Milestone2

Ziad Ashraf Ahmed Ahmed

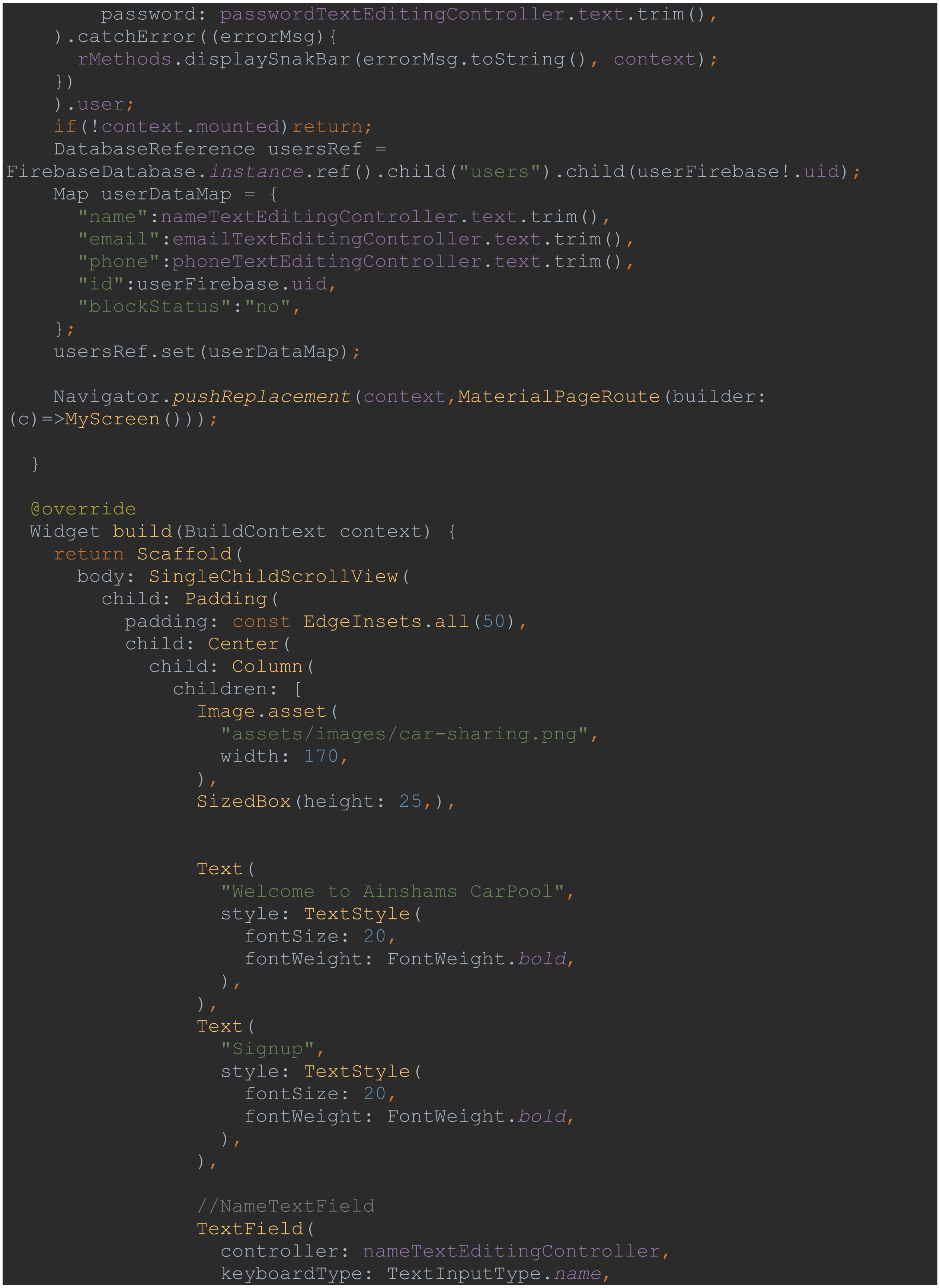
19p7095

MileStone2

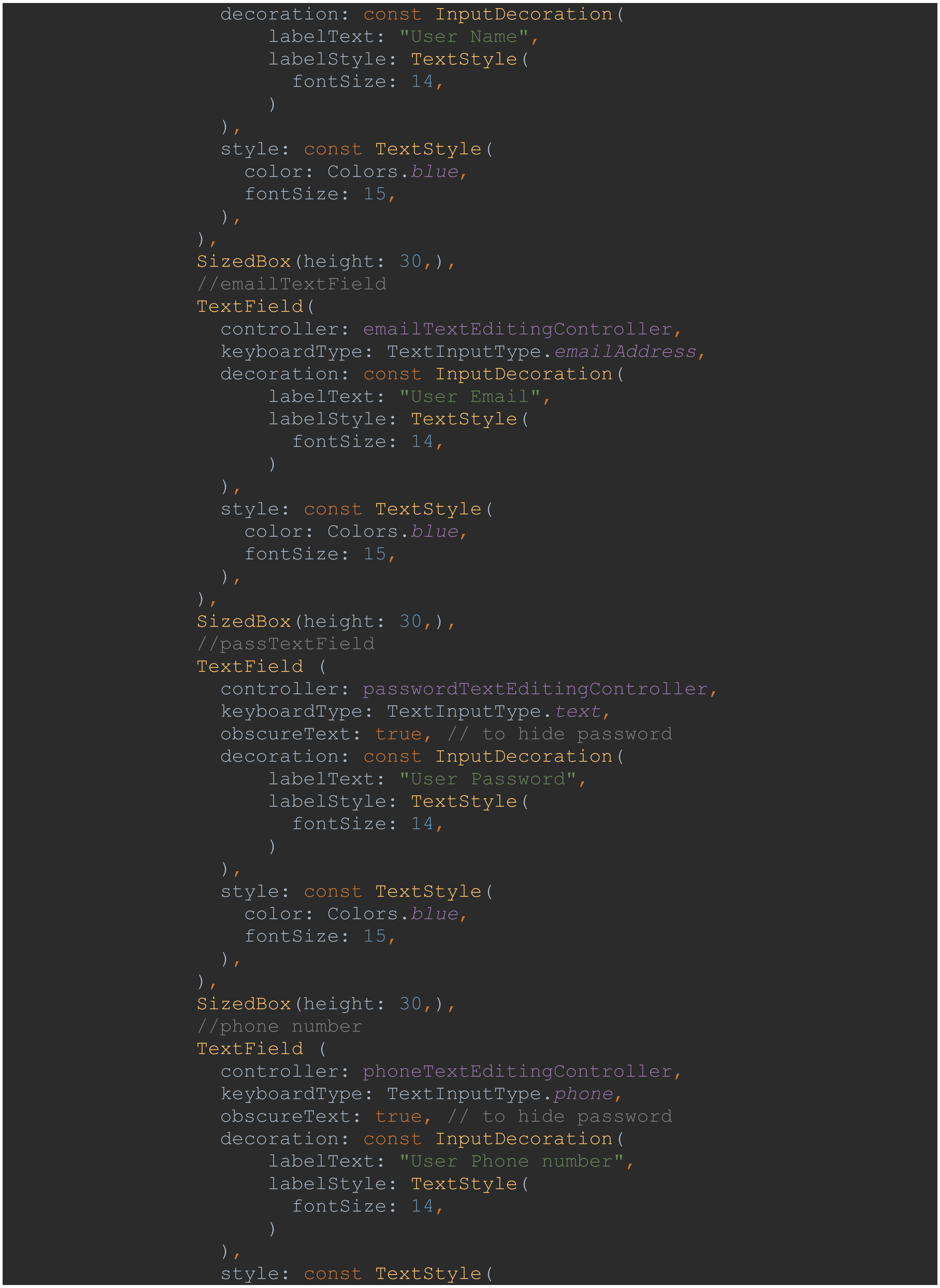
The second milestone is the basic function of the application of sign-up/login page, available routes. The Java/Kotlin/Dart code should be copied in a Word file, add screenshots of the layout, and list the testcase scenarios. Following that, convert the file into PDF and submit it on LMS.

