

# Shuttle Ride

by Vivek Tangudu

## Scope of the project

- **Users**
  - Book a bus seat
  - View Seat plan of the bus along with ETA
  - Cancel booking
  - Register in the website
- Admin
  - Add a bus
  - Modify bus

## Tech stack

- **Frontend:** React.js
- **Backend:** Springboot
- **Database:** Mysql ( Used an instance of AWS)

## Session management

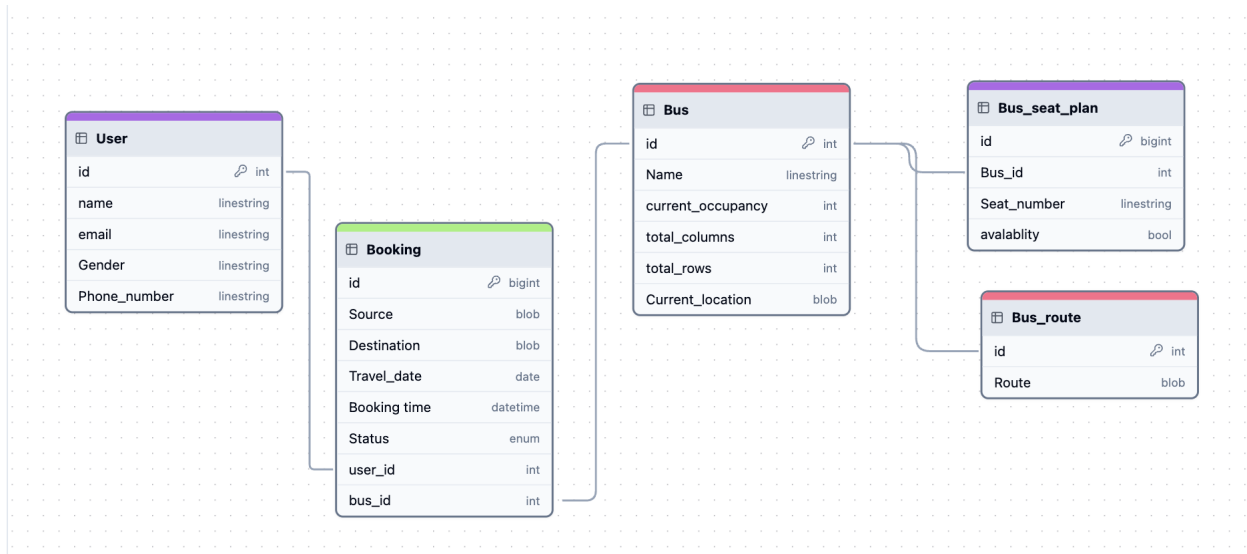
- We generate a JWT token after successful login and pass it to the frontend for storage in the webpage. For subsequent operations, we send this token back to the backend for verification. The JWT also serves as the user identifier—by decoding it, we can retrieve the user\_id. If the JWT has expired or is malfunctioning, authorization for any operations is denied. (Note: This process doesn't work in the Terminal.)

## Search

- Search for buses works using a pair of integers (representing longitude and latitude). We added stops using pairs of integers as well. To search for a

journey from one place to another, we use these integer pairs. (We need to improve this to find buses within a particular radius.)

DB design.



## API's List

- **Login**

- **Desc:** When a user wants to login, He uses this api to send his mobile number.
- **Endpoint:** `/auth/send_otp`
- **Method:** POST
- **Request Body:** Details of the user (mobile number).
- **Response:** Confirmation message or error response. (can be message where mobile number can't be found in the DB). For this I have twillio api to send an sms. (works for verified numbers).

