UNIVERSITY OF HOUSTON

Railway Management System Project

EDS 6397: Database Management Tools

Final Project

By

Group

Bagireddy Navya Madhuri

Damilola Eniola Olowu

Felipe Flores

Jaswanth Sai Mididodla

Keerthi Priya Veerapalli

Muhammad Asad Rehan

Onyinyechi C Ihesiulo

Poojitha Kandari

Quynh Nhu Thi Nguyen

Tejaswini Kasinadhuni

Course given by:

Dr. Lucy Nwosu

Department of Engineering

Engineering Data Science

ABSTRACT

This project focuses on developing a comprehensive web-based Railway Management System that streamlines railway operations, manages train schedules, facilitates ticket booking, and ensures efficient passenger transportation. The system's core functionality includes enabling users to inquire about available trains based on their source and destination, book and cancel tickets, and check the status of their booked tickets.

To achieve this, a robust database was designed and developed to maintain records of different trains, stations, and users. The train records encompass details such as train numbers, names, and operating days. Users can book tickets for trains with available seats by providing the desired train number and date of travel. Before confirming the booking, the system validates the train number and booking date. Once validated, it checks for seat availability. If seats are available, the ticket is booked with a confirmed status, and a unique ticket number is generated and stored along with the user's details.

The system also allows users to cancel their booked tickets at any time by providing the ticket ID, which serves as a unique key. The system searches for the ticket ID and deletes the corresponding record.

Emphasis was placed on designing a well-structured Entity Relationship Diagram (ERD) and Relational Database Design, adhering to established Business Rules and a comprehensive Data Dictionary. This approach ensured consistency and integrity across the system's components, including the website's functionality.

CONTENTS

1. INTRODUCTION

1.1. PROJECT OVERVIEW

The railway system is a vital transportation network that facilitates the efficient movement of passengers and goods across vast distances. With the increasing demand for convenient and reliable rail travel, there is a pressing need for a comprehensive and modern railway management system. This project aims to develop a robust web-based Railway Management System that streamlines railway operations, manages train schedules, facilitates ticket booking, and ensures efficient passenger transportation.

The Railway Management System is a comprehensive web application designed to revolutionize the way railway operations are managed and passenger experiences are delivered. It provides a centralized platform for users, railway staff, and administrators to interact and perform various tasks related to train schedules, ticket reservations, seat allocations, and passenger information management.

1.2. KEY MODULES

The Railway Management System comprises the following key modules:

1. **User Registration and Authentication:** Enables users to securely register, log in, and manage their accounts.

2. **Train Management:** Facilitates the management of train schedules, routes, and availability for railway staff and administrators.

3. **Ticket Booking:** Allows passengers to search for trains, view schedules, and book tickets seamlessly.

4. **Seat Reservation:** Empowers passengers to select and reserve their preferred seats on trains.

5. Payment Gateway Integration: Integrates with secure payment gateways to facilitate

convenient and safe ticket payments.

6. **User Database:** Comprehensive user management system for storing and managing user details, ticket reservations, and travel history.

7. **Admin Panel:** Provides a centralized interface for administrators to manage trains, tickets, passengers, and resolve operational issues.

1.3. USER ROLES

The Railway Management System caters to the needs of three distinct user roles:

1. **Passengers:** End-users who can search for train schedules, book tickets, and manage their travel plans.

2. **Railway Staff:** Authorized personnel responsible for managing train operations, schedules, and seat allocations.

3. **Administrators:** Privileged users with complete access to system administration, configuration, and reporting functionalities.

1.4. SECURITY AND COMPLIANCE

The Railway Management System prioritizes security and compliance by implementing robust user authentication, authorization mechanisms, secure encryption protocols, and adhering to regulatory requirements and data protection laws specific to the railway industry, ensuring the highest standards of data privacy and security.

1.5. TECHNOLOGIES USED

Front End: Combination of HTML, CSS, Javascript.

Back End: Flask, Python, SQLAlchemy

1.6. CHALLENGES ENCOUNTERED

The challenges encountered on this project include:

1. Front-end challenges due to inexperience with HTML, CSS, and JavaScript.

2. Back-end integration challenges to maintain consistency between the Entity Relationship Diagram, Relational Database Design, Business Rules, Data Dictionary, and the Website.