sign-up

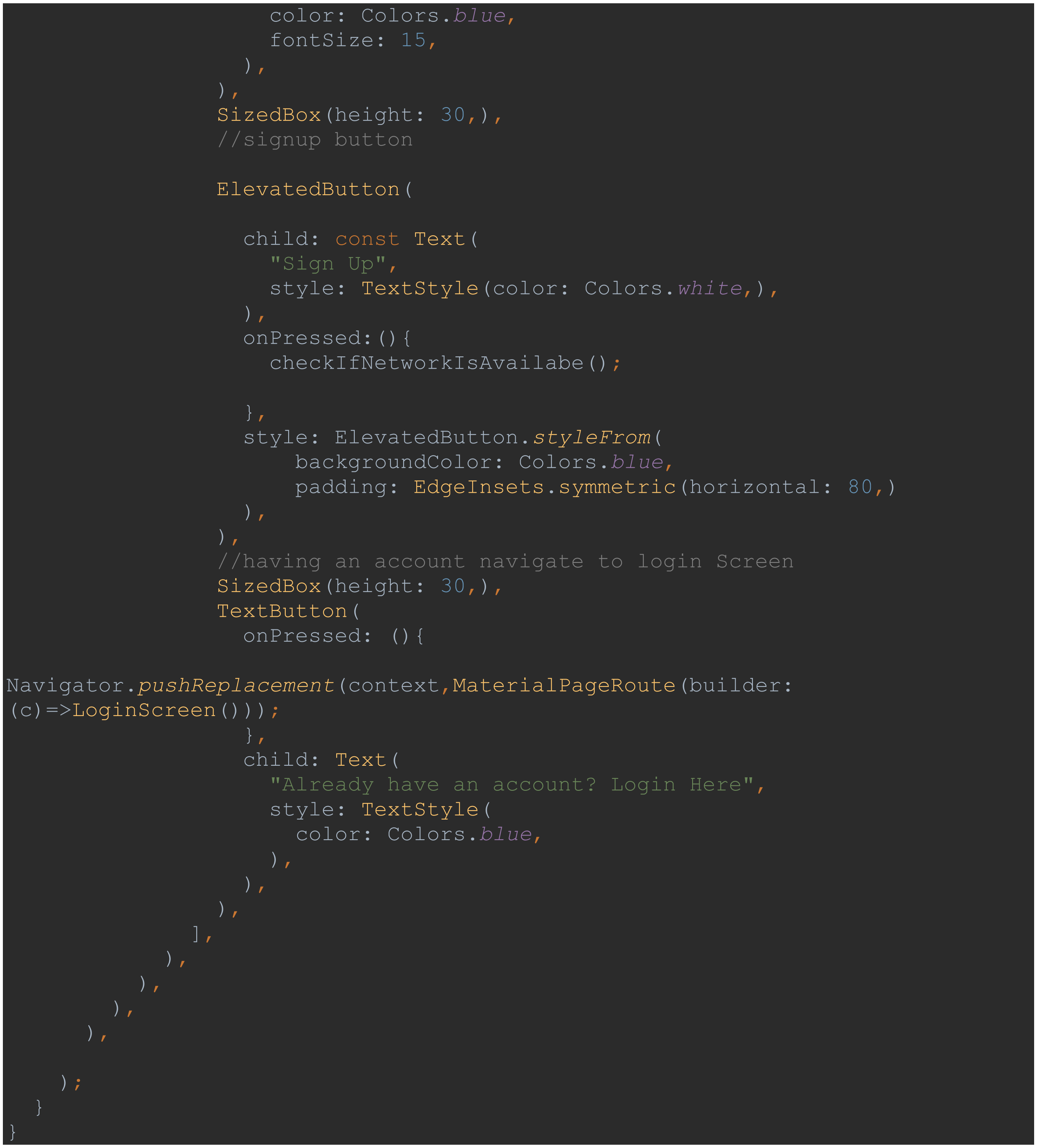
```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_database/firebase_database.dart';
import 'package:flutter/material.dart';
import 'package:project/home_screen.dart';
import 'package:project/login_screen.dart';
import 'package:project/methods/reusable_methods.dart';

class SignUpScreen extends StatefulWidget {
  const SignUpScreen({super.key});

  @override
  State<SignUpScreen> createState() => _SignUpScreenState();
}

class _SignUpScreenState extends State<SignUpScreen> {
  TextEditingController emailTextEditingController = TextEditingController();
  TextEditingController nameTextEditingController = TextEditingController();
  TextEditingController passwordTextEditingController =
TextEditingController();
  TextEditingController phoneTextEditingController = TextEditingController();
  ReusableMethods rMethods = ReusableMethods();

  checkIfNetworkIsAvailabe(){
    rMethods.checkConnectivity(context);
    signUpFormValidation();
  }

  signUpFormValidation(){
    if(nameTextEditingController.text.trim().length<3){
      rMethods.displaySnakBar("Name Must Be Atleast 4 charachters", context);
    }
    else if(!emailTextEditingController.text.endsWith("@eng.asu.edu.eg")){//
try to find method to check last few digits
      rMethods.displaySnakBar("Please SignUp with ASU Domain Email",
context);
    }
    else if(passwordTextEditingController.text.trim().length<6){
      rMethods.displaySnakBar("Password Must Be Atleast 6 Charachters",
context);
    }
    else if(phoneTextEditingController.text.trim().length<11){
      rMethods.displaySnakBar("Phone Number Must Be Atleast 11 Digets",
context);
    }
    else{
      registerNewUser();
    }

  }

  registerNewUser()async{
    final User? userFirebase = (
    await FirebaseAuth.instance.createUserWithEmailAndPassword(
        email: emailTextEditingController.text.trim(),
```



```dart
        password: passwordTextEditingController.text.trim(),
    ).catchError((errorMsg){
      rMethods.displaySnakBar(errorMsg.toString(), context);
    })
    ).user;
    if(!context.mounted)return;
    DatabaseReference usersRef =
FirebaseDatabase.instance.ref().child("users").child(userFirebase!.uid);
    Map userDataMap = {
      "name":nameTextEditingController.text.trim(),
      "email":emailTextEditingController.text.trim(),
      "phone":phoneTextEditingController.text.trim(),
      "id":userFirebase.uid,
      "blockStatus":"no",
    };
    usersRef.set(userDataMap);

    Navigator.pushReplacement(context,MaterialPageRoute(builder:
(c)=>MyScreen()));

  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        child: Padding(
          padding: const EdgeInsets.all(50),
          child: Center(
            child: Column(
              children: [
                Image.asset(
                  "assets/images/car-sharing.png",
                  width: 170,
                ),
                SizedBox(height: 25,),


                Text(
                  "Welcome to Ainshams CarPool",
                  style: TextStyle(
                    fontSize: 20,
                    fontWeight: FontWeight.bold,
                  ),
                ),
                Text(
                  "Signup",
                  style: TextStyle(
                    fontSize: 20,
                    fontWeight: FontWeight.bold,
                  ),
                ),

                //NameTextField
                TextField(
                  controller: nameTextEditingController,
                  keyboardType: TextInputType.name,
```

