19Z612 - APPLICATION DEVELOPMENT LABORATORY

Adarsh G - 20Z204
Jeevan Krishna K V - 20Z220
Nirmal M - 20Z267
Ajay Deepak P M - 21Z431

VEHICLE RENTAL SYSTEM

MARCH 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)
COIMBATORE – 641004

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641004

19Z612 - APPLICATION DEVELOPMENT LABORATORY

Bona fide record of work done by

Adarsh G - 20Z204

Jeevan Krishna K V - 20Z220

Nirmal M - 20Z267

Ajay Deepak P M - 21Z431

BACHELOR OF ENGINEERING

BRANCH: COMPUTER SCIENCE AND ENGINEERING

Application Development Laboratory report submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering, Branch: Computer Science and Engineering of PSG College of Technology.

…… ……

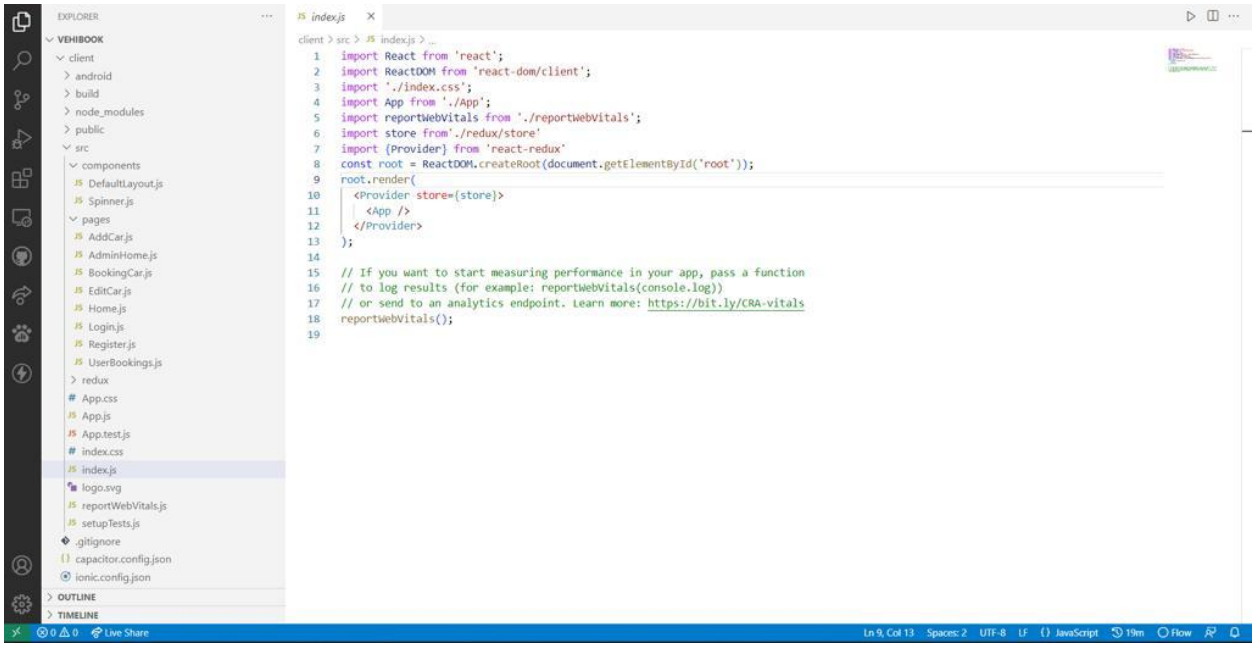
(Faculty Guide) (External Examiner)

OBJECTIVE

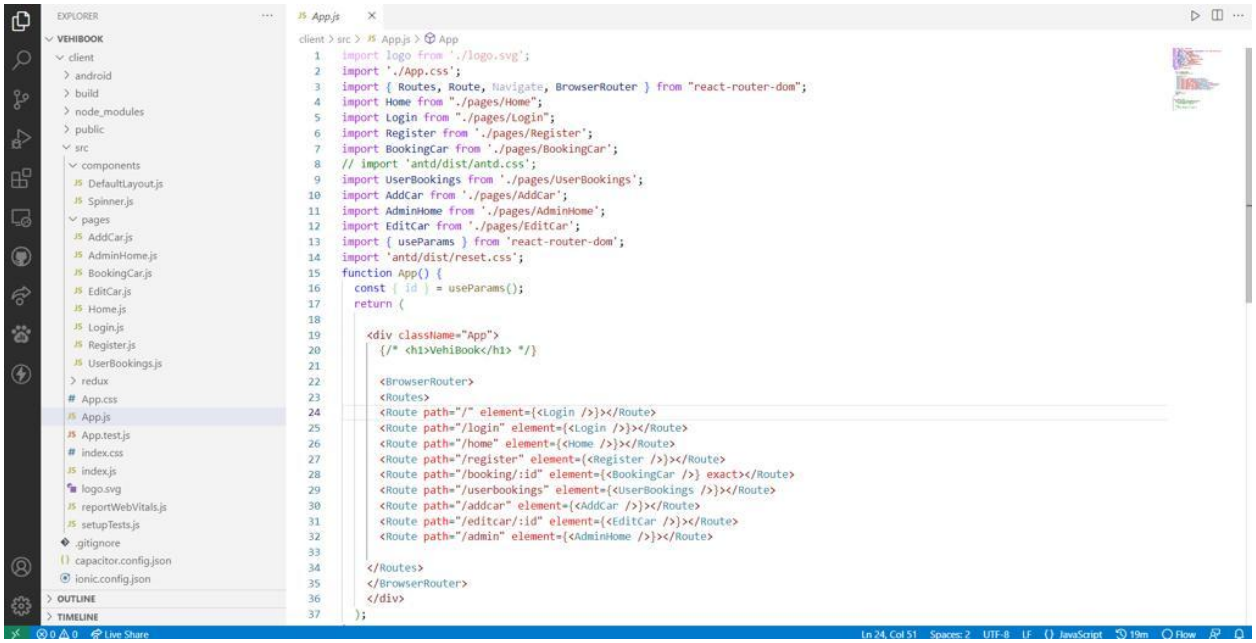
A vehicle rental system's goal is to give customers who need a car for a short period of time a convenient and adaptable mode of transportation. A vehicle rental system's main goals can be broadly divided into the following categories:

- Customer Satisfaction: The primary objective of any vehicle rental system is to provide a high level of customer satisfaction. This can be accomplished by offering customers a seamless and trouble-free rental experience, a variety of vehicles to suit their needs, and top-notch customer service.
- Vehicle Maintenance: The objective of a vehicle rental system is to maintain its vehicles in good condition. To keep the fleet's vehicles safe and roadworthy, this entails routine maintenance and servicing.
- Expansion: A vehicle rental system's expansion of operations and customer base is another goal. To provide it, add new rental locations, collaborate with tour operators, and offer new services to draw more clients.

CODE: WEB APPLICATION

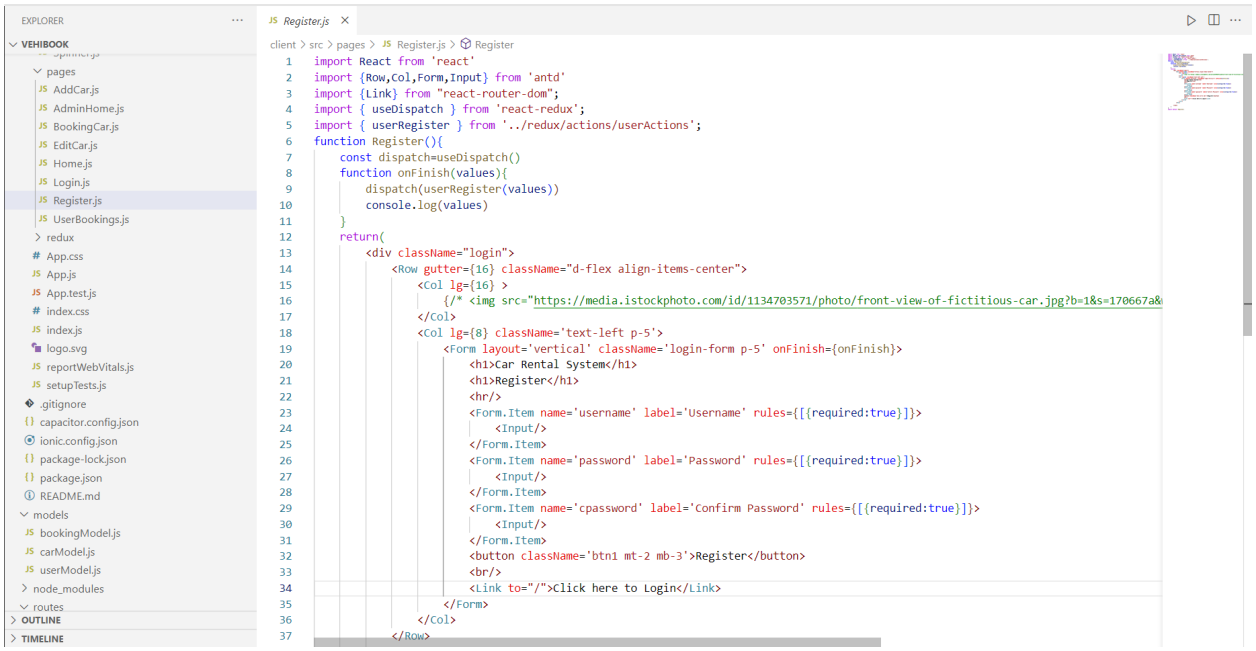


FRONT END



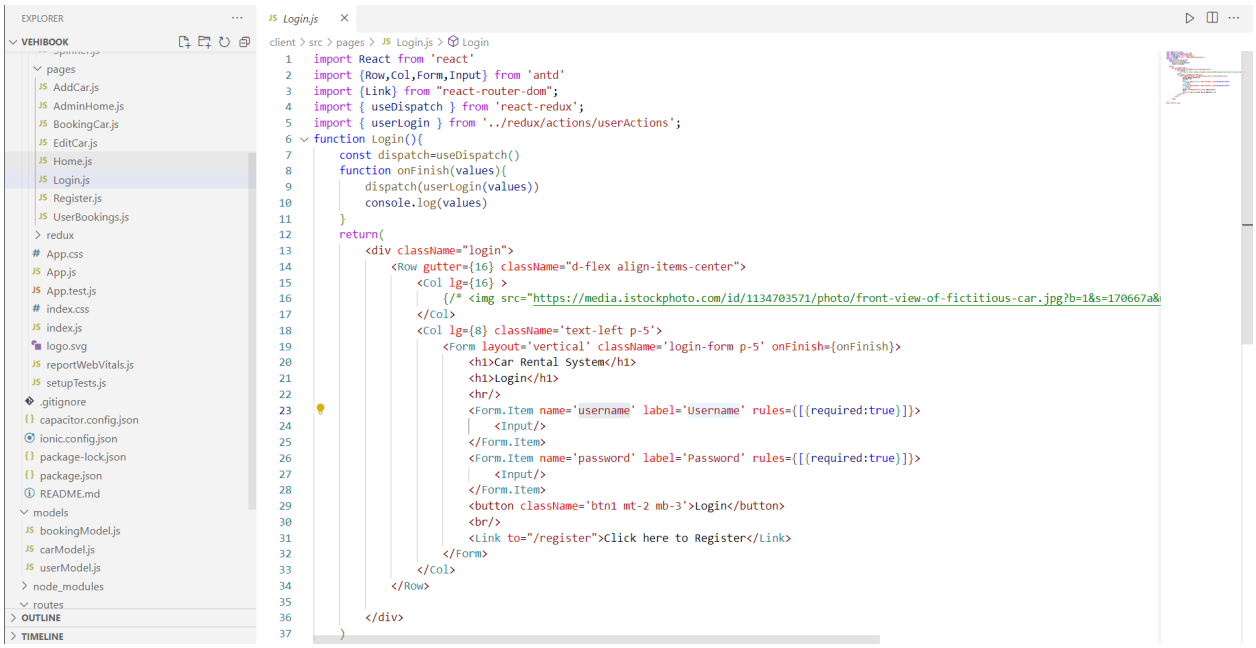
```
import logo from './logo.svg';
import './App.css';
import { Routes, Route, Navigate, BrowserRouter } from "react-router-dom";
import Home from "./pages/Home";
import Login from "./pages/Login";
import Register from './pages/Register';
import BookingCar from './pages/BookingCar';
// import 'antd/dist/antd.css';
import UserBookings from './pages/UserBookings';
import AddCar from './pages/AddCar';
import AdminHome from './pages/AdminHome';
import EditCar from './pages/EditCar';
import { useParams } from 'react-router-dom';
import 'antd/dist/reset.css';
function App() {
  const { id } = useParams();
  return (

    <div className="App">
      {/* <h1>VehiBook</h1> */}

      <BrowserRouter>
      <Routes>
      <Route path="/" element={<Login />}></Route>
      <Route path="/login" element={<Login />}></Route>
      <Route path="/home" element={<Home />}></Route>
      <Route path="/register" element={<Register />}></Route>
      <Route path="/booking/:id" element={<BookingCar />} exact></Route>
      <Route path="/userbookings" element={<UserBookings />}></Route>
      <Route path="/addcar" element={<AddCar />}></Route>
      <Route path="/editcar/:id" element={<EditCar />}></Route>
      <Route path="/admin" element={<AdminHome />}></Route>

      </Routes>
      </BrowserRouter>
      </div>
  );
```