- **Verify otp**
  - **Desc:** Verify his OTP
  - **Endpoint:** `/auth/verify_otp`
  - **Method:** POST
  - **Request Body:** mobile number, OTP.
  - **Response:** Confirmation message or error response. along with JWT token.
- **Verify JWT**
  - **Desc:** When a users opens home page, we verify his JWT token.
  - **Endpoint:** `/auth/jwt`
  - **Method:** POST
  - **Request Body:** JWT
  - **Response:** valid token or not. if it is invalid token, User moves to login page.
- **Register**
  - **Desc:** User wants to register himself.
  - **Endpoint:** `/users/create`
  - **Method:** POST
  - **Request Body:** mobileNumber, gender, email, Name
  - **Response:** True or false. And then he is moved to login page. While registering same email or phone\_number can't be used again. Sends an error message if you use it again.
- **getUserBookings**
  - **Desc:** User wants to have all his past bookings made by him.
  - **Endpoint:** `/users/bookings`

- **Method:** POST
- **Request Body:** JWT
- **Response:** List of all the bookings based on the user id.
- **book\_seat**
  - **Desc:** User selected a seat and wants to book the seat.
  - **Endpoint:** `/users/seat-booking`
  - **Method:** POST
  - **Request Body:** JWT, seat, source\_latitude, source\_longitude, destination\_latitude, destination\_longitude
  - **Response:** True or false. True means booking was confirm and False means failed to book the seat. If two people wants to book same seat at the same time, Only one can get the seats in the database seat changes to unavailable . So the other person can't.
- **Search\_buses**
  - **Desc:** User wants to search for buses between some source and destination.
  - **Endpoint:** `/users/search`
  - **Method:** POST
  - **Request Body:** JWT, seat, source\_latitude, source\_longitude, destination\_latitude, destination\_longitude
  - **Response:** True or false. True means booking was confirm and False means failed to book the seat.
- **Cancel\_booking**
  - **Desc:** User wants to cancel a booking.
  - **Endpoint:** `/users/cancel-booking`
  - **Method:** POST
  - **Request Body:** JWT, bookingId

- **Response:** True or false. True means the operation was Successful and false means failed to do.

**Interactions by the admin can be only done through terminal whereas the User can interaction with both Web and Terminal.**

## **User flow in terminal.**

```
Welcome to the Bus Booking System.  
Are you a user or an admin? (Enter 1 for User or 2 for Admin): 1  
Welcome, User! What would you like to do?  
1. Register  
2. Login  
7. Exit  
.
```

2 for login. After entering my mobile number. I received an OTP.

To: +91 57575701

Text Message • SMS  
Today, 1:12 PM

Your (SAMPLE TEST) verification code is: 855599

Today, 2:16 PM

Your (SAMPLE TEST) verification code is: 453899

Today, 4:16 PM

Your (SAMPLE TEST) verification code is: 839504

```
Enter your phone number (10 digits):
9441354555
Hibernate: select u1_0.user_id from user u1_0 where u1_0.phone_number=? limit ?
Hibernate: select u1_0.user_id from user u1_0 where u1_0.phone_number=? limit ?
pending
Enter OTP: 839504
Authentication Success!!
Hibernate: select u1_0.user_id,u1_0.email,u1_0.gender,u1_0.name,u1_0.phone_number from user u1_0 where u1_0.phone_number=?
Welcome, User! What would you like to do?
3. Book a bus seat
4. View Bus seat plan
5. Cancel a booking
6. Logout
7. Exit
```

```
3
Hibernate: select u1_0.user_id,u1_0.email,u1_0.gender,u1_0.name,u1_0.phone_number from user u1_0 where u1_0.user_id=?
Enter longitude for Source:
0
Enter latitude for Source:
0
Enter longitude for Destination:
10
Enter latitude for Destination:
0
```

And then List of busses are shown

```
Bus name: Zero,Id: 2,Occupancy Rate:Green
```

