

Experiment – 6: Connecting Flutter UI with firebase database.

Aim: To connect Flutter UI with firebase database.

Objectives: After study of this experiment, the student will be able to

- Create a production ready Flutter App by including files and firebase backend service.

Outcomes: After study of this experiment, the student will be able to

- Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android/iOS

Prerequisite: Dart Programming Language

Requirements: Android Studio, Flutter framework, Internet Connection.

Pre-Experiment Exercise:

Brief Theory:

Firebase is a platform developed by Google for creating mobile and web applications. Firebase is a set of server-side services and tools. If you use Firebase, you don't need to buy or rent your own server. Firebase is made up of over a dozen tools. Let's glance at these three:

- Cloud Firestore – A database with an API to read and write data
- Cloud Functions – Logic that is kicked off by an API call
- Authentication – Single sign-on to allow users to securely log in to your app using their social accounts or a username/password combination

Steps to connect UI with firebase database:

1. Visit <http://firebase.google.com> to register an app with Firebase.
2. Create a Firebase Project.
3. Create the database by using the option given on your project's dashboard.
4. Create an Android app.
 - a. Going through the next step while creating an android app, the wizard will create a google-services.json file and will tell you where to save it.
5. Add FlutterFire plugins.

Laboratory Exercise

A. Program

1. Connect your UI with the firebase database and show the values entered in UI being reflected in the database.

B. Result/Observation

Print out of program code and output.

Post-Experiments Exercise

A. Questions:

1. What are the other different databases that you can use to connect your UI?
2. Demonstrate the use of firebase Authentication using UI.

B. Conclusion:

1. Write what you have learnt in the experiment.

C. References:

1. Beginning App Development with Flutter: Create Cross-Platform Mobile Apps, By Rap Payne, 2019.
2. Google Flutter Mobile Development Quick Start Guide, Packt Publishing, 2019.

Login.dart

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

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

  @override
  State<MyLogin> createState() => _MyLoginState();
}

class _MyLoginState extends State<MyLogin> {
  TextEditingController email = TextEditingController();
  TextEditingController password = TextEditingController();

  Future<bool> handleAuth() async {
    try {
      await
      FirebaseAuth.instance.signInWithEmailAndPassword(
        email: email.text.trim(),
        password: password.text.trim(),
      );
      return true;
    } catch (e) {
      print('Auth error: $e');
      return false;
    }
  }

  @override
  Widget build(BuildContext context) {
    return Container(
      decoration: BoxDecoration(
        image: DecorationImage(
          image: AssetImage('assets/login.png'),
          fit: BoxFit.cover,
        ),
      ),
      child: Scaffold(
        backgroundColor: Colors.transparent,
        body: Stack(
          children: [
            Container(
              padding: EdgeInsets.only(left: 80, top: 130),
              child: Text(
                "Welcome \nback",
                style: TextStyle(color: Colors.white, fontSize: 33),
              ),
            ),
            SingleChildScrollView(
              child: Container(
                padding: EdgeInsets.only(
                  top: MediaQuery.of(context).size.height * 0.5,
                  right: 35,
                  left: 35,
                ),
                child: Column(
```

```
                SizedBox(height: 40),
                Row(
                  mainAxisAlignment:
                    MainAxisAlignment.spaceBetween,
                  children: [
                    Text(
                      "Sign In",
                      style: TextStyle(
                        color: Color(0xff4c505b),
                        fontSize: 27,
                        fontWeight: FontWeight.w700,
                      ),
                    ),
                    CircleAvatar(
                      radius: 30,
                      backgroundColor: Color(0xff4c505b),
                      child: IconButton(
                        color: Colors.white,
                        onPressed: () async {
                          bool isLoggedIn = await handleAuth();
                          if (isLoggedIn) {
                            Navigator.pushReplacementNamed(context,
                              'home');

