

Sneha Ramesh Sumbe
D15A/57

Experiment - 6

Aim: To connect Flutter UI with Firebase Database

Theory: Firebase, Inc. is a set of backend cloud computing services and application development platforms provided by Google. It hosts databases, services, authentication, and integration for a variety of applications, including Android, iOS

Steps for connecting to Firebase:

- Step 1: Install the required command line tools.
- Step 2: Configure your apps to use Firebase.
- Step 3: Initialize Firebase in your app.
- Step 4: Add Firebase plugins.

Code:

LoginPage.dart

```
import 'package:flutter/material.dart';  
import 'package:firebase_auth/firebase_auth.dart'; // Import Firebase Auth  
import 'SignupPage.dart'; // Import the SignupPage.dart file  
import 'account_page.dart';
```

```
class LoginPage extends StatelessWidget {  
  final FirebaseAuth _auth = FirebaseAuth.instance;
```

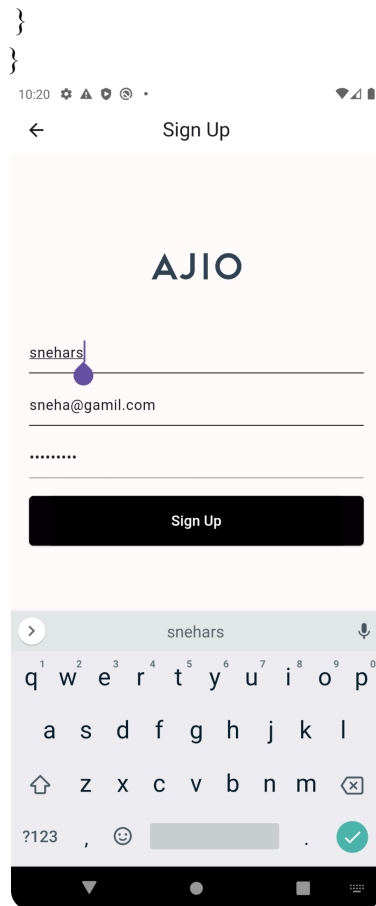
```
  // Function to handle login  
  Future<void> _loginWithEmailAndPassword(BuildContext context, String email, String password)  
  async {  
    try {  
      await _auth.signInWithEmailAndPassword(email: email, password: password);  
      // If login is successful, navigate to AccountPage  
      Navigator.pushReplacement(  
        context,  
        MaterialPageRoute(builder: (context) => AccountPage()),  
      );  
    } catch (e) {  
      print('Login Error: $e');  
      // Handle login error, show message to the user  
    }  
  }  
}
```

```
@override  
Widget build(BuildContext context) {
```

[illegible]

[illegible]

```
        fontSize: 16.0,  
      ),  
    ),  
  ),  
),  
SizedBox(height: 20.0),  
  
// New user link (Ajio-style button)  
ElevatedButton(  
  onPressed: () {  
    // Navigate to the SignupPage when the button is clicked  
    Navigator.push(  
      context,  
      MaterialPageRoute(  
        builder: (context) => SignupPage(),  
      ),  
    );  
  },  
  style: ButtonStyle(  
    backgroundColor: MaterialStateProperty.all<Color>(Colors.white,  
  ),  
  shape: MaterialStateProperty.all<RoundedRectangleBorder>(RoundedRectangleBorder(  
    borderRadius: BorderRadius.circular(5.0),  
    side: BorderSide(color: Colors.black),  
  ),  
  ),  
  child: Text(  
    'New to Ajio? Sign Up',  
    style: TextStyle(  
      color: Colors.black,  
      fontSize: 14.0,  
    ),  
  ),  
),  
],  
),  
),  
),  
);
```



```
117.100 ms
I/AssistStructure( 2048): Flattened final assist data: 508 bytes, containing 1 windows, 3 views
I/FirebaseAuth( 2048): Logging in as sneha@gmail.com with empty reCAPTCHA token
W/System ( 2048): Ignoring header X-Firebase-Locale because its value was null.
```