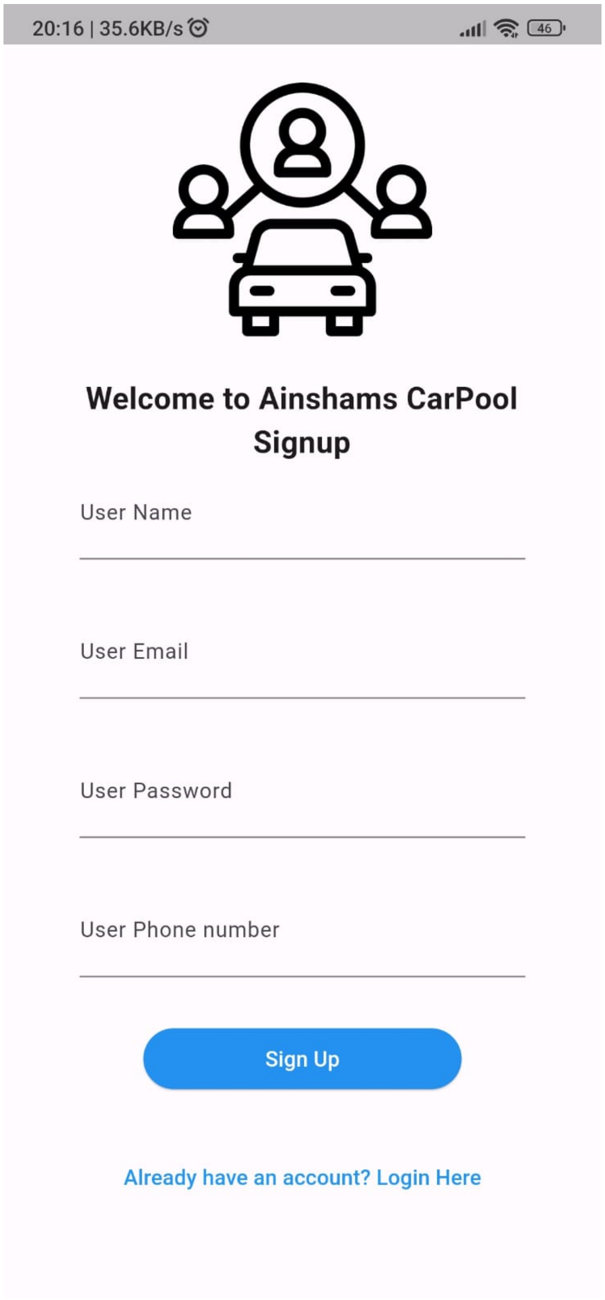


```
                  decoration: const InputDecoration(
                      labelText: "User Name",
                      labelStyle: TextStyle(
                        fontSize: 14,
                      )
                  ),
                  style: const TextStyle(
                    color: Colors.blue,
                    fontSize: 15,
                  ),
                ),
                SizedBox(height: 30,),
                //emailTextField
                TextField(
                  controller: emailTextEditingController,
                  keyboardType: TextInputType.emailAddress,
                  decoration: const InputDecoration(
                      labelText: "User Email",
                      labelStyle: TextStyle(
                        fontSize: 14,
                      )
                  ),
                  style: const TextStyle(
                    color: Colors.blue,
                    fontSize: 15,
                  ),
                ),
                SizedBox(height: 30,),
                //passTextField
                TextField (
                  controller: passwordTextEditingController,
                  keyboardType: TextInputType.text,
                  obscureText: true, // to hide password
                  decoration: const InputDecoration(
                      labelText: "User Password",
                      labelStyle: TextStyle(
                        fontSize: 14,
                      )
                  ),
                  style: const TextStyle(
                    color: Colors.blue,
                    fontSize: 15,
                  ),
                ),
                SizedBox(height: 30,),
                //phone number
                TextField (
                  controller: phoneTextEditingController,
                  keyboardType: TextInputType.phone,
                  obscureText: true, // to hide password
                  decoration: const InputDecoration(
                      labelText: "User Phone number",
                      labelStyle: TextStyle(
                        fontSize: 14,
                      )
                  ),
                  style: const TextStyle(
```



```
                    color: Colors.blue,
                    fontSize: 15,
                  ),
                ),
                SizedBox(height: 30,),
                //signup button

                ElevatedButton(

                  child: const Text(
                    "Sign Up",
                    style: TextStyle(color: Colors.white,),
                  ),
                  onPressed:(){
                    checkIfNetworkIsAvailabe();

                  },
                  style: ElevatedButton.styleFrom(
                      backgroundColor: Colors.blue,
                      padding: EdgeInsets.symmetric(horizontal: 80,)
                  ),
                ),
                //having an account navigate to login Screen
                SizedBox(height: 30,),
                TextButton(
                  onPressed: (){

Navigator.pushReplacement(context,MaterialPageRoute(builder:
(c)=>LoginScreen()));
                  },
                  child: Text(
                    "Already have an account? Login Here",
                    style: TextStyle(
                      color: Colors.blue,
                    ),
                  ),
                ),
              ],
            ),
          ),
        ),
      ),

    );
  }
}
```