                            ScaffoldMessenger.of(context).showSnackBar(
                              SnackBar(
                                content: Text("Login in Successfull"),
                              ),
                            );
                          } else {
                            ScaffoldMessenger.of(context).showSnackBar(
                              SnackBar(content: Text("Login failed")),
                            );
                          }
                        },
                        icon: Icon(Icons.arrow_forward),
                      ),
                    ),
                  ],
                ),
                SizedBox(height: 40),
                Row(
                  mainAxisAlignment:
                    MainAxisAlignment.spaceBetween,
                  children: [
                    TextButton(
                      onPressed: () {
                        Navigator.pushNamed(context, 'register');
                      },
                      child: Text(
                        "Sign Up",
                        style: TextStyle(
                          decoration: TextDecoration.underline,
                          fontSize: 18,
                          color: Color(0xff4c505b),
                        ),
                    ),
```

<pre> children: [TextField(controller: email, decoration: InputDecoration(fillColor: Colors.grey.shade100, filled: true, hintText: "Email", border: OutlineInputBorder(borderRadius: BorderRadius.circular(10),),),), SizedBox(height: 30), TextField(controller: password, obscureText: true, decoration: InputDecoration(fillColor: Colors.grey.shade100, filled: true, hintText: "Password", border: OutlineInputBorder(borderRadius: BorderRadius.circular(10),),),),], </pre>	<pre>), TextButton(onPressed: () {}, child: Text("Forgot Password", style: TextStyle(decoration: TextDecoration.underline, fontSize: 18, color: Color(0xff4c505b),),),),],),],),],),),); } } </pre>
--	--

Main.dart

