

Bus Booking Application - Project Documentation

1. Introduction

This Bus Booking Application is a web-based platform designed to streamline the process of booking bus tickets online. It supports two major user roles:

- **USER:** Can register, log in, browse bus schedules, book seats, and view their booking history.
- **ADMIN:** Has access to administrative features like bus management, booking overview, and user management.

Technologies Used

- **Frontend:** (Assumed Web Interface)
- **Backend:** Node.js (assumed)
- **Database:** MongoDB

The MongoDB database is used to store all essential records, including:

- Registered users
- Available buses
- Booking transactions

The screenshot shows the MongoDB Compass interface connected to the 'busbookingdb' database. The 'bookings' collection is selected, displaying six documents. The first document is expanded, showing its internal structure:

```
_id: ObjectId('682c186272fbab1d5d9bf1b')
paymentId : "pay_0X44UnByCCy/2"
orderId : "undefined"
receipt : "undefined"
paymentStatus : "Success"
userId : "68206bd119e3ef7616a827f3"
busId : "68206bd119e3ef7616a827f3"
bookingDate : 2025-05-21T18:30:00.000+00:00
totalAmount : 824
travelDate : 2025-05-21T18:30:00.000+00:00
email : "user@gmail.com"
passengers : Array (1)
  -class : "com.busBooking.Bus.model.Booking"
```

The second document is also partially visible below it.

MongoDB_Booking

MongoDB Compass - localhost/busbookingdb.buses

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (2)

Search connections

- localhost
 - admin
 - busbookingdb
 - bookings
 - buses
 - users
 - config
 - empdb
 - employeeedb
 - local
 - patientmedapp
 - productdb
 - task
- Test

localhost > busbookingdb > buses

Documents 11 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

25 1 - 11 of 11

[Explain](#) [Reset](#) [Find](#) [Options](#)

<code>_id: ObjectId('602b0d130e3e67616eb62f3')</code>	<code>operatorName: "PSR Travels"</code>
<code>from: "Amarapur"</code>	<code>to: "Kurnool"</code>
<code>departureTime: "13:00"</code>	<code>arrivalTime: "05:00"</code>
<code>seaterPrice: 501</code>	<code>sleeperPrice: 824</code>
<code>seaterCount: 28</code>	<code>sleeperCount: 15</code>
<code>totalSeats: 35</code>	<code>amenities: "WiFi,AC"</code>
<code>-class: "com.BusBooking.Bus.model.Bus"</code>	<code>-class: "com.BusBooking.Bus.model.Bus"</code>

<code>_id: ObjectId('602c13c2f2d0183c13a81f3c')</code>	<code>operatorName: "SRS Travels"</code>
<code>from: "Hyderabad"</code>	<code>to: "Chennai"</code>
<code>departureTime: "20:00"</code>	<code>arrivalTime: "06:00"</code>
<code>seaterPrice: 1201</code>	<code>sleeperPrice: 1421</code>
<code>seaterCount: 25</code>	<code>sleeperCount: 15</code>
<code>totalSeats: 40</code>	<code>amenities: "WiFi,AC,Blanket"</code>
<code>-class: "com.BusBooking.Bus.model.Bus"</code>	<code>-class: "com.BusBooking.Bus.model.Bus"</code>

MongoDB_Buses

MongoDB Compass - localhost/busbookingdb.users

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (2)

Search connections

- localhost
 - admin
 - busbookingdb
 - bookings
 - buses
 - users
 - config
 - empdb
 - employeeedb
 - local
 - patientmedapp
 - productdb
 - task
- Test

