BUS BOOKING RESERVATION SYSTEM

Submitted By:

NAME	ROLL NUMBER
GOUTHAM KRISHNA	AM.EN.U4CSE21125
JOE VARGHESE	AM.EN.U4CSE21129
P M ARJUN KRISHNA	AM.EN.U4CSE21142
YASWANTH S	AM.EN.U4CSE21163

Abstract

This project report outlines the development and implementation of a comprehensive full-stack Bus Booking Reservation System. The system is designed to simplify and streamline the process of bus reservations, offering powerful features for both administrators and end-users. Developed with Angular for the front-end, Spring Boot for the back-end, and MySQL for the database, the system aims to provide a seamless user experience, ensuring robust connectivity between the database and the front-end application, and allowing for efficient booking processes. The architecture has been designed to be scalable and maintainable, supporting both user-facing functionalities and administrative operations. Key features include:

- **Admin Functionality**: Allows for managing buses, specifying seat arrangements, defining pricing structures, and controlling amenities offered on different buses.
- **User Capabilities**: Enables users to search for available buses, book tickets, retrieve ticket details by ID, and cancel bookings as needed. Users can also manage their journey details and track their booking status in real time.

This report provides an overview of the system architecture, a breakdown of the core functionalities, integration of backend services, and visual representations of the user interface with key screenshots, offering a comprehensive understanding of the system's operation.

Introduction

In today’s fast-paced world, transportation management plays a critical role in ensuring smooth and efficient travel experiences for users. The Bus Booking Reservation System is designed to meet this need by providing a user-friendly and functional platform for managing bus reservations. The application is intended to streamline the process of booking tickets, with key functionalities accessible to both administrators responsible for managing bus operations and passengers who need to book and manage their tickets.

The system aims to offer an optimized solution that not only enhances user experience through a dynamic and responsive front-end but also ensures secure and efficient backend operations. The project’s main objectives include:

1. **Centralized Platform for Reservation Management**: The system aims to create a unified platform that can be easily used by both administrators for managing buses and users for booking tickets. The centralized nature of the system ensures that all reservations, pricing, and related data are handled from one location, improving the overall efficiency of the operations.
2. **Data Integrity and Security**: Ensuring the security and integrity of data is a key priority for this system. The backend is designed with robust business logic and security measures to handle user data, ticket bookings, payment details, and other sensitive information. This is achieved through well-structured APIs and secure data transmission protocols.
3. **User Experience Enhancement**: A primary goal of the system is to provide an intuitive and seamless experience for users. The front-end, developed using Angular, incorporates dynamic elements and responsive design principles to ensure ease of use and accessibility across a variety of devices. This provides users with a smooth and engaging experience as they search for buses, book tickets, and manage their bookings.

Additionally, the application ensures users and administrators can interact with the system in a way that supports real-time data updates. For instance, seat availability is updated as soon as a booking is confirmed, ensuring users do not encounter conflicts when attempting to reserve seats. With these features, the Bus Booking Reservation System optimizes the bus reservation process, ensuring convenience for users and efficient management for administrators.

Contents

1. System Overview

Features

- **Admin**: Add bus details, amenities, and seat arrangements with pricing.
- **User**: Search for buses, book tickets, retrieve and cancel tickets by ID.

Technologies Used

- **Front-End**: Angular
- **Back-End**: Spring Boot
- **Database**: MySQL

---

2. System Architecture

The architecture follows an MVC (Model-View-Controller) pattern:

- **Model**: Represents data structures and database operations.
- **View**: User interface created using Angular components.
- **Controller**: Handles requests and responses, implemented in Spring Boot.

---

3. Features Description

Admin Features

1. Login authentication.
2. Add and manage buses.
3. Define seat arrangements and pricing.
4. Specify amenities for buses.

User Features

1. Search for buses using date, source, and destination.
2. Book tickets and view booking details.
3. Search tickets by ID for quick access.
4. Cancel tickets as needed.

4. Working of the system

The Bus booking system operates seamlessly by integrating the front-end, back-end, and database components. Here is a detailed explanation of the working of the system:

User Interface (UI) - Angular Front-End:

- Bus Search: Users begin by providing the source, destination, and travel date in the search section. Upon submission, the Angular app communicates with the back-end Spring Boot API to fetch available buses based on the user's input.
- Seat Selection: Once the user selects a bus, they can view a dynamic visual seat map that shows available and occupied seats. The availability is updated in real-time to ensure no double booking.
- Voucher Application: After selecting seats, the user can apply applicable vouchers, provided the cart meets the eligibility criteria. The voucher system integrates with the cart system to provide a discount.
- Booking and Ticket Generation: The final step in the user flow is the booking process. After entering required details, the ticket is generated with a unique ticket ID, confirming the reservation. The user is provided with the ticket and can view or cancel it later by using the ticket ID.

