

## MPL Experiment 6

**Name:** Eesha Chavan

**Class:** D15A

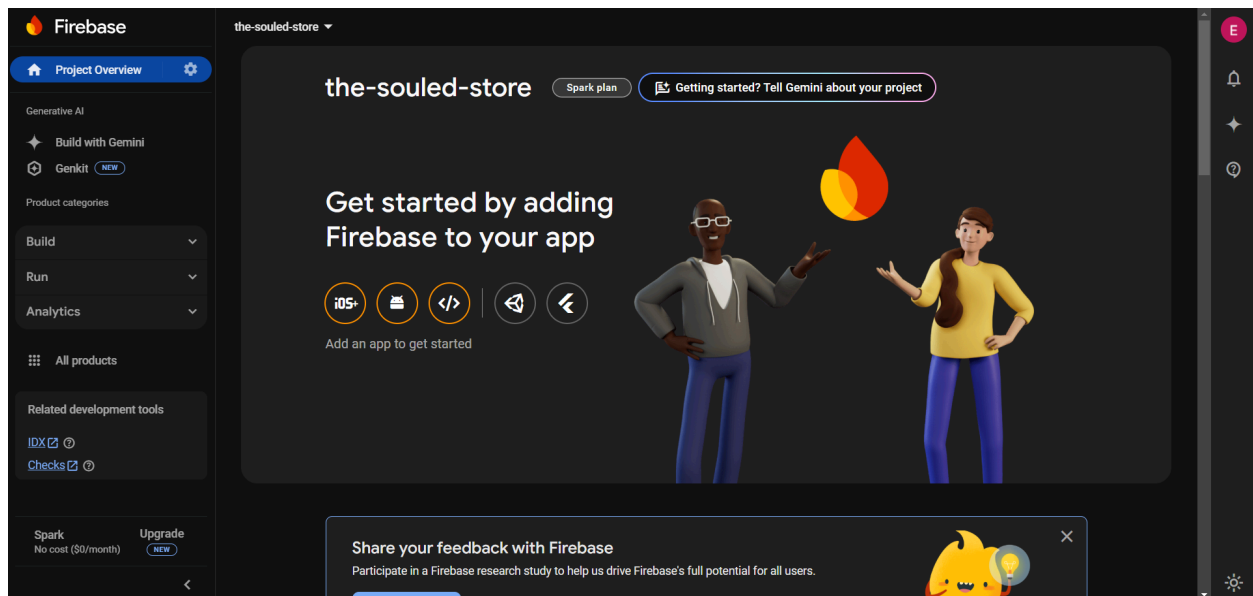
**Roll no:** 8

**Aim:** How To Set Up Firebase with Flutter for iOS and Android Apps

### **Steps to Set Up Firebase with Flutter:**

#### **Step 1:**

Go to the Firebase Console (<https://console.firebase.google.com/>). Click on "Add Project" and follow the steps to create your Firebase project. Once the project is created, select the Flutter option for connecting your app with Firebase.



#### **Step 2:**

Open **Windows PowerShell** (or any terminal you prefer).

Run the following commands to install Firebase CLI and verify the installation:

```
npm install -g firebase-tools
```

```
firebase --version
```

```
firebase login
```

This will install the **Firebase CLI**, check the version, and log you into your Firebase account.

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Eesha Chavan> npm install -g firebase-tools

added 631 packages in 1m

71 packages are looking for funding
  run 'npm fund' for details
npm notice
npm notice New major version of npm available! 10.2.4 -> 11.1.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.1.0
npm notice Run npm install -g npm@11.1.0 to update!
npm notice
PS C:\Users\Eesha Chavan> firebase --version
13.32.0
PS C:\Users\Eesha Chavan> firebase login
i Firebase optionally collects CLI and Emulator Suite usage and error reporting information to help improve our products. Data is collected in accordance with Google's privacy policy (https://policies.google.com/privacy) and is not used to identify you.

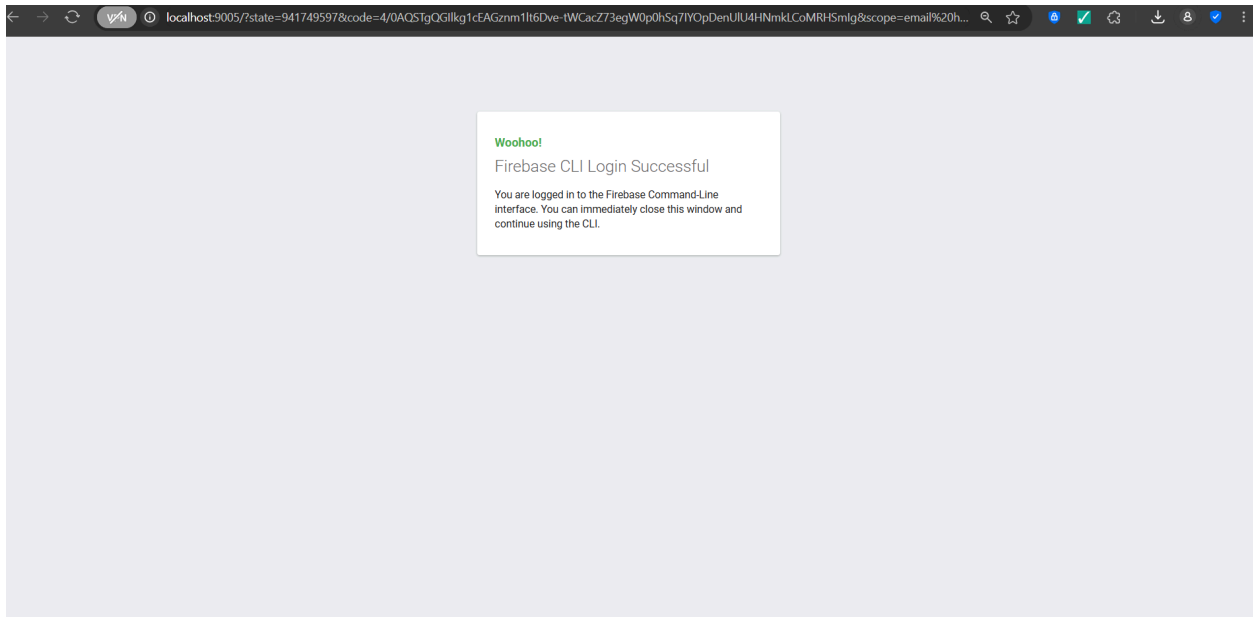
? Allow Firebase to collect CLI and Emulator Suite usage and error reporting information? Yes
i To change your data collection preference at any time, run 'firebase logout' and log in again.

