NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready workforce

Team Members

Student Name :DEEPA.K

Student ID :511321205005

KINGSTON ENGINEERING COLLEGE

VELLORE

CAPSTONE PROJECT SHOWCASE

Project Title

Building Bus Reservation System using Python and Django

Abstract | Problem Statement | Project Overview | Proposed Solution | Technology Used | Modelling & Results | Conclusion

Abstract

1. Project Setup:
•Install Django and set up a new project.
•Create a virtual environment to manage dependencies.
2. Database Design:
•Identify the entities involved (e.g., buses, routes, bookings, passengers).
•Design the database schema using Django's ORM (Object-Relational Mapping).
•Migrate the schema to the database.
3. Models:
•Define Django models corresponding to the database schema.
•Implement relationships between models (e.g., ForeignKey, ManyToManyField).
4. Views:
•Create views to handle user interactions and HTTP requests.
•Define views for displaying available buses, booking tickets, viewing bookings, etc.
•Implement CRUD (Create, Read, Update, Delete) operations for relevant models.
5. Templates:
•Design HTML templates for user interfaces.
•Integrate Django's template engine for dynamic content rendering.
•Ensure responsive design for various devices.

Source :

6. Forms:
- Build forms for user input validation and data submission.
- Utilize Django's form handling features to simplify form creation and processing.

7. Authentication and Authorization:
- Implement user authentication using Django's built-in authentication system or third-party libraries like Django-Allauth.
- Define user roles and permissions (e.g., admin, regular user).

8. Business Logic:
- Implement business rules and validations.
- Handle seat availability, fare calculation, booking confirmation, etc.

9. Integration:
- Integrate with payment gateways for handling transactions.
- Implement APIs for external services (e.g., SMS notifications, email confirmations).

10. Testing:
- Write unit tests to ensure code quality and functionality.
- Perform integration testing to check interactions between components.

11. Deployment:
- Deploy the application to a server (e.g., AWS, Heroku, DigitalOcean).
- Set up necessary configurations for production environment.

12. Monitoring and Maintenance:
- Implement logging for tracking errors and debugging.
- Monitor application performance and user feedback for continuous improvement.

Problem Statement

The aim of this project is to develop a web-based bus reservation system using Python and Django framework. The system should facilitate users to search for available buses, book tickets, and manage their bookings efficiently. The application should provide a seamless experience for both passengers and administrators.

Features to Implement:

User Registration and Authentication:

Users should be able to register for an account and log in securely.

Authentication mechanisms should be implemented to ensure secure access to user accounts.

Bus Management:

Admins should be able to add, update, and delete bus routes.

Each bus route should have details such as origin, destination, departure time, arrival time, fare, and available seats.

Booking Management:

Users should be able to search for available buses based on the origin, destination, and date of travel.

Upon selecting a bus, users should be able to book tickets by specifying the number of seats required.

Users should receive confirmation emails or SMS upon successful booking.

Admins should have access to view and manage bookings, including canceling bookings if necessary.

Seat Selection:

Users should be able to select seats from the available ones while booking tickets.

Seat availability should be updated in real-time to avoid overbooking.

Source :

1.Payment Integration:

- Integration with a payment gateway to facilitate secure online payments for booking tickets.
- Support for multiple payment methods such as credit/debit cards, net banking, and digital wallets.

2.User Profile:

- Users should have a profile section where they can view their booking history, update personal information, and change passwords.

3.Admin Dashboard:

- Admins should have access to a dashboard to manage buses, bookings, and user accounts.
- The dashboard should provide insights into booking trends, revenue, and other key metrics.

4.Email/SMS Notifications:

- Automated notifications should be sent to users regarding booking confirmations, ticket cancellations, and updates on their journey.

Technical Requirements:

- Python and Django framework for backend development.
- HTML, CSS, and JavaScript for frontend development.
- PostgreSQL or MySQL database for storing application data.
- Integration with third-party APIs for payment processing and notifications.
- Implementation of RESTful APIs for communication between frontend and backend components.

Deliverables:

- Fully functional web application deployed on a web server.
- Source code with proper documentation and comments for future maintenance.
- User manual or documentation explaining how to use the application.

Constraints:

- The project should be completed within a specified timeline and budget.