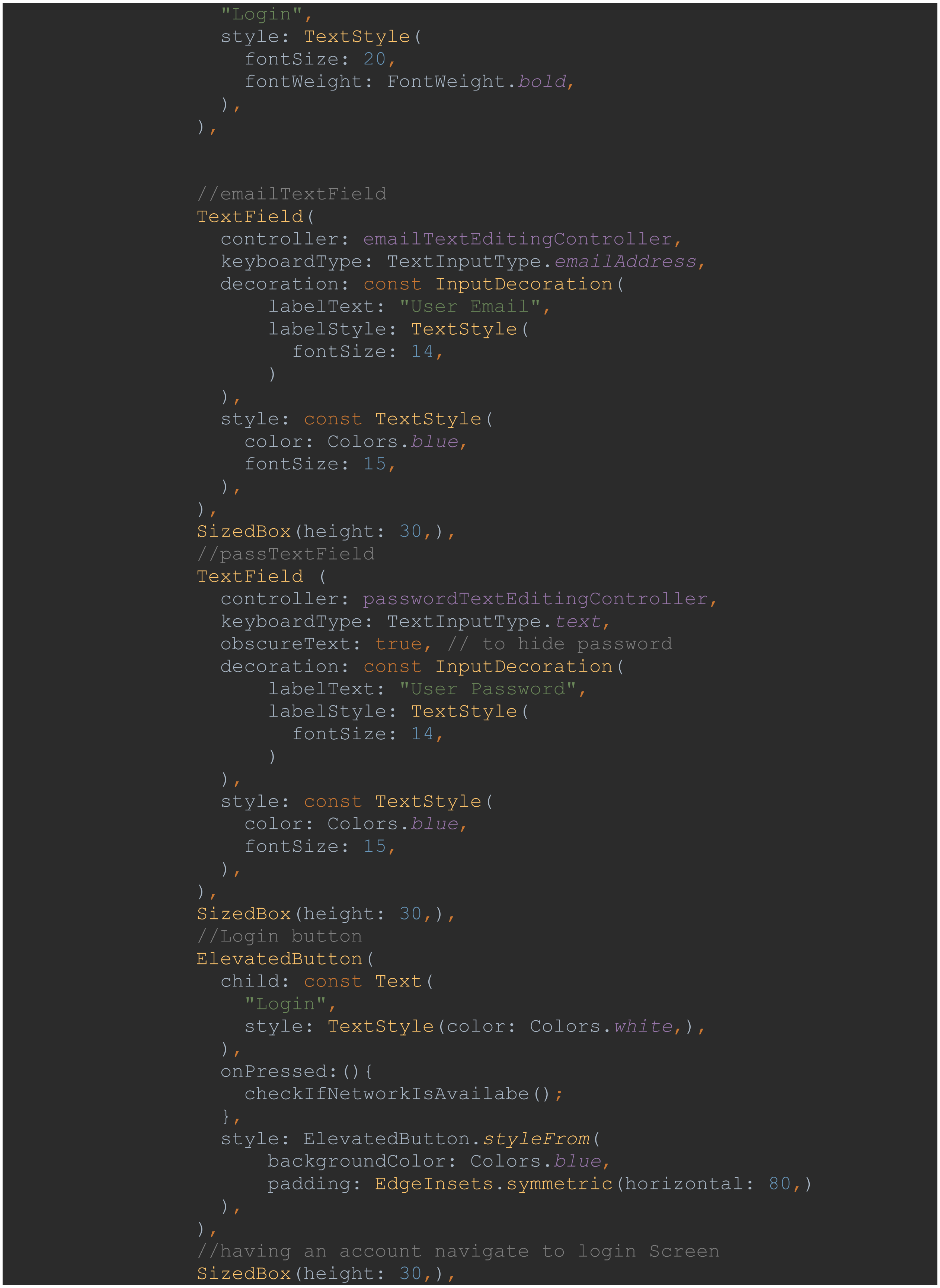
Screenshot



Welcome to Ainshams CarPool Signup

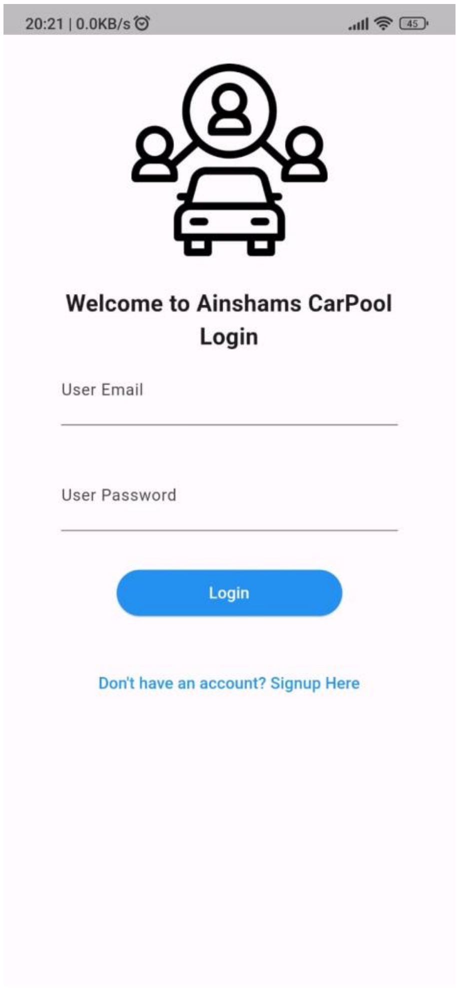
Login

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_database/firebase_database.dart';
import 'package:flutter/material.dart';
import 'package:project/home_screen.dart';
import 'package:project/signup_screen.dart';

import 'methods/reusable_methods.dart';
class LoginScreen extends StatefulWidget {
  const LoginScreen({super.key});

  @override
  State<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  TextEditingController emailTextEditingController = TextEditingController();
  TextEditingController passwordTextEditingController =
TextEditingController();
  ReusableMethods rMethods = ReusableMethods();

  checkIfNetworkIsAvailabe(){
    rMethods.checkConnectivity(context);
    logInFormValidation();
  }

  logInFormValidation(){
    if(!emailTextEditingController.text.endsWith("@eng.asu.edu.eg")){// try
to find method to check last few digits
      rMethods.displaySnakBar("Please SignUp with ASU Domain Email",
context);
    }
    else if(passwordTextEditingController.text.trim().length<6){
      rMethods.displaySnakBar("Password Must Be Atleast 6 Charachters",
context);
    }
    else{
      if(emailTextEditingController.text == "test@eng.asu.edu.eg"){// used to
bypass the authentication --for the aid of testing
        Navigator.pushReplacementNamed(context,'/home_screen');
      }
      else
        LogInUser();
    }

  }

  LogInUser()async{

    final User? userFirebase = (
        await FirebaseAuth.instance.signInWithEmailAndPassword(
          email: emailTextEditingController.text.trim(),
          password: passwordTextEditingController.text.trim(),
        ).catchError((errorMsg){
          rMethods.displaySnakBar(errorMsg.toString(), context);
        })
    ).user;

    if(!context.mounted)return;

    if(userFirebase != null){
      DatabaseReference usersRef =
FirebaseDatabase.instance.ref().child("users").child(userFirebase!.uid);
      usersRef.once().then((snap){
        if(snap.snapshot.value != null){

          if((snap.snapshot.value as Map)["blockStatus"] == "no"){
            Navigator.pushReplacementNamed(context,'/home_screen');
          }
          else{
            FirebaseAuth.instance.signOut();
            rMethods.displaySnakBar("This Account Is Blocked", context);
          }

        }else{
          FirebaseAuth.instance.signOut();
          rMethods.displaySnakBar("The Account Not Found As User", context);

        }


      });
    }

  }


  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        child: Padding(
          padding: const EdgeInsets.all(50),
          child: Center(
            child: Column(
              children: [
                Image.asset(
                  "assets/images/car-sharing.png",
                  width: 170,
                ),
                SizedBox(height: 25,),


                Text(
                  "Welcome to Ainshams CarPool",
                  style: TextStyle(
                    fontSize: 20,
                    fontWeight: FontWeight.bold,
                  ),
                ),
                Text(
```



```
                  "Login",
                  style: TextStyle(
                    fontSize: 20,
                    fontWeight: FontWeight.bold,
                  ),
                ),


