracing digitalization within the railway industry. With further development and refinement, it holds the potential to revolutionize the way passengers interact with and utilize railway services, paving the way for a more convenient, accessible, and user-centric travel experience.

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table** | **Page Number** |
| 1. | Flow Chart of the Methodology | 2 |
| 2. | ER Diagram for Railway Ticket Management System | 3 |
| 3. | Frontend Screenshot | 12 |
| 4. | XAMPP Server Screenshot | 14 |
| 5. | MyPHPAdmin Screenshot | 14 |

**INTRODUCTION**

**Objective:**

The Railway Ticket Management System (RTMS) project aims to develop a sophisticated software solution that streamlines and enhances the ticketing process for railway passengers. At its core, the objective is to create a user-friendly platform that simplifies ticket booking, reservation management, and administrative tasks for both passengers and railway authorities.

**Importance:**

In today's rapidly evolving transportation landscape, efficient ticket management systems are vital for ensuring smooth and convenient travel experiences. The RTMS project recognizes the significance of modernizing railway ticketing infrastructure to meet the growing demands of passengers while improving operational efficiency for railway authorities. By providing a comprehensive solution that integrates advanced technology and user-centric design principles, the RTMS project seeks to enhance the overall quality of service and customer satisfaction within the railway industry.

**Software Requirements:**

The successful implementation of the Railway Ticket Management System necessitates the utilization of various software tools and technologies. Key software requirements include:

1. Database Management System (MySQL): To store and manage passenger data, ticketing information, and other relevant records efficiently.

2. Web Development Framework (HTML, CSS, PHP): For building the user interface, backend logic, and web-based functionalities of the ticket booking system.

**Hardware Requirements:**

1. Web Server: To host the web application and handle incoming user requests. We have used XAMPP server.

2. Database Server: To host the database management system and ensure reliable data storage and retrieval we have used MySQL server.

3. Networking Equipment: Including routers, switches, and firewalls to establish network connectivity and ensure data security.

4. Client Devices: Such as desktop computers, laptops, tablets, and smartphones, which users will utilize to access the ticket booking system via web browsers.

By carefully addressing these software and hardware requirements, the RTMS project aims to create a robust and scalable ticket management solution that meets the needs of both passengers and railway authorities, ushering in a new era of efficiency and convenience in railway ticketing services.

METHODOLOGY

A computer diagram with text and images

Description automatically generated with medium confidence

1. **Requirements Analysis**

* Conducted a thorough analysis of the requirements for the Railway Ticket Management System.
* Identified key entities and relationships to develop an Entity-Relationship Diagram (ERD).
* Defined the functionalities and features required for the system.

1. **Database Design**

* Designed an ER Diagram based on the identified entities and relationships.
* Translated the ER Diagram into a relational database schema.
* Implemented the database schema in MySQL, ensuring normalization to eliminate data redundancy and maintain data integrity.

1. **Frontend Development**

* Created a user-friendly interface using HTML and CSS.
* Designed the webpage layout and visual elements to enhance user experience.
* Ensured compatibility across different devices and browsers for seamless access.

1. **Backend Development**

* Developed the backend logic using PHP to handle user requests and interactions.
* Established connection between the webpage and MySQL database for data retrieval and manipulation.

