Project Title: Uber App **Roll No.: 12**

MAD & PWA Lab Journal

Experiment No.	06
Experiment Title.	To Connect Flutter UI with fireBase database
Roll No.	12
Name	Aryan Dangat
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO3: Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS
Grade:	

Project Title: Uber App **Roll No.: 12**

AIM : To Connect Flutter UI with fireBase database

Theory :

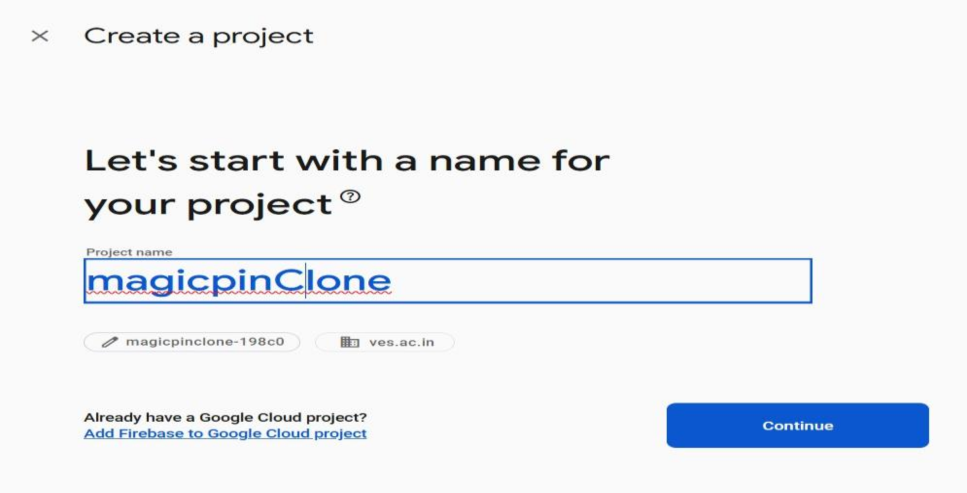
Firebase provides a powerful backend solution for Flutter applications, enabling features like authentication, real-time database, and cloud storage. By integrating Firebase, developers can easily manage user authentication and store data without setting up a separate backend. Firebase is a cloud-based platform by Google that provides backend services for mobile and web applications, eliminating the need to manage servers. It integrates seamlessly with Flutter to enhance app functionality, security, and performance.

It includes **Authentication** (email, social logins, phone sign-in), **Cloud Firestore & Realtime Database** for real-time data syncing, and **Cloud Storage** for handling images, videos, and documents. **Cloud Messaging (FCM)** enables push notifications, while **Crashlytics** and **Performance Monitoring** help in debugging and optimizing app performance. **Remote Config** allows updating app features without requiring updates, and **Analytics** provides insights into user behavior. With Firebase, Flutter apps can be more scalable, secure, and feature-rich without complex backend management.

Steps to Set Up Firebase with Android Apps

Step 1: Create a Firebase Project

1. Go to Firebase Console.
2. Click on **"Add Project"**, enter your project name, and proceed.



× Create a project

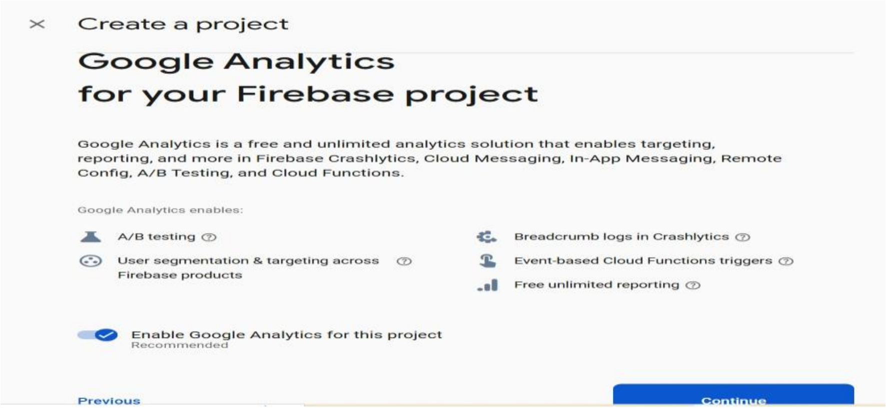
Let's start with a name for your project

Project name: magicpinClone

magicpinclone-198c0 ves.ac.in

Already have a Google Cloud project?
Add Firebase to Google Cloud project

3. Enable **Google Analytics** (optional) and complete the setup.



Create a project

Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.

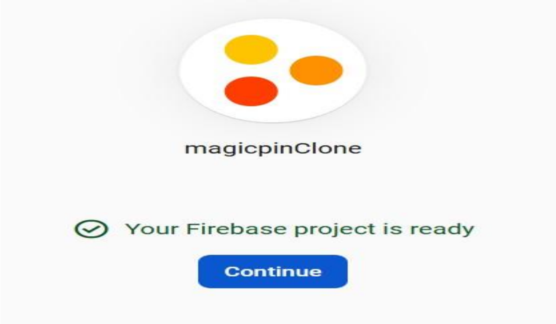
Google Analytics enables:

- A/B testing
- User segmentation & targeting across Firebase products
- Breadcrumb logs in Crashlytics
- Event-based Cloud Functions triggers
- Free unlimited reporting

☑ Enable Google Analytics for this project
Recommended

If you choose to use Google Analytics, you will need to review and accept the terms and conditions prior to project creation.

After pressing Continue, your project will be created and resources will be provisioned. You will then be directed to the dashboard for the new project.