Project Overview

The Bus Reservation System is a web-based application developed using Python and Django framework. It aims to provide a convenient platform for users to search for available buses, book tickets, and manage their bookings seamlessly. The system caters to both passengers and administrators, offering features for efficient bus management and booking management.

Source :

Proposed Solution

Setup Django Project: Start by creating a new Django project using `django-admin startproject project_name`.
1.Create Django App: Use `python manage.py startapp app_name` to create a new Django app within your project.
This app will contain the code for your bus reservation system.

Define Models: Design your database models to represent

Create Views: Define views to handle user interactions such as viewing available buses, reserving seats, etc.
2.Implement Forms: Create Django forms for user input, such as passenger information and seat selection.
3.Set Up URLs: Configure URL patterns to map views to specific URLs within your Django app.
4.Implement Logic: Write the necessary business logic to handle reservations, seat availability, etc.
5.Templates: Create HTML templates to render the views and forms to the users.

Source :

Technology Used

Front-end

Back-end

Source :

Modelling & Results

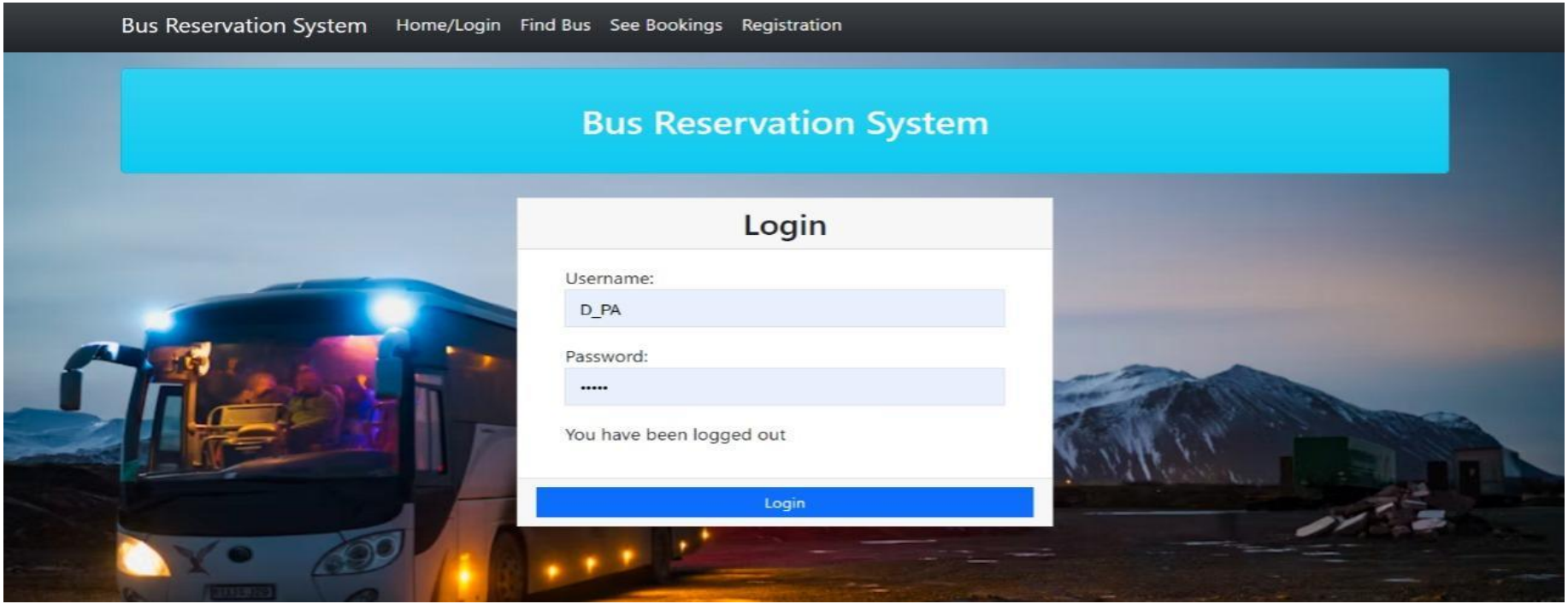
Set Up Django Project: Create a new Django project using the command django-admin startproject projectname

