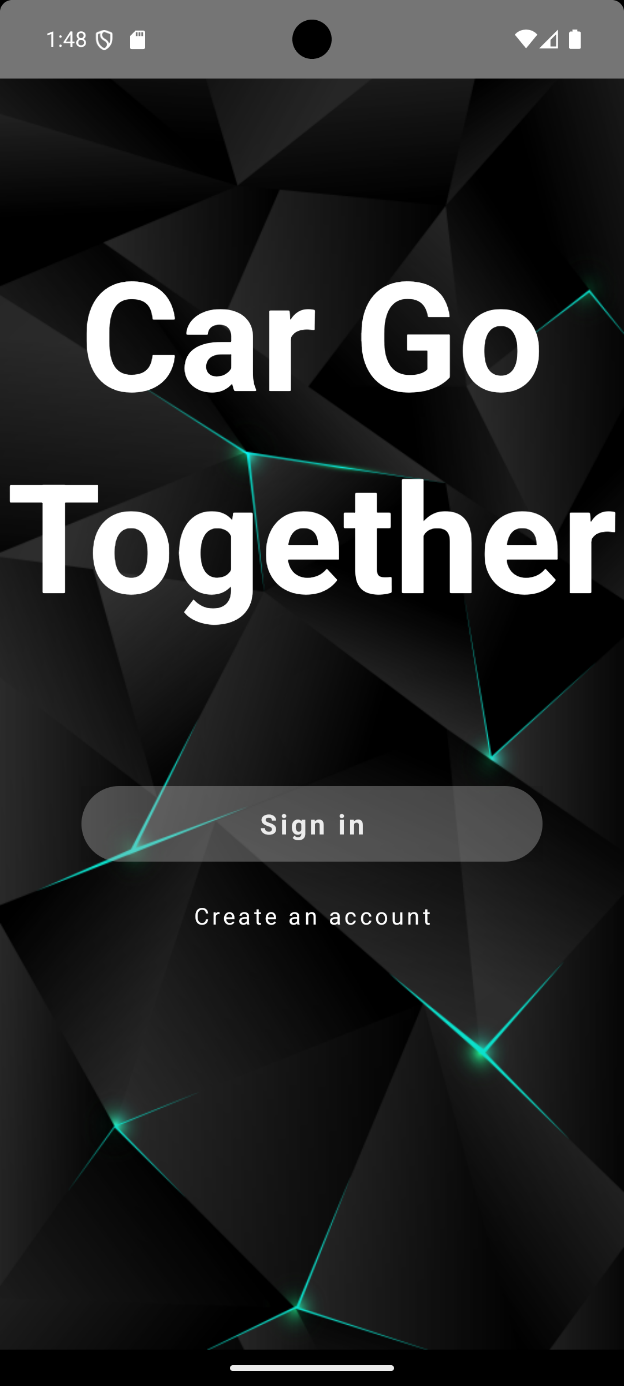
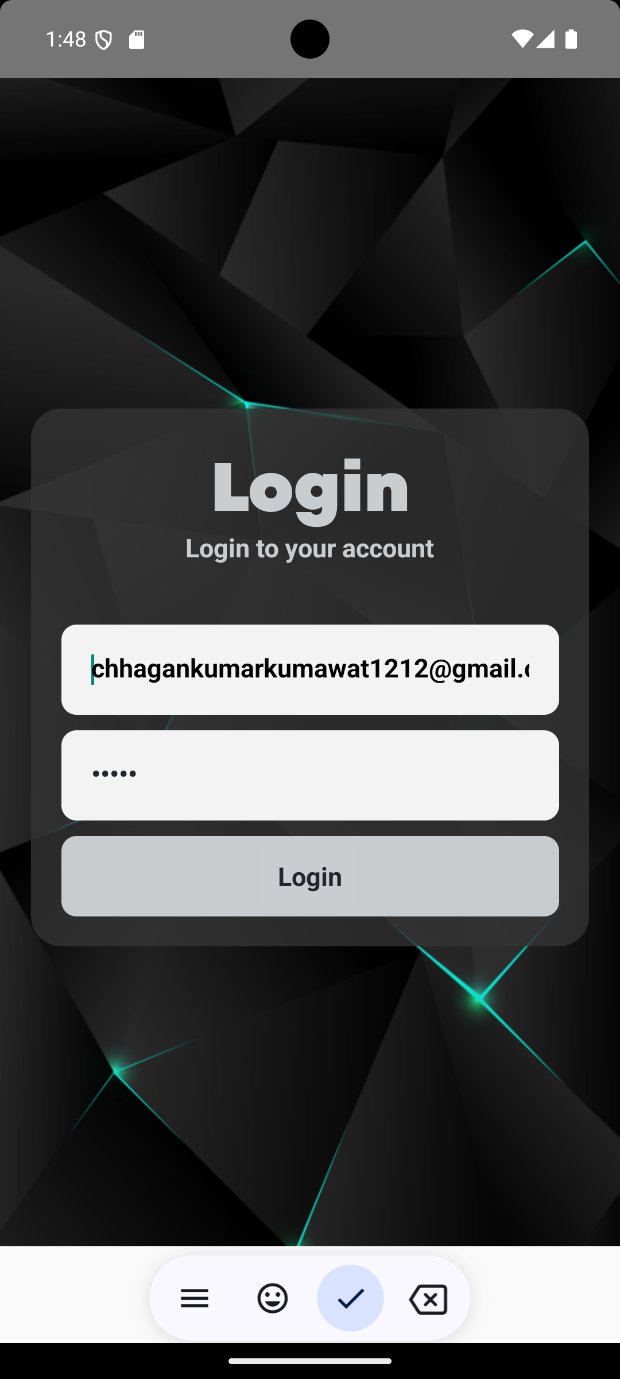
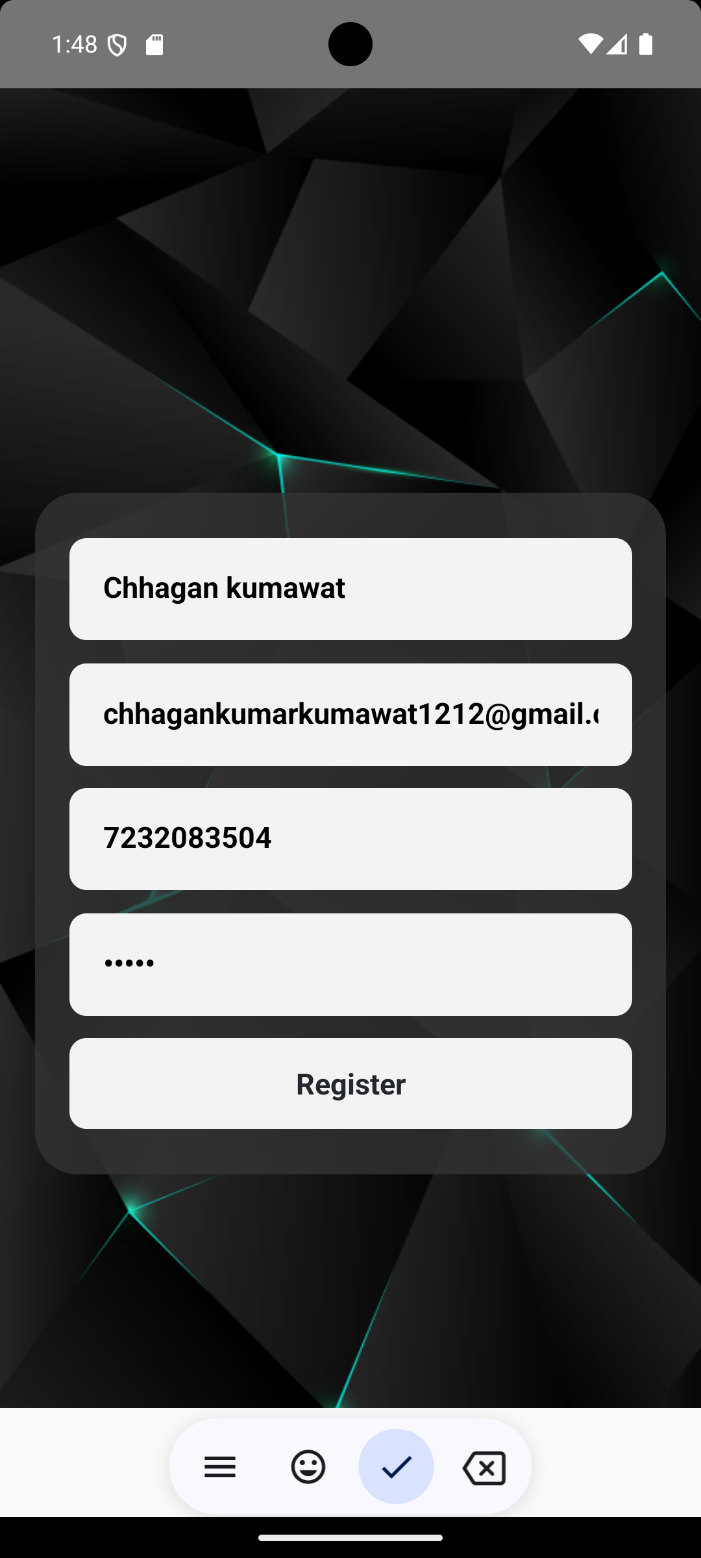
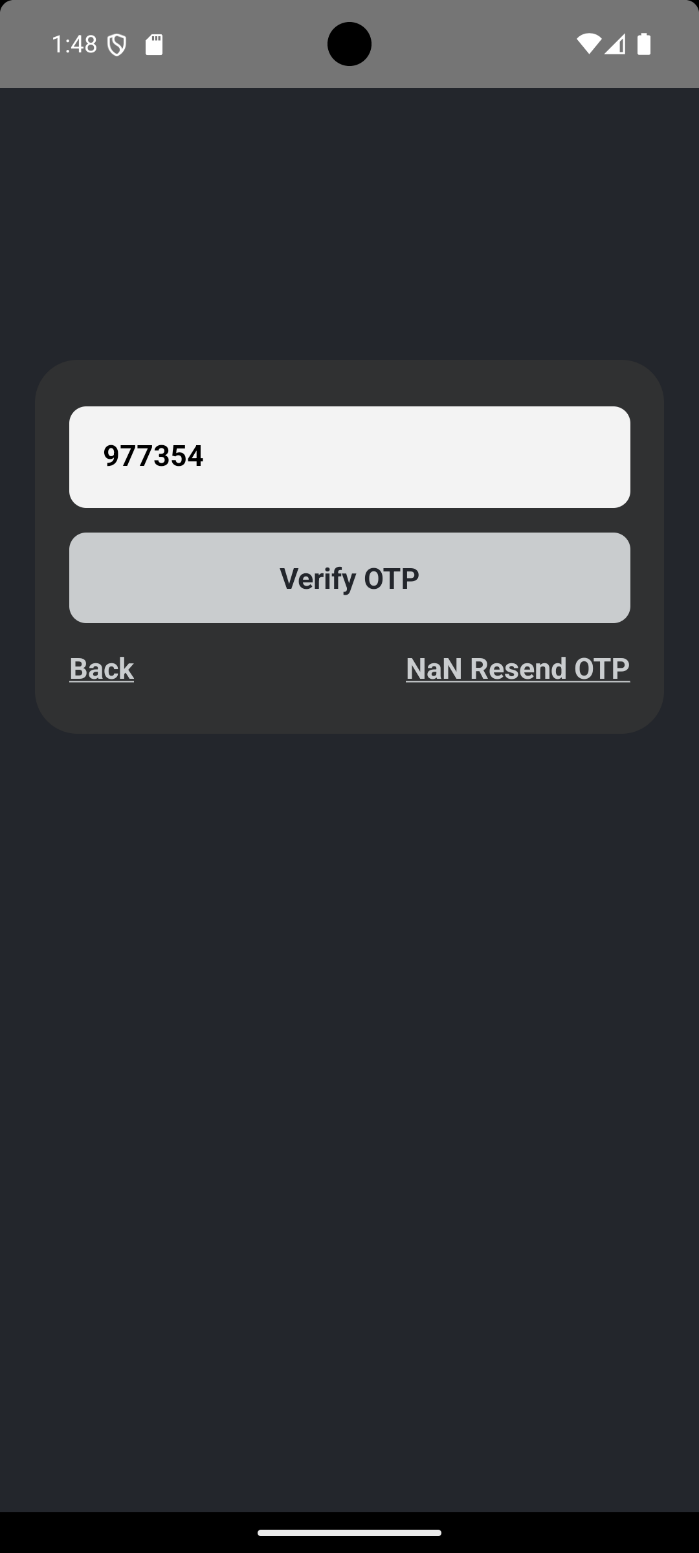
**CHAPTER 4 : USER MANUAL**