Admin Panel:

- Bus Management: Admins have access to a control panel where they can add new buses, specify seat arrangements, set pricing, and add amenities such as Wi-Fi, air conditioning, etc.
- Voucher Management: Admins can create new vouchers, specify discount criteria, and assign them to specific routes or bookings.
- Passenger Information: Admins can view and manage the passenger details for each bus, allowing them to monitor bookings and address any issues.

Backend - Spring Boot:

- Controller Layer: The back-end is designed using Spring Boot, which handles incoming HTTP requests from the front-end (Angular). The controllers process requests related to bus searches, seat availability, ticket bookings, cancellations, and more.
- Service Layer: The service layer performs the core business logic, such as applying vouchers, checking seat availability, calculating the ticket price based on seat selection, and handling booking cancellations with refunds.
- Database Interaction: The system uses MySQL as its database. It stores information about buses, seat arrangements, ticket bookings, users, and vouchers. Each user and admin interaction triggers appropriate database transactions to ensure data consistency and integrity.

Real-Time Updates:

- Seat Availability: A key feature is the real-time seat availability. As a user books a seat, the back-end ensures that the seat is marked as booked and unavailable for other users. This is managed through proper transaction handling and database updates.
- Dynamic Pricing: Ticket pricing can vary based on bus availability, selected seat class, and other factors. The back-end dynamically calculates the price based on these variables and passes it to the front-end for display.

5. Implementation Details

Backend Code

- **Database Connectivity**

```
server.port=9093

spring.datasource.url=jdbc:mysql://localhost:3306/happytrip?useSSL=false
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
logging.level.org.hibernate=DEBUG
logging.level.org.hibernate.SQL=DEBUG
•
```

- **DAO for bus**

```java
package com.busbooking.happytripbackend.dao;

import java.util.Date;
import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

import com.busbooking.happytripbackend.entity.busEntity;

@Repository
public interface busDao extends JpaRepository<busEntity, Long>{

    @Query("select be from busEntity be where be.source = ?1 and 
be.destination= ?2 and be.departureDate= ?3")
    public List<busEntity> getAllBusesUsingRouteAndDate(String source, 
String destination, String date);

    @Query("select be from busEntity be where be.busId= ?1 ")
    public busEntity getBusByBusId(Long busId);

}
```

- **Entity Model**

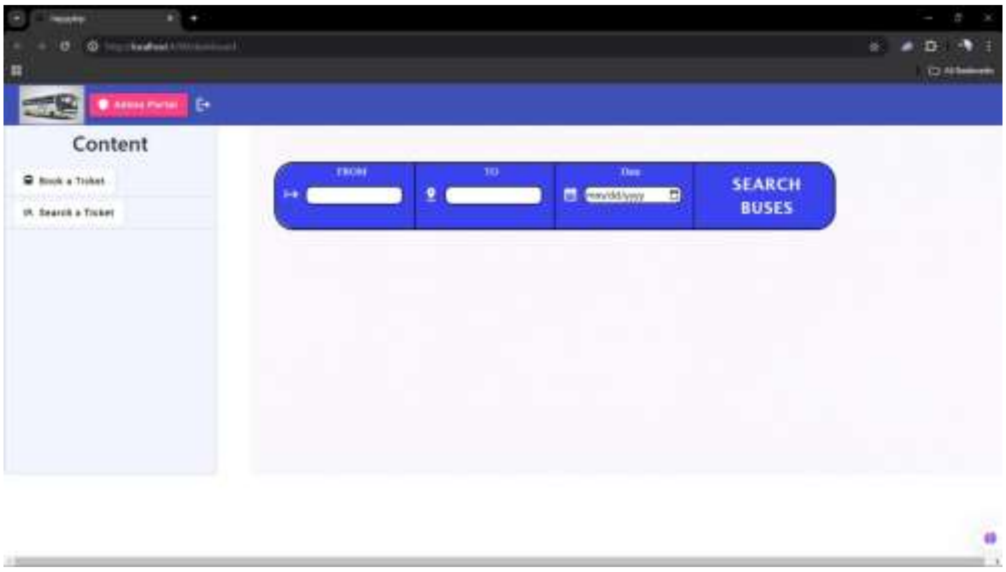
```
@Entity
@Table(name = "admin")
public class adminEntity {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private Long id;
    @Column(name = "username")
    private String userName;
    @Column(name = "password")
    private String password;

    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    @Override
    public String toString() {
        return "adminEntity [id=" + id + ", userName=" + userName + ",
password=" + password + "]";
    }

    public adminEntity(Long id, String userName, String password) {
        super();
        this.id = id;
        this.userName = userName;
        this.password = password;
    }
    public adminEntity() {
        super();
    }
}
```

