

MPL Prerequisites Document

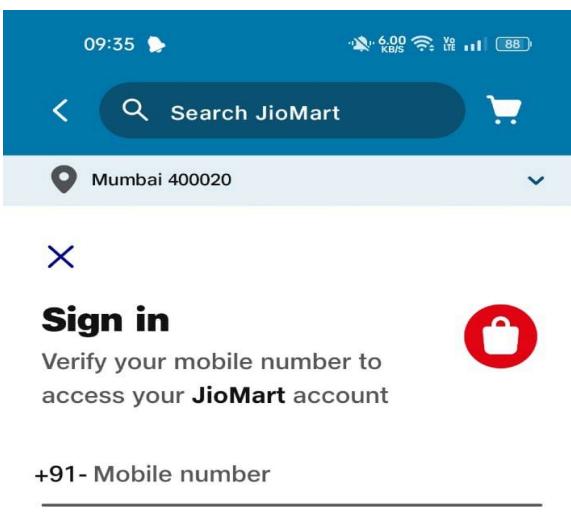
Name: Sneha Patra

Class: D15A

Roll no:40

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

Sr No	Screen Shot	Features
1)	 The screenshot shows the JioMart app's landing page. At the top, there is a header bar with the time (09:35), battery level (6.00), signal strength, and a search bar labeled "Search JioMart". Below the header, the location is set to "Mumbai 400020". The main content area has a blue header with the text "Sign in" and a red circular icon containing a shopping bag. Below this, it says "Verify your mobile number to access your JioMart account" and provides a text input field for "+91 - Mobile number". A large blue "Continue" button is at the bottom. At the very bottom of the screen, there is a footer with four icons: Home (house), Categories (grid), Wishlist (heart), and Orders (cube).	<p>1. Landing Page</p> <p>Firebase Authentication:</p> <p>Secure sign-in using email address.</p> <p>User details include:</p> <p>Username: Captures the user's display name.</p> <p>Email Address: For login credentials and communication.</p> <p>Password: Encrypted for security.</p> <p>Easy-to-use form for new user registration and existing user login.</p>

2)



2. User Homepage

Features:

Icons:

Notifications for updates and offers.

Profile management and user settings.

Cart for quick access to added items.

Search Bar:

Auto-suggestions for faster search results.

Best Offers Section:

Display curated deals and personalized offers

3)



3. Shopping List

Browse Categories:

Shop for groceries, electronics, and more, with dynamic updates based on trends.

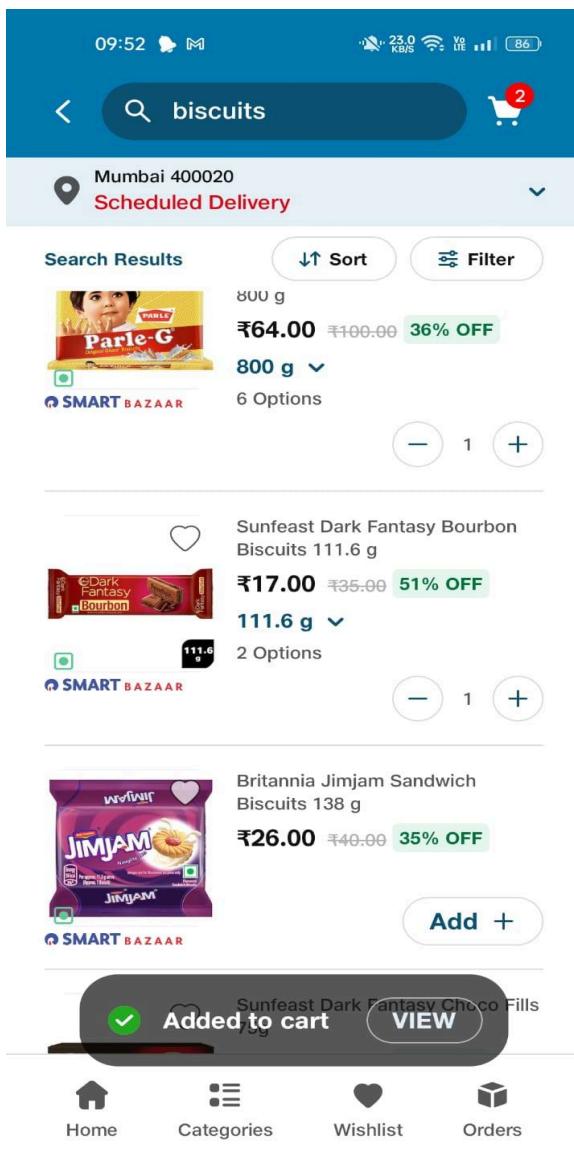
Personalized Shopping Lists:

Create and manage shopping lists for different occasions (e.g., monthly groceries).

List Management:

Edit quantities, remove items, and get notifications for stock updates or price drops.

4)



4. Search Results:-

Features:

Detailed product information displayed in search results.

Direct options to:

Add to Cart: Select items for purchase.

Remove from Cart: Update the cart instantly.

Filters for price, ratings, and categories to refine search results.

5)

09:37 56.0 KB/S Vo LTE 88

< My Cart

Mumbai 400020
Hyperlocal Delivery in 10 to 30 min

Hyperlocal Basket (3) ₹1,187.00

Yay! You get FREE delivery with this Basket

Tata Gold Leaf Tea 500 g
₹165.00 ₹330.00
You Save ₹165.00
Sold by: Reliance Retail
Size: 500 g

Save for Later - 1 +

Pears Pure & Gentle Soap with Natural Oils 125 g (Buy 4 Get 1 Free)
₹223.00 ₹372.00
You Save ₹149.00
Sold by: Reliance Retail

Save for Later - 1 +

Saffola Gold Rice Bran Based Blended Oil 3 L + India Gate Rozana Basmati Rice 5 kg
₹700.00

₹1,187.00 ✓
You Saved ₹771.00

Place Order

5. My Cart

Comprehensive view of all items added to the cart.

Features:

Modify the cart by removing unwanted items.

View product quantity and total price breakdown.

Proceed to place an order with a streamlined checkout process.

Integrated payment options for a secure and hassle-free transaction.

MPL Experiment 1

Name: Sneha Patra

Class: D15A

Roll no:40

Aim: Installation and Configuration of Flutter Environment.

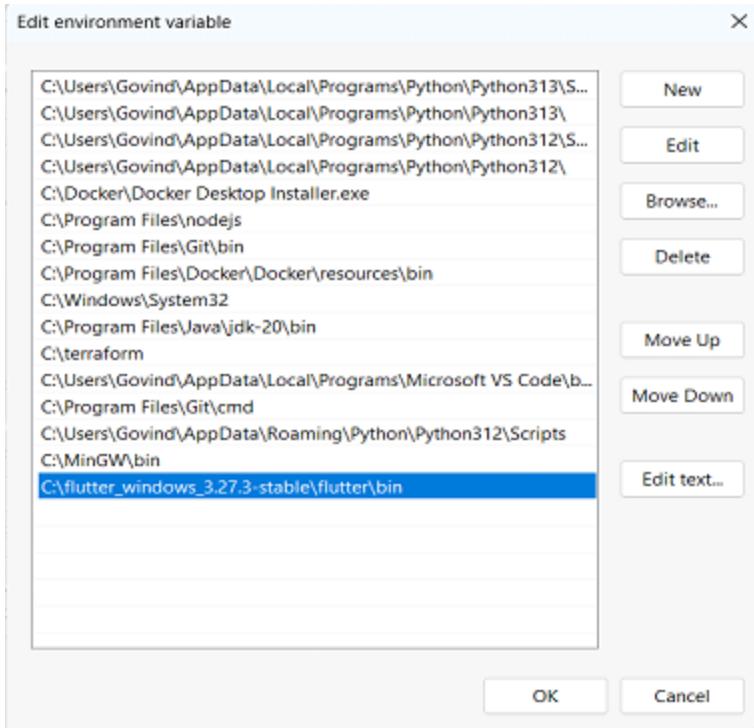
Step 1: Install Flutter

- 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 in the top right corner. 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 menu includes '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' (selected), 'macOS', 'Linux', and 'ChromeOS'. Below these options, a note for developers in China provides instructions for using Flutter in China. 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 official 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 contains the same navigation links as the previous page. The main content area has a heading 'Install the Flutter SDK' and a sub-section 'Download then install Flutter'. It provides instructions for downloading the Flutter SDK bundle and extracting it to a preferred location. A download link for 'flutter_windows_3.27.3-stable.zip' is shown. To the right of the main content, there is a 'Contents' sidebar with links to various setup and configuration guides, such as 'Verify system requirements', 'Install the Flutter SDK', 'Configure Android development', and 'Check your development setup'. The overall design is consistent with the first screenshot, maintaining a professional and user-friendly appearance.

- 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.

