EXPERIMENT NO: - 06

Name:- Himesh Pathai Class:- D15A Roll:No: - 34

AIM: - To connect Flutter UI with Firebase database.

Theory: -

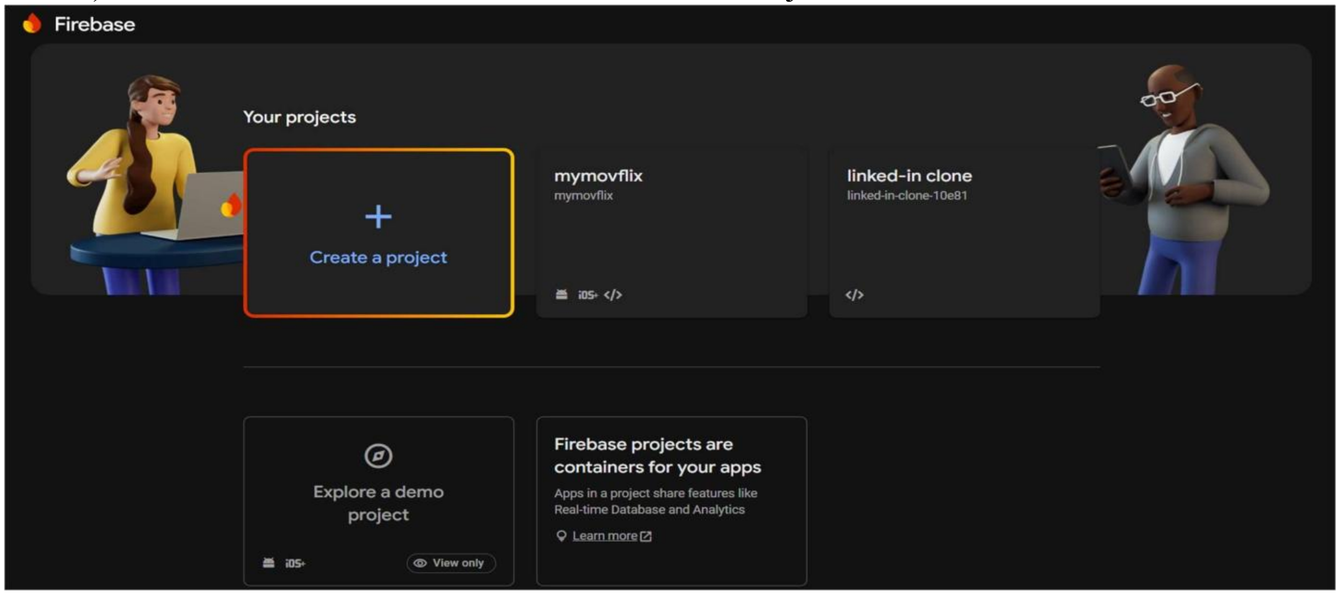
Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. Firebase, a Backend-as-a-Service (BaaS) platform, provides real-time database, authentication, and cloud storage services, making it a powerful backend solution for Flutter applications.

By integrating Firebase with Flutter, developers can store and retrieve data in real time, authenticate users, and manage cloud-based data efficiently. This is particularly useful for applications requiring dynamic content updates and user interactions.

Steps to Connect Flutter UI with Firebase Database

Step 1:

1.1) Go to Firebase Console and Create a Firebase Project



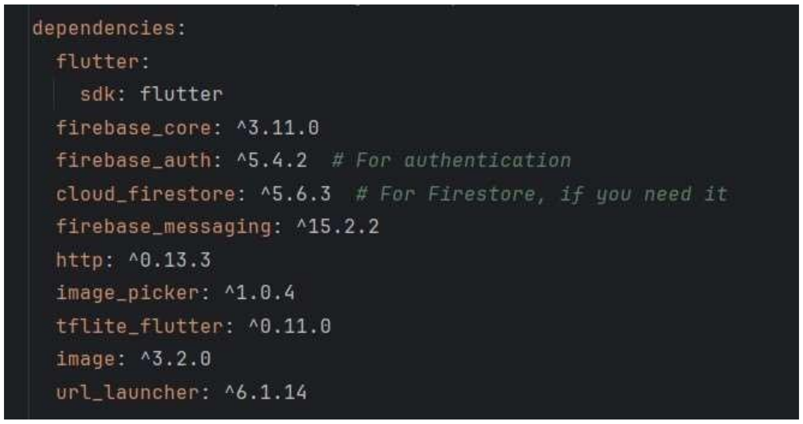
1.2) Click on Create a Project and give it a suitable name.