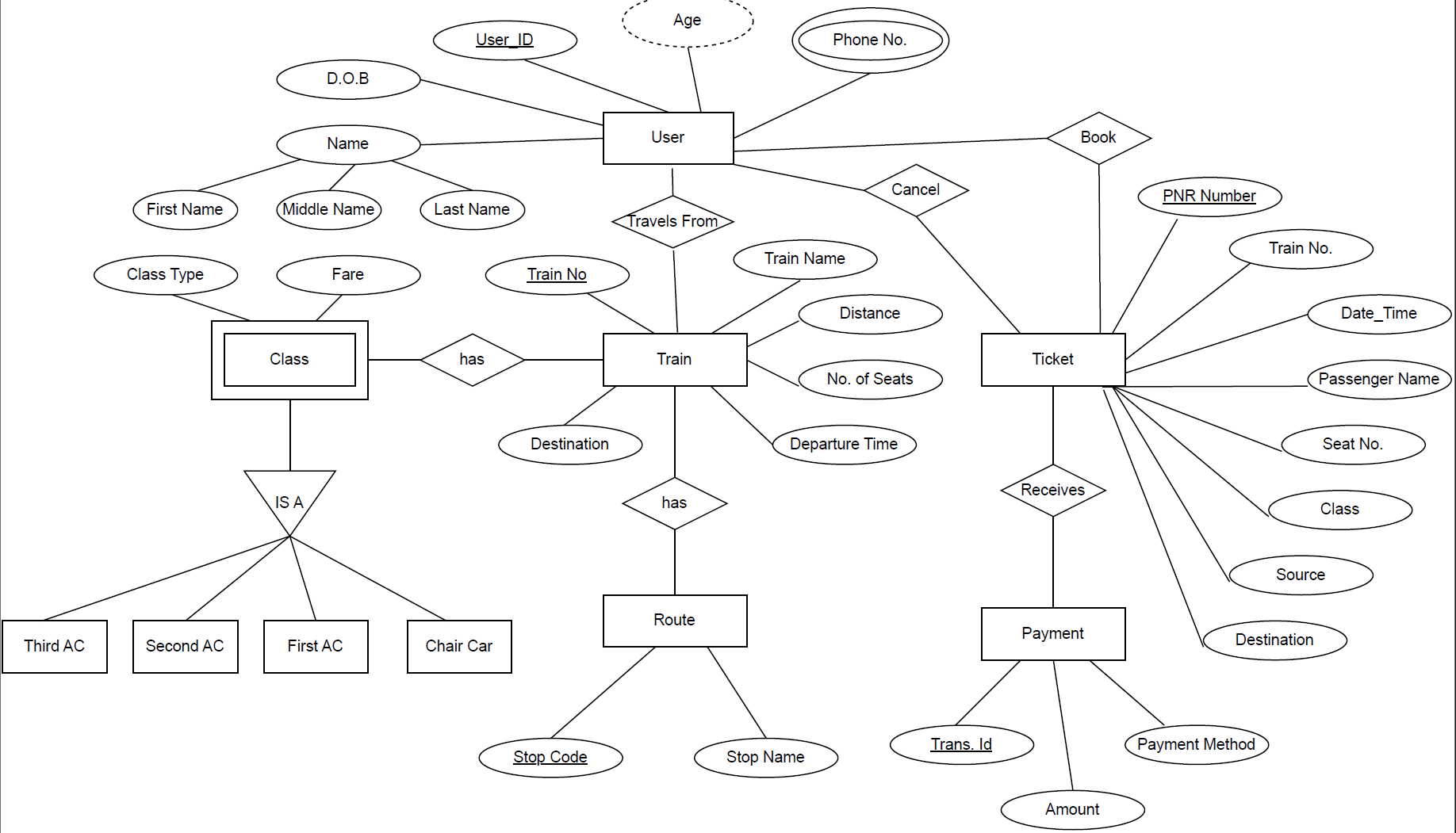
1. **Server Setup**

* Installed and configured XAMPP server to host the PHP web application locally.
* Deployed MySQL server to manage the database backend.
* Tested the server setup to ensure smooth functioning of the application.

1. **Integration and Testing**

* Integrated the frontend and backend components to create a cohesive system.
* Conducted extensive testing to validate the functionality and performance of the application.