**4.1 User Interface Screens (Input)**

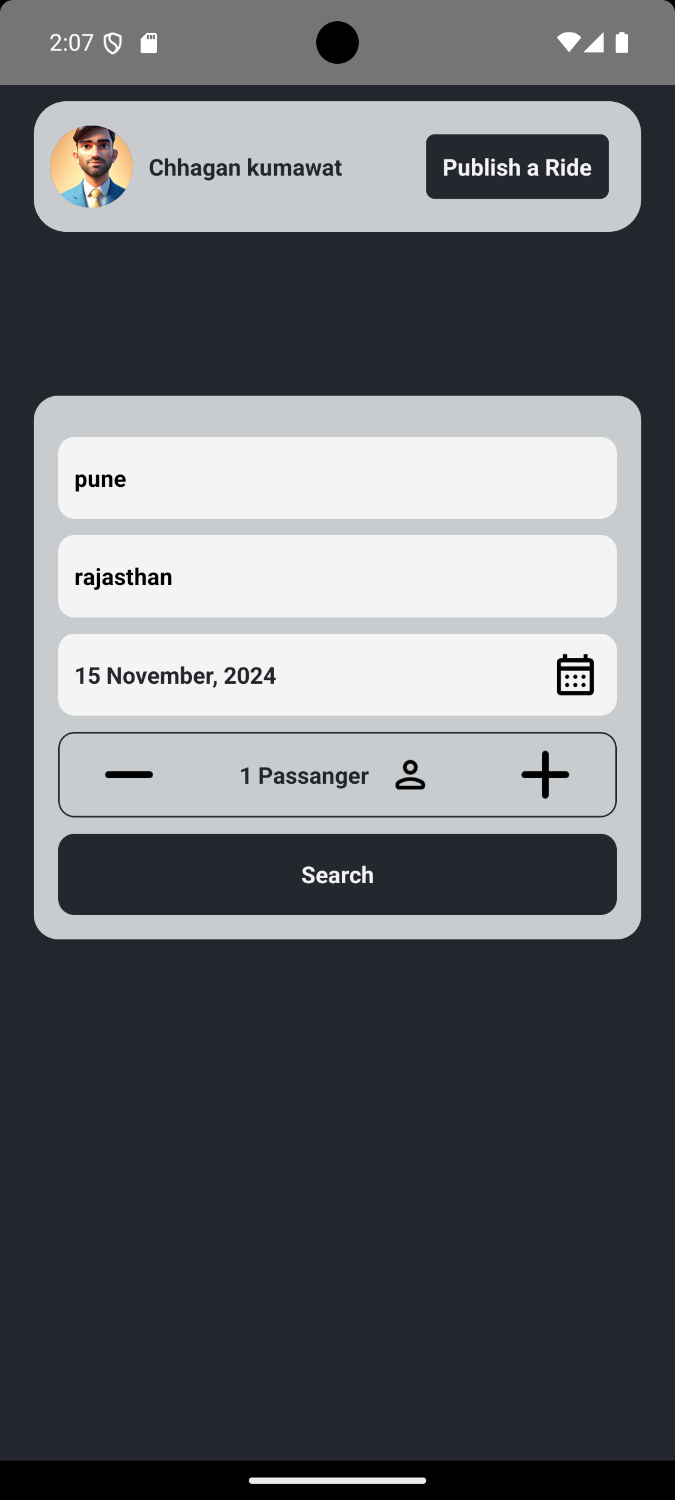
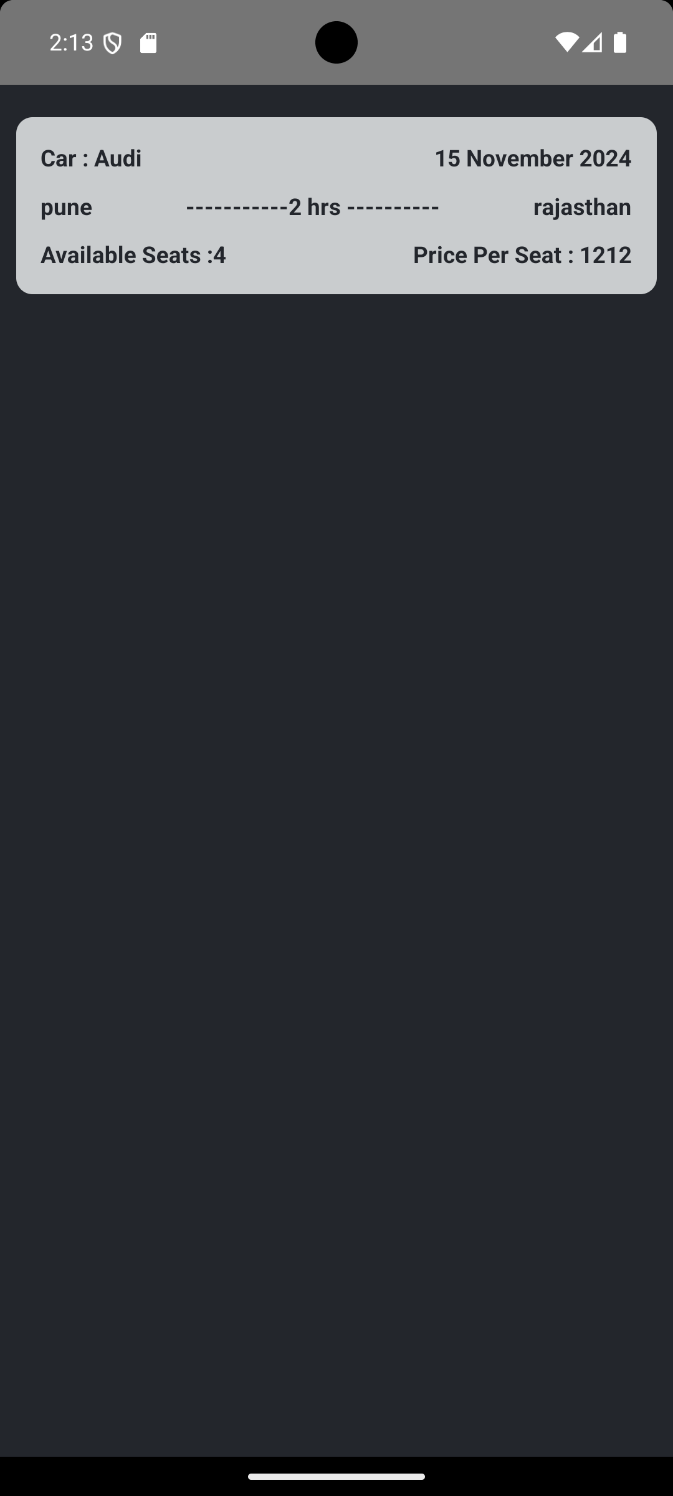
**A. Home Screen & Login Screen:**