Step 2:- Add Firebase to Your Flutter App

2.1) Click on Android/iOS/Web based on your Flutter application

Step 3: - Add Firebase Authentication to Your App

3.1) Add Firebase Authentication Dependencies



```
dependencies:
  flutter:
    sdk: flutter
  firebase_core: ^3.11.0
  firebase_auth: ^5.4.2  # For authentication
  cloud_firestore: ^5.6.3  # For Firestore, if you need it
  firebase_messaging: ^15.2.2
  http: ^0.13.3
  image_picker: ^1.0.4
  tflite_flutter: ^0.11.0
  image: ^3.2.0
  url_launcher: ^6.1.14
```

3.2) Enable Authentication in Firebase Console Go to
Firebase Console → Authentication.
Click on Sign-in method and enable Email/Password (or any other method like Google). Click Save

3.3) Implement Authentication in Flutter Modify
main.dart

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

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(MyApp());
}
```

Step 4: -Configure Firebase Realtime Database

4.1) Go to Firebase Console → Realtime Database.

4.2) Click Create Database → Choose location → Set rules (for development, set read/write to true).

4.3) Click Publish.

Code:-

Sign_in_page.dart

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

import 'sign_up_page.dart'; import
'package:movflix/screens/homescreen.dart';
import
'package:movflix/widgets/bottom_bar_nav.dart';

class SignInPage extends StatefulWidget {
  @override
  _SignInPageState createState() =>
_SignInPageState();
}

class _SignInPageState
extends State<SignInPage> {
final FirebaseAuth _auth =
FirebaseAuth.instance;
  final FirebaseFirestore _firestore =
FirebaseFirestore.instance;
  final TextEditingController _emailController =
TextEditingController();   final
TextEditingController
_passwordController = TextEditingController();
String _errorMessage = "";   bool _isLoading =
false;

  Future<void>     _signIn()     async     {
setState(() {
      _isLoading = true;
      _errorMessage = "";
    });

    try {
      // Firebase Authentication
        email: _emailController.text.trim(),
        password: _passwordController.text.trim(),
      );

      User? user = userCredential.user;
if (user != null) {
        // Ensure user document exists
before updating         var userDoc = await
_firestore.collection("users").doc(user.uid).get()
;
        if (userDoc.exists) {
await
_firestore.collection("users").doc(user.uid).upda
te({
            "lastLogin":
FieldValue.serverTimestamp(),
          });
        }

        Navigator.pushAndRemoveUntil(
context,
          MaterialPageRoute(builder: (context) =>
HomeScreen()),
            (route) => false,
        );
        Navigator.of(context).pushReplacement(
MaterialPageRoute(builder: (context) => const
BottomNavBar()),
        );
      }
    }  on  FirebaseAuthException  catch  (e)  {
setState(() {
        _errorMessage = e.message ?? "Error
signing in";
        _isLoading = false;
      });
    }       finally       {
setState(() {
        _isLoading = false;
      });
    }
```

```
}


  @override
  Widget build(BuildContext context) {
return Scaffold(      body: Container(
padding: EdgeInsets.symmetric(horizontal:
15, vertical: 15),
child: Column(
children: [
          _headerWidget(),
SizedBox(
height: 10,
          ),
          _formWidget(),
        ],
      ),
    ),
  );
  }

  Widget _headerWidget() {
return Row(      children: [
InkWell(         onTap: () {
          Navigator.pop(context);
        },
        child: Icon(Icons.arrow_back),
      ),
      SizedBox(
width: 10,
      ),
      Container(         height: 40,
child: Image.asset('assets/logo.png'),
      )
    ],
  );
  }

  Widget _formWidget() {
return Expanded(
child: Column(
mainAxisAlignment:
MainAxisAlignment.center,
        children: [
Container(
padding:
EdgeInsets.symmetric(horizontal: 8),
decoration: BoxDecoration(
color: Colors.grey[800],
              borderRadius: BorderRadius.all(
                Radius.circular(5),
              ),
            ),
            child: TextFormField(
controller: _emailController,
decoration: InputDecoration(
labelStyle: TextStyle(fontSize: 14, color:
Colors.white),              border:
InputBorder.none,               labelText: "Email
or phone number",
              ),
            ),
          ),
          SizedBox(
height: 10,
          ),
          Container(
padding:
EdgeInsets.symmetric(horizontal: 8),
decoration: BoxDecoration(
color: Colors.grey[800],
              borderRadius: BorderRadius.all(
                Radius.circular(5),
              ),
            ),
            child: TextFormField(
              controller: _passwordController,
obscureText: true,            decoration:
InputDecoration(              labelStyle:
TextStyle(fontSize: 14, color: Colors.white),
border: InputBorder.none,
labelText: "Password",
              ),
            ),
          ),
          SizedBox(
height: 15,
        InkWell(            onTap: _isLoading
? null : _signIn,            child: Container(
alignment: Alignment.center,
padding:
EdgeInsets.symmetric(vertical: 15),
width: double.maxFinite,
decoration: BoxDecoration(
color: _isLoading ? Colors.grey :
Colors.transparent,                border:
Border.all(                  color:
Colors.grey[600] ??
Colors.grey, width: 2),
            ),
            child: _isLoading
              ? CircularProgressIndicator(color:
Colors.white)
              : Text("Sign In"),
          ),
        ),
        if (_errorMessage.isNotEmpty)
Padding(              padding: const
EdgeInsets.all(8.0),              child: Text(
_errorMessage,                style:
TextStyle(color: Colors.red),
            ),
          ),
        SizedBox(
height: 15,
        ),
        Text("Need Help?"),
        SizedBox(
          height: 15,
        ),
        GestureDetector(
onTap: () {
Navigator.push(
context,
            MaterialPageRoute(builder: (context)
=> SignUpPage()),
          );
},          child:
Text(
          "New to Netflix? Sign up now.",
style: TextStyle(fontWeight:
FontWeight.bold, color: Colors.blue),
          ),
        ),
        SizedBox(
height: 20,
        ),
        Text(
          "Sign-in is protected by Google reCAPTCHA
to ensure you're not a bot. Learn more.",
          style: TextStyle(
fontSize: 12,
          ),
          textAlign: TextAlign.center,
        )
      ],
    ),
  );
 }
}
```