2. SYSTEM ANALYSIS AND METHODS

2.1. METHODOLOGY

• First create train(s)

• Assign route(s) to said train which would then create a set number of seats and tickets corresponding to the capacity of the train

• As a customer you can now select a route you would like to use (**Note:When selecting you will be able to picture the desired seat**)

• Be able to manage the tickets you have reserved by either checking the ticket information or canceling the ticket.

• Designing the database schema using an appropriate database management system (DBMS) such as MySQL.

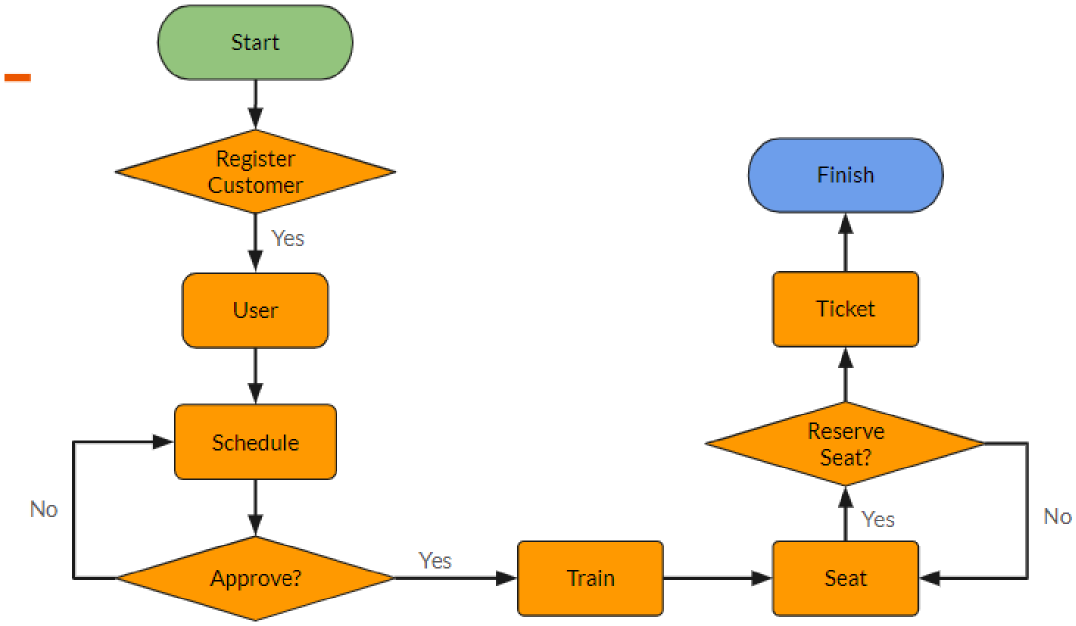


Figure 2.1: Flow map of Customer Booking Journey

2.2. BUSINESS RULES

1. **Relationships: User - Ticket**

   (a) One to many: only one registered user can book multiple tickets.

2. **Relationships: Train - Seat**

   (a) One to many: a train can has multiple seats.

3. **Relationships: Train - Ticket**

   (a) One to many: A train can have multiple tickets.

4. **Relationships: Train - Schedule**

   (a) One to many: multiple schedules can be assigned to a train.

5. **Relationships: Schedule - Ticket**

   (a) One to many: one schedule may have many tickets.

6. **Relationships: Schedule - Seat**

   (a) One to many: one schedule can be assigned to many seat.

7. **Relationships: Ticket - Seat**

   (a) One to one: one ticket can have only one seat.

2.3. LIST OF ENTITIES

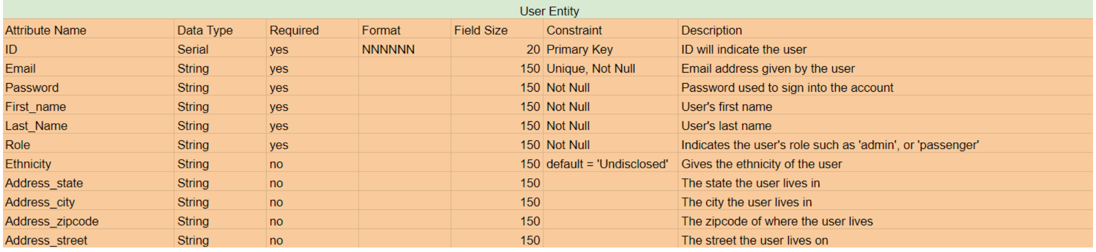
- **Passenger:** Represents registered users who can search for trains, book tickets, and make reservations. They must provide valid and accurate information during registration and ticket booking processes. Passengers can also apply promotional codes or special offers.