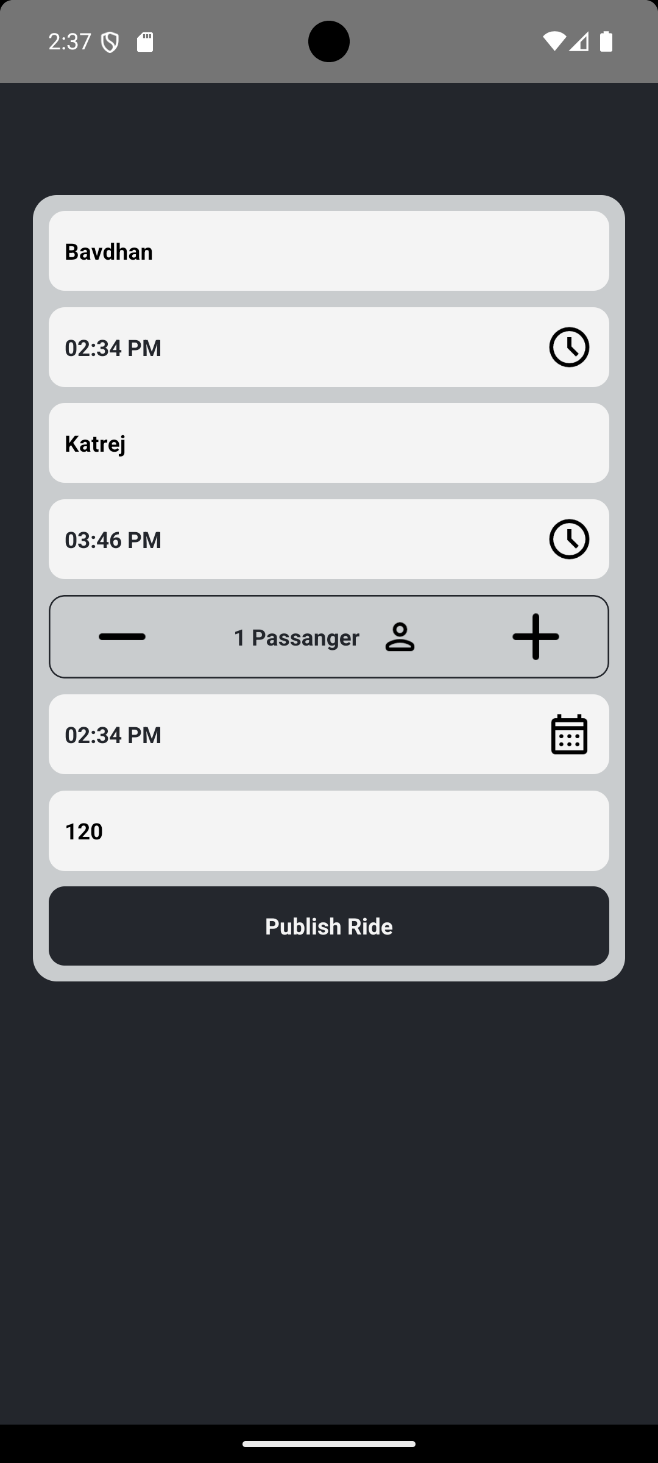
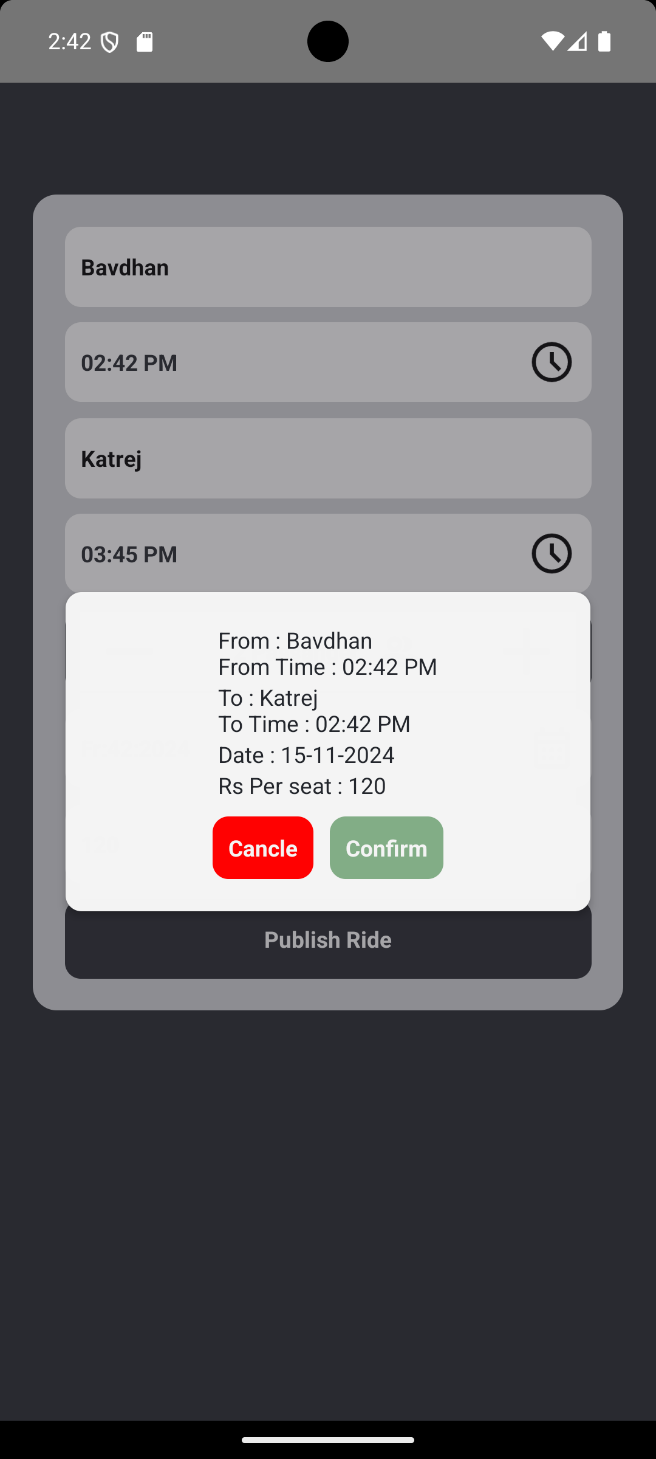
**B. Registration screen & OTP Generation:**