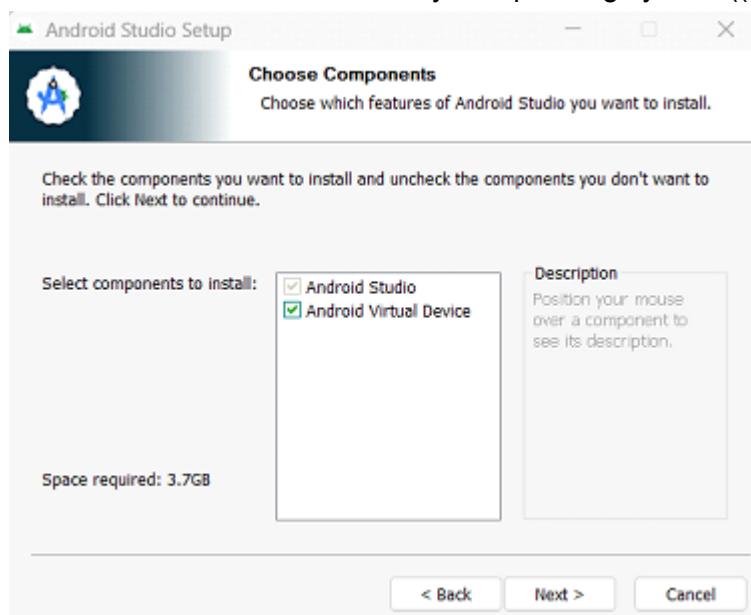
Telemetry is not sent on the very first run. To disable reporting of telemetry,
run this terminal command:

  flutter --disable-analytics

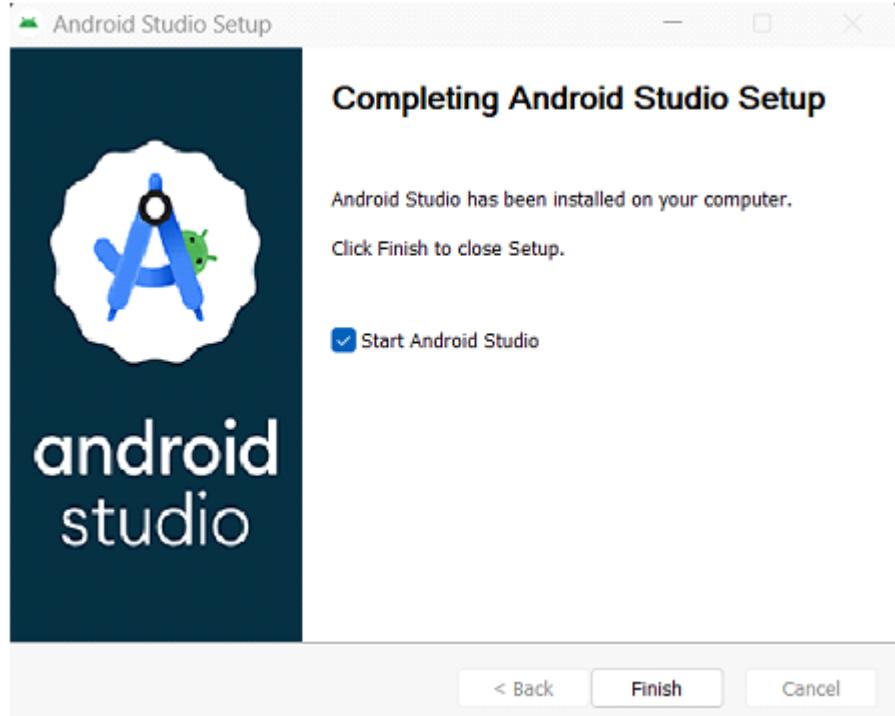
If you opt out of telemetry, an opt-out event will be sent, and then no further
```

Step 2: Install Android Studio

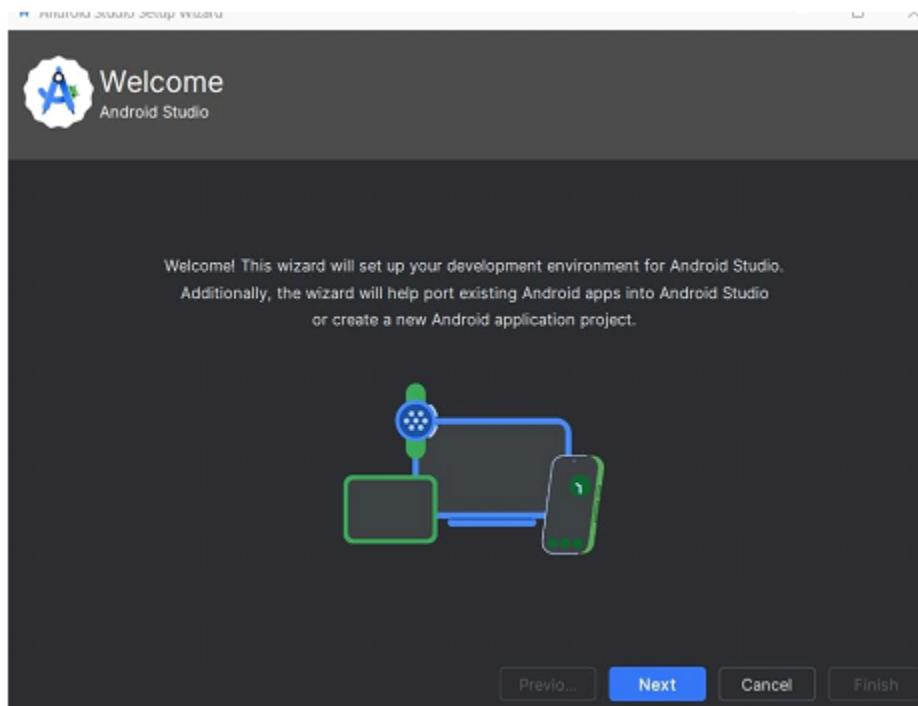
- Download Android Studio
- Go to the Android Studio website. (<https://developer.android.com/studio>)
- Download the installer for your operating system ((Windows, macOS, or Linux)).



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

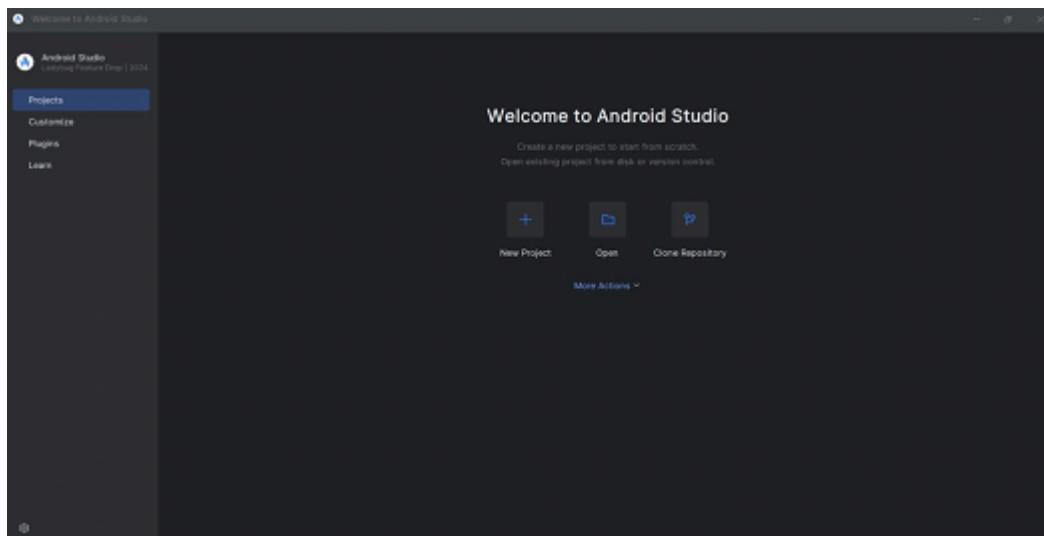
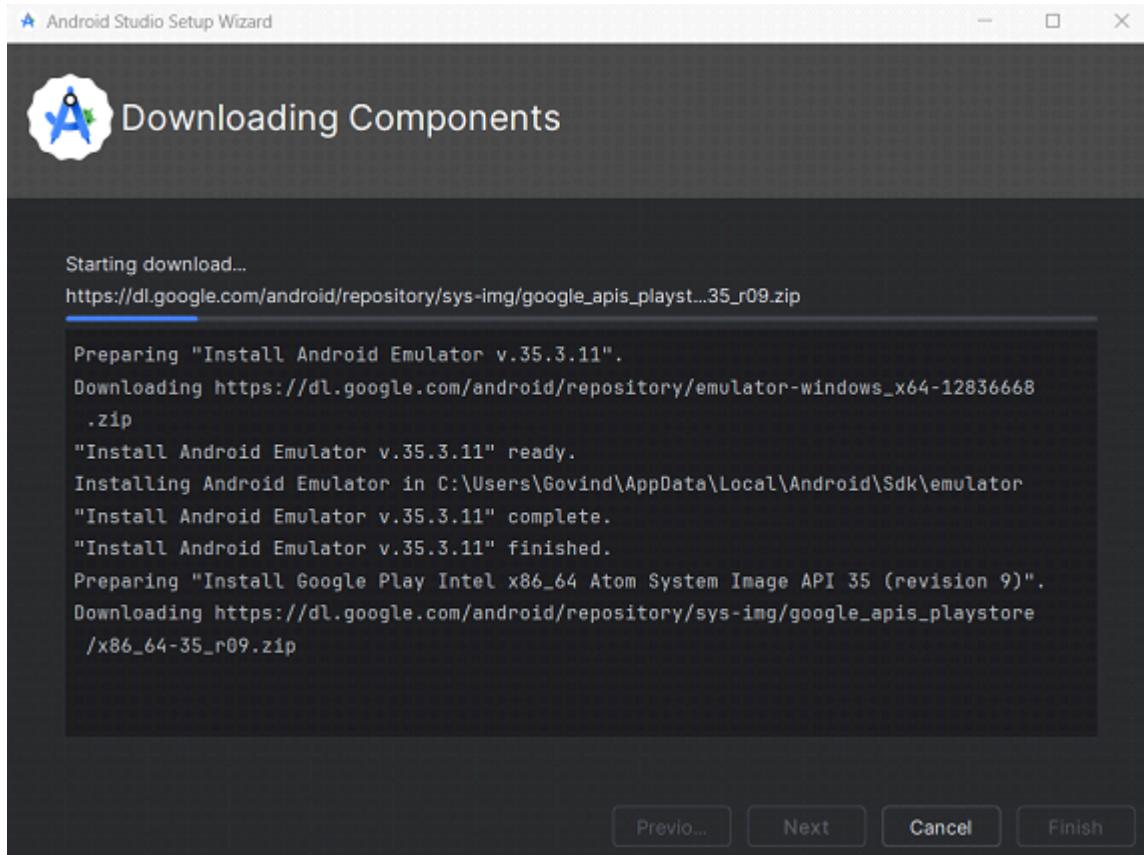


- Install Android SDK Tools
- Open Android Studio.



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

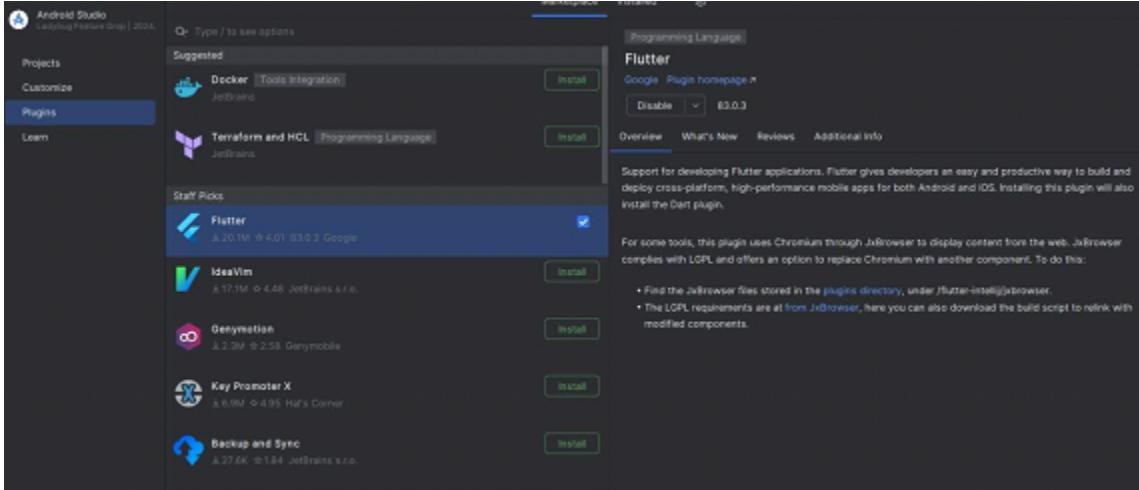
- Select the latest Android API level.
- Ensure "Android SDK Platform" and "Android Virtual Device (AVD)" are selected.
- Click "Apply" and wait for the components to install.



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

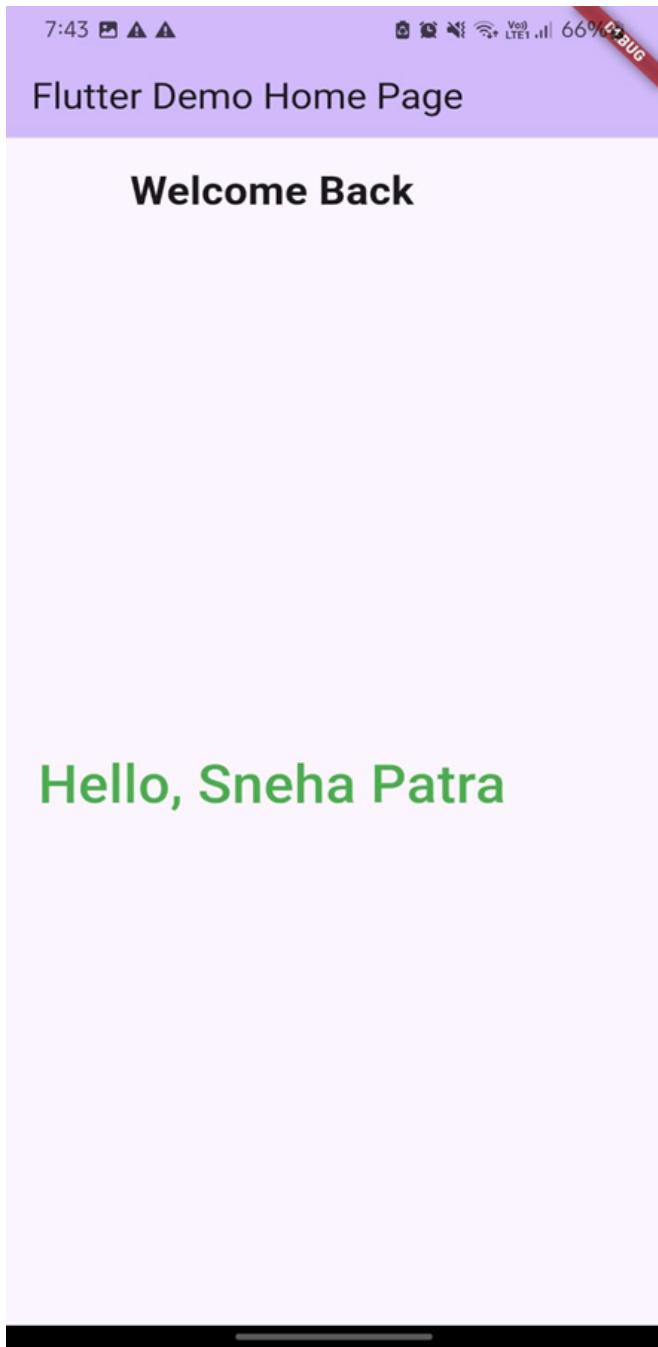
- Click on New Flutter Project.
- Enter project details and select the Flutter SDK path.
- Click "Finish" to create the project.
- Connect the USB to the device and run the flutter application.

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 Sneha Patra',  
        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: Sneha Patra

Class: D15A

Roll no:40

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 JioMart-like home screen, combining a ListView with a CarouselSlider, GridView for categories, Card for trending items, and an unchanged BottomNavigationBar can create a user-friendly experience.

Code:

```
import 'package:flutter/material.dart';
import './screens/sign_in_page.dart';
import './screens/cart_page.dart';
import './widgets/category_list.dart';
import './widgets/product_grid.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: const HomeScreen(),
    );
}
```

```
class HomeScreen extends StatefulWidget {
  const HomeScreen({Key? key}) : super(key: key);

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

class _HomeScreenState extends State<HomeScreen> {
  int _currentIndex = 0;
  final List<Widget> _pages = [
    HomeContent(),
    CategoryList(),
    ProductGrid(),
    SignInPage(),
  ];
}

@Override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: Colors.blueAccent,
      title: Row(
        children: [
          Expanded(
            child: TextField(
              decoration: InputDecoration(
                hintText: 'Search',
                border: InputBorder.none,
                prefixIcon: Icon(Icons.search, color: Colors.white),
              ),
            ),
          ),
        ],
      ),
      dropdownButton< String>(
        icon: Icon(Icons.arrow_drop_down, color: Colors.white),
        items: ["Mumbai", "Delhi", "Bangalore"]
        .map((String value) => DropdownMenuItem<String>(
          value: value,
          child: Text(value),
        ))
        .toList(),
        onChanged: (String? newValue) {},
      ),
    ),
  );
}
```

```
actions: [
    IconButton(
        icon: Icon(Icons.shopping_cart, color: Colors.white),
        onPressed: () {
            Navigator.push(
                context, MaterialPageRoute(builder: (context) => CartPage()));
        },
    ),
],
),
),
body: _pages[_currentIndex],
bottomNavigationBar: BottomNavigationBar(
    currentIndex: _currentIndex,
    type: BottomNavigationBarType.fixed,
    selectedItemColor: Colors.blueAccent,
    unselectedItemColor: Colors.grey,
    onTap: (index) {
        setState(() {
            _currentIndex = index;
        });
    },
    items: [
        BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'),
        BottomNavigationBarItem(icon: Icon(Icons.category), label: 'Categories'),
        BottomNavigationBarItem(icon: Icon(Icons.shopping_bag), label: 'Orders'),
        BottomNavigationBarItem(icon: Icon(Icons.person), label: 'Profile'),
    ],
),
);
}
}
```

```
class HomeContent extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return SingleChildScrollView(
            child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                    Padding(
                        padding: const EdgeInsets.all(8.0),
                        child: Row(
                            children: [
                                Icon(Icons.location_on, color: Colors.black54),
```

```
        SizedBox(width: 5),
        Text("Mumbai 400020", style: TextStyle(fontSize: 16)),
        Spacer(),
        Icon(Icons.arrow_drop_down, color: Colors.black54),
      ],
    ),
  ),
  SizedBox(height: 10),
  Container(
    height: 150,
    margin: EdgeInsets.symmetric(horizontal: 10),
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(10),
      image: DecorationImage(
        image: AssetImage('assets/banner.jpg'),
        fit: BoxFit.cover,
      ),
    ),
  ),
  SizedBox(height: 10),
  Padding(
    padding: const EdgeInsets.symmetric(horizontal: 10.0),
    child: Text(
      "BEST DEALS",
      style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
    ),
  ),
  SizedBox(height: 20),
  Wrap(
    spacing: 16,
    runSpacing: 16,
    alignment: WrapAlignment.center,
    children: [
      _dealCard(context, 'Up to 50% Off', 'Biscuits,Drinks', 'assets/biscuits.png'),
      _dealCard(context, 'Up to 30% Off', 'Cooking Essentials', 'assets/tea.png'),
      _dealCard(context, 'Up to 25% Off', 'Fruit & Vegetable', 'assets/juices.png'),
      _dealCard(context, 'Up to 25% Off', 'Personal Care', 'assets/juices.png'),
      _dealCard(context, 'Up to 25% Off', 'Beauty Products', 'assets/juices.png'),
      _dealCard(context, 'Up to 25% Off', 'Home Care', 'assets/juices.png'),
    ],
  ),
  SizedBox(height: 20),
],
),
```