- **Railway Staff:** Represents staff members with access to manage train schedules, routes, availability, and passenger information. They can only access and modify information relevant to their assigned tasks and responsibilities.

- **Role:** a defined set of responsibilities and expectations assigned to an individual, such as the railway staff member. The description of roles is managed by administration members.
- **Train:** Represents train information such as schedules, routes, and availability. Managed by Railway Staff members and Administrators.
- **Seat:** Represents seats on trains. Passengers can select and reserve multiple seats based on their specific needs, with a maximum of 2 seats per transaction.
- **Ticket:** Represents booked tickets, including payment information and cancellation policies. Each ticket booking must have a corresponding payment, and passengers are entitled to cancel their tickets within a specific period.
- **Payment:** Represents payment information and the payment gateway. Payment must be completed through the integrated payment gateway before tickets are confirmed.

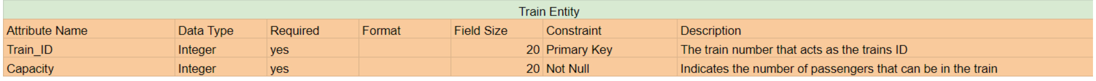
2.4. DATABASE DICTIONARY

Here shows the data dictionary for all the Entities.



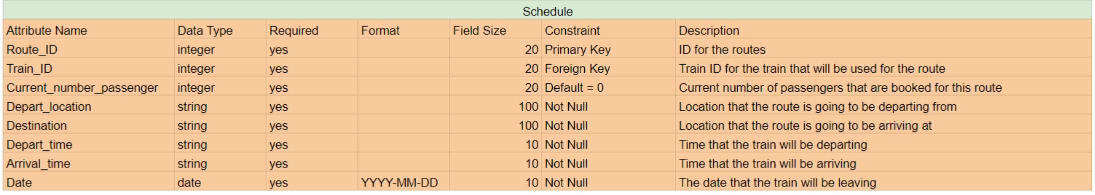
User Entity						
Attribute Name	Data Type	Required	Format	Field Size	Constraint	Description
ID	Serial	yes	NNNNNN	20	Primary Key	ID will indicate the user
Email	String	yes		150	Unique, Not Null	Email address given by the user
Password	String	yes		150	Not Null	Password used to sign into the account
First_name	String	yes		150	Not Null	User's first name
Last_Name	String	yes		150	Not Null	User's last name
Role	String	yes		150	Not Null	Indicates the user's role such as 'admin', or 'passenger'
Ethnicity	String	no		150	default = 'Undisclosed'	Gives the ethnicity of the user
Address_state	String	no		150		The state the user lives in
Address_city	String	no		150		The city the user lives in
Address_zipcode	String	no		150		The zipcode of where the user lives
Address_street	String	no		150		The street the user lives on

Figure 2.2: Data Dictionary for User Entity



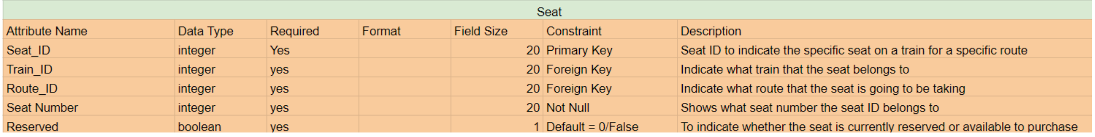
Train Entity						
Attribute Name	Data Type	Required	Format	Field Size	Constraint	Description
Train_ID	Integer	yes		20	Primary Key	The train number that acts as the trains ID
Capacity	Integer	yes		20	Not Null	Indicates the number of passengers that can be in the train