localhost > busbookingdb > users

Documents 4 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

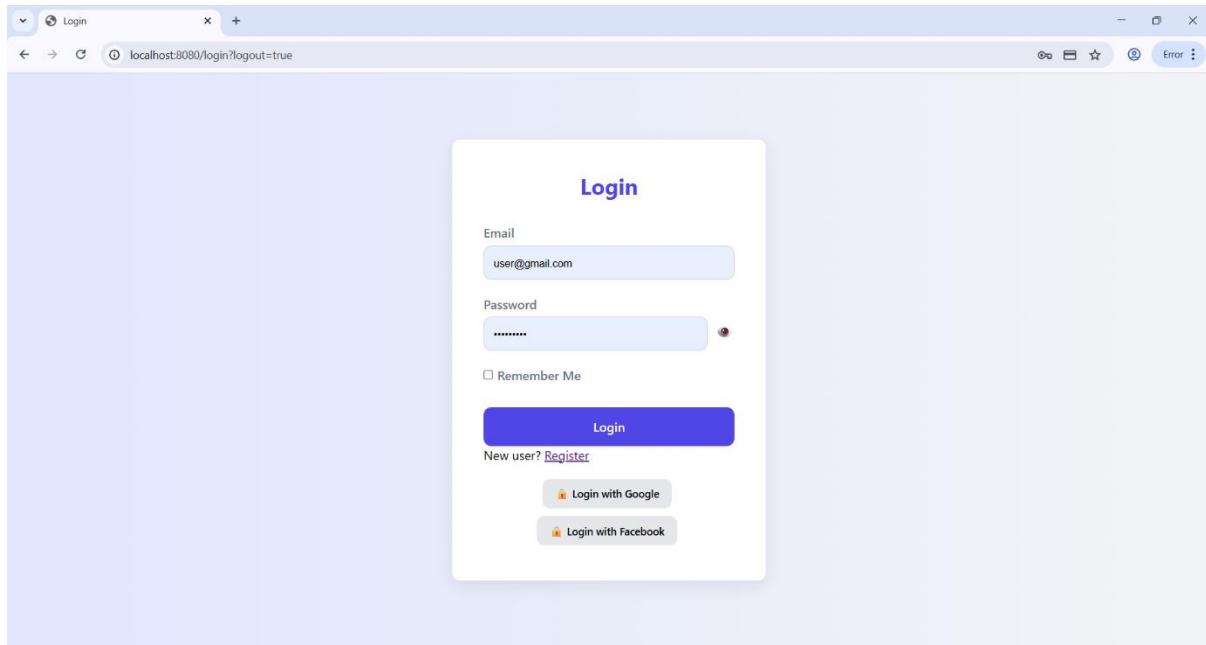
25 1 - 4 of 4

[Explain](#) [Reset](#) [Find](#) [Options](#)

<code>_class: "com.BusBooking.Bus.model.User"</code>	
<code>_id: ObjectId('602b2cce958609260b48563a')</code>	<code>name: "Test User"</code>
<code>email: "user@bus.com"</code>	<code>password: "52a10502053rhUCeGRlAnXQmbqHr35fz0rj2J0NEBx1bUH4mQ0VpFyC"</code>
<code>verified: false</code>	<code>role: "USER"</code>
<code>-class: "com.BusBooking.Bus.model.User"</code>	<code>-class: "com.BusBooking.Bus.model.User"</code>
<code>_id: ObjectId('602b2d5958609260b485645')</code>	<code>name: "user"</code>
<code>email: "user@gmail.com"</code>	<code>password: "52a10502053rhUCeGRlAnXQmbqHr35fz0rj2J0NEBx1bUH4mQ0VpFyC"</code>
<code>verified: false</code>	<code>role: "USER"</code>
<code>-class: "com.BusBooking.Bus.model.User"</code>	<code>-class: "com.BusBooking.Bus.model.User"</code>
<code>_id: ObjectId('602c125cf2d0183c13a81f3a')</code>	<code>name: "John"</code>
<code>email: "johnwill10@gmail.com"</code>	<code>password: "52a10502053rhUCeGRlAnXQmbqHr35fz0rj2J0NEBx1bUH4mQ0VpFyC"</code>
<code>verified: false</code>	<code>role: "USER"</code>
<code>-class: "com.BusBooking.Bus.model.User"</code>	<code>-class: "com.BusBooking.Bus.model.User"</code>