```
        );
    }

Widget _dealCard(BuildContext context, String discount, String title, String asset) {
    return GestureDetector(
        onTap: () {
            Navigator.push(
                context, MaterialPageRoute(builder: (context) => ProductGrid()));
        },
        child: Column(
            children: [
                Container(
                    height: 80,
                    width: 80,
                    decoration: BoxDecoration(
                        image: DecorationImage(image: AssetImage(asset), fit: BoxFit.cover),
                    ),
                ),
                SizedBox(height: 5),
                Text(discount, style: TextStyle(color: Colors.green, fontSize: 12)),
                Text(title, style: TextStyle(fontSize: 14, fontWeight: FontWeight.bold)),
            ],
        ),
    );
}

class CartPage extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(title: Text("My Cart")),
            body: Center(child: Text("Your cart is empty")),
        );
    }
}

//category list//
import 'package:flutter/material.dart';
```

```
class CategoryList extends StatelessWidget {
  const CategoryList({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.blue,
        title: const TextField(
          decoration: InputDecoration(
            hintText: "Search JioMart",
            hintStyle: TextStyle(color: Colors.white70),
            prefixIcon: Icon(Icons.search, color: Colors.white),
            border: InputBorder.none,
          ),
          style: TextStyle(color: Colors.white),
        ),
        actions: [
          IconButton(
            icon: const Icon(Icons.shopping_cart),
            onPressed: () {},
          ),
        ],
      ),
      body: SingleChildScrollView(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              // Shopping List card

              // Try something new section
              const Text(
                "Try something new",
                style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
              ),
              const SizedBox(height: 10),
              Wrap(
                spacing: 8.0,
                runSpacing: 8.0,
                children: [
                  _buildChip("oil"),

```

```
        _buildChip("sugar"),
        _buildChip("biscuits"),
        _buildChip("ghee"),
        _buildChip("atta"),
        _buildChip("surf excel powder"),
        _buildChip("mustard oil"),
        _buildChip("maggi"),
        _buildChip("rice"),
        _buildChip("soap"),
    ],
),
const SizedBox(height: 30),
// Most trending categories section
const Text(
    "Most trending categories",
    style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
),
const SizedBox(height: 10),
GridView.builder(
    shrinkWrap: true,
    physics: const NeverScrollableScrollPhysics(),
    gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
        crossAxisCount: 4,
        mainAxisSpacing: 16,
        crossAxisSpacing: 16,
),
itemCount: 8,
itemBuilder: (context, index) {
    return Column(
        children: [
            CircleAvatar(
                radius: 30,
                backgroundImage: NetworkImage(
                    "https://via.placeholder.com/100", // Replace with your image URL
                ),
            ),
            const SizedBox(height: 8),
            Text(
                ["Groceries", "Electronics", "Fashion", "Jewellery", "Toys", "Furniture",
                "Books", "Beauty"][index],
                style: const TextStyle(fontSize: 14),
            ),
        ],
);
}
```

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

Widget _buildChip(String label) {
    return Chip(
        label: Text(label),
        backgroundColor: Colors.grey[200],
    );
}
}
```

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

class ProductGrid extends StatelessWidget {
    const ProductGrid({Key? key}) : super(key: key);

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                backgroundColor: Colors.blue,
                title: const TextField(
                    decoration: InputDecoration(
                        hintText: "Search JioMart",
                        hintStyle: TextStyle(color: Colors.white70),
                        prefixIcon: Icon(Icons.search, color: Colors.white),
                        border: InputBorder.none,
                    ),
                    style: TextStyle(color: Colors.white),
                ),
                actions: [
                    IconButton(
                        icon: const Icon(Icons.shopping_cart),

```

```
    onPressed: () {
        // Add action for cart icon
    },
),
],
),
body: Padding(
    padding: const EdgeInsets.all(8.0),
    child: GridView.builder(
        gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 2,
            crossAxisSpacing: 8.0,
            mainAxisSpacing: 8.0,
            childAspectRatio: 0.75,
        ),
        itemCount: 10, // Example count
        itemBuilder: (context, index) {
            return Card(
                shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(10),
                ),
                elevation: 4,
                child: Column(
                    children: [
                        Expanded(
                            child: Container(
                                decoration: BoxDecoration(
                                    borderRadius: const BorderRadius.only(
                                        topLeft: Radius.circular(10),
                                        topRight: Radius.circular(10),
                                    ),
                                ),
                                image: DecorationImage(
                                    image: NetworkImage("https://via.placeholder.com/150"), // Example
image
                                    fit: BoxFit.cover,
                                ),
                            ),
                        ),
                    ],
                ),
            ),
            const Padding(
                padding: EdgeInsets.all(8.0),
                child: Text(
                    "Product Name",
                    style: TextStyle(

```

```
        fontSize: 16,
        fontWeight: FontWeight.bold,
    ),
    textAlign: TextAlign.center,
),
),
const Text(
"₹999",
style: TextStyle(
fontSize: 14,
color: Colors.green,
),
),
],
),
);
),
),
);
}
}
```

Output:

23:13 2023 11.0 KB/S 77%

Search 

LOWEST PRICES ON GROCERY
10-30 MINUTE DELIVERY

Download the App Now 

BEST DEALS


Up to 50% Off
Biscuits, Drinks


Up to 30% Off
Dairy Items


Up to 25% Off
Fruit & Vegetable


Up to 25% Off
Personal Care


Up to 20% Off
Beauty Products

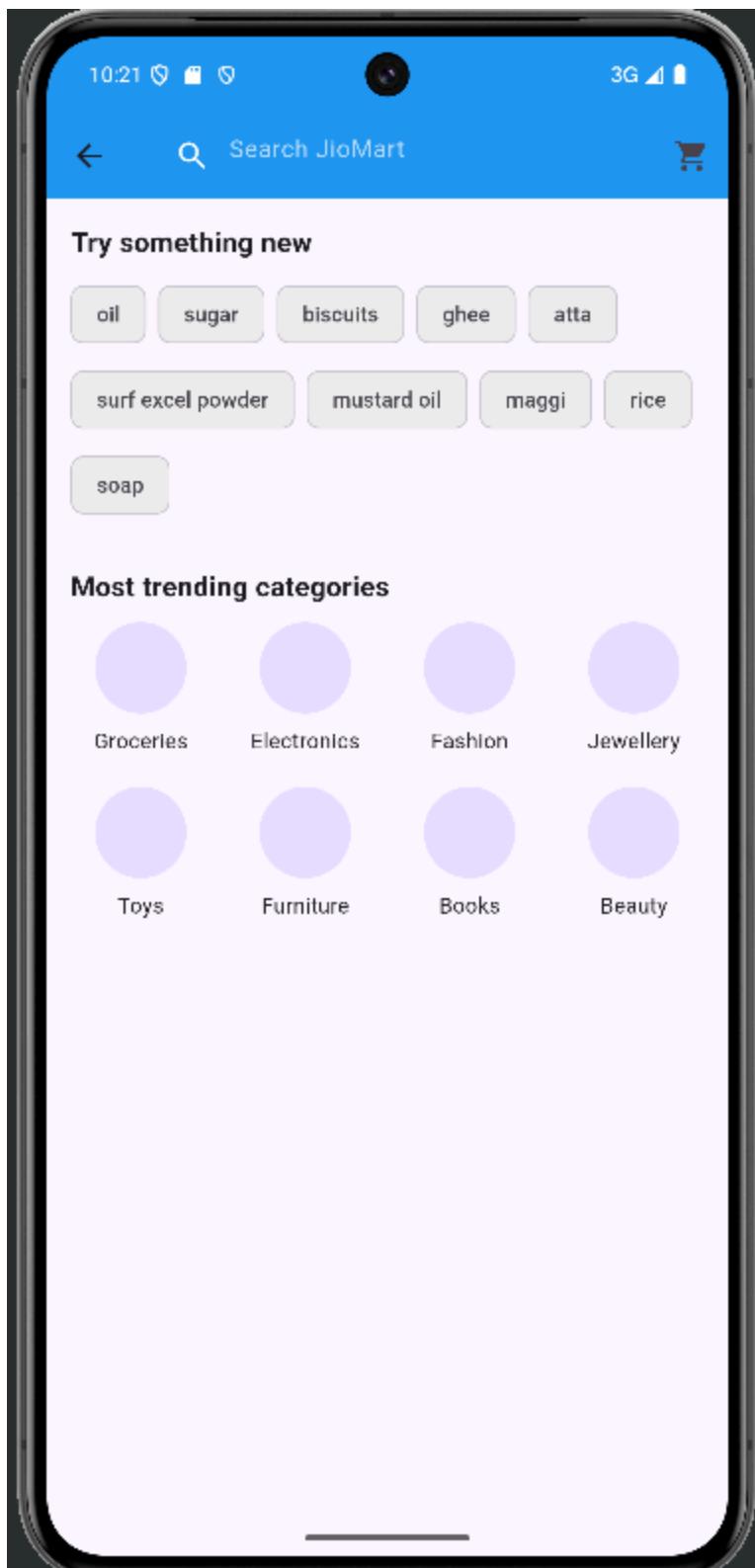

Up to 20% Off
Home Care

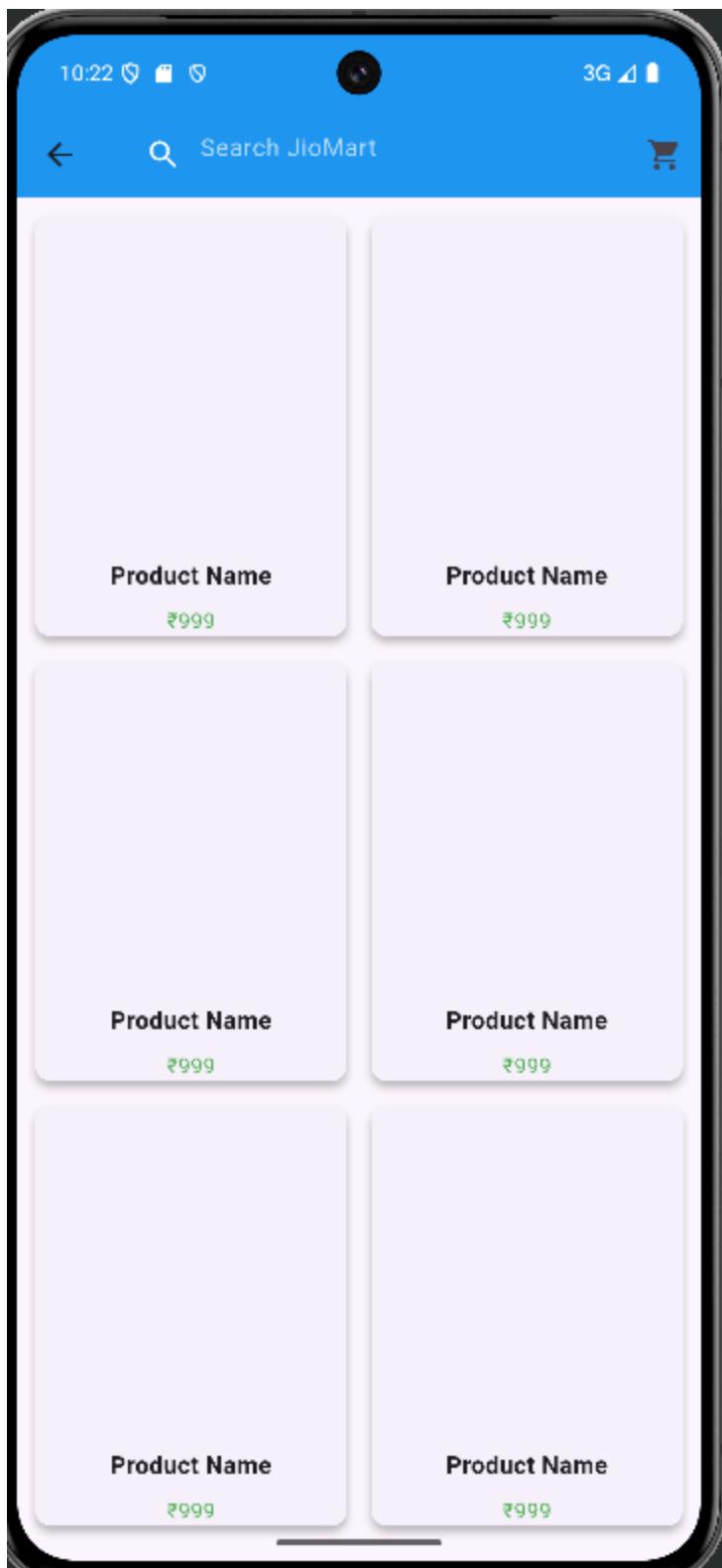
Get Grocery at Incredibly LOW PRICES.

FREE HOME DELIVERY NO MINIMUM ORDER


UP TO 50% OFF

 Home  Categories  Orders  Profile





MPL Experiment 3

Name: Sneha Patra

Class: D15A

Roll no:40

Aim: To include icons, images, fonts in Flutter app

Theory:

Including icons, images, and custom fonts in a Flutter app enhances visual appeal and user experience. Icons can be added using the Icon widget with built-in Icons class or custom icons via the flutter_launcher_icons package.

Images can be loaded from assets using the Image.asset() method or from the internet using Image.network().

To use local images, they must be placed in the assets folder and declared in pubspec.yaml. Custom fonts improve typography and branding; they can be added by placing font files in the assets/fonts directory and specifying them in pubspec.yaml under the flutter section.

Using TextStyle with the fontFamily property applies the custom font to text.

Code:

```
import 'package:flutter/material.dart';
import './screens/sign_in_page.dart';
import './screens/cart_page.dart';
import './widgets/category_list.dart';
import './widgets/product_grid.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: const HomeScreen(),
    );
}

class HomeScreen extends StatefulWidget {
  const HomeScreen({Key? key}) : super(key: key);
```

```
@override
_HOMEScreenState createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
int _currentIndex = 0;
final List<Widget> _pages = [
    HomeContent(),
    CategoryList(),
    ProductGrid(),
    SignInPage(),
];
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            backgroundColor: Colors.blueAccent,
            title: Row(
                children: [
                    Expanded(
                        child: TextField(
                            decoration: InputDecoration(
                                hintText: 'Search',
                                border: InputBorder.none,
                                prefixIcon: Icon(Icons.search, color: Colors.white),
                            ),
                        ),
                    ),
                    ],
                ),
            actions: [
                IconButton(
                    icon: Icon(Icons.shopping_cart, color: Colors.white),
                    onPressed: () {
                        Navigator.push(
                            context, MaterialPageRoute(builder: (context) => CartPage()));
                    },
                ),
            ],
        ),
        body: _pages[_currentIndex],
```

```
bottomNavigationBar: BottomNavigationBar(
  currentIndex: _currentIndex,
  type: BottomNavigationBarType.fixed,
  selectedItemColor: Colors.blueAccent,
  unselectedItemColor: Colors.grey,
  onTap: (index) {
    setState(() {
      _currentIndex = index;
    });
  },
  items: [
    BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'),
    BottomNavigationBarItem(icon: Icon(Icons.category), label: 'Categories'),
    BottomNavigationBarItem(icon: Icon(Icons.shopping_bag), label: 'Orders'),
    BottomNavigationBarItem(icon: Icon(Icons.person), label: 'Profile'),
  ],
),
);
},
);
}
}
```

```
class HomeContent extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return SingleChildScrollView(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: [
          Padding(
            padding: const EdgeInsets.all(8.0),
            child: Row(
              children: [
                Icon(Icons.location_on, color: Colors.black54),
                SizedBox(width: 5),
                Text("Mumbai 400020", style: TextStyle(fontSize: 16)),
                Spacer(),
                Icon(Icons.arrow_drop_down, color: Colors.black54),
              ],
            ),
          ),
          SizedBox(height: 10),
          Container(
            height: 150,
            margin: EdgeInsets.symmetric(horizontal: 10),

```

```
decoration: BoxDecoration(
  borderRadius: BorderRadius.circular(10),
  image: DecorationImage(
    image: AssetImage('assets/banner.jpg'),
    fit: BoxFit.cover,
  ),
),
),
),
),

SizedBox(height: 10),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 10.0),
  child: Text(
    "BEST DEALS",
    style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
  ),
),
),
SizedBox(height: 20),
Wrap(
  spacing: 16,
  runSpacing: 16,
  alignment: WrapAlignment.center,
  children: [
    _dealCard(context, 'Up to 50% Off', 'Biscuits,Drinks', 'assets/biscuit.png'),
    _dealCard(context, 'Up to 30% Off', 'Dairy Items', 'assets/dairy.png'),
    _dealCard(context, 'Up to 25% Off', 'Fruit & Vegetable', 'assets/fruits.png'),
    _dealCard(context, 'Up to 25% Off', 'Personal Care', 'assets/personal care.png'),
    _dealCard(context, 'Up to 20% Off', 'Beauty Products', 'assets/beauty.png'),
    _dealCard(context, 'Up to 20% Off', 'Home Care', 'assets/home care.png'),
  ],
),
SizedBox(height: 10),
Container(
  height: 150,
  margin: EdgeInsets.symmetric(horizontal: 10),
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(10),
    image: DecorationImage(
      image: AssetImage('assets/banner2.png'),
      fit: BoxFit.cover,
    ),
),
),
),
),
SizedBox(height: 20),
```

```

        ],
        ),
        );
    }

Widget _dealCard(BuildContext context, String discount, String title, String asset) {
    return GestureDetector(
        onTap: () {
            Navigator.push(
                context, MaterialPageRoute(builder: (context) => ProductGrid()));
        },
        child: Column(
            children: [
                Container(
                    height: 80,
                    width: 80,
                    decoration: BoxDecoration(
                        image: DecorationImage(image: AssetImage(asset), fit: BoxFit.cover),
                    ),
                ),
                SizedBox(height: 5),
                Text(discount, style: TextStyle(color: Colors.green, fontSize: 12)),
                Text(title, style: TextStyle(fontSize: 14, fontWeight: FontWeight.bold)),
            ],
        ),
    );
}