ER DIAGRAM

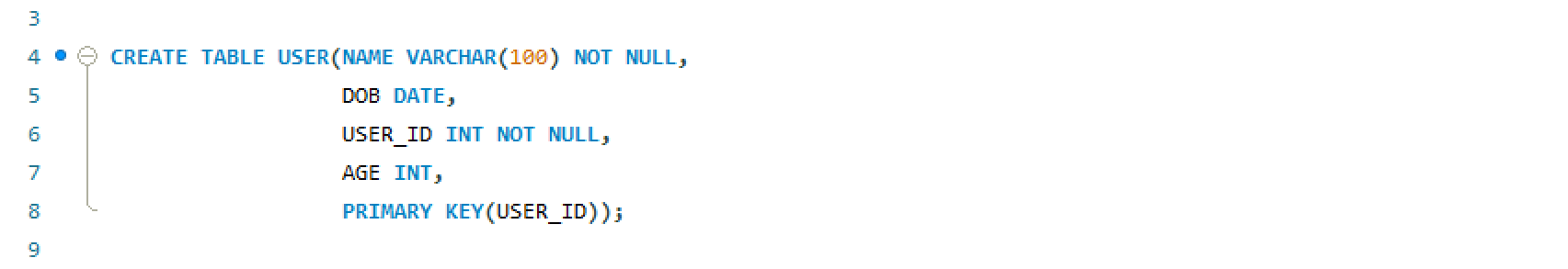


1. **User Table**:
   * Contains information about individual users.
   * Attributes:
     + **UserID**: Unique identifier for each user.
     + **Name**: Composite attribute with **First Name** and **Last Name**.
     + **Gender**: Multivalued attribute (can have multiple values, e.g., Male and Female).
     + **Email**: User’s email address.
     + **Phone No.**: Contact number.
     + **Age**: Age of the user.
     + **DOB**: Date of birth.
2. **Train Table**:
   * Holds data related to different trains.
   * Attributes:
     + **TrainID**: Unique identifier for each train.
     + **Name**: Name of the train.
     + **No of Seats**: Number of available seats in the train.
     + **Departure Time**: Scheduled departure time.
3. **Route Table**:
   * Stores information about various routes that trains take.
   * Attributes:
     + **RouteID**: Unique identifier for each route.
4. **Ticket Table**:
   * Contains data on tickets issued to passengers.
   * Attributes:
     + **TicketID**: Unique identifier for each ticket.
     + **Price**: Price of the ticket.
     + **Booked**: Indicates whether the ticket is booked or not.
     + **User**: Foreign key referencing the User table.
     + **Train**: Foreign key referencing the Train table.
5. **Payment Table**:
   * Holds records of payments made by users for their tickets.
   * Attributes:
     + **PaymentID**: Unique identifier for each payment.
     + **Date/Time**: Timestamp of the payment.
     + **Amount**: Payment amount.
     + **Ticket**: Foreign key referencing the Ticket table.

KEY OPERATIONS IN DATABASE

Creation of Tables :

* 1. User Table :



* 1. User1 and DOB Table :

A screenshot of a computer code

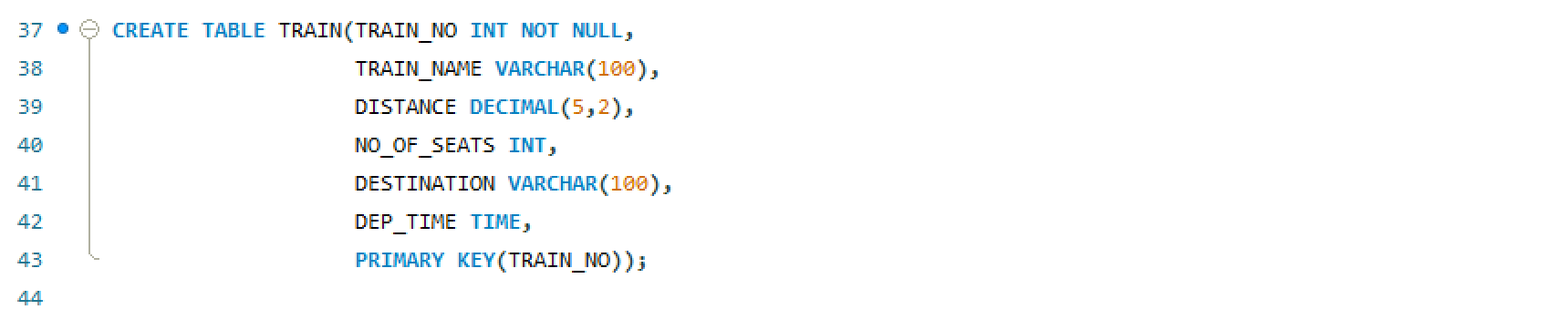
Description automatically generated

* 1. Phone\_no Table :

A white background with text

Description automatically generated

* 1. Train Table :



* 1. Ticket Table :

A computer code with text

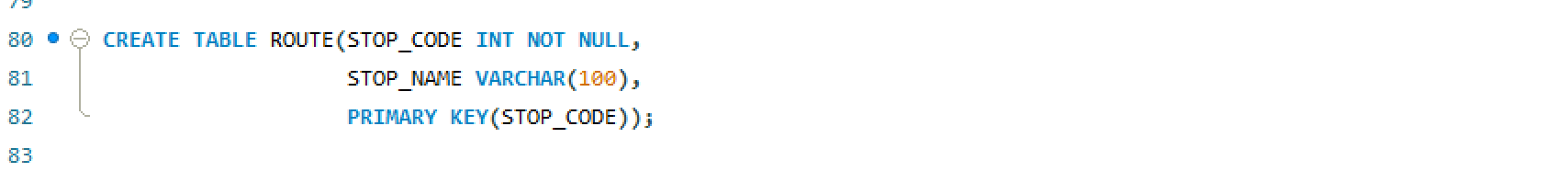
Description automatically generated with medium confidence