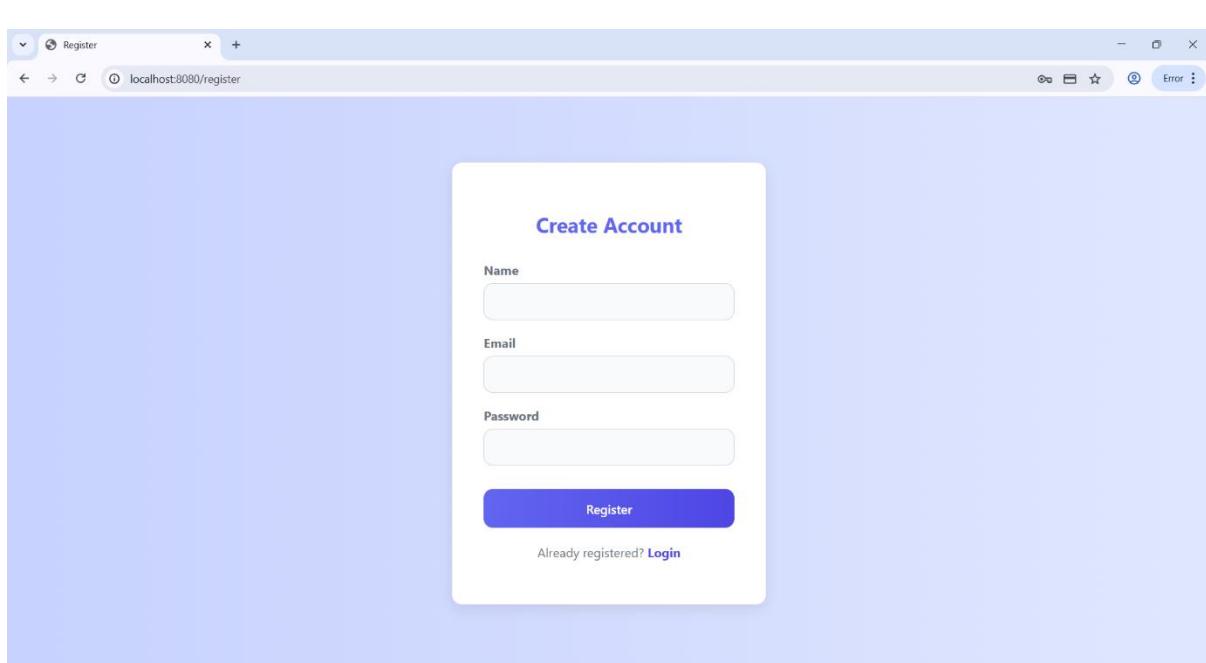
MongoDB_Users

2. Explanation



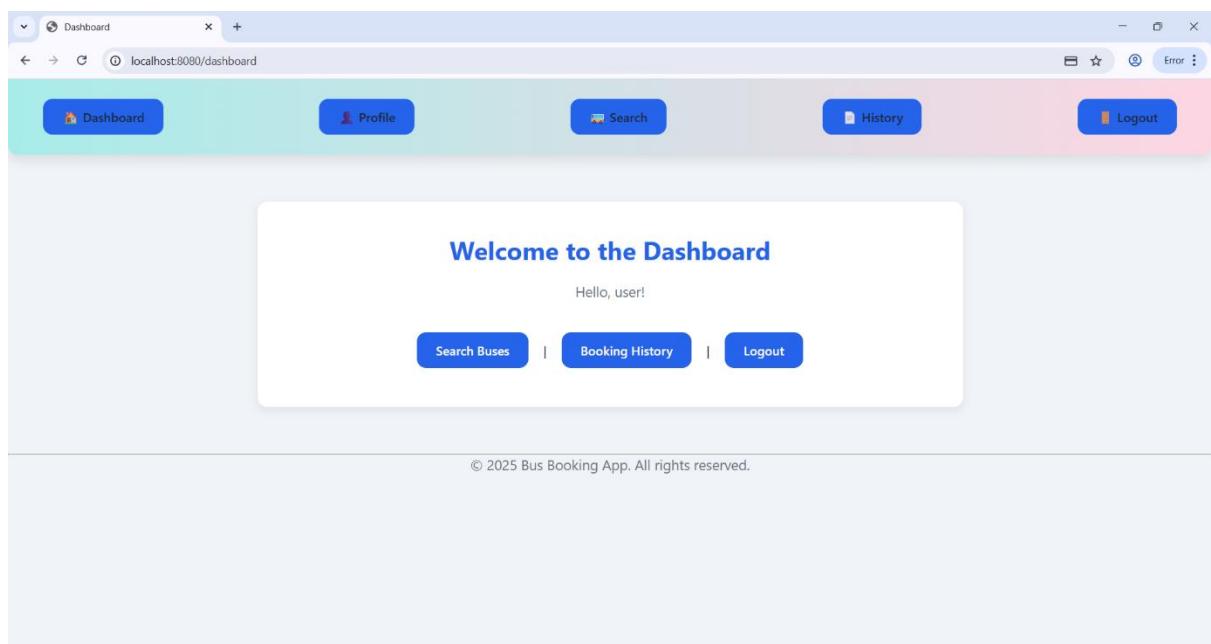
Login Page

This screen is the main entry point to the application. Users enter their registered credentials to authenticate and access their personalized dashboards. The form validates the inputs in real time and communicates with the backend to verify the credentials against the MongoDB database. In case of incorrect credentials, appropriate error messages guide users to correct them. This page also includes links to sign up or reset passwords. It plays a key role in ensuring secure access and session control.