ProductDetailScreen.dart

```
import 'package:flutter/material.dart';
import 'category.dart';
import 'account_page.dart';
import 'main.dart';
import 'explore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'LoginPage.dart';
```

```
User? getCurrentUser() {
  return FirebaseAuth.instance.currentUser;
}
```

```
class ProductDetailScreen extends StatelessWidget {
  final String productName;
  final String productImage;
  final double productPrice;
  final double productRating;

  ProductDetailScreen({
    required this.productName,
    required this.productImage,
    required this.productPrice,
    required this.productRating,
  });

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Product Details'),
      ),
      body: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Image.network(
            productImage,
            width: double.infinity,
            fit: BoxFit.cover,
          ),
          Padding(
            padding: EdgeInsets.all(16),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  productName,
                  style: TextStyle(
                    fontSize: 24,
                    fontWeight: FontWeight.bold,
                  ),
                ),
                SizedBox(height: 8),
              ],
            ),
          ),
        ],
      ),
    );
  }
}
```

```
Text(
  '\$${productPrice}',
  style: TextStyle(
    fontSize: 18,
    color: Colors.grey[800],
  ),
),
SizedBox(height: 8),
Row(
  children: [
    Icon(
      Icons.star,
      color: Colors.amber,
      size: 20,
    ),
    SizedBox(width: 4),
    Text(
      '$productRating',
      style: TextStyle(
        fontSize: 18,
      ),
    ),
  ],
),
SizedBox(height: 16),
Row(
  children: [
    ElevatedButton(
      onPressed: () {
        handleBuyButton(context);
      },
      child: Text('Buy'),
    ),
    SizedBox(width: 16),
    ElevatedButton(
      onPressed: () {
        handleAddToCartButton(context);
      },
      child: Text('Add to Cart'),
    ),
  ],
),
```

```

        ],
      ),
    ],
  ),
),
],
),
bottomNavigationBar: BottomNavigationBar(
  items: [
    BottomNavigationBarItem(
      icon: Icon(Icons.store),
      label: 'Switch Stores',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.home),
      label: 'Explore',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.trending_up),
      label: 'TRENDin',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.category),
      label: 'Categories',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.account_circle),
      label: 'Account',
    ),
  ],
  selectedItemColor: Colors.black,
  unselectedItemColor: Colors.black,
  showUnselectedLabels: true,
  type: BottomNavigationBarType.fixed,
  onTap: (int index) {
    // Handle bottom navigation item clicks
    switch (index) {
      case 0:
        // Handle 'Switch Stores' action
        Navigator.push(

```



```

        context,
        MaterialPageRoute(builder: (context) => MyHomePage()),
    );
    break;
case 1:
// Handle 'Explore' action
    Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => Explore()),
    );
    break;
case 2:
// Handle 'TRENDin' action
    break;
case 3:
// Navigate to the category section page
    Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => CategorySection()),
    );
    break;
case 4:
// Handle 'Account' action
    Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => AccountPage()),
    );
    break;
default:
}
},
),
);
}

```

```

void handleBuyButton(BuildContext context) {
    final user = getCurrentUser();
    if (user != null) {
        // User is logged in, store product details in Firestore
        storeProductDetails('buy');
    }
}

```

```

    } else {
        // User is not logged in, navigate to login page
        Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => LoginPage()),
        );
    }
}

```

```

void handleAddToCartButton(BuildContext context) {
    final user = getCurrentUser();
    if (user != null) {
        // User is logged in, store product details in Firestore
        storeProductDetails('cart');
    } else {
        // User is not logged in, navigate to login page
        Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => LoginPage()),
        );
    }
}

```

```

void storeProductDetails(String collection) {
    final user = getCurrentUser();
    if (user != null) {
        // Add product details along with user email to Firestore
        FirebaseFirestore.instance.collection(collection).add({
            'productName': productName,
            'productImage': productImage,
            'productPrice': productPrice,
            'productRating': productRating,
            'userEmail': user.email,
        }).then((value) {
            // Successfully stored product details
            print('Product details stored successfully');
        }).catchError((error) {
            // Failed to store product details
            print('Failed to store product details: $error');
        });
    }
}

```

```
}  
}  
}
```

```
I/flutter ( 2048): Product details stored successfully  
W/om.example.dem( 2048): Accessing hidden method Lsun/misc/Unsafe;->putObject(Ljava/lang/Object  
W/om.example.dem( 2048): Accessing hidden method Lsun/misc/Unsafe;->getInt(Ljava/lang/Object
```