- **Model (addnewUserRequest)**

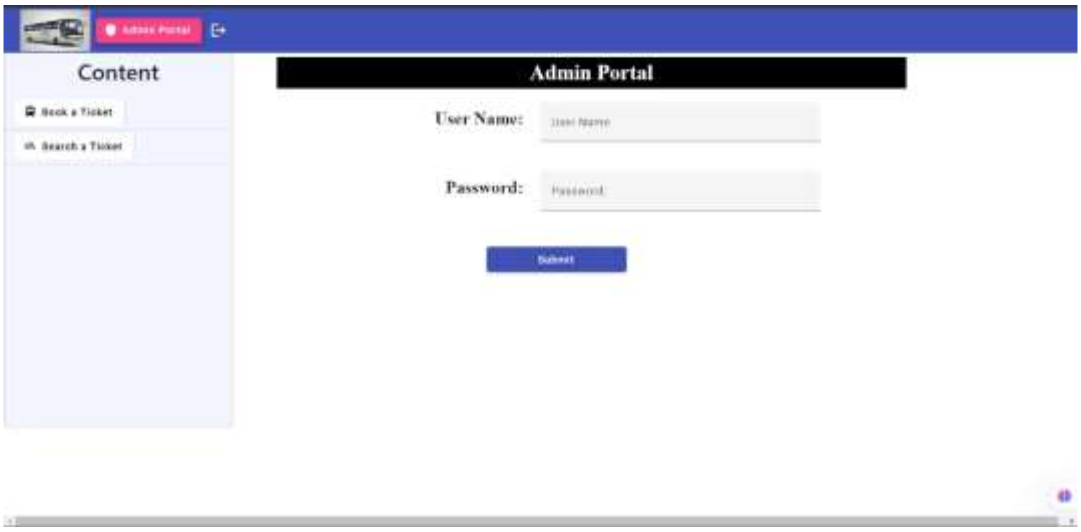
```
public class addNewUserRequest {

    private String userName;
    private String emailId;
    private String phoneNumber;

    public String getUserName() {
        return userName;
    }

    public void setUserName(String userName) {
        this.userName = userName;
    }

    public String getEmailId() {
        return emailId;
    }

    public void setEmailId(String emailId) {
        this.emailId = emailId;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public addNewUserRequest(String userName, String emailId, String
phoneNumber) {
        super();
        this.userName = userName;
        this.emailId = emailId;
        this.phoneNumber = phoneNumber;
    }
    @Override
    public String toString() {
        return "addNewUserRequest [userName=" + userName + ", emailId=" +
emailId + ", phoneNumber=" + phoneNumber
                + "]";
    }
    public addNewUserRequest() {
        super();
    }
}
```