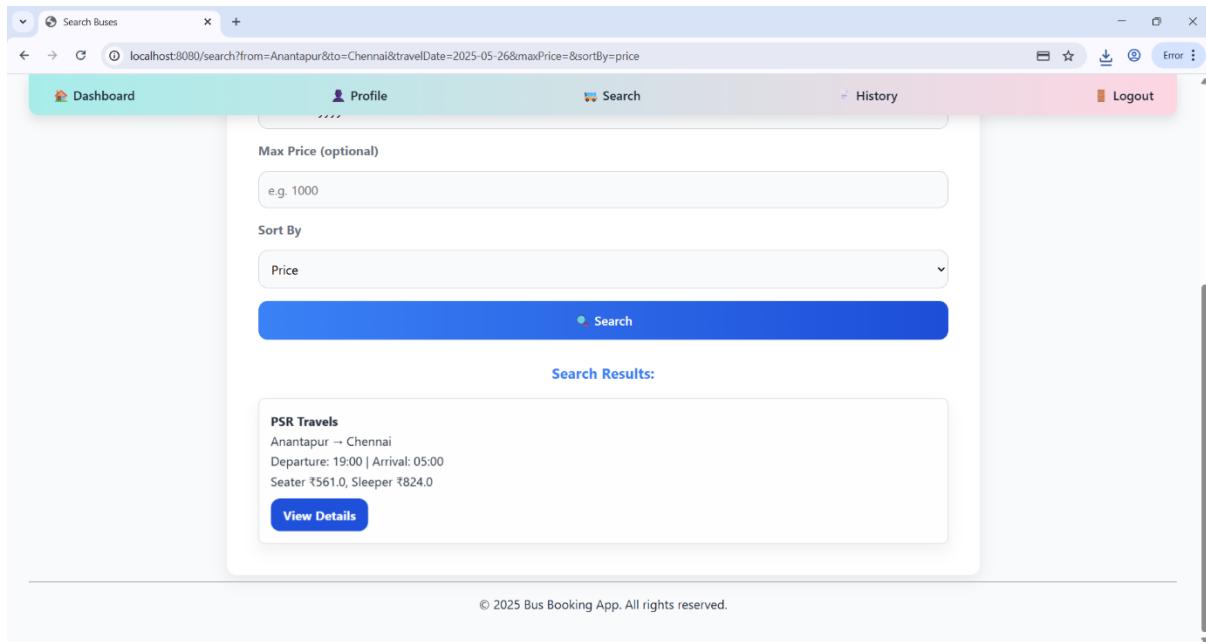
User Registration

This page allows new users to register with the system by providing personal details such as name, email, and password. The inputs are validated both on the client and server side to ensure accuracy and uniqueness. Upon successful registration, the user's data is stored securely in MongoDB. The form layout is designed to be intuitive, ensuring a smooth onboarding experience. It also prevents duplicate accounts and redirects users to the login page once registration is complete.