* 1. Payment Table :

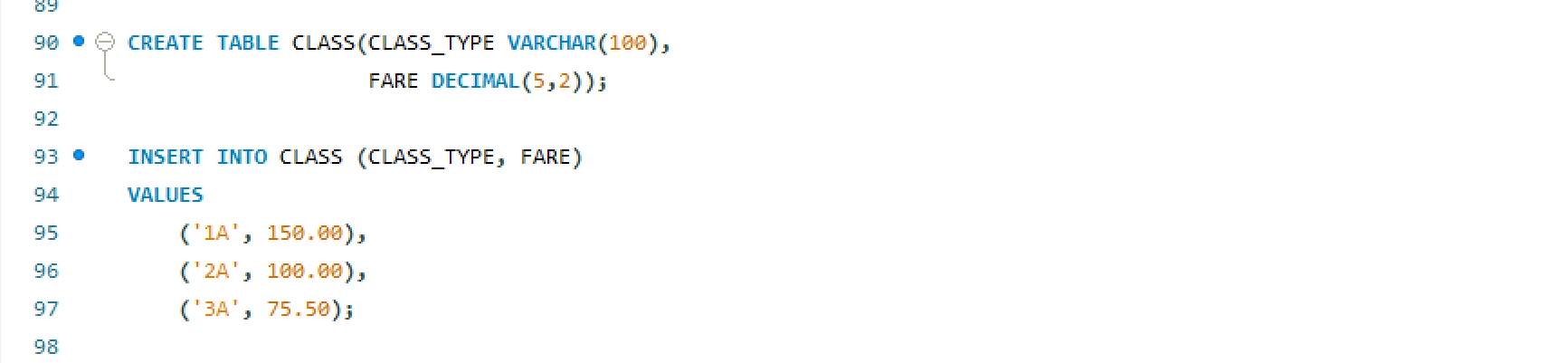
A close-up of a blue and white object

Description automatically generated

* 1. Route Table :



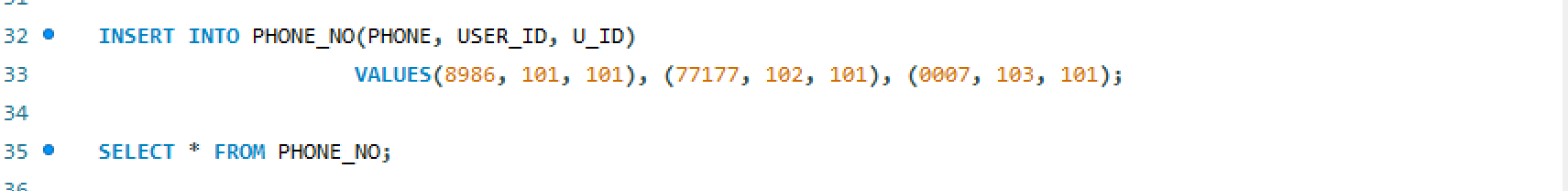
* 1. Class Table :



* 1. DOB Table :



Insertion Of Data in Tables :



