

EXPERIMENT NO 6

NAME-SPANDAN DEB

CLASS-D15A

ROLL NO-13

AIM-To connect Flutter UI with firebase.

THEORY-

Firebase helps developers to manage their mobile app easily. It is a service provided by Google. Firebase has various functionalities available to help developers manage and grow their mobile apps.

Steps to Add firebase to our Flutter app using Firebase CLI

1.Install the Firebase CLI and log in (run firebase login)

2.From any directory, run this command:

- `dart pub global activate flutterfire_cli`

3.Then, at the root of your Flutter project directory, run this command:

- `flutterfire configure --project=questitnextjs`

4.This automatically registers your per-platform apps with Firebase and adds a `lib/firebase_options.dart` configuration file to your Flutter project.

5.To initialise Firebase, call `Firebase.initializeApp` from the `firebase_core` package with the configuration from your new `firebase_options.dart` file:

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

```
import 'firebase_options.dart';  
await Firebase.initializeApp(  
  options: DefaultFirebaseOptions.currentPlatform,  
);
```

6. Add the dependencies in the pubspec.yaml file

```
FlutterFire : ^0.5.4  
firebase_core : ^10.12.0  
firebase_auth : ^10.12.0
```

SYNTAX

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

```
Future<void> signInUser(String email, String password) async {  
  try {  
    await FirebaseAuth.instance.signInWithEmailAndPassword(  
      email: email,  
      password: password,  
    );  
    print("User Signed In Successfully!");  
  } catch (e) {  
    print("Error: $e");  
  }  
}
```

Widget Properties

1)Firebase_Auth

- currentUser → Returns the currently signed-in user.
- signInWithEmailAndPassword(email, password) → Logs in a user.
- createUserWithEmailAndPassword(email, password) → Registers a new user.
- signOut() → Logs out the current user.

2)Firestore

- collection("name") → Accesses a Firestore collection.
- doc("id") → Refers to a specific document.
- set(Map<String, dynamic> data) → Adds or updates data.
- get() → Fetches document data.
- delete() → Deletes a document.

CODE

```
import 'package:flutter/material.dart';  
import 'package:font_awesome_flutter/font_awesome_flutter.dart';  
import 'package:firebase_auth/firebase_auth.dart';  
import 'package:my_app/pages/mainpage.dart';
```

```
class CreateAccount extends StatefulWidget {  
  const CreateAccount({super.key});  
  
  @override  
  _CreateAccountState createState() => _CreateAccountState();  
}
```

```
class _CreateAccountState extends State<CreateAccount> {  
  final _formKey = GlobalKey<FormState>();  
  final _usernameController = TextEditingController();
```

```

final _emailController = TextEditingController();
final _dobController = TextEditingController();
final _passwordController = TextEditingController();
final FirebaseAuth _auth = FirebaseAuth.instance; // Firebase Authentication Instance

bool _isLoading = false;

// Function to handle Firebase registration
Future<void> _registerUser() async {
  if (!_formKey.currentState!.validate()) return;

  setState() {
    _isLoading = true;
  });

  try {
    UserCredential userCredential = await _auth.createUserWithEmailAndPassword(
      email: _emailController.text.trim(),
      password: _passwordController.text.trim(),
    );

    User? user = userCredential.user;
    if (user != null) {
      print("User Registered: ${user.email}");
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text("Account Created Successfully!")),
      );

      // Navigate to Home Page after successful signup

```

```

    Navigator.pushReplacement(
      context,
      MaterialPageRoute(builder: (context) => TwitterHomePage()), // Update with the
correct main page
    );
  }
} on FirebaseAuthException catch (e) {
  print("Firebase Auth Error: ${e.message}");
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text("Error: ${e.message}")),
  );
}

```

```

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

```

@override

```

Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      leading: IconButton(
        icon: const Icon(Icons.arrow_back, color: Colors.black),
        onPressed: () {
          Navigator.pop(context);
        },
      ),
    centerTitle: true,
  );
}

```

```

title: const FaIcon(FontAwesomeIcons.twitter, color: Colors.blue, size: 30),
backgroundColor: Colors.transparent,
elevation: 0,
),
body: Padding(
padding: const EdgeInsets.all(20),
child: Form(
key: _formKey,
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
// Username
const Text("Username", style: TextStyle(fontSize: 16, fontWeight:
FontWeight.bold)),
TextFormField(
controller: _usernameController,
decoration: InputDecoration(
hintText: "Enter your username",
border: OutlineInputBorder(borderRadius: BorderRadius.circular(10)),
),
validator: (value) => value!.isEmpty ? "Username cannot be empty" : null,
),
const SizedBox(height: 15),

// Email
const Text("Email", style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold)),
TextFormField(
controller: _emailController,
decoration: InputDecoration(

```

```
      hintText: "Enter your email",
      border: OutlineInputBorder(borderRadius: BorderRadius.circular(10)),
    ),
    validator: (value) => value!.contains("@") ? null : "Enter a valid email",
  ),
  const SizedBox(height: 15),

  // Date of Birth
  const Text("Date of Birth", style: TextStyle(fontSize: 16, fontWeight:
FontWeight.bold)),
  TextFormField(
    controller: _dobController,
    decoration: InputDecoration(
      hintText: "DD/MM/YYYY",
      border: OutlineInputBorder(borderRadius: BorderRadius.circular(10)),
    ),
    keyboardType: TextInputType.datetime,
  ),
  const SizedBox(height: 15),

  // Password
  const Text("Password", style: TextStyle(fontSize: 16, fontWeight:
FontWeight.bold)),
  TextFormField(
    controller: _passwordController,
    obscureText: true,
    decoration: InputDecoration(
      hintText: "Enter your password",
      border: OutlineInputBorder(borderRadius: BorderRadius.circular(10)),
    ),
  ),
```

```

        validator: (value) => value!.length < 6 ? "Password must be at least 6 characters" :
null,

    ),

  ],

),

),

),

// Floating Action Button for "Next"
floatingActionButton: Padding(
  padding: const EdgeInsets.only(bottom: 30), // Adjust to move the button up
  child: FloatingActionButton(
    onPressed: _isLoading ? null : _registerUser,
    backgroundColor: _isLoading ? Colors.grey : Colors.blue,
    child: _isLoading
      ? const CircularProgressIndicator(color: Colors.white)
      : const Icon(Icons.arrow_forward, color: Colors.white),
  ),
),
floatingActionButtonLocation: FloatingActionButtonLocation.endFloat,
);
}
}

```


OUTPUT

Project Overview

Generative AI

Build with Gemini

Genkit

Project shortcuts

Authentication

Product categories

Build

Run

Analytics

Spark

No-cost (\$0/month)

Upgrade

NEW

Twitter

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

aryan@gmail.com		Feb 10, 2025	Feb 10, 2025	0Sd0li5UsZSJ5jcuZdl0kJP4vd...
nitish@gmail.com		Feb 10, 2025	Feb 10, 2025	3yVYwBR3CVcyJP7ba5UDDkz...
sd13@gmail.com		Feb 9, 2025	Feb 9, 2025	UCK3li8IPUSHrYNNve3MTDYs2...
spandan.deb04@gmail...		Feb 9, 2025	Feb 22, 2025	XY07gtwP38hRfw5eGMavlJ...

Rows per page: 50

1 - 4 of 4