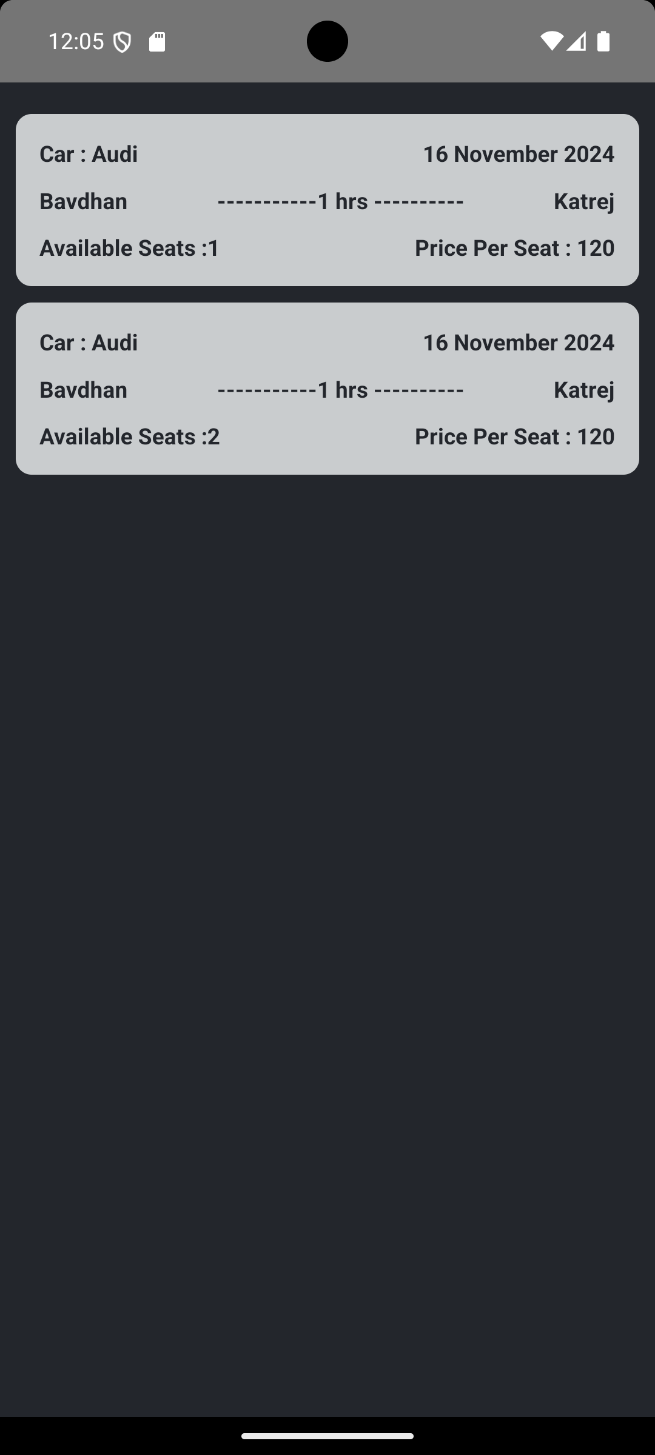


**C. Dashboard And Booking List:**



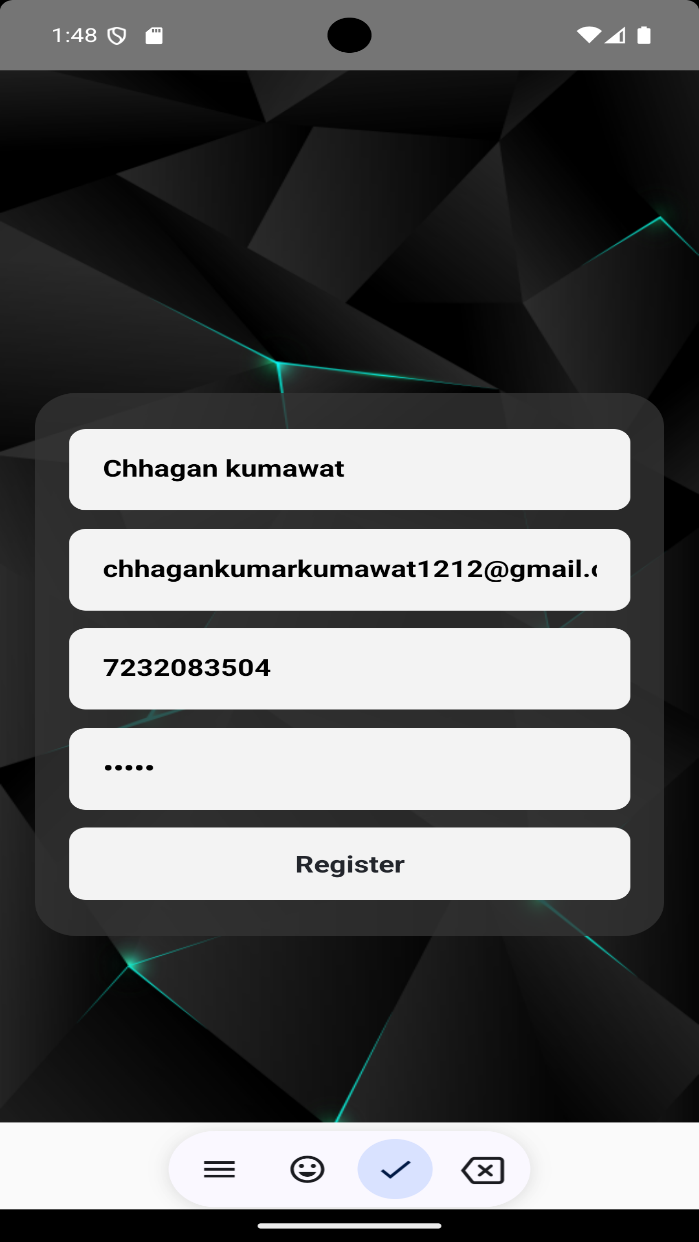
**D. Create Ride & Ride List:**