}

class CartPage extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(title: Text("My Cart")),
            body: Center(child: Text("Your cart is empty")),
        );
    }
}

```

Output:

23:13

11.0 KB/S

77

Search



**LOWEST PRICES
ON GROCERY**

**10-30 MINUTE
DELIVERY**

Download the App Now



BEST DEALS



Up to 50% Off
Biscuits,Drinks



Up to 30% Off
Dairy Items



Up to 25% Off
Fruit & Vegetable



Up to 25% Off
Personal Care



Up to 20% Off
Beauty Products



Up to 20% Off
Home Care

Get Grocery at Incredibly **LOW PRICES.**

FREE HOME
DELIVERY

NO MINIMUM
ORDER



Home

Categories

Orders

Profile



Search JioMart



Try something new

oil

sugar

biscuits

ghee

atta

surf excel powder

mustard oil

maggi

rice

soap

Most trending categories



Groceries



Electronic



Fashion



Jewellery



Toys



Furniture



Books



Beauty



Home



Categories



Orders



Profile

MPL Experiment 4

Name: Sneha Patra

Class: D15A

Roll no:40

Aim: To create an interactive Form using form widgets.

Theory:

Creating an interactive form in Flutter requires using form-related widgets to collect and validate user input efficiently. The Form widget, combined with TextFormField, provides a structured way to manage input fields. 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 or ElevatedButton can trigger validation and submission logic. For a responsive experience, proper padding, spacing, and InputDecoration enhance usability.

Code:

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

class SignInPage extends StatefulWidget {
  const SignInPage({Key? key}) : super(key: key);

  @override
  State<SignInPage> createState() => _SignInPageState();
}

class _SignInPageState extends State<SignInPage> {
  final _formKey = GlobalKey<FormState>();
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();

  String? _validateEmail(String? value) {
    if (value == null || value.isEmpty) {
      return 'Please enter your email';
    }
    const emailPattern = r'^(?![_.])(?!.*\.(?!.*\.))([a-zA-Z][a-zA-Z0-9]*[a-zA-Z])((\.[a-zA-Z][a-zA-Z0-9]*){1,4})$';
    if (!RegExp(emailPattern).hasMatch(value)) {
      return 'Enter a valid email address';
    }
  }
}
```

```
        return null;
    }

String? _validatePassword(String? value) {
    if (value == null || value.isEmpty) {
        return 'Please enter your password';
    }
    if (value.length < 6) {
        return 'Password must be at least 6 characters';
    }
    return null;
}

void _onSubmit() {
    if (_formKey.currentState!.validate()) {
        // All fields are valid
        ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(content: Text('Signing in...')),
        );
        // Add your sign-in logic here
    }
}

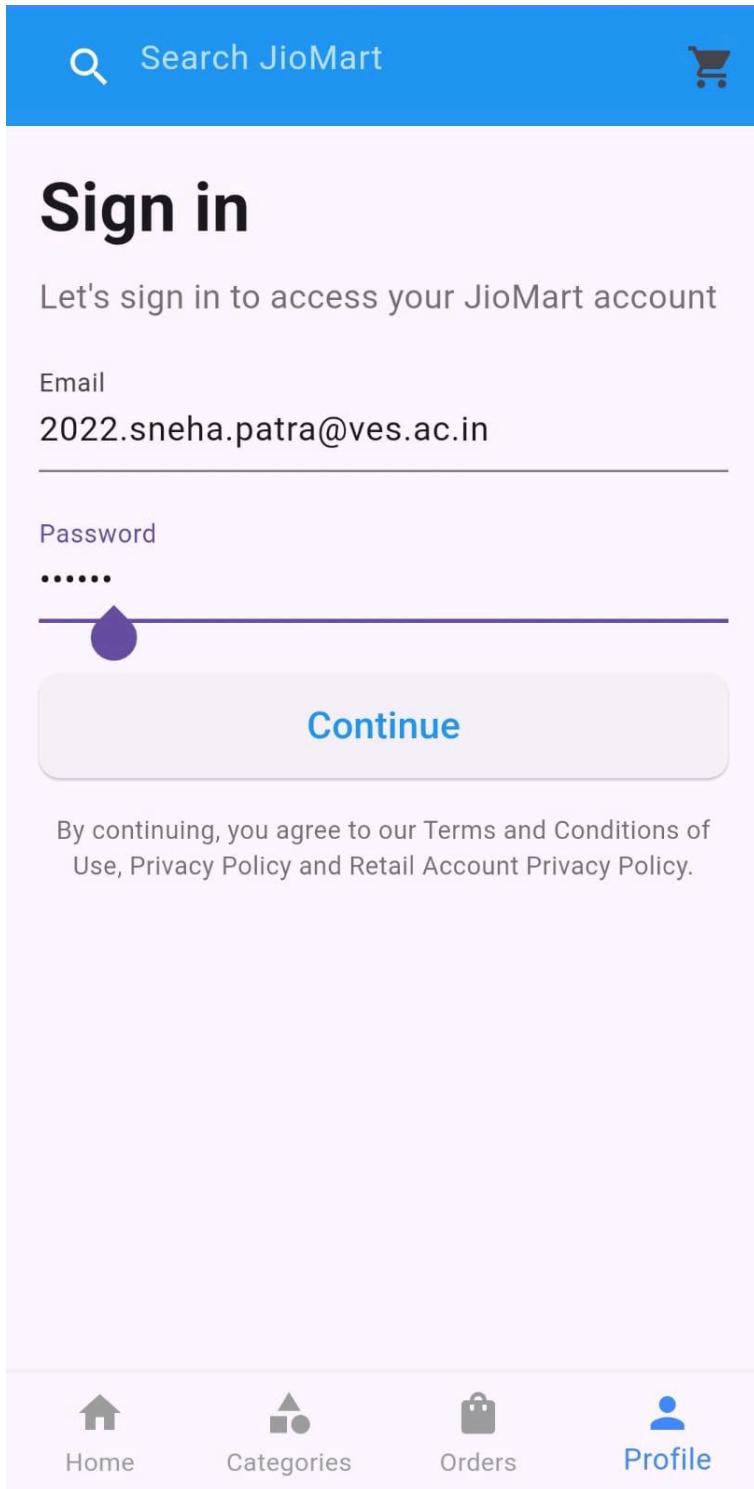
@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            backgroundColor: Colors.blue,
            title: const TextField(
                decoration: InputDecoration(
                    hintText: "Search JioMart",
                    hintStyle: TextStyle(color: Colors.white70),
                    prefixIcon: Icon(Icons.search, color: Colors.white),
                    border: InputBorder.none,
                ),
                style: TextStyle(color: Colors.white),
            ),
            actions: [
                IconButton(
                    icon: const Icon(Icons.shopping_cart),
                    onPressed: () {},
                ),
            ],
        ),
        body: Padding(
            padding: const EdgeInsets.all(16.0),

```

```
child: Form(
  key: _formKey,
  child: Column(
    mainAxisAlignment: MainAxisAlignment.start,
    children: [
      const Text(
        "Sign in",
        style: TextStyle(fontSize: 32, fontWeight: FontWeight.bold),
      ),
      const SizedBox(height: 8),
      const Text(
        "Let's sign in to access your JioMart account",
        style: TextStyle(fontSize: 16, color: Colors.black54),
      ),
      const SizedBox(height: 16),
      TextFormField(
        controller: _emailController,
        decoration: const InputDecoration(
          labelText: "Email",
          border: UnderlineInputBorder(),
        ),
        keyboardType: TextInputType.emailAddress,
        validator: _validateEmail,
      ),
      const SizedBox(height: 16),
      TextFormField(
        controller: _passwordController,
        decoration: const InputDecoration(
          labelText: "Password",
          border: UnderlineInputBorder(),
        ),
        obscureText: true,
        validator: _validatePassword,
      ),
      const SizedBox(height: 24),
      SizedBox(
        width: double.infinity,
        height: 50,
        child: ElevatedButton(
          onPressed: _onSubmit,
          style: ElevatedButton.styleFrom(
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(10),
            ),
          ),
          child: const Text(
            "Continue",
            style: TextStyle(fontSize: 18, color: Colors.blue),
          ),
        ),
      ),
    ],
  ),
);
```

```
        ) ,
    ) ,
const SizedBox(height: 16) ,
const Text(
    "By continuing, you agree to our Terms and Conditions of Use,
Privacy Policy and Retail Account Privacy Policy.",
    style: TextStyle(fontSize: 12, color: Colors.black54),
    textAlign: TextAlign.center,
),
],
),
),
),
);
}
}
```

Output:





Search JioMart



Sign in

Let's sign in to access your JioMart account

Email

sneha

Enter a valid email address

Password

....

>Password must be at least 6 characters

Continue

By continuing, you agree to our Terms and Conditions of Use, Privacy Policy and Retail Account Privacy Policy.