Visit this URL on this device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=563584335869-fgrhgm47bqnekij5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&response_type=code&state=9417495976&redirect_uri=http%3A%2F%2Flocalhost%3A9005

Waiting for authentication...

+ Success! Logged in as eeshachavan17@gmail.com
PS C:\Users\Eesha Chavan>

```



### Step 3:

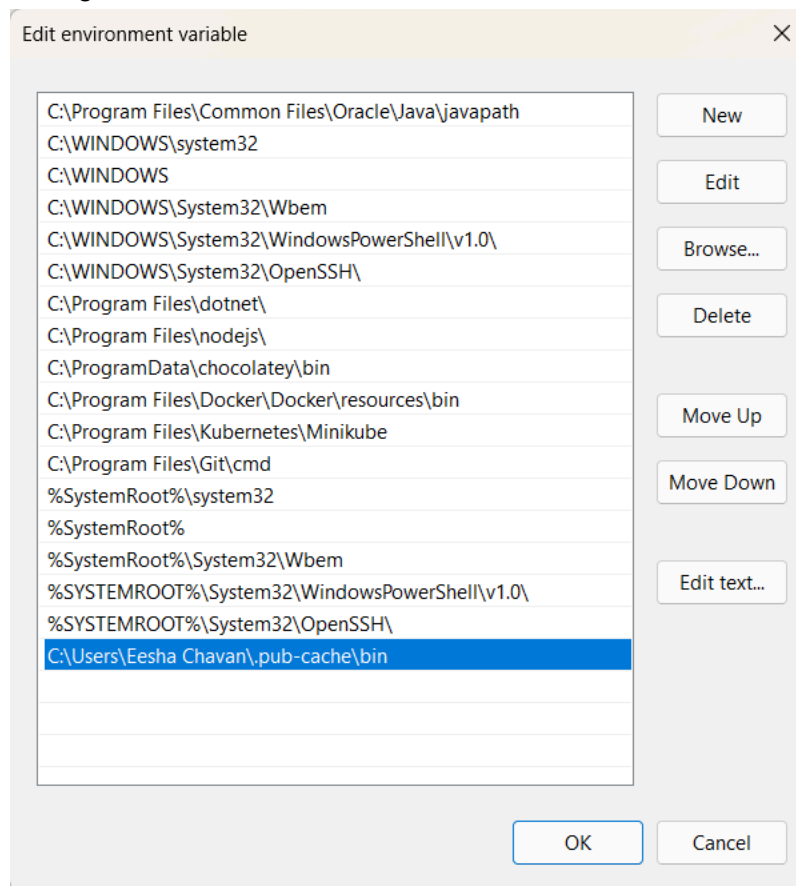
Open your Flutter app in **Android Studio**.

In the terminal of Android Studio, run the following command to activate flutterfire\_cli:

```
dart pub global activate flutterfire_cli
```

Add flutterfire to your Environment Variables. You may need to restart Android Studio after

adding it.



#### Step 4:

Run the following command to configure Firebase with your project:

```
flutterfire configure --project=dmart-c9f2c
```

Replace `dmart-c9f2c` with your Firebase project ID.

```
PROBLEMS 43  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
PS C:\Users\Eesha Chavan\Documents\TheSouledStore> dart pub global activate flutterfire_cli
Package flutterfire_cli is currently active at version 1.1.0.
Downloading packages... . (3.2s)
The package flutterfire_cli is already activated at newest available version.
To recompile executables, first run `dart pub global deactivate flutterfire_cli`.
Installed executable flutterfire.
Activated flutterfire_cli 1.1.0.
```

```
PROBLEMS 43 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS powershell
PS C:\Users\Eesha Chavan\Documents\TheSouledStore> cd the_souled_store
PS C:\Users\Eesha Chavan\Documents\TheSouledStore\the_souled_store> flutterfire configure --project=the-souled-store-d1cf4
>>
i Found 3 Firebase projects. Selecting project the-souled-store-d1cf4.
✓ Which platforms should your configuration support (use arrow keys & space to select)? · android, ios, macos, web, windows
i Firebase android app com.example.the_souled_store is not registered on Firebase project the-souled-store-d1cf4.
i Registered a new Firebase android app on Firebase project the-souled-store-d1cf4.
i Firebase ios app com.example.theSouledStore is not registered on Firebase project the-souled-store-d1cf4.
i Registered a new Firebase ios app on Firebase project the-souled-store-d1cf4.
i Firebase macos app com.example.theSouledStore registered.
i Firebase web app the_souled_store (web) is not registered on Firebase project the-souled-store-d1cf4.
i Registered a new Firebase web app on Firebase project the-souled-store-d1cf4.
i Firebase windows app the_souled_store (windows) is not registered on Firebase project the-souled-store-d1cf4.
i Registered a new Firebase windows app on Firebase project the-souled-store-d1cf4.

Firebase configuration file lib\firebase_options.dart generated successfully with the following Firebase apps:

Platform  Firebase App Id
web       1:979874017282:web:722c0abfa259bdd013f71b
android   1:979874017282:android:940798bf0f78b9de13f71b
ios       1:979874017282:ios:80441bd729f2ce3d13f71b
macos     1:979874017282:ios:80441bd729f2ce3d13f71b
windows   1:979874017282:web:f633643be0222d1413f71b

Learn more about using this file and next steps from the documentation:
> https://firebase.google.com/docs/flutter/setup
PS C:\Users\Eesha Chavan\Documents\TheSouledStore\the_souled_store>
```

## Step 5:

Run this command to add Firebase Core dependency to your app:

```
flutter pub add firebase_core
```

```

PS C:\Users\Eesha Chavan\Documents\TheSouledStore\the_souled_store> flutter pub add firebase_core
Resolving dependencies...
Downloading packages... (6.7s)
  async 2.11.0 (2.13.0 available)
  boolean_selector 2.1.1 (2.1.2 available)
  characters 1.3.0 (1.4.0 available)
  clock 1.1.1 (1.1.2 available)
  collection 1.19.0 (1.19.1 available)
  fake_async 1.3.1 (1.3.3 available)
+ firebase_core 3.12.1
+ firebase_core_platform_interface 5.4.0
+ firebase_core_web 2.21.1
  leak_tracker 10.0.7 (10.0.9 available)
  leak_tracker_flutter_testing 3.0.8 (3.0.9 available)
  matcher 0.12.16+1 (0.12.17 available)
  material_color_utilities 0.11.1 (0.12.0 available)
  meta 1.15.0 (1.16.0 available)
  path 1.9.0 (1.9.1 available)
+ plugin_platform_interface 2.1.8
  source_span 1.10.0 (1.10.1 available)
  stack_trace 1.12.0 (1.12.1 available)
  stream_channel 2.1.2 (2.1.4 available)
  string_scanner 1.3.0 (1.4.1 available)
  term_glyph 1.2.1 (1.2.2 available)
  test_api 0.7.3 (0.7.4 available)
  vm_service 14.3.0 (15.0.0 available)
Changed 4 dependencies!
19 packages have newer versions incompatible with dependency constraints.
Try `flutter pub outdated` for more information.

```

## Firebase Authentication SetUp

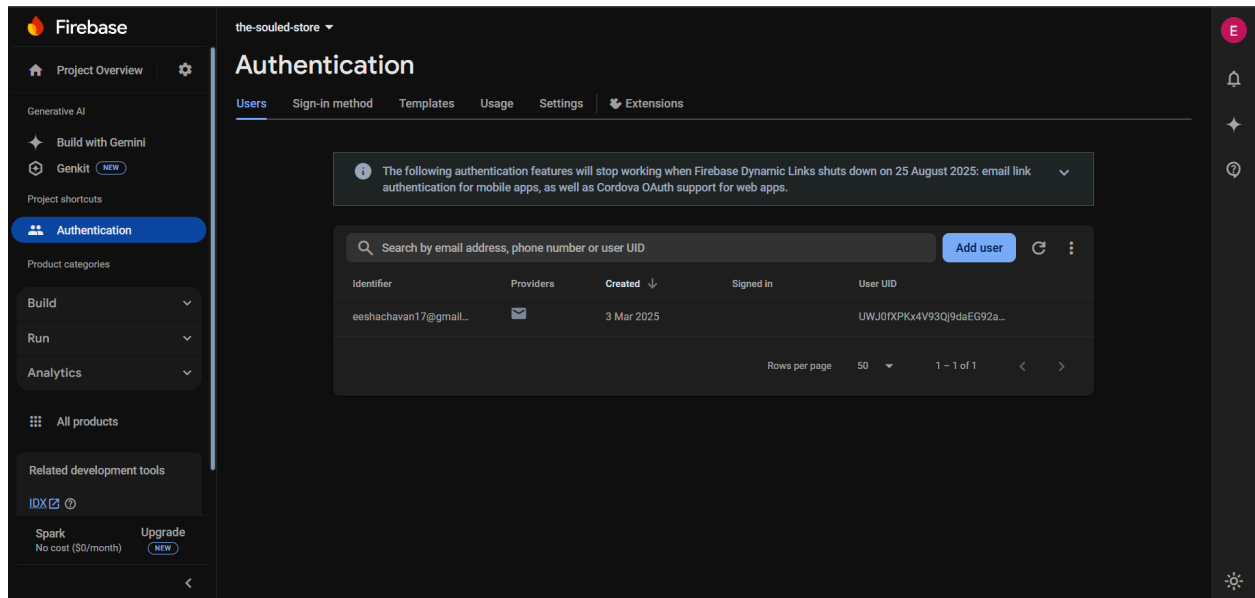
### Step 1:

Go to the Firebase Console (<https://console.firebase.google.com/>). In your Firebase project, navigate to **Authentication**. Under the **Sign-in method** tab, enable **Email/Password** sign-in. Once enabled, go to the **Users** section and click on **Add User**. Enter a **username** (email) and a

password for the new user.

The screenshot shows the Firebase Authentication console for the project 'the-souled-store'. The 'Sign-in method' tab is selected. Under 'Sign-in providers', the 'Email/Password' provider is listed with a status of 'Enabled'. An 'Add new provider' button is in the top right. Below, the 'Advanced' section features a card for 'SMS multi-factor authentication' with a description and a 'Learn more' link. At the bottom of this card is a 'Upgrade to enable' button.

This screenshot shows the 'Users' tab in the Firebase Authentication console. A modal is open to 'Add an Email/Password user'. The 'Email' field contains 'eeshachavan17@gmail.com' and the 'Password' field contains 'eeshachavanTSS\$'. 'Add user' and 'Cancel' buttons are at the bottom right of the modal. Above the modal, a warning message states: 'The following authentication features will stop working when Firebase Dynamic Links shuts down on 25 August 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.' Below the modal, a message says 'No users for this project yet'.



## Step 2:

Open your app in **Android Studio**.

In the terminal, run the following command to add the Firebase Authentication dependency:

```
flutter pub add firebase_auth
```

```

PS C:\Users\Eesha Chavan\Documents\TheSouledStore\the_souled_store> flutter pub add firebase_auth
Resolving dependencies...
Downloading packages... (6.9s)
+ _flutterfire_internals 1.3.53
  async 2.11.0 (2.13.0 available)
  boolean_selector 2.1.1 (2.1.2 available)
  characters 1.3.0 (1.4.0 available)
  clock 1.1.1 (1.1.2 available)
  collection 1.19.0 (1.19.1 available)
  fake_async 1.3.1 (1.3.3 available)
+ firebase_auth 5.5.1
+ firebase_auth_platform_interface 7.6.1
+ firebase_auth_web 5.14.1
+ http_parser 4.1.2
  leak_tracker 10.0.7 (10.0.9 available)
  leak_tracker_flutter_testing 3.0.8 (3.0.9 available)
  matcher 0.12.16+1 (0.12.17 available)
  material_color_utilities 0.11.1 (0.12.0 available)
  meta 1.15.0 (1.16.0 available)
  path 1.9.0 (1.9.1 available)
  source_span 1.10.0 (1.10.1 available)
  stack_trace 1.12.0 (1.12.1 available)
  stream_channel 2.1.2 (2.1.4 available)
  string_scanner 1.3.0 (1.4.1 available)
  term_glyph 1.2.1 (1.2.2 available)
  test_api 0.7.3 (0.7.4 available)
+ typed_data 1.4.0
  vm_service 14.3.0 (15.0.0 available)
Changed 6 dependencies!
19 packages have newer versions incompatible with dependency constraints.
Try `flutter pub outdated` for more information.

```

### Step 3:

If you encounter any issues, add the following configuration to your `android/app/build.gradle` file:

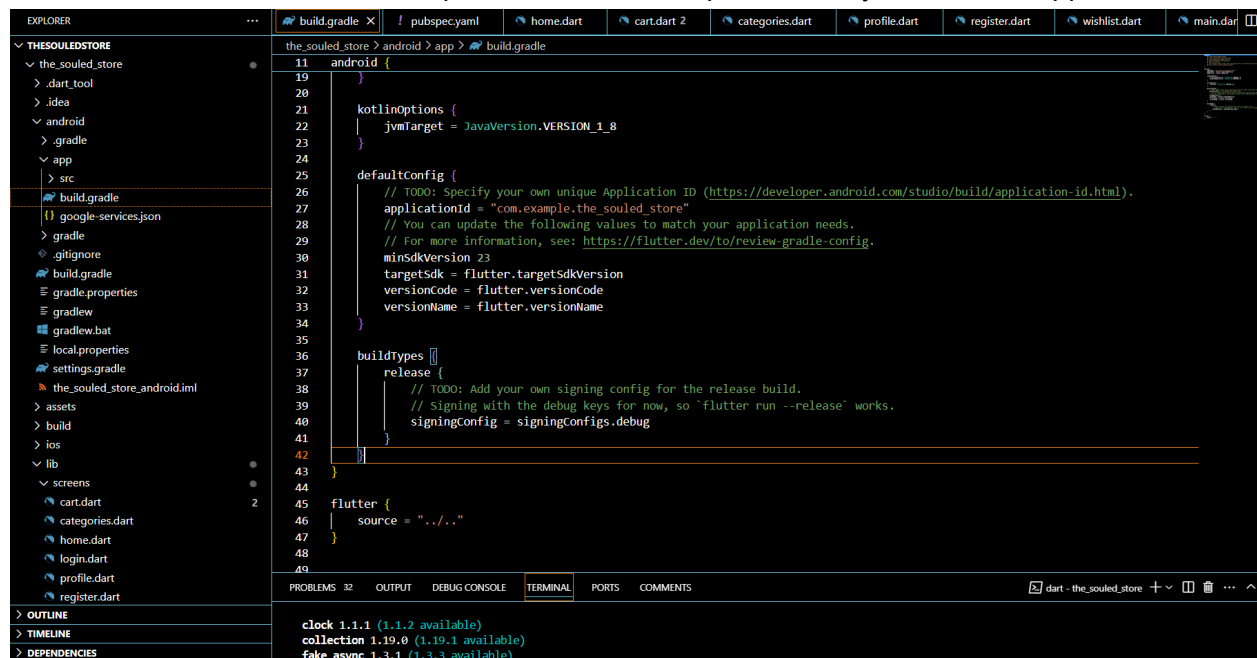
```