                //emailTextField
                TextField(
                  controller: emailTextEditingController,
                  keyboardType: TextInputType.emailAddress,
                  decoration: const InputDecoration(
                      labelText: "User Email",
                      labelStyle: TextStyle(
                        fontSize: 14,
                      )
                  ),
                  style: const TextStyle(
                    color: Colors.blue,
                    fontSize: 15,
                  ),
                ),
                SizedBox(height: 30,),
                //passTextField
                TextField (
                  controller: passwordTextEditingController,
                  keyboardType: TextInputType.text,
                  obscureText: true, // to hide password
                  decoration: const InputDecoration(
                      labelText: "User Password",
                      labelStyle: TextStyle(
                        fontSize: 14,
                      )
                  ),
                  style: const TextStyle(
                    color: Colors.blue,
                    fontSize: 15,
                  ),
                ),
                SizedBox(height: 30,),
                //Login button
                ElevatedButton(
                  child: const Text(
                    "Login",
                    style: TextStyle(color: Colors.white,),
                  ),
                  onPressed:(){
                    checkIfNetworkIsAvailabe();
                  },
                  style: ElevatedButton.styleFrom(
                      backgroundColor: Colors.blue,
                      padding: EdgeInsets.symmetric(horizontal: 80,)
                  ),
                ),
                //having an account navigate to login Screen
                SizedBox(height: 30,),
                TextButton(
                  onPressed: (){

Navigator.pushReplacement(context,MaterialPageRoute(builder:
(c)=>SignUpScreen()));
                  },
                  child: Text(
                    "Don\'t have an account? Signup Here",
                    style: TextStyle(
                      color: Colors.blue,
                    ),
                  ),
                ),
              ],
            ),
          ),
        ),
      ),




    );;
  }
}
```

Screenshot

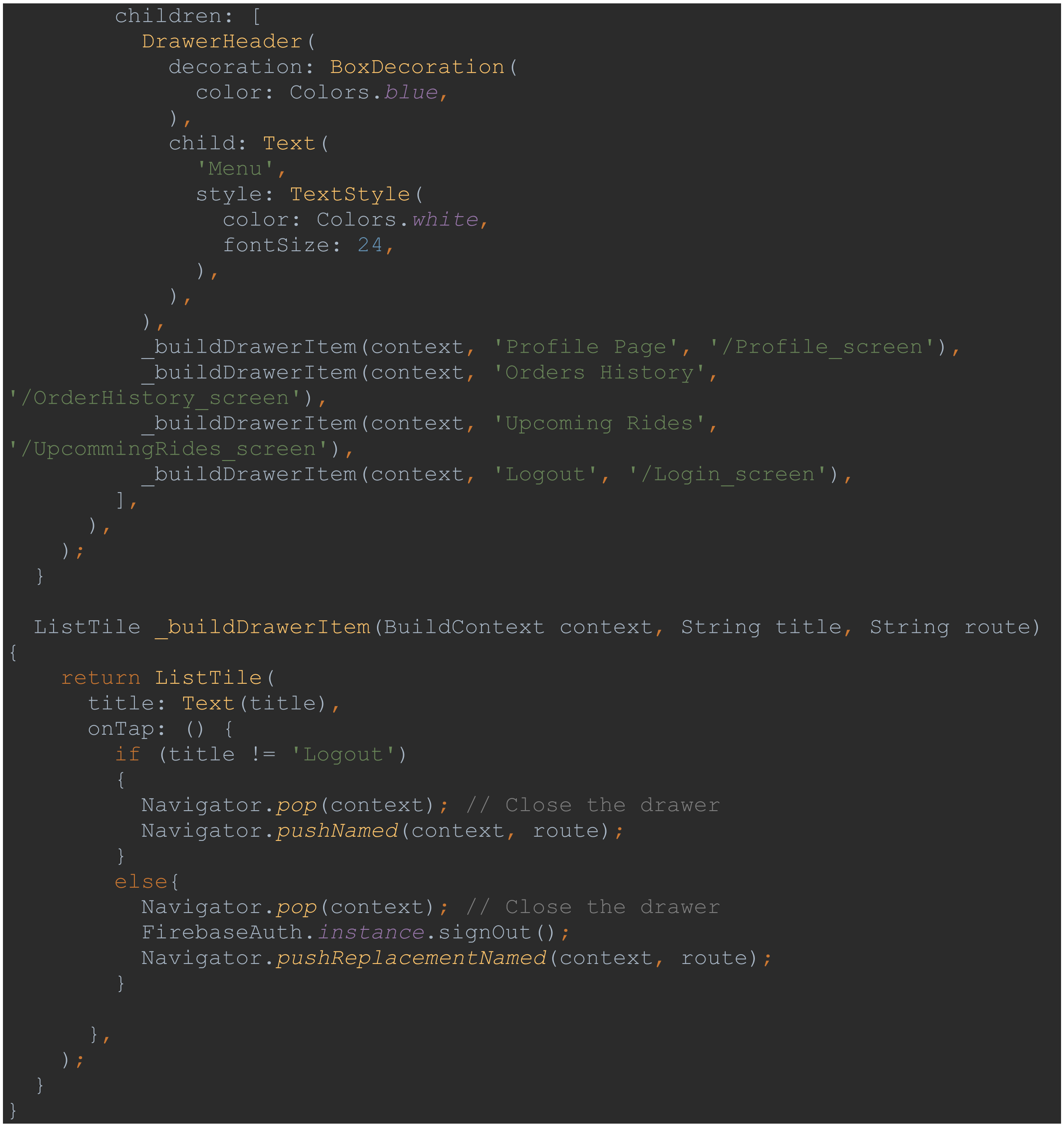


Welcome to Ainshams CarPool Login

Home_screen / User App

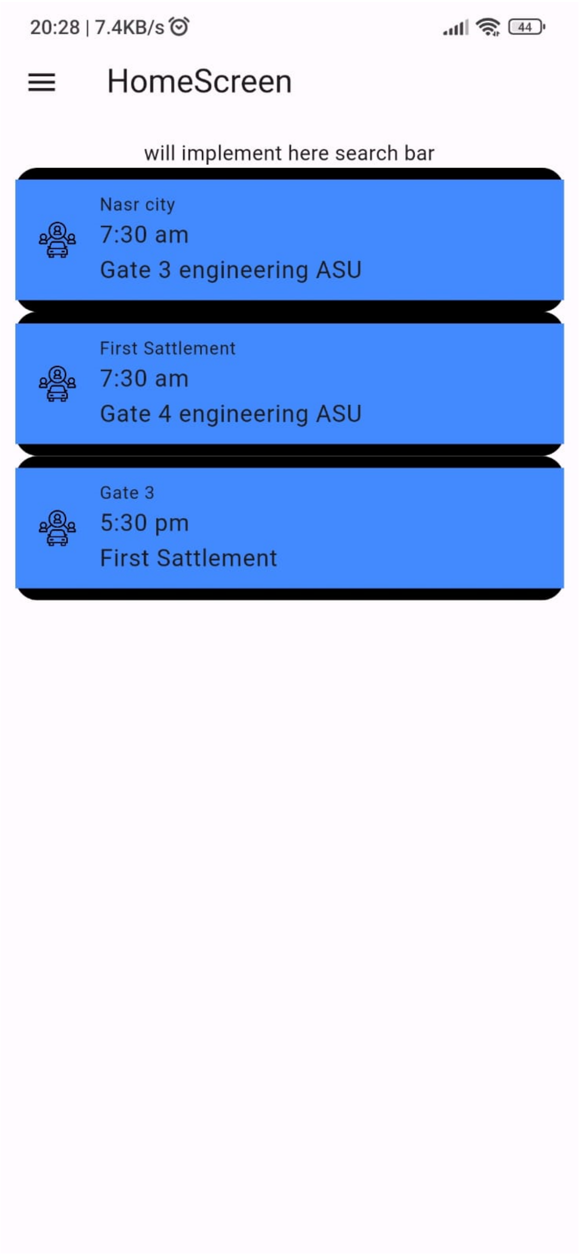
```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:firebase_database/firebase_database.dart';
void main() {
  runApp(MyScreen());
}