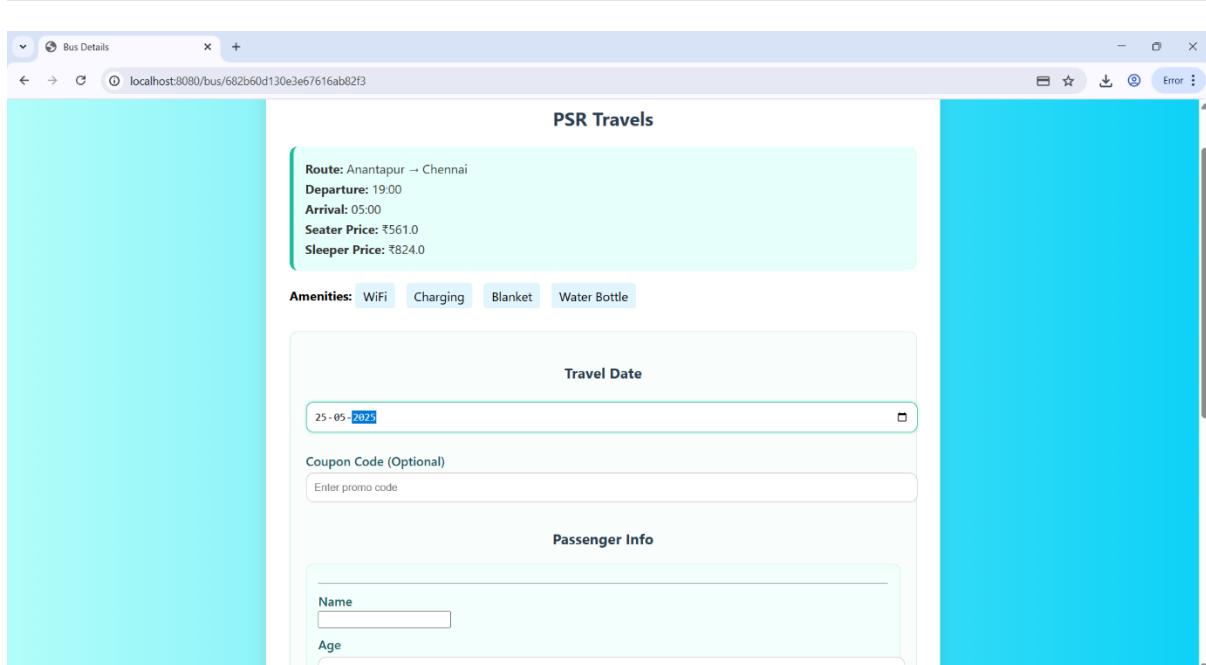
User Dashboard

After a successful login, the user is redirected to this dashboard. It provides quick access to key features like viewing buses, managing bookings, and updating profiles. The interface displays personalized data such as user name and active bookings. Navigation options are clearly marked and user-centric. The dashboard uses backend APIs to fetch relevant data from MongoDB and is designed for optimal responsiveness and usability.



Available Buses List

This screen displays a list of all available buses, filtered according to the user's search parameters like date, destination, and time. Each listing shows critical information including timing, available seats, and pricing. Clicking a listing opens detailed bus and seat information. This page dynamically interacts with MongoDB to fetch updated listings. It offers a real-time view of all bus options and helps users make informed decisions.



Bus Details Page

This screen provides detailed information about a selected bus. It displays departure/arrival time, bus number, route, and an interactive seat map. Users can view

and select available seats in real time. Backend logic ensures selected seats are locked temporarily to prevent double bookings. It's connected to MongoDB to track live availability and to confirm selected options before proceeding to booking.

This screenshot shows the 'Passenger Info' section of the booking interface. It includes fields for Name, Age, Gender (Male), and Seat Type (Seater). Below these, a 'Select Seat' grid displays 16 numbered boxes (1-16) arranged in four rows of four. Seats 1, 2, 3, and 4 are highlighted in purple, while others are white. At the bottom are buttons for '+ Add Passenger' and 'Pay & Book'.

This screenshot shows the same 'Passenger Info' section as above, but with different seat selection status. Seats 1, 2, 3, and 4 are now dark gray, indicating they are selected or locked. The other seats (5-16) are white. The rest of the interface remains the same.

Booking Interface

Once seats are selected, users arrive at this confirmation screen. It summarizes the trip details, selected seats, and total fare. The system re-validates all data before storing the booking. Users can confirm or go back to change selections. This screen is key in

ensuring transactional integrity, and all successful bookings are logged to MongoDB. It also provides helpful error messages or success alerts.

Online Bus Ticket

Booking ID: 682b31cc1c30dc050eb86352

Booking Date: Mon May 19 18:57:40 IST 2025

Email: null

Trip Details

Bus Operator: Kallada Travels

Route: Hyderabad Chennai

Departure Time: 8:00

Arrival Time: 21:00

Travel Date: Thu May 22 00:00:00 IST 2025

Total Paid: 1134.0

Total Seats Booked: 2

Passenger Details

- John | Age: 20 | Gender: Male | Seat: 9 (Sleeper)

- Vesli | Age: 25 | Gender: Male | Seat: 10 (Sleeper)

Male Passengers: 2, Female Passengers: 0

Boarding Instructions

Boarding Time: Please report 30 minutes before departure.

Boarding Point: Main Bus Terminal (Gate 3)