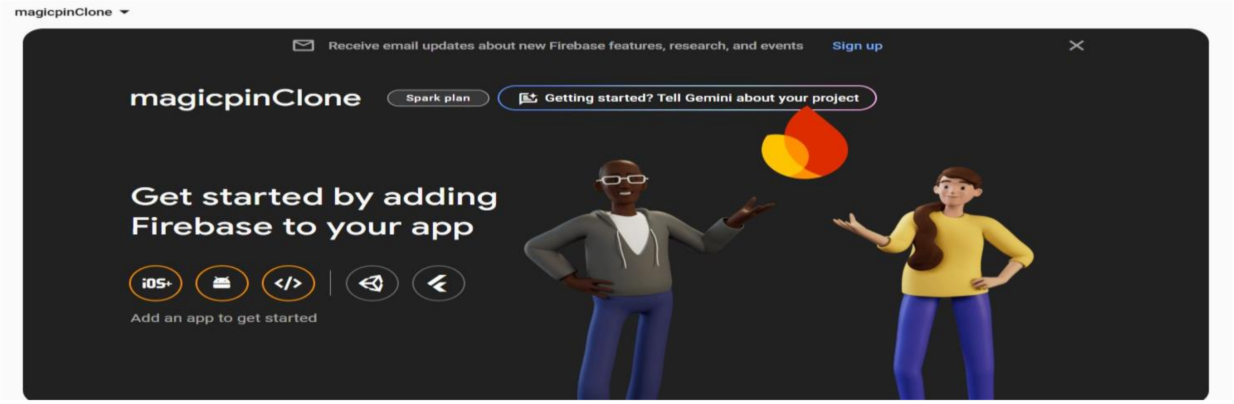
magicpinClone

Your Firebase project is ready

Step 2: Add Firebase to Your Flutter App

Adding Android support :

1. Click **"Add App"** → Select **Android**.



magicpinClone

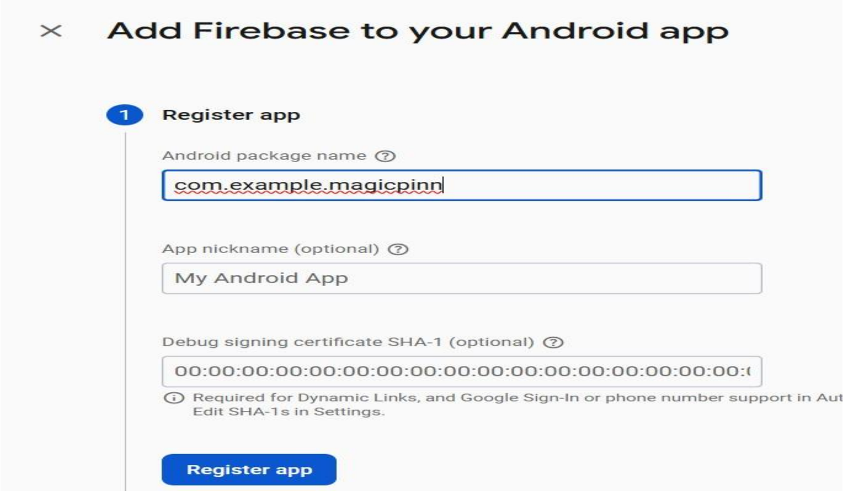
Receive email updates about new Firebase features, research, and events Sign up

magicpinClone Spark plan Getting started? Tell Gemini about your project

Get started by adding Firebase to your app

Add an app to get started

In order to add Android support to our Flutter application, select the Android logo from the dashboard. This brings us to the following screen:



Add Firebase to your Android app

1 Register app

Android package name: com.example.magicpinn

App nickname (optional): My Android App

Debug signing certificate SHA-1 (optional): 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:(

Required for Dynamic Links, and Google Sign-In or phone number support in Aut
Edit SHA-1s in Settings.

2. Enter the package name (found in android/app/build.gradle).

   The most important thing here is to match up the Android package name that you choose here with the one inside of our application.

   The structure consists of at least two segments. A common pattern is to use a domain name, a company name, and the application name:

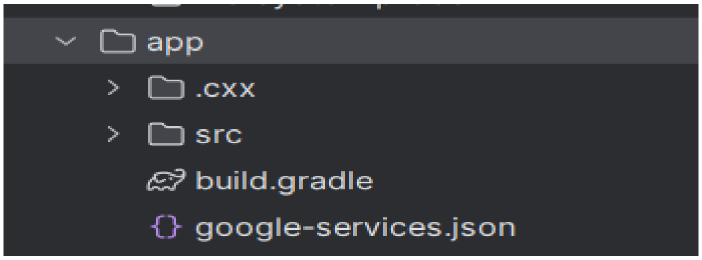
   com.example.flutterfirebaseexample

   You can copy it from applicationId in android/app/build.gradle in your code editor android/app/build.gradle

   You can skip the app nickname and debug signing keys at this stage. Select Register app to continue.

3. Download the **google-services.json** file and place it inside android/app/.

   This is important as it contains the API keys and other critical information for Firebase to use.



```
app
  .cxx
  src
  build.gradle
  google-services.json
```

Adding the Firebase SDK

We’ll now need to update our Gradle configuration to include the Google Services plugin.

Open android/build.gradle in your code editor and modify it to include the following:

```
dependencies {

   classpath  'com.google.gms:google-services:4.3.10'

}
```

android/buiild.gradle

```
buildscript {

   ext.kotlin_version = "1.9.22"

   repositories {

       google()

       mavenCentral()

   }

   dependencies {

       classpath  'com.google.gms:google-services:4.4.0'

       classpath  "org.jetbrains.kotlin:kotlin-gradle-plugin:1.9.22"

   }
```

Finally, update the app level file at android/app/build.gradle to include the following:

```
             apply plugin:  'com.google.gms.google-services'
```

android/app/build.gradle

```
apply plugin: 'com.android.application'
// Add this line
apply plugin:  'com.google.gms.google-services'

dependencies {
  // Import the Firebase BoM
  implementation platform('com.google.firebase:firebase-bom:28.0.0')

}

plugins {
    id "com.android.application"
    id "kotlin-android"
    id "dev.flutter.flutter-gradle-plugin"
     id 'com.google.gms.google-services'
}

dependencies {
  // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:33.9.0')
  // TODO: Add the dependencies for Firebase products you want to use
   // When using the BoM, don't specify versions in Firebase dependencies

 // Add the dependencies for any other desired Firebase products
//  https://firebase.google.com/docs/android/setup#available-libraries
}
```

Or else

apply plugin: 'com.google.gms.google-services'   at the end

With this update, we’re essentially applying the Google Services plugin as well as looking at how other Flutter Firebase plugins can be activated such as Analytics.

