

EXPERIMENT NO: - 06

Name:-Snehal Patil

Class:- D15A

Roll:No: - 38

AIM: - To connect Flutter UI with Firebase database.

Theory: -

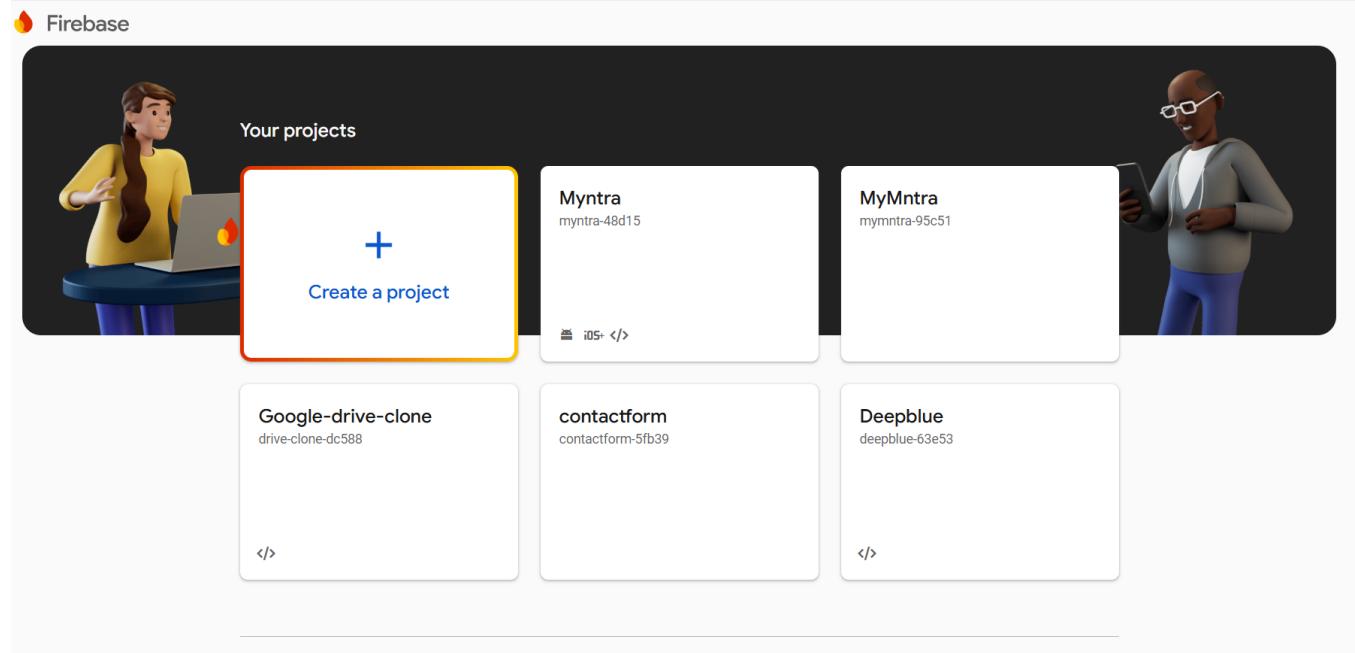
Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. Firebase, a Backend-as-a-Service (BaaS) platform, provides real-time database, authentication, and cloud storage services, making it a powerful backend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