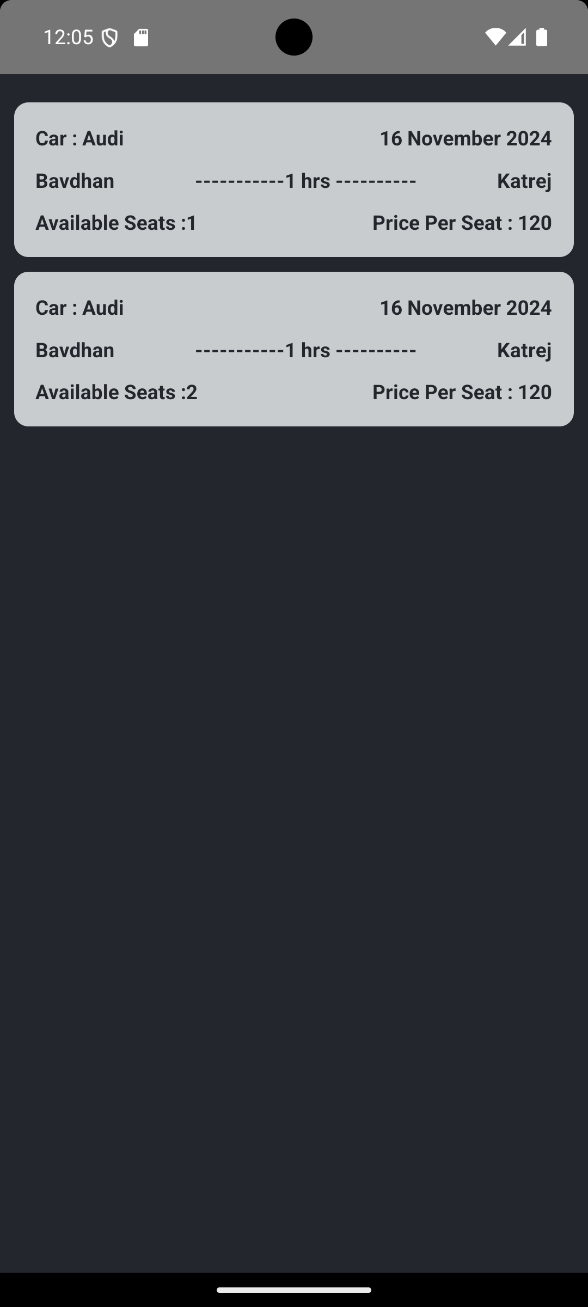
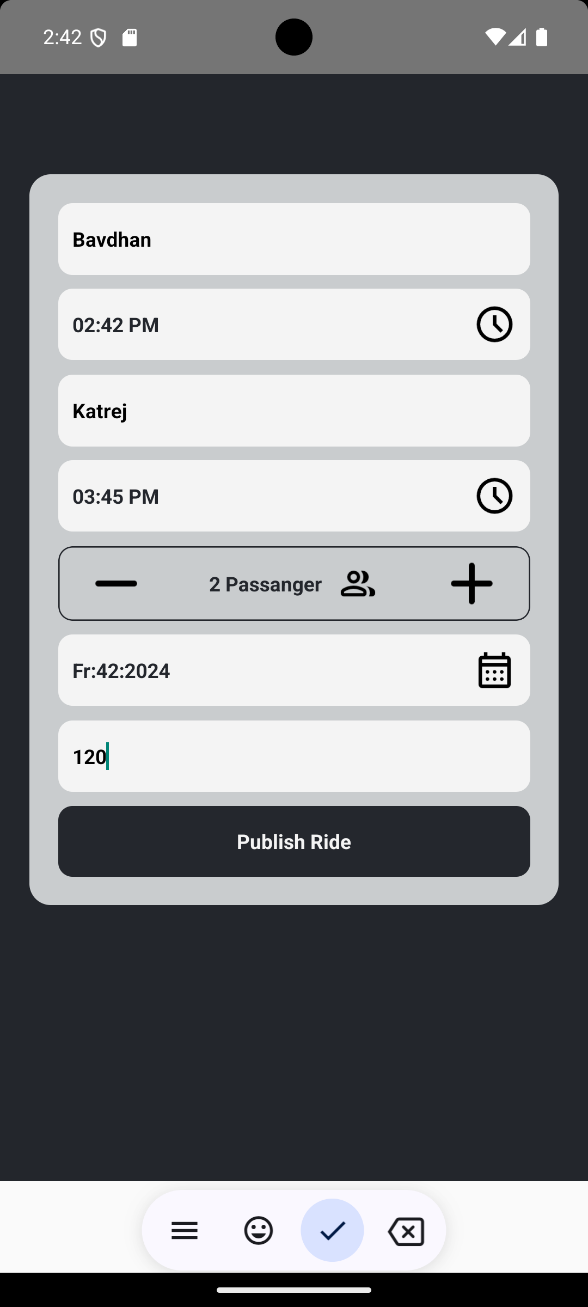
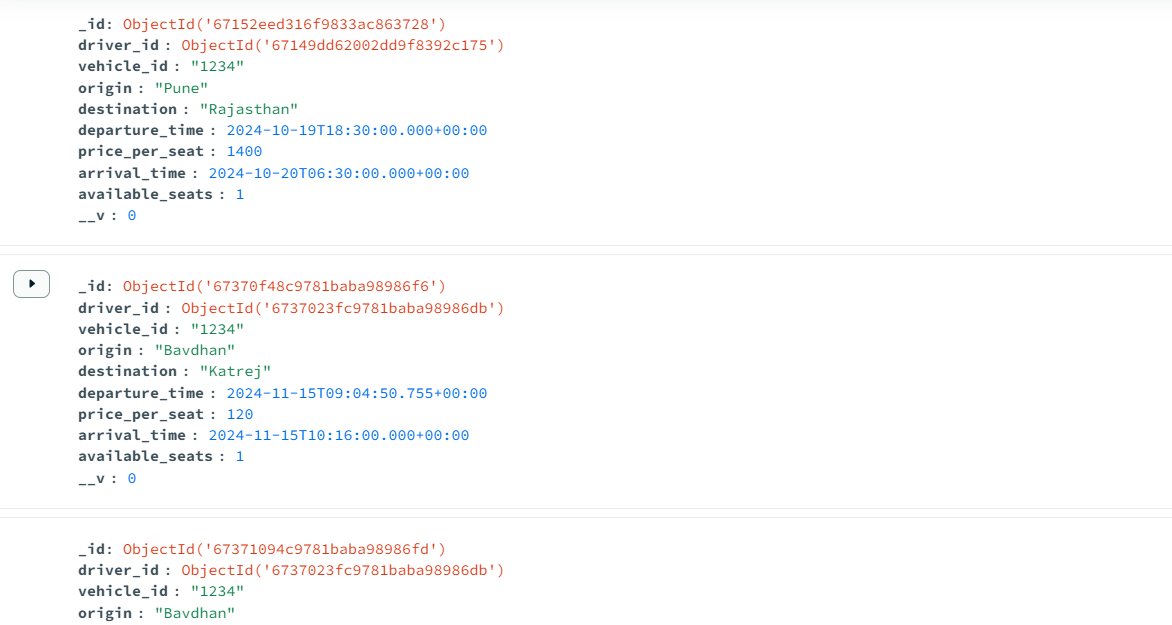
**4.2 Output Screens with data :**