Figure 2.3: Data Dictionary for Train Entity



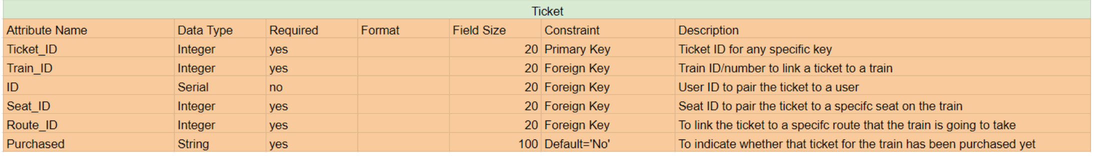
Schedule						
Attribute Name	Data Type	Required	Format	Field Size	Constraint	Description
Route_ID	integer	yes		20	Primary Key	ID for the routes
Train_ID	integer	yes		20	Foreign Key	Train ID for the train that will be used for the route
Current_number_passenger	integer	yes		20	Default = 0	Current number of passengers that are booked for this route
Depart_location	string	yes		100	Not Null	Location that the route is going to be departing from
Destination	string	yes		100	Not Null	Location that the route is going to be arriving at
Depart_time	string	yes		10	Not Null	Time that the train will be departing
Arrival_time	string	yes		10	Not Null	Time that the train will be arriving
Date	date	yes	YYYY-MM-DD	10	Not Null	The date that the train will be leaving

Figure 2.4: Data Dictionary for Schedule Entity



Seat						
Attribute Name	Data Type	Required	Format	Field Size	Constraint	Description
Seat_ID	integer	Yes		20	Primary Key	Seat ID to indicate the specific seat on a train for a specific route
Train_ID	integer	yes		20	Foreign Key	Indicate what train that the seat belongs to
Route_ID	integer	yes		20	Foreign Key	Indicate what route that the seat is going to be taking
Seat Number	integer	yes		20	Not Null	Shows what seat number the seat ID belongs to
Reserved	boolean	yes		1	Default = 0/False	To indicate whether the seat is currently reserved or available to purchase

Figure 2.5: Data Dictionary for Seat Entity



Ticket						
Attribute Name	Data Type	Required	Format	Field Size	Constraint	Description
Ticket_ID	Integer	yes		20	Primary Key	Ticket ID for any specific key
Train_ID	Integer	yes		20	Foreign Key	Train ID/number to link a ticket to a train
ID	Serial	no		20	Foreign Key	User ID to pair the ticket to a user
Seat_ID	Integer	yes		20	Foreign Key	Seat ID to pair the ticket to a specifc seat on the train
Route_ID	Integer	yes		20	Foreign Key	To link the ticket to a specifc route that the train is going to take
Purchased	String	yes		100	Default='No'	To indicate whether that ticket for the train has been purchased yet