class MyScreen extends StatefulWidget {
  @override
  State<MyScreen> createState() => _MyScreenState();
}

class _MyScreenState extends State<MyScreen> {
  late DatabaseReference _databaseReference;
  List<Map<String, String>> mapRoutes = [];

  TextEditingController fromController = TextEditingController();
  TextEditingController toController = TextEditingController();
  TextEditingController timeController = TextEditingController();

  void initState() {
    super.initState();
    _databaseReference =
FirebaseDatabase.instance.reference().child("routes");
    _setupDataListener();
  }

  void _setupDataListener() {
    _databaseReference.onValue.listen((event) {
      if (event.snapshot.value != null) {
        print("Retrieved Data: ${event.snapshot.value}");

        // The retrieved data is a map, convert entries to a list
        var data = (event.snapshot.value as Map<dynamic, dynamic>).entries;

        if (data != null) {
          setState(() {
            mapRoutes = data
                .map((entry) => Map<String, String>.from({
              'From': '${entry.value['From']}',
              'To': '${entry.value['To']}',
              'Time': entry.value['Time'],
            }))
                .toList();
          });
        }
      } else {
        print("Snapshot value is null");
      }
    });
  }
```

```dart
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('HomeScreen'),
      ),
      drawer: _buildSideDrawer(context),
      body: Container(
        padding: EdgeInsets.all(10),
        child: Column(
          children: [
            Text("will implement here search bar"),
            Expanded(
              child: ListView.builder(
                itemCount: mapRoutes.length,
                itemBuilder: (context, index) {
                  return Container(
                    decoration: BoxDecoration(
                      color: Colors.black,
                      borderRadius: BorderRadius.circular(15),
                    ),
                    padding: const EdgeInsets.symmetric(vertical: 8.0),
                    child: Container(
                      color: Colors.blueAccent,
                      child: ListTile(
                        title: Column(
                          crossAxisAlignment: CrossAxisAlignment.start,
                          children: [
                            Text("${mapRoutes[index]['From']}",style:
TextStyle(fontSize: 12),),
                            Text("${mapRoutes[index]['Time']}"),
                            Text("${mapRoutes[index]['To']}"),
                          ],
                        ),
                        leading: Image.asset(
                          "assets/images/car-sharing.png",
                          height: 25,
                        ),
                        onTap: () {
                          Navigator.pushNamed(context, "/Cart_screen");
                        },
                      ),
                    ),
                  );
                },
              ),
            ),
          ],
        ),
      ),
    );
  }

  Drawer _buildSideDrawer(BuildContext context) {
    return Drawer(
      child: ListView(
        padding: EdgeInsets.zero,
```



```
        children: [
          DrawerHeader(
            decoration: BoxDecoration(
              color: Colors.blue,
            ),
            child: Text(
              'Menu',
              style: TextStyle(
                color: Colors.white,
                fontSize: 24,
              ),
            ),
          ),
          _buildDrawerItem(context, 'Profile Page', '/Profile_screen'),
          _buildDrawerItem(context, 'Orders History',
'/OrderHistory_screen'),
          _buildDrawerItem(context, 'Upcoming Rides',
'/UpcommingRides_screen'),
          _buildDrawerItem(context, 'Logout', '/Login_screen'),
        ],
      ),
    );
  }

  ListTile _buildDrawerItem(BuildContext context, String title, String route)
{
    return ListTile(
      title: Text(title),
      onTap: () {
        if (title != 'Logout')
        {
          Navigator.pop(context); // Close the drawer
          Navigator.pushNamed(context, route);
        }
        else{
          Navigator.pop(context); // Close the drawer
          FirebaseAuth.instance.signOut();
          Navigator.pushReplacementNamed(context, route);
        }