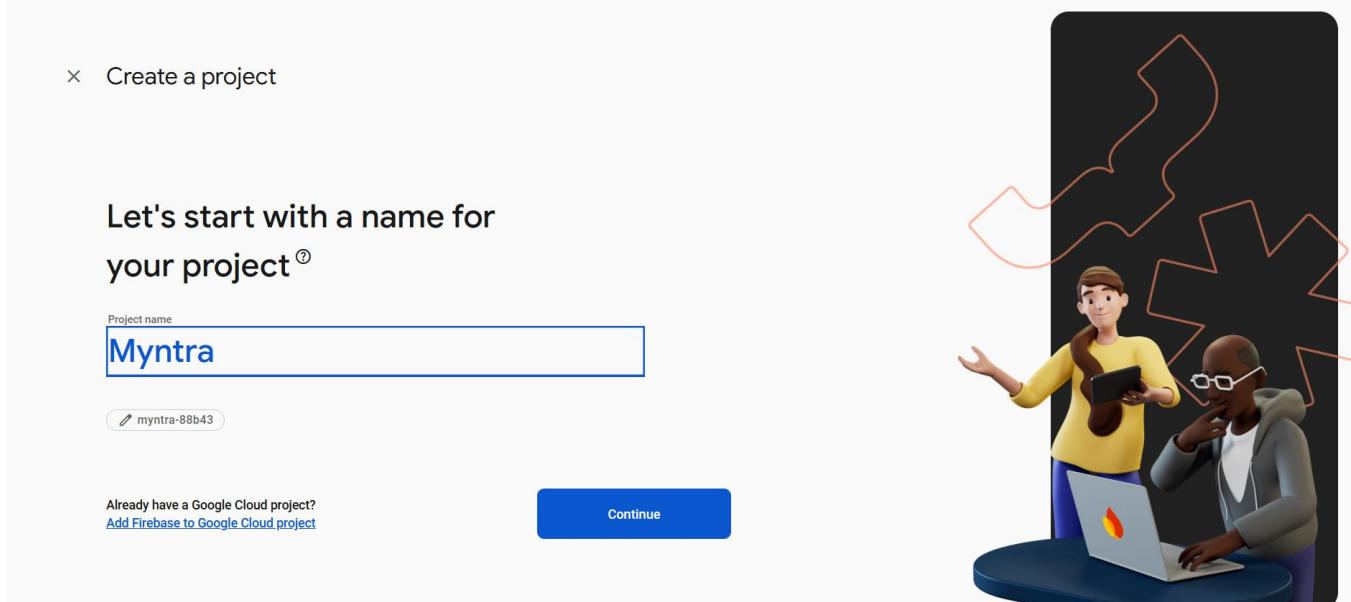
➤ Steps to Connect Flutter UI with Firebase Database

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

The screenshot shows the 'Project Overview' page for the 'Myntra' project in the Firebase console. The sidebar on the left includes sections for Generative AI, Build with Gemini, Genkit, Authentication, and Firestore Database. The main content area is titled 'Build' and shows Firestore metrics: 'Reads (current)' at 0 and 'Writes (current)' at 0. A modal window titled 'Share your feedback with Firebase' is open, encouraging participation in a Firebase research study. The overall interface is dark-themed.

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

The screenshot shows the Firebase Authentication screen for a project named "Mynta". On the left, there's a sidebar with navigation links like Project Overview, Generative AI, Build with Gemini, Genkit, Project shortcuts, Authentication (which is selected), Firestore Database, Product categories, Build, Run, Analytics, All products, Related development tools (Spark and Upgrade), and Support.

The main content area is titled "Authentication" and has tabs for Users, Sign-in method, Templates, Usage, Settings, and Extensions. A prominent message at the top states: "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." Below this, there's a search bar and a table listing users:

Identifier	Providers	Created	Signed In	User UID
xyz@gmail.com	[Email icon]	Feb 19, 2025	Feb 19, 2025	hXAwa0Pj8lcr6A6B9YQyyUxV...
abc@gmail.com	[Email icon]	Feb 11, 2025	Mar 5, 2025	h2n7j3ya0UNbSQNQlxJexMY...
snehalpatil302004@gm...	[Email icon]	Feb 10, 2025	Feb 10, 2025	ZqoTbADhkAecisV4YJ7DcPiv...

At the bottom of the table, there are pagination controls: "Rows per page: 50", "1 – 3 of 3", and navigation arrows.