6. Screenshots

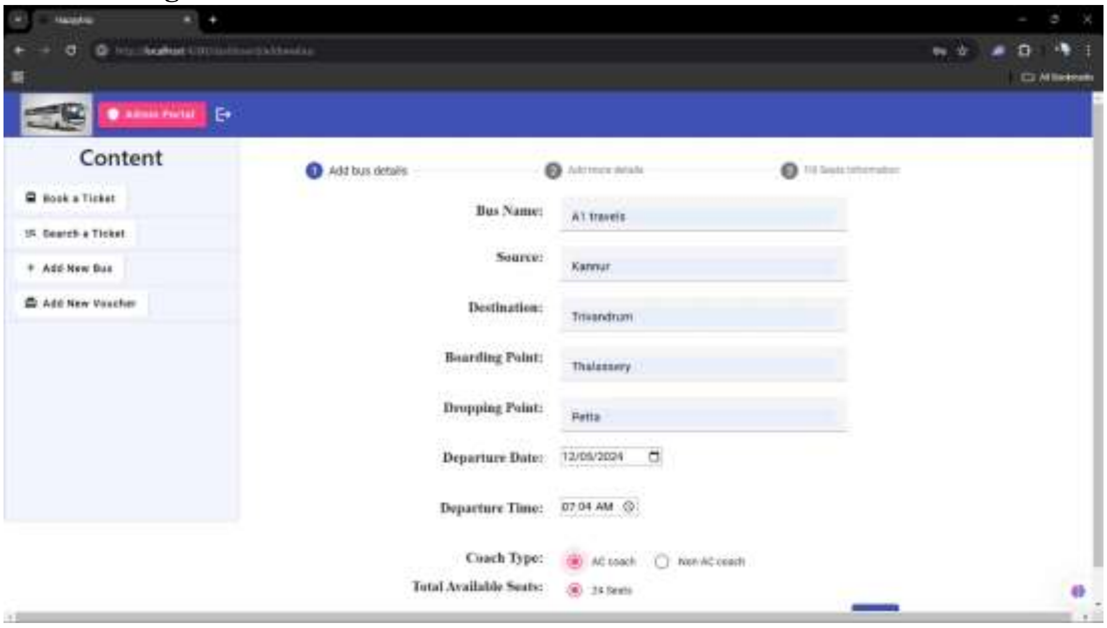
- **Homepage**



- **Admin Dashboard**



- **Bus Add Page**



Bus Name: A1 travels

Source: Kannur

Destination: Trivandrum

Boarding Point: Thalassery

Dropping Point: Petta

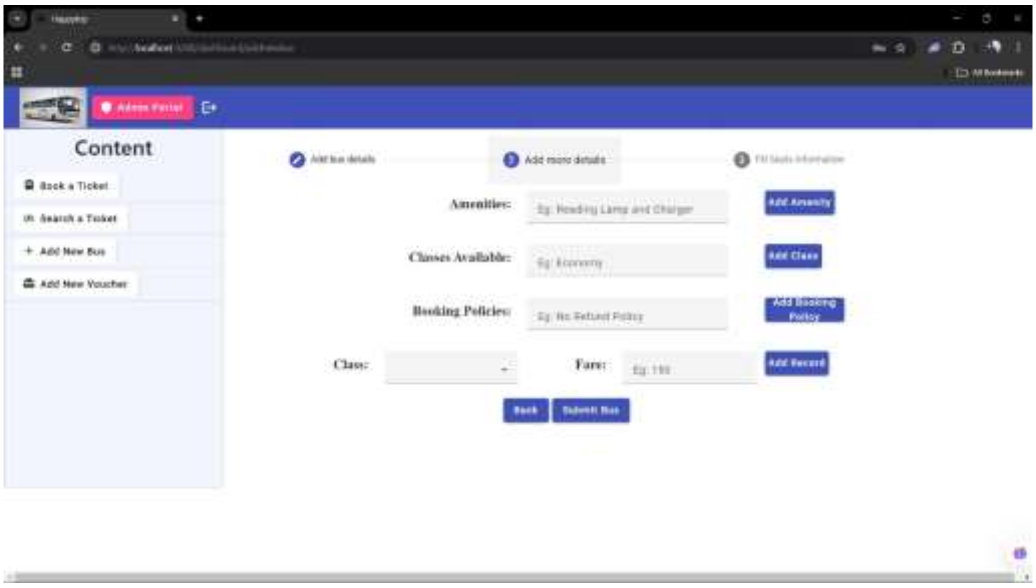
Departure Date: 12/05/2024

Departure Time: 07:04 AM

Coach Type: [✓ AC coach] Non AC coach

Total Available Seats: [✓ 24 Seats]

- **Amenities and Class Page**



Amenities: Eg: Reading Lamp and Charger

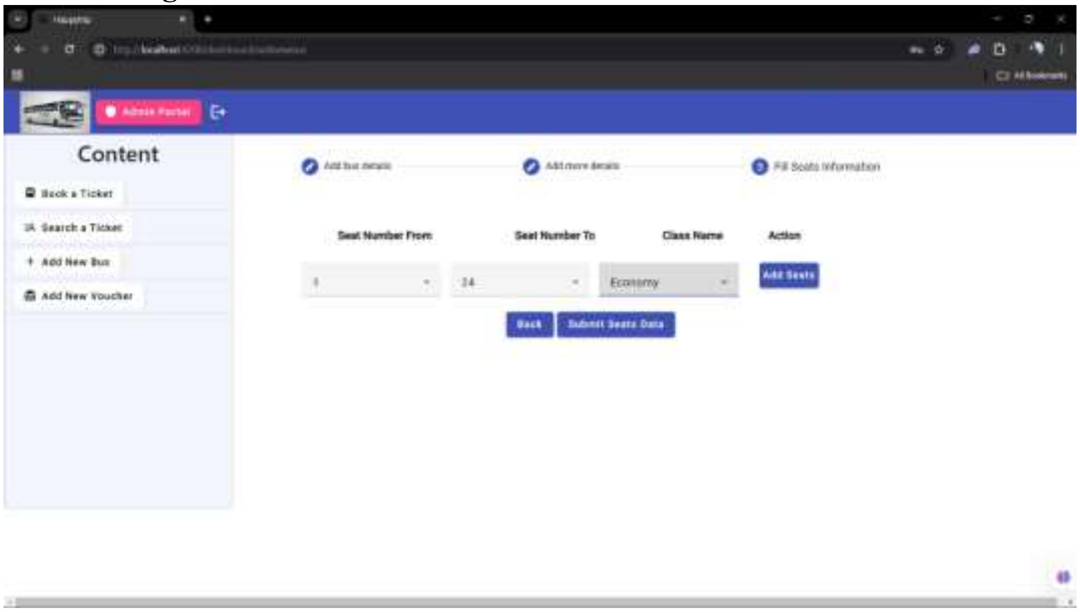
Classes Available: Eg: Economy

Booking Policies: Eg: No Refund Policy

Class:

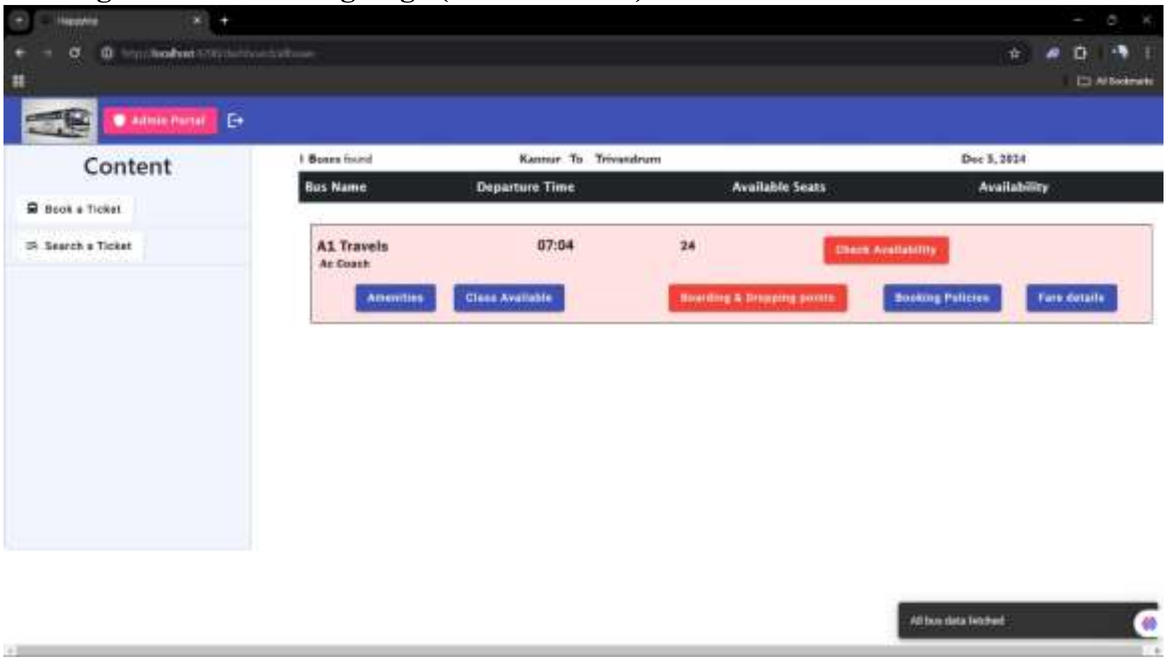
Fare: Eg: 199

- **Seat Data Page**



Seat Number From	Seat Number To	Class Name	Action
1	24	Economy	Add Seats

- **Passenger Ticket Booking Page (Available Bus)**

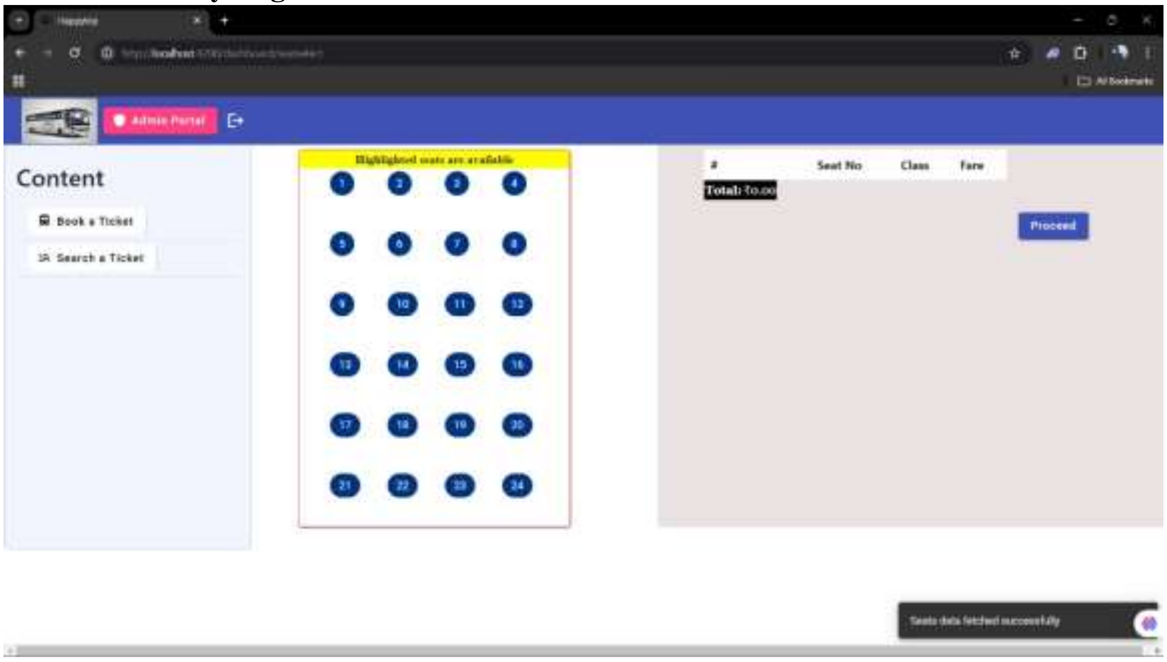


1 Buses found Kannur To Trivandrum Dec 5, 2024

Bus Name	Departure Time	Available Seats	Availability
A1 Travels Ac Coach	07:04	24	Check Availability

