

# MAD and PWA Lab

**Name: Sarvadnya Awaghad**

**Class: D15A**

**Roll no:04**

## Experiment - 6

**Aim: To Connect Flutter UI with FireBase database**

### Theory:

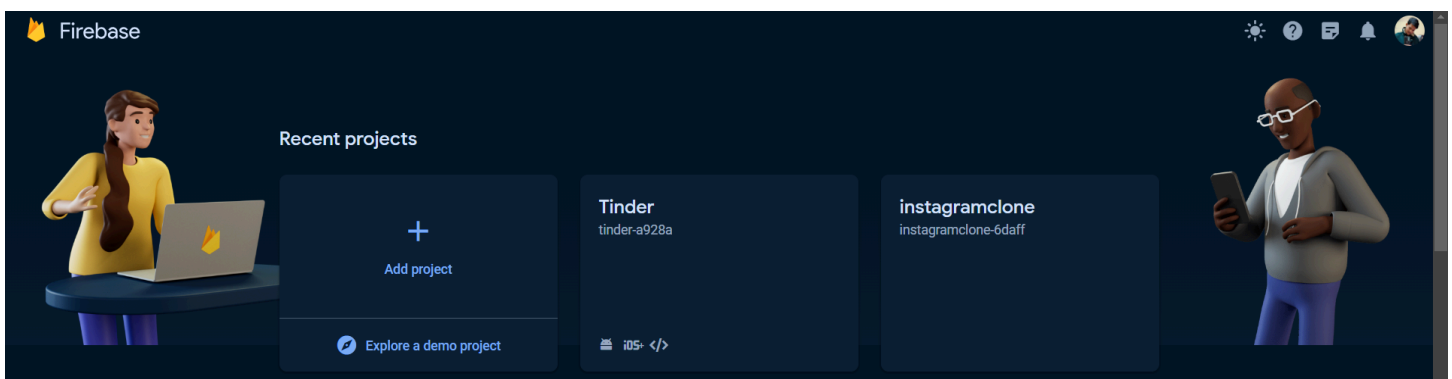
### Prerequisites

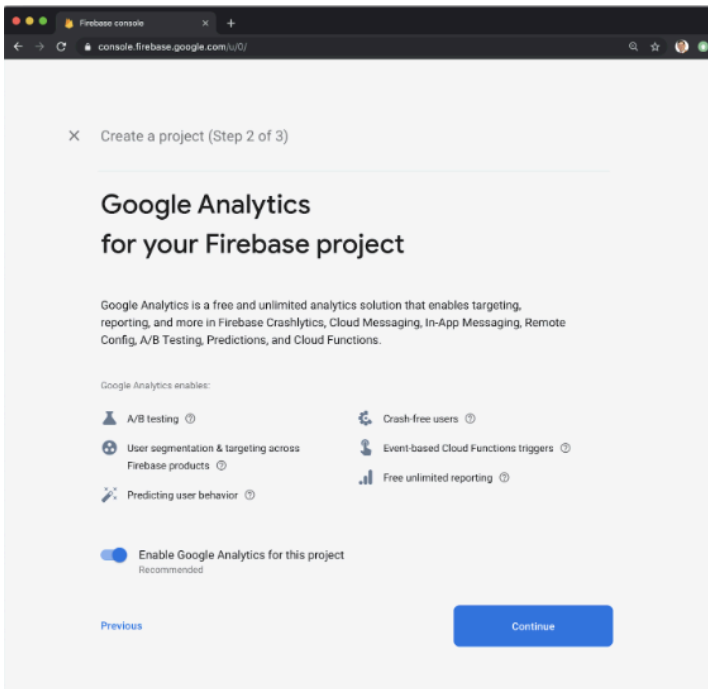
To complete this tutorial, you will need:

- A Google account to use Firebase.
- Developing for iOS will require XCode.
- To download and install Flutter.
- To download and install Android Studio and Visual Studio Code.
- It is recommended to install plugins for your code editor:
  - [Flutter](#) and [Dart](#) plugins installed for Android Studio.
  - [Flutter](#) extension installed for Visual Studio Code.