Step 3: Install Firebase Dependencies

Open pubspec.yaml and add:

```
dependencies:

  firebase_core: latest_version

  firebase_auth: latest_version

  cloud_firestore: latest_version
```

Then run :

```
flutter pub get
```

Step 4: Initialize Firebase in Flutter

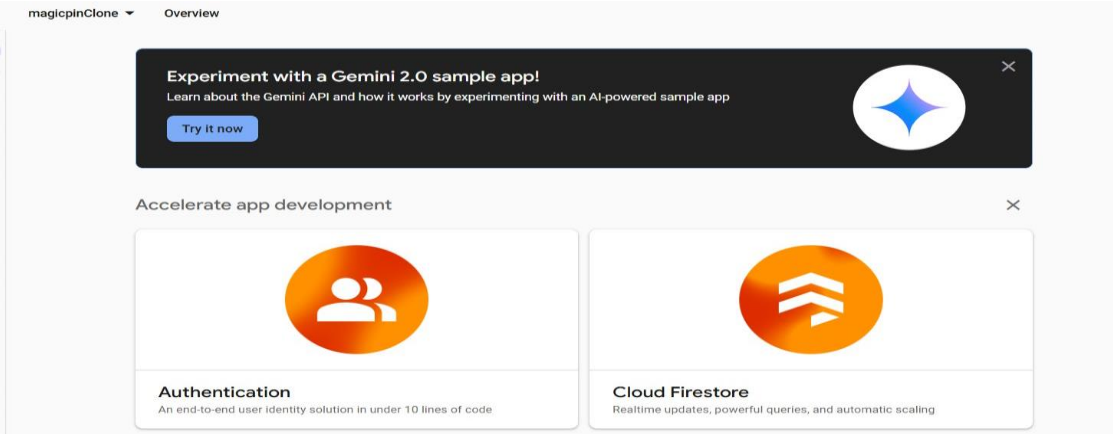
Open main.dart and update the main() function:

```
import 'package:flutter/material.dart';

import 'package:firebase_core/firebase_core.dart';

void main() async {

WidgetsFlutterBinding.ensureInitialized();

  await Firebase.initializeApp();

  runApp(MyApp());

}
```

Ensure that MyApp() is wrapped inside MaterialApp().

Step 5: Enable Authentication in Firebase

1. In Firebase Console, go to **Authentication** > **Sign-in method**.



magicpinClone ▾ Overview

Experiment with a Gemini 2.0 sample app!

Learn about the Gemini API and how it works by experimenting with an AI-powered sample app

Try it now

Accelerate app development

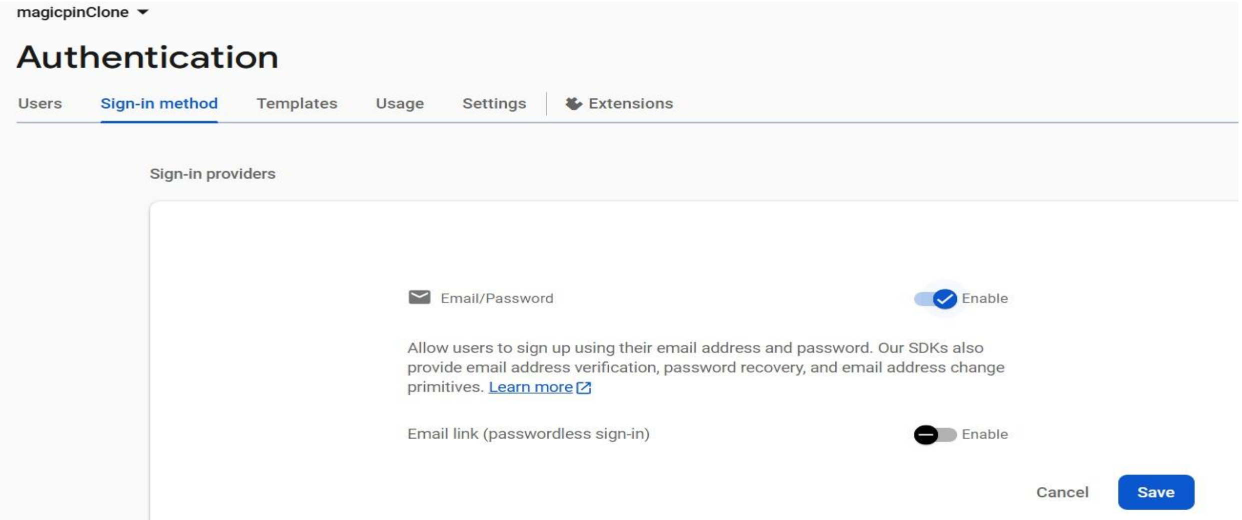
Authentication

An end-to-end user identity solution in under 10 lines of code

Cloud Firestore

Realtime updates, powerful queries, and automatic scaling

2. Enable **Email/Password Authentication** (or any other method).



magicpinClone ▾

Authentication

Users | Sign-in method | Templates | Usage | Settings | Extensions

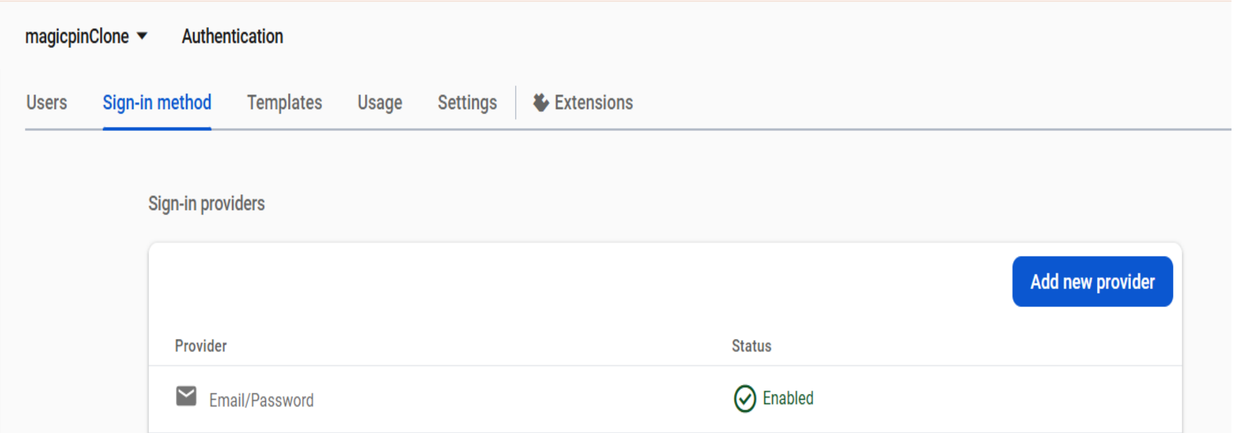
Sign-in providers

Email/Password ☑ Enable

Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery, and email address change primitives. Learn more