Figure 2.6: Data Dictionary for Ticket Entity

3. RELATIONAL DIAGRAM

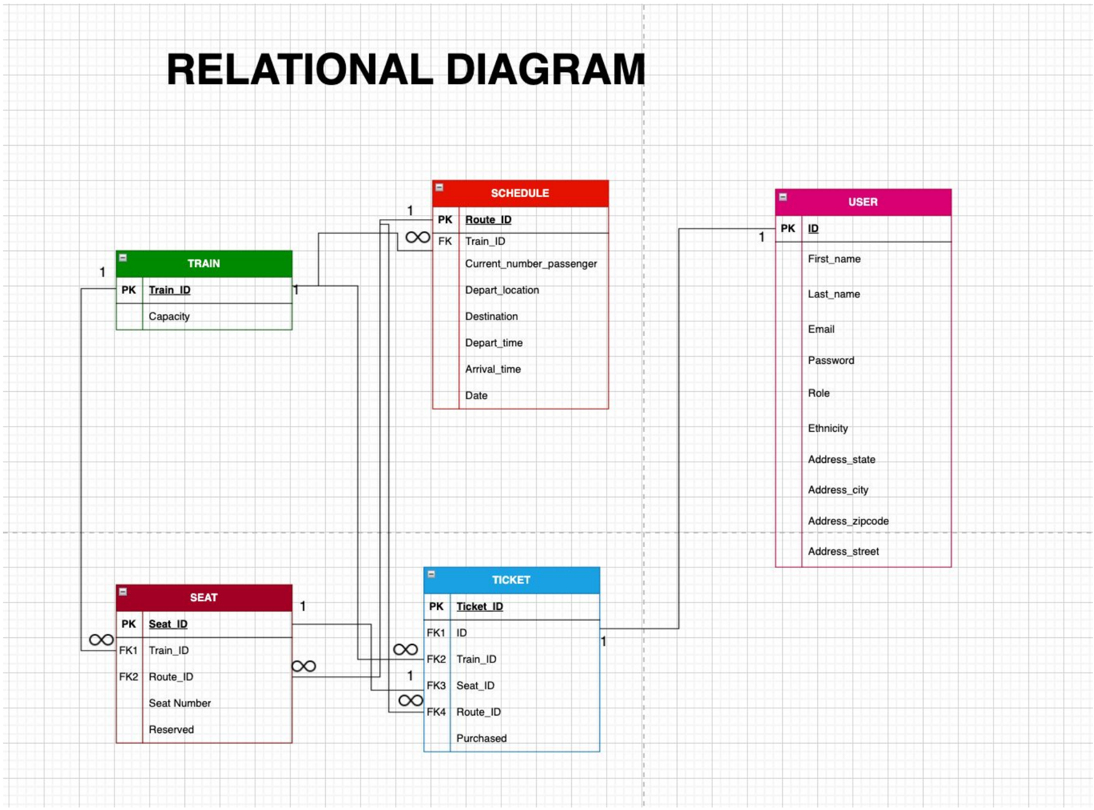


Figure 3.1: Relational Diagram

4. LOGICAL ENTITY RELATIONAL DIAGRAM (ERD)

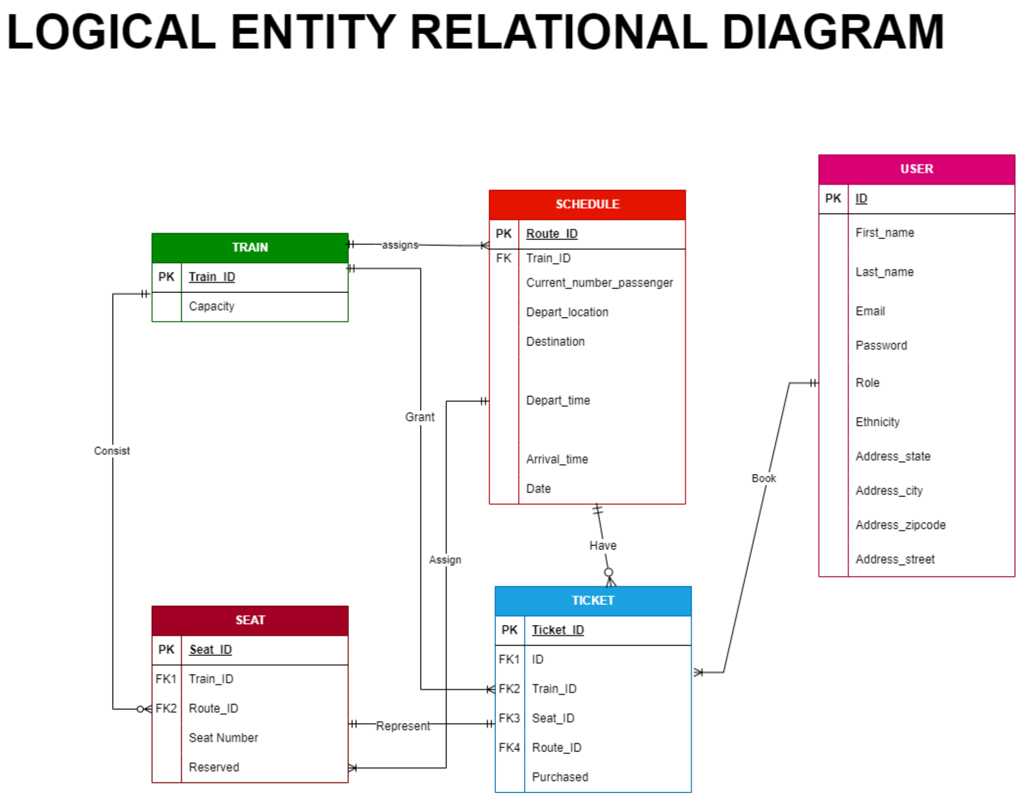


Figure 4.1: Logical Entity Relational Diagram

5. SQL QUERIES

```
    CREATE TABLE TRAIN (
    train_id INTEGER NOT NULL,
    capacity INTEGER NOT NULL,
    PRIMARY KEY (train_id)
    );
```