All bus data fetched

- **Seat Availability Page**

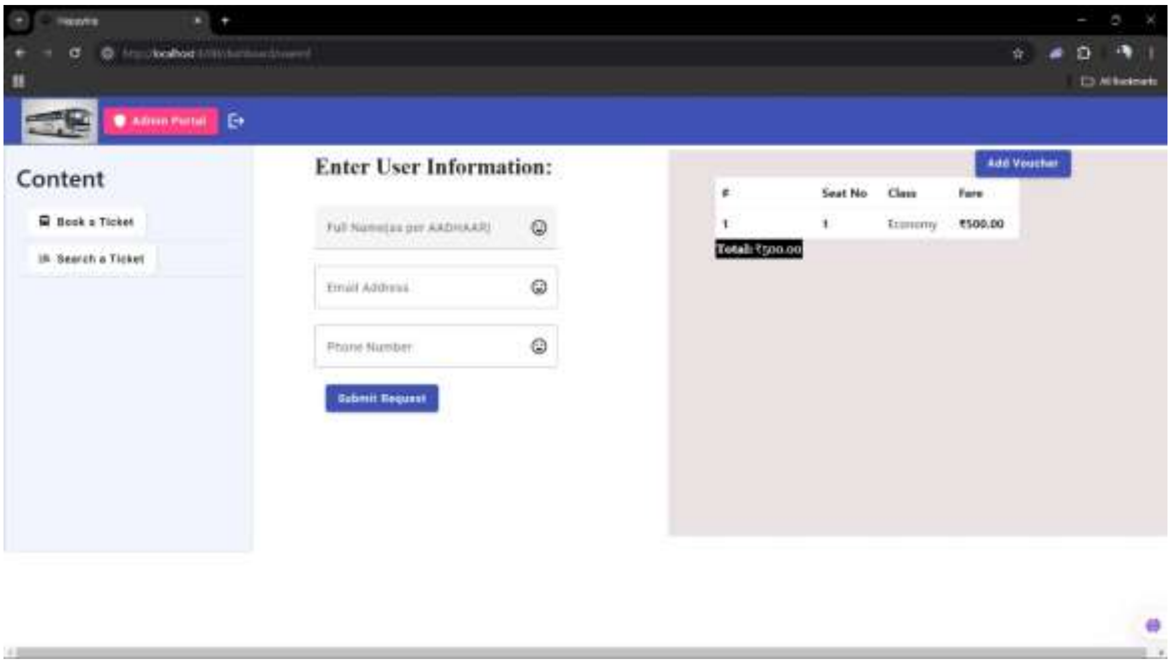


#	Seat No	Class	Fare

Total: ₹0.00

Seats data fetched successfully

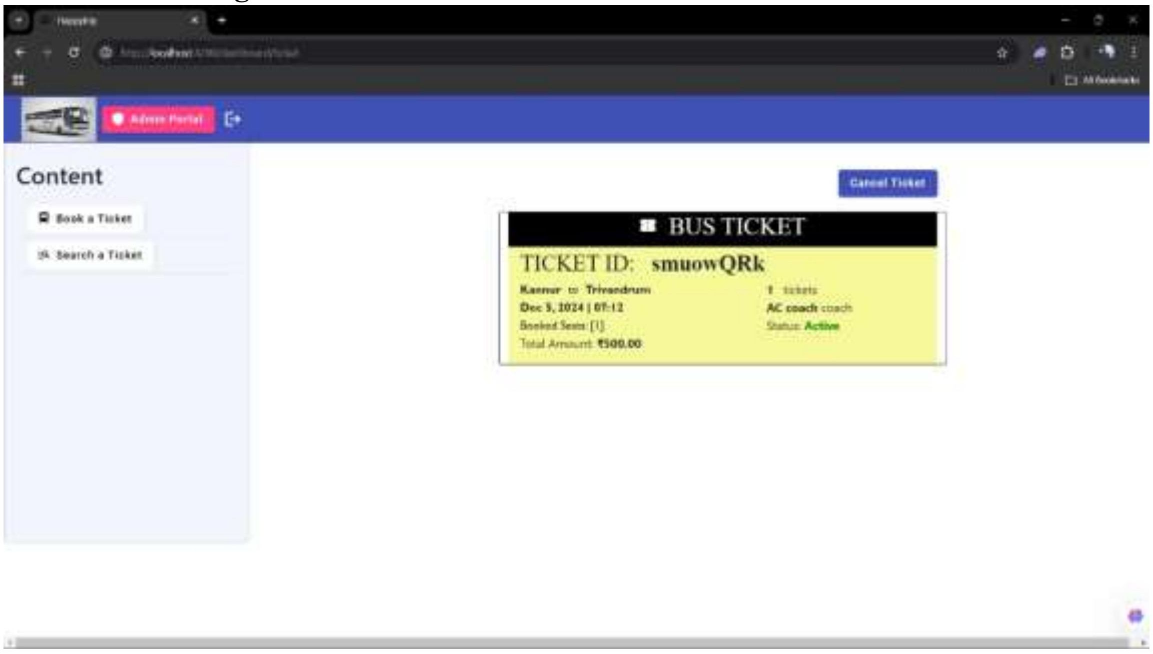
- **User Information Page**