Home



Categories



Orders



Profile

MPL Experiment 5

Name: Sneha Patra

Class: D15A

Roll no:40

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:

```
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:jiomart/Screens/product_grid.dart';
import './screens/cart_page.dart';
import './screens/myaccount.dart';
import './widgets/category_list.dart';
```

```
import '../Screens/orders_page.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: const HomeScreen(),
    );
  }
}

class HomeScreen extends StatefulWidget {
  const HomeScreen({Key? key}) : super(key: key);

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

class _HomeScreenState extends State<HomeScreen> {
  int _currentIndex = 0;

  final List<Widget> _pages = [
    HomeContent(),
    CategoryList(),
    OrdersPage(),
    MyAccountPage(),
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.blueAccent,
        title: Text(
          "JioMart",
          style: TextStyle(
            color: Colors.white,
          ),
        ),
      ),
      body: _pages[_currentIndex],
    );
  }
}
```

```
        fontSize: 22,
        fontWeight: FontWeight.bold,
        fontFamily: 'Roboto', // Change to any font you like
    ),
),
),
centerTitle: true, // Centers the logo text

actions: [
    IconButton(
        icon: Icon(Icons.search, color: Colors.white),
        onPressed: () {
            // Add search functionality if needed
        },
    ),
],
),
),

body: _pages[_currentIndex],
bottomNavigationBar: BottomNavigationBar(
    currentIndex: _currentIndex,
    type: BottomNavigationBarType.fixed,
    selectedItemColor: Colors.blueAccent,
    unselectedItemColor: Colors.grey,
    onTap: (index) {
        setState(() {
            _currentIndex = index;
        });
    },
    items: [
        BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'),
        BottomNavigationBarItem(icon: Icon(Icons.category), label: 'Categories'),
        BottomNavigationBarItem(icon: Icon(Icons.shopping_bag), label: 'Orders'),
        BottomNavigationBarItem(icon: Icon(Icons.person), label: 'Profile'),
    ],
),
);
},
}
}

class HomeContent extends StatefulWidget {
    @override
    _HomeContentState createState() => _HomeContentState();
}
```

```
class _HomeContentState extends State<HomeContent> {
  Map<String, int> cartQuantities = {};

  @override
  void initState() {
    super.initState();
    // Initialize quantities from existing cart items
    for (var item in cartItems) {
      cartQuantities[item['name']] = item['quantity'];
    }
  }

  // Function to fetch products from Firestore
  Future<List<Map<String, dynamic>>> fetchProductsForCategory(String category) async
  {
    List<Map<String, dynamic>> products = [];
    try {
      DocumentSnapshot categorySnapshot =
        await FirebaseFirestore.instance.collection("deals").doc(category).get();

      if (categorySnapshot.exists) {
        var categoryData = categorySnapshot.data() as Map<String, dynamic>?;
        if (categoryData != null && categoryData.containsKey('products')) {
          var productList = categoryData['products'] as List<dynamic>;
          for (var product in productList) {
            products.add({
              'name': product['name'],
              'price': product['price'],
              'imageUrl': product['imageUrl'],
            });
          }
        }
      }
    } catch (e) {
      print("Error fetching category products: $e");
    }
    return products;
  }

  // Function to add/update product in cart
  void _updateCart(Map<String, dynamic> product, int quantity) {
    if (quantity <= 0) {
      // Remove product from cart if quantity is 0
      setState(() {
```

```
        cartItems.removeWhere((item) => item['name'] == product['name']);
        cartQuantities[product['name']] = 0;
    });
    return;
}

// Check if product already exists in cart
int existingIndex = cartItems.indexWhere((item) => item['name'] == product['name']);

setState() {
    if (existingIndex >= 0) {
        // Update existing product quantity
        cartItems[existingIndex]['quantity'] = quantity;
    } else {
        // Add new product to cart
        cartItems.add({
            'name': product['name'],
            'price': product['price'],
            'imageUrl': product['imageUrl'],
            'quantity': quantity,
        });
    }
    // Update local quantity tracker
    cartQuantities[product['name']] = quantity;
});
}

@Override
Widget build(BuildContext context) {
    return SingleChildScrollView(
        child: Column(
            mainAxisAlignment: MainAxisAlignment.start,
            children: [
                Padding(
                    padding: const EdgeInsets.all(8.0),
                    child: Row(
                        children: [
                            Icon(Icons.location_on, color: Colors.black54),
                            SizedBox(width: 5),
                            Text("Mumbai 400020", style: TextStyle(fontSize: 16)),
                            Spacer(),
                            Icon(Icons.arrow_drop_down, color: Colors.black54),
                        ],
                    ),
                ),
            ],
        ),
    );
}
```

```
),
SizedBox(height: 10),
Container(
  height: 150,
  margin: EdgeInsets.symmetric(horizontal: 10),
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(10),
    image: DecorationImage(
      image: AssetImage('assets/banner.jpg'),
      fit: BoxFit.cover,
    ),
  ),
),
SizedBox(height: 10),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 10.0),
  child: Text(
    "BEST DEALS",
    style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
  ),
),
SizedBox(height: 20),
Wrap(
  spacing: 16,
  runSpacing: 16,
  alignment: WrapAlignment.center,
  children: [
    _dealCard(context, 'Up to 50% Off', 'Biscuits', 'assets/biscuit.png'),
    _dealCard(context, 'Up to 30% Off', 'dairy', 'assets/dairy.png'),
    _dealCard(context, 'Up to 25% Off', 'fruits', 'assets/fruits.png'),
    _dealCard(context, 'Up to 25% Off', 'personal care', 'assets/personal care.png'),
    _dealCard(context, 'Up to 20% Off', 'beauty', 'assets/beauty.png'),
    _dealCard(context, 'Up to 20% Off', 'home care', 'assets/home care.png'),
  ],
),
SizedBox(height: 10),
Container(
  height: 150,
  margin: EdgeInsets.symmetric(horizontal: 10),
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(10),
    image: DecorationImage(
      image: AssetImage('assets/banner2.png'),
      fit: BoxFit.cover,
```

```

        ),
        ),
        ),
        SizedBox(height: 20),
    ],
),
);
}

/// ✅ Fixed ` _dealCard` method to fetch products before navigating
Widget _dealCard(BuildContext context, String discount, String category, String asset) {
  return FutureBuilder<List<Map<String, dynamic>>>(
    future: fetchProductsForCategory(category),
    builder: (context, snapshot) {
      if (snapshot.connectionState == ConnectionState.waiting) {
        return CircularProgressIndicator(); // Show loading indicator while fetching
      } else if (snapshot.hasError) {
        return Text('Error: ${snapshot.error}');
      } else if (!snapshot.hasData || snapshot.data!.isEmpty) {
        return Text('No products available');
      } else {
        List<Map<String, dynamic>> products = snapshot.data!;
        return GestureDetector(
          onTap: () {
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => ProductGrid(products: products),
              ),
            ).then((_) {
              setState(() {}); // Refresh the HomeContent when returning from ProductGrid
            });
          },
          child: Column(
            children: [
              Container(
                height: 80,
                width: 80,
                decoration: BoxDecoration(
                  image: DecorationImage(image: AssetImage(asset), fit: BoxFit.cover),
                ),
              ),
              SizedBox(height: 5),
              Text(discount, style: TextStyle(color: Colors.green, fontSize: 12)),
            ],
          ),
        );
      }
    }
  );
}

```

```
        Text(category.toUpperCase(), style: TextStyle(fontSize: 14, fontWeight:  
FontWeight.bold)),  
        ],  
      ),  
    );  
  },  
);  
}  
}
```

Output:

23:13

11.0 KB/S

77

Search



**LOWEST PRICES
ON GROCERY**

**10-30 MINUTE
DELIVERY**

Download the App Now



BEST DEALS



Up to 50% Off
Biscuits,Drinks



Up to 30% Off
Dairy Items



Up to 25% Off
Fruit & Vegetable



Up to 25% Off
Personal Care



Up to 20% Off
Beauty Products



Up to 20% Off
Home Care

Get Grocery at Incredibly **LOW PRICES.**

FREE HOME
DELIVERY

NO MINIMUM
ORDER



Home

Categories

Orders

Profile

01:07 • 5.00 KB/S 79

JioMart



DAIRY



FRUITS



VEGETABLES



COOKING ESSENTIALS



BABYCARE



STATIONERY



KITCHENWARE



Home



Categories



Orders



Profile

01:07 8.00 KB/S 79

JioMart



My Orders

Your Last Order:

No orders yet!



Home



Categories



Orders



Profile

01:07



JioMart



Sneha Patra

patrasneha21@gmail.com

My Account

My Notifications >

My List >

Delivery Addresses >

PAN Card Information >

MilkBasket >

Payment Modes

JioMart Wallet >

Sign Out



Home



Categories



Orders



Profile

01:08 Snapchat 📲 VI 🎵 🔍 • 🔁 2.00 KB/S ⌂ 79 🔋

← My Cart



Oreo Pokemon Cream...

₹30

- 1 +



Apple Red Delicious 4 ...

₹189

- 1 +



Total: ₹219.00

Proceed to Checkout

01:08 3.00 KB/S 79%

← Checkout Page

Delivery Address

ion, Mumbai, Maharashtra 400022, India

Payment Method

Cash on Delivery

Place Order

MPL Experiment 6

Name: Sneha Patra

Class: D15A

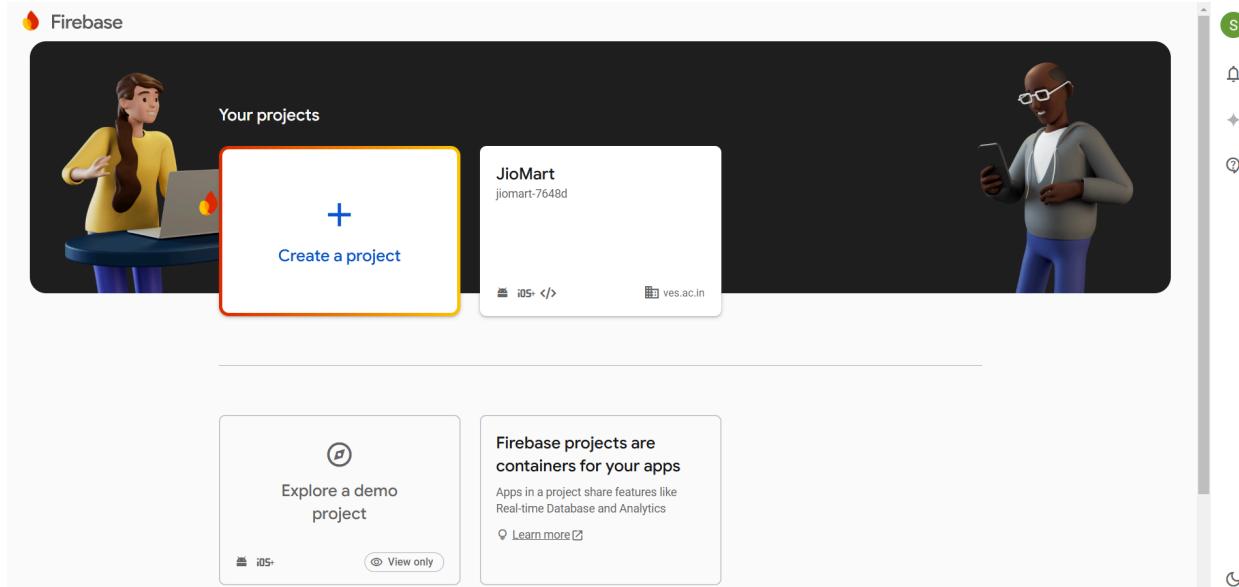
Roll no: 40

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.