Email link (passwordless sign-in) □ Enable

Cancel Save



magicpinClone ▾ Authentication

Users | Sign-in method | Templates | Usage | Settings | Extensions

Sign-in providers

Add new provider

Provider	Status
Email/Password	✓ Enabled

Step 6: Implement Firebase Authentication in Flutter

1. Register a New User (Signup)

```
import 'package:firebase_auth/firebase_auth.dart';

Future<void> signUpUser(String email, String password) async {
  try {
    UserCredential userCredential = await FirebaseAuth.instance.createUserWithEmailAndPassword(
      email: email,
      password: password,
    );
    print("User registered: ${userCredential.user?.uid}");
  } catch (e) {
    print("Signup failed: $e");
  }
}
```

2. Login User

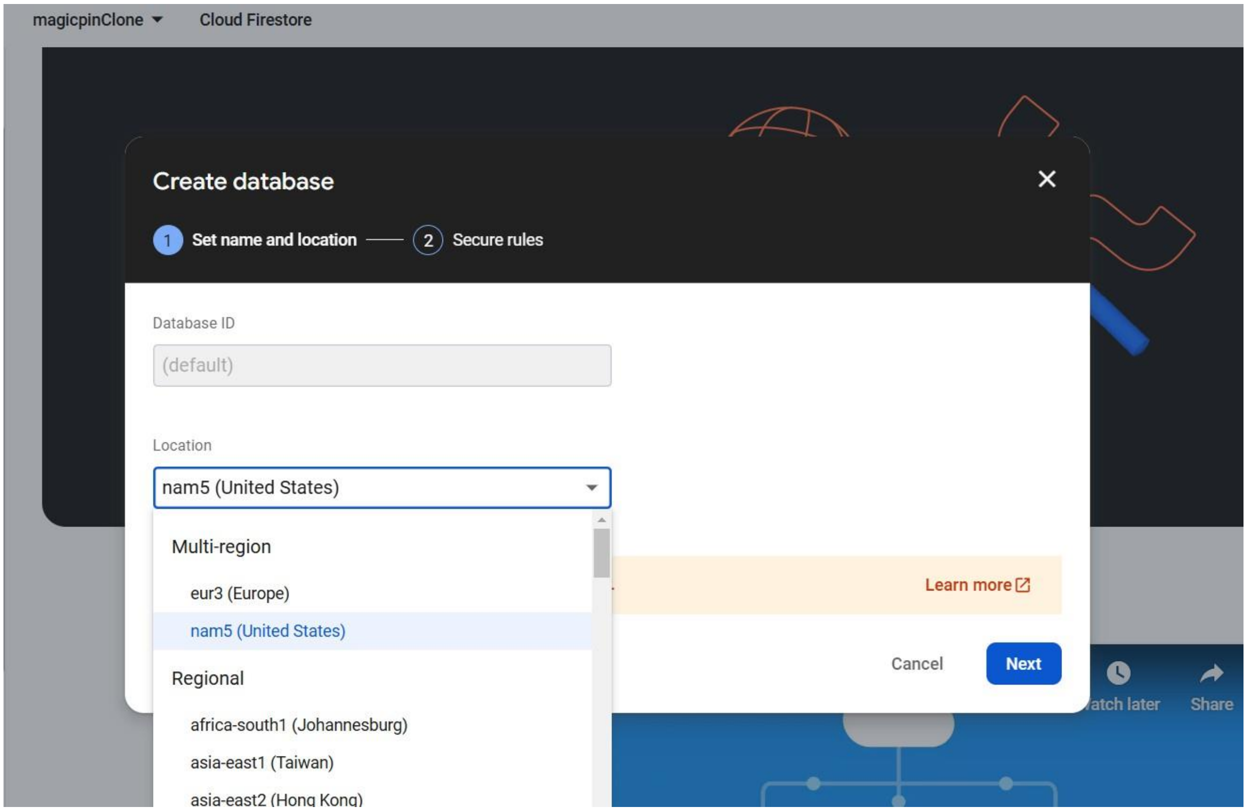
```
Future<void> loginUser(String email, String password) async {
  try {
    UserCredential userCredential = await FirebaseAuth.instance.signInWithEmailAndPassword(
      email: email,
      password: password,
    );
    print("User logged in: ${userCredential.user?.uid}");
  } catch (e) {
    print("Login failed: $e");
  }
}
```

3. Logout User

```
Future<void> logoutUser() async {
  await FirebaseAuth.instance.signOut();
  print("User logged out");
}
```

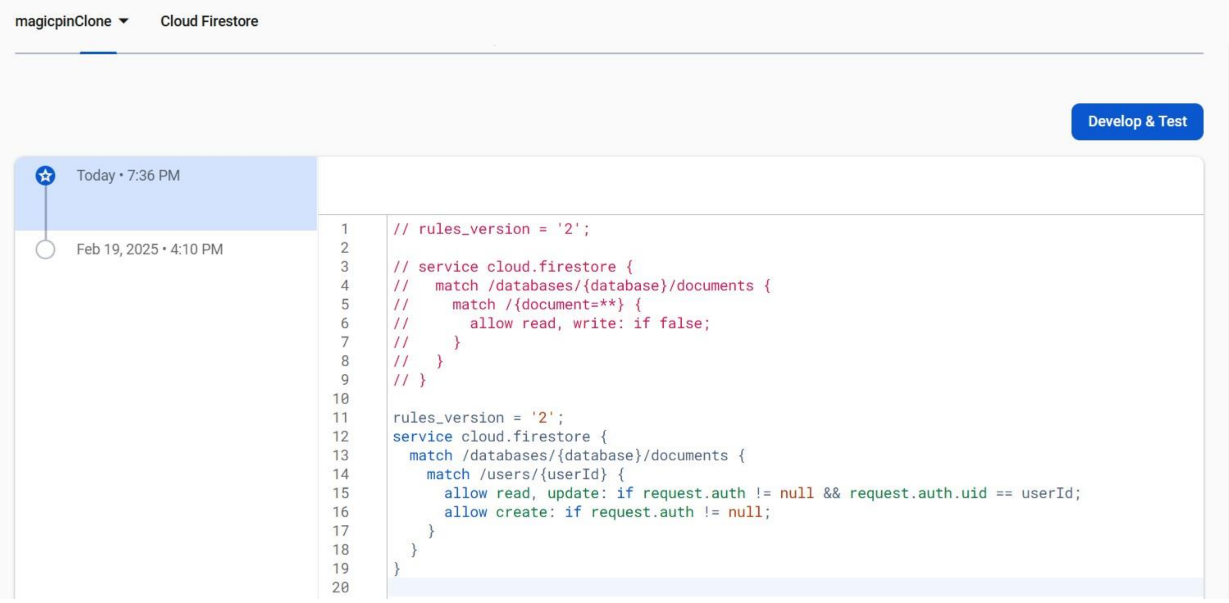
Step 7: Enable Firestore Database in Firebase

1. In Firebase Console, go to **Firestore Database**.
2. Click **Create Database**, select **Start in Test Mode**, and enable Firestore.