**A. Registration Screen:**





**B. Create Ride Screen:**

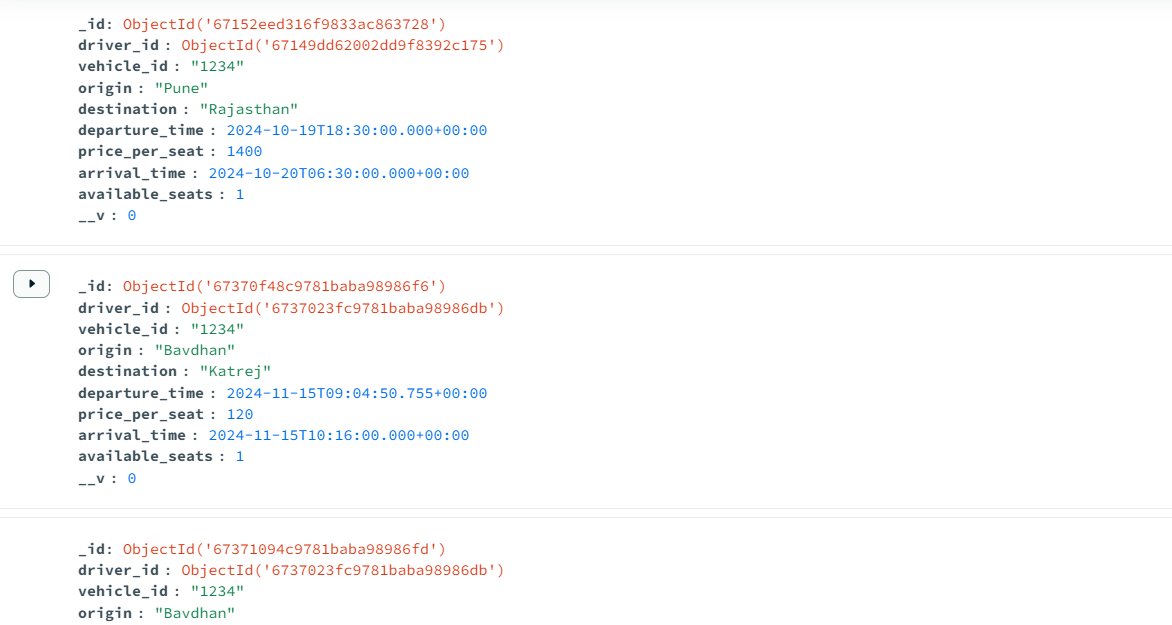


**4.3 Data Reports:**

**A. Registration Data:**



**B. Rides List:**



**4.4 Test Procedures and cases:**

**A. User Registration Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Input Data** | **Expected Result** | **Pass/Fail** |
| TC001 | Validate empty username | Username: "" | Error message: "Username is required." | Fail |
| TC002 | Validate special characters | Username: "User@123" | Error message: "Username cannot contain special characters." | Fail |
| TC003 | Validate valid username | Username: "JohnDoe" | User registration proceeds successfully. | Pass |
| TC004 | Validate empty email | Email: "" | Error message: "Email is required." | Fail |
| TC005 | Validate invalid email format | Email: "userexample.com" | Error message: "Enter a valid email address." | Fail |
| |  | | --- | | TC006 |  |  | | --- | |  | | Validate valid email | Email: "chhagankumarkumawat1212@gmail.com" | User registration proceeds successfully. | Pass |
| TC007 | Validate empty mobile number | Mobile Number: "" | Error message: "Mobile number is required." | Fail |
| TC00  8 | Validate invalid mobile number | Mobile Number: "12345" | Error message: "Enter a valid mobile number." | Fail |
| TC009 | Validate valid mobile number | Mobile Number: "7232083504" | User registration proceeds successfully. | Pass |
| TC010 | Validate empty password | Password: "" | Error message: "Password is required." | Fail |
| TC011 | Validate password length | Password: "123" | Error message: "Password must be at least 8 characters long." | Fail |
| TC012 | Validate valid password | Password: "12345" | |  | | --- | |  |  |  | | --- | | User registration proceeds successfully. | | Pass |

**B. User Login Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Input Data** | **Expected Result** | **Pass/Fail** |
| TC001 | Validate empty email | Email: "", Password: "P@ssw0rd" | Error message: "Email is required." | Fail |
| TC002 | Validate empty password | Email: " chhagankumarkumawat1212@gmail.com ", Password: "" | Error message: "Password is required." | Fail |
| TC003 | Validate both fields empty | Email: "", Password: "" | Error message: "Email and Password are required." | Fail |
| TC004 | Validate invalid email format | Email: " chhagankumarkumawat1212@gmail.com ", Password: "P@ssw0rd" | Error message: "Enter a valid email address." | Fail |
| TC005 | Validate email not registered | Email: " chhagankumarkumawat1212@gmail.com ", Password: "P@ssw0rd" | Error message: "Email not found. Please register." | Fail |
| TC006 | Validate incorrect password | Email: " chhagankumarkumawat1212@gmail.com ", Password: "wrongpassword" | Error message: "Incorrect password. Please try again." | Fail |
| TC007 | Validate valid login | Email: " chhagankumarkumawat1212@gmail.com ", Password: "123456" | User successfully logged in. | Pass |
| TC008 | Validate case sensitivity in email | Email: " chhagankumarkumawat1212@gmail.com ", Password: "123456" | User successfully logged in (if case-insensitive match). | Pass |
| TC009 | Validate SQL injection attempt | Email: " chhagankumarkumawat1212@gmail.com ", Password: "' OR '1'='1" | Error message: "Invalid credentials." | Fail |
| TC010 | Validate XSS attack in email | Email: "<script>alert('XSS')</script>", Password: "P@ssw0rd" | Error message: "Invalid email address." | Fail |
| TC011 | Validate locked account after failures | Email: " chhagankumarkumawat1212@gmail.com ", Password: "wrongpassword" (3+ attempts) | Error message: "Account locked due to multiple failed attempts. Please reset your password." | Fail |
| TC012 | Validate login with inactive account | Email: " chhagankumarkumawat1212@gmail.com ", Password: "P@ssw0rd" | Error message: "Account is inactive. Please contact support." | Fail |