      },
    );
  }
}
```

Screen Shot



20:28 | 7.4KB/s

HomeScreen

will implement here search bar

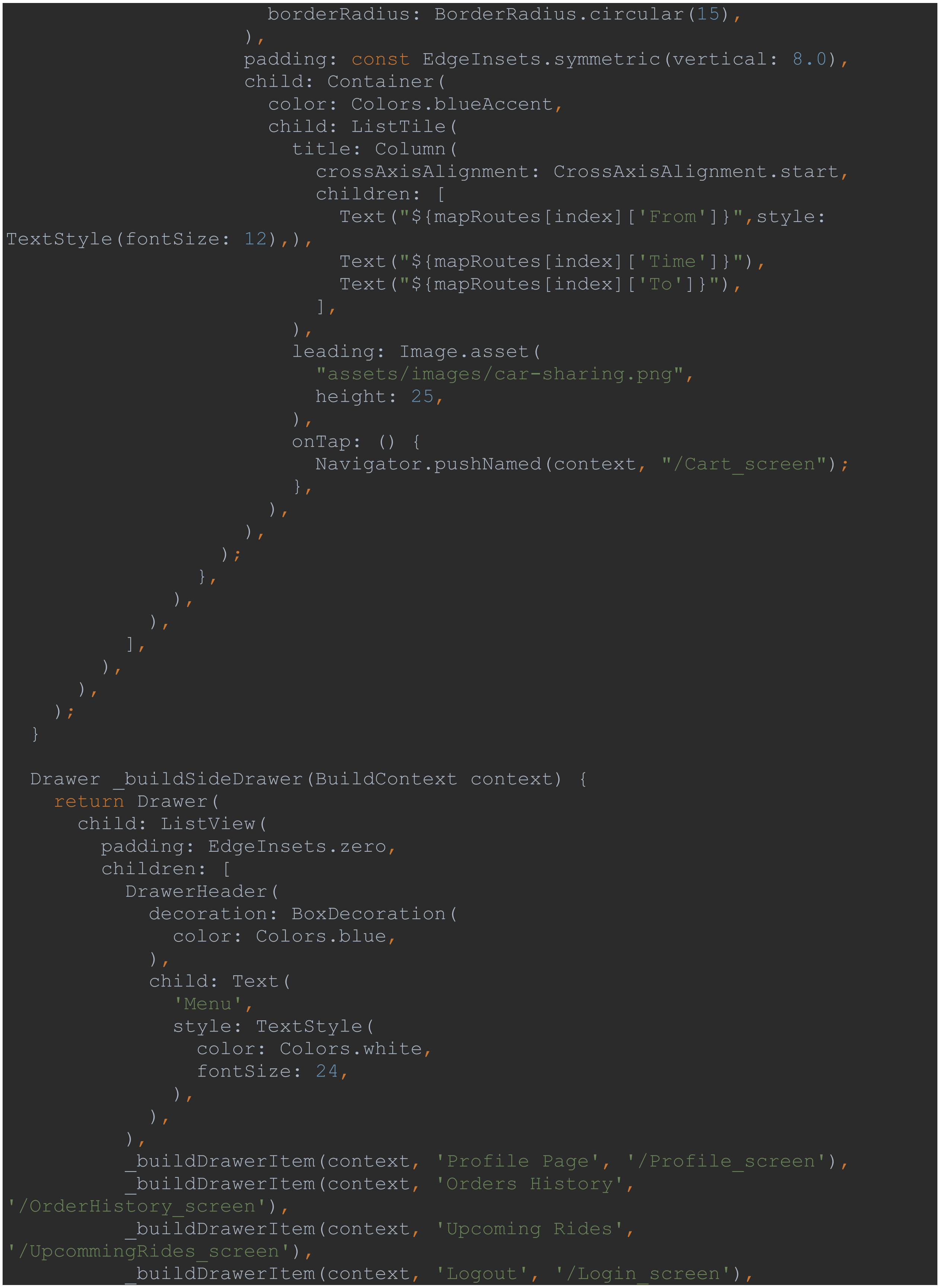
Nasr city
7:30 am
Gate 3 engineering ASU

First Sattlement
7:30 am
Gate 4 engineering ASU

Gate 3
5:30 pm
First Sattlement

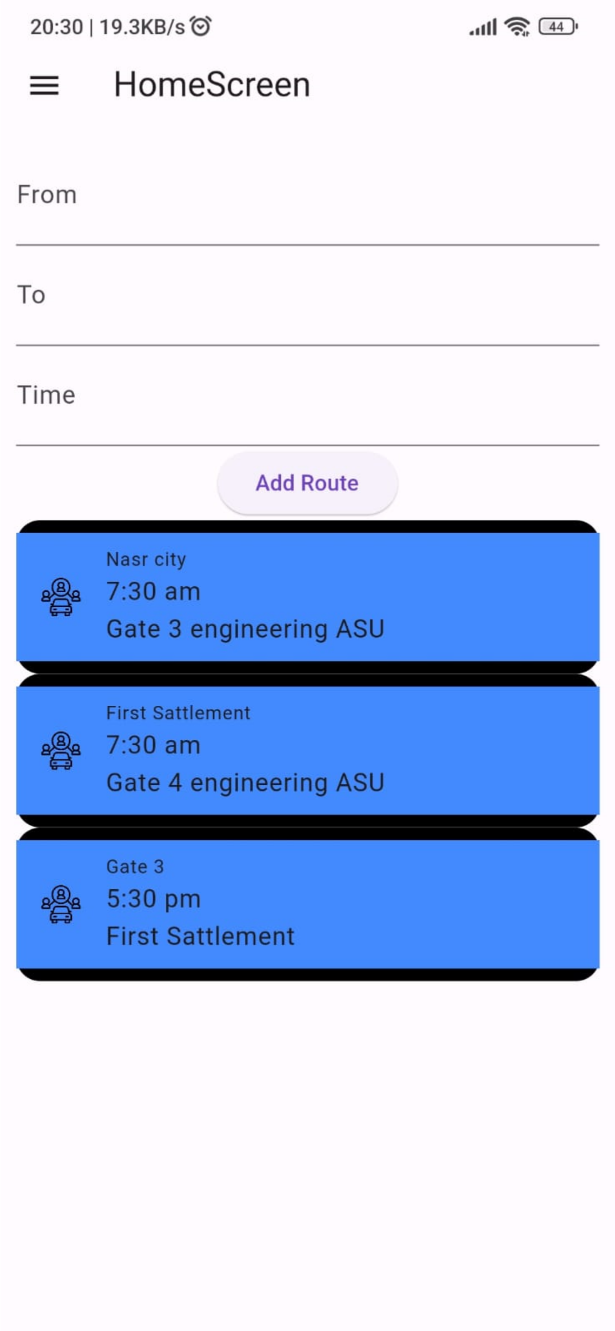
Home_screen / Driver

```
import 'package:flutter/material.dart';
import 'package:firebase_database/firebase_database.dart';
void main() {
  runApp(MyScreen());
}

class MyScreen extends StatefulWidget {
  @override
  State<MyScreen> createState() => _MyScreenState();
}

class _MyScreenState extends State<MyScreen> {
  late DatabaseReference _databaseReference;
  List<Map<String, String>> mapRoutes = [];

  TextEditingController fromController = TextEditingController();
  TextEditingController toController = TextEditingController();
  TextEditingController timeController = TextEditingController();

  void initState() {
    super.initState();
    _databaseReference =
FirebaseDatabase.instance.reference().child("routes");
    _setupDataListener();
  }

  void _setupDataListener() {
    _databaseReference.onValue.listen((event) {
      if (event.snapshot.value != null) {
        print("Retrieved Data: ${event.snapshot.value}");

        // The retrieved data is a map, convert entries to a list
        var data = (event.snapshot.value as Map<dynamic, dynamic>)?.entries;

        if (data != null) {
          setState(() {
            mapRoutes = data
                .map((entry) => Map<String, String>.from({
              'From': '${entry.value['From']}',
              'To': '${entry.value['To']}',
              'Time': entry.value['Time'],
            }))
                .toList();
          });
        }
      } else {
        print("Snapshot value is null");
      }
    });
  }


  void _addRoute() {
    String from = fromController.text.trim();
    String to = toController.text.trim();
    String time = timeController.text.trim();

    if (from.isNotEmpty && to.isNotEmpty && time.isNotEmpty) {
      _databaseReference.push().set({
        'From': '$from',
        'To': '$to',
        'Time': time,
      });

      // Clear the text controllers after adding a route
      fromController.clear();
      toController.clear();
      timeController.clear();
    } else {
      print("One or more fields are empty");
    }
  }


  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('HomeScreen'),
      ),
      drawer: _buildSideDrawer(context),
      body: Container(
        padding: EdgeInsets.all(10),
        child: Column(
          children: [
            TextField(
              controller: fromController,
              decoration: InputDecoration(labelText: 'From'),
            ),
            TextField(
              controller: toController,
              decoration: InputDecoration(labelText: 'To'),
            ),
            TextField(
              controller: timeController,
              decoration: InputDecoration(labelText: 'Time'),
            ),
            ElevatedButton(
              onPressed: () {
                _addRoute();
                // Adding print statement to check mapRoutes after adding a
route
                print("Updated mapRoutes: $mapRoutes");
              },
              child: Text('Add Route'),
            ),
            Expanded(
              child: ListView.builder(
                itemCount: mapRoutes.length,
                itemBuilder: (context, index) {
                  return Container(
                    decoration: BoxDecoration(
                      color: Colors.black,
```