Database ID: (default)

Location: nam5 (United States)



```
// rules_version = '2';

// service cloud.firestore {
//   match /databases/{database}/documents {
//     match /{document=**} {
//       allow read, write: if false;
//     }
//   }
// }

rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /users/{userId} {
      allow read, update: if request.auth != null && request.auth.uid == userId;
      allow create: if request.auth != null;
    }
  }
}
```

Step 8: Perform Database Operations in Flutter

1. Add Data to Firestore

```
import 'package:cloud_firestore/cloud_firestore.dart';

void addUser() {

FirebaseFirestore.instance.collection('users').add({

    'name': 'John Doe',

    'email': 'johndoe@example.com',

  });

}
```

2. Read Data from Firestore

```
void fetchUsers() {

  FirebaseFirestore.instance.collection('users').get().then((snapshot) {

  for (var doc in snapshot.docs) {

      print(doc.data());

    }

  });

}
```

3. Update Data in Firestore

```
void updateUser(String docID) {

  FirebaseFirestore.instance.collection('users').doc(docID).update({

  'name': 'Jane Doe',

  });

}
```

4. Delete Data from Firestore

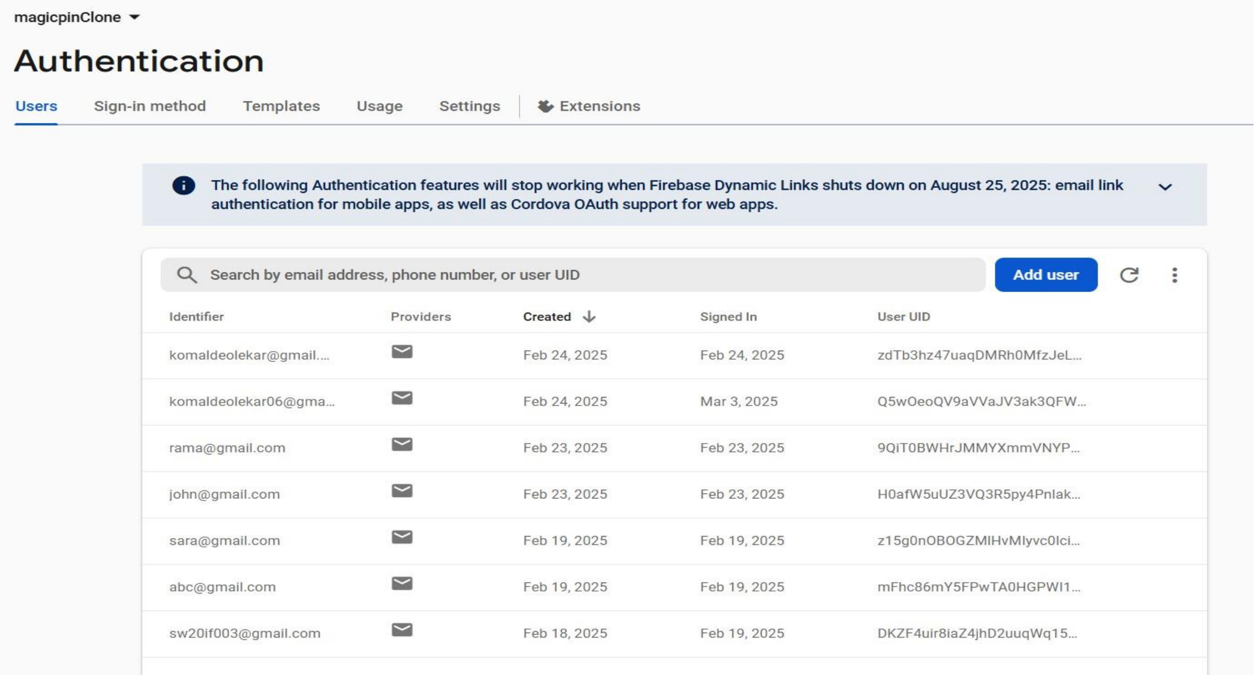
```
void deleteUser(String docID) {

  FirebaseFirestore.instance.collection('users').doc(docID).delete();

}
```

Step 9: Test Firebase Connection

Now update the codes and run the app using:

```
flutter run
```

1. Perform actions like adding, reading, updating, and deleting data.
2. Check Firebase Console to verify the changes.

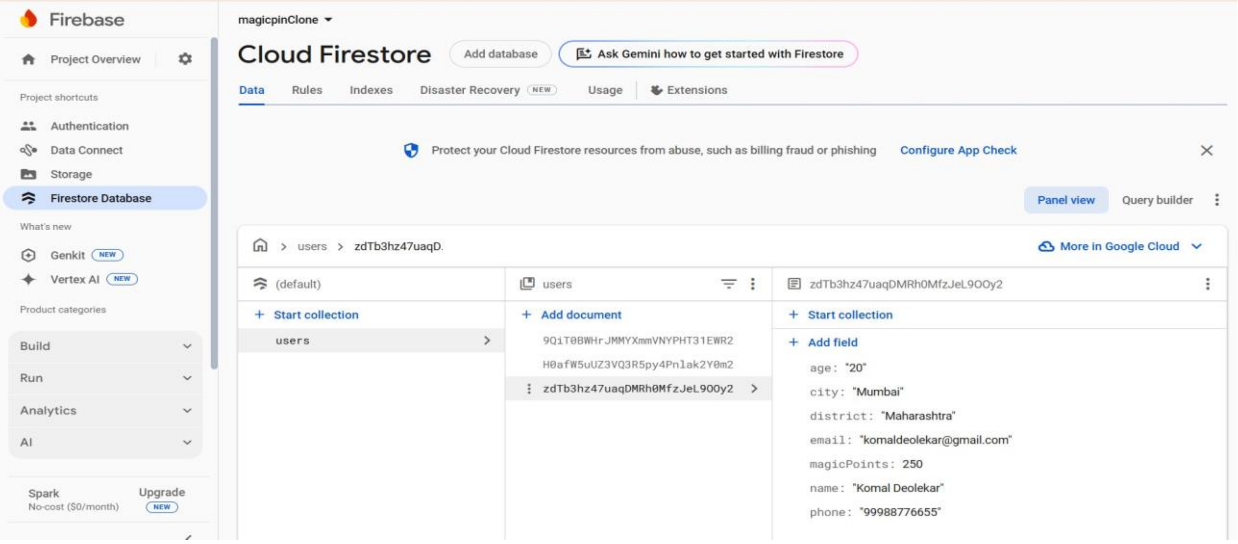


Authentication

The following Authentication features will stop working when Firebase Dynamic Links shuts down on August 25, 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.