android {
    defaultConfig {
        minSdk = 23
        targetSdk = flutter.targetSdkVersion
        versionCode = flutter.versionCode
        versionName = flutter.versionName
    }
}

```



This ensures that the Firebase dependencies are compatible with your Android app.

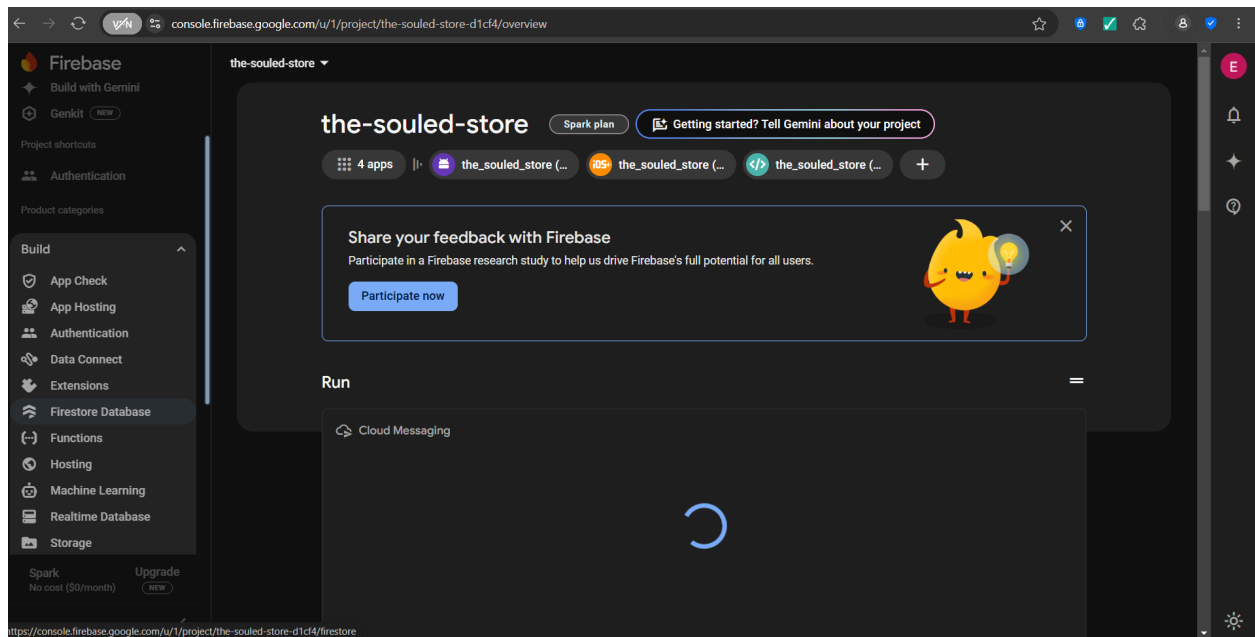


Now, firebase is successfully connected to our app.

## Firestore Database Setup

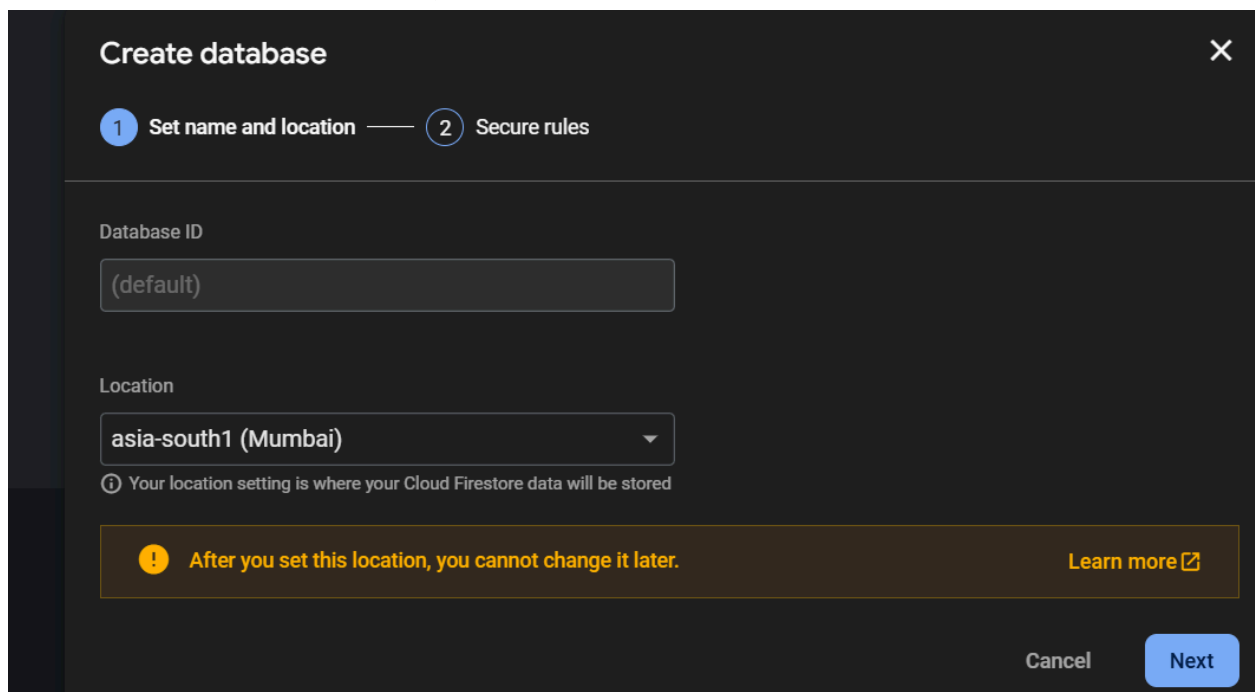
### Step 1:

Select **Firestore Database** from **Build** in the sidebar.



Then click on Create database. For location select asia-south1(Mumbai). Then choose **Start in test mode** (for development).

Click **Next** and choose a location, then **Enable**.



Create database

Set name and location

2

Secure rules

After you've defined your data structure, you will need to write rules to secure your data.

[Learn more](#)

Start in **Production mode**

Your data is private by default. Client read/write access will only be granted as specified by your security rules.

Start in **test mode**

Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```

rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
        request.time < timestamp.date(2025, 4, 2);
    }
  }
}