#	Seat No	Class	Fare
1	1	Economy	₹500.00

Total: ₹500.00

- **Confirmation Page**



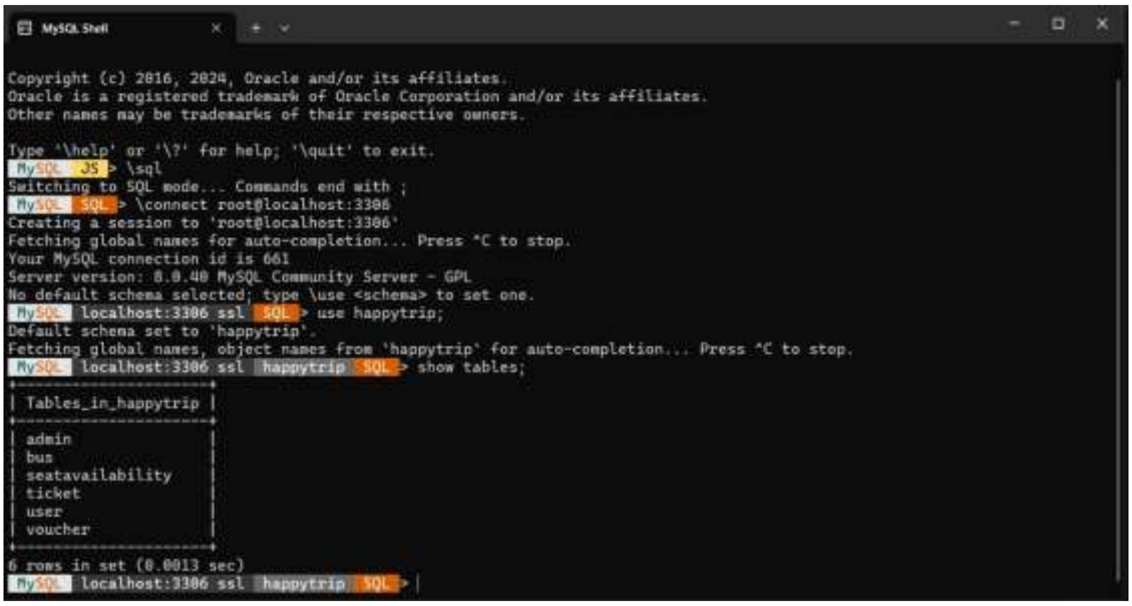
BUS TICKET

TICKET ID: **smuowQRk**

Kannur to **Trivandrum**
Dec 5, 2024 | 07:12
Booked Seats: [1]
Total Amount: **₹500.00**