### Create a Firebase Project:

First, log in with your Google account to manage your Firebase projects. From within the Firebase dashboard, select the Create new project button and give it a name:





Go to the Firebase Console and create a new project.  
Add your Flutter app to the Firebase project:

Register your app in the Firebase project, and follow the instructions to download the configuration files (google-services.json for Android, GoogleService-Info.plist for iOS).

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

Once you've decided on a name, open android/app/build.gradle in your code editor and update the applicationId to match the Android package name:

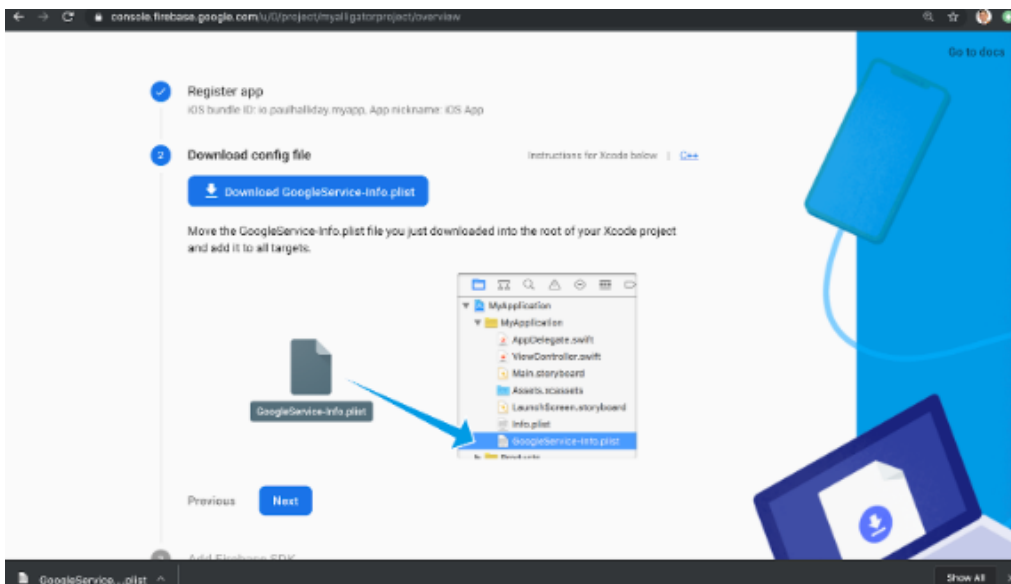
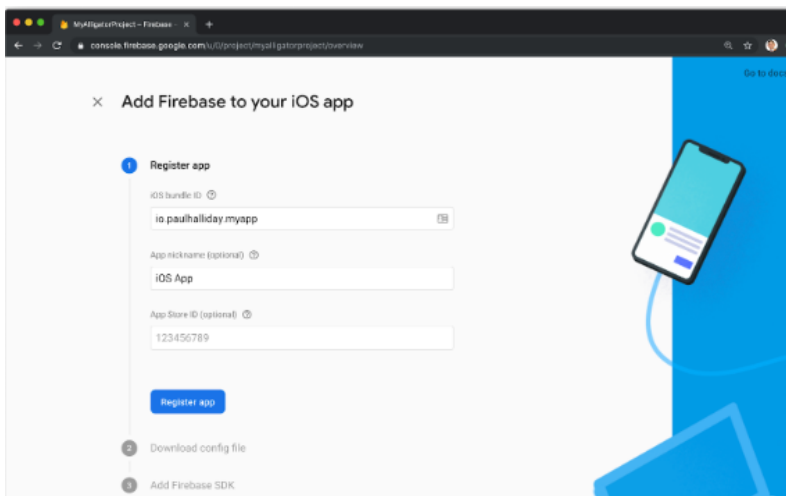
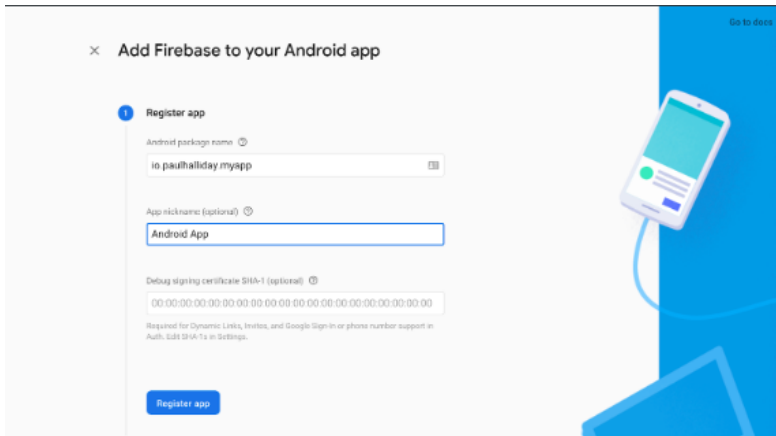
android/app/build.gradle

```
...
defaultConfig {
    // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).
    applicationId 'com.example.flutterfirebaseexample'
}
...
```

## Downloading the Config File

The next step is to add the Firebase configuration file into our Flutter project. This is important as it contains the API keys and other critical information for Firebase to use.

Select Download `google-services.json` from this page:



2. Add Firebase to your Flutter project:  
Add Dependencies:

Open your pubspec.yaml file and add the necessary dependencies:

yaml

```
dependencies:
  flutter:
    sdk: flutter
  cupertino_icons: ^1.0.2

  swipe_cards: ^2.0.0+1
  google_fonts: ^6.1.0
  flutter_svg: ^2.0.9
  cached_network_image: ^3.3.1
  font_awesome_flutter: ^10.7.0
  faker: ^2.1.0
  flutter_custom_clippers: ^2.1.0
  firebase_core: ^2.25.3
  firebase_auth: ^4.17.3
  cloud_firestore: ^4.15.4
  google_sign_in: ^6.2.1
```

Code:

main.dart

```
import 'package:firebase_core/firebase_core.dart';
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(MyApp());
}
```

login.dart

```
import 'dart:developer';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_sign_in/google_sign_in.dart';
import 'package:tinder/authenticationScreen/main_page.dart';
import 'package:tinder/authenticationScreen/registration_screen.dart';
import 'package:tinder/widgets/custom_text_field_widget.dart';
class LoginScreen extends StatefulWidget {
  const LoginScreen({Key? key}) : super(key: key);
  @override
```

```

_LoginScreenState createState() => _LoginScreenState();
}
class _LoginScreenState extends State<LoginScreen> {
  TextEditingController emailTextEditingController = TextEditingController();
  TextEditingController passwordTextEditingController = TextEditingController();
  bool showProgressBar = false;
  void login() async {
    String email = emailTextEditingController.text.trim();
    String password = passwordTextEditingController.text.trim();
    if (email == "" || password == "") {
      print('please fill all details');
    } else {
      try {
        UserCredential userCredential = await FirebaseAuth.instance
          .signInWithEmailAndPassword(email: email, password: password);
        if (userCredential != null) {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => MainPage()),
          );
        }
      } on FirebaseAuthException catch (ex) {
        log(ex.code.toString());
      }
    }
  }
  _signInWithGoogle() async {
    final GoogleSignIn _googleSignIn = GoogleSignIn();
    try {
      final GoogleSignInAccount? googleSignInAccount =
        await _googleSignIn.signIn();
      if (googleSignInAccount != null) {
        final GoogleSignInAuthentication googleSignInAuthentication =
          await googleSignInAccount.authentication
        final AuthCredential credential = GoogleAuthProvider.credential(
          idToken: googleSignInAuthentication.idToken,
          accessToken: googleSignInAuthentication.accessToken,
        );
      }
    }
  }
}

```

```

    await FirebaseAuth.instance.signInWithCredential(credential);
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => MainPage()),
    );
  }
} catch (e) {
  log("some error occurred $e");
}
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      decoration: BoxDecoration(
        gradient: LinearGradient(
          colors: [Colors.pink, Colors.orange],
          begin: Alignment.topLeft,
          end: Alignment.bottomRight,
        ),
      ),
    child: SingleChildScrollView(
      child: Center(
        child: Column(
          children: [
            const SizedBox(
              height: 220,
            ),
            Image.asset(
              'images/tinder_logo.png',
              width: 100,
            ),
            const SizedBox(
              height: 60,
            ),
            Center(
              child: Padding(

```

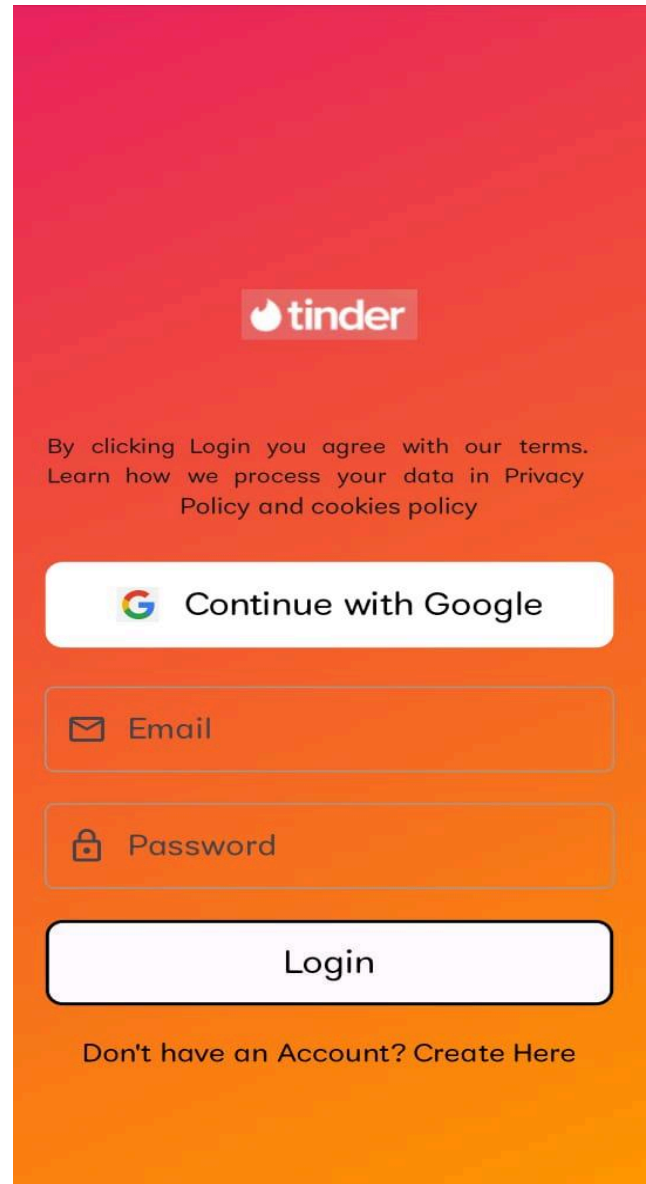
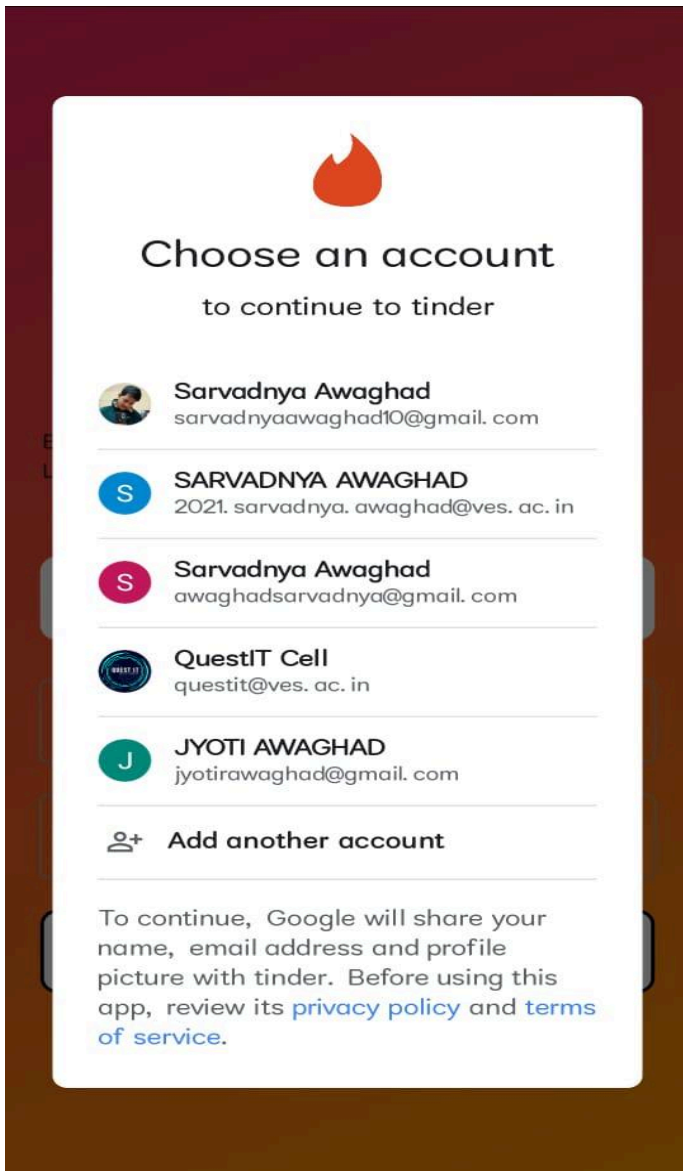
```

padding: EdgeInsets.symmetric(
  horizontal: 20,
),
child: Text(
  "By clicking Login you agree with our terms. Learn how we process
your data in Privacy ",
  style: TextStyle(
    fontSize: 13,
    wordSpacing: 5,
  ),
),
),
),
Text('Policy and cookies policy'),
const SizedBox(
  height: 25,
),
ElevatedButton(
  onPressed: () {
    // Implement your Google login logic here
    _signInWithGoogle();
  },
  style: ElevatedButton.styleFrom(
    primary: Colors.white,
    onPressed: Colors.black,
    elevation: 0,
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(10),
    ),
  )
)
showProgressbar
? const CircularProgressIndicator(
  valueColor: AlwaysStoppedAnimation<Color>(Colors.pink),
)
: Container()
],
),
),

```

```
),  
,  
);  
}  
}
```

Output:





Firebase Authentication interface showing a list of users.

Search by email address, phone number or user UID

Identifier	Providers	Created	Signed In	User UID
sarvadnyaawaghad10...	Google	13 Feb 2024	13 Feb 2024	ual4EficNyYspqV2RV0BzDlyb...
shubh@gmail.com	Google	12 Feb 2024	12 Feb 2024	foNUfxqNrSbHMCWEw9dacF...
sarv3@gmail.com	Google	9 Feb 2024	9 Feb 2024	1RKZiZuxz6XoETpRocoFwE4...
sarv2@gmail.com	Google	9 Feb 2024	9 Feb 2024	j7g9BKPZbgT6HRCgJoMNmo...
sarv@gmail.com	Google	9 Feb 2024	13 Feb 2024	eYDmw8JpSTXieBoVSZ7KtAq...

Rows per page: 50 | 1 - 5 of 5

Firebase Build interface showing performance metrics for the Tinder app.

Tinder Spark plan

4 apps | com.example.tinder | + Add app

### Build

Firestore

Reads (current): 13

Writes (current): 5

Timeline: Feb 5 to Feb 11

Legend: — This week — Last week

## Conclusion:

In this experiment, we have successfully connected firebase database and authenticated using google signin and email and password with out flutter application successfully.