```
                      borderRadius: BorderRadius.circular(15),
                    ),
                    padding: const EdgeInsets.symmetric(vertical: 8.0),
                    child: Container(
                      color: Colors.blueAccent,
                      child: ListTile(
                        title: Column(
                          crossAxisAlignment: CrossAxisAlignment.start,
                          children: [
                            Text("${mapRoutes[index]['From']}",style:
TextStyle(fontSize: 12),),
                            Text("${mapRoutes[index]['Time']}"),
                            Text("${mapRoutes[index]['To']}"),
                          ],
                        ),
                        leading: Image.asset(
                          "assets/images/car-sharing.png",
                          height: 25,
                        ),
                        onTap: () {
                          Navigator.pushNamed(context, "/Cart_screen");
                        },
                      ),
                    ),
                  );
                },
              ),
            ),
          ],
        ),
      ),
    );
  }

  Drawer _buildSideDrawer(BuildContext context) {
    return Drawer(
      child: ListView(
        padding: EdgeInsets.zero,
        children: [
          DrawerHeader(
            decoration: BoxDecoration(
              color: Colors.blue,
            ),
            child: Text(
              'Menu',
              style: TextStyle(
                color: Colors.white,
                fontSize: 24,
              ),
            ),
          ),
          _buildDrawerItem(context, 'Profile Page', '/Profile_screen'),
          _buildDrawerItem(context, 'Orders History',
'/OrderHistory_screen'),
          _buildDrawerItem(context, 'Upcoming Rides',
'/UpcommingRides_screen'),
          _buildDrawerItem(context, 'Logout', '/Login_screen'),
        ],
      ),
    );
  }

  ListTile _buildDrawerItem(BuildContext context, String title, String route)
{
    return ListTile(
      title: Text(title),
      onTap: () {
        Navigator.pop(context); // Close the drawer
        Navigator.pushNamed(context, route);
      },
    );
  }
}
```

Screen shot.



From

To

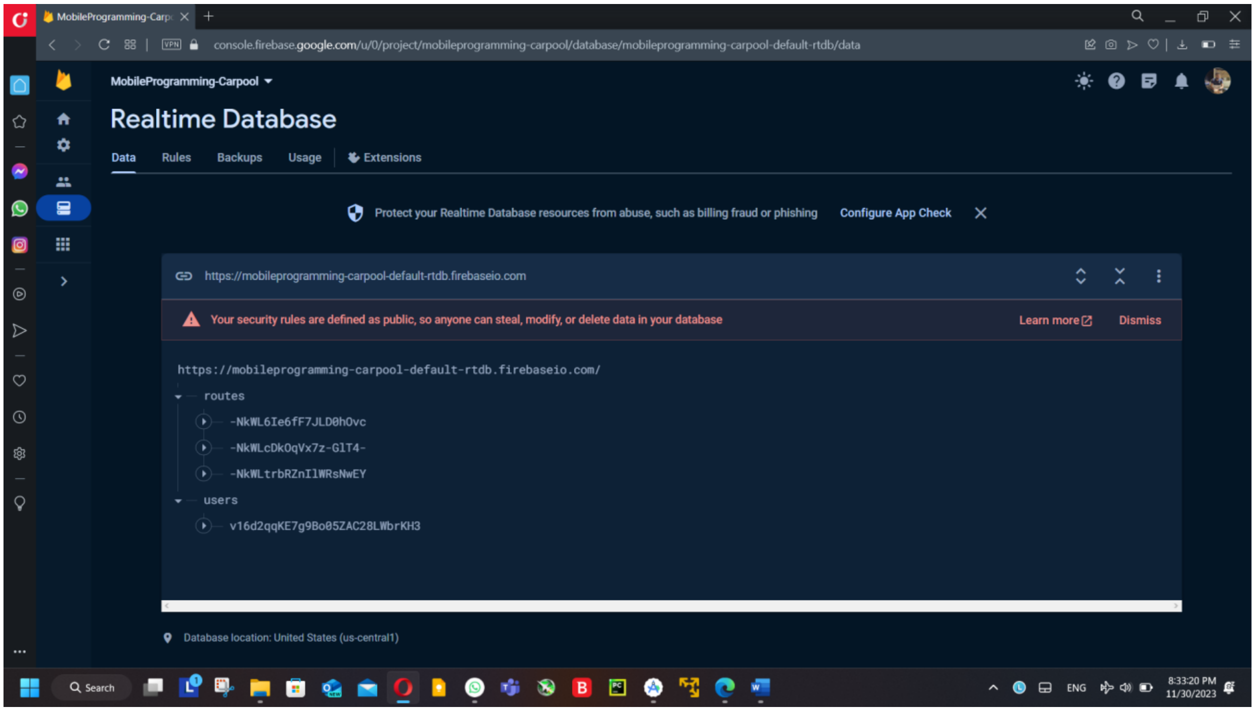
Time

Nasr city
7:30 am
Gate 3 engineering ASU

First Sattlement
7:30 am
Gate 4 engineering ASU

Gate 3
5:30 pm
First Sattlement

Firebase Screenshot



MobileProgramming-Carp
console.firebase.google.com/u/0/project/mobileprogramming-carpool/database/mobileprogramming-carpool-default-rtdb/data
MobileProgramming-Carpool

Realtime Database

Data
Rules
Backups
Usage
Extensions

Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://mobileprogramming-carpool-default-rtdb.firebaseio.com

Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

```
https://mobileprogramming-carpool-default-rtdb.firebaseio.com/
▾ routes
   ▸ -NkWL6Ie6fF7JLD0hOvc
   ▸ -NkWLcDkOqVx7z-GlT4-
   ▸ -NkWLtrbRZnIlWRsNwEY
▾ users
   ▸ v16d2qqKE7g9Bo05ZAC28LWbrKH3
```

Database location: United States (us-central1)

Testcases scenarios

User Authentication (Firebase Authentication):

1. **User Registration:**
    - Input valid registration details (email with collage domain, password, username, phonenumber).
    - Ensure the user account is created successfully.
    - Verify the user data in the Firebase Authentication console.
2. **User Successfully Login:**
    - Provide valid login credentials.
    - Verify that the user is logged in successfully.
    - Check for appropriate error handling when providing invalid credentials.
3. **Logout:**
    - Log out the user and ensure they are redirected to the login screen.
    - Verify that the user cannot access protected routes after logging out.

User Home Screen:

1. **Route Retrieval:**
    - Ensure that the available routes are fetched from the database and displayed on the home screen.

Driver's Home Screen and Data Submission:

1. **Driver Home Screen Trip Data Submission:**
    - Submit the data and ensure it is successfully stored in the Firebase real-time database.
    - Check for appropriate error handling for invalid or incomplete data submissions.
    - Check that the submitted data is fetched correctly from the user application

Interaction Between User and Driver:

1. **Real-time Data Update:**
    - After a driver submits a new trip, ensure that the user's home screen is automatically updated with the new route information without needing a manual refresh.

Error Handling:

1. **Invalid Input Handling:**
    - Test the application with invalid inputs during registration, login, and data submission.
    - Ensure appropriate error messages are displayed.
2. **Network Issues:**
    - Simulate network issues and check if the application handles them gracefully, providing informative error messages.

Navigation:

1. **Navigation Between Screens:**
    - Verify that users can navigate seamlessly between all screens

Note:

For the Aid of testing, we have email: test@eng.asu.edu.eg and password 123456789

This account don't go through the firebase authentication process in order to decrease the test cases dependencies in case something went wrong.