Helpline: +91-98765-43210

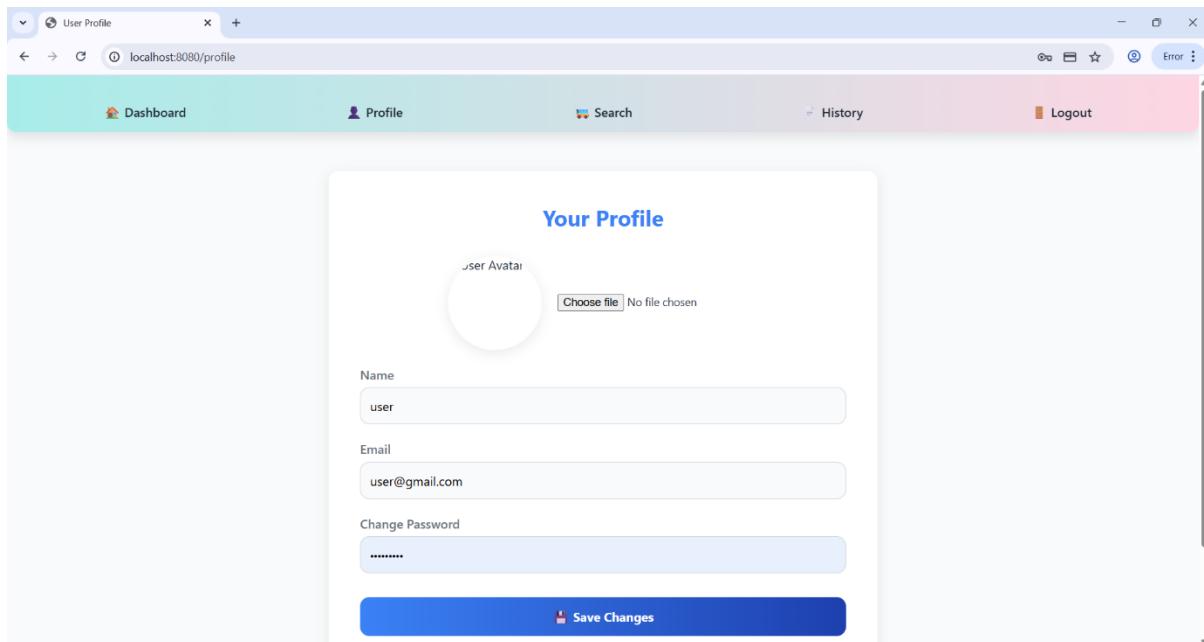
Scan this QR code for quick ticket info



Booking Summary

Here, users can view a detailed summary of their booking after confirmation. It

includes ticket ID, travel details, seat numbers, fare breakdown, and passenger information. A print or download option is typically available for generating the ticket. This screen reassures users their booking is confirmed and serves as a digital receipt. The details are fetched securely from MongoDB based on user session and booking ID.



User Profile Page

This page displays all user information including name, contact, and booking history. Users can update personal details, change passwords, or manage upcoming trips. The system fetches real-time data from MongoDB and allows secure updates. A logout option and navigation to the main dashboard are also provided. This helps users manage their data and preferences within the platform efficiently.

Admin Login Screen

This secure login screen is for administrative users only. It authenticates admin credentials using role-based access control. Unlike the regular user login, it unlocks the backend features like analytics and bus management. It includes security checks to ensure only authorized users gain access. Proper feedback is given in case of incorrect login attempts.

Admin Panel - Manage Buses

Total Buses: 11

Avg Seater Price: ₹

Avg Sleeper Price: ₹

Operator	From	To	Departure	Arrival	Seater ₹	Sleeper ₹	Seater Seats	Sleeper Seats	Total Seats	Status	Actions
Orange Travels	Hyderabad	Delhi	15:00	20:00	781.0	868.0	30	12	42	Active	
SRS Travels	Hyderabad	Delhi	9:00	20:00	391.0	715.0	28	13	41	Active	
Kallada Travels	Hyderabad	Chennai	8:00	21:00	626.0	567.0	27	14	41	Active	
Seabird Tourists	Bangalore	Mumbai	17:00	22:00	304.0	697.0	30	13	43	Active	

Admin Dashboard

The admin dashboard gives a comprehensive overview of system statistics such as total users, bookings, and buses in operation. From here, the admin can access booking records, user data, and performance analytics. This screen pulls summarized data from MongoDB collections and displays it in a user-friendly format. It is the control center for all backend activities and ensures administrative efficiency.

+ Add New Bus

Operator Name

From

To

Departure Time

-- : --

Arrival Time

-- : --

Seater Price

Sleeper Price

Seater Seats

Add/Update Bus Info

Admins can use this interface to add new buses or update existing ones. It includes

fields for route, schedule, seat count, and pricing. The form validates entries and submits them to MongoDB. This screen plays a vital role in ensuring accurate scheduling and capacity management. Admins can also manage recurring routes efficiently through this panel.