Register



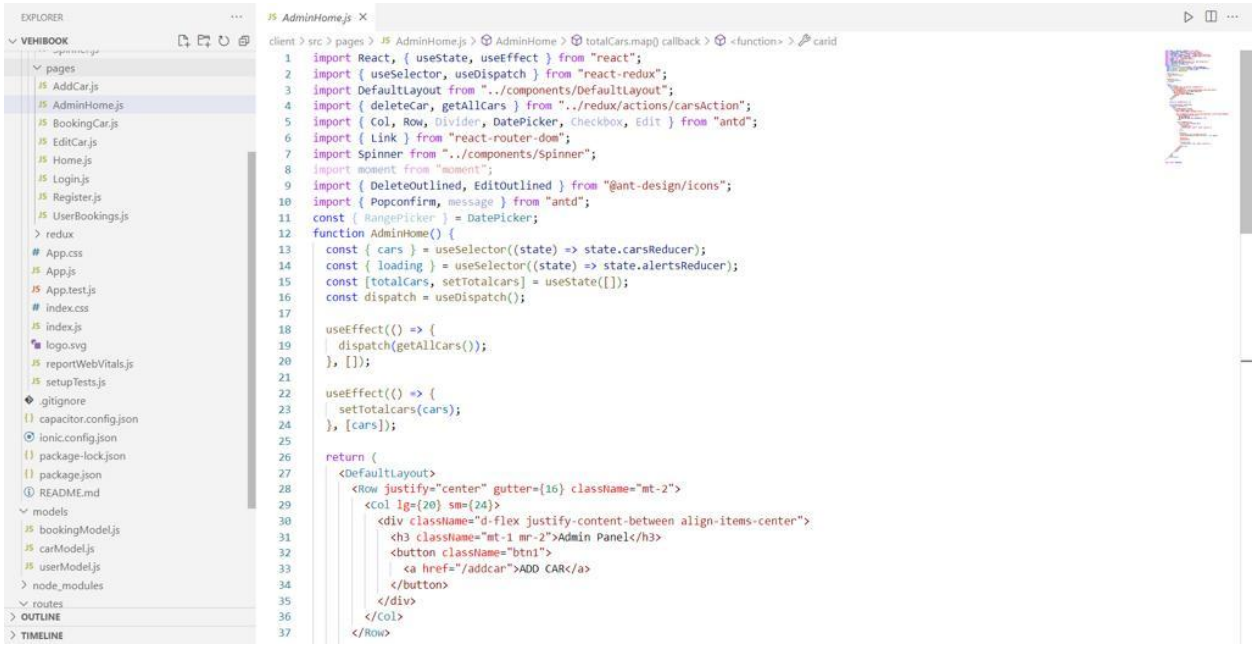
```
import React from 'react'
import {Row,Col,Form,Input} from 'antd'
import {Link} from "react-router-dom";
import { useDispatch } from 'react-redux';
import { userRegister } from '../redux/actions/userActions';
function Register(){
    const dispatch=useDispatch()
    function onFinish(values){
        dispatch(userRegister(values))
        console.log(values)
    }
    return(
        <div className="login">
            <Row gutter={16} className="d-flex align-items-center">
                <Col lg={16} >
                    {/* <img src="https://media.istockphoto.com/id/1134703571/photo/front-view-of-fictitious-car.jpg?b=1&s=170667a&
                </Col>
                <Col lg={8} className='text-left p-5'>
                    <Form layout='vertical' className='login-form p-5' onFinish={onFinish}>
                        <h1>Car Rental System</h1>
                        <h1>Register</h1>
                        <hr/>
                        <Form.Item name='username' label='Username' rules={[{required:true}]}>
                            <Input/>
                        </Form.Item>
                        <Form.Item name='password' label='Password' rules={[{required:true}]}>
                            <Input/>
                        </Form.Item>
                        <Form.Item name='cpassword' label='Confirm Password' rules={[{required:true}]}>
                            <Input/>
                        </Form.Item>
                        <button className='btn1 mt-2 mb-3'>Register</button>
                        <br/>
                        <Link to="/">Click here to Login</Link>
                    </Form>
                </Col>
            </Row>
```

Login



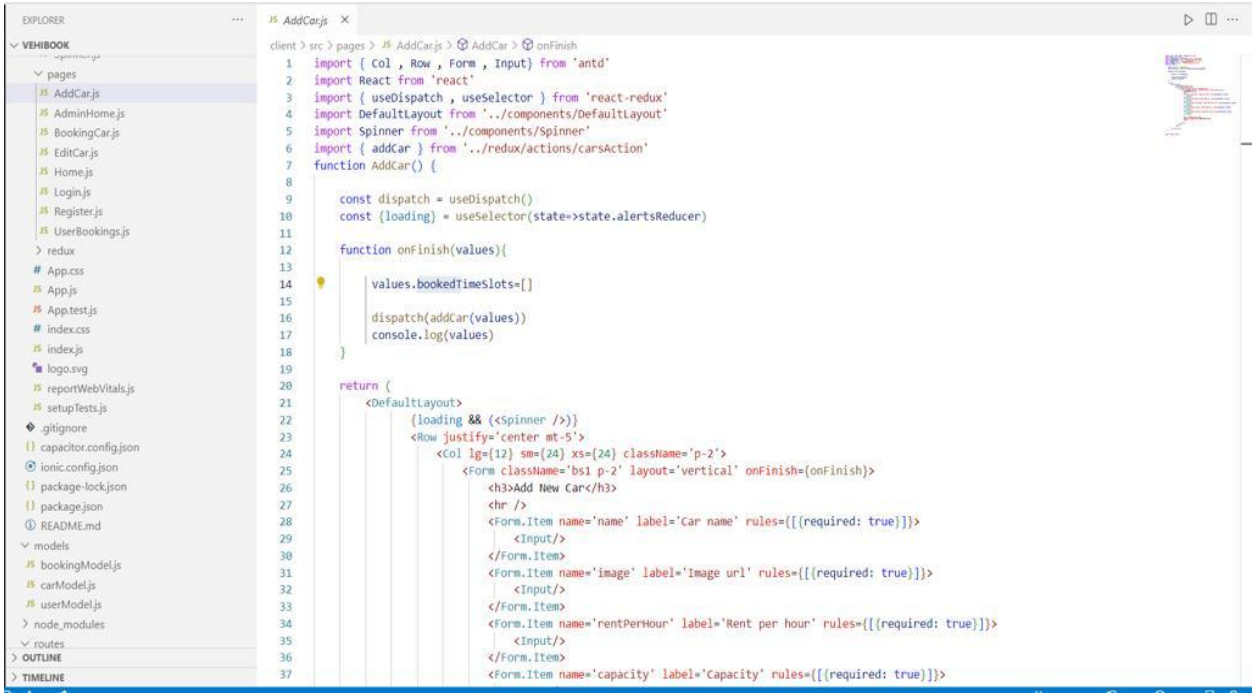
```
import React from 'react'
import {Row,Col,Form,Input} from 'antd'
import {Link} from "react-router-dom";
import { useDispatch } from 'react-redux';
import { userLogin } from '../redux/actions/userActions';
function Login(){
    const dispatch=useDispatch()
    function onFinish(values){
        dispatch(userLogin(values))
        console.log(values)
    }
    return(
        <div className="login">
            <Row gutter={16} className="d-flex align-items-center">
                <Col lg={16} >
                    {/* <img src="https://media.istockphoto.com/id/1134703571/photo/front-view-of-fictitious-car.jpg?b=1&s=170667a&
                </Col>
                <Col lg={8} className='text-left p-5'>
                    <Form layout='vertical' className='login-form p-5' onFinish={onFinish}>
                        <h1>Car Rental System</h1>
                        <h1>Login</h1>
                        <hr/>
                        <Form.Item name='username' label='Username' rules={[{required:true}]}>
                            <Input/>
                        </Form.Item>
                        <Form.Item name='password' label='Password' rules={[{required:true}]}>
                            <Input/>
                        </Form.Item>
                        <button className='btn1 mt-2 mb-3'>Login</button>
                        <br/>
                        <Link to="/register">Click here to Register</Link>
                    </Form>
                </Col>
            </Row>

        </div>
    )
```

Admin Home



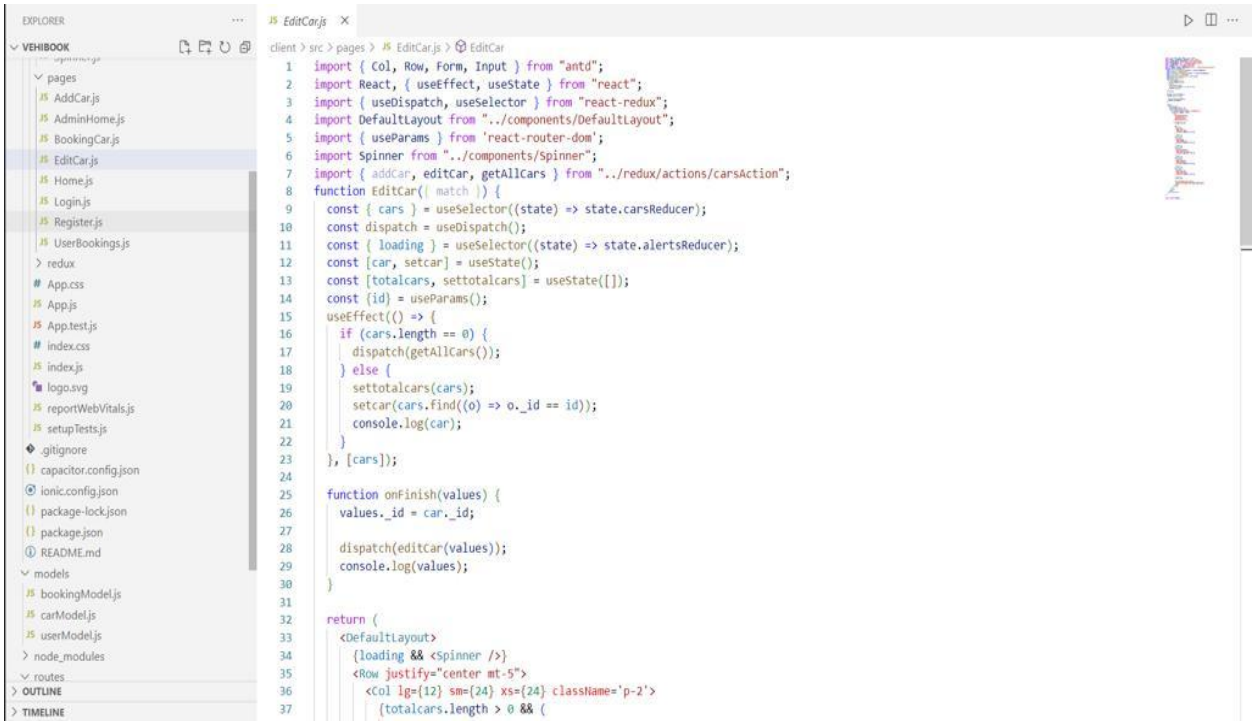
```
import React, { useState, useEffect } from "react";
import { useSelector, useDispatch } from "react-redux";
import DefaultLayout from "../components/DefaultLayout";
import { deleteCar, getAllCars } from "../redux/actions/carsAction";
import { Col, Row, Divider, DatePicker, Checkbox, Edit } from "antd";
import { Link } from "react-router-dom";
import Spinner from "../components/Spinner";
import moment from "moment";
import { DeleteOutlined, EditOutlined } from "@ant-design/icons";
import { Popconfirm, message } from "antd";
const { RangePicker } = DatePicker;
function AdminHome() {
  const { cars } = useSelector((state) => state.carsReducer);
  const { loading } = useSelector((state) => state.alertsReducer);
  const [totalCars, setTotalcars] = useState([]);
  const dispatch = useDispatch();

  useEffect(() => {
    dispatch(getAllCars());
  }, []);

  useEffect(() => {
    setTotalcars(cars);
  }, [cars]);

  return (
    <DefaultLayout>
      <Row justify="center" gutter={16} className="mt-2">
        <Col lg={20} sm={24}>
          <div className="d-flex justify-content-between align-items-center">
            <h3 className="mt-1 mr-2">Admin Panel</h3>
            <button className="btn1">
              <a href="/addcar">ADD CAR</a>
            </button>
          </div>
        </Col>
      </Row>
```