Search by email address, phone number, or user UID

Identifier	Providers	Created ↓	Signed In	User UID
komaldeolekar@gmail....	✉	Feb 24, 2025	Feb 24, 2025	zdTb3hz47uaqDMRh0MfzJeL...
komaldeolekar06@gma...	✉	Feb 24, 2025	Mar 3, 2025	Q5wOeoQV9aVVaJV3ak3QFW...
rama@gmail.com	✉	Feb 23, 2025	Feb 23, 2025	9QiT0BWHrJMMYXmmVNYP...
john@gmail.com	✉	Feb 23, 2025	Feb 23, 2025	H0afW5uUZ3VQ3R5py4Pnlak...
sara@gmail.com	✉	Feb 19, 2025	Feb 19, 2025	z15g0nOBOGZMlHvMlyvc0lci...
abc@gmail.com	✉	Feb 19, 2025	Feb 19, 2025	mFhc86mY5FPwTA0HGPWI1...
sw20if003@gmail.com	✉	Feb 18, 2025	Feb 19, 2025	DKZF4uir8iaZ4jhD2uuqWq15...



Cloud Firestore

Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing

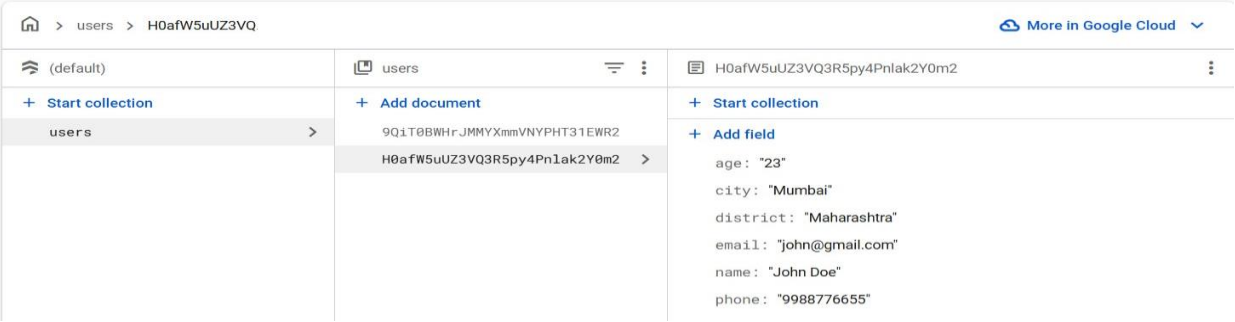
users > zdTb3hz47uaqD.

(default)
- users

users
- 9QiT0BWHrJMMYXmmVNYPHT31EWR2
- H0afW5uUZ3VQ3R5py4Pnlak2Y0m2
- zdTb3hz47uaqDMRh0MfzJeL900y2

zdTb3hz47uaqDMRh0MfzJeL9OOy2

age: "20"
city: "Mumbai"
district: "Maharashtra"
email: "komaldeolekar@gmail.com"
magicPoints: 250
name: "Komal Deolekar"
phone: "99988776655"



users > H0afW5uUZ3VQ.

(default)
- users

users
- 9QiT0BWHrJMMYXmmVNYPHT31EWR2
- H0afW5uUZ3VQ3R5py4Pnlak2Y0m2

H0afW5uUZ3VQ3R5py4Pnlak2Y0m2

age: "23"
city: "Mumbai"
district: "Maharashtra"
email: "john@gmail.com"
name: "John Doe"
phone: "9988776655"

Codes :

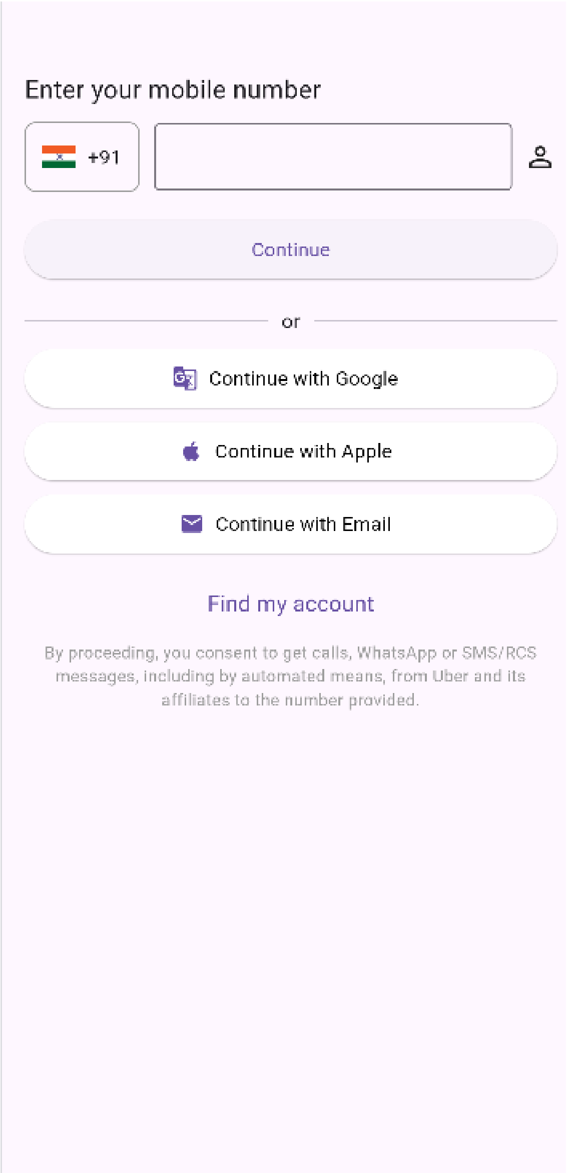
Login_page.dart

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'otp_page.dart';
import 'signup_page.dart';

class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final TextEditingController _phoneController = TextEditingController();
  // Initialize Firebase Auth
  final FirebaseAuth _auth = FirebaseAuth.instance;
  bool _isLoading = false;

  @override
  void initState() {
    super.initState();
    // Check if user is already logged in
    _auth.authStateChanges().listen((User? user) {
      if (user != null) {
        // User is signed in, you can navigate to home page if needed
        // Navigator.pushReplacementNamed(context, '/home');
      }
    });
  }

  void _navigateToOtpPage() {
    if (_phoneController.text.length == 10) {
      setState(() {
        _isLoading = true;
      });

      // Format phone number with country code
      String phoneNumber = '+91${_phoneController.text}';

      _auth.verifyPhoneNumber(
        phoneNumber: phoneNumber,
        verificationCompleted: (PhoneAuthCredential credential) async {
          // Auto-verification on Android
          await _auth.signInWithCredential(credential);
          setState(() {
            _isLoading = false;
          });
          // Navigate to home page or wherever you want
          // Navigator.pushReplacementNamed(context, '/home');
        },
        verificationFailed: (FirebaseAuthException e) {
          setState(() {
            _isLoading = false;
          });
          ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(content: Text(e.message ?? "Verification failed")),
          );
        },
        codeSent: (String verificationId, int? resendToken) {
          setState(() {
            _isLoading = false;
          });
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => OtpPage(
                phoneNumber: _phoneController.text,
                verificationId: verificationId,
              ),
            ),
          );
        },
        codeAutoRetrievalTimeout: (String verificationId) {},
      );
    } else {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text("Please enter a valid 10-digit number")),
      );
    }
  }

  void _navigateToSignupPage() {
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => SignupPage()),
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: _isLoading
            ? Center(child: CircularProgressIndicator())
            : Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  SizedBox(height: 50),
                  Text("Enter your mobile number",
                      style: TextStyle(fontSize: 18)),
                  SizedBox(height: 10),
                  Row(
                    children: [
                      Container(
                        decoration: BoxDecoration(
                          border: Border.all(color: Colors.grey),
                          borderRadius: BorderRadius.circular(8),
                        ),
                        padding:
                            EdgeInsets.symmetric(horizontal: 12, vertical: 14),
                        child: Row(
                          children: [
                            Image.asset("/india_flag.png",
                                width: 24), // Update asset path
                            SizedBox(width: 8),
                            Text("+91"),
                          ],
                        ),
                      ),
                      SizedBox(width: 10),
                      Expanded(
                        child: TextField(
                          controller: _phoneController,
                          keyboardType: TextInputType.phone,
                          decoration: InputDecoration(
                            border: OutlineInputBorder(),
                          ),
                        ),
                      ),
                      SizedBox(width: 8),
                      Icon(Icons.person_outline),
                    ],
                  ),
                  SizedBox(height: 20),
                  ElevatedButton(
                    onPressed: _navigateToOtpPage,
                    child: Text("Continue"),
                    style: ElevatedButton.styleFrom(
                      minimumSize: Size(double.infinity, 50),
                    ),
                  ),
                  SizedBox(height: 20),
                  Row(children: [
                    Expanded(child: Divider()),
                    Padding(
                      padding: const EdgeInsets.symmetric(horizontal: 10),
                      child: Text("or"),
                    ),
                    Expanded(child: Divider()),
                  ]),
                  SizedBox(height: 10),
                  ElevatedButton.icon(
                    onPressed: _navigateToSignupPage,
                    icon: Icon(Icons.g_translate),
                    label: Text("Continue with Google"),
                    style: ElevatedButton.styleFrom(
                      backgroundColor: Colors.white,
                      foregroundColor: Colors.black,
                      minimumSize: Size(double.infinity, 50),
                    ),
                  ),
                  SizedBox(height: 10),
                  ElevatedButton.icon(
                    onPressed: _navigateToSignupPage,
                    icon: Icon(Icons.apple),
                    label: Text("Continue with Apple"),
                    style: ElevatedButton.styleFrom(
                      backgroundColor: Colors.white,
                      foregroundColor: Colors.black,
                      minimumSize: Size(double.infinity, 50),
                    ),
                  ),
                  SizedBox(height: 10),
                  ElevatedButton.icon(
                    onPressed: _navigateToSignupPage,
                    icon: Icon(Icons.email),
                    label: Text("Continue with Email"),
                    style: ElevatedButton.styleFrom(
                      backgroundColor: Colors.white,
                      foregroundColor: Colors.black,
                      minimumSize: Size(double.infinity, 50),
                    ),
                  ),
                  SizedBox(height: 20),
                  Center(
                    child: TextButton(
                      onPressed: _navigateToSignupPage,
                      child: Text("Find my account",
                          style: TextStyle(fontSize: 16)),
                    ),
                  ),
                  SizedBox(height: 10),
                  Text(
                    "By proceeding, you consent to get calls, WhatsApp or SMS/RCS messages, including by automated means, from Uber and its affiliates to the number provided.",
                    textAlign: TextAlign.center,
                    style: TextStyle(fontSize: 12, color: Colors.grey),
                  ),
                ],
              ),
      ),
    );
  }
}
```

Otp_page.dart

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'accept_page.dart';

class OtpPage extends StatefulWidget {
  final String phoneNumber;
  final String verificationId; // Added verificationId

  OtpPage({required this.phoneNumber, required this.verificationId});

  @override
  _OtpPageState createState() => _OtpPageState();
}

class _OtpPageState extends State<OtpPage> {
  final TextEditingController _otpController = TextEditingController();
  final FirebaseAuth _auth = FirebaseAuth.instance;
  bool _isLoading = false;

  void _verifyOtp() async {
    String smsCode = _otpController.text.trim();

    if (smsCode.length == 6) {
      setState(() {
        _isLoading = true;
      });

      try {
        // Create PhoneAuthCredential
        PhoneAuthCredential credential = PhoneAuthProvider.credential(
          verificationId: widget.verificationId,
          smsCode: smsCode,
        );

        // Sign in the user
        await _auth.signInWithCredential(credential);

        setState(() {
          _isLoading = false;
        });

        // Navigate to AcceptPage after successful login
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(builder: (context) => AcceptPage()),
        );
      } catch (e) {
        setState(() {
          _isLoading = false;
        });
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text("Invalid OTP. Please try again.")),
        );
      }
    } else {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text("Enter a valid 6-digit OTP")),
      );
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Verify OTP")),
      body: Padding(
   padding: const EdgeInsets.all(16.0),
   child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
     Text(
      "Enter the 6-digit code sent via SMS to ${widget.phoneNumber}",
      style: TextStyle(fontSize: 18),
     ),
     SizedBox(height: 10),
     TextField(
      controller: _otpController,
      keyboardType: TextInputType.number,
      maxLength: 6, // Updated OTP length to 6 digits
      textAlign: TextAlign.center,
      decoration: InputDecoration(counterText: ""),
     ),
     SizedBox(height: 20),
     TextButton(onPressed: () {}, child: Text("Resend code via SMS")),
     TextButton(onPressed: () {}, child: Text("Call me with code")),
     Spacer(),
     _isLoading
       ? Center(child: CircularProgressIndicator())
       : Row(
         mainAxisAlignment: MainAxisAlignment.spaceBetween,
         children: [
          IconButton(
           icon: Icon(Icons.arrow_back),
           onPressed: () => Navigator.pop(context),
          ),
          ElevatedButton(
           onPressed: _verifyOtp,
           child: Text("Verify & Continue"),
          ),
         ],
        ),
    ],
   ),
  ),
 );
}
```