A close-up of a number

Description automatically generated

A white background with black text

Description automatically generated

A white background with black and white clouds

Description automatically generated

A group of text on a white background

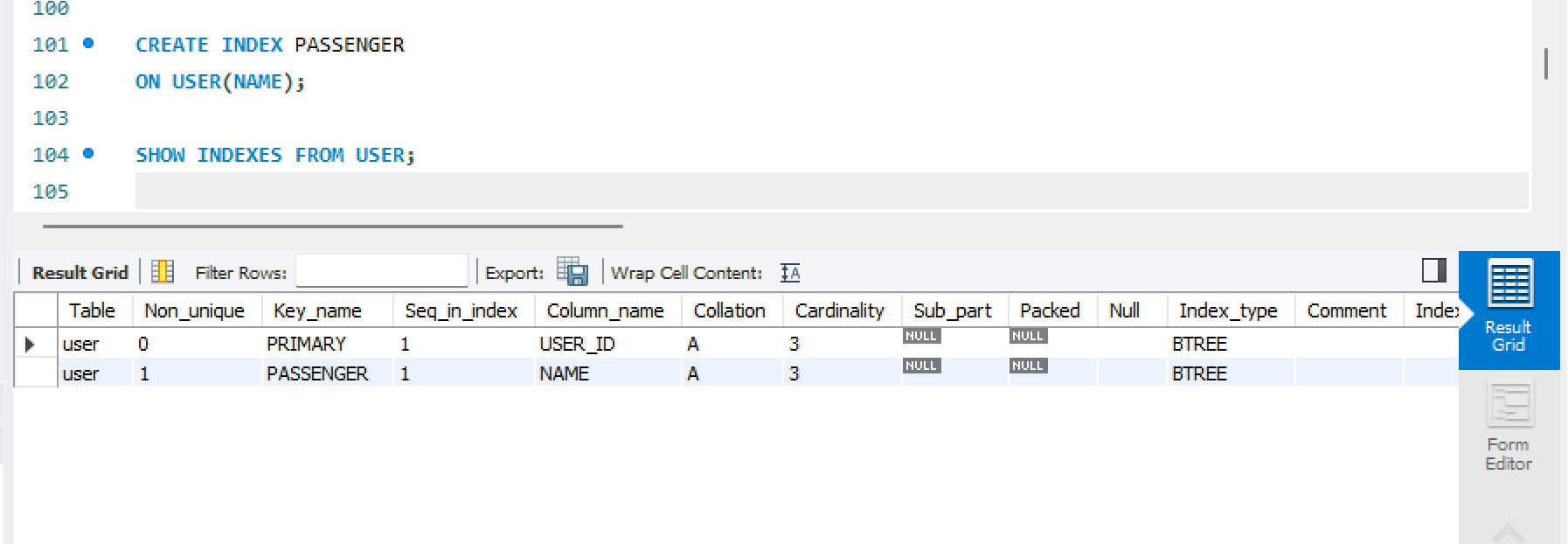
Description automatically generated

A white background with black and white clouds

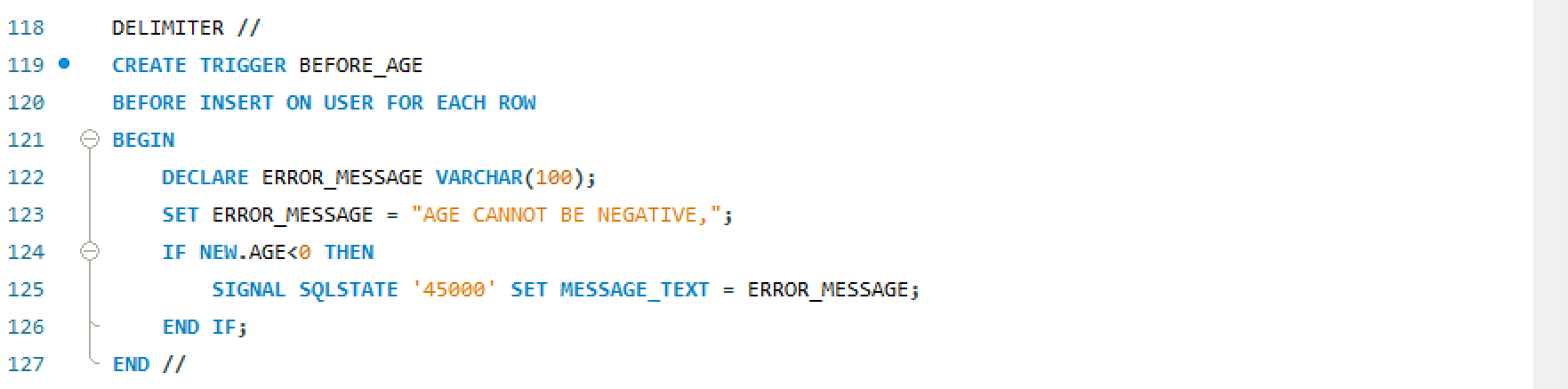
Description automatically generated

KEY FEATURES IN DATABASE

Indexes :



Trigger :



View :

A white background with black and white clouds

Description automatically generated

Stored Procedures :