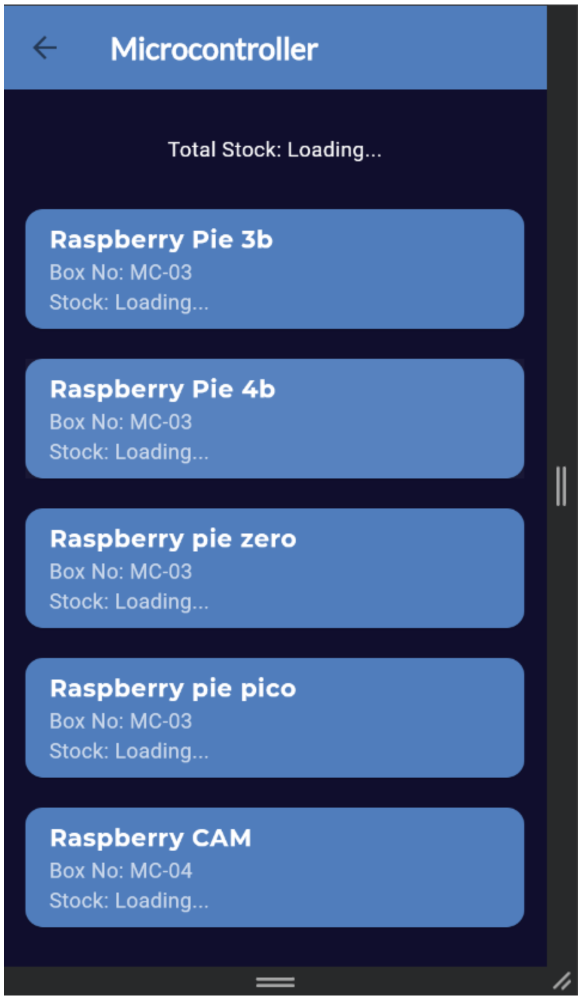
ign_up_page.dart

```
import 'package:flutter/material.dart';
import 'package:flutter/rendering.dart';
import
'package:movflix/widgets/header_widget.da
rt';
import 'sign_in_page.dart';
import
'package:movflix/screens/homescreen.dart';
import
'package:movflix/widgets/bottom_bar_nav.d
art';
import
'package:firebase_auth/firebase_auth.dart';
import
'package:cloud_firestore/cloud_firestore.dart
';

class SignUpPage extends StatefulWidget {
  @override
  _SignUpPageState createState() =>
_SignUpPageState();
}

class _SignUpPageState
extends State<SignUpPage> {
final FirebaseAuth _auth =
FirebaseAuth.instance;
  final FirebaseFirestore _firestore
= FirebaseFirestore.instance;   final
TextEditingController
_emailController = TextEditingController();
final TextEditingController
_passwordController =
TextEditingController();
bool _isCheck = false;
String _errorMessage = "";
  bool _isLoading = false;

  Future<void>     _signUp()     async     {
setState(() {
      _isLoading = true;
      _errorMessage = "";
    });
```

```
  var existingUser = await _firestore

  .collection('users')
        .where('email', isEqualTo:
_emailController.text.trim())
        .get();

    if       (existingUser.docs.isNotEmpty)       {
setState(() {
        _errorMessage = "User already exists.
Please log in.";
        _isLoading = false;
      });
return;
    }

    // Create user in Firebase Authentication
    UserCredential userCredential = await
_auth.createUserWithEmailAndPassword(
email: _emailController.text.trim(),
      password: _passwordController.text.trim(),
    );

    // Store user details in Firestore
await
_firestore.collection('users').doc(userCredential.
user!.uid).set({
      'email': _emailController.text.trim(),
      'password':_passwordController.text.trim(),
      'createdAt': FieldValue.serverTimestamp(),
    });

    // Navigate to Home Page
    Navigator.pushAndRemoveUntil(
context,
      MaterialPageRoute(builder: (context) =>
HomeScreen()),
        (route) => false,
    );

    Navigator.of(context).pushReplacement(
MaterialPageRoute(builder: (context) =>
const BottomNavBar()),
    );
    try {
      // Check if the user already exists in
Firestore
```

```
   }   on   FirebaseAuthException   catch   (e)   {
setState(() {
        _errorMessage = e.message ?? "Error
signing up";
      });
```