Now you can view the seat plan

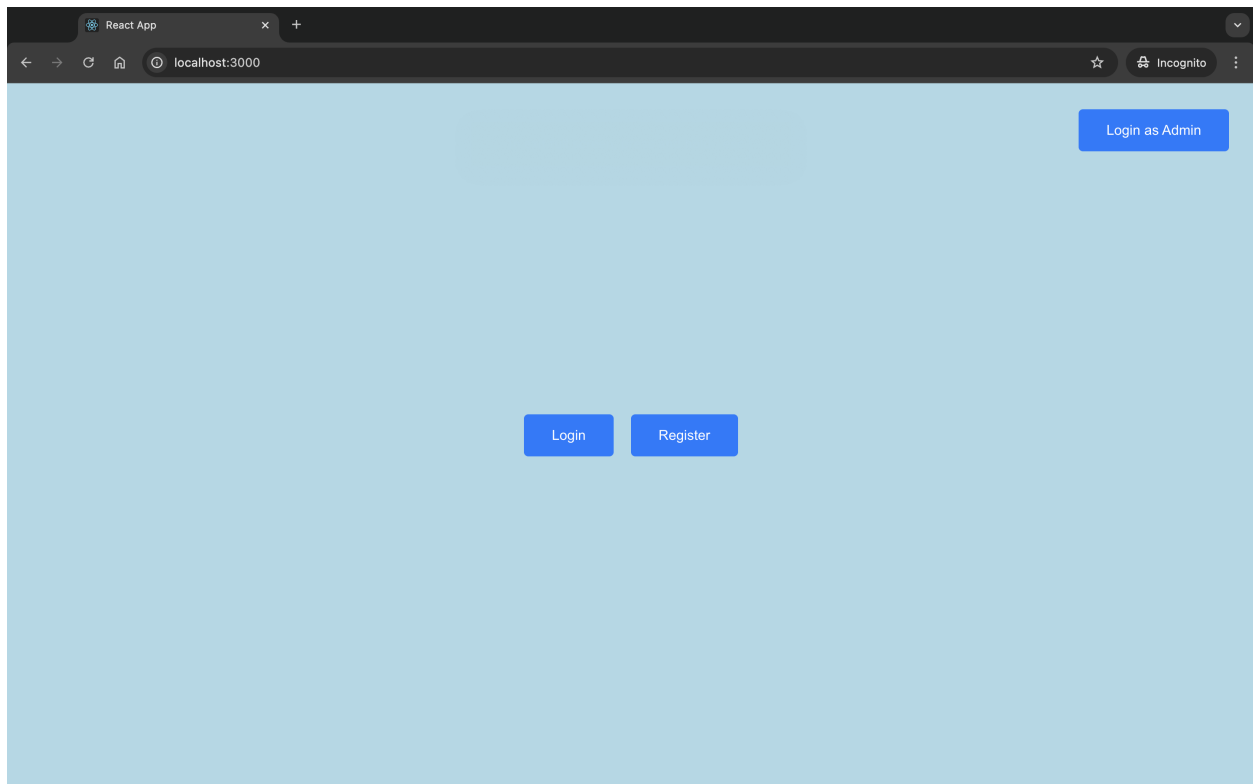
```
Do you want to Show the seat map of any of the buses, exit(-1)

2
1A 1B 1C
2A 2B 2C
3A 3B 3C
4A 4B 4C
```

red seats are unavailable and remaining are available.