A screenshot of a computer

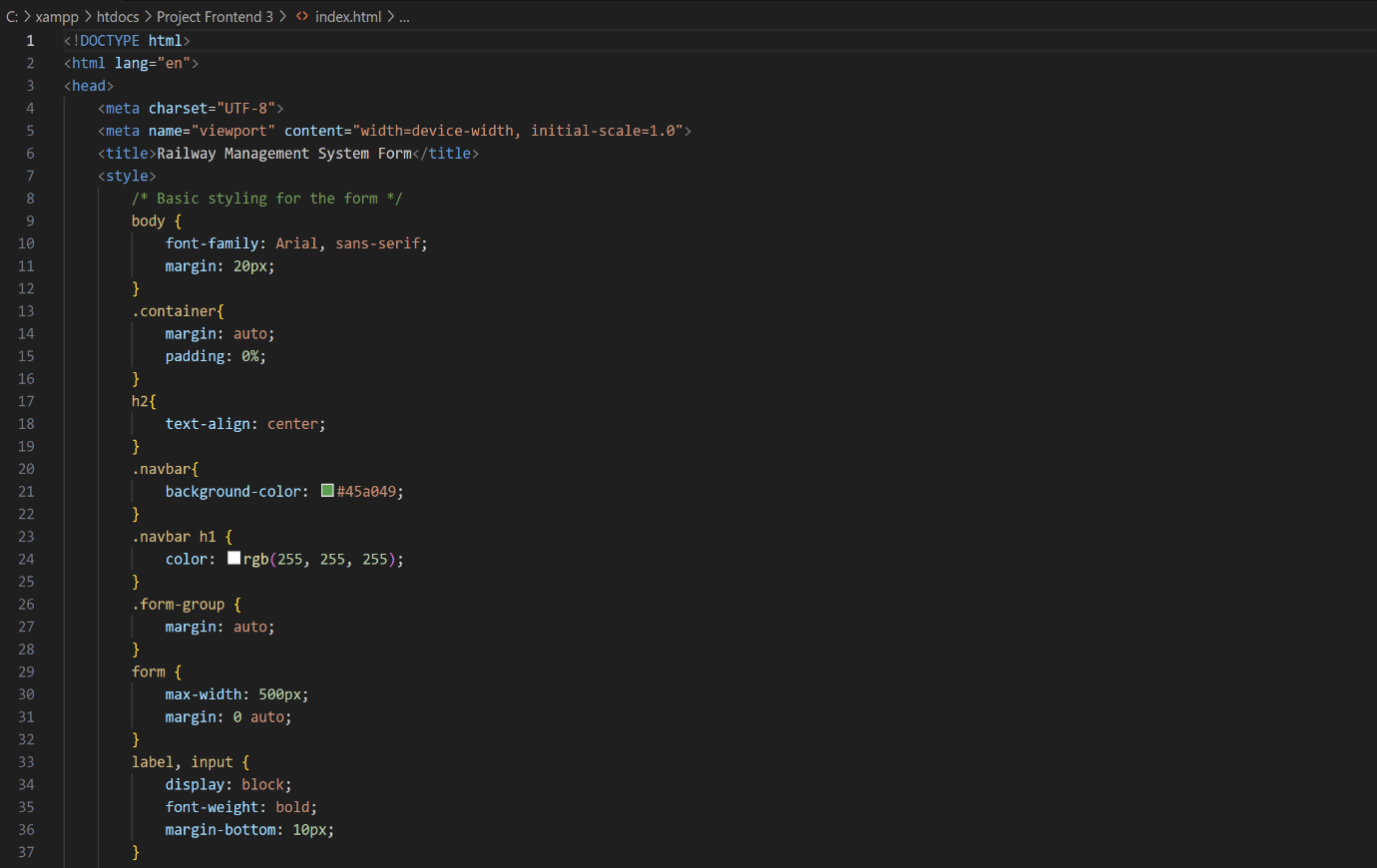
Description automatically generated

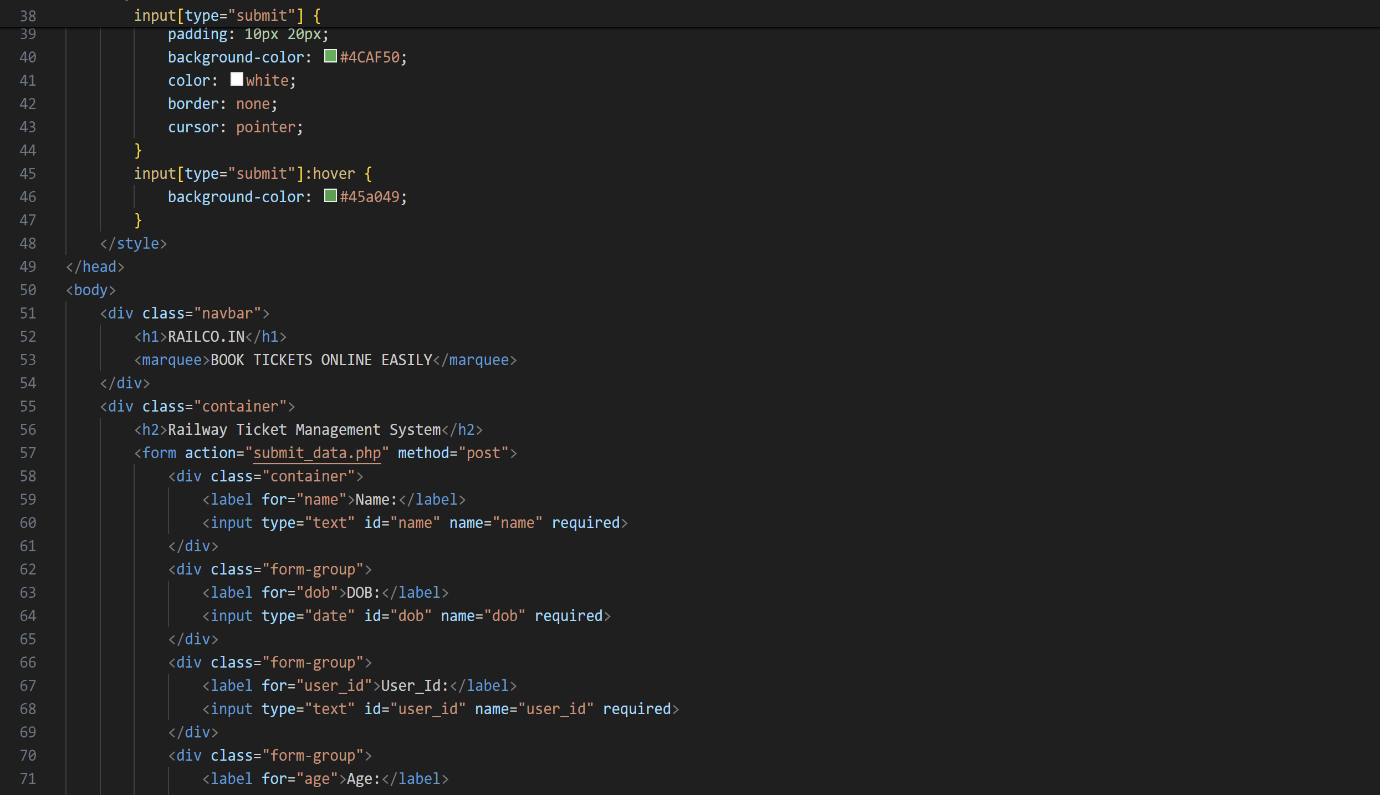
A screenshot of a computer