Output:



Microcontroller

Total Stock: Loading...

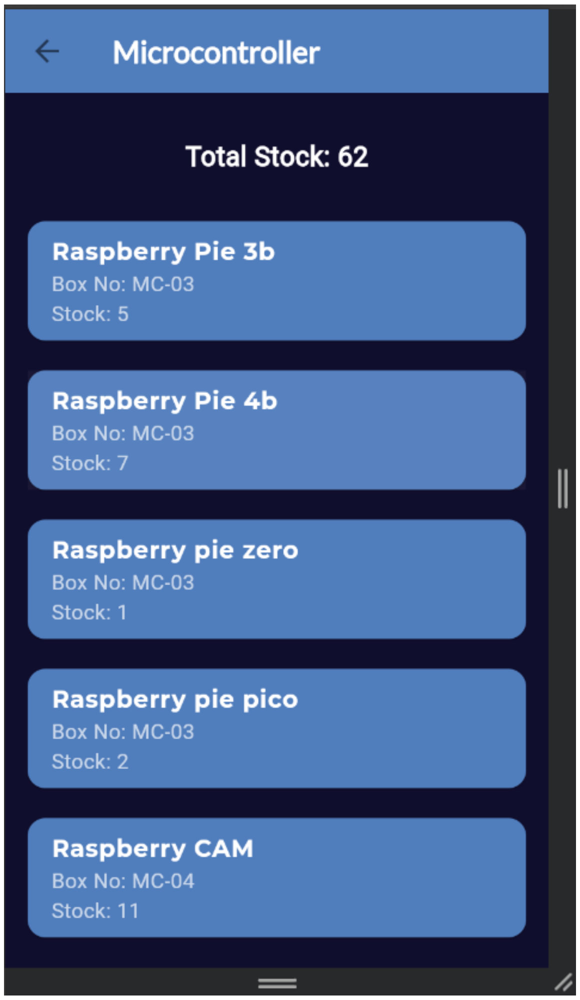
Raspberry Pie 3b
Box No: MC-03
Stock: Loading...

Raspberry Pie 4b
Box No: MC-03
Stock: Loading...

Raspberry pie zero
Box No: MC-03
Stock: Loading...

Raspberry pie pico
Box No: MC-03
Stock: Loading...

Raspberry CAM
Box No: MC-04
Stock: Loading...



Microcontroller

Total Stock: 62

Raspberry Pie 3b
Box No: MC-03
Stock: 5

Raspberry Pie 4b
Box No: MC-03
Stock: 7

Raspberry pie zero
Box No: MC-03
Stock: 1

Raspberry pie pico
Box No: MC-03
Stock: 2

Raspberry CAM
Box No: MC-04
Stock: 11

Fetching Data From the Database

‘Stock: Loading…’

Once the Data is Fetched

‘Stock: Nos’