Listing 5.1: SQL Code for Creating the TRAIN Table

```
    new_train = Train(capacity=capacity)
    db.session.add(new_train)
    db.session.commit()

    In SQL this would be the following:

    INSERT INTO Train (capacity) VALUES (value_of_capacity);
```

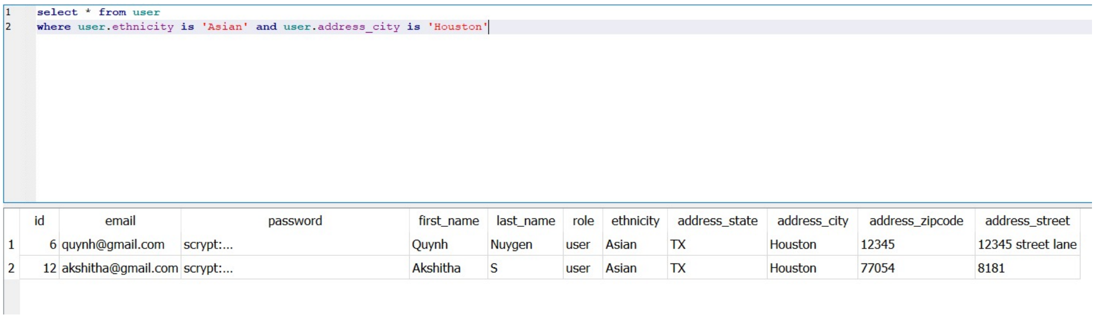
Listing 5.2: Queries being made using SQLAlchemy and their SQL Counterparts

```
routes = db.session.query(Schedule, Train).join(Schedule).filter(Schedule.date == query).all()

In SQL this would be the following:

SELECT schedule.*, train.*
FROM schedule
JOIN train ON schedule.train_id = train.train_id
WHERE schedule.date = 'query_date';
```

Listing 5.3: Questies being made using SQLAlchemy and their SQL Counterparts



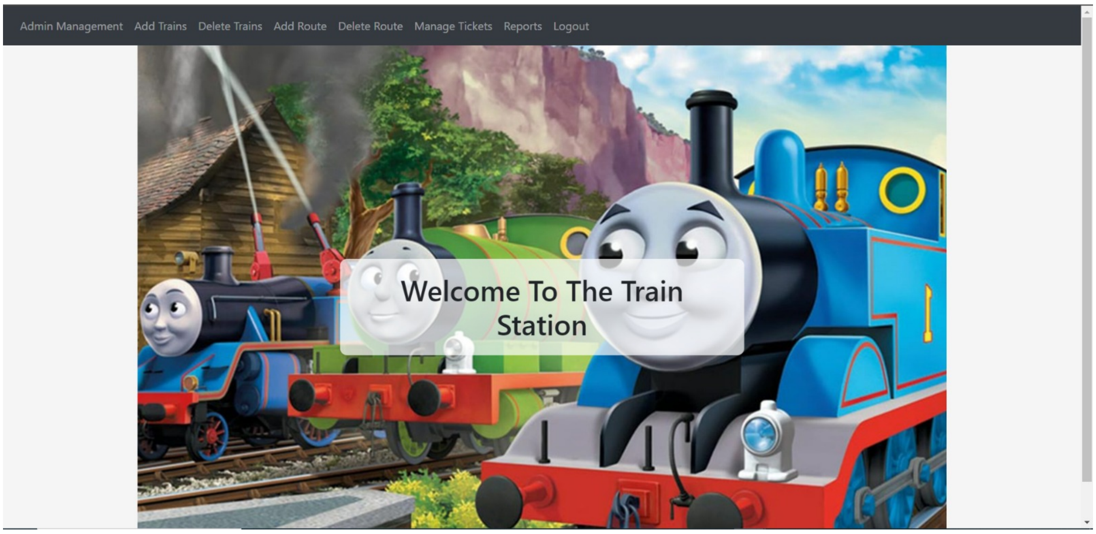
```
select * from user
where user.ethnicity is 'Asian' and user.address_city is 'Houston'
```

	id	email	password	first_name	last_name	role	ethnicity	address_state	address_city	address_zipcode	address_street
1	6	quynh@gmail.com	scrypt:...	Quynh	Nuygen	user	Asian	TX	Houston	12345	12345 street lane
2	12	akshitha@gmail.com	scrypt:...	Akshitha	S	user	Asian	TX	Houston	77054	8181