Description automatically generated

FRONTEND

HTML Code :





A black screen with white text

Description automatically generated

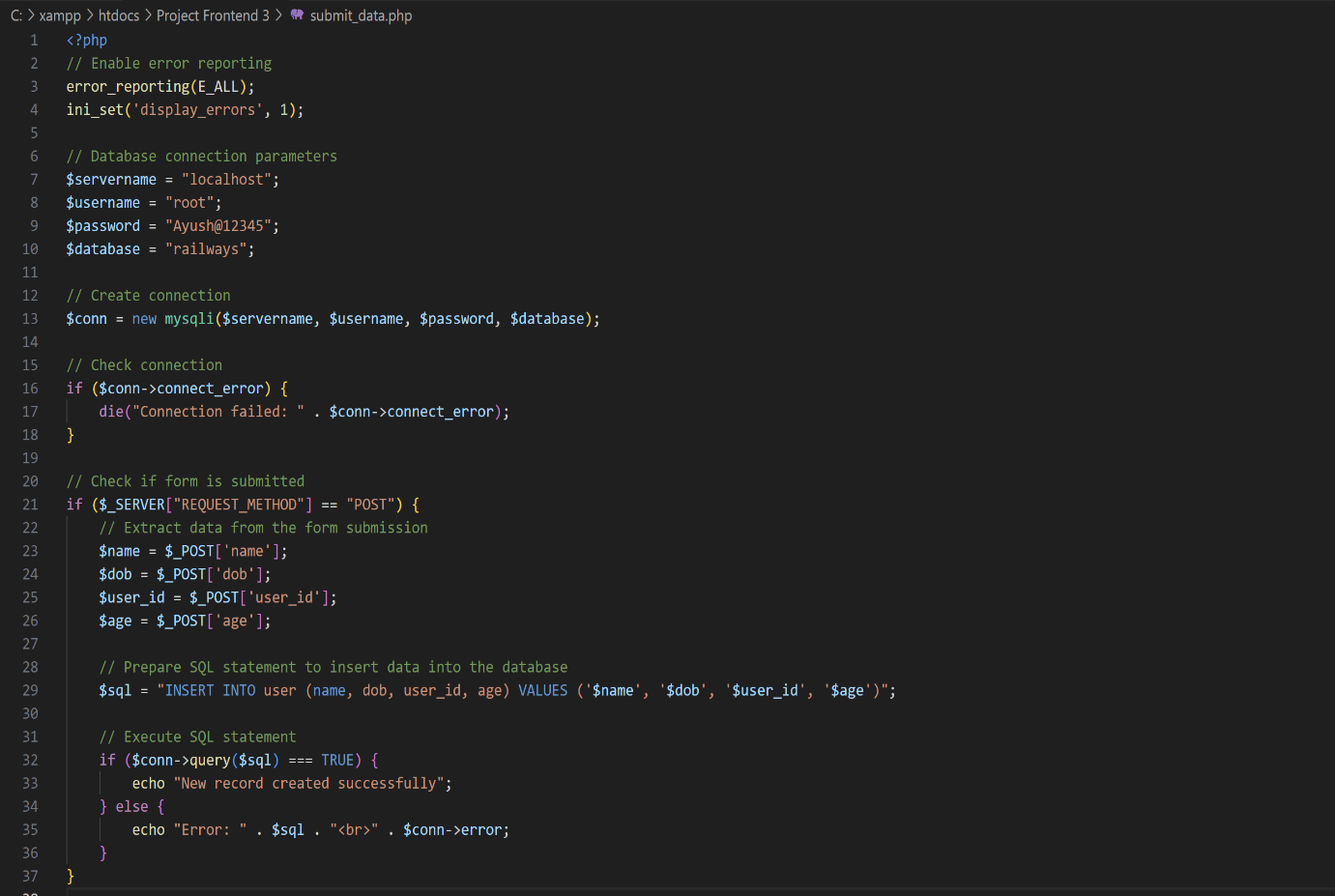
Frontend Screenshot :