Add Car



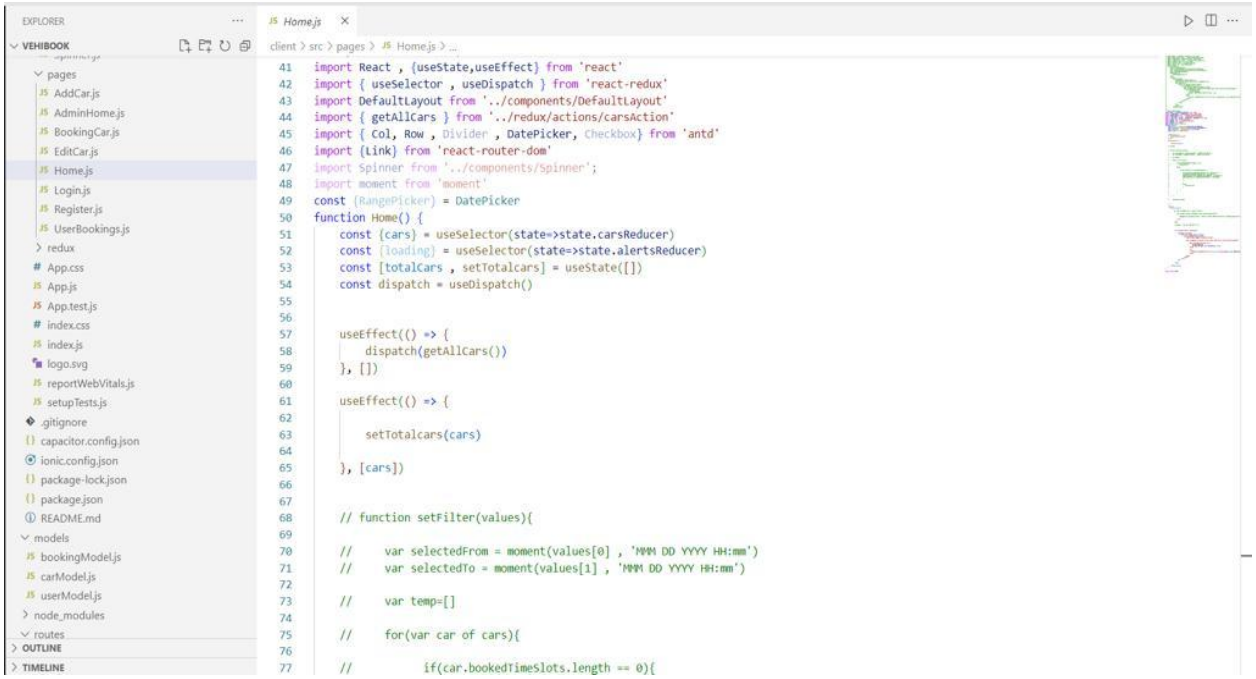
```
import { Col , Row , Form , Input} from 'antd'
import React from 'react'
import { useDispatch , useSelector } from 'react-redux'
import DefaultLayout from '../components/DefaultLayout'
import Spinner from '../components/Spinner'
import { addCar } from '../redux/actions/carsAction'
function AddCar() {

    const dispatch = useDispatch()
    const {loading} = useSelector(state=>state.alertsReducer)

    function onFinish(values){

        values.bookedTimeSlots=[]

        dispatch(addCar(values))
        console.log(values)
    }

    return (
        <DefaultLayout>
                {loading && (<Spinner />)}
                <Row justify='center mt-5'>
                    <Col lg={12} sm={24} xs={24} className='p-2'>
                        <Form className='bs1 p-2' layout='vertical' onFinish={onFinish}>
                            <h3>Add New Car</h3>
                            <hr />
                            <Form.Item name='name' label='Car name' rules={[{required: true}]}>
                                <Input/>
                            </Form.Item>
                            <Form.Item name='image' label='Image url' rules={[{required: true}]}>
                                <Input/>
                            </Form.Item>
                            <Form.Item name='rentPerHour' label='Rent per hour' rules={[{required: true}]}>
                                <Input/>
                            </Form.Item>
                            <Form.Item name='capacity' label='Capacity' rules={[{required: true}]}>
```

Edit Car

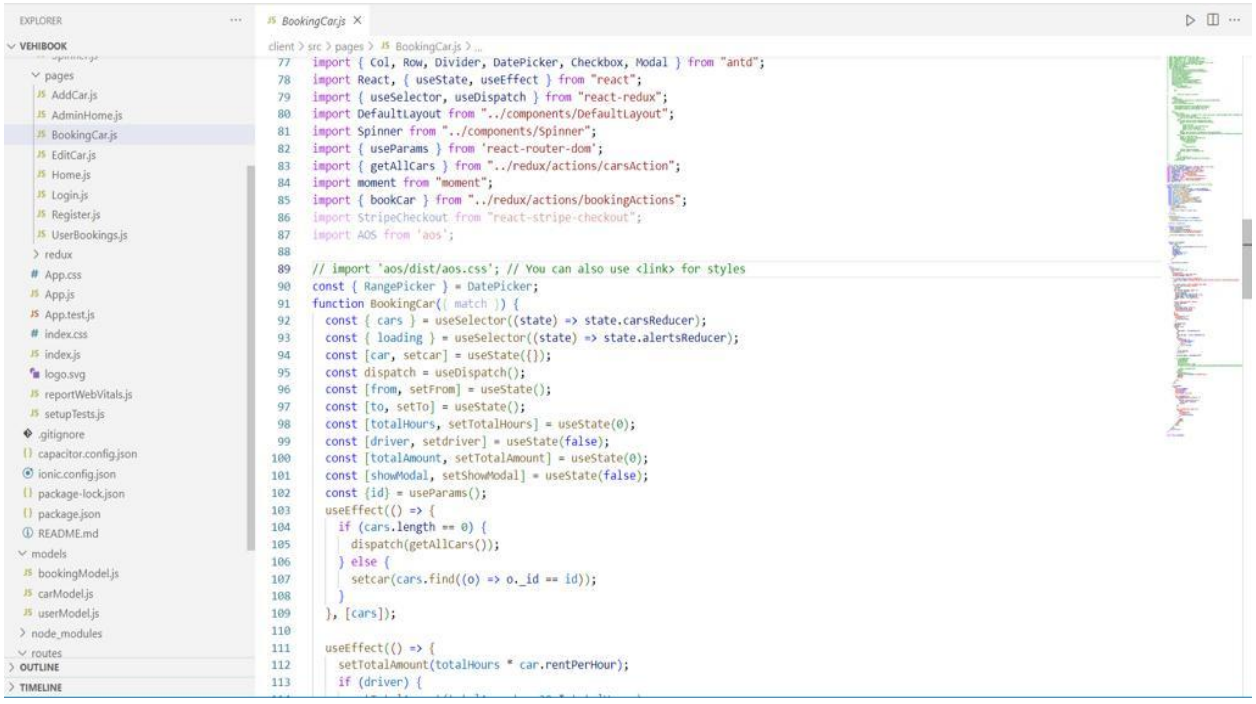


```
import { Col, Row, Form, Input } from "antd";
import React, { useEffect, useState } from "react";
import { useDispatch, useSelector } from "react-redux";
import DefaultLayout from "../components/DefaultLayout";
import { useParams } from 'react-router-dom';
import Spinner from "../components/Spinner";
import { addCar, editCar, getAllCars } from "../redux/actions/carsAction";
function EditCar({ match }) {
  const { cars } = useSelector((state) => state.carsReducer);
  const dispatch = useDispatch();
  const { loading } = useSelector((state) => state.alertsReducer);
  const [car, setcar] = useState();
  const [totalcars, settotalcars] = useState([]);
  const {id} = useParams();
  useEffect(() => {
    if (cars.length == 0) {
      dispatch(getAllCars());
    } else {
      settotalcars(cars);
      setcar(cars.find((o) => o._id == id));
      console.log(car);
    }
  }, [cars]);

  function onFinish(values) {
    values._id = car._id;

    dispatch(editCar(values));
    console.log(values);
  }

  return (
    <DefaultLayout>
      {loading && <Spinner />}
      <Row justify="center mt-5">
        <Col lg={12} sm={24} xs={24} className='p-2'>
          {totalcars.length > 0 && (
```