Figure 5.1: Data Retrieval

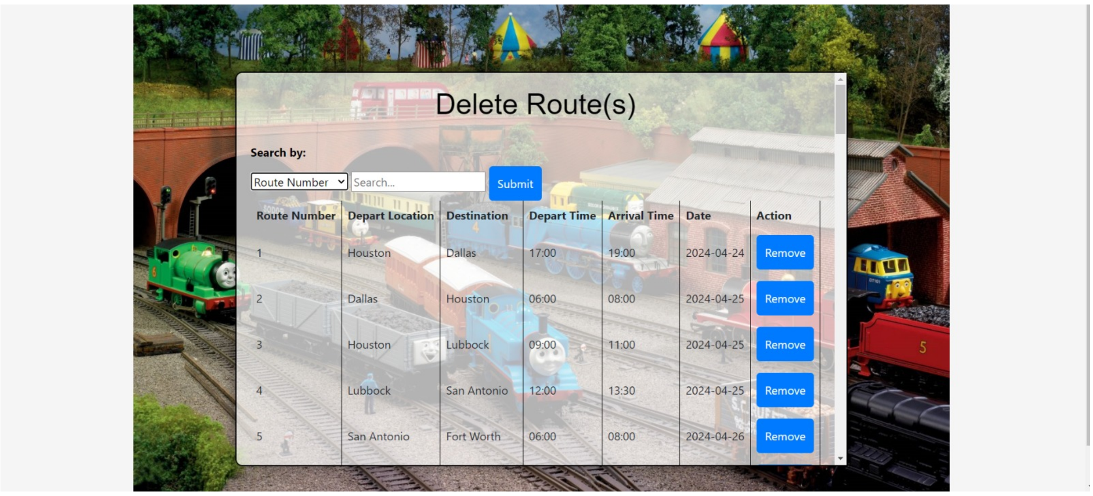
6. RESULTS

Here shows the Website Layout of Home Page.



Welcome To The Train Station

Figure 6.1: Home Page

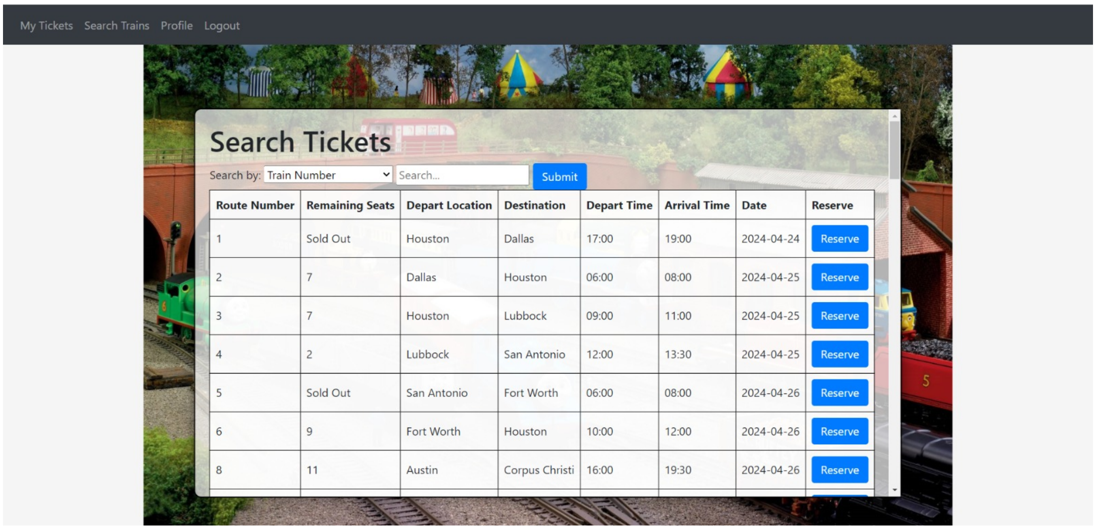


Delete Route(s)

Search by:

Route Number	Depart Location	Destination	Depart Time	Arrival Time	Date	Action
1	Houston	Dallas	17:00	19:00	2024-04-24	Remove
2	Dallas	Houston	06:00	08:00	2024-04-25	Remove
3	Houston	Lubbock	09:00	11:00	2024-04-25	Remove
4	Lubbock	San Antonio	12:00	13:30	2024-04-25	Remove
5	San Antonio	Fort Worth	06:00	08:00	2024-04-26	Remove