The screenshot shows a web-based admin interface titled "All Bookings (Admin View)". The top navigation bar includes links for Dashboard, Profile, Admin Panel, Bookings, and Logout. A filter dropdown at the top left says "Filter by Status: All". The main content area displays a table with columns: USER, TRAVEL DATE, PASSENGERS, TOTAL PAID, and STATUS. The table contains six rows of booking data:

USER	TRAVEL DATE	PASSENGERS	TOTAL PAID	STATUS
Test User user@bus.com	19-05-2025	• ⚡ Passenger 1 Age: 22 Gender: Male Seat: 5 (Seater) • ⚡ Passenger 2 Age: 23 Gender: Female Seat: 6 (Sleeper) • ⚡ Passenger 3 Age: 24 Gender: Male Seat: 7 (Seater)	₹1425.0	CONFIRMED ▾
user user@gmail.com	22-05-2025	• ⚡ John Age: 20 Gender: Male Seat: 9 (Sleeper) • ⚡ Vesli Age: 25 Gender: Male Seat: 10 (Sleeper)	₹1134.0	CONFIRMED ▾
user user@gmail.com	25-05-2025	• ⚡ Vesli Age: 20 Gender: Male Seat: 16 (Seater) • ⚡ John Age: 15 Gender: Male Seat: 17 (Seater)	₹1252.0	CONFIRMED ▾
user user@gmail.com	25-05-2025		₹0.0	CONFIRMED ▾
user user@gmail.com	22-05-2025	• ⚡ John Age: 25 Gender: Male Seat: 15 (Sleeper)	₹824.0	CONFIRMED ▾
John johnvesli10@gmail.com	25-05-2025	• ⚡ John Age: 21 Gender: Male Seat: 3 (Sleeper) • ⚡ Vesli Age: 28 Gender: Male Seat: 4 (Sleeper)	₹1648.0	CONFIRMED ▾

Bookings List (Admin View)

This admin screen displays a table of all user bookings. Each record includes user info, selected bus, travel date, and status. Admins can update status, issue refunds, or make changes as needed. It connects directly with MongoDB to reflect live data. This interface helps in customer support and operational monitoring.

The screenshot shows a web-based admin dashboard titled "Admin Dashboard" at the URL "localhost:8090/admin/dashboard". The main content area is titled "Dashboard" and contains a table with the following data:

Operator	From	To	Departure	Arrival	Seater ₹	Sleeper ₹	Seater Seats	Sleeper Seats	Total Seats	Status	Actions
Orange Travels	Hyderabad	Delhi	15:00	20:00	781.0	868.0	30	12	42	Active	Edit Delete
SRS Travels	Hyderabad	Delhi	9:00	20:00	391.0	715.0	28	13	41	Active	Edit Delete
Kallada Travels	Hyderabad	Chennai	8:00	21:00	626.0	567.0	27	14	41	Active	Edit Delete
Seabird Tourists	Bangalore	Mumbai	17:00	22:00	304.0	697.0	30	13	43	Active	Edit Delete
APSRTC - Amaravati/Indra	Hyderabad	Chennai	10:00	22:00	301.0	783.0	23	15	38	Active	Edit Delete
National Travels	Chennai	Mumbai	14:00	18:00	557.0	562.0	20	11	31	Active	Edit Delete
Intricity SmartBus	Hyderabad	Mumbai	16:00	23:00	680.0	970.0	22	13	35	Active	Edit Delete
GreenLine Travels	Chennai	Bangalore	17:00	21:00	553.0	1038.0	24	14	38	Active	Edit Delete

Bus List Management

This screen allows admins to view, edit, or delete bus listings. Each row shows a bus's unique ID, route, and status. Buttons for editing or removal ensure that outdated or canceled buses can be easily managed. MongoDB updates occur instantly based on admin inputs. It keeps the available bus listings clean and up-to-date.

User Profiles (Admin)

Admin users can view individual user profiles, including personal details and booking history. This screen helps in resolving user issues and managing account-level access. It serves as a monitoring tool for ensuring proper user behavior and account management. MongoDB acts as the backend source for all user-related data shown here.

All Bookings - Admin

localhost:8080/admin/bookings

All Bookings (Admin View)

Filter by Status: All

User	Travel Date	Passengers	Total Paid	Status
Test User user@bus.com	19-05-2025	• ⚡ Passenger 1 Age: 22 Gender: Male Seat: 5 (Seater) • ⚡ Passenger 2 Age: 23 Gender: Female Seat: 6 (Sleeper) • ⚡ Passenger 3 Age: 24 Gender: Male Seat: 7 (Seater)	₹1425.0	CONFIRMED
user user@gmail.com	22-05-2025	• ⚡ John Age: 20 Gender: Male Seat: 9 (Sleeper) • ⚡ Vesli Age: 25 Gender: Male Seat: 10 (Sleeper)	₹1134.0	CONFIRMED
user user@gmail.com	25-05-2025	• ⚡ Vesli Age: 20 Gender: Male Seat: 16 (Seater) • ⚡ John Age: 15 Gender: Male Seat: 17 (Seater)	₹1252.0	CONFIRMED
user user@gmail.com	25-05-2025		₹0.0	CONFIRMED
user user@gmail.com	22-05-2025	• ⚡ John Age: 25 Gender: Male Seat: 15 (Sleeper)	₹824.0	CONFIRMED
John johnvesli10@gmail.com	25-05-2025	• ⚡ John Age: 21 Gender: Male Seat: 3 (Sleeper) • ⚡ Vesli Age: 28 Gender: Male Seat: 4 (Sleeper)	₹1648.0	CONFIRMED