```

The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

Cancel

Create

## Step 2:

Start by creating a **collection**. Add a **document** within the collection. Define the **fields** as per your project requirements.

Build with Gemini

Genkit

Project shortcuts

Authentication

Firestore Database

Product categories

Build

Run

Analytics

All products

Related development tools

IDX

Checks

Spark

Upgrade

the-souled-store

Cloud Firestore

Panel view

Query builder

categories

men

(default)

categories

men

+ Start collection

+ Add document

categories

men

+ Start collection

+ Add field

accessories

bestsellers

bottomwear

image: "https://th.bing.com/th/id/OIP/M85oFXh9e3ZY0JWx\_SSQ3w=182&h=243&c=7&r=0&o=5&dpr=1.3&pid=1.7"

name: "Iron-Man Billionaire Eau De Parfum"

price: 1799

image: "https://th.bing.com/th/id/OIP/GEe8Xbz6s8rK0yJch7JSfWHw=182&h=243&c=7&r=0&o=5&dpr=1.3&pid=1.7"

name: "Black and White High Top Shoes"

price: 2499

Database location: asia-south1

### Step 3:

Run the following command to add Firestore to your Flutter app:

```
flutter pub add cloud_firestore
```

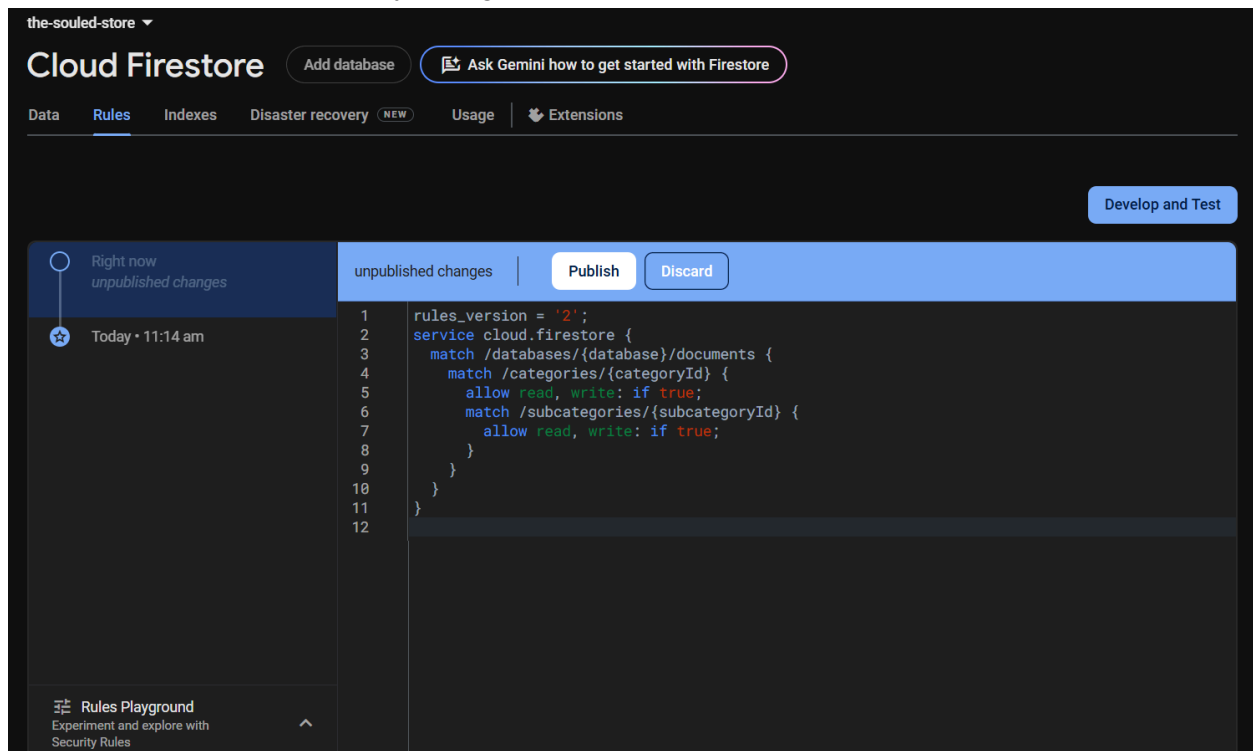
### Step 4:

Go to Firebase Console → Firestore Database → Rules

Set the following rules:

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /categories/{categoryId} {
      allow read, write: if true;
      match /subcategories/{subcategoryId} {
        allow read, write: if true;
      }
    }
  }
}
```

and then click **Publish** to apply changes.



**Code:**

login.dart:

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

class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController =
    TextEditingController();
  final FirebaseAuth _auth = FirebaseAuth.instance;

  void _login() async {
    try {
      await _auth.signInWithEmailAndPassword(
        email: emailController.text.trim(),
        password: passwordController.text.trim(),
      );
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => HomePage()),
      );
    } catch (e) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text("Login failed: ${e.toString()}")),
      );
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: Column(
```

```

children: [
  Expanded(
    flex: 45,
    child: Container(
      decoration: BoxDecoration(
        image: DecorationImage(
          image: NetworkImage(
            "https://rukminim2.flixcart.com/image/612/612/xif0q/t-shirt/c/e/o/m-195139
            -the-souled-store-original-imaghy3nqsm7ctkg.jpeg?q=70",
          ),
          fit: BoxFit.cover,
        ),
      ),
    ),
  ),
  Expanded(
    flex: 55,
    child: Container(
      padding: EdgeInsets.symmetric(horizontal: 24, vertical: 25),
      decoration: BoxDecoration(
        color: Colors.white,
        borderRadius: BorderRadius.vertical(top:
Radius.circular(20)),
      ),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          SizedBox(height: 10),
          TextField(
            controller: emailController,
            keyboardType: TextInputType.emailAddress,
            decoration: InputDecoration(
              labelText: "Email",
              border: OutlineInputBorder(),
            ),
          ),
          SizedBox(height: 10),
          TextField(
            controller: passwordController,

```

```

        obscureText: true,
        decoration: InputDecoration(
          labelText: "Password",
          border: OutlineInputBorder(),
        ),
      ),
    ),
    SizedBox(height: 20),
    ElevatedButton(
      style: ElevatedButton.styleFrom(
        backgroundColor: Color(0xFF31827C),
        minimumSize: Size(double.infinity, 50),
        shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(8)),
      ),
      onPressed: _login,
      child: Text(
        "LOGIN",
        style: TextStyle(color: Colors.white, fontSize: 16,
fontWeight: FontWeight.bold),
      ),
    ),
    ),
    SizedBox(height: 20),
    Center(
      child: GestureDetector(
        onTap: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) =>
RegisterPage()),
          );
        },
      child: RichText(
        text: TextSpan(
          style: TextStyle(fontSize: 14, color:
Colors.grey),
          children: [
            TextSpan(text: "Don't have an account? "),
            TextSpan(
              text: "REGISTER NOW",

```

```

                                style: TextStyle(color: Colors.red,
decoration: TextDecoration.underline),
                                ),
                                1,
                                ),
                                ),
                                ),
                                ),
                                1,
                                ),
                                ),
                                ),
                                1,
                                ),
                                ),
                                ),
                                1,
                                ),
                                );
    }
}

```

signup.dart:

```

import 'package:flutter/material.dart'
import 'package:firebase_auth/firebase_auth.dart'; // Import Firebase Auth
import 'login.dart'; // Importing login page for navigation

class RegisterPage extends StatefulWidget {
  @override
  _RegisterPageState createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {
  DateTime? selectedDate;
  String? selectedGender;

  // Form Key for validation
  final _formKey = GlobalKey<FormState>();

  // Controllers for text fields
  final TextEditingController _firstNameController =
TextEditingController();

```



```

    final TextEditingController _lastNameController =
    TextEditingController();

    final TextEditingController _phoneController = TextEditingController();
    final TextEditingController _emailController = TextEditingController();
    final TextEditingController _passwordController =
    TextEditingController();

    // Firebase Auth instance
    final FirebaseAuth _auth = FirebaseAuth.instance;

    // Function to handle user registration
    Future<void> _registerUser() async {
        if (_formKey.currentState!.validate()) {
            try {
                // Create user with email and password
                UserCredential userCredential = await
                _auth.createUserWithEmailAndPassword(
                    email: _emailController.text.trim(),
                    password: _passwordController.text.trim(),
                );

                // If registration is successful, print user details
                print("User Registered: ${userCredential.user!.email}");

                // Navigate to the login page after successful registration
                Navigator.pushReplacement(
                    context,
                    MaterialPageRoute(builder: (context) => LoginPage()),
                );
            } on FirebaseAuthException catch (e) {
                // Handle registration errors
                String errorMessage = "Registration failed. Please try again.";
                if (e.code == 'weak-password') {
                    errorMessage = "The password provided is too weak.";
                } else if (e.code == 'email-already-in-use') {
                    errorMessage = "The account already exists for that email.";
                }

                // Show error message to the user
                ScaffoldMessenger.of(context).showSnackBar(

```

[illegible]

```

                style: TextStyle(fontSize: 18, fontWeight:
FontWeight.bold, color: Colors.red),
            ),
            SizedBox(height: 5),
            Text(
                "Get 10% additional cashback as TSS points on every
purchase.",
                textAlign: TextAlign.center,
                style: TextStyle(fontSize: 14, color: Colors.grey),
            ),
        ],
    ),
),
),
SizedBox(height: 20),

// First Name & Last Name
Row(
    children: [
        Expanded(
            child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                    Text("First Name", style: TextStyle(fontSize: 14,
color: Colors.black)),
                    TextFormField(
                        controller: _firstNameController,
                        decoration: InputDecoration(
                            hintText: "First Name *",
                            hintStyle: TextStyle(color: Colors.grey),
                            border: OutlineInputBorder(),
                        ),
                    ),
                    validator: (value) {
                        if (value == null || value.isEmpty) {
                            return 'First Name is required';
                        }
                        return null;
                    },
                ],
            ),
        ],
    ),
),

```

```

    ),
    SizedBox(width: 10),
    Expanded(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text("Last Name", style: TextStyle(fontSize: 14,
color: Colors.black)),
          TextFormField(
            controller: _lastNameController,
            decoration: InputDecoration(
              hintText: "Last Name *",
              hintStyle: TextStyle(color: Colors.grey),
              border: OutlineInputBorder(),
            ),
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'Last Name is required';
              }
              return null;
            },
          ),
        ],
      ),
    ),
    SizedBox(height: 15),

    // Mobile Number
    TextFormField(
      controller: _phoneController,
      keyboardType: TextInputType.phone,
      decoration: InputDecoration(
        hintText: "+91 *",
        hintStyle: TextStyle(color: Colors.grey),
        border: OutlineInputBorder(),
      ),
      validator: (value) {
        if (value == null || value.isEmpty) {

```

```

        return 'Phone number is required';
    } else if (value.length != 10) {
        return 'Phone number must be 10 digits';
    }
    return null;
},
),
SizedBox(height: 15),

// Email ID
Text("Email ID", style: TextStyle(fontSize: 14, color:
Colors.black)),
TextFormField(
  controller: _emailController,
  keyboardType: TextInputType.emailAddress,
  decoration: InputDecoration(
    hintText: "Email ID *",
    hintStyle: TextStyle(color: Colors.grey),
    border: OutlineInputBorder(),
  ),
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Email is required';
    } else if
(!RegExp(r'^[\w-\.] +@([\w-]+\.)+[\w-]{2,4}$').hasMatch(value)) {
      return 'Enter a valid email address';
    }
    return null;
  },
),
SizedBox(height: 15),

// Password
Text("Password", style: TextStyle(fontSize: 14, color:
Colors.black)),
TextFormField(
  controller: _passwordController,
  obscureText: true,
  decoration: InputDecoration(
    hintText: "Password *",