Figure 6.2: Page Shows to Delete trains by Admin

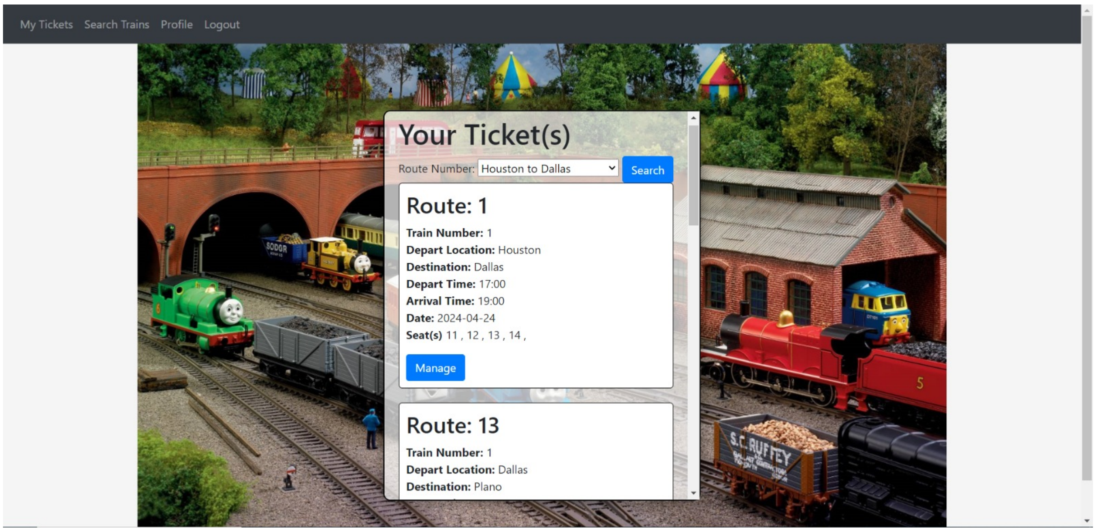


Search Tickets

Search by: Train Number | Search... | Submit

Route Number	Remaining Seats	Depart Location	Destination	Depart Time	Arrival Time	Date	Reserve
1	Sold Out	Houston	Dallas	17:00	19:00	2024-04-24	Reserve
2	7	Dallas	Houston	06:00	08:00	2024-04-25	Reserve
3	7	Houston	Lubbock	09:00	11:00	2024-04-25	Reserve
4	2	Lubbock	San Antonio	12:00	13:30	2024-04-25	Reserve
5	Sold Out	San Antonio	Fort Worth	06:00	08:00	2024-04-26	Reserve
6	9	Fort Worth	Houston	10:00	12:00	2024-04-26	Reserve
8	11	Austin	Corpus Christi	16:00	19:30	2024-04-26	Reserve

Figure 6.3: Page Shows the select the trains for different destinations



Your Ticket(s)

Route Number: Houston to Dallas | Search

Route: 1

Train Number: 1
Depart Location: Houston
Destination: Dallas
Depart Time: 17:00
Arrival Time: 19:00
Date: 2024-04-24
Seat(s) 11 , 12 , 13 , 14 ,

Manage

Route: 13

Train Number: 1
Depart Location: Dallas
Destination: Plano

Figure 6.4: Page Shows to With Ticket Confirmation

7. CONCLUSION AND FUTURE OUTLOOK

In our project, the Railway Management System project created a complete system for managing user information, ticket bookings, travel records, and generating useful reports to make good decisions. The team overcame difficulties in building the user interface and linking all the back-end parts together. The system follows important security and compliance rules for the railway industry. With its strong features and following best practices, the system is a valuable tool for better railway operations and improved user services.

Our project link: click here

Future Outlook:

- Our Railway Management system, evolving with AI, facilitates personalized bookings and displays pricing details in mobile apps for iOS and Android.
- Real-time updates and payment receipts enhance user experience.
- Collaborations offer integrated travel like Promotional codes and offers.
- Data analytics inform decisions. Security and sustainability initiatives ensure efficiency and eco-friendliness.