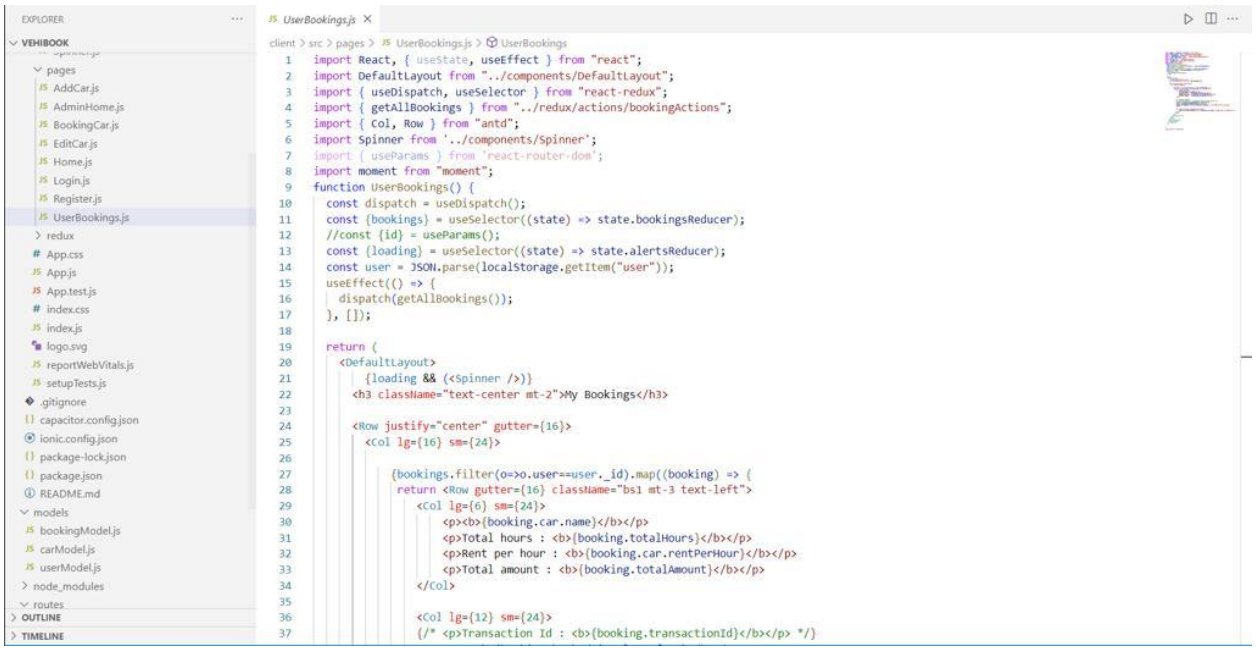
User Home



Booking Car

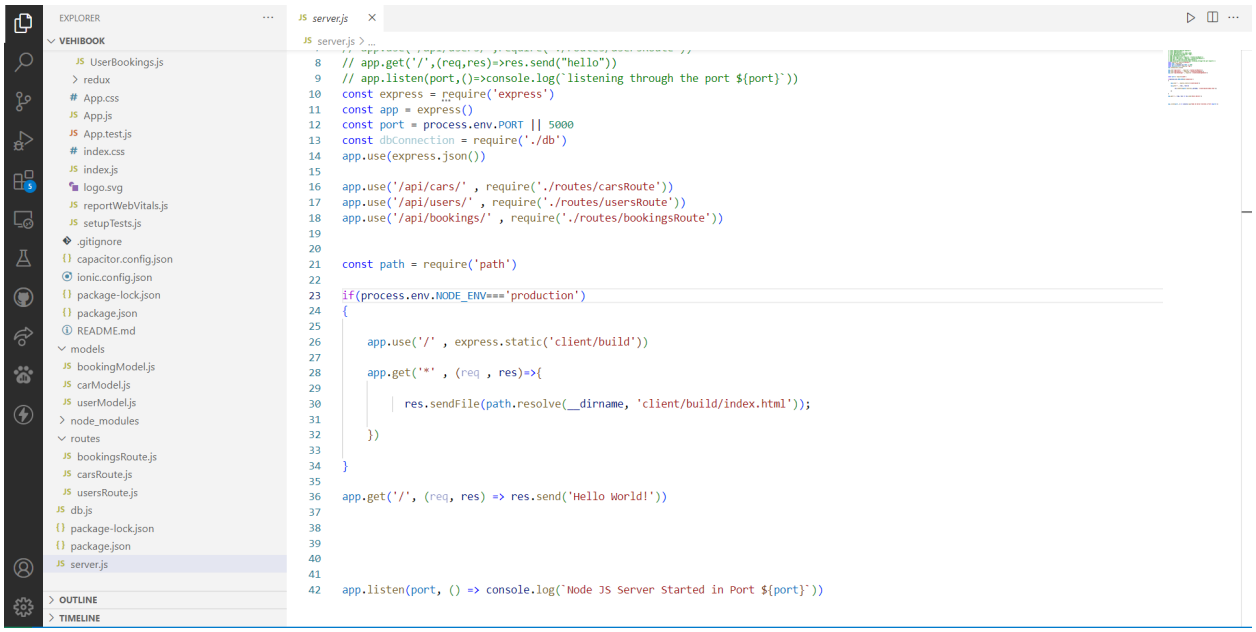


User Booking



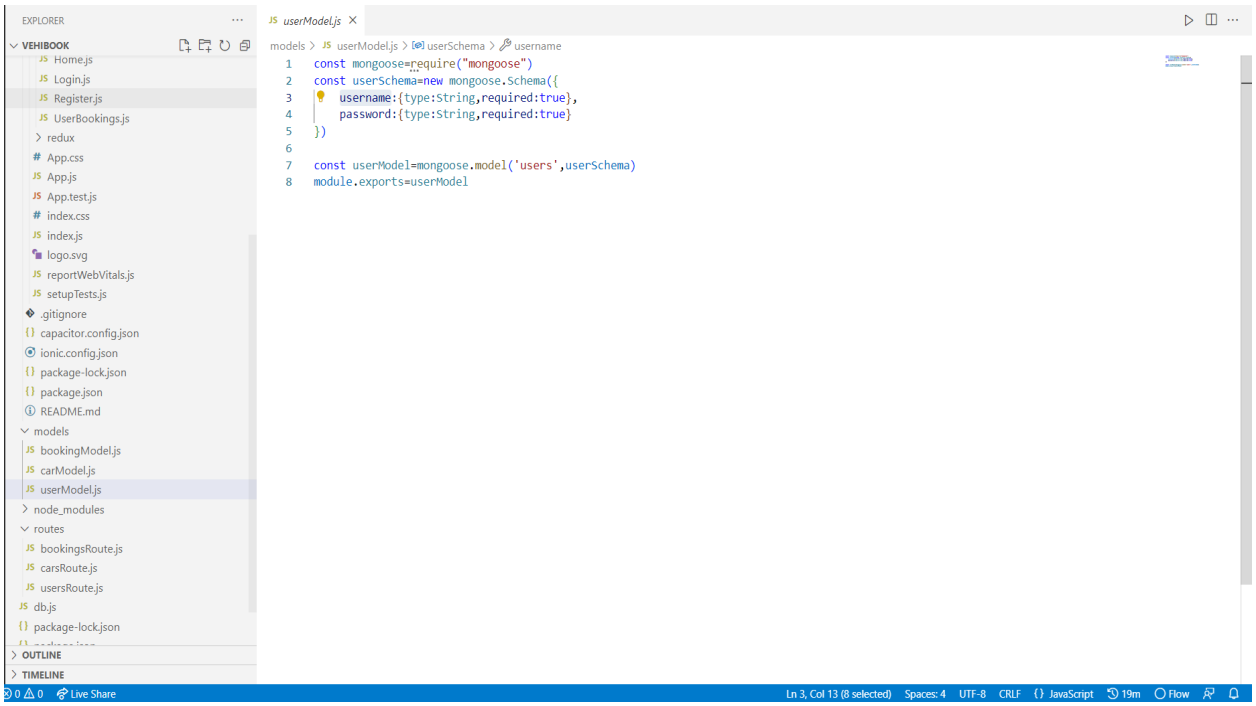
```
EXPLORER
VEHIBOOK
  pages
    AddCar.js
    AdminHome.js
    BookingCar.js
    EditCar.js
    Home.js
    Login.js
    Register.js
    UserBookings.js
  redux
  App.css
  App.js
  App.test.js
  index.css
  index.js
  logo.svg
  reportWebVitals.js
  setupTests.js
  .gitignore
  capacitor.config.json
  ionic.config.json
  package-lock.json
  package.json
  README.md
  models
    bookingModel.js
    carModel.js
    userModel.js
  node_modules
  routes
OUTLINE
TIMELINE

UserBookings.js
client > src > pages > UserBookings.js > UserBookings

import React, { useState, useEffect } from "react";
import DefaultLayout from "../components/DefaultLayout";
import { useDispatch, useSelector } from "react-redux";
import { getAllBookings } from "../redux/actions/bookingActions";
import { Col, Row } from "antd";
import Spinner from '../components/Spinner';
import { useParams } from 'react-router-dom';
import moment from "moment";
function UserBookings() {
  const dispatch = useDispatch();
  const {bookings} = useSelector((state) => state.bookingsReducer);
  //const {id} = useParams();
  const {loading} = useSelector((state) => state.alertsReducer);
  const user = JSON.parse(localStorage.getItem("user"));
  useEffect(() => {
    dispatch(getAllBookings());
  }, []);

  return (
    <DefaultLayout>
        {loading && (<Spinner />)}
      <h3 className="text-center mt-2">My Bookings</h3>

      <Row justify="center" gutter={16}>
        <Col lg={16} sm={24}>

          {bookings.filter(o=>o.user==user._id).map((booking) => {
          return <Row gutter={16} className="bs1 mt-3 text-left">
              <Col lg={6} sm={24}>
                  <p><b>{booking.car.name}</b></p>
                  <p>Total hours : <b>{booking.totalHours}</b></p>
                  <p>Rent per hour : <b>{booking.car.rentPerHour}</b></p>
                  <p>Total amount : <b>{booking.totalAmount}</b></p>
              </Col>

              <Col lg={12} sm={24}>
              {/* <p>Transaction Id : <b>{booking.transactionId}</b></p> */}
```

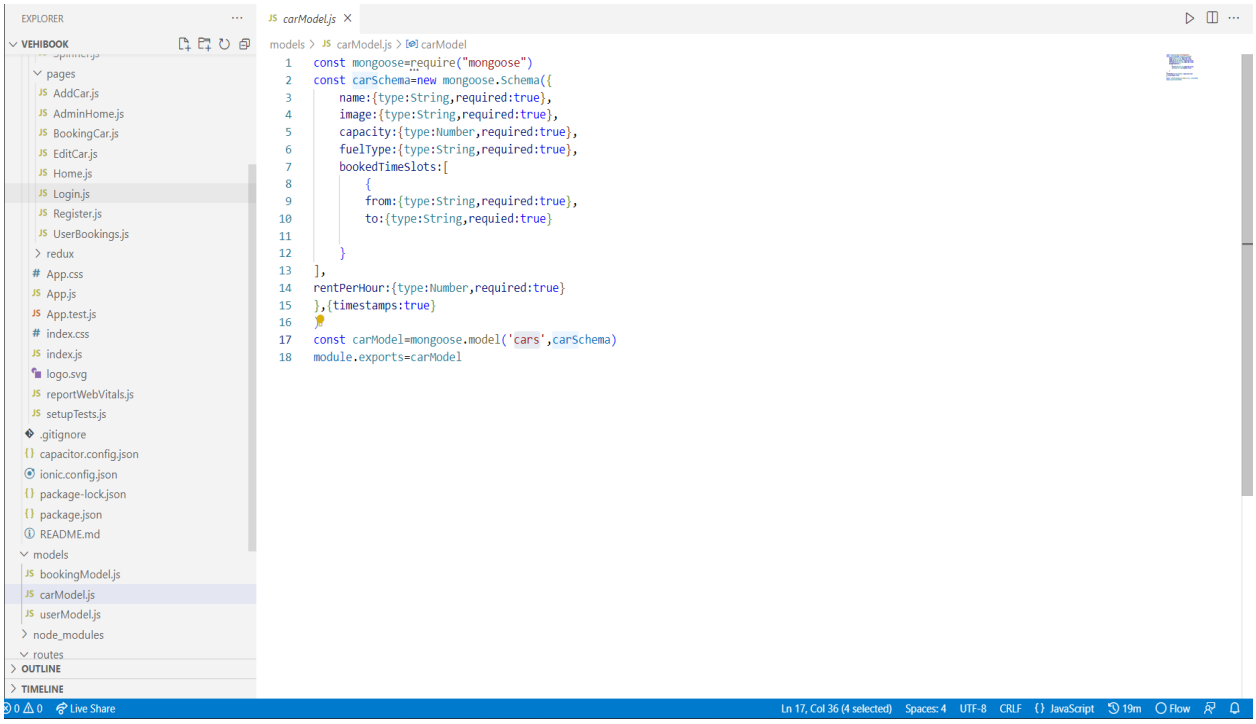
BACK END

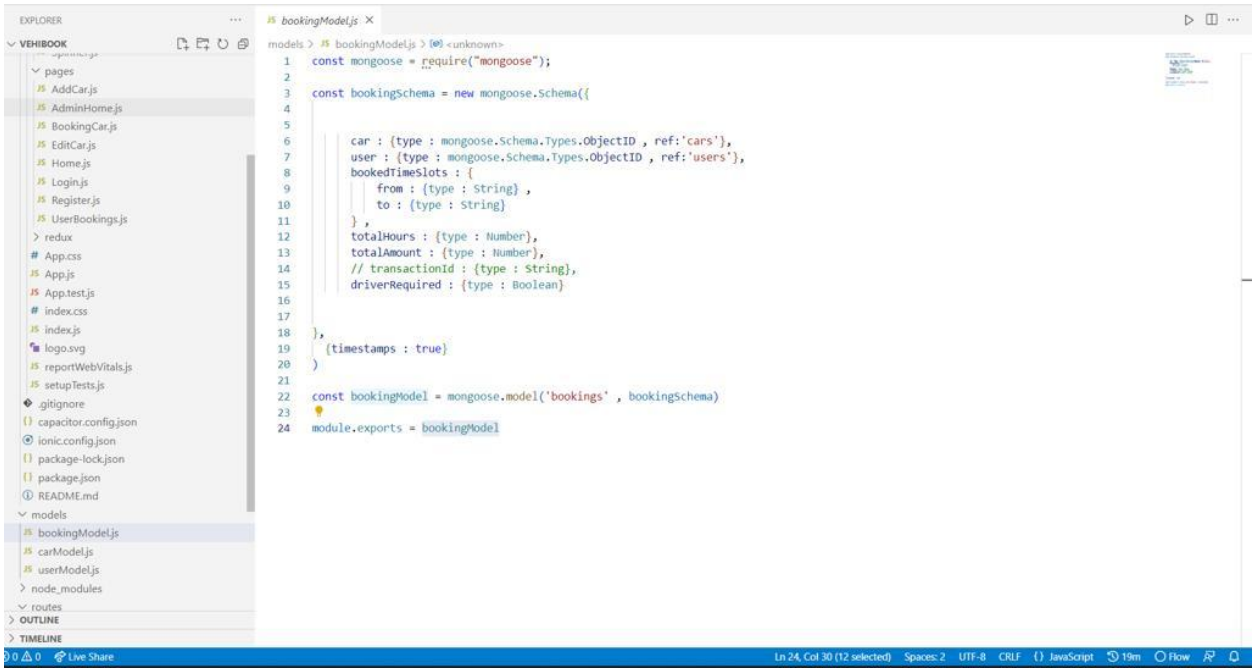


```
EXPLORER
VEHIBOOK
  UserBookings.js
  redux
  App.css
  App.js
  App.test.js
  index.css
  index.js
  logo.svg
  reportWebVitals.js
  setupTests.js
  .gitignore
  capacitor.config.json
  ionic.config.json
  package-lock.json
  package.json
  README.md
  models
    bookingModel.js
    carModel.js
    userModel.js
  node_modules
  routes
    bookingsRoute.js
    carsRoute.js
    usersRoute.js
  db.js
  package-lock.json
  package.json
  server.js
OUTLINE
TIMELINE

server.js
server.js > ...

// app.get('/',(req,res)=>res.send("hello"))
// app.listen(port,()=>console.log(`listening through the port ${port}`))
const express = require('express')
const app = express()
const port = process.env.PORT || 5000
const dbConnection = require('./db')
app.use(express.json())

app.use('/api/cars/' , require('./routes/carsRoute'))
app.use('/api/users/' , require('./routes/usersRoute'))
app.use('/api/bookings/' , require('./routes/bookingsRoute'))


const path = require('path')

if(process.env.NODE_ENV==='production')
{

    app.use('/' , express.static('client/build'))

    app.get('*' , (req , res)=>{

        res.sendFile(path.resolve(__dirname, 'client/build/index.html'));

    })
}

app.get('/', (req, res) => res.send('Hello World!'))



app.listen(port, () => console.log(`Node JS Server Started in Port ${port}`))
```