```

```

        hintStyle: TextStyle(color: Colors.grey),
        border: OutlineInputBorder(),
        suffixIcon: Icon(Icons.visibility_off, color:
Colors.grey),
    ),
    validator: (value) {
        if (value == null || value.isEmpty) {
            return 'Password is required';
        }
        return null;
    },
),
 SizedBox(height: 15),

// Birthdate
Text("Birthdate", style: TextStyle(fontSize: 14, color:
Colors.black)),
 TextFormField(
    readOnly: true,
    decoration: InputDecoration(
        hintText: "DD-MM-YYYY *",
        hintStyle: TextStyle(color: Colors.grey),
        border: OutlineInputBorder(),
        suffixIcon: Icon(Icons.calendar_today, color:
Colors.grey),
    ),
    onTap: () async {
        DateTime? pickedDate = await showDatePicker(
            context: context,
            initialDate: DateTime.now(),
            firstDate: DateTime(1900),
            lastDate: DateTime.now(),
            builder: (context, child) {
                return Theme(
                    data: ThemeData.light().copyWith(
                        primaryColor: Colors.grey,
                        colorScheme: ColorScheme.light(primary:
Colors.grey),

                        buttonTheme: ButtonThemeData(textTheme:
ButtonTextTheme.primary),

```

```

        ),
        child: child!,
      ),
    ),
  ),

  if (pickedDate != null) {
    setState(() {
      selectedDate = pickedDate;
    });
  },
  validator: (value) {
    if (selectedDate == null) {
      return 'Birthdate is required';
    }
    return null;
  },
),
 SizedBox(height: 5),

// Birthday Discount Note
Text(
  "Avail 10% birthday discount as a member",
  style: TextStyle(fontSize: 12, color: Colors.grey),
),
 SizedBox(height: 20),

// Gender Selection
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    _genderOption("Male"),
    _genderOption("Female"),
    _genderOption("Other"),
  ],
),
 SizedBox(height: 20),

// Register Button

```

```

        SizedBox(
          width: double.infinity,
          child: ElevatedButton(
            style: ElevatedButton.styleFrom(
              backgroundColor: Color(0xFF31827C), // New Darker Teal
Color
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(8),
              ),
            ),
            onPressed: _registerUser, // Call the registration
function
            child: Padding(
              padding: EdgeInsets.symmetric(vertical: 14),
              child: Text("REGISTER", style: TextStyle(fontSize: 16,
color: Colors.white)),
            ),
          ),
        ),
        SizedBox(height: 10),

        // Already Registered? LOG IN
        Center(
          child: GestureDetector(
            onTap: () {
              Navigator.pushReplacement(
                context,
                MaterialPageRoute(builder: (context) =>
LoginPage()),
              );
            },
            child: RichText(
              text: TextSpan(
                text: "Already Registered? ",
                style: TextStyle(color: Colors.grey, fontSize: 14),
                children: [
                  TextSpan(
                    text: "LOG IN",
                    style: TextStyle(
                      color: Colors.red,

```





```
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'home.dart';
import 'wishlist.dart';
import 'profile.dart';

class CategoriesPage extends StatefulWidget {
  @override
  _CategoriesPageState createState() => _CategoriesPageState();
}

class _CategoriesPageState extends State<CategoriesPage> {
  int _selectedIndex = 2; // Default index for Categories
  int _selectedTab = 0; // 0: MEN, 1: WOMEN, 2: KIDS
  int _selectedCategory = -1; // Track selected category

  // Categories for Men, Women, and Kids
  final List<String> _menCategories = [
    'topwear',
    'bottomwear',
    'footwear',
    'bestsellers',
    'accessories',
  ];

  final List<String> _womenCategories = [
    'topwear',
    'bottomwear',
    'footwear',
    'bestsellers',
    'accessories',
  ];

  final List<String> _kidsCategories = [
    'just_launched',
    'bestsellers',
    'boys',
    'girls',
    'accessories',
  ];
}
```

```

};

// Display names for categories (for UI)
final Map<String, String> _displayNames = {
  'topwear': 'Topwear',
  'bottomwear': 'Bottomwear',
  'footwear': 'Footwear',
  'bestsellers': 'Bestsellers',
  'accessories': 'Accessories',
  'just_launched': 'Just Launched',
  'boys': 'Boys',
  'girls': 'Girls',
};

// Function to get collection name based on selected tab
String _getCollectionName() {
  switch (_selectedTab) {
    case 0:
      return 'men';
    case 1:
      return 'women';
    case 2:
      return 'kids';
    default:
      return 'men';
  }
}

void _onItemTapped(int index) {
  if (index == 1) {
    // Navigate to Wishlist when Wishlist icon is tapped
    Navigator.pushReplacement(
      context,
      MaterialPageRoute(builder: (context) => WishListPage()),
    );
  } else if (index == 0) {
    // Navigate to Home when Home icon is tapped
    Navigator.pushReplacement(

```

```

        context,
        MaterialPageRoute(builder: (context) => HomePage()),
    );
} else if (index == 2) {
    // Already on categories page
    setState(() {
        _selectedIndex = index;
    });
} else if (index == 4) {
    // Navigate to Profile when Profile icon is tapped
    Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => ProfilePage()),
    );
} else {
    setState(() {
        _selectedIndex = index;
    });
}
}

List<Widget> _buildProductGrid(List<dynamic>? products) {
    if (products == null || products.isEmpty) {
        return [
            Center(
                child: Text(
                    "No products available",
                    style: TextStyle(fontSize: 16),
                ),
            ),
        ];
    }

    return List.generate(products.length, (index) {
        final product = products[index];
        return Card(
            elevation: 2,
            child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [

```

```

    AspectRatio(
      aspectRatio: 1,
      child: product['image'] != null
        ? Image.network(
            product['image'],
            fit: BoxFit.cover,
            loadingBuilder: (context, child, loadingProgress) {
              if (loadingProgress == null) return child;
              return Center(
                child: CircularProgressIndicator(
                  value: loadingProgress.expectedTotalBytes !=
null
                    ? loadingProgress.cumulativeBytesLoaded /
                      loadingProgress.expectedTotalBytes!
                    : null,
                ),
              );
            },
            errorBuilder: (context, error, stackTrace) {
              return Container(
                color: Colors.grey[200],
                child: Icon(Icons.image_not_supported, size:
50),
              );
            },
          )
        : Container(
            color: Colors.grey[200],
            child: Icon(Icons.image_not_supported, size: 50),
          ),
    ),
    Padding(
      padding: const EdgeInsets.all(8.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(
            product['name'] ?? 'No Name',
            style: TextStyle(
              fontWeight: FontWeight.bold,