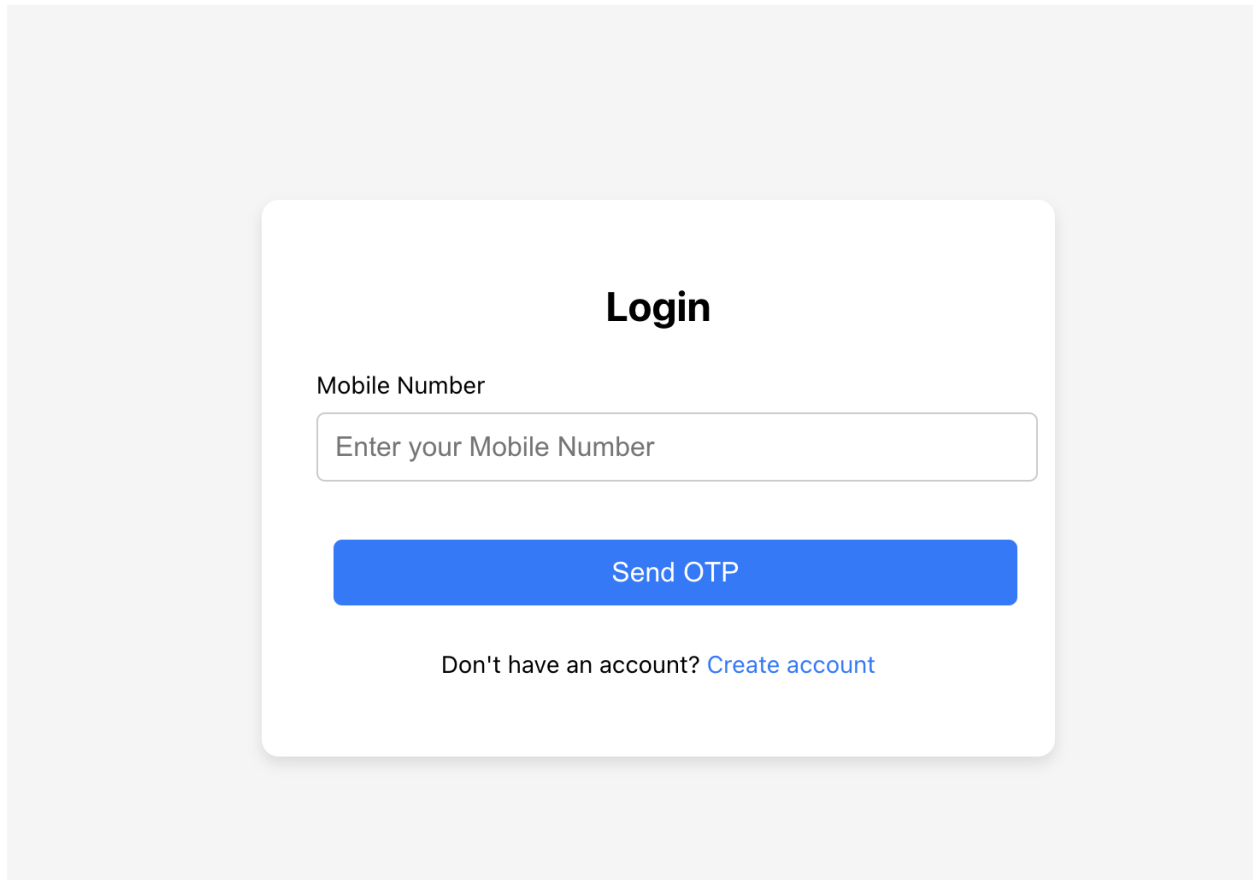
By selecting a Seat, he will get confirmation message.

## User flow in web



Landing page.





The image shows a login form centered on a light gray background. The form is a white rounded rectangle. At the top, the word "Login" is written in bold black text. Below it, the label "Mobile Number" is positioned to the left of a text input field. The input field has a light gray border and contains the placeholder text "Enter your Mobile Number". Below the input field is a solid blue button with the text "Send OTP" in white. At the bottom of the form, there is a link that says "Don't have an account? Create account", where "Create account" is in blue text.

**Login**

Mobile Number

Enter your Mobile Number

Send OTP

Don't have an account? [Create account](#)

login page

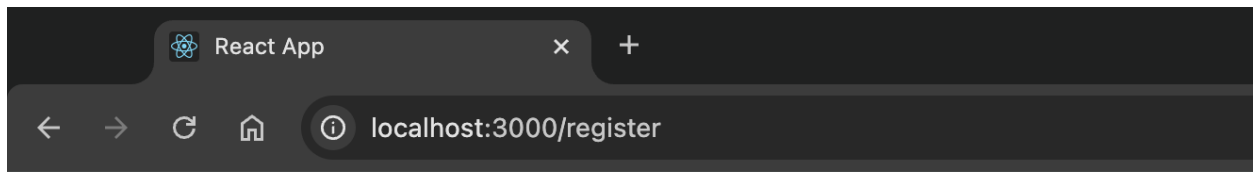
## Login

Enter OTP

123456

Verify OTP

enter otp



## Registration

Full Name \*

Gender \*

Mobile Number \*

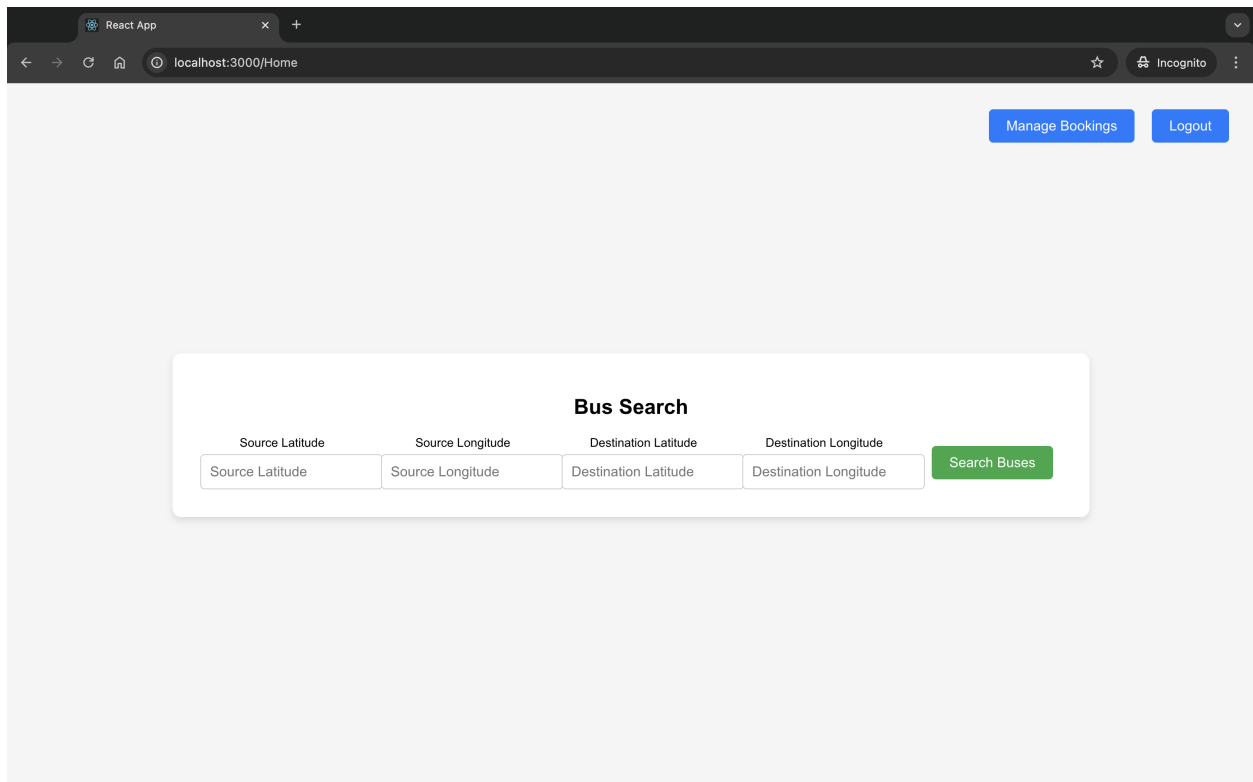
Verify

Email \*

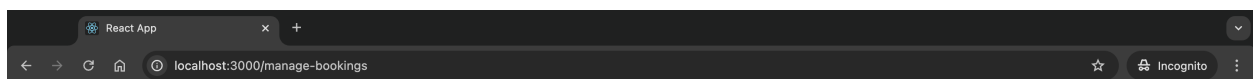
Verified

Register

Already Registered? [Login](#)



## Home page



Manage Bookings					
Booking ID	Bus name	Seat Number	Booking Time	Status	Action
5	Tata	3A	Date: 24:10:2024 Time: 14:18	CANCELLED	Not cancellable

Booking page. ( another button will be present, if the booking was not cancelled).

Manage Bookings					
Booking ID	Bus name	Seat Number	Booking Time	Status	Action
1	Tata	1A	Date: 24:10:2024 Time: 13:54	SUCCESSFUL	<button>Cancel Booking</button>
2	Zero	1A	Date: 24:10:2024 Time: 14:7	SUCCESSFUL	<button>Cancel Booking</button>
3	Zero	1B	Date: 24:10:2024 Time: 14:11	SUCCESSFUL	<button>Cancel Booking</button>

Search results.

Source Latitude

0

Source Longitude

0

Destination Latitude

10

Destination Longitude

0

Search Buses

Zero (ID: 2)

Total Seats:

Current Occupancy: 2

Not Live

View Seat Plan

Source Latitude

0

Source Longitude

0

Destination Latitude

6

Destination Longitude

0

Search Buses

Tata (ID: 1)

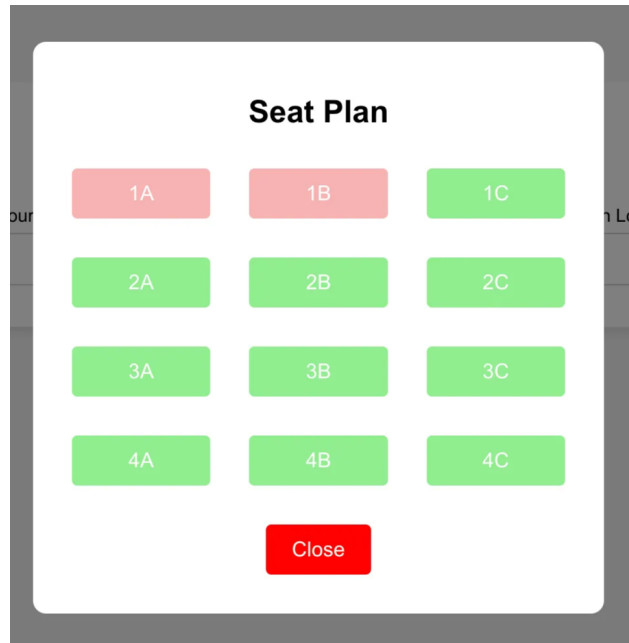
Total Seats:

Current Occupancy: 7

Not Live

View Seat Plan

colour of the bus changes, based on the availability.



Select a seat if you want to book and press book. To confirm booking, Booking status will given as an alert.

## Admin flow

```
Welcome, Admin! What would you like to do?
0. Login
4. Exit
0
Enter your phone number (10 digits):
01234556789
Enter OTP: 111111
Welcome, Admin! What would you like to do?
1. Add a bus
2. Modify bus details
3. Logout
4. Exit
1
Enter Bus Name:
Highway
Enter Total Rows: (default value)
5
Enter Total Columns: (default value)
2
Enter Current Occupancy (number of booked seats): 0
Enter number of stops in the route:
3
Enter longitude for stop 1:
0
Enter latitude for stop 1:
0
Enter longitude for stop 2:
5
Enter latitude for stop 2:
0
Enter longitude for stop 3:
10
```

As of now admin can only add a bus.

## Future scope

- Use a builder method to create different types of Buses ( economy bus, luxury bus). Can also do for seats.
- Have a better seat locking mechanism, (scalability is an issue).
- Notifications.
  - Different types of notifications, using interfaces ( important, promotional) and sending them in different ways.( text, mail).
  - Notifications in a trip. Set of passengers that belongs to that trip, sending them about their trip.
- Strategy pattern
  - Searching buses based on ETA or based on duration.
  - Search buses around particular radius( 1KM , stops around 1km vicinity of user).

<https://github.com/vivektangudu123/moveinsync>

## Refernce project

<https://github.com/vivektangudu123/clickcare>