**C. OTP Test Cases :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Input Data** | **Expected Result** | **Pass/Fail** |
| TC001 | Validate OTP field empty | Email: " chhagankumarkumawat1212@gmail.com ", OTP: "" | Error message: "OTP is required." | Fail |
| TC002 | Validate incorrect OTP | Email: " chhagankumarkumawat1212@gmail.com ", OTP: "123456" | Error message: "Invalid OTP. Please try again." | Fail |
| TC003 | Validate expired OTP | Email: " chhagankumarkumawat1212@gmail.com ", OTP: "789456" (expired) | Error message: "OTP has expired. Please request a new one." | Fail |
| TC004 | Validate valid OTP | Email: " chhagankumarkumawat1212@gmail.com ", OTP: "654321" | User successfully logs in. | Pass |
| TC005 | Validate OTP length | Email: "user@example.com", OTP: "12345" | Error message: "OTP must be 6 digits." | Fail |
| TC006 | Validate special characters in OTP | Email: " chhagankumarkumawat1212@gmail.com ", OTP: "12@#45" | Error message: "OTP must contain only numbers." | Fail |
| TC007 | Validate multiple incorrect attempts | Email: " chhagankumarkumawat1212@gmail.com ", OTP: "111111" (3+ attempts) | Error message: "Too many incorrect attempts. OTP disabled." | Fail |
| TC008 | Validate OTP request for unregistered email | Email: " chhagankumAarkumawat12@gmail.com ", OTP: "123456" | Error message: "Email not found. Please register first." | Fail |
| TC009 | Validate OTP delivery for valid email | Email: " chhagankumarkumawat1212@gmail.com " (Request OTP) | Success message: "OTP sent to your email." | Pass |
| TC010 | Validate OTP delivery failure | Email: "invalidemail@example.com" | Error message: "Failed to send OTP. Check email address." | Fail |
| TC011 | Validate login bypassing OTP | Email: " chhagankumarkumawat1212@gmail.com ", No OTP entered | Error message: "OTP verification is required to log in." | Fail |
| TC012 | Validate resend OTP functionality | Email: " chhagankumarkumawat1212@gmail.com " (Request resend) | Success message: "New OTP sent to your email." | Pass |