```
import 'package:firebase/home.dart';
import 'package:firebase/login.dart';
import 'package:firebase/register.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(
    MaterialApp(
      debugShowCheckedModeBanner: false,
      initialRoute: 'login',
      routes: {
        'login': (context) => MyLogin(),
        'register': (context) => MyRegister(),
        'home': (context) => HomePage(),
      },
    ),
  );
}
```

Register.dart

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

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

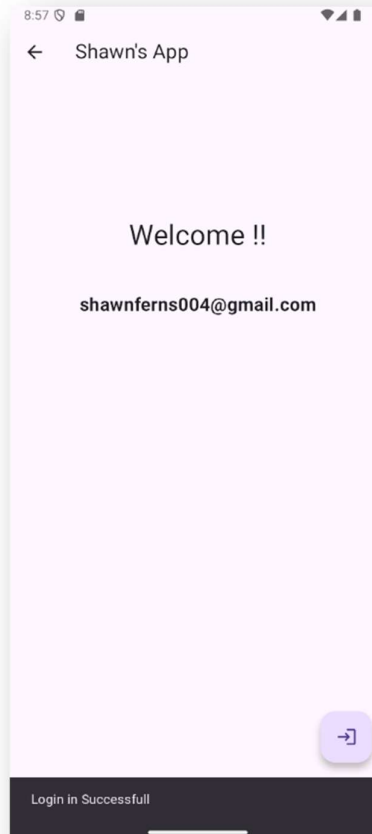
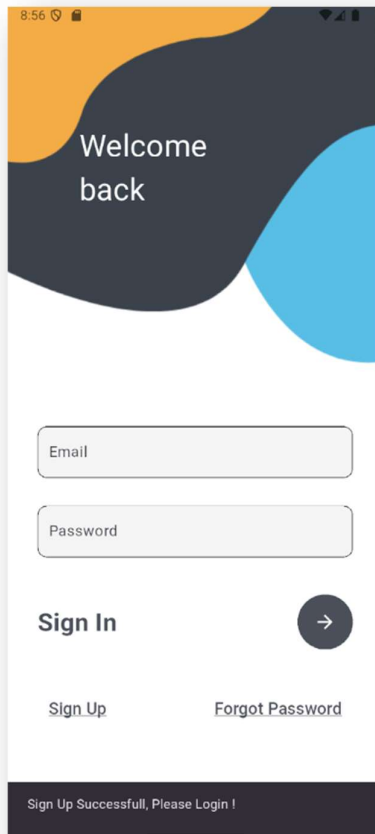
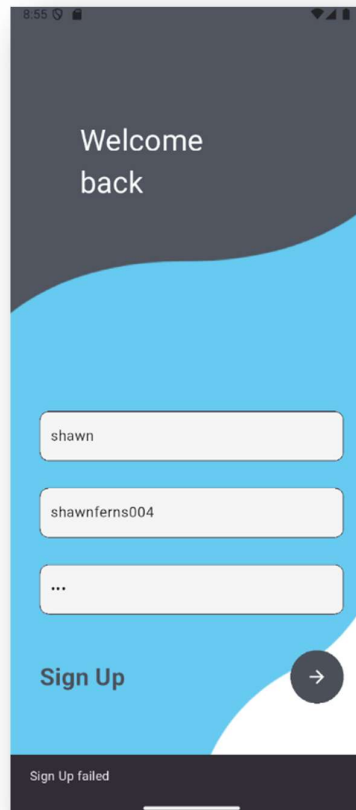
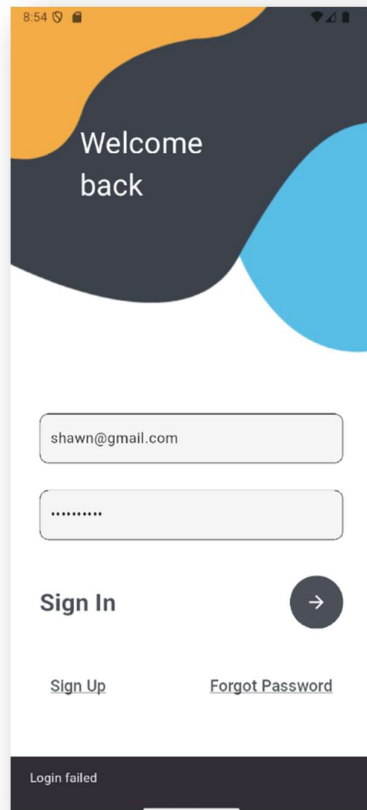
  @override
  State<MyRegister> createState() => _MyRegisterState();
}

class _MyRegisterState extends State<MyRegister> {
  TextEditingController email = TextEditingController();
  TextEditingController password = TextEditingController();

  Future<bool> handleAuth() async {
    try {
      await
        FirebaseAuth.instance.createUserWithEmailAndPassword(
          email: email.text.trim(),
          password: password.text.trim(),
        );
      return true;
    } catch (e) {
      print('Auth error: $e');
      return false;
    }
  }

  @override
  Widget build(BuildContext context) {
    return Container(
      decoration: BoxDecoration(
        image: DecorationImage(
          image: AssetImage('assets/register.png'),
          fit: BoxFit.cover,
        ),
      ),
      child: Scaffold(
        backgroundColor: Colors.transparent,
        body: Stack(
          children: [
            Container(
              padding: EdgeInsets.only(left: 80, top: 130),
              child: Text(
                "Welcome \nback",
                style: TextStyle(color: Colors.white, fontSize: 33),
              ),
            ),
            SingleChildScrollView(
              child: Container(
                padding: EdgeInsets.only(
                  top: MediaQuery.of(context).size.height * 0.5,
                  right: 35,
                  left: 35,
                ),
                child: Column(
                  children: [
```

```
                    fillColor: Colors.grey.shade100,
                    filled: true,
                    hintText: "Password",
                    border: OutlineInputBorder(
                      borderRadius: BorderRadius.circular(10),
                    ),
                  ),
                ),
              SizedBox(height: 40),
              Row(
                mainAxisAlignment:
                  MainAxisAlignment.spaceBetween,
                children: [
                  Text(
                    "Sign Up",
                    style: TextStyle(
                      color: Color(0xff4c505b),
                      fontSize: 27,
                      fontWeight: FontWeight.w700,
                    ),
                  ),
                  CircleAvatar(
                    radius: 30,
                    backgroundColor: Color(0xff4c505b),
                    child: IconButton(
                      color: Colors.white,
                      onPressed: () async {
                        bool isLoggedIn = await handleAuth();
                        if (isLoggedIn) {
                          Navigator.pushReplacementNamed(
                            context,
                            'login',
                          );
                        }
                        ScaffoldMessenger.of(context).showSnackBar(
                          SnackBar(
                            content: Text(
                              "Sign Up Successfull, Please Login!",
                            ),
                          ),
                        );
                      } else {
                        ScaffoldMessenger.of(context).showSnackBar(
                          SnackBar(content: Text("Login failed!")),
                        );
                      }
                    },
                    icon: Icon(Icons.arrow_forward),
                  ),
                ],
              ),
              SizedBox(height: 40),
              Row(
                mainAxisAlignment:
                  MainAxisAlignment.spaceBetween,
                children: [
```

Authentication

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

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

Add user

Identifier	Providers	Created ↓	Signed In	User UID
shawnfers004@gmail...		Apr 11, 2025	Apr 11, 2025	pbz2hSaMYJTPcTXX0MSCNF...

Rows per page: 50 1 - 1 of 1