- 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_up_page.dart

```
import 'package:flutter/material.dart';
import
'package:firebase_auth/firebase_auth.dart';
import 'package:fluttertoast/fluttertoast.dart';
import
'package:myntra_clone/pages/home_page.dart';
// Import your home page

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

class _SignUpPageState extends
State<SignUpPage> {
  final TextEditingController nameController =
  TextEditingController();
  final TextEditingController emailController =
  TextEditingController();
  final TextEditingController mobileController =
  TextEditingController();
  final TextEditingController
passwordController = TextEditingController();
  final FirebaseAuth _auth =
  FirebaseAuth.instance;
  bool _isLoading = false;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: SafeArea(
        child: SingleChildScrollView(
          child: Padding(
            padding: const
EdgeInsets.symmetric(horizontal: 16.0),
            child: Column(
              children: [
                // Full Banner Section with Image
                Container(
                  width: double.infinity,
                  height: 200,
                  decoration: BoxDecoration(
                    gradient: LinearGradient(
                      colors: [Colors.yellow.shade100,
Colors.pink.shade100],
                    begin: Alignment.topLeft,
                    end: Alignment.bottomRight,
                  ),
                ),
                child: Image.asset(
                  'assets/images/shopping_cart.png',
                  fit: BoxFit.cover,
                ),
              ],
            ),
            const SizedBox(height: 20),
            // Sign Up Text
            Row(
              children: [
                Text(
                  "Create ",
                  style: TextStyle(
                    fontSize: 18,
                    fontWeight: FontWeight.bold,
                    color: Colors.black,
                  ),
                ),
                Text(
                  "Account",
                  style: TextStyle(
                    fontSize: 18,
                    color: Colors.black54,
                  ),
                ),
              ],
            ),
            const SizedBox(height: 20),
            // Name Field
            TextField(

```

```
        controller: nameController,
        decoration: InputDecoration(
            hintText: "Full Name",
            hintStyle: TextStyle(color:
                Colors.grey),
            border: OutlineInputBorder(
                borderRadius:
                    BorderRadius.circular(8),
            ),
            focusedBorder:
                OutlineInputBorder(
                    borderRadius:
                        BorderRadius.circular(8),
                    borderSide: BorderSide(
                        color: Colors.pink.shade300,
                        width: 2,
                    ),
                ),
            contentPadding: const
                EdgeInsets.symmetric(horizontal: 16),
        ),
    ),
    const SizedBox(height: 12),
}

// Mobile Number Field
TextField(
    controller: mobileController,
    keyboardType: TextInputType.phone,
    decoration: InputDecoration(
        hintText: "Mobile Number",
        hintStyle: TextStyle(color:
            Colors.grey),
        border: OutlineInputBorder(
            borderRadius:
                BorderRadius.circular(8),
        ),
        focusedBorder: OutlineInputBorder(
            borderRadius:
                BorderRadius.circular(8),
            borderSide: BorderSide(
                color: Colors.pink.shade300,
                width: 2,
            ),
        ),
        contentPadding: const
            EdgeInsets.symmetric(horizontal: 16),
    ),
),
const SizedBox(height: 12),

// Email Field
TextField(
    controller: emailController,
    keyboardType:
        TextInputType.emailAddress,
    decoration: InputDecoration(
        hintText: "Email Address",
        hintStyle: TextStyle(color:
            Colors.grey),
    ),
    border: OutlineInputBorder(
        borderRadius:
            BorderRadius.circular(8),
    ),
    focusedBorder:
        OutlineInputBorder(
            borderRadius:
                BorderRadius.circular(8),
            borderSide: BorderSide(
                color: Colors.pink.shade300,
                width: 2,
            ),
        ),
    contentPadding: const
        EdgeInsets.symmetric(horizontal: 16),
),
const SizedBox(height: 12),

// Password Field
TextField(
    controller: passwordController,
    obscureText: true,
    decoration: InputDecoration(
        hintText: "Password",
        hintStyle: TextStyle(color:
            Colors.grey),
    ),
    border: OutlineInputBorder(
        borderRadius:
            BorderRadius.circular(8),
    ),
    focusedBorder: OutlineInputBorder(
        borderRadius:
            BorderRadius.circular(8),
            borderSide: BorderSide(
                color: Colors.pink.shade300,
                width: 2,
            ),
        ),
    contentPadding: const
        EdgeInsets.symmetric(horizontal: 16),
)
```



```

        if (!RegExp(r"^[0-
9]{10}$").hasMatch(mobile)) {
            Fluttertoast.showToast(msg: "Enter a valid
10-digit mobile number");
            setState(() {
                _isLoading = false;
            });
            return;
        }

        try {
            await
            _auth.createUserWithEmailAndPassword(email
            : email, password: password);
            Fluttertoast.showToast(msg: "Sign Up
Successful!");
            Navigator.pushReplacement(
                context,
                MaterialPageRoute(builder: (context) =>
HomePage()),
            );
        } catch (e) {
            Fluttertoast.showToast(msg: "Error:
${e.toString()}");
        }

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

```

Sign In Page:

```

import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import
'package:cloud_firestore/cloud_firestore.dart'; // 🔘
Firestore Import
import 'package:fluttertoast/fluttertoast.dart';
import
'package:myntra_clone/pages/home_page.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;
    final FirebaseFirestore _firestore =
    FirebaseFirestore.instance; // 🔘 Firestore Instance
    bool _isLoading = false;

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            backgroundColor: Colors.white,
            body: SafeArea(
                child: SingleChildScrollView(
                    child: Padding(
                        padding: const
                    EdgeInsets.symmetric(horizontal: 16.0),
                    child: Column(
                        children: [
                            // Full Banner Section with Image
                            Container(
                                width: double.infinity,
                                height: 200,
                                decoration: BoxDecoration(
                                    gradient: LinearGradient(
                                        colors: [Colors.yellow.shade100,
Colors.pink.shade100],
                                begin: Alignment.topLeft,
                                end: Alignment.bottomRight,
                            ),

```

```

),
child: Image.asset(
  'assets/images/shopping_cart.png',
  fit: BoxFit.cover,
),
),
),
const SizedBox(height: 20),

// Login Text
Row(
  children: [
    Text(
      "Login",
      style: TextStyle(
        fontSize: 18,
        fontWeight: FontWeight.bold,
        color: Colors.black,
      ),
    ),
    Text(
      "to your account",
      style: TextStyle(
        fontSize: 18,
        color: Colors.black54,
      ),
    ),
  ],
),
const SizedBox(height: 20),

// Email Input Field
TextField(
  controller: emailController,
  keyboardType:
  TextInputType.emailAddress,
  decoration: InputDecoration(
    hintText: "Email Address",
    hintStyle: TextStyle(color:
    Colors.grey),
  border: OutlineInputBorder(
    borderRadius:
    BorderRadius.circular(8),
  ),
  focusedBorder:
  OutlineInputBorder(
    borderRadius:
    BorderRadius.circular(8),
    borderSide: BorderSide(
      color: Colors.pink.shade300,
      width: 2,
    ),
  ),
  contentPadding: const
  EdgeInsets.symmetric(horizontal: 16),
),
),
const SizedBox(height: 12),

// Password Input Field
TextField(
  controller: passwordController,
  obscureText: true,
  decoration: InputDecoration(
    hintText: "Password",
    hintStyle: TextStyle(color:
    Colors.grey),
  border: OutlineInputBorder(
    borderRadius:
    BorderRadius.circular(8),
  ),
  focusedBorder: OutlineInputBorder(
    borderRadius:
    BorderRadius.circular(8),
    borderSide: BorderSide(
      color: Colors.pink.shade300,
      width: 2,
    ),
  ),
  contentPadding: const
  EdgeInsets.symmetric(horizontal: 16),
),
),
const SizedBox(height: 20),

// Login Button
SizedBox(
  width: double.infinity,
  child: ElevatedButton(
    onPressed: _isLoading ? null :
    _loginUser,
    style: ElevatedButton.styleFrom(
      backgroundColor:
      Colors.pink.shade300,
      padding: const
      EdgeInsets.symmetric(vertical: 16),
      shape: RoundedRectangleBorder(
        borderRadius:
        BorderRadius.circular(8),
      ),
    ),
  ),
),

```

```

),
),
child: _isLoading
?
CircularProgressIndicator(color: Colors.white)
: Text(
  "LOGIN",
  style: TextStyle(
    fontSize: 16,
    color: Colors.white,
    fontWeight:
FontWeight.bold,
),
),
),
),
),
),
),
const SizedBox(height: 20),
// Don't have an account? Sign up
Row(
  mainAxisAlignment:
MainAxisAlignment.center,
  children: [
    Text(
      "Don't have an account? ",
      style: TextStyle(color:
Colors.black54),
),
    GestureDetector(
      onTap: () {
        Navigator.pushNamed(context,
'/signup');
),
    child: Text(
      "Sign up",
      style: TextStyle(
        color: Colors.pink.shade300,
        fontWeight: FontWeight.bold,
),
),
),
],
),
],
),
),
),
),
);
}
// Function to handle user login
void _loginUser() async {
  setState(() {
    _isLoading = true;
  });

  String email = emailController.text.trim();
  String password =
passwordController.text.trim();

  if (email.isEmpty || password.isEmpty) {
    Fluttertoast.showToast(msg: "Please enter email and password");
    setState(() {
      _isLoading = false;
    });
    return;
  }

  try {
    UserCredential userCredential = await
_auth.signInWithEmailAndPassword(email: email,
password: password);

    // 🔔 Ensure Firestore User Document Exists
    await
_checkAndCreateUser(userCredential.user);

    Fluttertoast.showToast(msg: "Login Successful!");
    Navigator.pushReplacement(
context,
MaterialPageRoute(builder: (context) =>
HomePage()),
);
  } catch (e) {
    Fluttertoast.showToast(msg: "Error:
${e.toString()}");
  }

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

  // 🔔 Function to check and create Firestore user document
  Future<void> _checkAndCreateUser(User? user)
async {
  if (user == null) return;
}

```

```
DocumentReference userRef =  
_firestore.collection('users').doc(user.uid);  
DocumentSnapshot userDoc = await  
userRef.get();  
  
if (!userDoc.exists) {  
    await userRef.set({  
        'name': user.displayName ?? "Unknown",  
        'email': user.email ?? "No Email",  
        'rewardPoints': 0, // Default reward points  
for new users  
        'createdAt': FieldValue.serverTimestamp(),  
    });  
    print("🔥 User document created in  
Firestore!");  
} else {  
    print("✅ User already exists in Firestore.");  
}  
}  
}
```

Main.dart

```
import 'package:flutter/material.dart';
import
'package:firebase_core/firebase_core.dart';
import
'package:myntra_clone/pages/login_page.dart';
// import
'package:myntra_clone/pages/productDetails_pa
ge.dart';
import
'package:myntra_clone/pages/profile_page.dart';
import
'package:myntra_clone/pages/signup_page.dart';
import
'package:myntra_clone/pages/home_page.dart';
import
'package:myntra_clone/pages/categories_page.d
art';
import
'package:myntra_clone/pages/product_page.dart
';
// import;
'package:myntra_clone/pages/productDetails_pa
ge.dart';

void main() async {
    WidgetsFlutterBinding.ensureInitialized(); //  Ensures Flutter binding is initialized
    await Firebase.initializeApp(); //  Initializes
Firebase before running the app
    runApp(const MyApp());
}

class MyApp extends StatelessWidget {
    const MyApp({super.key});

    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            title: 'Mynta Clone',
            debugShowCheckedModeBanner: false,
            initialRoute: '/login', //  Set Login Page
as the default route
            routes: {
                '/login': (context) => LoginPage(),
                '/signup': (context) => SignUpPage(),
                '/home': (context) => HomePage(),
                '/categories': (context) =>
CategoriesPage(),
                '/profile': (context) => ProfilePage(),
                '/product': (context) => ProductPage(),
                '/productDetails': (context) =>
ProductDetailsPage(product: {}),
            },
        );
    }
}
```

Checkout page:

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

class CheckoutPage extends StatefulWidget {
  final Map<String, dynamic> product;

  CheckoutPage({required this.product});

  @override
  _CheckoutPageState createState() =>
  _CheckoutPageState();
}

class _CheckoutPageState extends
State<CheckoutPage> {
  final TextEditingController nameController =
  TextEditingController();
  final TextEditingController addressController =
  TextEditingController();
  final TextEditingController phoneController =
  TextEditingController();

  double totalCost = 0.0; // Store total cost after
  applying reward points
  int rewardPoints = 0;
  int userPoints = 0;
  bool useRewards = false;
  String? _selectedPaymentMethod;
  final List<String> paymentMethods =
  ["Credit/Debit Card", "UPI / Wallet", "Cash on
  Delivery"];

  @override
  void initState() {
    super.initState();
    _fetchUserPoints();
    totalCost =
    widget.product['price']?.toDouble() ?? 0.0; //
  Ensure price is treated as double
  }

  /// Fetches the user's current reward points
  /// from Firestore
  void _fetchUserPoints() async {
    final user =
    FirebaseAuth.instance.currentUser;
```

```
      if (user != null) {
        final userDoc = await
        FirebaseFirestore.instance.collection('users').doc
        (user.uid).get();
        setState(() {
          userPoints = userDoc.exists ?
          (userDoc.data()?[rewardPoints] ?? 0) : 0;
        });
      }
    }

    /// Updates the user's reward points in Firestore
    Future<void> _updateUserPoints(int
    earnedPoints) async {
      final user =
      FirebaseAuth.instance.currentUser;
      if (user != null) {
        final userRef =
        FirebaseFirestore.instance.collection('users').doc
        (user.uid);

        await
        FirebaseFirestore.instance.runTransaction((trans
        action) async {
          final userDoc = await
          transaction.get(userRef);
          int currentPoints = userDoc.exists ?
          (userDoc.data()?[rewardPoints] ?? 0) : 0;

          int newPoints = currentPoints +
          earnedPoints;
          transaction.update(userRef,
          {'rewardPoints': newPoints});
        });
      }

      // Refresh user points after updating
      _fetchUserPoints();
    }

    /// Places an order and updates reward points
    void _placeOrder() async {
      try {
        if (nameController.text.isEmpty ||
        addressController.text.isEmpty ||
        phoneController.text.isEmpty) {
          throw Exception("Please fill in all required
          fields.");
        }
        if (_selectedPaymentMethod == null) {
          throw Exception("Please select a payment
```

```

method.");

final user =
FirebaseAuth.instance.currentUser;
if (user != null) {
  final userRef =
FirebaseFirestore.instance.collection('users').doc
(user.uid);
  final userDoc = await userRef.get();

  int currentPoints = userDoc.exists ?
(userDoc.data()?'rewardPoints' ?? 0) : 0;
  int earnedPoints = ((widget.product['price']
~/ 100) * 10); // 10 points per ₹100 spent
  int newPoints = currentPoints +
earnedPoints;

  if (useRewards) {
    newPoints -= userPoints;
    if (newPoints < 0) newPoints = 0; //
Prevent negative points
  }

  await userRef.update({'rewardPoints':
newPoints});

  ScaffoldMessenger.of(context).showSnackBar
Bar(
  SnackBar(content: Text("Order Placed!
 You earned $earnedPoints points.")),
);

Future.delayed(Duration(seconds: 2), () {
  Navigator.pushReplacementNamed(conte
xt, '/home');
});
} catch (e) {
  ScaffoldMessenger.of(context).showSnackBarB
ar(
  SnackBar(content: Text("Error:
${e.toString()}")),
);
}
}

@Override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Checkout"),
      backgroundColor: Colors.white,
      elevation: 0,
      leading: IconButton(
        icon: Icon(Icons.arrow_back, color:
Colors.black),
        onPressed: () => Navigator.pop(context),
      ),
      body: SingleChildScrollView(
        padding: EdgeInsets.all(16.0),
        child: Column(
          mainAxisAlignment:
CrossAxisAlignment.start,
          children: [
            _buildInputField("Full Name",
nameController),
            _buildInputField("Shipping Address",
addressController),
            _buildInputField("Phone Number",
phoneController, keyboardType:
TextInputType.phone),
            SizedBox(height: 20),
            // Earned Rewards Section
            Text("  Earn Rewards", style:
TextStyle(fontSize: 18, fontWeight:
FontWeight.bold)),
            SizedBox(height: 5),
            Text("You will earn **$rewardPoints
points** on this purchase.", style:
TextStyle(fontSize: 16, color: Colors.blue)),
            SizedBox(height: 10),
            // Display Total Cost
            Text(
              "Total Cost: ₹$totalCost",
              style: TextStyle(fontSize: 18,
fontWeight: FontWeight.bold, color:
Colors.black),
            ),
            SizedBox(height: 10),
            // Checkbox for redeeming points
            CheckboxListTile(
              title: Text("Redeem $userPoints points
for discount"),
              value: useRewards,
              onChanged: (bool? value) {
                setState(() {
                  useRewards = value ?? false;
                });
              },
            );
          ],
        ),
      ),
    ),
  );
}

```

```

totalCost = useRewards
    ? (widget.product['price'] -
userPoints).clamp(0,
widget.product['price']).toDouble()
    :
widget.product['price'].toDouble();
    });
},
),
),

SizedBox(height: 20),

// Select Payment Method
Text("Select Payment Method",
style: TextStyle(fontSize: 18, fontWeight:
FontWeight.bold)),
SizedBox(height: 10),
DropdownButtonFormField<String>(
value: _selectedPaymentMethod,
items: paymentMethods.map((method)
{
    return DropdownMenuItem<String>(
        value: method,
        child: Text(method),
    );
}).toList(),
onChanged: (value) {
    setState(() {
        _selectedPaymentMethod = value;
    });
},
decoration: InputDecoration(
    border: OutlineInputBorder(),
    hintText: "Choose payment method",
),
),
SizedBox(height: 20),

// Confirm Order Button
SizedBox(
    width: double.infinity,
    child: ElevatedButton(
        style: ElevatedButton.styleFrom(
            padding:
EdgeInsets.symmetric(vertical: 14),
            backgroundColor: Colors.green,
            shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(8)),
),
 onPressed: _placeOrder,
),
),
),
),
],
),
),
);
}

Widget _buildInputField(String label,
TextEditingController controller,
{TextInputType keyboardType =
TextInputType.text}) {
return Padding(
padding: const EdgeInsets.only(bottom: 12),
child: TextField(
controller: controller,
keyboardType: keyboardType,
decoration: InputDecoration(labelText:
label, border: OutlineInputBorder()),
),
);
}
}
}

```

Firebase Project Overview | Cloud Firestore

Myatra ▾ Cloud Firestore

Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing Configure App Check

Panel view Query builder ⋮

users > Q6QG2atL91Fzu

(default) users Q6QG2atL91FzuAhD9Nq

+ Start collection + Add document + Start collection

users > h2n7j3yaoUNbSQNQIxJexMYBcRD3

+ Add field

email: "john@example.com"
name: "John Doe"
rewardPoints: 50
userId: "UID123456"

More in Google Cloud ⋮

Generative AI Build Run Analytics All products

Authentication Firestore Database Product categories

Build Run Analytics All products

Related development tools Spark Upgrade NEW

Firebase Project Overview | Cloud Firestore

Myatra ▾ Cloud Firestore

Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing Configure App Check

Panel view Query builder ⋮

users > h2n7j3yaoUNbSQNQIxJexMYBcRD3

(default) users h2n7j3yaoUNbSQNQIxJexMYBcRD3

+ Start collection + Add document + Start collection

users > h2n7j3yaoUNbSQNQIxJexMYBcRD3

+ Add field

rewardPoints: 250

More in Google Cloud ⋮

Generative AI Build Run Analytics All products

Authentication Firestore Database Product categories

Build Run Analytics All products

Related development tools Spark Upgrade NEW

Checkout page:

Android Emulator - flutter_emulator:5554

LTE 11:30

← Checkout

Full Name —
abc

Shipping Address —
abc

Phone Number —
1234567890

💎 Earn Rewards
You will earn **0 points** on this purchase.

Total Cost: ₹4750.0

Redeem 250 points for discount

Select Payment Method

Choose payment method

Confirm Order