**4.5 Sample program code:**

**A. Home page**

**import React, { useEffect } from "react";**

**import {**

**Image,**

**ImageBackground,**

**StyleSheet,**

**Text,**

**Touchable,**

**TouchableOpacity,**

**View,**

**} from "react-native";**

**import { Button } from "@rneui/themed";**

**import bg from "../assets/images/BG-01.jpg";**

**import { useDispatch, useSelector } from "react-redux";**

**function Home(props) {**

**const count = useSelector((state) => state.loginData);**

**const dispatch = useDispatch();**

**useEffect(() => {**

**console.log(count);**

**}, []);**

**return (**

**<View style={styles.container}>**

**<ImageBackground source={bg} resizeMode="cover" style={styles.image}>**

**<View**

**style={{**

**alignItems: "center",**

**marginTop: 100,**

**}}**

**>**

**<Text**

**style={{**

**fontSize: 100,**

**color: "#ffffff",**

**fontWeight: "bold",**

**}}**

**>**

**Car Go**

**</Text>**

**<Text**

**style={{**

**fontSize: 100,**

**color: "#ffffff",**

**fontWeight: "bold",**

**}}**

**>**

**Together**

**</Text>**

**</View>**

**<TouchableOpacity**

**style={{**

**backgroundColor: "#e2e2e254",**

**width: 304,**

**height: 50,**

**borderRadius: 40,**

**marginTop: 100,**

**marginLeft: "auto",**

**marginRight: "auto",**

**alignItems: "center",**

**justifyContent: "center",**

**}}**

**onPress={() => {**

**props.navigation.navigate("Login");**

**}}**

**>**

**<Text**

**style={{**

**color: "#ffffff",**

**fontSize: 18,**

**letterSpacing: 2,**

**fontWeight: "bold",**

**}}**

**>**

**Sign in**

**</Text>**

**</TouchableOpacity>**

**<TouchableOpacity**

**style={{**

**marginTop: 25,**

**marginLeft: "auto",**

**marginRight: "auto",**

**alignItems: "center",**

**justifyContent: "center",**

**}}**

**onPress={() => {**

**props.navigation.navigate("Register");**

**}}**

**>**

**<Text**

**style={{**

**color: "#ffffff",**

**fontSize: 15,**

**letterSpacing: 2,**

**}}**

**>**

**Create an account**

**</Text>**

**</TouchableOpacity>**

**</ImageBackground>**

**</View>**

**);**

**}**

**const styles = StyleSheet.create({**

**container: {**

**flex: 1,**

**},**

**button: {**

**flex: 1,**

**marginTop: 30,**

**marginLeft: "auto",**

**marginRight: "auto",**

**width: 304,**

**borderRadius: 10,**

**},**

**signUp: {**

**marginTop: 30,**

**marginLeft: "auto",**

**marginRight: "auto",**

**width: 10,**

**height: 50,**

**backgroundColor: "rgba(5,70,82,1)",**

**borderRadius: 10,**

**},**

**Text: {**

**fontFamily: "helvetica-regular",**

**fontWeight: "bold",**

**color: "rgba(5,70,82,1)",**

**width: 304,**

**fontSize: 25,**

**textAlign: "center",**

**marginTop: 30,**

**marginLeft: "auto",**

**marginRight: "auto",**

**},**

**image: {**

**flex: 1,**

**width: "100%",**

**height: "100%",**

**display: "flex",**

**},**

**});**

**export default Home;**

**B. Registration page:**

**import {**

**View,**

**Text,**

**StyleSheet,**

**TouchableOpacity,**

**TextInput,**

**ImageBackground,**

**Image,**

**} from "react-native";**

**import React, { useState } from "react";**

**import { PostRequestCall } from "../apicall/PostReq";**

**import { black, lWhite, white } from "./Color";**

**import { useDispatch, useSelector } from "react-redux";**

**import { setEmailRedux } from "../redux/features/data/DataSlice";**

**import bg from "../assets/images/BG-01.jpg";**

**export default function Register(props) {**

**const dispatch = useDispatch();**

**const Remail = useSelector((state) => state.emailData.value);**

**const [fullName, setFullName] = useState("");**

**const [email, setEmail] = useState("");**

**const [phone, setPhone] = useState("");**

**const [password, setPassword] = useState("");**

**const [toggleOTP, setToggleOTP] = useState(true);**

**const [age, setAge] = useState("20");**

**const [otp, setOTP] = useState(null);**

**const handleRegister = async () => {**

**let data = JSON.stringify({**

**name: fullName,**

**email: email,**

**phone\_number: phone,**

**password: password,**

**});**

**console.log(data);**

**dispatch(setEmailRedux(email));**

**const res = await PostRequestCall("api/auth/register", data, "", false);**

**console.log(JSON.stringify(res, null, 2));**

**if (res.status === 201) {**

**props.navigation.navigate("verifyOTP");**

**}**

**};**

**return (**

**<View style={styles.container}>**

**<ImageBackground source={bg} resizeMode="cover" style={styles.image}>**

**{/\* <View**

**style={{**

**backgroundColor: white,**

**flex: 1,**

**alignItems: "center",**

**}}**

**> \*/}**

**<View**

**style={{**

**//   height: 400,**

**width: "90%",**

**backgroundColor: 'rgba(52, 52, 52, 0.8)',**

**marginTop: 100,**

**borderRadius: 25,**

**padding: 20,**

**// paddingTop: 10,**

**// paddingBottom: 10,**

**// paddingLeft: 20,**

**// paddingRight: 20,**

**flexDirection: "column",**

**justifyContent: "space-around",**

**height: 400,**

**}}**

**>**

**<TextInput**

**placeholderTextColor={black}**

**placeholder="Full Name"**

**onChangeText={(text) => {**

**setFullName(text);**

**}}**

**style={styles.TextInput}**

**></TextInput>**

**<TextInput**

**placeholderTextColor={black}**

**placeholder="Email"**

**onChangeText={(email) => {**

**setEmail(email);**

**}}**

**style={styles.TextInput}**

**></TextInput>**

**<TextInput**

**placeholderTextColor={black}**

**placeholder="Phone"**

**onChangeText={(text) => {**

**setPhone(text);**

**}}**

**style={styles.TextInput}**

**></TextInput>**

**<TextInput**

**placeholderTextColor={black}**

**placeholder="Password"**

**secureTextEntry={true}**

**onChangeText={(text) => {**

**setPassword(text);**

**}}**

**style={styles.TextInput}**

**></TextInput>**

**<TouchableOpacity**

**onPress={() => {**

**handleRegister();**

**}}**

**>**

**<Text**

**style={{**

**backgroundColor: white,**

**padding: 15,**

**borderRadius: 10,**

**// marginTop: 10,**

**textAlign: "center",**

**fontFamily: "normalText",**

**fontWeight: "bold",**

**fontSize: 17,**

**color: black,**

**}}**

**>**

**Register**

**</Text>**

**</TouchableOpacity>**

**</View>**

**{/\* </View> \*/}**

**</ImageBackground>**

**</View>**

**C. API Call (postReq):**

**import axios from 'axios';**

**export const PostRequestCall = async (apiPath, payload, token, formData) => {**

**let auth = {};**

**if (typeof token !== 'undefined' && token !== null && token !== '') {**

**auth = {**

**Authorization: `Bearer ${token}`,**

**};**

**}**

**let headers = {**

**...auth,**

**Accept: 'application/json',**

**'Content-Type': formData ? 'multipart/form-data' : 'application/json',**

**};**

**const postResponse = await axios**

**.post(`http://192.168.55.70:5000/${apiPath}`, payload, {**

**headers: headers,**

**})**

**.then(function (result) {**

**return result;**

**})**

**.catch(err => err.response);**

**return postResponse;**

**};**

**D. Server File:**

**// // server.js**

**// const express = require("express");**

**// const dotenv = require("dotenv");**

**// const cors = require("cors");**

**// const connectDB = require("./config/db");**

**// const swaggerDocs = require("./swagger");**

**// // Load environment variables**

**// dotenv.config();**

**// // Connect to MongoDB**

**// connectDB();**

**// const app = express();**

**// swaggerDocs(app);**

**// app.use(express.json());**

**// app.use(cors());**

**// // Routes**

**// app.use("/api/auth", require("./routes/authRoutes"));**

**// app.use("/api/rides", require("./routes/rideRoutes"));**

**// const PORT = process.env.PORT || 5000;**

**// app.listen(PORT, () => console.log(`Server running on port ${PORT}`));**

**const express = require("express");**

**const mongoose = require("mongoose");**

**const authRoutes = require("./routes/authRoutes");**

**const rideRoutes = require("./routes/rideRoutes");**

**const bookingRoutes = require("./routes/bookingRoutes");**

**require("dotenv").config();**

**const app = express();**

**const port = process.env.PORT || 5000;**

**// Middleware**

**app.use(express.json());**

**app.use(express.urlencoded({ extended: true }));**

**// Routes**

**app.use("/api/auth", authRoutes);**

**app.use("/api/rides", rideRoutes); // Ensure this is correctly configured**

**app.use("/api/bookings", bookingRoutes); // Ensure this is correctly configured**

**// Connect to MongoDB**

**mongoose**

**.connect(process.env.MONGO\_URI, {**

**useNewUrlParser: true,**

**useUnifiedTopology: true,**

**})**

**.then(() => console.log("MongoDB connected"))**

**.catch((err) => console.error("MongoDB connection error:", err));**

**// Start the server**

**app.listen(port, "192.168.55.70", () => {**

**console.log(`Server running on port ${port}`);**

**});**

**4.6 Limitations and Bibliography:**

**Core Documentation and Resources:**

1. **React Native Documentation**  
   Meta Platforms, Inc.  
   Official documentation for React Native, covering setup, components, APIs, and guides.  
   Available at: https://reactnative.dev/docs
2. **JavaScript Documentation**  
   Mozilla Developer Network (MDN).  
   Comprehensive guide and reference for JavaScript, the primary language used in React Native.  
   Available at: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
3. **React Documentation**  
   Meta Platforms, Inc.  
   Documentation for React, the library that underpins React Native's core principles.  
   Available at: https://reactjs.org/docs

**Libraries and Tools**

1. **Expo Documentation**  
   Expo Team.  
   Reference for using Expo CLI and libraries for quick development in React Native.  
   Available at: https://docs.expo.dev
2. **Redux Toolkit Documentation**  
   Redux Team.  
   Simplified state management for React and React Native.  
   Available at: https://redux-toolkit.js.org
3. **React Navigation**  
   React Navigation Team.  
   Documentation for implementing navigation in React Native applications.  
   Available at: https://reactnavigation.org/docs
4. **Axios Documentation**  
   Axios Developers.  
   Promise-based HTTP client for making API requests.  
   Available at: <https://axios-http.com/docs>

**Community Resources**

1. **Stack Overflow**  
   Various contributors.  
   Solutions and discussions related to common React Native development challenges.  
   Available at: <https://stackoverflow.com>
2. **GitHub Repositories**  
   GitHub Inc.  
   Various open-source projects and code samples for React Native.  
   Examples: React Native repository (<https://github.com/facebook/react-native>)
3. **Blog Articles and Tutorials**  
   Authors: Multiple.  
   Tutorials and best practices shared by developers. Examples include blogs on Medium, Dev.to, or freeCodeCamp.

**Design and UI References**

1. **Material Design Guidelines**Google.  
   Reference for implementing material design principles in React Native applications.  
   Available at: https://material.io/design
2. **Figma for Mobile Design**Figma Team.  
   Design and prototype guidelines for mobile apps.  
   Available at: <https://www.figma.com>