Project Title: Uber App **Roll No.: 12**



Enter your mobile number

+91

Continue

or

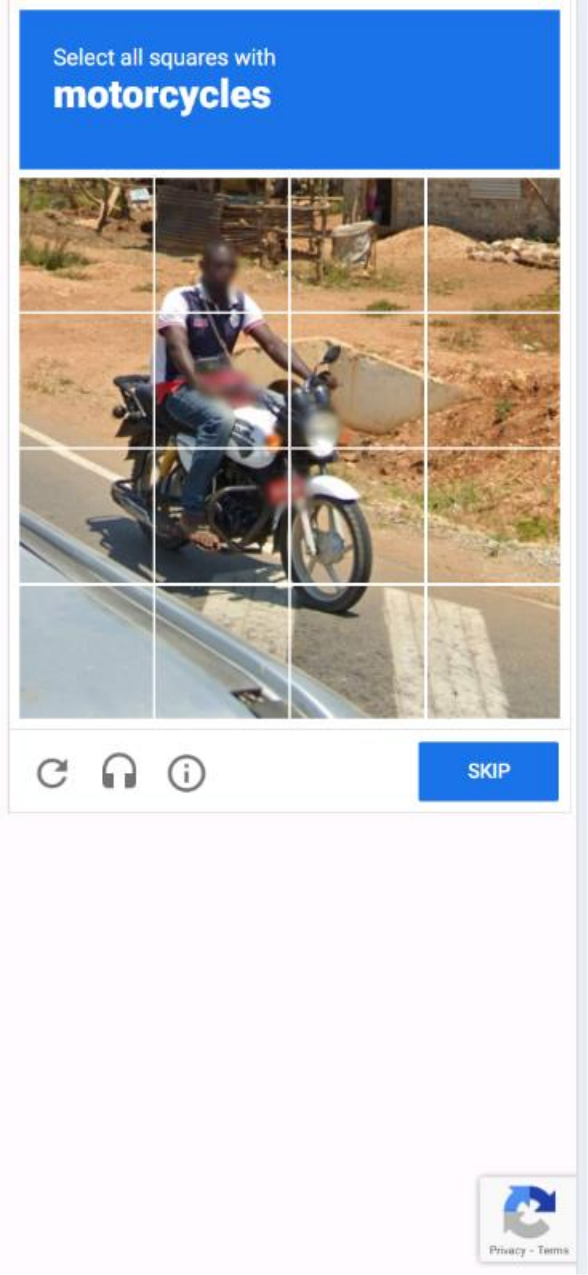
Continue with Google

Continue with Apple

Continue with Email

Find my account

By proceeding, you consent to get calls, WhatsApp or SMS/RCS messages, including by automated means, from Uber and its affiliates to the number provided.



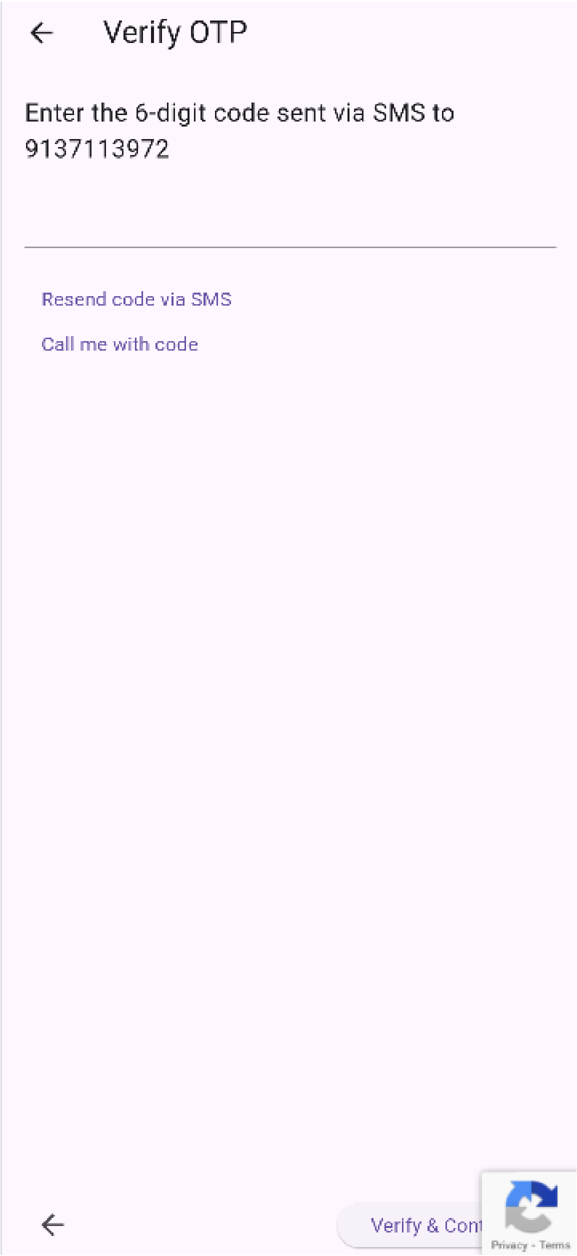
Select all squares with

motorcycles

SKIP

Privacy - Terms

ScreenShots :



Verify OTP

Enter the 6-digit code sent via SMS to 9137113972