Booking Status Controls

This screen allows admins to approve, reject, or mark bookings as completed. Each entry includes status toggles and audit logs. This control helps in managing exceptions like seat cancellations or no-shows. Admins can ensure that real-world operations are synchronized with database records. MongoDB is updated accordingly to maintain accurate status.

Admin Dashboard

localhost:8080/admin/dashboard

Operator

Operator	From	To	Departure	Arrival	Seater ₹	Sleeper ₹	Seater Seats	Sleeper Seats	Total Seats	Status	Actions
Orange Travels	Hyderabad	Delhi	15:00	20:00	781.0	868.0	30	12	42	<input checked="" type="checkbox"/> Active	
SRS Travels	Hyderabad	Delhi	9:00	20:00	391.0	715.0	28	13	41	<input checked="" type="checkbox"/> Active	
Kallada Travels	Hyderabad	Chennai	8:00	21:00	626.0	567.0	27	14	41	<input checked="" type="checkbox"/> Active	
Seabird Tourists	Bangalore	Mumbai	17:00	22:00	304.0	697.0	30	13	43	<input checked="" type="checkbox"/> Active	
APSRTC - Amaravati/Indra	Hyderabad	Chennai	10:00	22:00	301.0	783.0	23	15	38	<input checked="" type="checkbox"/> Active	
National Travels	Chennai	Mumbai	14:00	18:00	557.0	562.0	20	11	31	<input checked="" type="checkbox"/> Active	
Inticity SmartBus	Hyderabad	Mumbai	16:00	23:00	680.0	970.0	22	13	35	<input checked="" type="checkbox"/> Active	
GreenLine Travels	Chennai	Bangalore	17:00	21:00	553.0	1038.0	24	14	38	<input checked="" type="checkbox"/> Active	

Analytics Page

Provides graphs and tables showing booking trends, peak usage times, and route popularity. This dashboard helps admins make data-driven decisions. The backend

aggregates records from MongoDB collections and displays summaries using charts. It's essential for performance monitoring and business forecasting.

Operator	From	To	Departure	Arrival	Seater ₹	Sleeper ₹	Seater Seats	Sleeper Seats	Total Seats	Status	Actions
Orange Travels	Hyderabad	Delhi	15:00	20:00	781.0	868.0	30	12	42	Active	
SRS Travels	Hyderabad	Delhi	9:00	20:00	391.0	715.0	28	13	41	Active	
Kallada Travels	Hyderabad	Chennai	8:00	21:00	626.0	567.0	27	14	41	Active	
Seabird Tourists	Bangalore	Mumbai	17:00	22:00	304.0	697.0	30	13	43	Active	
APSRTC - Amaravati/Indra	Hyderabad	Chennai	10:00	22:00	301.0	783.0	23	15	38	Active	
National Travels	Chennai	Mumbai	14:00	18:00	557.0	562.0	20	11	31	Active	
Inticity SmartBus	Hyderabad	Mumbai	16:00	23:00	680.0	970.0	22	13	35	Active	
GreenLine Travels	Chennai	Bangalore	17:00	21:00	553.0	1038.0	24	14	38	Active	

Settings Panel

Admins can configure system-wide settings like roles, notifications, and access rights. This screen is key to managing internal rules and configurations. It ensures that the application remains flexible and secure. All changes are saved in MongoDB and applied immediately.

Feedback & Support

Users can submit feedback or request support using this screen. It includes a message box, subject line, and email autofill. Feedback is stored in MongoDB and may be reviewed by support staff. It improves user satisfaction and helps with issue resolution.

3. Summary

The Bus Booking Application offers an efficient solution for both travellers and bus operators. It provides a robust system for searching, booking, and managing trips. Admin features allow oversight and control over operations, while MongoDB ensures fast and scalable data handling.

4. Future Improvements

-  Send SMS/email notifications for booking updates
-  Add real-time bus tracking and GPS support
-  Improve responsive design for mobile users
-  Enhance admin dashboards with predictive analytics
-  Include OTP-based login and stronger data encryption