A screenshot of a computer

Description automatically generated

BACKEND CONNECTIVITY

PHP Code :



A black rectangle with white text

Description automatically generated

XAMPP Server :

A screenshot of a computer

Description automatically generated

MyPHPAdmin :

A screenshot of a computer

Description automatically generated

CONCLUSION

In this partially completed Railway Ticket Management System project, significant progress has been made in laying the foundation for a robust database management system. The project utilizes a combination of MySQL, PHP, HTML, CSS, and XAMPP server to create a platform for efficient ticket management in the railway domain.

MySQL serves as the backbone for storing and organizing data related to tickets, trains, schedules, and passengers. Despite being partially completed, the database design has been structured based on sound principles, ensuring data integrity and efficiency through normalization techniques.

PHP has been employed to develop backend logic, enabling dynamic content generation and interaction with the MySQL database. Although the backend functionalities are not fully implemented, PHP provides the flexibility needed to handle various operations such as ticket booking, schedule management, and user authentication.

On the frontend side, HTML and CSS have been used to design a user-friendly interface. The interface, although not fully polished, lays the groundwork for a visually appealing and intuitive user experience. Responsive design principles have been incorporated to ensure compatibility across different devices and screen sizes.

The XAMPP server setup allows for local hosting of the project, providing a stable environment for development and testing. Apache serves the PHP files seamlessly, while MySQL handles database operations efficiently.

While this project is partially completed, it represents a solid foundation for further development. Future work includes implementing remaining backend functionalities, refining the frontend interface, and conducting thorough testing to ensure the system's reliability and performance. With continued effort and refinement, this Railway Ticket Management System has the potential to streamline ticketing processes and improve the overall efficiency of railway operations.

BIBLIOGRAPHY

1. **YouTube**:
   * Various video tutorials on building railway ticket management systems were referenced for understanding concepts such as database management, user interface design, and system architecture.
   * Example: HTML and CSS for Beginners, SuperSimpleDev, <https://youtu.be/G3e-> cpL7ofc?si=CaYQ4PHqg-cujewR
2. **W3Schools**:
   * W3Schools was used for learning web development technologies like HTML, CSS, JavaScript, and server-side scripting language PHP.
3. **Tutorials Point**:
   * Tutorials Point provided in-depth tutorials on various programming languages, databases, and frameworks. It was used for learning topics related to backend development.
4. **Google**:
   * Google was used extensively for finding relevant articles, documentation, and troubleshooting solutions during the development process.
   * Various online resources found through Google search were referred to for specific coding problems, architectural decisions, and best practices.