```

```

        fontSize: 14,
      ),
      maxLines: 2,
      overflow: TextOverflow.ellipsis,
    ),
    SizedBox(height: 4),
    Text(
      '₹${product['price']?.toString() ?? '0'}',
      style: TextStyle(
        color: Colors.black,
        fontWeight: FontWeight.w500,
        fontSize: 14,
      ),
    ),
  ),
],
),
),
],
),
),
],
),
);
});
}

@override
Widget build(BuildContext context) {
  List<String> currentCategories = _selectedTab == 0
    ? _menCategories
    : _selectedTab == 1
      ? _womenCategories
      : _kidsCategories;

  return Scaffold(
    backgroundColor: Colors.white,
    appBar: AppBar(
      backgroundColor: Colors.white,
      elevation: 0,
      leading: IconButton(
        icon: Icon(Icons.menu, color: Colors.black, size: 28),
        onPressed: () {},
      ),
    ),
  );
}

```



```

        color: Colors.black,
      ),
    ),
    if (_selectedTab == index)
      Container(
        margin: EdgeInsets.only(top: 4),
        height: 3,
        width: 50,
        color: Colors.teal,
      ),
    ],
  ),
);
}).toList(),
),
// Categories and Products
Expanded(
  child: Row(
    children: [
      // Left Side: Categories List
      Container(
        width: MediaQuery.of(context).size.width * 0.3,
        color: Colors.grey[100],
        child: ListView(
          children:
currentCategories.asMap().entries.map((entry) {
          int index = entry.key;
          String category = entry.value;
          return GestureDetector(
            onTap: () => setState(() {
              _selectedCategory = index;
            }),
            child: Container(
              color: _selectedCategory == index
                ? Colors.grey[300]
                : Colors.grey[100],
              padding:
                EdgeInsets.symmetric(vertical: 16,
horizontal: 8),

```



```

        child: Text(
          _displayNames[category] ?? category,
          style: TextStyle(
            fontSize: 16,
            fontWeight: FontWeight.w500,
            color: Colors.black,
          ),
        ),
      ),
    ),
  );
  }).toList(),
),
),
// Right Side: Products
Expanded(
  child: _selectedCategory == -1
    ? Center(
        child: Text(
          "Select a category to view products",
          style: TextStyle(fontSize: 16),
        ),
      )
    : FutureBuilder<DocumentSnapshot>(
        future: FirebaseFirestore.instance
          .collection('categories')
          .doc(_getCollectionName())
          .get(),
        builder: (context, snapshot) {
          if (snapshot.connectionState ==
              ConnectionState.waiting) {
            return Center(child:
CircularProgressIndicator());
          }

          if (snapshot.hasError) {
            return Center(
              child: Text(
                "Error loading products:
${snapshot.error}",
                style: TextStyle(color: Colors.red),

```

```

        ),
    );
}

if (!snapshot.hasData ||
!snapshot.data!.exists) {
    return Center(
        child: Text("No data available"),
    );
}

// Get the category field name
String categoryField =
    currentCategories[_selectedCategory];

// Get products from the selected category
field

List<dynamic>? products =
    (snapshot.data!.data() as Map<String,
dynamic>)[categoryField];

return Padding(
    padding: const EdgeInsets.all(8.0),
    child: GridView.count(
        crossAxisCount: 2,
        childAspectRatio: 0.7,
        crossAxisSpacing: 10,
        mainAxisSpacing: 10,
        children: _buildProductGrid(products),
    ),
);
},
),
),
1,
),
),
1,
),
bottomNavigationBar: BottomNavigationBar(

```

```

        currentIndex: _selectedIndex,
        onTap: _onItemTapped,
        selectedItemColor: Colors.teal,
        unselectedItemColor: Colors.black,
        items: [
            BottomNavigationBarItem(icon: Icon(Icons.home), label: "Home"),
            BottomNavigationBarItem(icon: Icon(Icons.favorite), label:
"Wishlist"),
            BottomNavigationBarItem(icon: Icon(Icons.category), label:
"Categories"),
            BottomNavigationBarItem(icon: Icon(Icons.card_membership),
label: "Membership"),
            BottomNavigationBarItem(icon: Icon(Icons.person), label:
"Profile"),
        ],
    ),
);
}
}

```

Output:

Login Page:







LOGIN


Don't have an account? [REGISTER NOW](#)




**Signup page:**

01:25

 • 

 100%



First Name

First Name \*

First Name is required

Last Name

Last Name \*

Last Name is required

9986548649469

Phone number must be 10 digits


Email ID

eesha

Enter a valid email address

Password


Password \*



Password is required

Birthdate

DD-MM-YYYY \*



Birthdate is required

Avail 10% birthday discount as a member

☐ Male

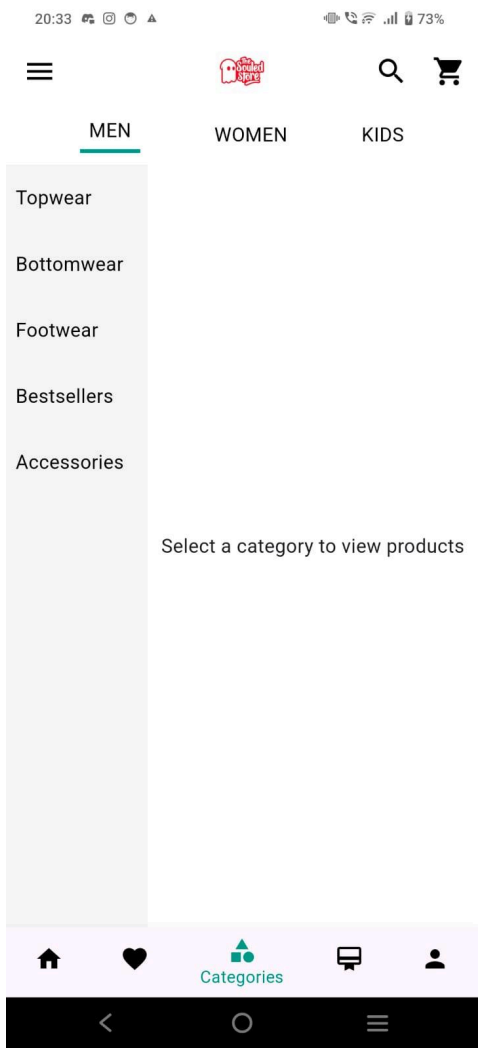
☐ Female

☐ Other

REGISTER

Already Registered? [LOG IN](#)

Category Page:



Product List Page:

