|  |  |
| --- | --- |
| Experiment No.: 6 | Connect Flutter UI with firebase database |
| Aim: | To Connect Flutter UI with firebase database |
| Lab Outcome | Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS. |

.

Step 1: Create Flutter project as firebase\_demo

Step 2: Add dependencies

firebase\_auth: ^1.0.1 # add this line

firebase\_core: ^1.0.2 # add this line

Step 3: run the flutter pub get command

Step 4: Visit the firebase.google.com and perform the following steps:

1. Visit the Firebase Console.
2. Start by adding a New Project
3. Giving the project a name and agreeing to all they’ve asked
4. Click ‘Create Project.’

Once you’re done creating the project, it’s time to integrate it with Android and iOS applications.

Step 5: **Configure Firebase with Android and iOS Application**

**Register Android application**

For registering the android app, you have to provide a unique package name. For that, you can find it at *android>app>build.gradle*, by default it will take *com.example.application\_name*. The next **field App nickname** is optional.

**Download Android Config file**

After filling the required fields, you’ll see something like this. Download *google-services.json* and place it in *MyApplication/app* folder.

You can skip the next step as we are using: [*Firebase core*](https://pub.dev/packages/firebase_core)*and*[*Firebase auth*](https://pub.dev/packages/firebase_auth)*packages*. Click ‘Continue to Console’ to complete the process.

**Enable Email/Password Authentication in Firebase**

On visiting the Firebase dashboard, click ‘Authentication.’

Under the Sign-in method click ‘Email/Password’ and enable it using the toggle button. Use the below screenshots for your reference.

**Initialize Firebase App**

Open *main.dart* and use the following code snippet to initialize Firebase App.

future main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp();

runApp(MyApp());

}

Create a file authentication.dart and use the following code for creating helper class for authentication.

import 'package:firebase\_auth/firebase\_auth.dart';

class AuthenticationHelper {

final FirebaseAuth \_auth = FirebaseAuth.instance;

get user => \_auth.currentUser;

//SIGN UP METHOD

Future signUp({String email, String password}) async {

try {

await \_auth.createUserWithEmailAndPassword(

email: email,

password: password,

);

return null;

} on FirebaseAuthException catch (e) {

return e.message;

}

}

//SIGN IN METHOD

Future signIn({String email, String password}) async {

try {

await \_auth.signInWithEmailAndPassword(email: email, password: password);

return null;

} on FirebaseAuthException catch (e) {

return e.message;

}

}

//SIGN OUT METHOD

Future signOut() async {

await \_auth.signOut();

print('signout');

}

}

**Signup.dart**

// Get username and password from the user.Pass the data to

// helper method

AuthenticationHelper()

.signUp(email: email, password: password)

.then((result) {

if (result == null) {

Navigator.pushReplacement(context,

MaterialPageRoute(builder: (context) => Home()));

} else {

Scaffold.of(context).showSnackBar(SnackBar(

content: Text(

result,

style: TextStyle(fontSize: 16),

),

));

}

});

**Login.dart**

AuthenticationHelper()

.signIn(email: email, password: password)

.then((result) {

if (result == null) {

Navigator.pushReplacement(context,

MaterialPageRoute(builder: (context) => Home()));

} else {

Scaffold.of(context).showSnackBar(SnackBar(

content: Text(

result,

style: TextStyle(fontSize: 16),

),

));

}

});