

MPL Prerequisites Document

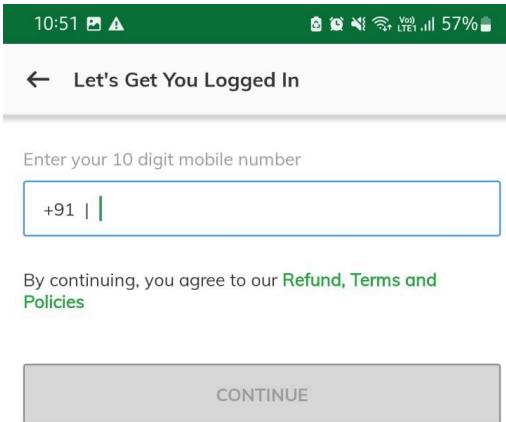
Name: Anuprita Mhapankar

Class: D15A

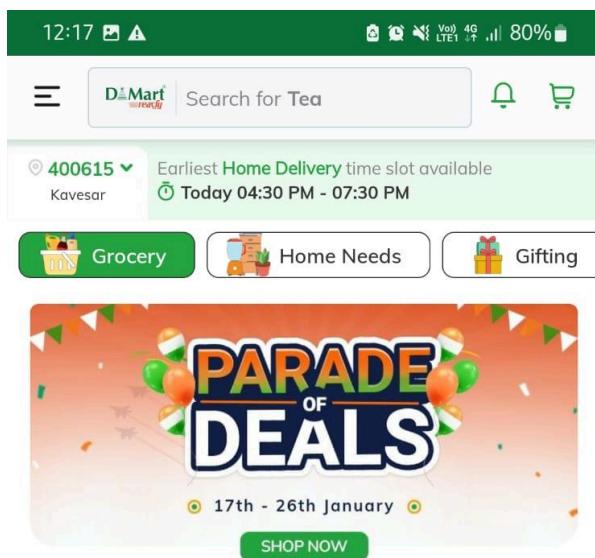
Roll no:28

Aim: Selecting features for application development, the features should comprise of:

- Common widgets
- Should include icons, images, charts etc.
- Should have an interactive Form
- Should apply navigation, routing and gestures
- Should connect with FireBase database

| No. | Screen Shot | Features |
|-----|--|---|
| 1 |  | <p>Landing Page :</p> <p>Sign in using email address Firebase for authentication</p> <p>Firebase : User : Username , email and password</p> |

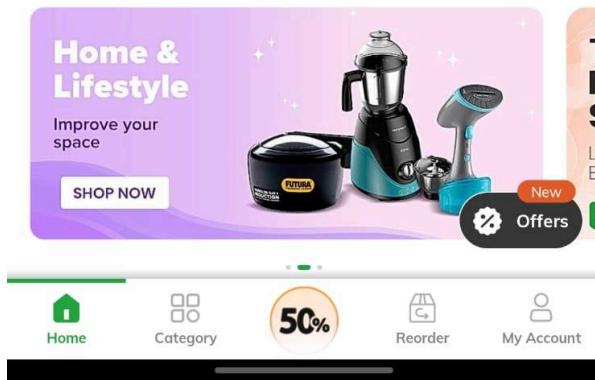
2



DMart Exclusive



Best Offers



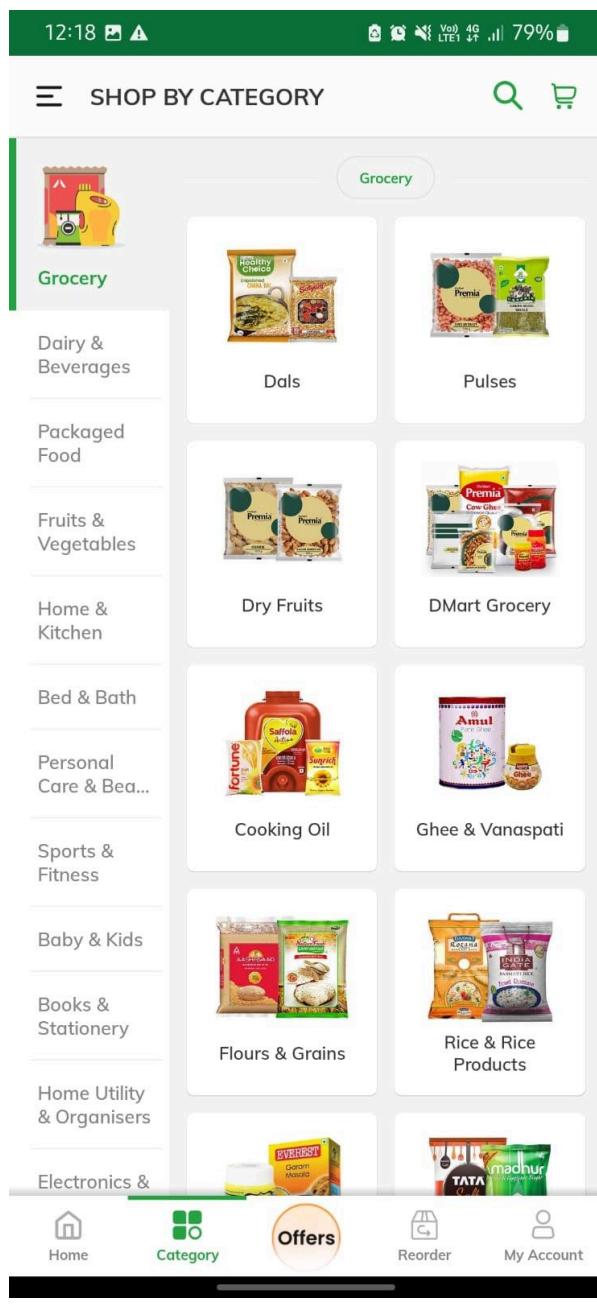
User Homepage :

Icons for notifications, cart, home, category, reorder, profile.

Search bar

Display best offers

3



Shop by Category :

Display various categories of products and the subcategories for products.

For navigation the app uses the left sidebar for category navigation.

For each active bar, an image will be displayed along with the category name.

4

12:19 4G 79%

← Chocolates & Candies

All Chocolates Candies Compound Chocolates

Cadbury Dairy Milk Chocolate
Mini Treats : 91 g
91 g (₹ 1.05 / 1 gm)
You Pay ₹96 You Save ₹24

Nestle Kitkat Chocolate : 119 gms
119 g (₹ 0.84 / 1 gm)
DMart ₹100 MRP ₹149 Incl. of all taxes ₹49 OFF

Ferrero Rocher - 16 Pieces : 200 gms
200 g (₹ 2.50 / 1 gm)
DMart ₹499 MRP ₹589 Incl. of all taxes ₹90 OFF

Cadbury Dairy Milk Silk
Chocolate Mini Treats : 153 g
153 gm (₹ 0.86 / 1 gm)

₹49 Saved ₹266 Cart Total 2 Item(s) >>

Search Results :

Display product details, name of the product, quantity, price, saving price and image of the product.

Add to cart/ Remove from cart.

Increment the item already present in the cart and decrement the product already present in the cart.

Like the product, add to your favorite list.

5

The screenshot shows a mobile application interface for a shopping cart. At the top, there is a green header bar with the time '12:19' and battery level '79%'. Below the header, the word 'Cart' is displayed with a back arrow icon. To the right of 'Cart' are icons for search, a shopping cart containing '₹266', and a user profile.

Below the header, there is a section for 'Savings ₹49' and 'Cart Total ₹266'. A message states 'All offer(s) will be applied on checkout'.

The main area is titled 'My Cart (2 item(s))' and includes a 'SHARE CART' button. It lists two items:

- Cadbury Dairy Milk Chocolate Mini Treats ...**
91 g
You Pay ₹96 You Save ₹24
Quantity: 1 Buttons: - (trash), + (increase)
- Cadbury Dairy Milk Silk Roast Almond Bar...**
143 g
You Pay ₹170 You Save ₹25
Quantity: 1 Buttons: - (trash), + (increase)

A 'Remove all' button is located below the item list. At the bottom of the screen, there is a 'PROCEED TO CHECKOUT' button and a note: 'Your minimum order value should be ₹ 500.' A black horizontal bar is at the very bottom.

Cart:

Display all items added to cart.

Increment the item already present in the cart and decrement the product already present in the cart.

Remove all the products from the cart using the "Remove all" button.

MPL Experiment 1

Name: Anuprita Mhapankar

Class: D15A

Roll no:28

Aim: Installation and Configuration of Flutter Environment.

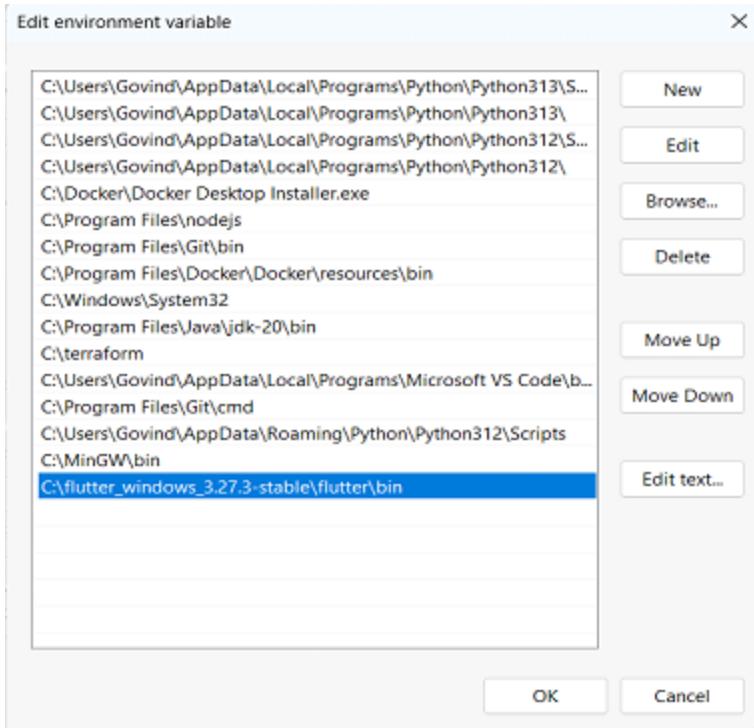
Step 1: Install Flutter

- Download Flutter SDK from the official Flutter website (<https://flutter.dev/>).
- Download the Flutter SDK for your operating system (Windows, macOS, or Linux).

The screenshot shows the official Flutter website's 'Get started' page. At the top, there is a navigation bar with links for Multi-Platform, Development, Ecosystem, Showcase, Docs, and a search icon. A prominent blue button labeled 'Get started' is located on the right. Below the navigation, a banner reads 'Celebrating Flutter's production era! Learn more' and 'Also, check out What's new on the website.' On the left, a sidebar contains links for Get started, Set up Flutter, Learn Flutter, Stay up to date, App solutions, User interface, Introduction, Widget catalog, Layout, Adaptive & responsive design, and Design & theming. The main content area features a heading 'Choose your development platform to get started' with four options: Windows (Current device), macOS, Linux, and ChromeOS. A note below the Windows option states: 'Developing in China' and provides instructions for using Flutter in China if you're developing there. The overall layout is clean and modern, with a dark header and light body text.

The screenshot shows the 'Install the Flutter SDK' page from the Flutter website. The layout is similar to the previous 'Get started' page, with a sidebar on the left and a main content area on the right. The sidebar includes links for Get started, Set up Flutter, Learn Flutter, Stay up to date, App solutions, User interface, Introduction, Widget catalog, Layout, Adaptive & responsive design, Design & theming, Interactivity, and Assets & media. The main content area has a heading 'Install the Flutter SDK' with a sub-instruction: 'To install the Flutter SDK, you can use the VS Code Flutter extension or download and install the Flutter bundle yourself.' Below this are two buttons: 'Use VS Code to install' and 'Download and install'. To the right of the main content, there is a 'Contents' sidebar with links for Verify system requirements, Hardware requirements, Software requirements, Configure a text editor or IDE, Install the Flutter SDK, Configure Android development, Configure the Android toolchain in Android Studio, Configure your target Android device, Agree to Android licenses, Check your development setup, Run Flutter doctor, Troubleshoot Flutter doctor issues, Start developing Android on Windows apps with Flutter, and Start developing iOS apps with Flutter. The 'Install the Flutter SDK' section also includes a note about the flutter_windows_3.27.3-stable.zip file and instructions for older builds.

- Extract the downloaded zip file to a preferred location on your computer (e.g., C:\src\flutter for Windows).
- Add Flutter to the PATH
- Locate the flutter\bin directory in the extracted Flutter folder.
- Add this directory to your system's PATH environment variable.



Verify the Installation

- Open a terminal or command prompt.
- Run the command: **flutter** and **flutter doctor**.

```
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 3.27.1, on Microsoft Windows [Version 10.0.22631.4751], locale en-IN)
[!] Windows Version (the doctor check crashed)
  X Due to an error, the doctor check did not complete. If the error message below is not helpful, please let us know
    about this issue at https://github.com/flutter/flutter/issues.
  X ProcessException: Failed to find "powershell" in the search path.
    Command: powershell
[!] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
[!] Chrome - develop for the web
[!] Visual Studio - develop Windows apps
  X Visual Studio not installed; this is necessary to develop Windows apps.
    Download at https://visualstudio.microsoft.com/downloads/.
    Please install the "Desktop development with C++" workload, including all of its default components
[!] Android Studio (version 2023.3)
[!] VS Code (version 1.96.4)
[!] Connected device (4 available)
[!] Network resources

! Doctor found issues in 2 categories.

C:\Users\User>
```

```
Command Prompt - flutter  + - 

Welcome to Flutter! - https://flutter.dev

The Flutter tool uses Google Analytics to anonymously report feature usage
statistics and basic crash reports. This data is used to help improve
Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable
reporting, type 'flutter config --no-analytics'. To display the current
setting, type 'flutter config'. If you opt out of analytics, an opt-out
event will be sent, and then no further information will be sent by the
Flutter tool.

By downloading the Flutter SDK, you agree to the Google Terms of Service.
The Google Privacy Policy describes how data is handled in this service.

Moreover, Flutter includes the Dart SDK, which may send usage metrics and
crash reports to Google.

Read about data we send with crash reports:
https://flutter.dev/to/crash-reporting

See Google's privacy policy:
https://policies.google.com/privacy

To disable animations in this tool, use
'flutter config --no-cli-animations'.

The Flutter CLI developer tool uses Google Analytics to report usage and diagnostic
data along with package dependencies, and crash reporting to send basic crash
reports. This data is used to help improve the Dart platform, Flutter framework,
and related tools.

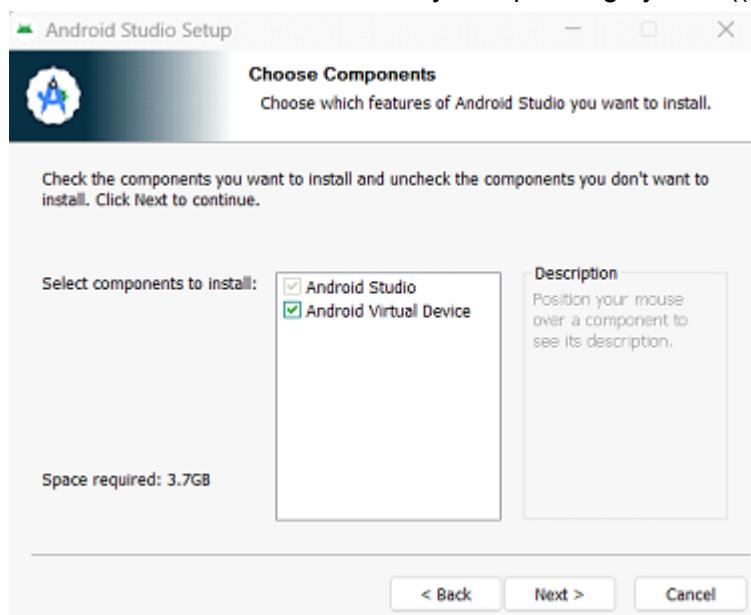
Telemetry is not sent on the very first run. To disable reporting of telemetry,
run this terminal command:

  flutter --disable-analytics

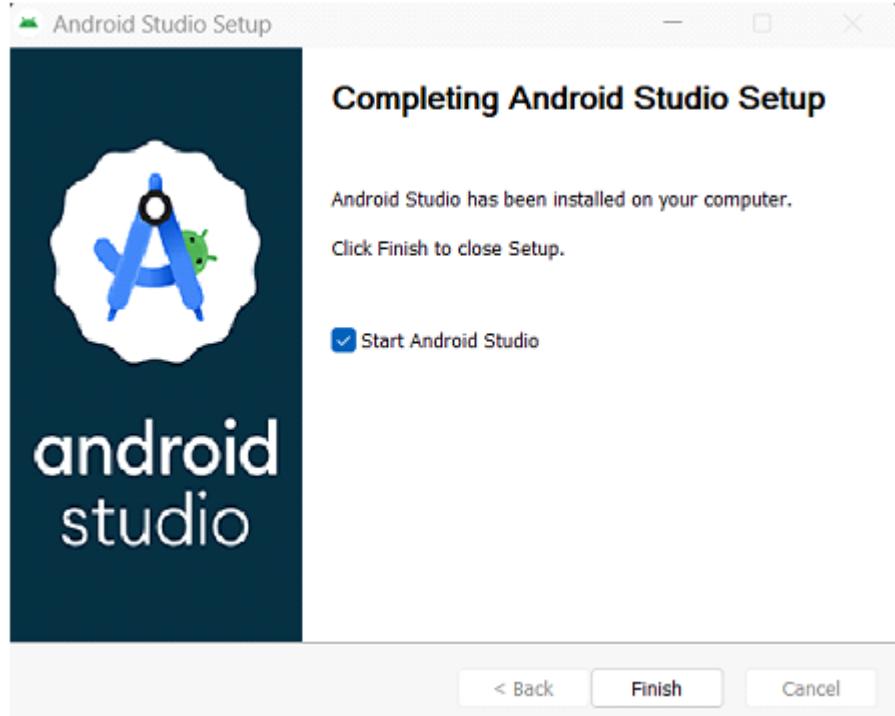
If you opt out of telemetry, an opt-out event will be sent, and then no further
```

Step 2: Install Android Studio

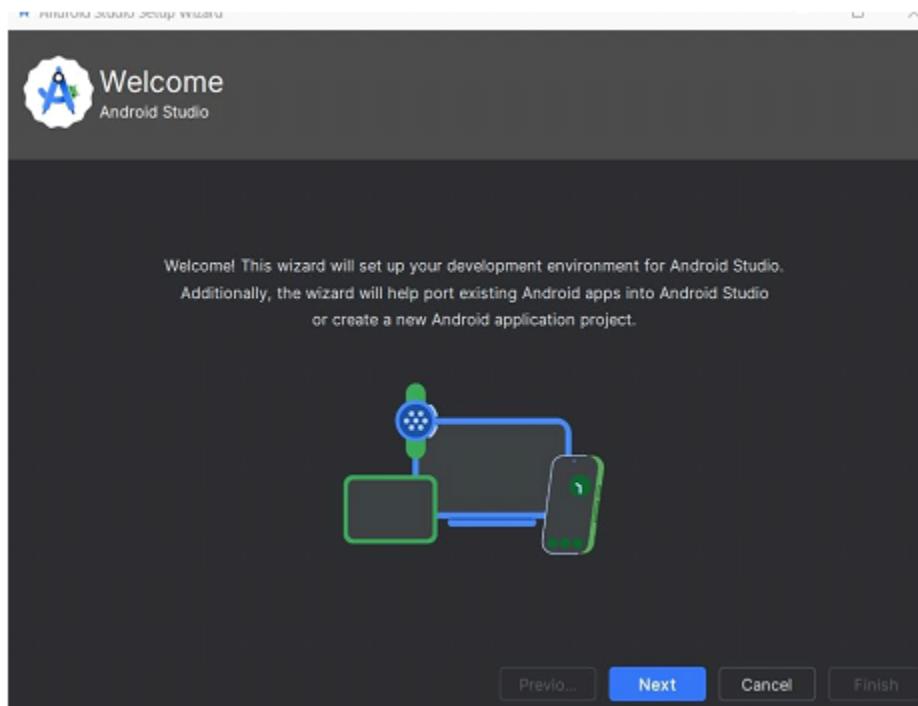
- Download Android Studio
- Go to the Android Studio website. (<https://developer.android.com/studio>)
- Download the installer for your operating system ((Windows, macOS, or Linux)).



- Run the installer and follow the setup wizard.
- Choose the standard installation option.

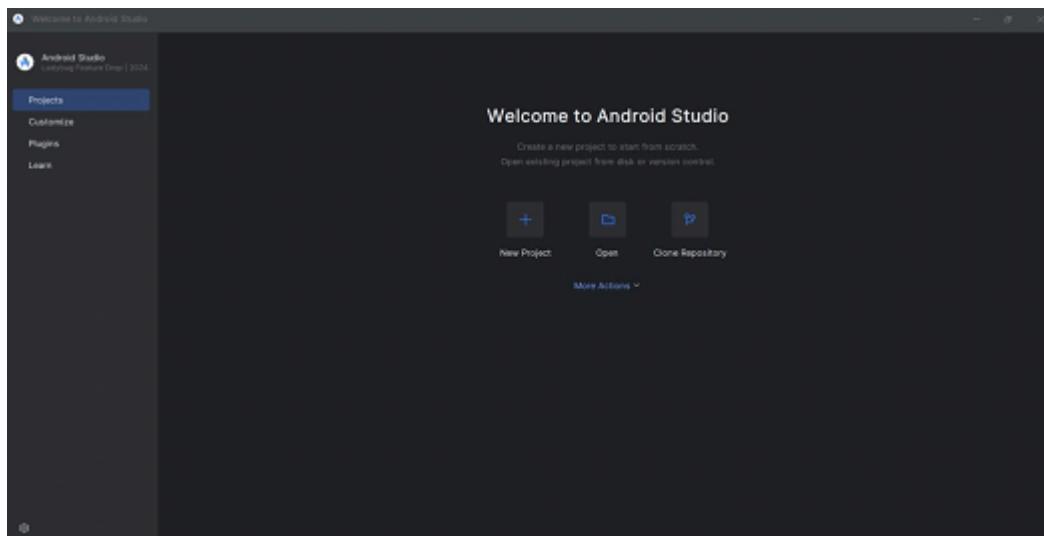
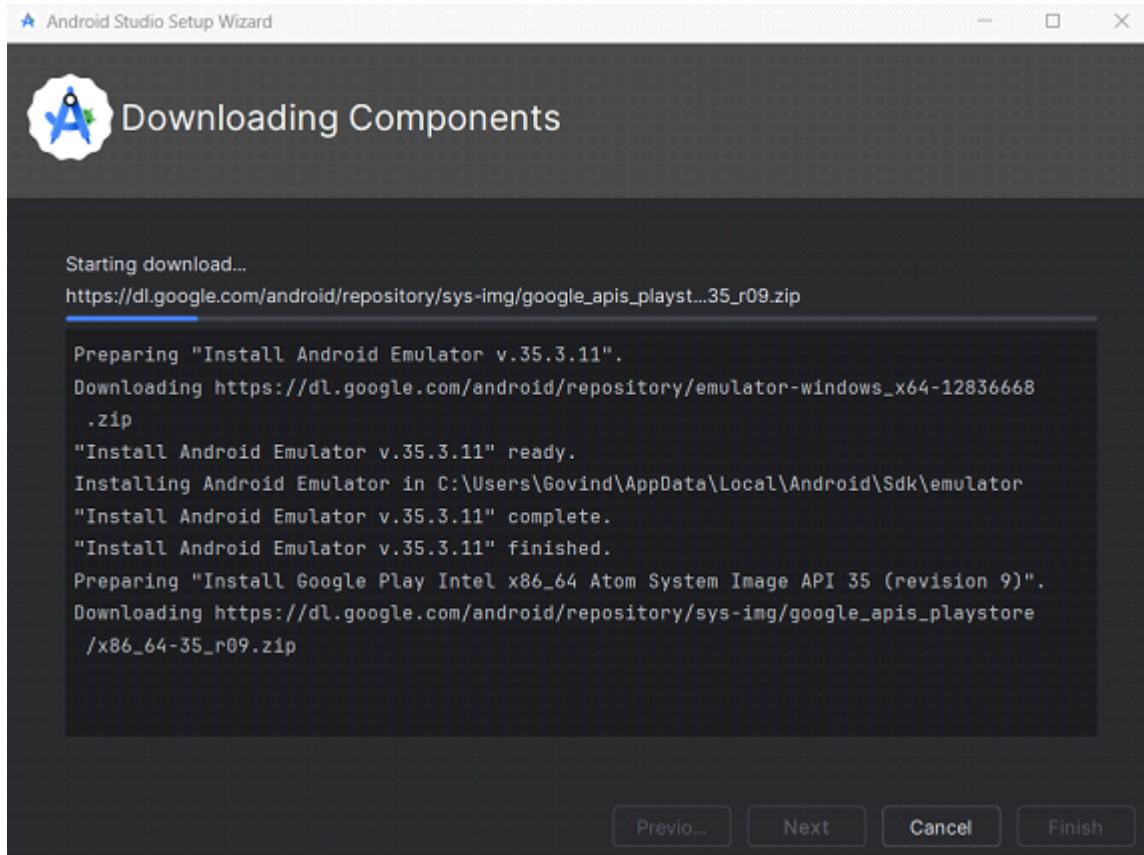


- Install Android SDK Tools
- Open Android Studio.



- Go to Settings/Preferences > Appearance & Behavior > System Settings > Android SDK.

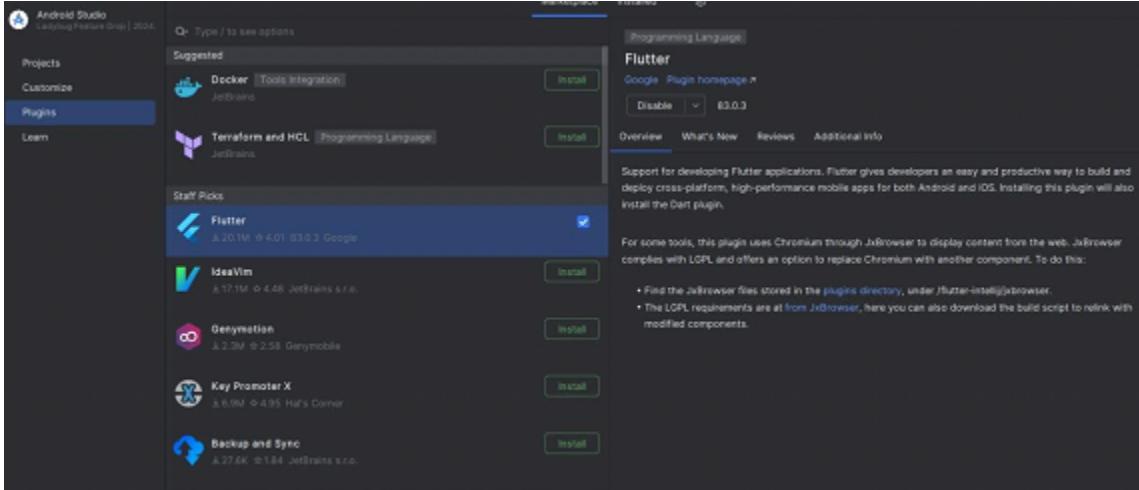
- Select the latest Android API level.
- Ensure "Android SDK Platform" and "Android Virtual Device (AVD)" are selected.
- Click "Apply" and wait for the components to install.



Step 3: Connect Flutter with Android Studio

- Install Flutter and Dart Plugins
- Open Android Studio. Go to File > Settings (Windows/Linux) > Plugins.

- Search for "Flutter" and click "Install." Dart will be installed automatically.
- Restart Android Studio.



Step 4: Create a New Flutter Project

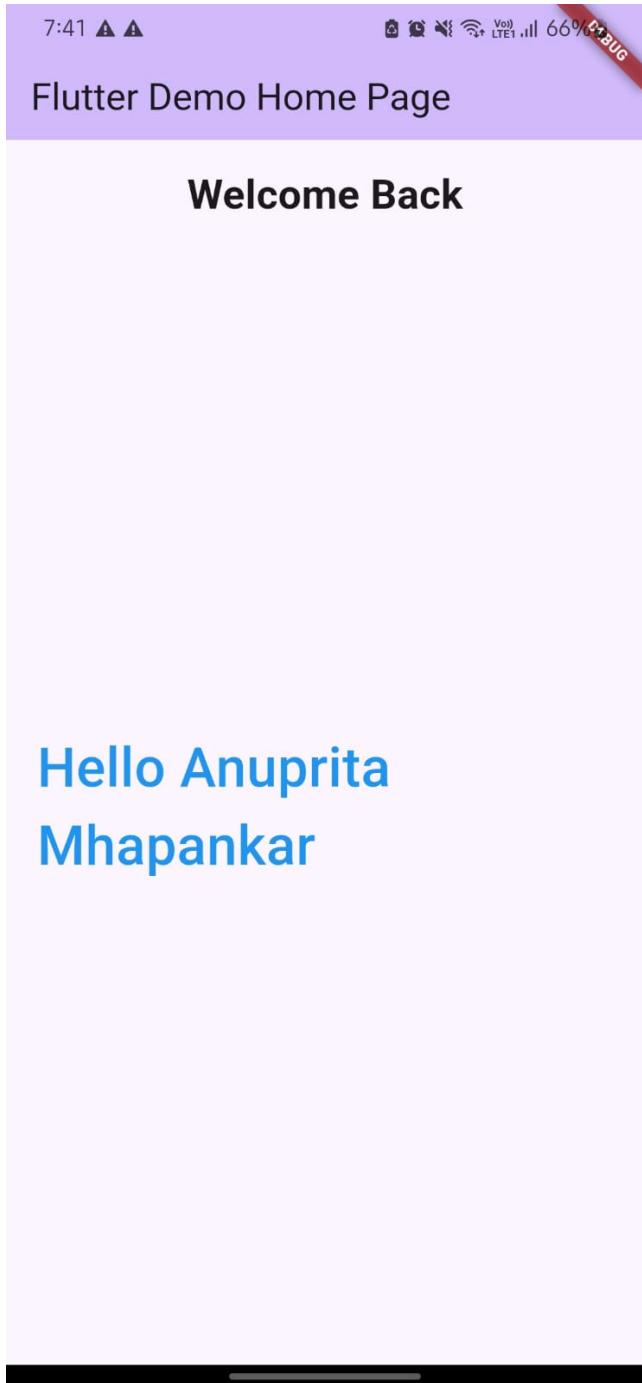
- Click on New Flutter Project.
- Enter project details and select the Flutter SDK path.
- Click "Finish" to create the project.
- Connect the USB to the device and run the flutter application.

NOTE: In your mobile device, make sure the USB debugging is turned on.

Code:

```
class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: Text(widget.title),
      ),
      body: Column(
        children: [
          const Padding(
            padding: EdgeInsets.all(16.0),
            child: Text(
              'Welcome Back',
              style: TextStyle(
                fontSize: 24,
                fontWeight: FontWeight.bold,
              ),
            ),
          ),
        ],
      ),
      const Spacer(),
    );
  }
}
```

```
const Padding(  
    padding: EdgeInsets.all(20.0),  
    child: Text(  
        'Hello Anuprita Mhapankar',  
        style: TextStyle(  
            fontSize: 32,  
            color: Colors.blue,  
            fontWeight: FontWeight.w500,  
        ),  
    ),  
) ,  
    const Spacer(),  
],  
) ;  
}  
}
```



Conclusion: Hello World, is successfully run on the flutter app.

MPL Experiment 2

Name: Anuprita Mhapankar

Class: D15A

Roll no:28

Aim: To design Flutter UI by including common widgets.

Theory:

Designing a Flutter UI involves using common widgets effectively to create a structured and interactive interface. Layout widgets like `Container`, `Column`, `Row`, and `GridView` help organize elements, while interactive widgets like `GestureDetector`, `InkWell`, and `TextField` enhance user engagement.

For displaying content, widgets like `Text`, `Image`, `Card`, and `ListView` are essential. Navigation widgets such as `BottomNavigationBar`, `Drawer`, and `Navigator` enable seamless movement between screens. Managing state using `setState`, `Provider`, or advanced solutions like `Riverpod` ensures smooth UI updates.

For a **Dmart-like home screen**, a combination of the following widgets can be used:

- `ListView` with a `CarouselSlider` for displaying promotional banners.
- `GridView` for categorizing products.
- `Card` widget to showcase trending items.
- `BottomNavigationBar` for consistent navigation between screens.

Steps:

Step 1: Create a new Flutter project or open an existing one.

Step 2: Design the layout using `Scaffold`, incorporating `AppBar` and `BottomNavigationBar`.

Step 3: Use `CarouselSlider` inside a `ListView` for banners.

Step 4: Implement `GridView` to display product categories.

Step 5: Use `Card` widgets inside a `ListView` to show trending items.

Step 6: Ensure navigation using `BottomNavigationBar` and `Navigator`.

Step 7: Apply `setState` or `Provider` for state management.

Code:

```
//home_page.dart
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:dmart/screens/login/login_page.dart';
import 'package:carousel_slider/carousel_slider.dart';
import 'package:dmart/common_components/header.dart';

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

  Future<void> _logout(BuildContext context) async {
    await FirebaseAuth.instance.signOut();

    // Clear login state from shared preferences
    SharedPreferences prefs = await SharedPreferences.getInstance();
    await prefs.setBool('isLoggedIn', false);

    Navigator.pushReplacement(
      context,
      MaterialPageRoute(builder: (context) => const LoginPage()),
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: const Header(),
      body: SingleChildScrollView(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.start,
          children: [
            // Parade of Deals Banner
            Container(
              width: double.infinity,
              height: 150,
              padding: const EdgeInsets.all(16.0),
            ),
          ],
        ),
      ),
    );
  }
}
```

```
        alignment: Alignment.center,
        color: Colors.blueAccent,
        child: const Text(
            'Parade of Deals',
            style: TextStyle(fontSize: 24, fontWeight:
FontWeight.bold, color: Colors.white),
            textAlign: TextAlign.center,
        ),
    ),
}

// Dmart Exclusive
const Padding(
    padding: EdgeInsets.all(16.0),
    child: Text(
        'Dmart Exclusive',
        style: TextStyle(fontSize: 20, fontWeight:
FontWeight.bold),
    ),
),
CarouselSlider(
    options: CarouselOptions(enableInfiniteScroll: false),
    items: List.generate(5, (index) {
        return Card(
            margin: const EdgeInsets.symmetric(horizontal: 8),
            child: Container(
                height: 150,
                width: double.infinity,
                alignment: Alignment.center,
                child: Text('Exclusive Item ${index + 1}'),
            ),
        );
    }),
),
}

// Best Offer
const Padding(
    padding: EdgeInsets.all(16.0),
    child: Text(
```

```
        'Best Offer',
        style: TextStyle(fontSize: 20, fontWeight:
FontWeight.bold),
    ),
),
CarouselSlider(
options: CarouselOptions(enableInfiniteScroll: false),
items: List.generate(5, (index) {
return Card(
margin: const EdgeInsets.symmetric(horizontal: 8),
child: Container(
height: 150,
width: double.infinity,
alignment: Alignment.center,
child: Text('Best Offer Item ${index + 1}'),
),
),
);
}),
),
],
),
);
);
}
}
}
```

```
//account_page.dart
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:dmart/screens/login/login_page.dart';
```

```
import 'package:dmart/common_components/header.dart';

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

  Future<void> _logout(BuildContext context) async {
    await FirebaseAuth.instance.signOut();

    // Clear login state from shared preferences
    SharedPreferences prefs = await SharedPreferences.getInstance();
    await prefs.setBool('isLoggedIn', false);

    // Navigate back to login screen
    Navigator.pushAndRemoveUntil(
      context,
      MaterialPageRoute(builder: (context) => const LoginPage()),
      (Route<dynamic> route) => false, // Remove all previous
    routes
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: const Header(),
      body: Column(
        children: [
          Expanded(
            child: ListView(
              children: [
                _buildAccountOption(Icons.person, "My Profile"),
                _buildAccountOption(Icons.help, "Help @ DMart Ready"),
                _buildAccountOption(Icons.logout, "Sign Out",
isLogout: true, onTap: () => _logout(context)),
                const SizedBox(height: 10),
                _buildPromoBanner(),
              ],
            ),
          ),
        ],
      ),
    );
  }
}
```

```
        ),
    ],
),
);
}

Widget _buildAccountOption(IconData icon, String title, {bool
isLogout = false, VoidCallback? onTap}) {
    return ListTile(
        leading: Icon(icon, color: isLogout ? Colors.red :
Colors.black54),
        title: Text(title, style: TextStyle(fontSize: 16, fontWeight:
FontWeight.w500)),
        trailing: const Icon(Icons.arrow_forward_ios, size: 16, color:
Colors.black54),
        onTap: onTap ?? () {}, // Provide navigation or functionality
    );
}

Widget _buildPromoBanner() {
    return Padding(
        padding: const EdgeInsets.all(12.0),
        child: Container(
            decoration: BoxDecoration(
                color: Colors.green.shade100,
                borderRadius: BorderRadius.circular(8),
            ),
            padding: const EdgeInsets.all(12),
            child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                    const Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                            Text(
                                "You're SAVING a LOT on every order",
                                style: TextStyle(fontSize: 14, fontWeight:
FontWeight.bold),
                            ),
                            Text("Up to 50% off selected items"),
                        ],
                    ),
                ],
            ),
        ),
    );
}
```

```
        ),
        SizedBox(height: 5),
        Text(
            "Check My Savings",
            style: TextStyle(fontSize: 14, fontWeight:
FontWeight.bold, color: Colors.green),
            ),
        ],
    ],
),
),
);
}
}
```

```
//sidebar.dart
import 'package:flutter/material.dart';

class Sidebar extends StatelessWidget {
@override
Widget build(BuildContext context) {
```

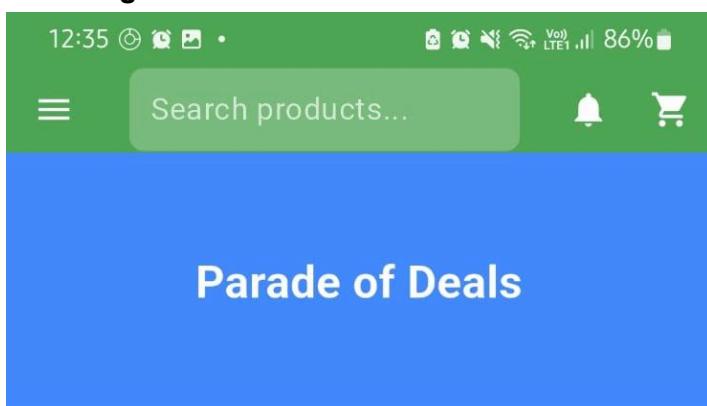
```
return Drawer(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.start,
    children: [
      UserAccountsDrawerHeader(
        decoration: BoxDecoration(color: Colors.green),
        accountName: Text(
          "Hi, Anuprita",
          style: TextStyle(fontSize: 18, fontWeight:
FontWeight.bold),
        ),
        accountEmail: Text("400615, Thane (West)"),
        currentAccountPicture: CircleAvatar(
          backgroundColor: Colors.white,
          child: Icon(Icons.person, size: 40, color:
Colors.green),
        ),
      ),
      _buildMenuItem(Icons.category, "Shop by Category", context),
      _buildMenuItem(Icons.help, "Help @ Dmart Ready", context),
      _buildMenuItem(Icons.policy, "Refund, Terms and Policies",
context),
      _buildMenuItem(Icons.question_answer, "Frequently Asked
Questions", context),
      _buildMenuItem(Icons.info, "About Us", context),
      Spacer(),
      Padding(
        padding: const EdgeInsets.all(16.0),
        child: Text("Version 5.1.9", style: TextStyle(color:
Colors.grey)),
      ),
    ],
  ),
);
}

Widget _buildMenuItem(IconData icon, String title, BuildContext
context) {
```

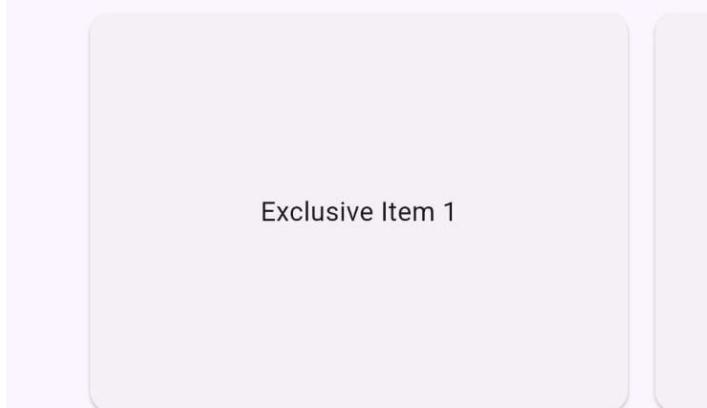
```
        return ListTile(
            leading: Icon(icon, color: Colors.green),
            title: Text(title, style: TextStyle(fontSize: 16)),
            onTap: () {
                Navigator.pop(context); // Close drawer when an item is tapped
            },
        );
    }
}
```

Output:

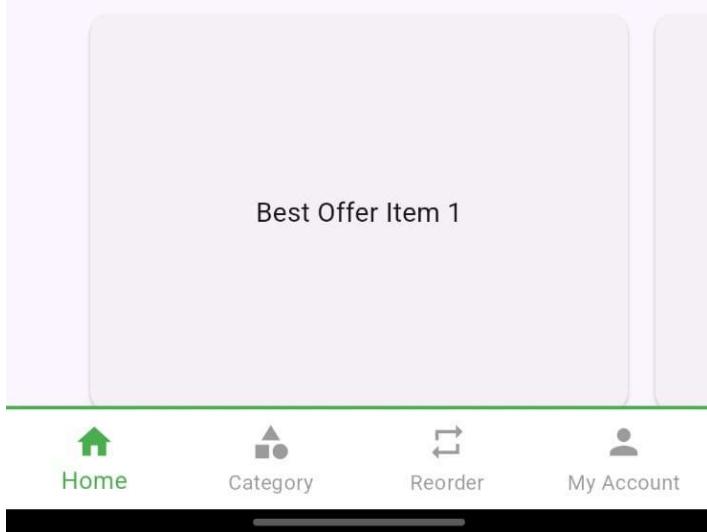
HomePage:



Dmart Exclusive



Best Offer



MyAccount:

The screenshot shows the 'MyAccount' screen of the DMart Ready app on a smartphone. At the top, there is a green header bar with the time '12:01' and battery level '48%'. Below the header is a navigation bar with a search bar containing 'Search products...', a bell icon, and a shopping cart icon.

The main content area displays three menu items:

- My Profile** (indicated by a person icon)
- Help @ DMart Ready** (indicated by a question mark icon)
- Sign Out** (indicated by a red exit icon)

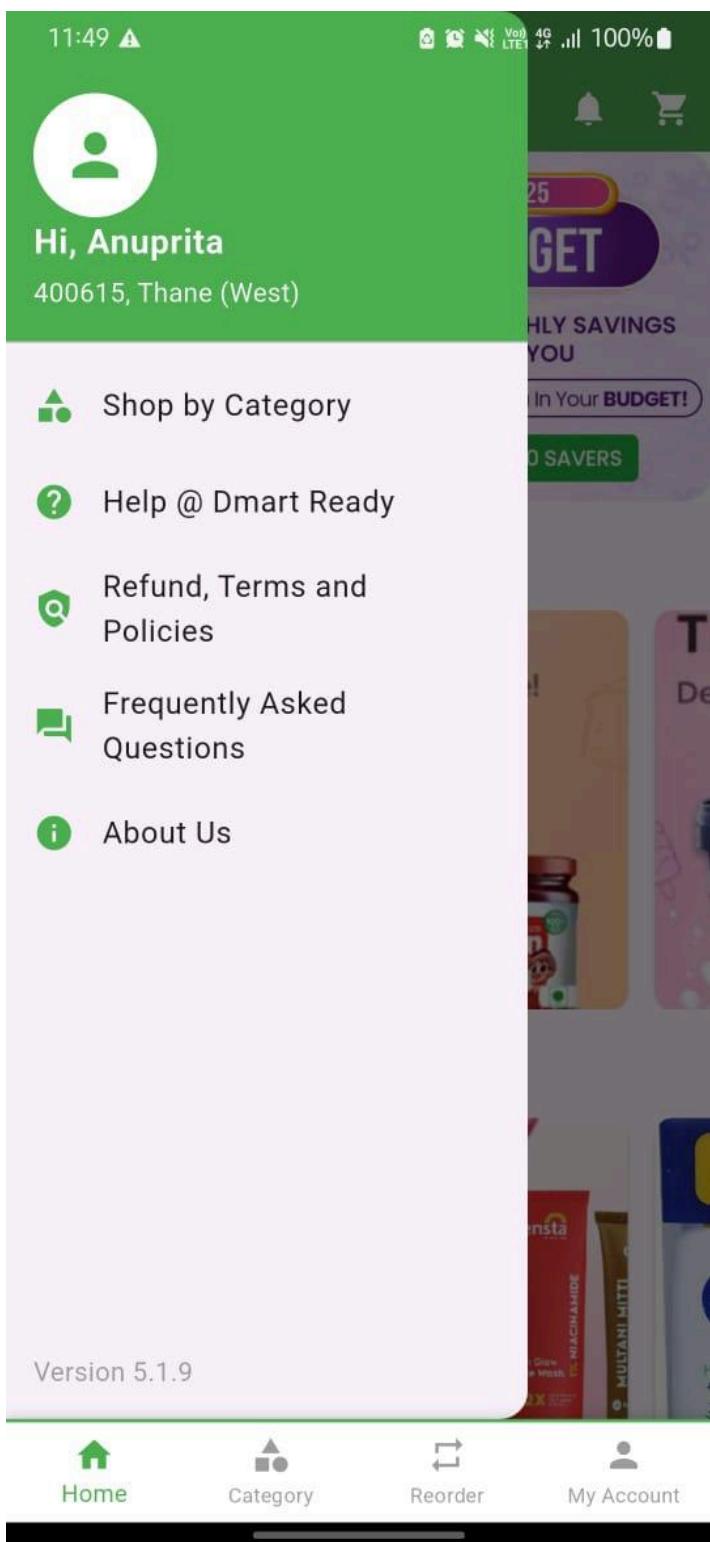
Below these items is a green callout box with the text 'You're SAVING a LOT on every order' and a 'Check My Savings' button.

At the bottom of the screen are four navigation icons:

- Home** (house icon)
- Category** (grid icon)
- Reorder** (refresh icon)
- My Account** (person icon)

A black navigation bar is visible at the very bottom of the phone screen.

Sidebar:



MPL Experiment 3

Name: Anuprita Mhapankar

Class: D15A

Roll no:28

Aim: To include icons, images, Fonts in Flutter app.

Theory:

Incorporating icons, images, and custom fonts in a Flutter application enhances the visual appeal and improves the user experience. Flutter provides different ways to add these elements:

1. Icons:

Icons can be added using the built-in `Icons` class with the `Icon` widget. Custom icons can also be used via the `flutter_launcher_icons` package.

2. Images:

Images can be displayed in two ways:

- **From assets:** Images stored locally in the app's assets folder can be loaded using the `Image.asset()` method.
- **From the internet:** Images can be fetched dynamically from a URL using the `Image.network()` method.

Examples:

- **Loading an image from assets:**

```
Image.asset(  
  'assets/images/sample.png',  
  fit: BoxFit.cover,  
  width: double.infinity,  
)
```

- **Loading an image from a URL:**

```
Image.network(  
  'https://example.com/sample.jpg',
```

```

        width: double.infinity,
        fit: BoxFit.cover,
    )

```

3. Fonts:

Custom fonts improve typography and branding. To add custom fonts, font files must be placed in the `assets/fonts/` directory and declared in `pubspec.yaml` under the `flutter` section. Using `TextStyle` with the `fontFamily` property applies the custom font to text elements.

Steps:

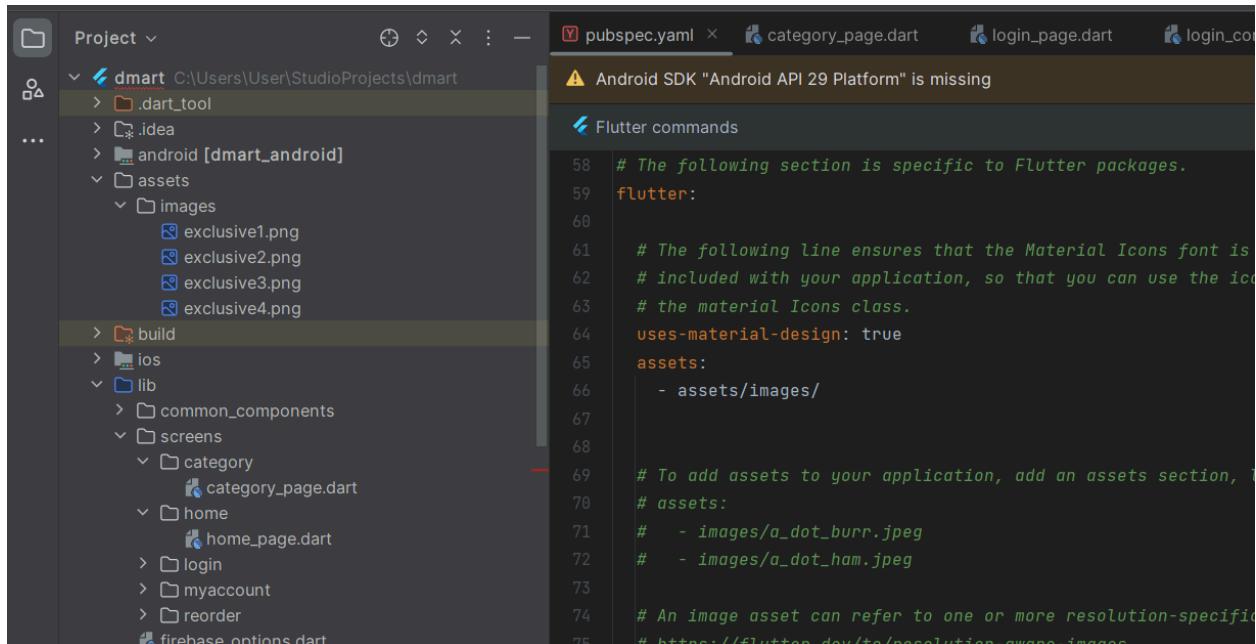
Step 1: Create an `assets` folder inside the project directory. Inside the `assets` folder, create an `images` folder and add the required images.

Step 2: Open `pubspec.yaml` and add the following under the `flutter` section:

```
flutter:
```

```
  assets:
```

```
    - assets/images/
```



Step 3: Run the following command in the terminal to apply the changes:

```
flutter pub get
```

Code:

```
//home_page.dart
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:dmart/screens/login/login_page.dart';
import 'package:carousel_slider/carousel_slider.dart';
import 'package:dmart/common_components/header.dart';
import 'package:dmart/common_components/sidebar.dart';

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

  Future<void> _logout(BuildContext context) async {
    await FirebaseAuth.instance.signOut();

    // Clear login state from shared preferences
    SharedPreferences prefs = await SharedPreferences.getInstance();
    await prefs.setBool('isLoggedIn', false);

    Navigator.pushReplacement(
      context,
      MaterialPageRoute(builder: (context) => const LoginPage()),
    );
  }

  @override
  Widget build(BuildContext context) {
    // Image Lists
    final List<String> dmartExclusiveImages = [
      'assets/images/exclusive1.png',
      'assets/images/exclusive2.png',
      'assets/images/exclusive3.png',
      'assets/images/exclusive4.png',
    ];

    final List<String> bestOfferImages = [
```

```
'https://clensta.com/cdn/shop/files/mob-banner4_600x.jpg?v=1737626058'
,

'https://rukminim2.flixcart.com/fk-p-harbour/600/600/images/incom/images/product/Nivea-Body-Lotion-Buy-3-Get-1-Free-Aloe-Hydration-2-x-75-ml-Extra-Whitening-75-mlBody-Milk-75-ml-1601886438-10076795-1.jpg?q=60',
];

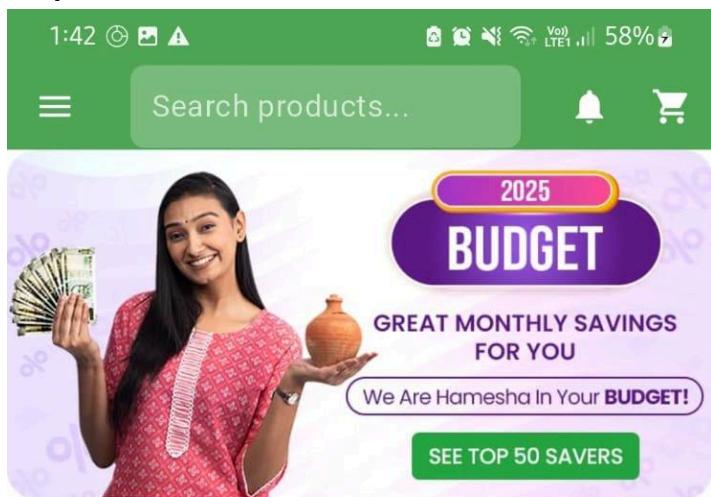
return Scaffold(
  appBar: const Header(),
  drawer: Sidebar(),
  body: SingleChildScrollView(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.start,
      children: [
        // Parade of Deals Banner
        SizedBox(
          width: double.infinity,
          child: Image.network(
            'https://cdn.dmart.in/images/rwd-mobile/banners/curated/29jan25-curated-budget-hp.jpg',
            fit: BoxFit.cover, // Ensures the image covers the full space
          ),
        ),
        // Dmart Exclusive
        const Padding(
          padding: EdgeInsets.all(16.0),
          child: Text(
            'Dmart Exclusive',
            style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
          ),
        ),
        CarouselSlider(
          options: CarouselOptions(enableInfiniteScroll: false),
```

```
        items: dmartExclusiveImages.map((imagePath) {
            return Card(
                margin: const EdgeInsets.symmetric(horizontal: 8),
                child: ClipRRect(
                    borderRadius: BorderRadius.circular(10),
                    child: Image.asset(imagePath, fit: BoxFit.cover,
width: double.infinity),
                ),
            );
        }).toList(),
    ),
}

// Best Offer
const Padding(
    padding: EdgeInsets.all(16.0),
    child: Text(
        'Best Offer',
        style: TextStyle(fontSize: 20, fontWeight:
FontWeight.bold),
    ),
),
CarouselSlider(
    options: CarouselOptions(enableInfiniteScroll: false),
    items: bestOfferImages.map((imageUrl) {
        return Card(
            margin: const EdgeInsets.symmetric(horizontal: 8),
            child: ClipRRect(
                borderRadius: BorderRadius.circular(10),
                child: Image.network(
                    imageUrl,
                    width: double.infinity,
                    fit: BoxFit.cover,
                ),
            ),
        );
    }).toList(),
),
],
```

) ,
),
);
}
}

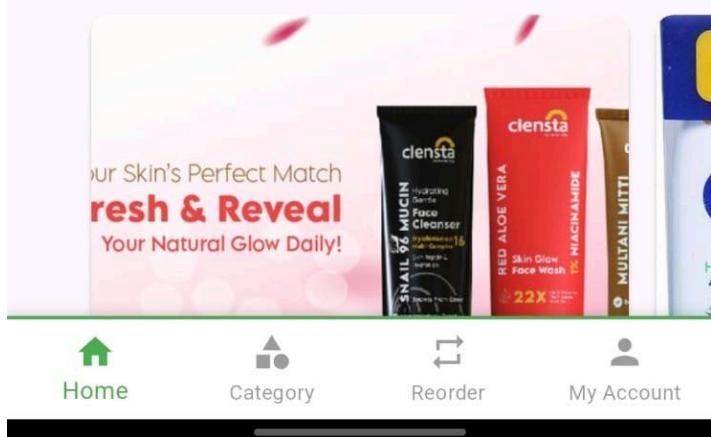
Output:

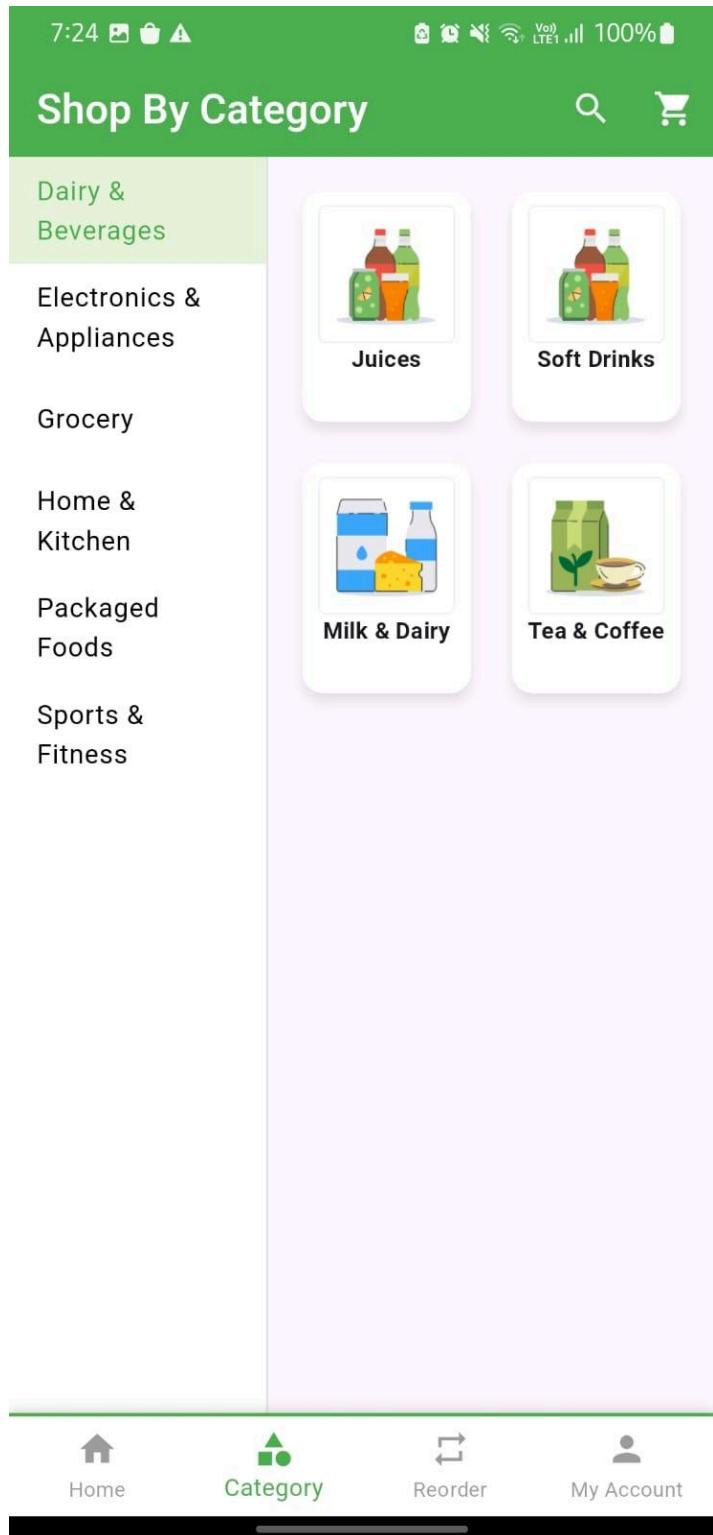


Dmart Exclusive



Best Offer





7:44

100%



Cooking Oil

**Fortune Sunlite Sunflower Oil**

Qty: 910 g

DMart: ₹150

₹40 OFF

MRP: ₹190



ADD

**Gemini Refined Sunflower Oil**

Qty: 4.550 kg

DMart: ₹782

₹208 OFF

MRP: ₹990



ADD

**Dhara Refined Sunflower Oil**

Qty: 13.65 kg

DMart: ₹2219

₹751 OFF

MRP: ₹2970



ADD

**Dalda Refined Sunflower Oil**

Qty: 910 g

DMart: ₹142

₹43 OFF

MRP: ₹185



ADD

MPL Experiment 4

Name: Anuprita Mhapankar

Class: D15A

Roll no:28

Aim: To create an interactive Form using form widget

Theory:

Creating an interactive form in Flutter requires using form-related widgets to collect and validate user input efficiently. The `Form` widget, combined with `TextField`, provides a structured way to manage input fields. Various input widgets like `TextField`, `DropdownButton`, `Checkbox`, `Radio`, and `Switch` allow users to enter data in different formats.

Validation and state management can be handled using the `GlobalKey<FormState>` to validate inputs before submission. Wrapping the form in a `SingleChildScrollView` ensures smooth scrolling when multiple fields are present.

A `RaisedButton` (deprecated) or `ElevatedButton` can trigger validation and submission logic. To enhance usability and create a responsive experience, proper padding, spacing, and `InputDecoration` should be applied.

Steps:

Step 1: Create a new Flutter project or open an existing one.

Step 2: Define a `Form` widget inside a `StatefulWidget` to manage user input.

Step 3: Use `TextField` for text input fields with validation logic.

Step 4: Include other input widgets such as `DropdownButton`, `Checkbox`, `Radio`, and `Switch` for additional user selections.

Step 5: Wrap the form inside a `SingleChildScrollView` to ensure smooth scrolling.

Step 6: Implement an `ElevatedButton` to trigger form validation and submission.

Step 7: Use `GlobalKey<FormState>` to manage form validation.

Code:

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:dmart/screens/home/home_page.dart';
import 'package:shared_preferences/shared_preferences.dart';
```

```
class LoginPage extends StatefulWidget {
  const LoginPage({super.key});

  @override
  State<LoginPage> createState() => _LoginPageState();
}

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

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

  Future<void> _login() async {
    if (_formKey.currentState!.validate()) {
      return; // Stop execution if validation fails
    }

    try {
      await _auth.signInWithEmailAndPassword(
        email: _emailController.text.trim(),
        password: _passwordController.text.trim(),
      );

      // Save login state
      SharedPreferences prefs = await SharedPreferences.getInstance();
      await prefs.setBool('isLoggedIn', true);

      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => const HomePage()),
      );
    } catch (e) {
      ScaffoldMessenger.of(context).showSnackBar(

```

```
        SnackBar(content: Text('Login failed: ${e.toString()}'))),
    );
}
}

String? _validateEmail(String? value) {
if (value == null || value.isEmpty) {
    return "Email cannot be empty";
}
String emailPattern =
r'^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'*+/=?^_`{|}~-]+@[a-zA-Z0-9]+\.[a-zA-Z]+'
;
RegExp regex = RegExp(emailPattern);
if (!regex.hasMatch(value)) {
    return "Enter a valid email";
}
return null;
}

String? _validatePassword(String? value) {
if (value == null || value.isEmpty) {
    return "Password cannot be empty";
}
if (value.length < 6) {
    return "Password must be at least 6 characters";
}
return null;
}

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

```
        },
    ),
    backgroundColor: Colors.white,
    elevation: 0,
    title: const Text(
        "Let's Get You Logged In",
        style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold,
color: Colors.black),
    ),
    centerTitle: false,
),
body: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Form(
        key: _formKey, // Wrap the form for validation
        child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
                const Text(
                    "Enter your Email",
                    style: TextStyle(fontSize: 14, color: Colors.grey),
                ),
                const SizedBox(height: 8),
                TextFormField(
                    controller: _emailController,
                    decoration: InputDecoration(
                        border: OutlineInputBorder(
                            borderRadius: BorderRadius.circular(8),
                            borderSide: const BorderSide(color: Colors.grey),
                        ),
                        focusedBorder: OutlineInputBorder(
                            borderRadius: BorderRadius.circular(8),
                            borderSide: const BorderSide(color: Colors.blue),
                        ),
                    ),
                    keyboardType: TextInputType.emailAddress,
                    validator: _validateEmail,
                ),
            ],
        ),
    ),
);
```

```
        const SizedBox(height: 15),
        const Text(
            "Enter your Password",
            style: TextStyle(fontSize: 14, color: Colors.grey),
        ),
        const SizedBox(height: 8),
        TextFormField(
            controller: _passwordController,
            obscureText: true,
            decoration: InputDecoration(
                border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(8),
                    borderSide: const BorderSide(color: Colors.grey),
                ),
                focusedBorder: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(8),
                    borderSide: const BorderSide(color: Colors.blue),
                ),
            ),
            validator: _validatePassword,
        ),
        const SizedBox(height: 20),
        SizedBox(
            width: double.infinity,
            height: 50, // Match button height in image
            child: ElevatedButton(
                onPressed: _login,
                style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.green, // Match button
color
                    foregroundColor: Colors.white,
                    shape: RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(8), //
Rectangular button
                ),
            ),
            child: const Text("LOGIN", style:
TextStyle(fontSize: 16)),
```

```
) ,  
),  
],  
) ,  
) ,  
) ;  
}  
}
```

Output:

Login Page

12:00



48%

← Let's Get You Logged In

Enter your Email

Enter your Password

LOGIN

11:14

VoIP 4G 100%

← Let's Get You Logged In

Enter your Email

Email cannot be empty

Enter your Password

Password cannot be empty

LOGIN

MPL Experiment 5

Name: Anuprita Mhapankar

Class: D15A

Roll no:28

Aim: To apply navigation, routing and gestures in Flutter App.

Theory:

Navigation & Routing:

Flutter provides multiple ways to navigate between screens (pages):

- 1. Using Navigator.push() and Navigator.pop()**

Push a new screen onto the stack and pop it to return to the previous one.

- 2. Named Routes**

Define routes in MaterialApp and navigate using Navigator.pushNamed().

- 3. GoRouter Package**

A declarative approach to handle navigation more efficiently.

Gestures:

Flutter's GestureDetector and InkWell widgets help in detecting user interactions like taps, swipes, and long presses. Some common gestures include:

- **onTap:** Detects a tap event.
- **onDoubleTap:** Recognizes double taps.
- **onLongPress:** Detects a long press on a widget.
- **onHorizontalDrag & onVerticalDrag:** Detects drag/swipe motions.

Steps:

Step 1: Create a Flutter project and define multiple screens.

Step 2: Set up named routes in `MaterialApp`.

Step 3: Implement navigation using `Navigator.push()` and `Navigator.pushNamed()`.

Step 4: Use `GestureDetector` to detect taps, swipes, and long presses.

Step 5: Add buttons or swipes to navigate between screens.

Code:

```
//bottom_tab.dart
import 'package:flutter/material.dart';
import 'package:dmart/screens/home/home_page.dart';
import 'package:dmart/screens/category/category_page.dart';
import 'package:dmart/screens/favorite/favorite_page.dart';
import 'package:dmart/screens/myaccount/account_page.dart';
```

```
class BottomTab extends StatefulWidget {
  const BottomTab({super.key});

  @override
  State<BottomTab> createState() => _BottomTabState();
}

class _BottomTabState extends State<BottomTab> {
  int _currentIndex = 0;
  final PageController _pageController = PageController();

  final List<Widget> _pages = [
    const HomePage(),
    const CategoryPage(),
    const FavoritePage(),
    const MyAccountPage(),
  ];

  @override
  void dispose() {
    _pageController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SizedBox(
        height: double.infinity,
        child: PageView(
          controller: _pageController,
          physics:
            const NeverScrollableScrollPhysics(), // Prevent swipe
        navigation
          children: _pages,
          onPageChanged: (index) {
            setState(() {

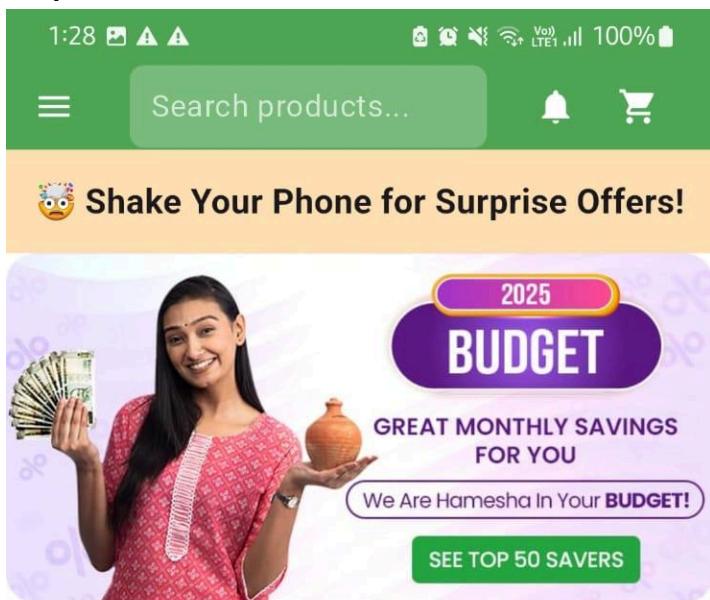
```

```
        _currentIndex = index;
    });
},
),
),
bottomNavigationBar: Container(
decoration: const BoxDecoration(
border: Border(
top: BorderSide(
width: 2, color: Colors.green), // Green border at the
top
),
boxShadow: [
BoxShadow(
color: Colors.black12,
blurRadius: 5,
offset: Offset(0, -2), // Slight elevation effect
),
],
),
child: BottomNavigationBar(
type: BottomNavigationBarType.fixed,
backgroundColor: Colors.white,
elevation: 10,
selectedItemColor: Colors.green, // Active icon color
unselectedItemColor: Colors.grey, // Inactive icon color
showUnselectedLabels: true,
showSelectedLabels: true,
currentIndex: _currentIndex,
onTap: (int index) {
    _pageController.jumpToPage(index);
},
items: const [
BottomNavigationBarItem(
icon: Icon(Icons.home),
label: 'Home',
),
BottomNavigationBarItem(

```

```
        icon: Icon(Icons.category),
        label: 'Category',
    ),
    BottomNavigationBarItem(
        icon: Icon(Icons.favorite),
        label: 'Favorite',
    ),
    BottomNavigationBarItem(
        icon: Icon(Icons.person),
        label: 'My Account',
    ),
],
),
),
);
}
}
```

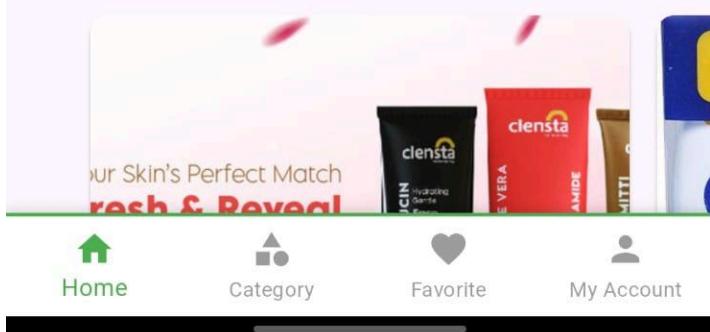
Output:



Dmart Exclusive



Best Offer



1:28

100%

Shop By Category



Dairy &
Beverages

Electronics &
Appliances

Grocery

Home &
Kitchen

Packaged
Foods

Sports &
Fitness



Juices



Soft Drinks



Milk & Dairy



Tea & Coffee



Home



Category



Favorite



My Account

1:29

100%

Soft Drinks



Thumbs Up

Qty: 2.25 L

DMart: ₹75

₹25 OFF

MRP: ₹100



1



Sprite

Qty: 2.25 L

DMart: ₹62

₹38 OFF

MRP: ₹100



ADD

1:29

100%

← My Cart



Thumbs Up

Qty: 2.25 L

DMart: ₹75.0

₹25 OFF

MRP: ₹100.0



- 1 +

Total Amount: ₹75.00

Clear Cart

Checkout

1:28 ▲ ▲

✉️ 🔍 🔁 🔍 🔁 100% 🔋

Favorites



Real Fruit Power Mixed Fruit

Qty: 1 L

DMart: ₹96.00

₹19.20 OFF

MRP: ₹115.20



B Natural Mixed Fruit Beverage

Qty: 1 L

DMart: ₹70.00

₹14.00 OFF

MRP: ₹84.00



Home



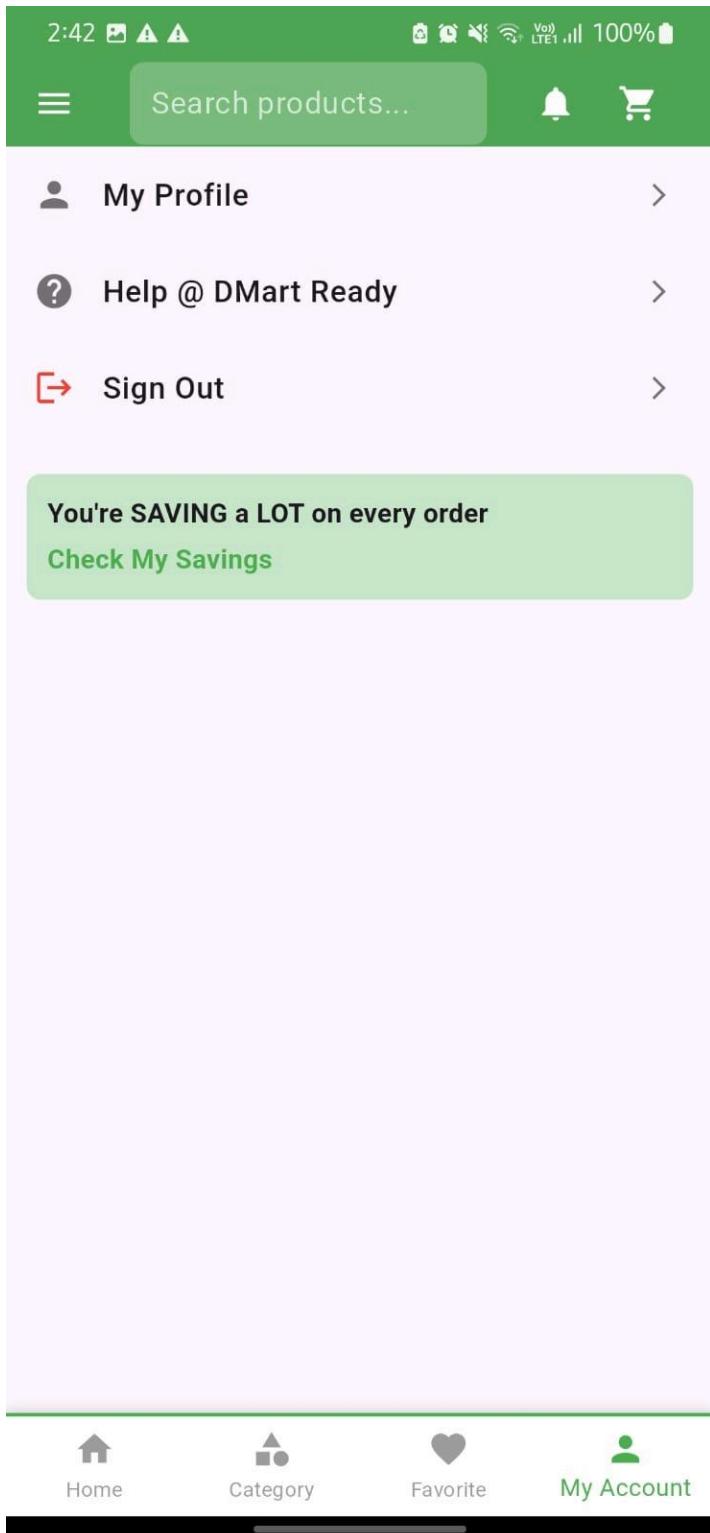
Category



Favorite



My Account



MPL Experiment 6

Name: Anuprita Mhapankar

Class: D15A

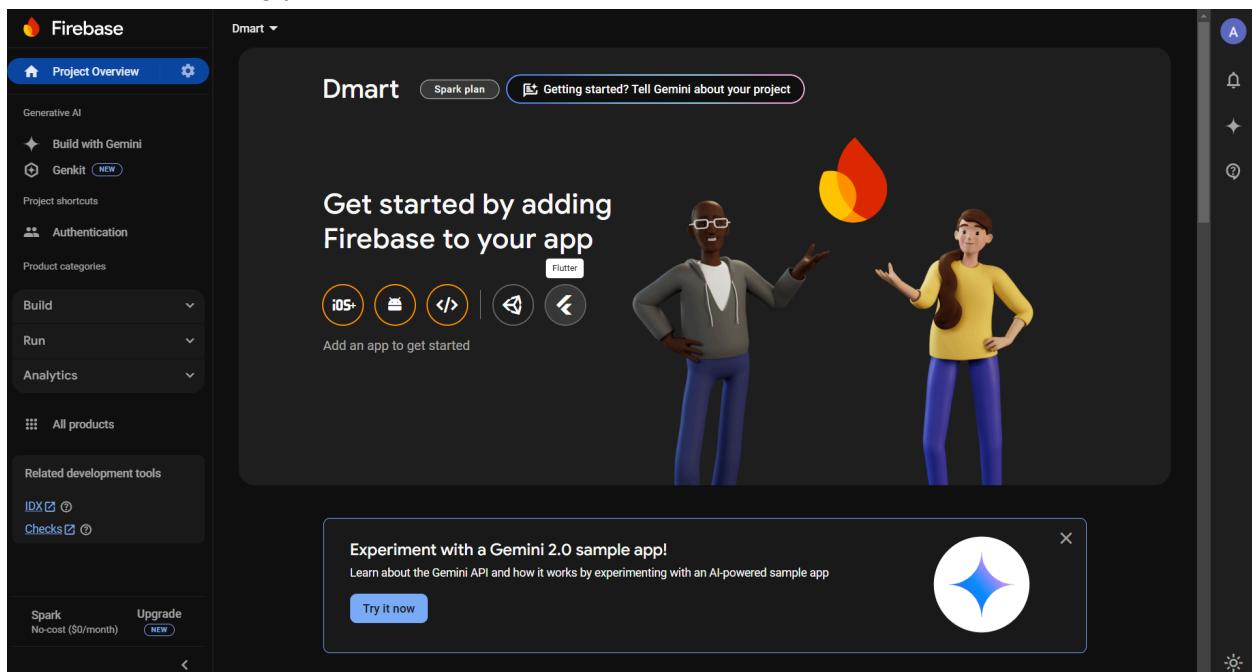
Roll no:28

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.firebaseio.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

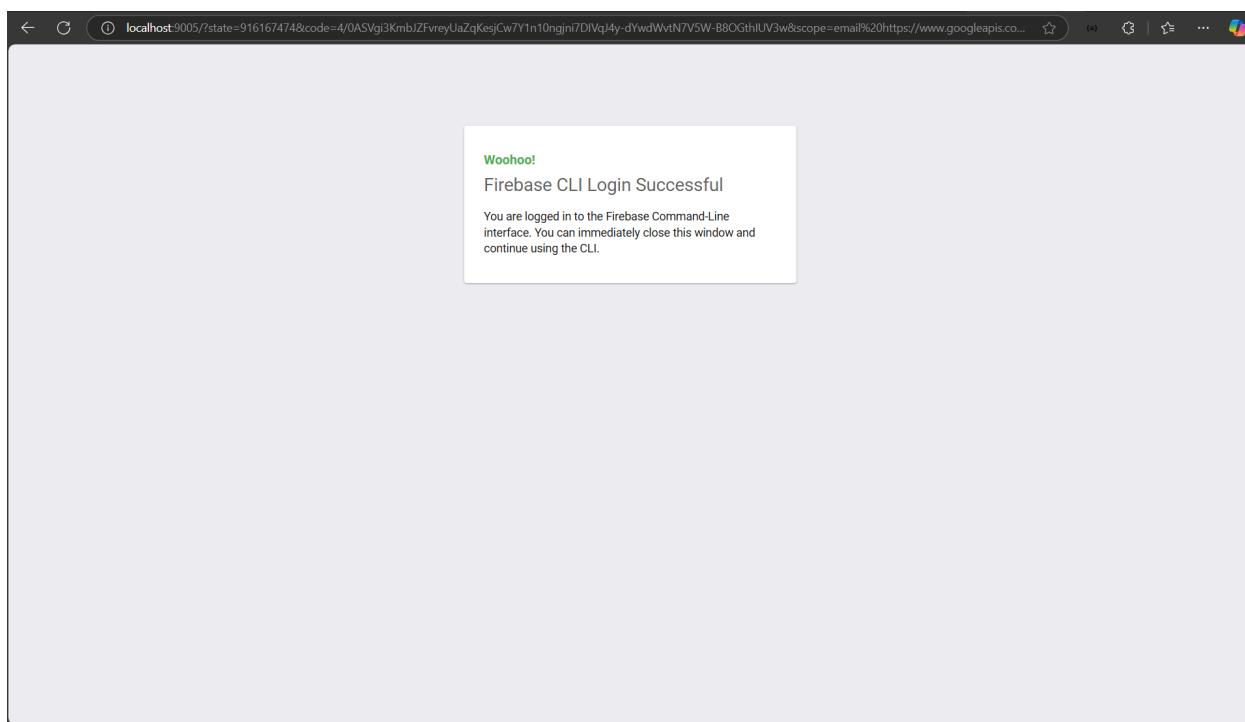
PS C:\Users\User> npm install -g firebase-tools
changed 635 packages in 33s

70 packages are looking for funding
  run `npm fund` for details
PS C:\Users\User> firebase --version
13.29.3
PS C:\Users\User> 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-fgrhgmdu7bqnekij5i8b5pr03ho849e6.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=916167474&redirect_uri=http%3A%2F%2Flocalhost%3A9005

Waiting for authentication...
+ Success! Logged in as 2022.anuprita.mhapankar@ves.ac.in
PS C:\Users\User>
```



Step 3:

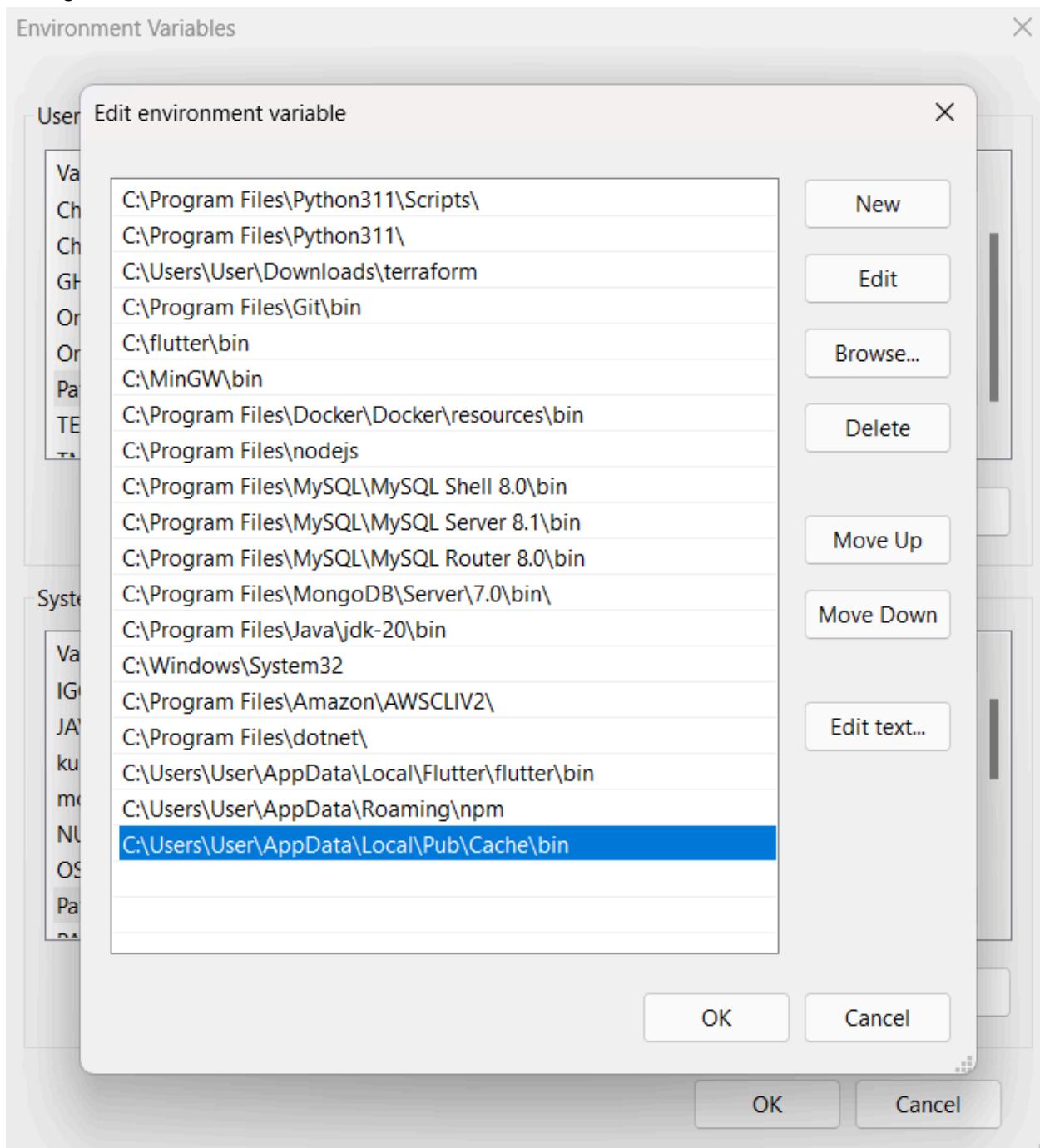
Open your Flutter app in **Android Studio**.

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.

Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

```
C:\Users\user\StudioProjects\dmart>dart pub global activate flutterfire_cli
The Dart CLI developer tool uses Google Analytics to report usage and diagnostic
data along with package dependencies, and crash reporting to send basic crash
reports. This data is used to help improve the Dart platform, Flutter framework,
and related tools.

Telemetry is not sent on the very first run. To disable reporting of telemetry,
run this terminal command:

  dart --disable-analytics

If you opt out of telemetry, an opt-out event will be sent, and then no further
information will be sent. This data is collected in accordance with the Google
Privacy Policy (https://policies.google.com/privacy).

Downloading packages... (1.4s)
+ ansi_styles 0.3.2+1
+ args 2.6.0
+ async 2.12.0
+ boolean_selector 2.1.2
+ characters 1.4.0
+ ci 0.1.0
+ cli_util 0.4.2
+ clock 1.1.2
+ collection 1.19.1
+ dart_console 1.2.0 (4.1.1 available)
+ deep_pink 1.1.0
+ ffi 2.1.3
+ file 7.0.1
```

18:37 CRLF UTF-8 2 spaces

Android SDK "Android API 29 Platform" is missing

```
# Read more about Android versioning at https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString while build-number is used as CFBundleVersion.
# Read more about iOS versioning at
# https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html#//apple\_ref/doc/uid/TP40000965-CH105-SW1
# In Windows, build-name is used as the major, minor, and patch parts
# of the product and file versions while build-number is used as the build suffix.
version: 1.0.0+1

environment:
  sdk: ^3.6.0

# Dependencies specify other packages that your package needs in order to work.
# To automatically upgrade your package dependencies to the latest versions
# consider running 'flutter pub upgrade --major-versions'. Alternatively,
```

Document 1/1 : name:

```
C:\Users\user\StudioProjects\dmart>flutterfire --version
1.0.1

C:\Users\user\StudioProjects\dmart>flutterfire configure --project=dmart-c9f2c
1 Found 1 Firebase projects. Selecting project dmart-c9f2c.
1 Which platforms should your configuration support (use arrow keys & space to select)? · android, ios, macos, web, windows
1 Firebase android app com.example.dmart is not registered on Firebase project dmart-c9f2c.
1 Registered a new Firebase android app on Firebase project dmart-c9f2c.
1 Firebase ios app com.example.dmart is not registered on Firebase project dmart-c9f2c.
1 Registered a new Firebase ios app on Firebase project dmart-c9f2c.
1 Firebase macos app com.example.dmart registered.
1 Firebase web app dmart (web) is not registered on Firebase project dmart-c9f2c.
1 Registered a new Firebase web app on Firebase project dmart-c9f2c.
1 Firebase windows app dmart (windows) is not registered on Firebase project dmart-c9f2c.
1 Registered a new Firebase windows app on Firebase project dmart-c9f2c.
```

1:1 CRLF UTF-8 2 spaces Schema: pubspec.json

```
dmart C:\Users\User\StudioProjects\dmart
  > dart_tool
  > idea
  > android [dmart_android]
  > build
  > lib
    > firebase_options.dart
    > main.dart
  > linux
  > macos
  > test
  > web
  > windows
  > ignore
  > metadata
  > analysis_options.yaml
  > dmart.iml
  > firebase.json

lib\firebase_options.dart
main.dart
pubspec.yaml

Android SDK "Android API 29 Platform" is missing
Flutter commands
Configure... | Pub get | Pub upgrade | Pub outdated | Flutter doctor
version: 1.0.0+1
environment:
  sdk: ^3.6.0
# Dependencies specify other packages that your package needs in order to work.
# To automatically upgrade your package dependencies to the latest versions

Document 1/1 > name:

Terminal Local x + 
1 Registered a new Firebase windows app on Firebase project dmart-c9f2c.

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

Platform  Firebase App Id
web      1:57333588780:web:5be92821a66b967dcbc289
android  1:57333588780:android:13d27f0178e184a8cbc289
ios       1:57333588780:ios:ed173068ff5a2a5cbc289
macos     1:57333588780:ios:ed173068ff5a2a5cbc289
windows   1:57333588780:web:60734d0072449c12cbc289

Learn more about using this file and next steps from the documentation:
> https://firebase.google.com/docs/flutter/setup

C:\Users\User\StudioProjects\dmart>
```

Step 5:

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

```
flutter pub add firebase_core
```

```
C:\Users\User\StudioProjects\dmart>flutter pub add firebase_core
Resolving dependencies...
Downloading packages...
async 2.11.8 (2.12.0 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 3.10.1
firebase_core_platform_interface 5.4.0
firebase_core_web 2.19.0
flutter_web_plugins 0.0.0 from sdk flutter
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)

C:\Users\User\StudioProjects\dmart> pubspec.yaml

dmart C:\Users\User\StudioProjects\dmart
  > dart_tool
  > idea
  > android [dmart_android]
  > build
  > lib
    > firebase_options.dart
    > main.dart
  > linux
  > macos
  > test
  > web
  > windows
  > ignore
  > metadata
  > analysis_options.yaml
  > dmart.iml
  > firebase.json

lib\firebase_options.dart
main.dart
pubspec.yaml

Android SDK "Android API 29 Platform" is missing
Flutter commands
Configure... | Pub get | Pub upgrade | Pub outdated | Flutter doctor
version: 1.0.0+1
environment:
  sdk: ^3.6.0
# Dependencies specify other packages that your package needs in order to work.
# To automatically upgrade your package dependencies to the latest versions

Document 1/1 > name:

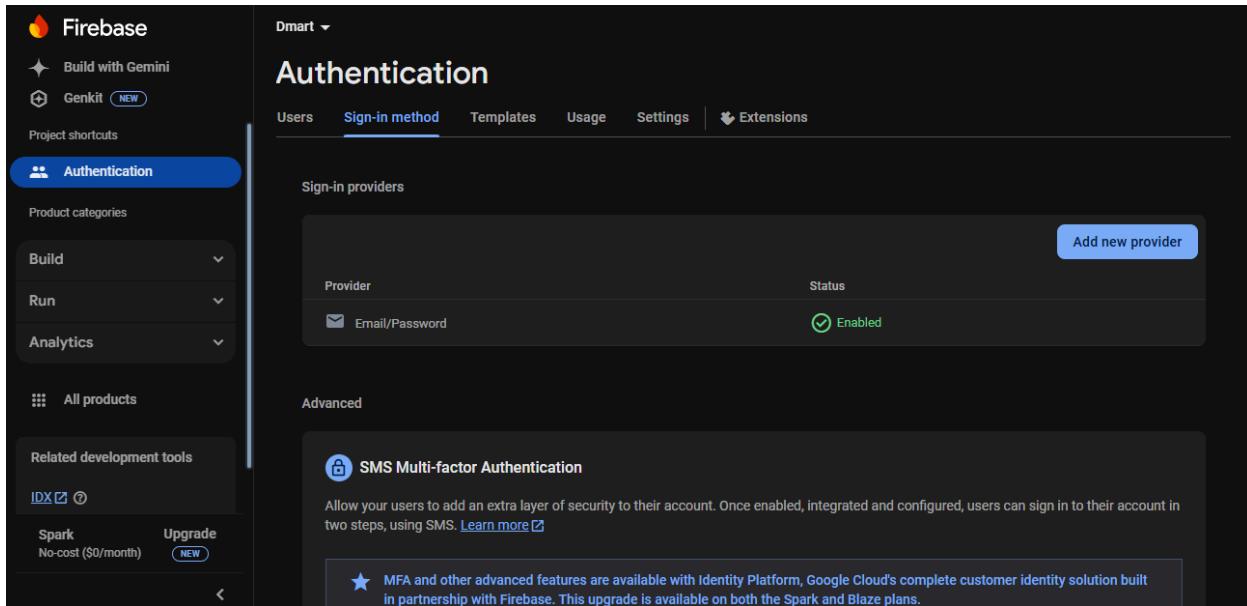
Terminal Local x + 
C:\Users\User\StudioProjects\dmart> flutter pub add firebase_core
Resolving dependencies...
Downloading packages...
async 2.11.8 (2.12.0 available)
boolean_selector 2.1.1 (2.1.2 available)
characters 1.3.0 (1.4.0 available)
clock 1.1.1 (1.1.2 available)
collection 1.19.0 (1.19.1 available)
fake_async 1.3.1 (1.3.3 available)
firebase_core 3.10.1
firebase_core_platform_interface 5.4.0
firebase_core_web 2.19.0
flutter_web_plugins 0.0.0 from sdk flutter
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)

C:\Users\User\StudioProjects\dmart>
```

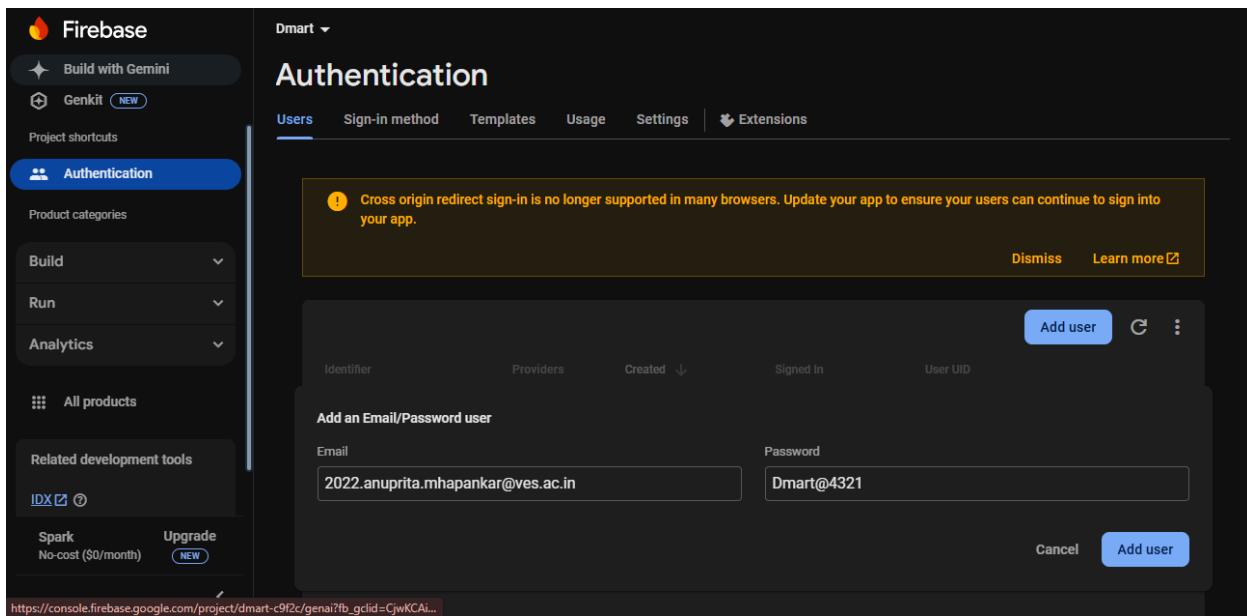
Firebase Authentication SetUp

Step 1:

Go to the Firebase Console (<https://console.firebaseio.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 interface. On the left sidebar, 'Authentication' is selected. The main 'Sign-in method' tab is active. Under 'Sign-in providers', the 'Email/Password' provider is listed with a status of 'Enabled'. Below this, there's a section for 'SMS Multi-factor Authentication' with a note about enabling two-step verification. A dark banner at the bottom provides information about MFA and Identity Platform.



The screenshot shows the Firebase Authentication 'Users' page. A yellow warning banner at the top informs users that cross-origin redirect sign-in is no longer supported and suggests updating their app. Below the banner, a modal window is open for adding a new user. The 'Add an Email/Password user' form has 'Email' set to '2022.anuprita.mhapankar@ves.ac.in' and 'Password' set to 'Dmart@4321'. Buttons for 'Cancel' and 'Add user' are visible at the bottom right of the modal.

The screenshot shows the Firebase Authentication console. On the left, there's a sidebar with navigation links like 'Build with Gemini', 'Genkit', 'Project shortcuts', 'Authentication' (which is selected and highlighted in blue), 'Analytics', 'All products', and 'Related development tools'. The main area is titled 'Authentication' and has tabs for 'Users', 'Sign-in method', 'Templates', 'Usage', 'Settings', and 'Extensions'. A prominent yellow warning box at the top says: 'Cross origin redirect sign-in is no longer supported in many browsers. Update your app to ensure your users can continue to sign into your app.' Below this, there's a table with columns 'Identifier', 'Providers', 'Created', 'Signed In', and 'User UID'. One user is listed: '2022 anuprita.mhapank...' with an email icon, 'Jan 29, 2025' under 'Created', and 'QwwusXHJFpcS1BcTzbJQWE...' under 'User UID'. At the bottom of the table, there are buttons for 'Rows per page' (set to 50), '1 - 1 of 1', and navigation arrows.

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
```

The screenshot shows the Android Studio interface. On the left, the project structure shows a folder named 'dmart' containing various files and folders like 'dart_tool', 'idea', 'android', 'ios', 'lib', 'test', 'web', 'windows', 'flutter-plugins', 'flutter-plugins-dependencies', 'ignore', 'metadata', 'analysis_options.yaml', 'dmart.lint', 'firebase.json', 'pubspec.lock', and 'pubspec.yaml'. The 'pubspec.yaml' file is open in the code editor. The code in the editor is as follows:

```

 flutter:
   sdk: flutter

   # The following adds the Cupertino Icons font to your application.
   # Use with the CupertinoIcons class for iOS style icons.
  /cupertino_icons: ^1.0.8
  /firebase_core: ^3.10.1
  /firebase_auth: ^5.4.1

 dev_dependencies:
   flutter_test:
     sdk: flutter

   # The "flutter_lints" package below contains a set of recommended lints to
   # encourage good coding practices. The lint set provided by the package is
   # activated in the "analysis_options.yaml" file located at the root of your
   # package. See that file for information about deactivating specific lint
   # rules and activating additional ones.

```

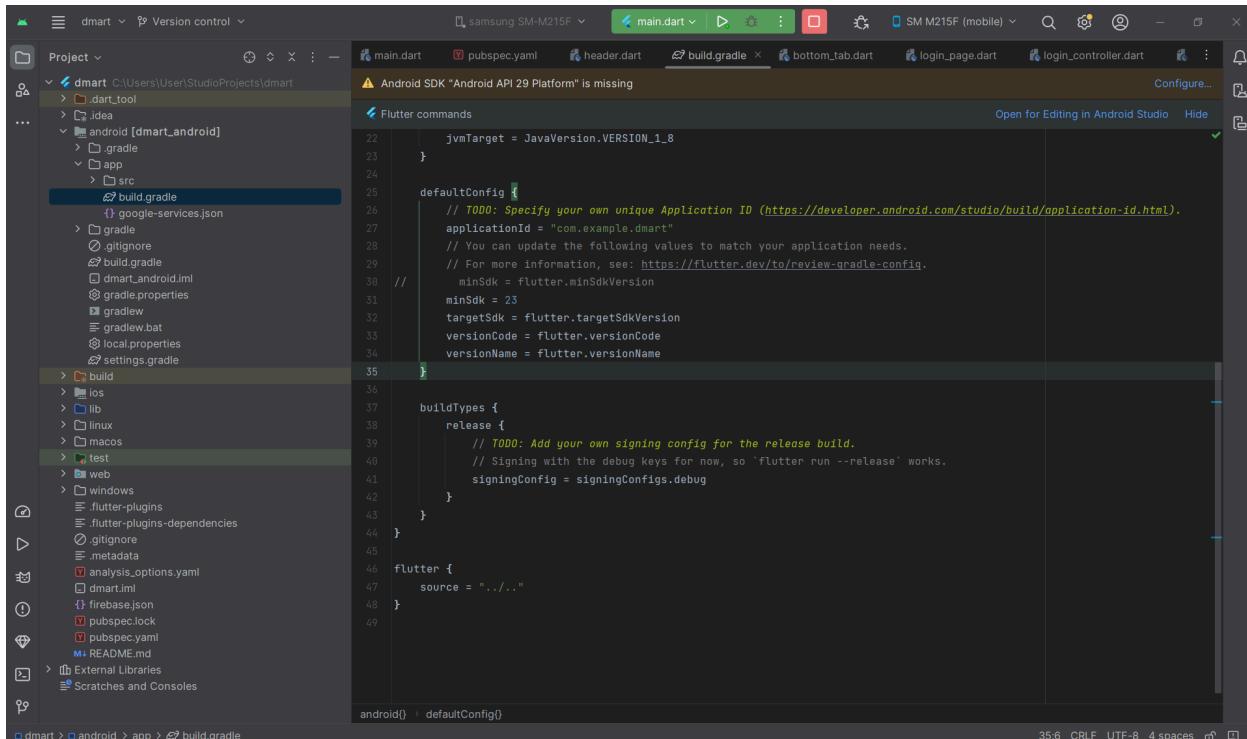
At the bottom of the terminal window, the command 'flutter pub add firebase_auth' is shown being run, followed by the output of the dependency resolution process.

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.

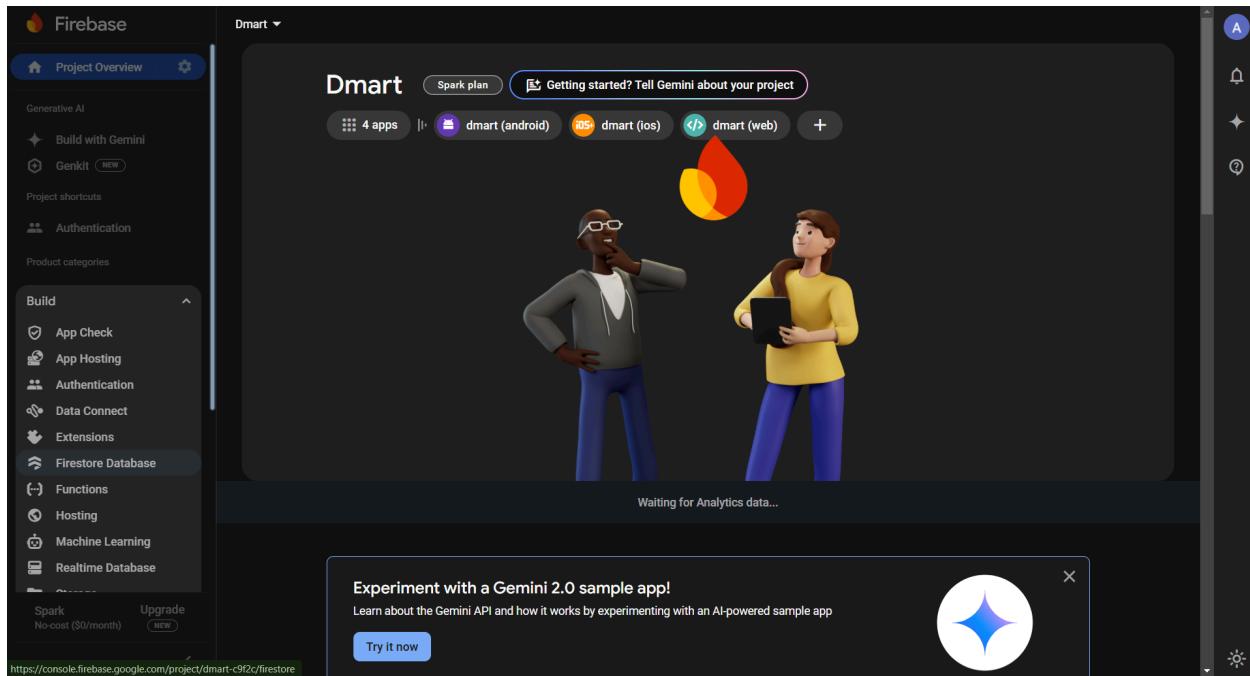


Now, firebase is successfully connected to our app.

Firestore Database Setup

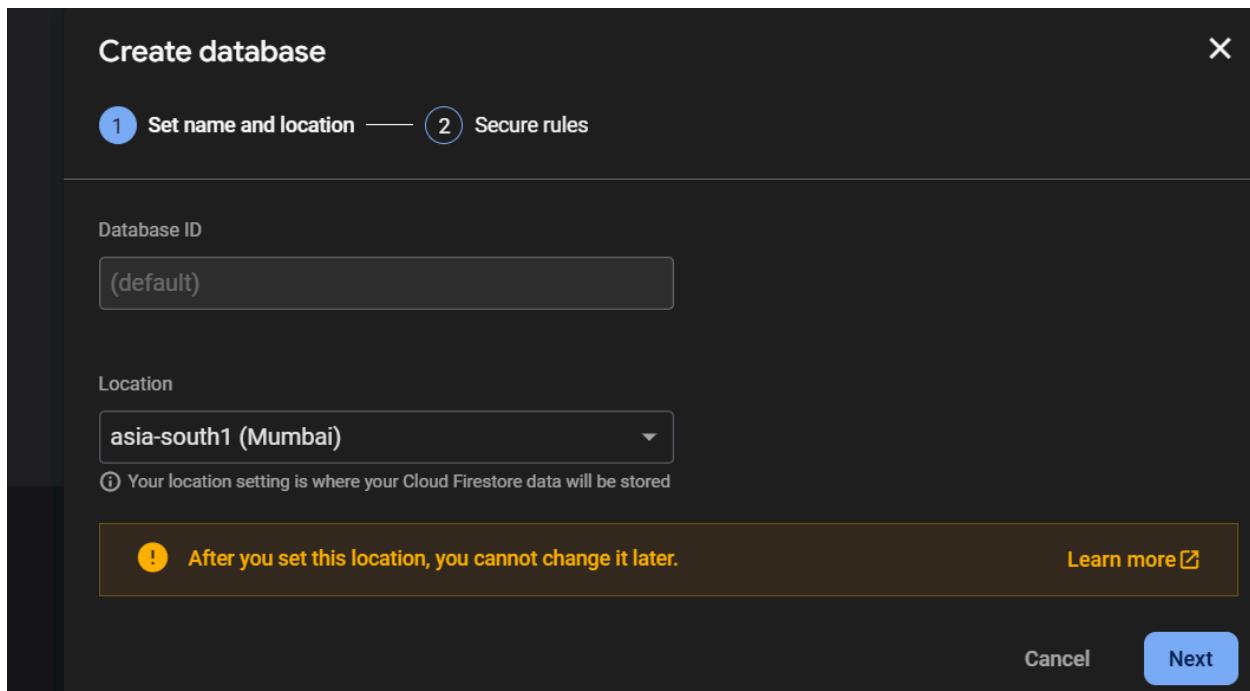
Step 1:

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

X

 Set name and location —  Secure rules

After you define 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, 3, 10);
        }
    }
}
```

 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.

Start a collection

- 1 Give the collection an ID
- 2 Add its first document

Parent path

/

Collection ID ?

categories

Cancel

Next

Start a collection

1 Give the collection an ID ————— 2 Add its first document

Document parent path

/categories

Document ID [?](#)

grocery

| Field | Type | Value |
|-------|----------|---------|
| name | = string | Grocery |

| Field | Type |
|---------------|-------|
| subcategories | = map |

| Field | Type |
|-------|-------|
| dals | = map |

| Field | Type | Value |
|-------|----------|-------|
| name | = string | Dals |

| Field | Type | Value |
|-------|----------|---|
| image | = string | https://cdn.dmar |

[+ Add field](#)

| Field | Type |
|-------|------|
|-------|------|

The screenshot shows the Firebase Cloud Firestore interface. On the left, the navigation bar includes 'Project Overview', 'Generative AI', 'Build with Gemini', 'Genkit (NEW)', 'Authentication', and 'Firestore Database' (which is highlighted). Below these are sections for 'Build', 'Run', 'Analytics', 'All products', and 'Related development tools'. At the bottom of the sidebar are 'Spark' (No-cost (\$0/month)) and 'Upgrade (NEW)'. The main area displays a hierarchical database structure under 'categories': 'categories' (with 'dairy_beverages', 'electronics', 'grocery', 'home_kitchen', 'packaged_foods', and 'sports' as children), 'sports' (with 'name: "Sports & Fitness"', 'subcategories' (with 'apparels' and 'exercise' as children), and 'apparels' (with 'image: "https://cdn.dmart.in/images/categories/clothing-accessories-131022.svg"' and 'name: "Apparels & Accessories"'). Other children of 'sports' include 'exercise' (with 'image: "https://cdn.dmart.in/images/products/SEP280000013xx1"' and 'name: "Exercise & Fitness"') and 'indoorSports' (with 'image: "https://cdn.dmart.in/images/categories/aesc-sports-fitnessc2.svg"' and 'name: "Indoor Sports"'). A message at the top right says 'Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing' with a 'Configure App Check' link. The bottom of the screen shows '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:

```
//category_page.dart
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:dmart/screens/category/productList_page.dart';
import 'package:provider/provider.dart';
import 'package:dmart/providers/cart_provider.dart';
import 'package:dmart/screens/cart/cart_page.dart';

class CategoryPage extends StatefulWidget {
  const CategoryPage({super.key});

  @override
  State<CategoryPage> createState() => _CategoryPageState();
}

class _CategoryPageState extends State<CategoryPage> {
  List<Map<String, dynamic>> categories = [];
  String selectedCategoryId = '';

  @override
  void initState() {
    super.initState();
    testFirestoreConnection();
    fetchCategories();
  }
}
```

```
void testFirestoreConnection() async {
  try {
    // Fetch the Firestore collection
    var snapshot =
      await FirebaseFirestore.instance.collection('categories').get();

    // Print each document's data to the console
    for (var doc in snapshot.docs) {
      print("Category ID: ${doc.id}, Data: ${doc.data()}");
    }

    // Optionally, log the total count of documents
    print("Total Categories: ${snapshot.docs.length}");
  } catch (e) {
    print("Error fetching categories: $e"); // Error handling
  }
}

Future<void> fetchCategories() async {
  try {
    var snapshot =
      await FirebaseFirestore.instance.collection('categories').get();
    setState(() {
      categories = snapshot.docs.map((doc) {
        Map<String, dynamic> data = doc.data();
        return {
          'id': doc.id,
          'name': data['name'] ?? 'Unknown Category',
          'subcategories':
            Map<String, dynamic>.from(data['subcategories'] ?? {});
        };
      }).toList();
    });

    if (categories.isNotEmpty) {
      selectedCategoryId = categories[0]['id'];
    }
  });
} catch (e) {
  print("Error fetching categories: $e");
}
}

@Override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: Colors.green,
```

```
title: const Text(
    'Shop By Category',
    style: TextStyle(
        color: Colors.white, // White text color
        fontWeight: FontWeight.w600, // Semi-bold font weight
    ),
),
actions: [
    IconButton(
        icon: const Icon(Icons.search, color: Colors.white),
        onPressed: () {
            // Implement search functionality
        },
    ),
    Padding(
        padding: const EdgeInsets.only(right: 20), // Add right padding here
        child: Consumer<CartProvider>(
            builder: (_, cart, ch) => cart.itemCount > 0
                ? Badge(
                    label: Text(cart.itemCount.toString()),
                    child: IconButton(
                        icon: const Icon(Icons.shopping_cart,
                            color: Colors.white),
                        onPressed: () {
                            Navigator.push(
                                context,
                                MaterialPageRoute(
                                    builder: (context) => const CartPage()),
                            );
                        },
                    ),
                )
                : IconButton(
                    icon:
                        const Icon(Icons.shopping_cart, color: Colors.white),
                    onPressed: () {
                        Navigator.push(
                            context,
                            MaterialPageRoute(
                                builder: (context) => const CartPage()),
                            );
                    },
                ),
            ],
),
)
```

```
body: Row(
  children: [
    // Left side - Category List
    Container(
      width: 150,
      color: Colors.white,
      child: ListView.builder(
        itemCount: categories.length,
        itemBuilder: (context, index) {
          final category = categories[index];
          final isSelected = category['id'] == selectedCategoryId;
          return Container(
            color: isSelected ? Colors.lightGreen.withOpacity(0.2) : null,
            child: ListTile(
              title: Text(
                category['name'],
                style: TextStyle(
                  fontSize: 14,
                  color: isSelected ? Colors.green : Colors.black,
                ),
              ),
              onTap: () {
                setState(() {
                  selectedCategoryId = category['id'];
                });
              },
            ),
          );
        },
      ),
    ),
    // Vertical Divider
    Container(
      width: 1,
      color: Colors.grey[300],
    ),
    // Right side - Subcategories Grid
    Expanded(
      child: categories.isEmpty
        ? const Center(child: CircularProgressIndicator())
        : buildSubcategoriesGrid(),
    ),
  ],
),
);
```

```

Widget buildSubcategoriesGrid() {
    final selectedCategory = categories.firstWhere(
        (category) => category['id'] == selectedCategoryId,
       orElse: () => categories[0],
    );

    final subcategories =
        selectedCategory['subcategories'] as Map<String, dynamic>;

    return GridView.builder(
        padding: const EdgeInsets.all(16),
        gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 2, // Keeps two tiles per row
            crossAxisSpacing: 16,
            mainAxisSpacing: 16,
            childAspectRatio: 0.75, // Decreased to make tiles bigger
        ),
        itemCount: subcategories.length,
        itemBuilder: (context, index) {
            String subcategoryId = subcategories.keys.elementAt(index);
            Map<String, dynamic> subcategoryData = subcategories[subcategoryId];

            return GestureDetector(
                onTap: () {
                    Navigator.push(
                        context,
                        MaterialPageRoute(
                            builder: (context) => ProductListPage(
                                categoryId: selectedCategoryId,
                                subcategoryId: subcategoryId,
                                subcategoryName: subcategoryData['name'] ?? 'Unknown',
                            ),
                        ),
                    );
                },
                child: buildSubcategoryTile(
                    subcategoryData['name'] ?? 'Unknown',
                    subcategoryData['image'] ?? '',
                ),
            );
        },
    );
}

Widget buildSubcategoryTile(String title, String imageUrl) {
    return SizedBox(
        width: 160,

```

```
height: 180, // Adjusted height to prevent overflow
child: Card(
  color: Colors.white,
  elevation: 5,
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(12),
  ),
  shadowColor: Colors.black38,
  child: Padding(
    padding: const EdgeInsets.all(8.0),
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      mainAxisSize: MainAxisSize.min, // Ensures content doesn't expand unnecessarily
      children: [
        SizedBox(
          width: 80,
          height: 80,
          child: imageUrl.isNotEmpty
            ? Image.network(
                imageUrl,
                width: 80,
                height: 80,
                fit: BoxFit.cover,
                errorBuilder: (context, error, stackTrace) {
                  return const Icon(Icons.image_not_supported,
                    size: 80);
                },
            )
            : const Icon(Icons.image_not_supported, size: 80),
        ),
        Expanded(
          // This ensures text doesn't push content beyond limits
          child: Text(
            title,
            style: const TextStyle(
              fontWeight: FontWeight.bold,
              fontSize: 12,
            ),
            textAlign: TextAlign.center,
            maxLines: 2,
            overflow: TextOverflow.ellipsis,
          ),
        ),
      ],
    ),
  ),
),
```

) ,
);
}
}

```
//productList_page.dart
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:dmart/providers/cart_provider.dart';
import 'package:dmart/providers/favorite_provider.dart';
import 'package:dmart/screens/cart/cart_page.dart';

class ProductListPage extends StatefulWidget {
    final String categoryId;
    final String subcategoryId;
    final String subcategoryName;

    const ProductListPage({
        super.key,
        required this.categoryId,
        required this.subcategoryId,
        required this.subcategoryName,
    });

    @override
    State<ProductListPage> createState() => _ProductListPageState();
}

class _ProductListPageState extends State<ProductListPage> {
    List<Map<String, dynamic>> products = [];
    // Map<String, int> productQuantities = {};
    // Map<String, bool> wishlist = {};
    bool isLoading = true;

    @override
    void initState() {
        super.initState();
        fetchProducts();
    }

    Future<void> fetchProducts() async {
        try {
            var doc = await FirebaseFirestore.instance
                .collection('categories')
                .doc(widget.categoryId)
                .get();
            if (!doc.exists) {
                setState(() => isLoading = false);
                return;
            }
            var data = doc.data();
        }
    }
}
```

```

    if (data == null) {
        setState(() => isLoading = false);
        return;
    }
    var subcategories = data['subcategories'] as Map<String, dynamic>?;
    var subcategory =
        subcategories?[widget.subcategoryId] as Map<String, dynamic>?;
    var productsList = subcategory?['products'] as List<dynamic>?;
    if (productsList != null) {
        setState(() {
            products = productsList.map((product) {
                return {
                    'id': product['id'] ?? '',
                    'name': product['name'] ?? 'Unknown Product',
                    'imageUrl': product['imageUrl'] ?? '',
                    'dmartPrice': product['dmartPrice']?.toString() ?? '0',
                    'mrpPrice': product['mrpPrice']?.toString() ?? '0',
                    'quantity': product['quantity'] ?? '',
                    'isOutOfStock': product['isOutOfStock'] ?? false,
                };
            }).toList();
            isLoading = false;
        });
    } else {
        setState(() => isLoading = false);
    }
} catch (e) {
    print("Error fetching products: $e");
    setState(() => isLoading = false);
}
}

void incrementQuantity(String productId) {
    final cartProvider = Provider.of<CartProvider>(context, listen: false);
    final product = products.firstWhere((p) => p['id'] == productId);

    cartProvider.addItem(
        productId: productId,
        name: product['name'],
        dmartPrice: double.parse(product['dmartPrice']),
        mrpPrice: double.parse(product['mrpPrice']),
        imageUrl: product['imageUrl'],
        quantity: product['quantity'],
    );
}

void decrementQuantity(String productId) {
    final cartProvider = Provider.of<CartProvider>(context, listen: false);
    cartProvider.removeItem(productId);
}

```



```

                builder: (context) => const CartPage()),
            );
        },
    ),
),
),
],
),
body: isLoading
? const Center(child: CircularProgressIndicator())
: products.isEmpty
? const Center(
    child: Text(
        'No products available',
        style: TextStyle(fontSize: 16),
    ),
)
: ListView.builder(
    padding: const EdgeInsets.all(8),
    itemCount: products.length,
    itemBuilder: (context, index) {
        final product = products[index];
        return buildProductTile(product);
    },
),
),
);
}
}

Widget buildProductTile(Map<String, dynamic> product) {
String productId = product['id'];

print("productId ID: ${productId}");
print("productcheck: ${product}");

return Consumer2<CartProvider, FavoriteProvider>(
builder: (context, cart, favorite, child) {
int quantity = cart.getItemQuantity(productId);
bool isWishlisted = favorite isInFavorites(productId);

return Card(
margin: const EdgeInsets.symmetric(horizontal: 10, vertical: 5),
elevation: 4,
shadowColor: Colors.black26,
color: Colors.white,
shape:
RoundedRectangleBorder(borderRadius: BorderRadius.circular(10)),
child: Padding(

```

```
padding: const EdgeInsets.all(10.0),
child: Row(
  mainAxisAlignment: MainAxisAlignment.start,
  children: [
    ClipRRect(
      borderRadius: BorderRadius.circular(8),
      child: Image.network(
        product['imageUrl'] ?? '',
        width: 100,
        height: 120,
        fit: BoxFit.cover,
        errorBuilder: (context, error, stackTrace) =>
          const Icon(Icons.image_not_supported, size: 100),
      ),
    ),
    const SizedBox(width: 10),
    Expanded(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: [
          Text(
            product['name'] ?? 'Unknown Product',
            style: const TextStyle(
              fontWeight: FontWeight.bold,
              fontSize: 16,
            ),
            maxLines: 2,
            overflow: TextOverflow.ellipsis,
          ),
          const SizedBox(height: 5),
          Align(
            alignment: Alignment.centerLeft,
            child: Container(
              width: double.infinity,
              padding: const EdgeInsets.symmetric(
                horizontal: 6, vertical: 4),
              decoration: BoxDecoration(
                border: Border.all(color: Colors.grey, width: 1),
                borderRadius: BorderRadius.circular(5),
              ),
              child: Text(
                "Qty: ${product['quantity']}",
                style: const TextStyle(
                  fontSize: 12, fontWeight: FontWeight.w500),
              ),
            ),
          ),
        ],
      ),
    ),
  ],
),
```

```
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  crossAxisAlignment: CrossAxisAlignment.center,
  children: [
    Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(
          "DMart: ₹${product['dmartPrice']}",
          style: const TextStyle(
            fontSize: 14,
            color: Colors.green,
            fontWeight: FontWeight.bold,
          ),
        ),
        const SizedBox(height: 2),
        Text(
          "MRP: ₹${product['mrpPrice']}",
          style: const TextStyle(
            fontSize: 14,
            decoration: TextDecoration.lineThrough,
            color: Colors.grey,
          ),
        ),
        ],
      ),
      const SizedBox(width: 10),
      Expanded(
        child: Align(
          alignment: Alignment.centerRight,
          child: Container(
            padding: const EdgeInsets.symmetric(
              horizontal: 8, vertical: 4),
            decoration: BoxDecoration(
              color: Colors.amber.shade100,
              borderRadius: BorderRadius.circular(10),
            ),
            child: Text(
              "₹${(int.parse(product['mrpPrice']) -
int.parse(product['dmartPrice']))} OFF",
              style: const TextStyle(
                fontSize: 12,
                fontWeight: FontWeight.bold,
                color: Colors.brown,
              ),
            ),
          ),
        ),
      ),
    ),
  ],
);
```

```
        ),
        ),
    ],
),
const SizedBox(height: 10),
Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
        Container(
            decoration: BoxDecoration(
                color: Colors.white,
                border:
                    Border.all(color: Colors.white70, width: 2),
                borderRadius: BorderRadius.circular(5),
                boxShadow: [
                    BoxShadow(
                        color: Colors.grey.withOpacity(0.5),
                        blurRadius: 4,
                        offset: const Offset(2, 2),
                    ),
                ],
            ),
            child: IconButton(
                onPressed: () {
                    // Toggle favorite status
                    favorite.toggleFavorite(
                        productId: productId,
                        name: product['name'],
                        dmartPrice:
                            double.parse(product['dmartPrice']),
                        imageUrl: product['imageUrl'],
                        quantity: product['quantity'],
                    );
                },
                icon: Icon(
                    favorite isInFavorites(productId)
                        ? Icons.favorite
                        : Icons.favorite_border,
                    color: Colors.green,
                ),
            ),
        ),
        ),
    ],
),
product['isOutOfStock']
    ? const Text(
        "Out Of Stock",
        style: TextStyle(
            color: Colors.red,
```

```
        fontWeight: FontWeight.bold,
    ),
)
: quantity == 0
? SizedBox(
    width: 120,
    child: ElevatedButton(
        onPressed: () {
            cart.addItem(
                productId: productId,
                name: product['name'],
                dmartPrice: double.parse(
                    product['dmartPrice']),
                mrpPrice: double.parse(
                    product['mrpPrice']),
                imageUrl: product['imageUrl'],
                quantity: product['quantity'],
            );
        },
        style: ElevatedButton.styleFrom(
            backgroundColor: Colors.green,
            shape: RoundedRectangleBorder(
                borderRadius:
                    BorderRadius.circular(5),
            ),
            padding: const EdgeInsets.symmetric(
                vertical: 12),
        ),
        child: const Row(
            mainAxisAlignment:
                MainAxisAlignment.center,
            children: [
                Icon(Icons.shopping_cart,
                    size: 18, color: Colors.white),
                SizedBox(width: 5),
                Text(
                    'ADD',
                    style: TextStyle(
                        color: Colors.white,
                        fontWeight: FontWeight.w500,
                    ),
                ),
            ],
        ),
    ),
)
: Container(
```

```
width: 120,
decoration: BoxDecoration(
    border: Border.all(
        color: Colors.green, width: 2),
    borderRadius: BorderRadius.circular(5),
),
child: Row(
    children: [
        Expanded(
            child: SizedBox(
                height: 40,
                child: ElevatedButton(
                    onPressed: () =>
                        cart.removeItem(productId),
                    style: ElevatedButton.styleFrom(
                        backgroundColor: Colors.green,
                        alignment: Alignment.center,
                        minimumSize:
                            const Size(0, 40),
                        padding: EdgeInsets.zero,
                        shape:
                            const RoundedRectangleBorder(
                                borderRadius:
                                    BorderRadius.only(
                                        topLeft:
                                            Radius.circular(3),
                                        bottomLeft:
                                            Radius.circular(3),
                                    ),
                                ),
                    ),
                    child: const Icon(Icons.remove,
                        color: Colors.white,
                        size: 24),
                ),
            ),
        ),
    ],
),
Container(
    alignment: Alignment.center,
    width: 40,
    child: Text(
        quantity.toString(),
        style: const TextStyle(
            fontSize: 16,
            fontWeight: FontWeight.bold,
        ),
    ),
),
```



```
) ,  
),  
);  
},  
);  
}  
}
```

Output:

Login Page:

12:00



← Let's Get You Logged In

Enter your Email

Enter your Password

LOGIN

Category Page:

1:28 100%

Shop By Category

| Category | Icon | Sub-Categories |
|--------------------------|------|----------------|
| Dairy & Beverages | | |
| Electronics & Appliances | | Juices |
| Grocery | | Soft Drinks |
| Home & Kitchen | | Milk & Dairy |
| Packaged Foods | | Tea & Coffee |
| Sports & Fitness | | |

Home Category Favorite My Account

Product List Page:

1:29 100%

Soft Drinks

Thumbs Up



Qty: 2.25 L

DMart: ₹75

MRP: ₹100

- 1 +

Sprite



Qty: 2.25 L

DMart: ₹62

MRP: ₹100

ADD

Cart Page:

1:29 100%

My Cart

Thumbs Up

Qty: 2.25 L

DMart: ₹75.0

MRP: ₹100.0 **₹25 OFF**

- **1** +

Total Amount: ₹75.00

Clear Cart

Checkout

Favorite Page:

1:28 ▲ ▲ 📦 🔍 ⚡ LTE 100%

Favorites

Real Fruit Power Mixed Fruit

Qty: 1 L

DMart: ₹96.00 ₹19.20 OFF

MRP: ₹115.20



B Natural Mixed Fruit Beverage

Qty: 1 L

DMart: ₹70.00 ₹14.00 OFF

MRP: ₹84.00



 Home  Category  Favorite  My Account