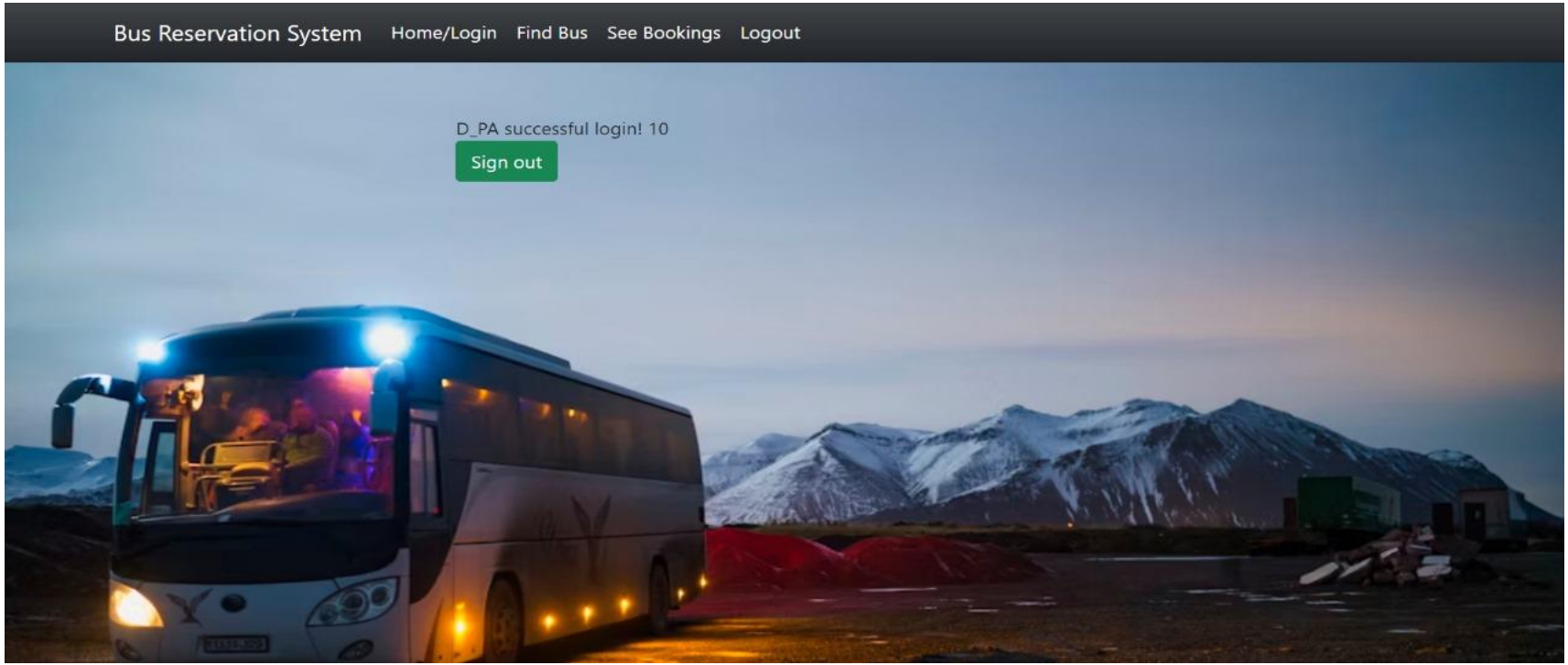
Create Django App: Inside the project directory, create a new Django app using python manage.py startapp appname.Define Models: Define models for buses, passengers, reservations, and any other necessary entitie

Set Up Admin Panel: Register the models in the Django admin panel for easy management

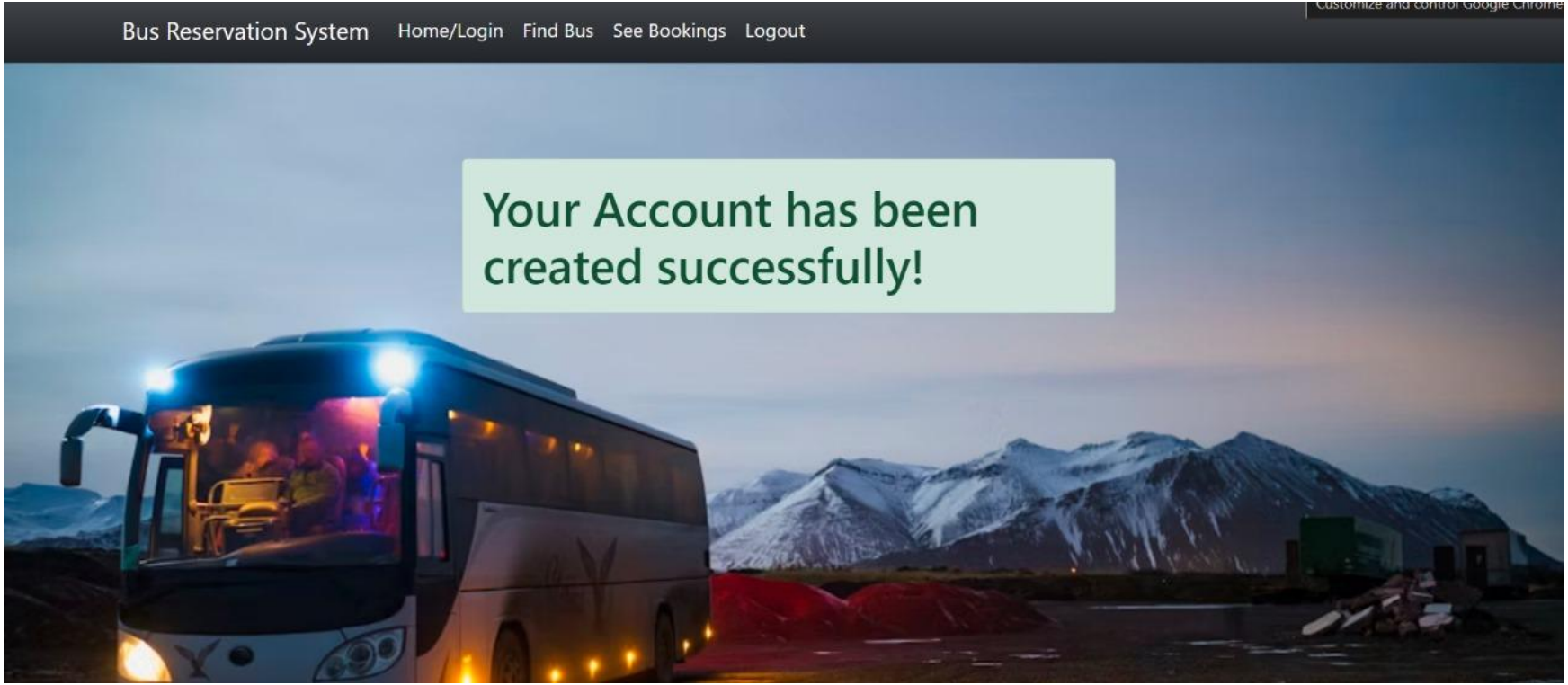
Source :

Homepage

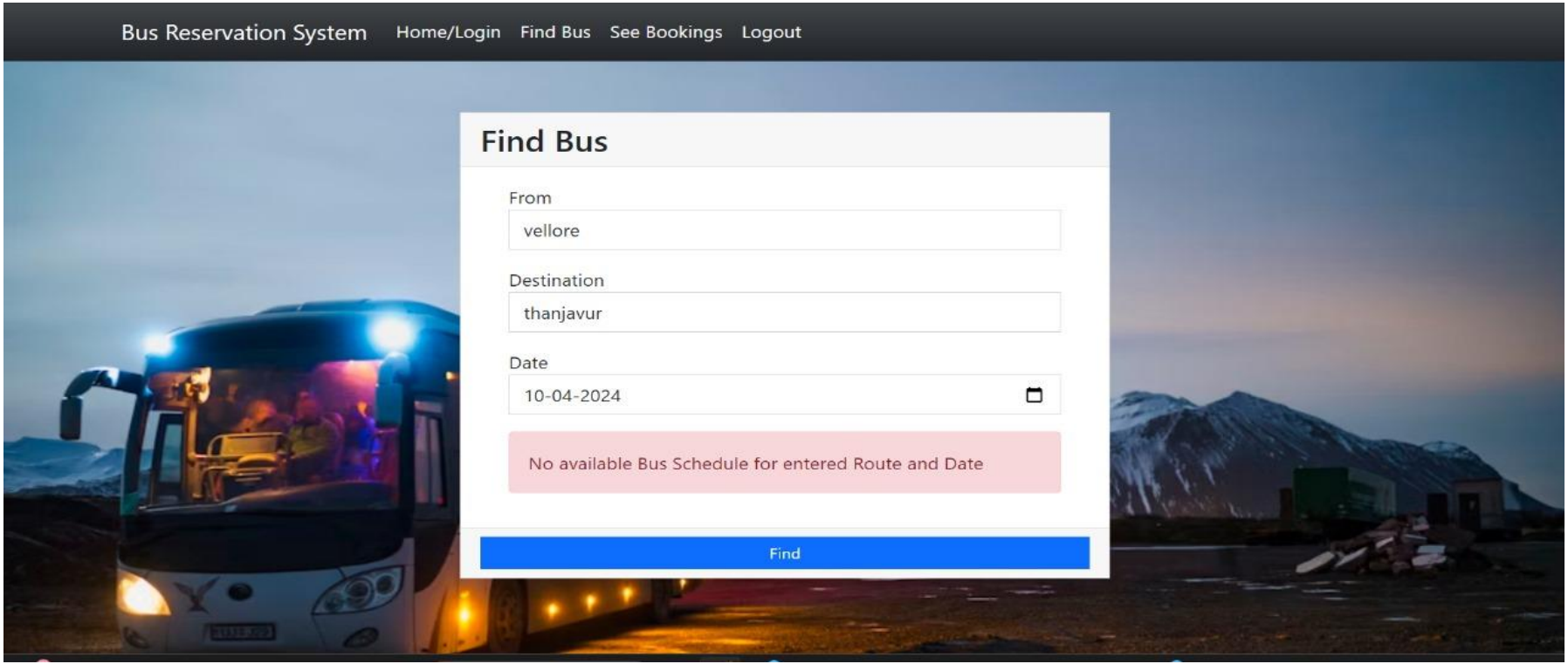

Bus Reservation System
Home/Login
Find Bus
See Bookings
Registration
Bus Reservation System
Login
Username:
D_PA
Password:
You have been logged out
Login


Bus Reservation System
Home/Login
Find Bus
See Bookings
Logout
D_PA successful login! 10
Sign out

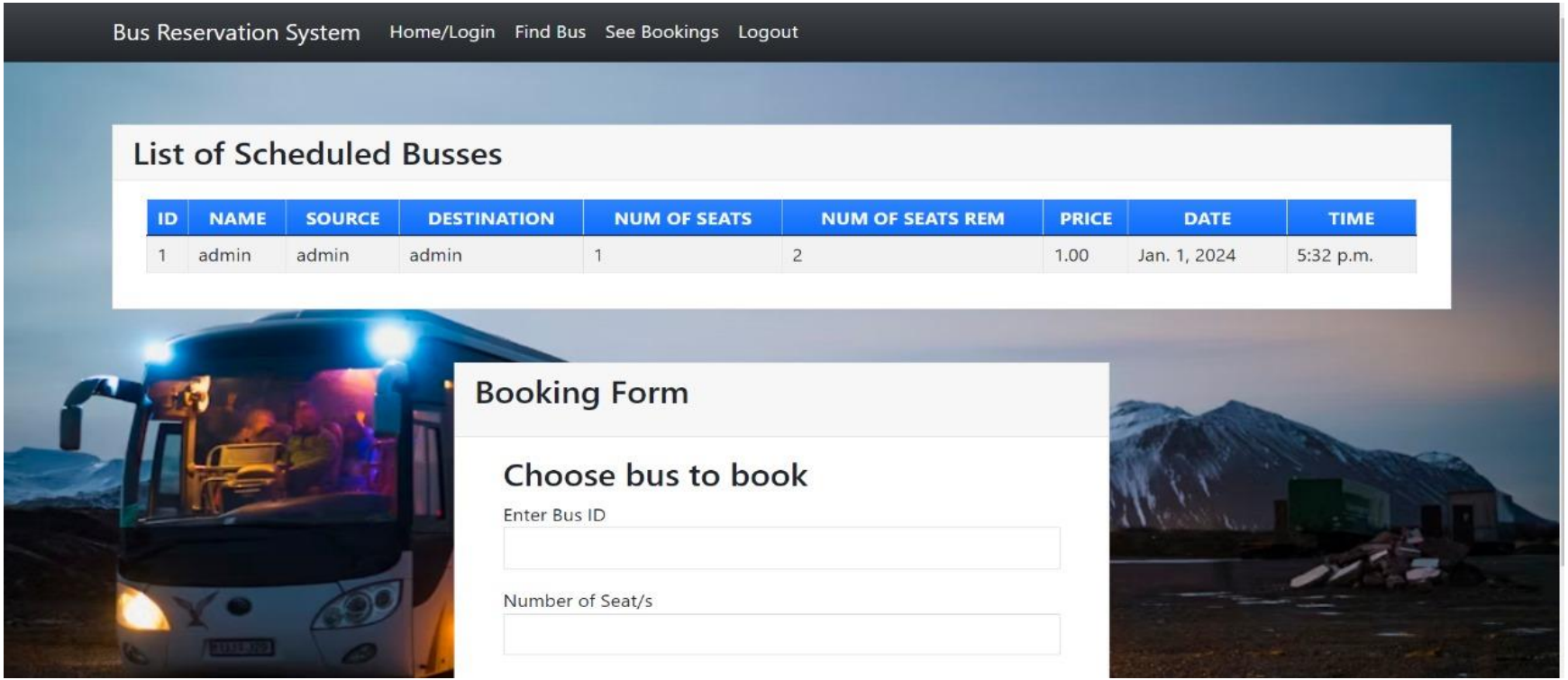
Account created page



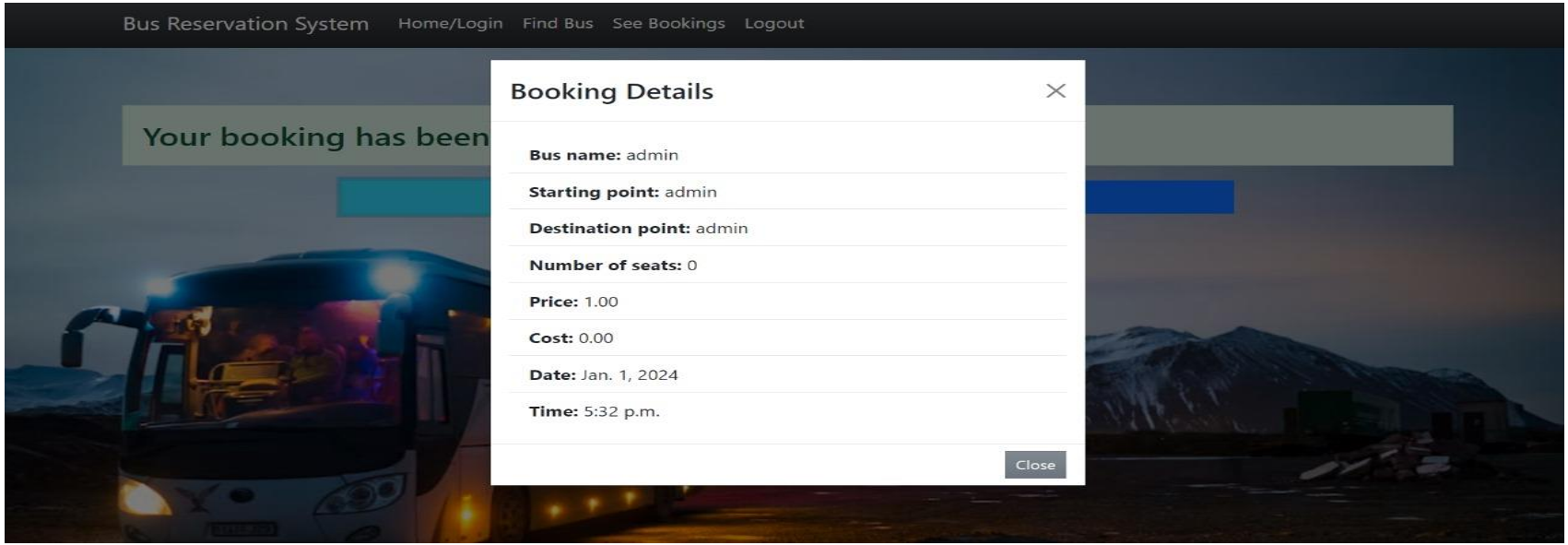
FIND BUS


Bus Reservation System
Home/Login
Find Bus
See Bookings
Logout
Find Bus
From
vellore
Destination
thanjavur
Date
10-04-2024
No available Bus Schedule for entered Route and Date
Find

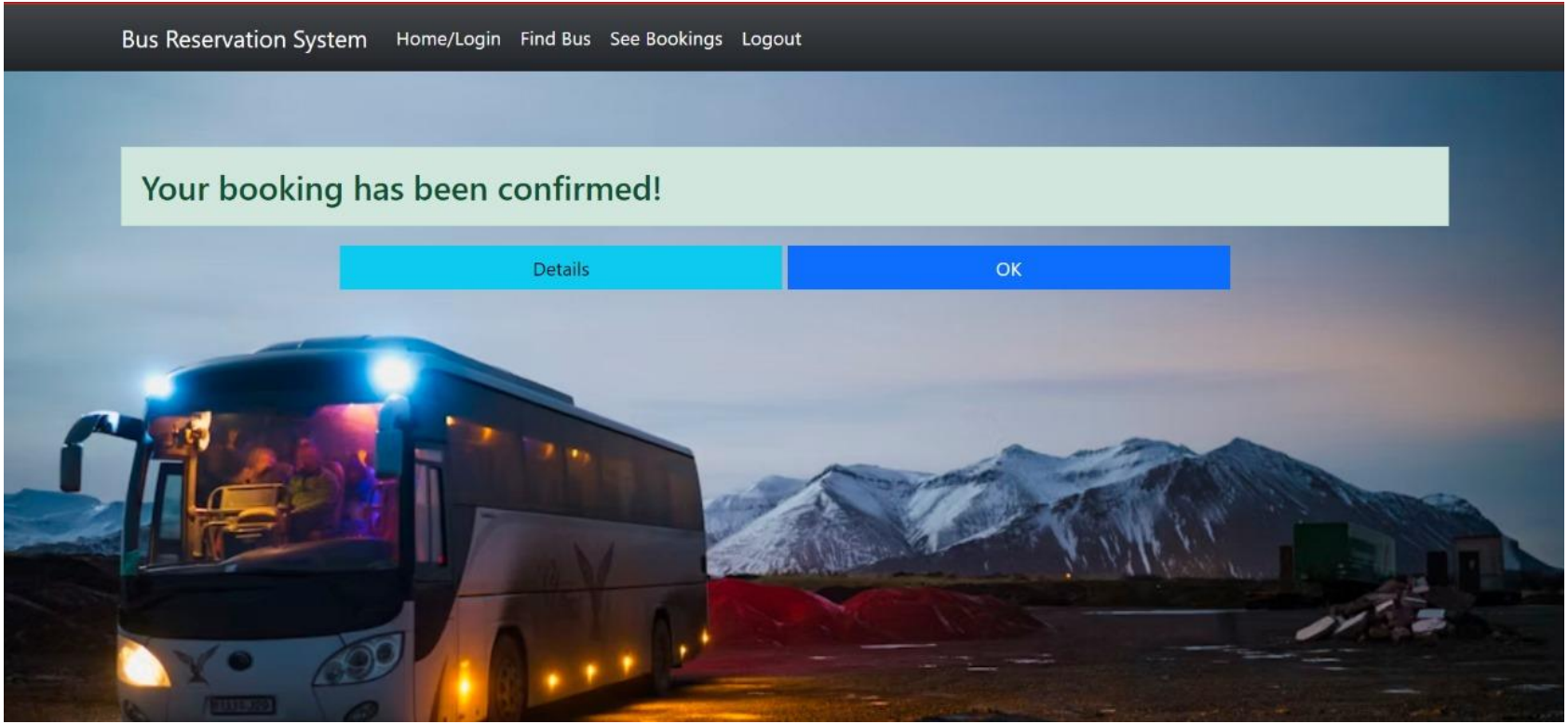
LIST O SCHEDULED BUSES



BOOKING DETAILS



BOOKING STATUS



Conclusion

The bus reservation system developed using Python and Django offers a comprehensive solution for managing bus bookings and reservations. By leveraging Django's powerful features such as models, views, templates, and the admin panel, we were able to create a user-friendly and efficient application.

Overall, the bus reservation system streamlines the booking process, enhances user experience, and improves the efficiency of bus management operations. With further enhancements and optimizations, the system can be tailored to meet the specific requirements of bus companies

Source :

Thank You!