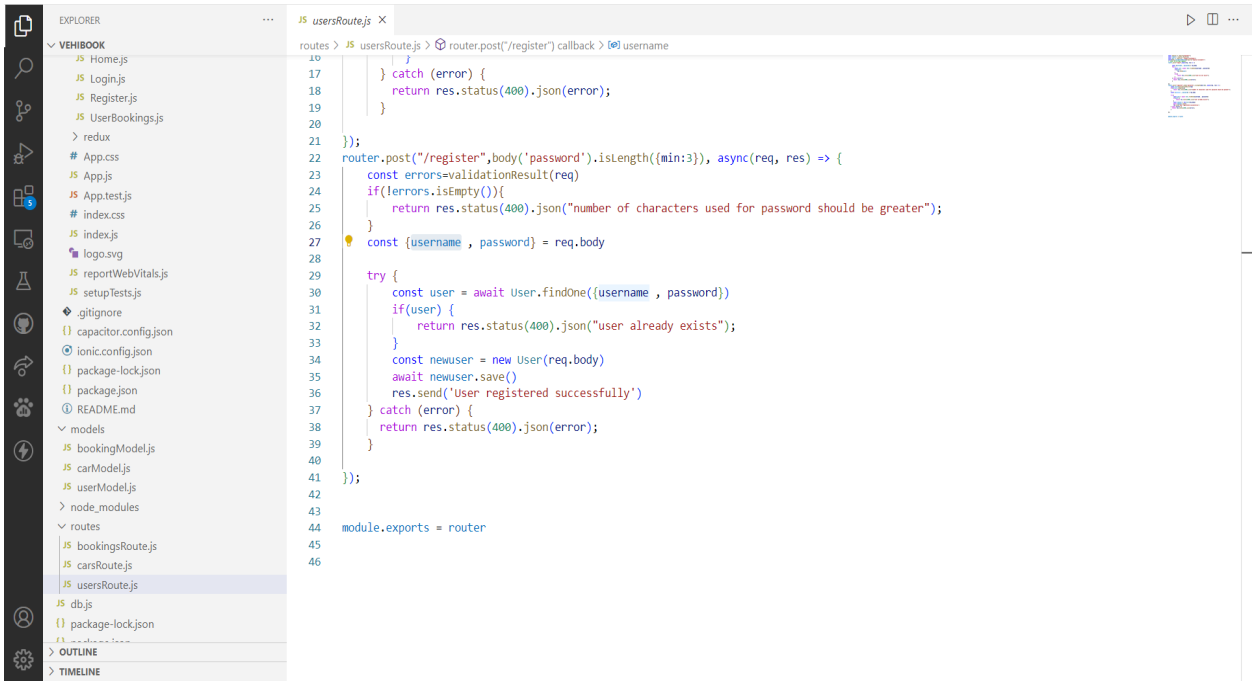
Schema

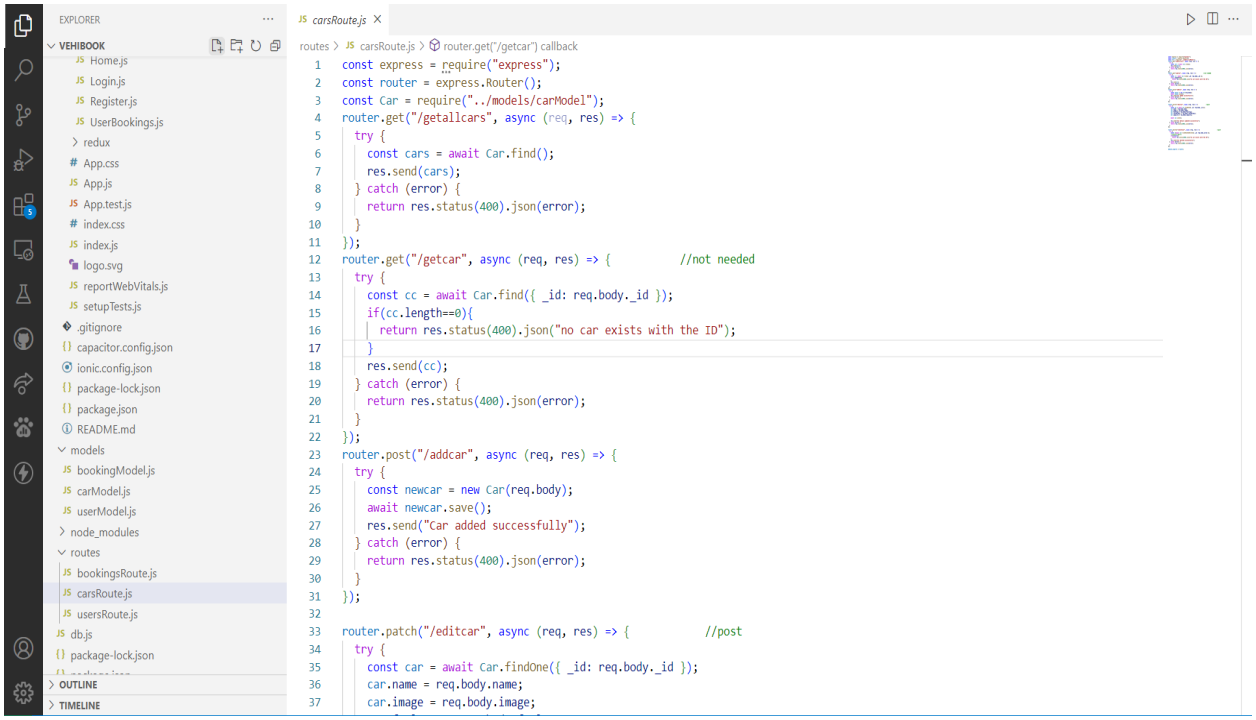

EXPLORER
VEHIBOOK
Home.js
Login.js
Register.js
UserBookings.js
redux
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
capacitor.config.json
ionic.config.json
package-lock.json
package.json
README.md
models
bookingModel.js
carModel.js
userModel.js
node_modules
routes
bookingsRoute.js
carsRoute.js
usersRoute.js
db.js
package-lock.json
OUTLINE
TIMELINE
userModel.js
models > userModel.js > userSchema > username
1 const mongoose=require("mongoose")
2 const userSchema=new mongoose.Schema({
3 username:{type:String,required:true},
4 password:{type:String,required:true}
5 })
7 const userModel=mongoose.model('users',userSchema)
8 module.exports=userModel
0 0 Live Share
Ln 3, Col 13 (8 selected) Spaces: 4 UTF-8 CRLF JavaScript 19m Flow


EXPLORER
VEHIBOOK
pages
AddCar.js
AdminHome.js
BookingCar.js
EditCar.js
Home.js
Login.js
Register.js
UserBookings.js
redux
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
capacitor.config.json
ionic.config.json
package-lock.json
package.json
README.md
models
bookingModel.js
carModel.js
userModel.js
node_modules
routes
OUTLINE
TIMELINE
carModel.js
models > carModel.js > carModel
1 const mongoose=require("mongoose")
2 const carSchema=new mongoose.Schema({
3 name:{type:String,required:true},
4 image:{type:String,required:true},
5 capacity:{type:Number,required:true},
6 fuelType:{type:String,required:true},
7 bookedTimeSlots:[
8 {
9 from:{type:String,required:true},
10 to:{type:String,requied:true}
12 }
13],
14 rentPerHour:{type:Number,required:true}
15 },{timestamps:true}
17 const carModel=mongoose.model('cars',carSchema)
18 module.exports=carModel
0 0 Live Share
Ln 17, Col 36 (4 selected) Spaces: 4 UTF-8 CRLF JavaScript 19m Flow


EXPLORER
VEHIBOOK
pages
AddCar.js
AdminHome.js
BookingCar.js
EditCar.js
Home.js
Login.js
Register.js
UserBookings.js
redux
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
capacitor.config.json
ionic.config.json
package-lock.json
package.json
README.md
models
bookingModel.js
carModel.js
userModel.js
node_modules
routes
OUTLINE
TIMELINE
bookingModel.js
models > bookingModel.js > <unknown>
const mongoose = require("mongoose");
const bookingSchema = new mongoose.Schema({
car : {type : mongoose.Schema.Types.ObjectID , ref:'cars'},
user : {type : mongoose.Schema.Types.ObjectID , ref:'users'},
bookedTimeSlots : {
from : {type : String} ,
to : {type : String}
} ,
totalHours : {type : Number},
totalAmount : {type : Number},
// transactionId : {type : String},
driverRequired : {type : Boolean}
},
{timestamps : true}
)
const bookingModel = mongoose.model('bookings' , bookingSchema)
module.exports = bookingModel
Live Share
Ln 24, Col 30 (12 selected)
Spaces: 2
UTF-8
CRLF
JavaScript
19m
Flow

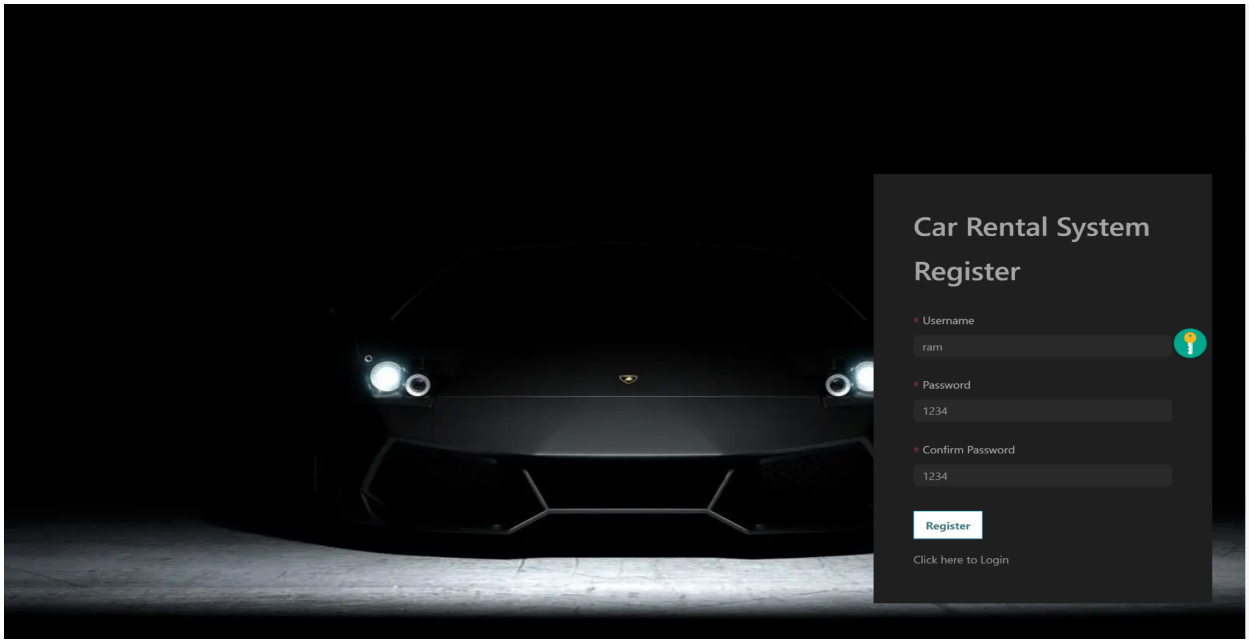
Routes


EXPLORER
VEHIBOOK
Home.js
Login.js
Register.js
UserBookings.js
redux
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
capacitor.config.json
ionic.config.json
package-lock.json
package.json
README.md
models
bookingModel.js
carModel.js
userModel.js
node_modules
routes
bookingsRoute.js
carsRoute.js
usersRoute.js
db.js
package-lock.json
OUTLINE
TIMELINE
usersRoute.js
routes > usersRoute.js > router.post("/register") callback > username
} catch (error) {
return res.status(400).json(error);
}
});
router.post("/register",body('password').isLength({min:3}), async(req, res) => {
const errors=validationResult(req)
if(!errors.isEmpty()){
return res.status(400).json("number of characters used for password should be greater");
}
const {username , password} = req.body
try {
const user = await User.findOne({username , password})
if(user) {
return res.status(400).json("user already exists");
}
const newuser = new User(req.body)
await newuser.save()
res.send('User registered successfully')
} catch (error) {
return res.status(400).json(error);
}
});
module.exports = router

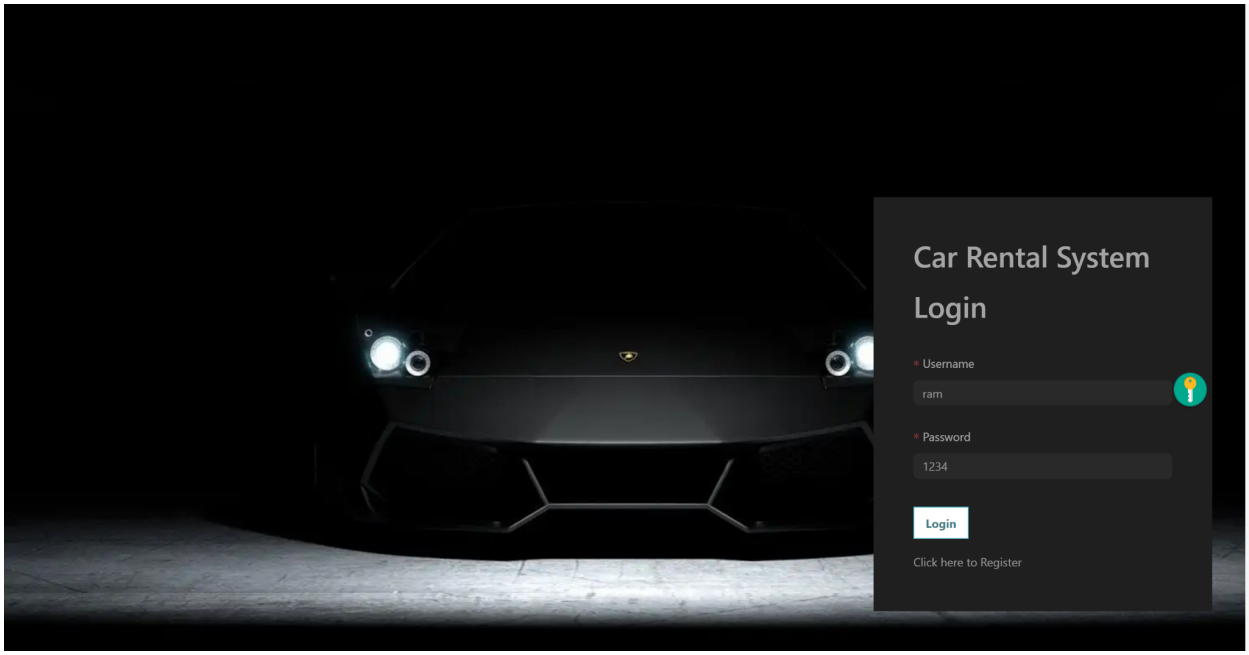

EXPLORER
VEHIBOOK
Home.js
Login.js
Register.js
UserBookings.js
redux
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
capacitor.config.json
ionic.config.json
package-lock.json
package.json
README.md
models
bookingModel.js
carModel.js
userModel.js
node_modules
routes
bookingsRoute.js
carsRoute.js
usersRoute.js
db.js
package-lock.json
OUTLINE
TIMELINE
carsRoute.js
routes > carsRoute.js > router.get("/getcar") callback
const express = require("express");
const router = express.Router();
const Car = require("../models/carModel");
router.get("/getallcars", async (req, res) => {
try {
const cars = await Car.find();
res.send(cars);
} catch (error) {
return res.status(400).json(error);
}
});
router.get("/getcar", async (req, res) => { //not needed
try {
const cc = await Car.find({ _id: req.body._id });
if(cc.length==0){
return res.status(400).json("no car exists with the ID");
}
res.send(cc);
} catch (error) {
return res.status(400).json(error);
}
});
router.post("/addcar", async (req, res) => {
try {
const newcar = new Car(req.body);
await newcar.save();
res.send("Car added successfully");
} catch (error) {
return res.status(400).json(error);
}
});
router.patch("/editcar", async (req, res) => { //post
try {
const car = await Car.findOne({ _id: req.body._id });
car.name = req.body.name;
car.image = req.body.image;