1 tickets
AC coach coach
Status: **Active**

- **DATABASE STRUCTURE**



```
Copyright (c) 2016, 2024, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
 MySQL  JS > \sql
Switching to SQL mode... Commands end with ;
 MySQL  SQL > \connect root@localhost:3306
Creating a session to 'root@localhost:3306'
Fetching global names for auto-completion... Press ^C to stop.
Your MySQL connection id is 661
Server version: 8.0.40 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
 MySQL  localhost:3306 ssl  SQL > use happytrip;
Default schema set to 'happytrip'.
Fetching global names, object names from 'happytrip' for auto-completion... Press ^C to stop.
 MySQL  localhost:3306 ssl  happytrip  SQL > show tables;
+---------------------+
| Tables_in_happytrip |
+---------------------+
| admin               |
| bus                 |
| seatavailability    |
| ticket              |
| user                |
| voucher             |
+---------------------+
6 rows in set (0.0013 sec)
 MySQL  localhost:3306 ssl  happytrip  SQL > |
```

7. Conclusion

The Bus Booking Reservation System has been successfully developed and implemented, achieving its primary goal of creating an efficient, user-friendly platform for managing bus reservations. By utilizing Angular for the front-end, Spring Boot for the back-end, and MySQL for the database, the system ensures that all users—whether administrators or passengers—can interact with a seamless, reliable, and dynamic application. The system's architecture is built to be robust and scalable, ensuring that it can handle increasing demand and traffic as the user base grows.

Key features, such as admin functionalities for bus and seat management, ticket booking, and cancellation, as well as user-friendly interfaces for bus search and ticketing, have been successfully implemented. The system allows administrators to easily manage buses, pricing, and other resources, while providing users with the ability to search for buses, book tickets, retrieve booking information, and cancel reservations when necessary. The efficient back-end integration ensures smooth data flow, enhancing both the user and administrator experience.

The project stands as a solid foundation for a bus reservation platform, with the potential for future improvements and added functionality. Through careful planning, development, and implementation, the system meets the immediate needs of bus reservation management and lays the groundwork for future growth and scalability. It effectively bridges the gap between user expectations and administrative needs, making it a valuable tool for the bus transportation industry.

8. Future Enhancements

While the Bus Booking Reservation System currently meets the core requirements, several future enhancements could further improve the system's capabilities and user experience. The following are some potential upgrades and features for future development:

1. **Integration with Payment Gateways for Online Transactions**: To make the booking process more convenient, integrating with popular payment gateways such as PayPal, Stripe, or credit card processors would allow users to make secure and hassle-free payments for their tickets directly within the system. This would streamline the booking process, eliminate the need for offline payments, and provide a more seamless experience for users.
2. **Real-Time Tracking of Bus Locations**: Integrating GPS tracking into the system would allow users to track the location of buses in real-time. This would provide passengers with more accurate information on bus arrival times and any delays, making it easier for them to plan their journeys. Additionally, real-time tracking could be useful for administrators to monitor bus operations and optimize route planning.
3. **Push Notifications for Ticket Updates and Reminders**: By integrating push notifications, the system could send timely updates to users regarding their booking status, reminders for upcoming trips, and alerts about cancellations or delays. This would enhance the user experience by keeping passengers informed in real-time, improving engagement, and reducing the chances of no-shows.

4. **Mobile Application Development**: Developing a mobile app for the system would provide users with an even more accessible and convenient way to book tickets, manage reservations, and track buses. A mobile app could offer additional features, such as offline access to booking details, a simplified interface for faster interactions, and better overall usability.
5. **AI-Powered Recommendations and Personalization**: Implementing artificial intelligence and machine learning algorithms could allow the system to suggest personalized routes and buses based on user preferences and past booking history. This would not only improve user satisfaction but also provide a tailored experience for passengers, further enhancing the value of the system.
6. **Multi-Language and Multi-Currency Support**: Expanding the system to support multiple languages and currencies would open up the platform to a broader international audience, enabling users from different regions to interact with the system in their native languages and process payments in their local currencies.
7. **Enhanced Reporting and Analytics for Administrators**: Adding advanced analytics and reporting tools would allow administrators to gain valuable insights into bus operations, such as occupancy rates, revenue generation, and user demographics. These insights could be used to optimize bus routes, adjust pricing strategies, and enhance overall service delivery.
8. **Social Media Integration**: Enabling users to share their bookings, itineraries, or travel experiences on social media platforms could increase user engagement and visibility for the system. Social media integration could also facilitate easy sharing of promotions, offers, and updates, helping to attract new customers.