IMAGES: WEB APPLICATION

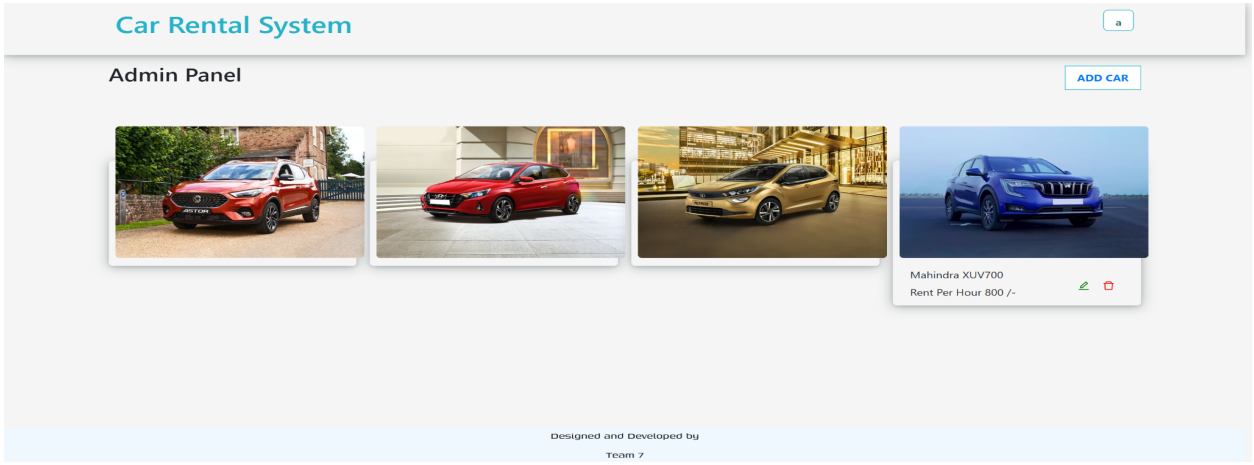
Register



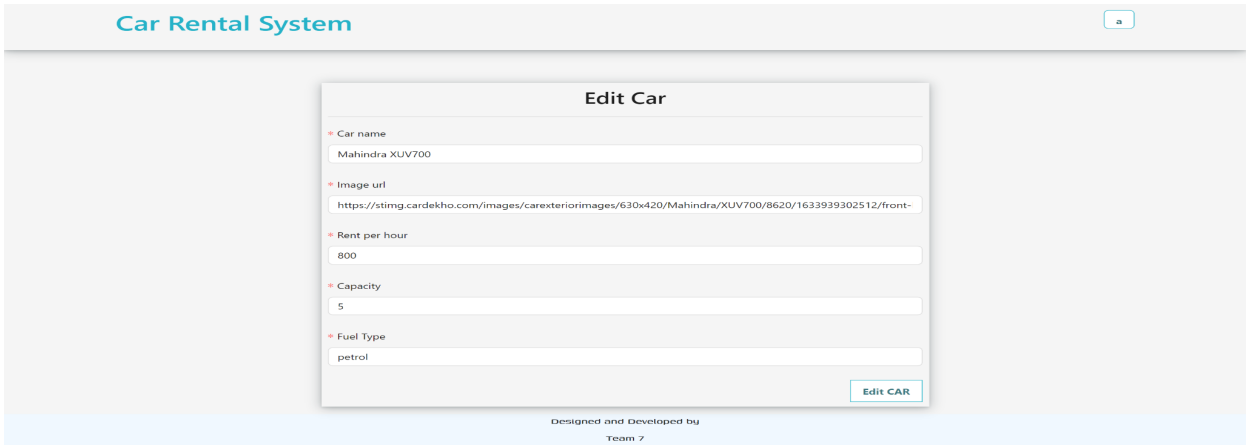
Login



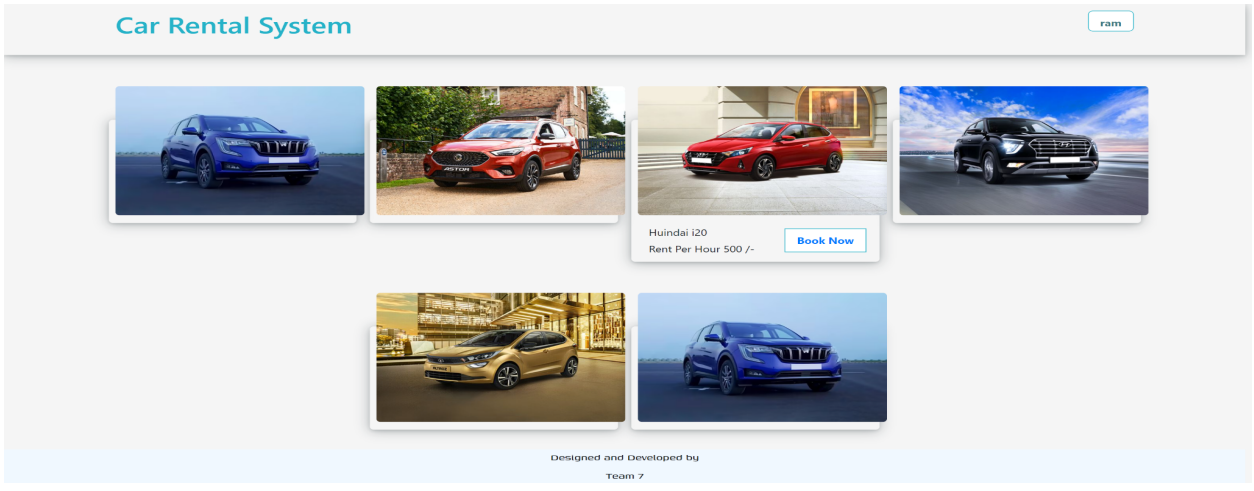
Admin Panel



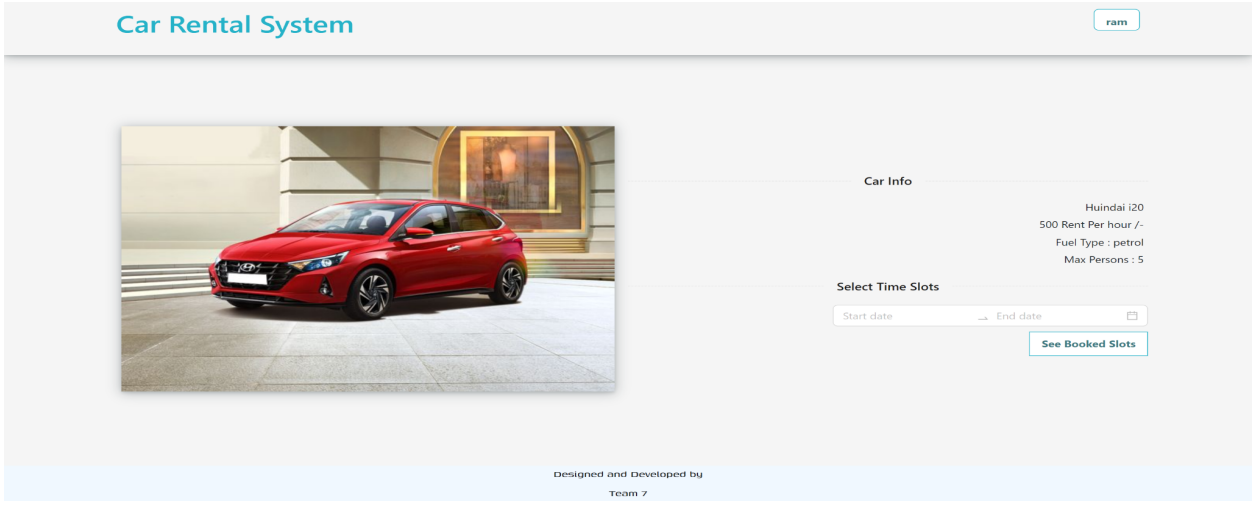
Edit Car

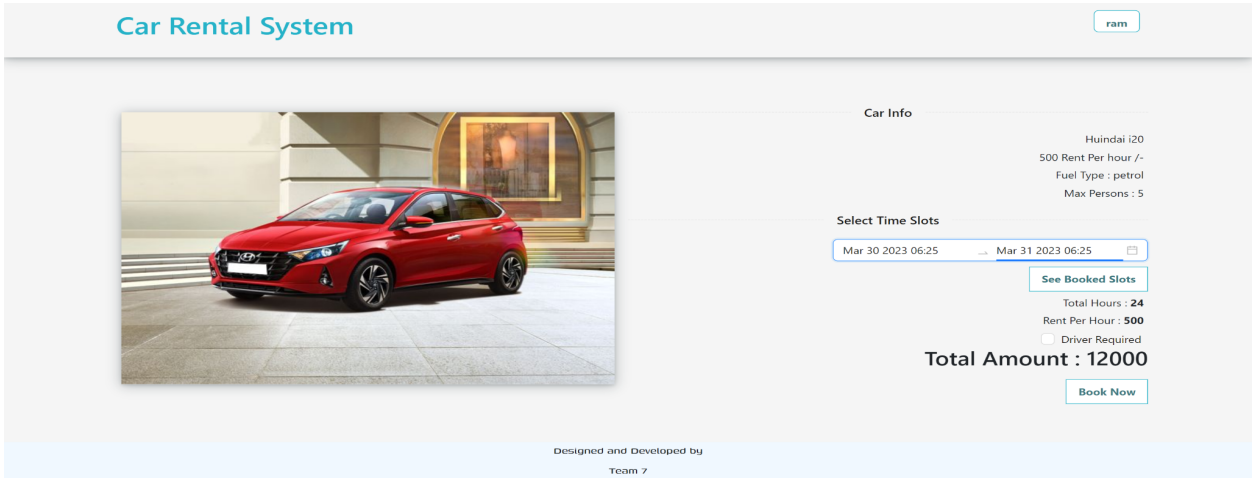


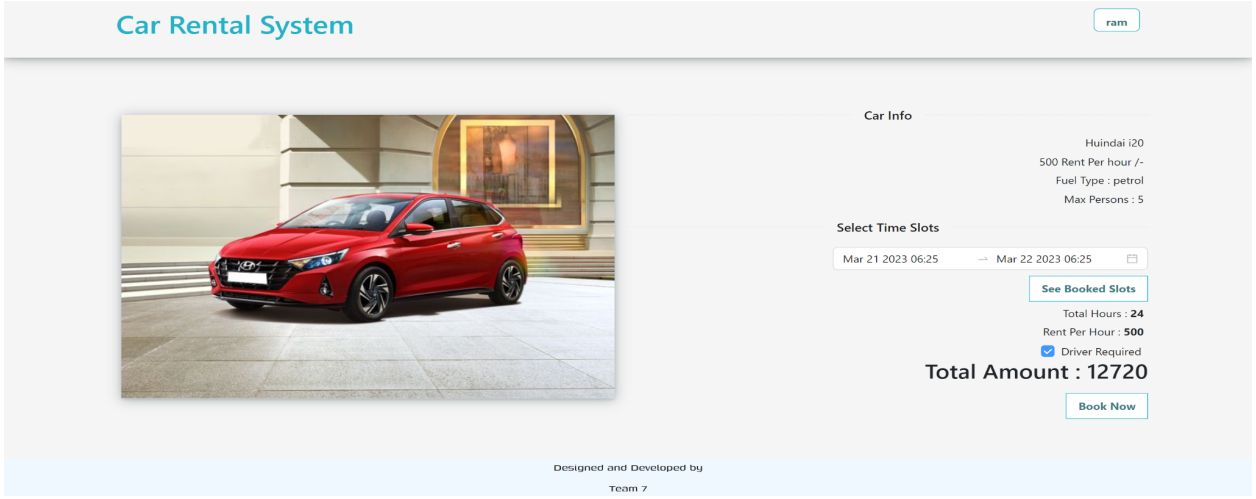
User Panel



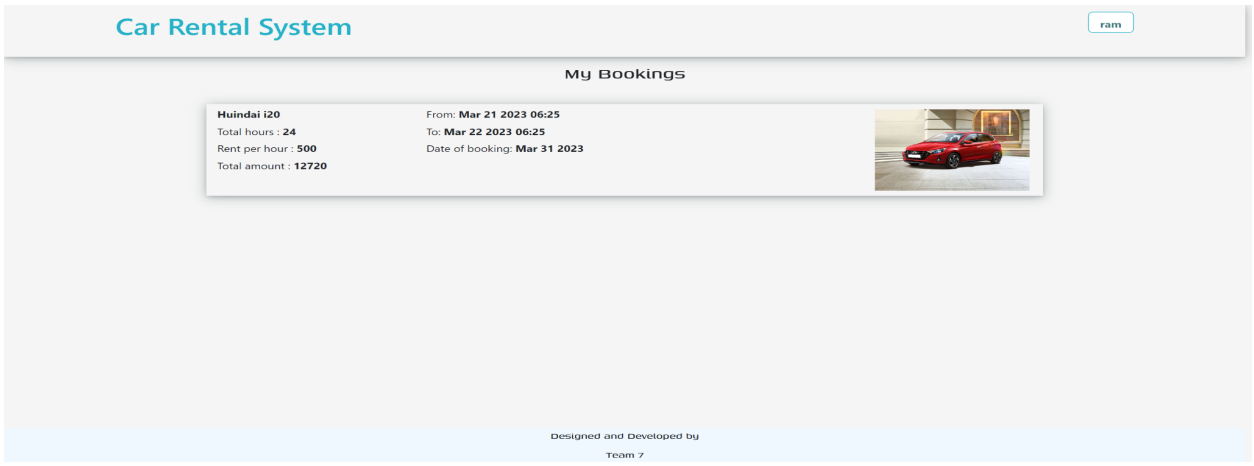
Booking


Car Rental System
ram
Car Info
Huindai i20
500 Rent Per hour /-
Fuel Type : petrol
Max Persons : 5
Select Time Slots
Start date
End date
See Booked Slots
Designed and Developed by
Team 7

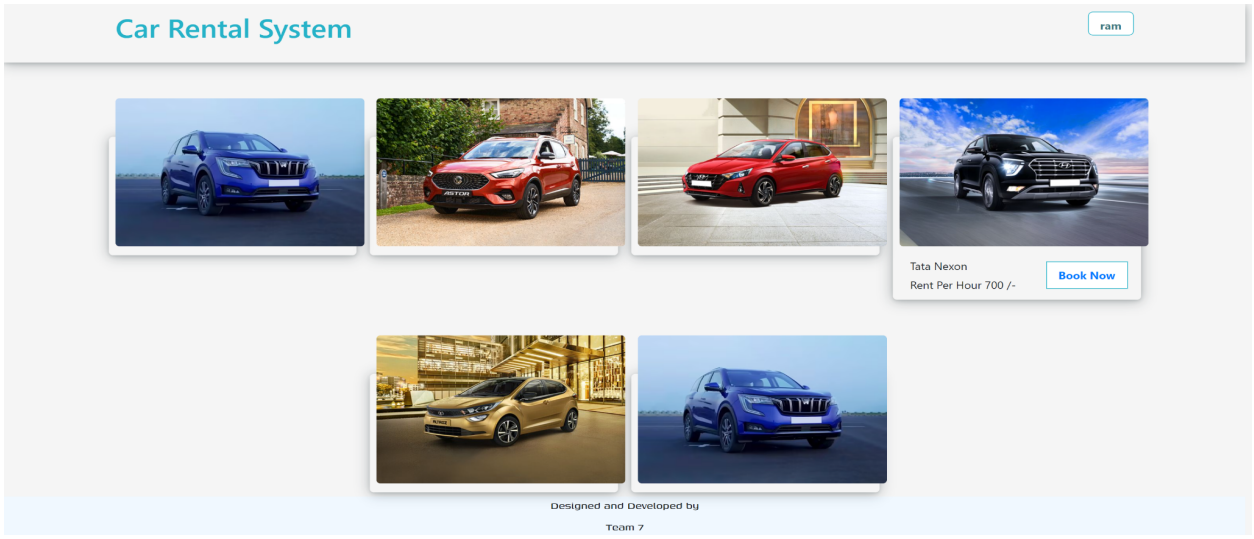

Car Rental System
ram
Car Info
Huindai i20
500 Rent Per hour /-
Fuel Type : petrol
Max Persons : 5
Select Time Slots
Mar 30 2023 06:25
Mar 31 2023 06:25
See Booked Slots
Total Hours : 24
Rent Per Hour : 500
Driver Required
Total Amount : 12000
Book Now
Designed and Developed by
Team 7

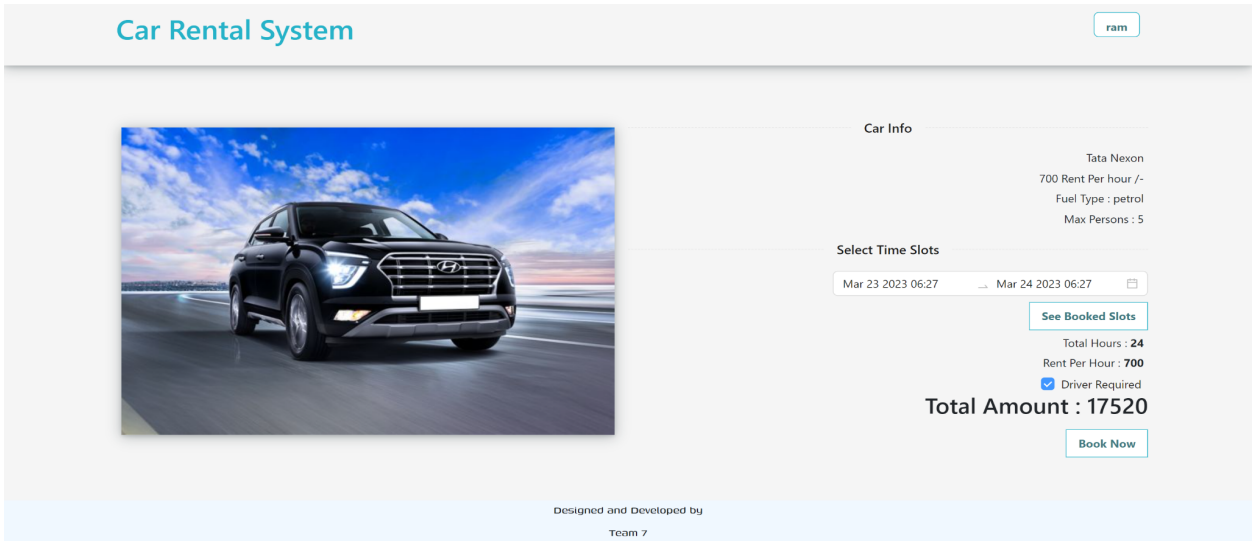

Car Rental System
ram
Car Info
Huindai i20
500 Rent Per hour /-
Fuel Type : petrol
Max Persons : 5
Select Time Slots
Mar 21 2023 06:25
Mar 22 2023 06:25
See Booked Slots
Total Hours : 24
Rent Per Hour : 500
Driver Required
Total Amount : 12720
Book Now
Designed and Developed by
Team 7

User Bookings



Booked Time slots





Car Rental System

ram

My Bookings

Tata Nexon
Total hours : **24**
Rent per hour : **700**
Total amount : **17520**

From: **Mar 23 2023 06:27**
To: **Mar 24 2023 06:27**
Date of booking: **Mar 31 2023**

Designed and Developed by
Team 7

Car Rental System

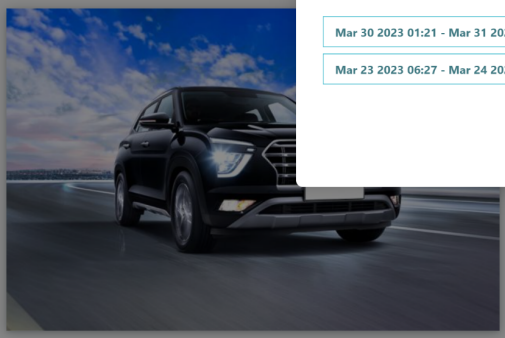
ram

Booked time slots

Mar 30 2023 01:21 - Mar 31 2023 01:21

Mar 23 2023 06:27 - Mar 24 2023 06:27

CLOSE



Car Info

Tata Nexon
700 Rent Per hour /-
Fuel Type : petrol
Max Persons : 5

Select Time Slots

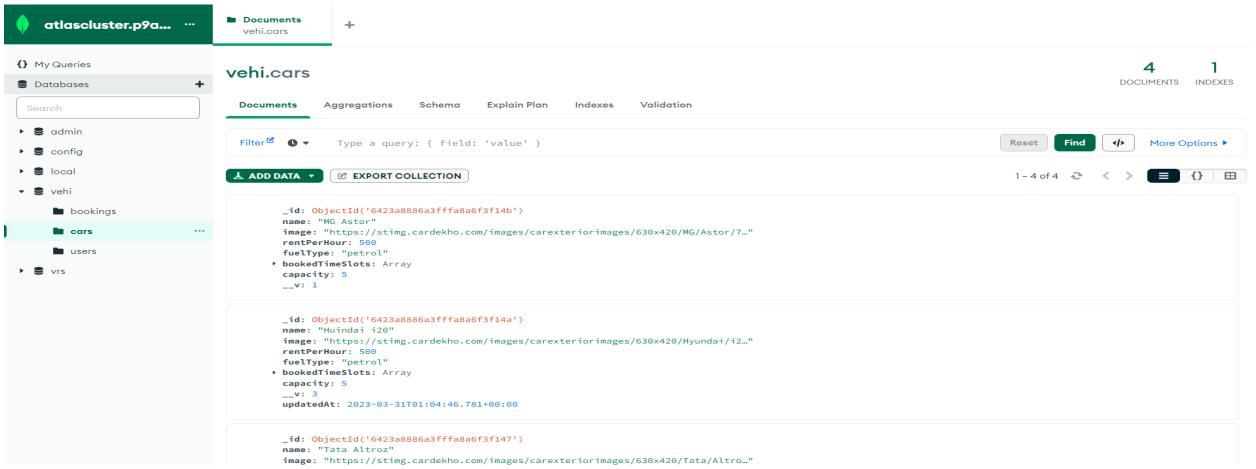
Start date — End date

See Booked Slots

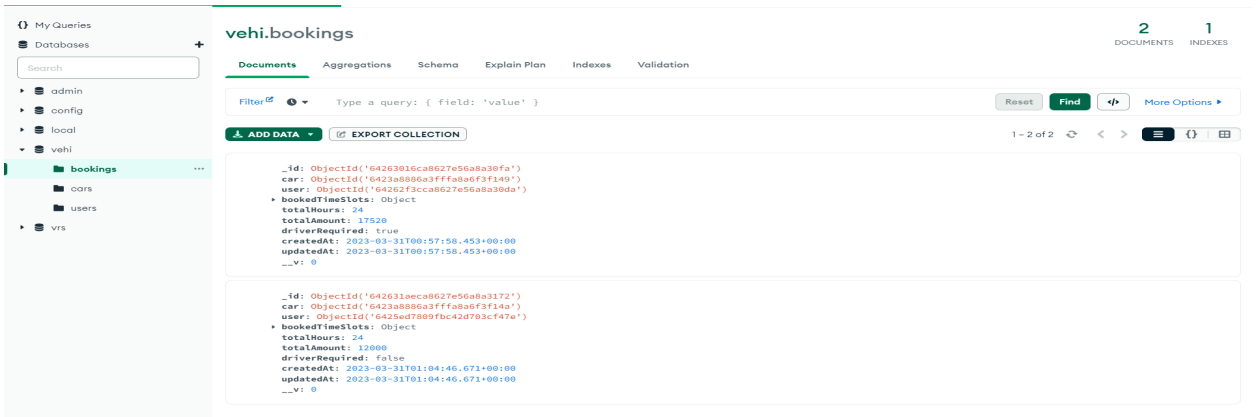
Designed and Developed by
Team 7

Database

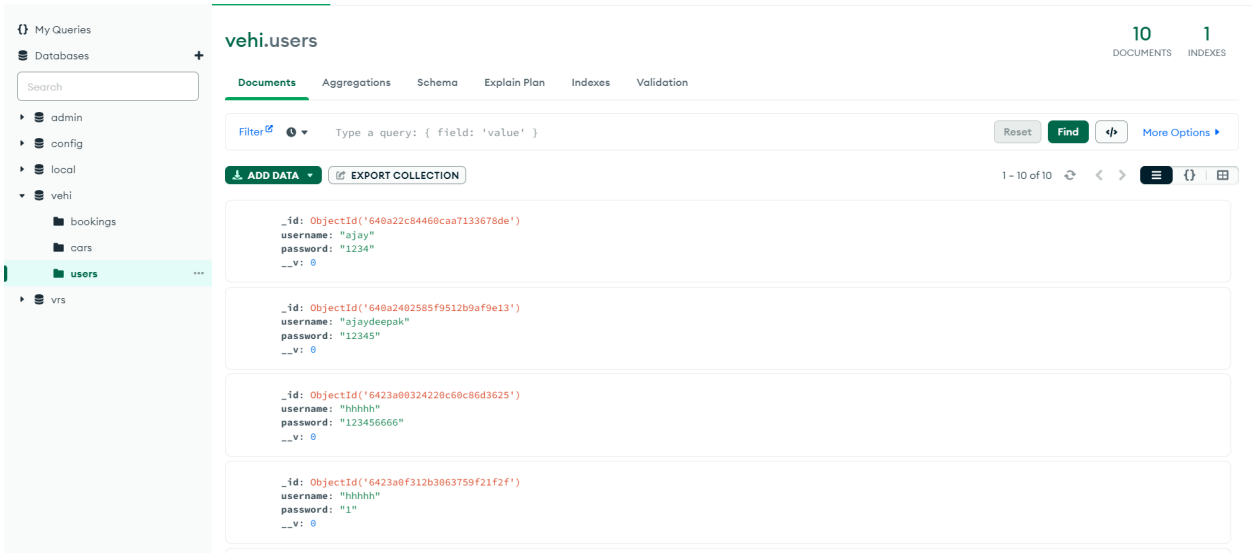
Cars



Bookings

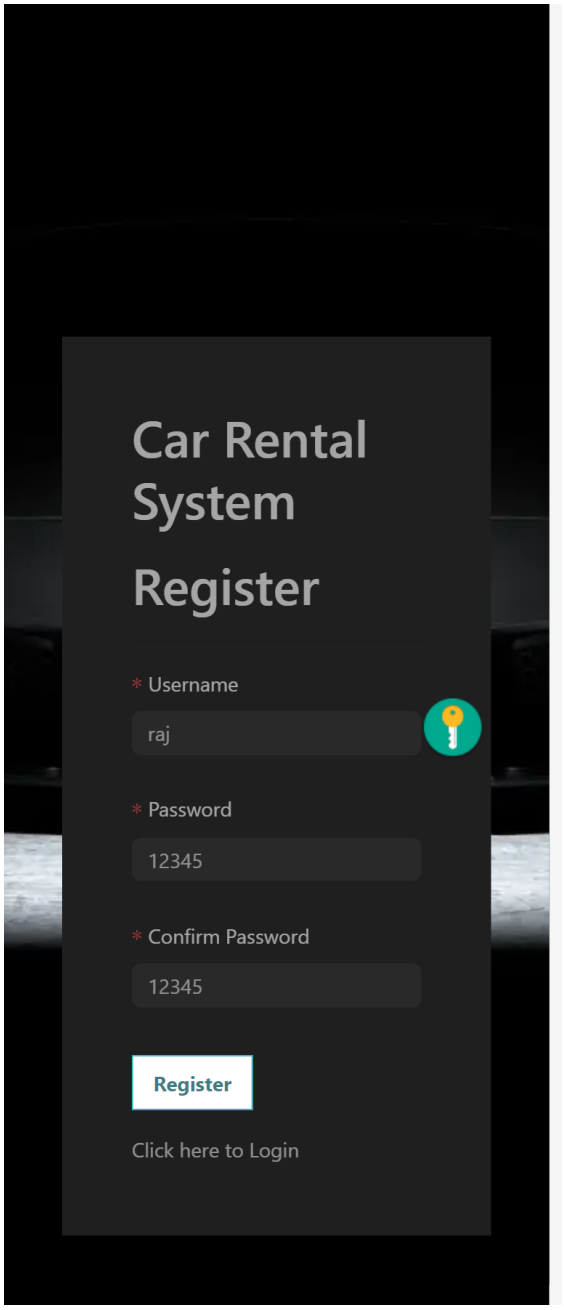


Users

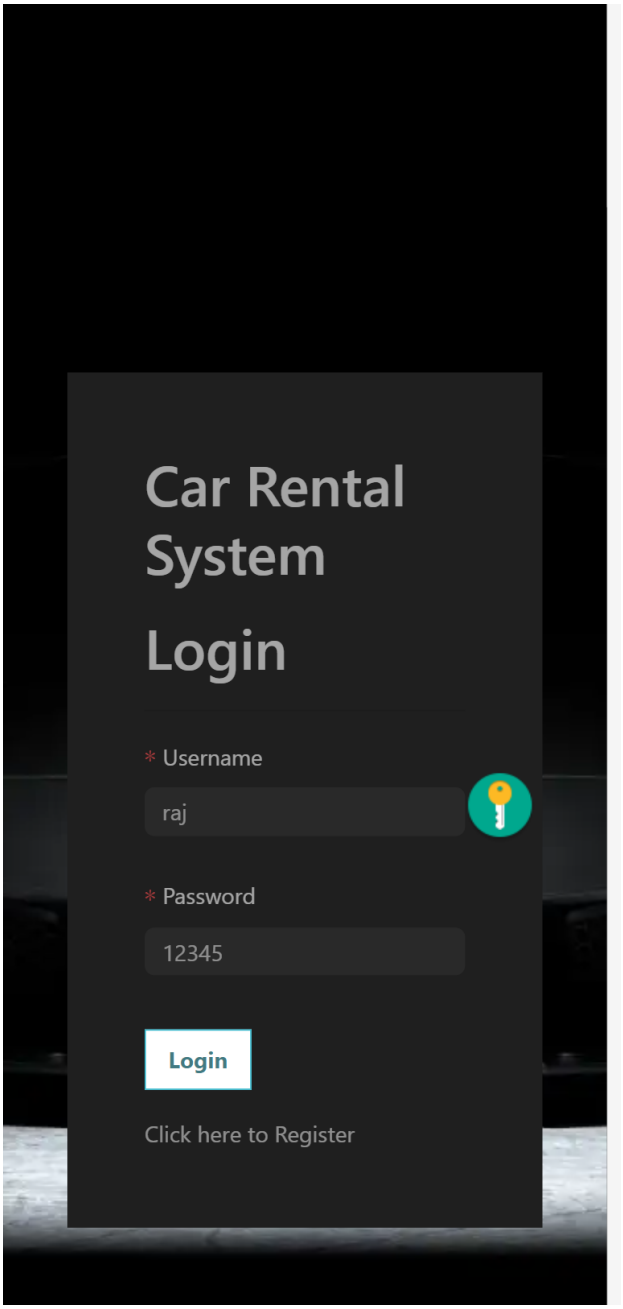


IMAGES: MOBILE APPLICATION

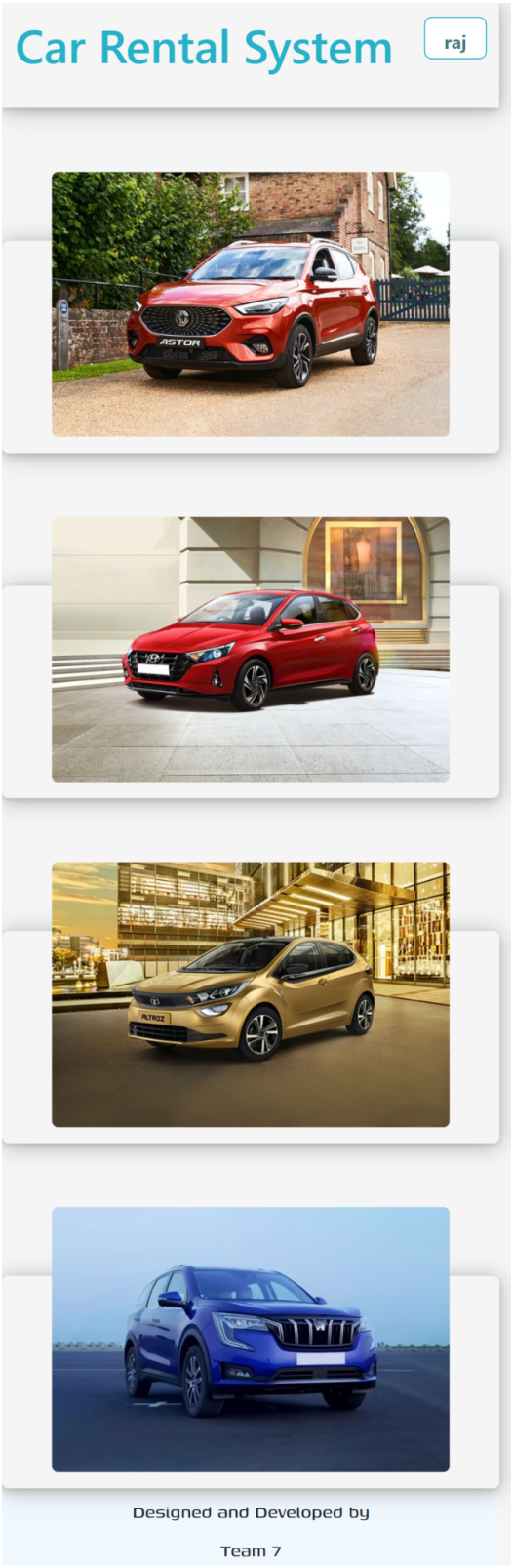
Register

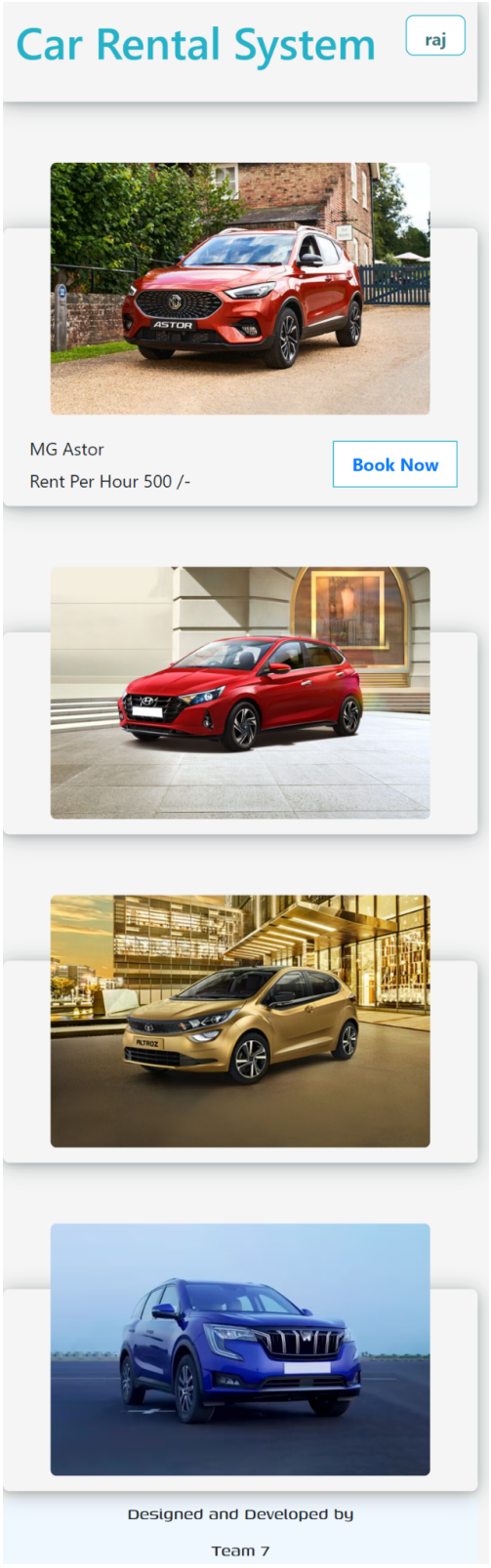


Login



User Panel


Car Rental System
raj
Designed and Developed by
Team 7


Car Rental System
raj
MG Astor
Rent Per Hour 500 /-
Book Now
Designed and Developed by
Team 7

Booking

Car Rental System

raj

Car Info

MG Astor

500 Rent Per hour /-

Fuel Type : petrol

Max Persons : 5

Select Time Slots

Mar 23 2023 06:42 — Mar 25 2023 06:42

See Booked Slots

Total Hours : **48**

Rent Per Hour : **500**

Driver Required

Total Amount : 24000

Book Now

Designed and Developed by

Team 7

My Bookings

Car Rental System

raj

My Bookings

MG Astor

Total hours : **48**

Rent per hour : **500**

Total amount : **24000**

From: **Mar 23 2023 06:42**

To: **Mar 25 2023 06:42**

Date of booking: **Mar 31 2023**

Designed and Developed by

Team 7

Booked Time Slots