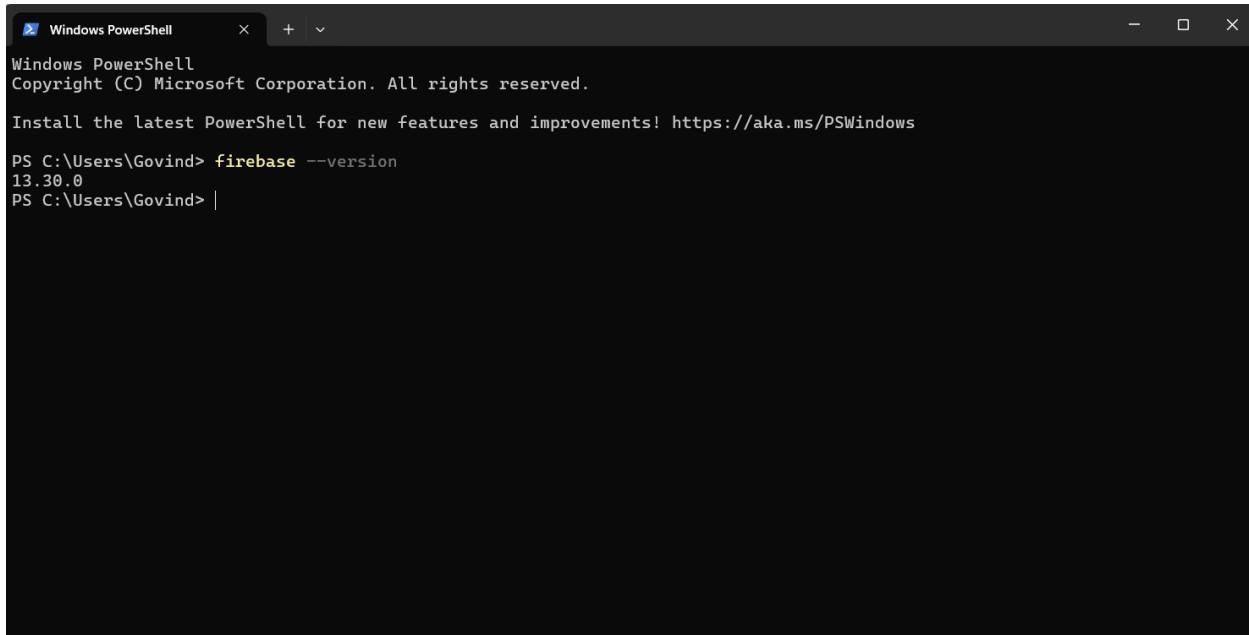
Step 2:

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
```

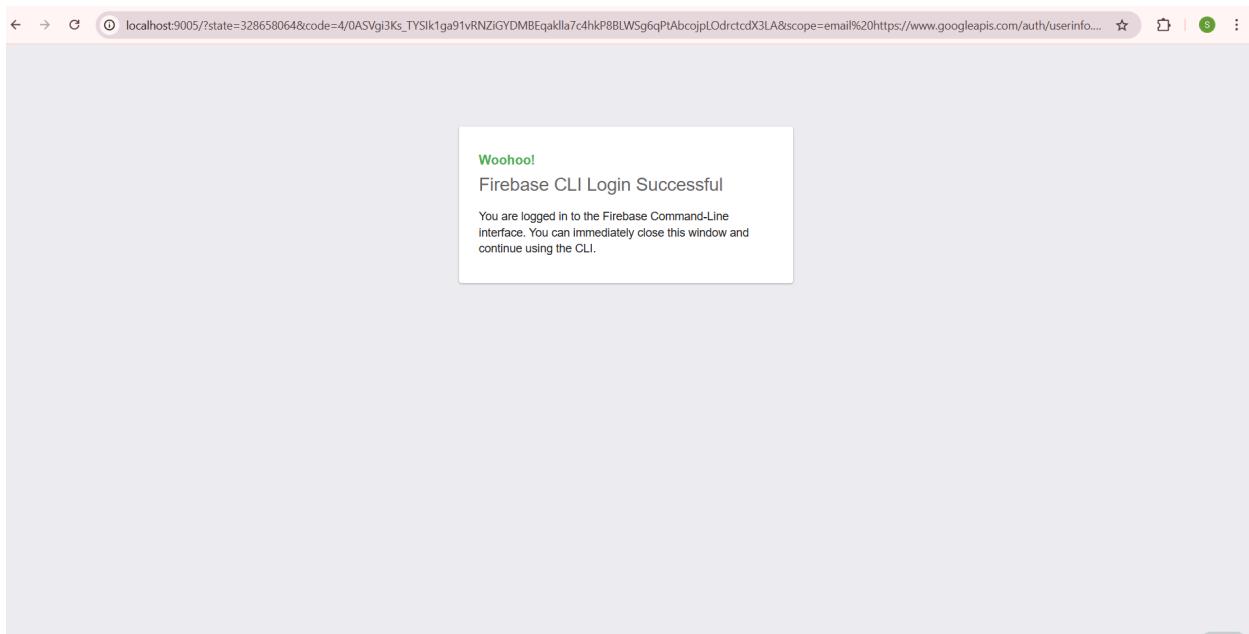
This will install the **Firebase CLI**, check the version, and log you into your Firebase account.



```
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\Govind> firebase --version
13.30.0
PS C:\Users\Govind> |
```



Step 3:

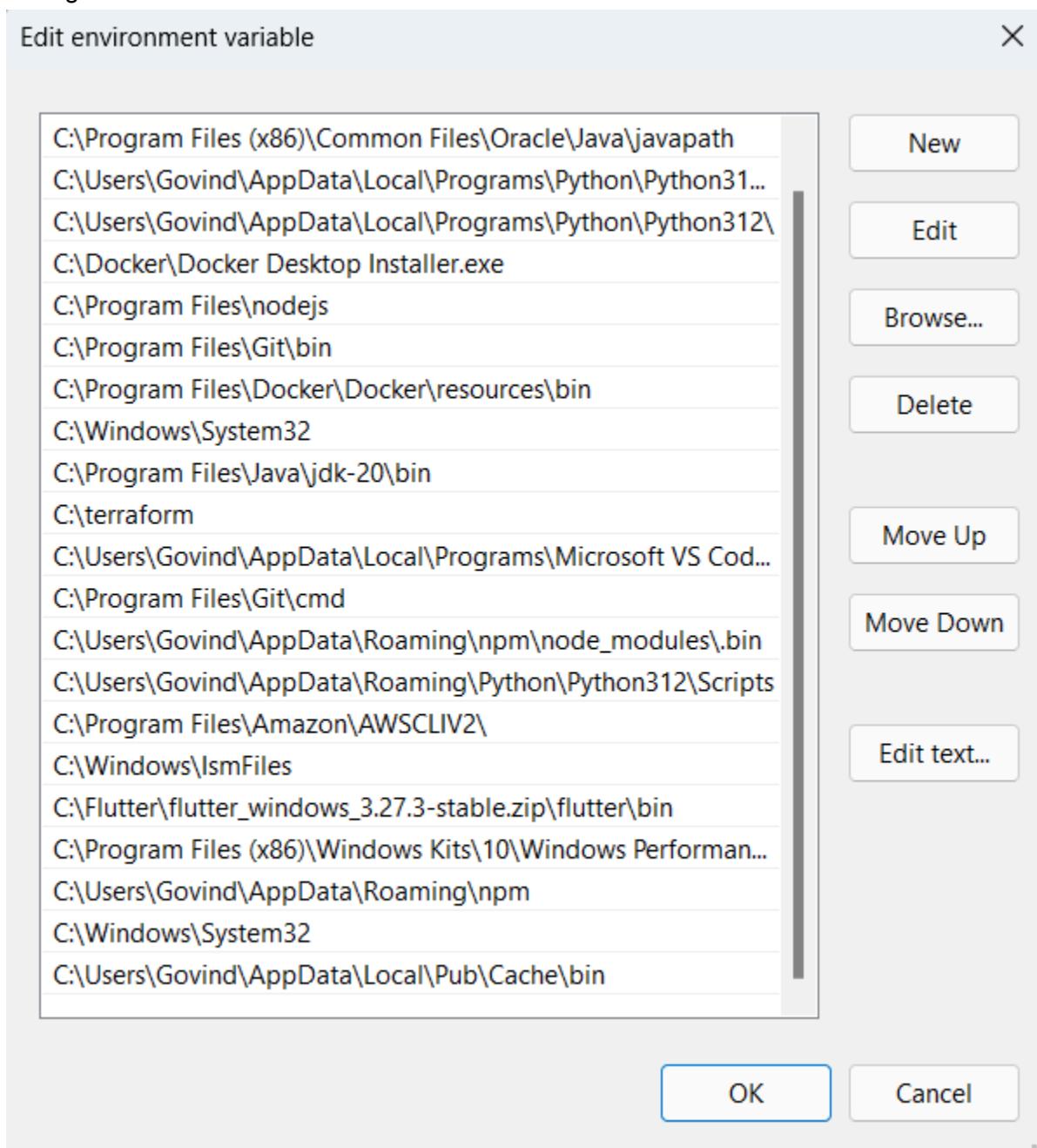
Open your Flutter app in **Android Studio**.

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.

```
Terminal Local + ▾
...
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

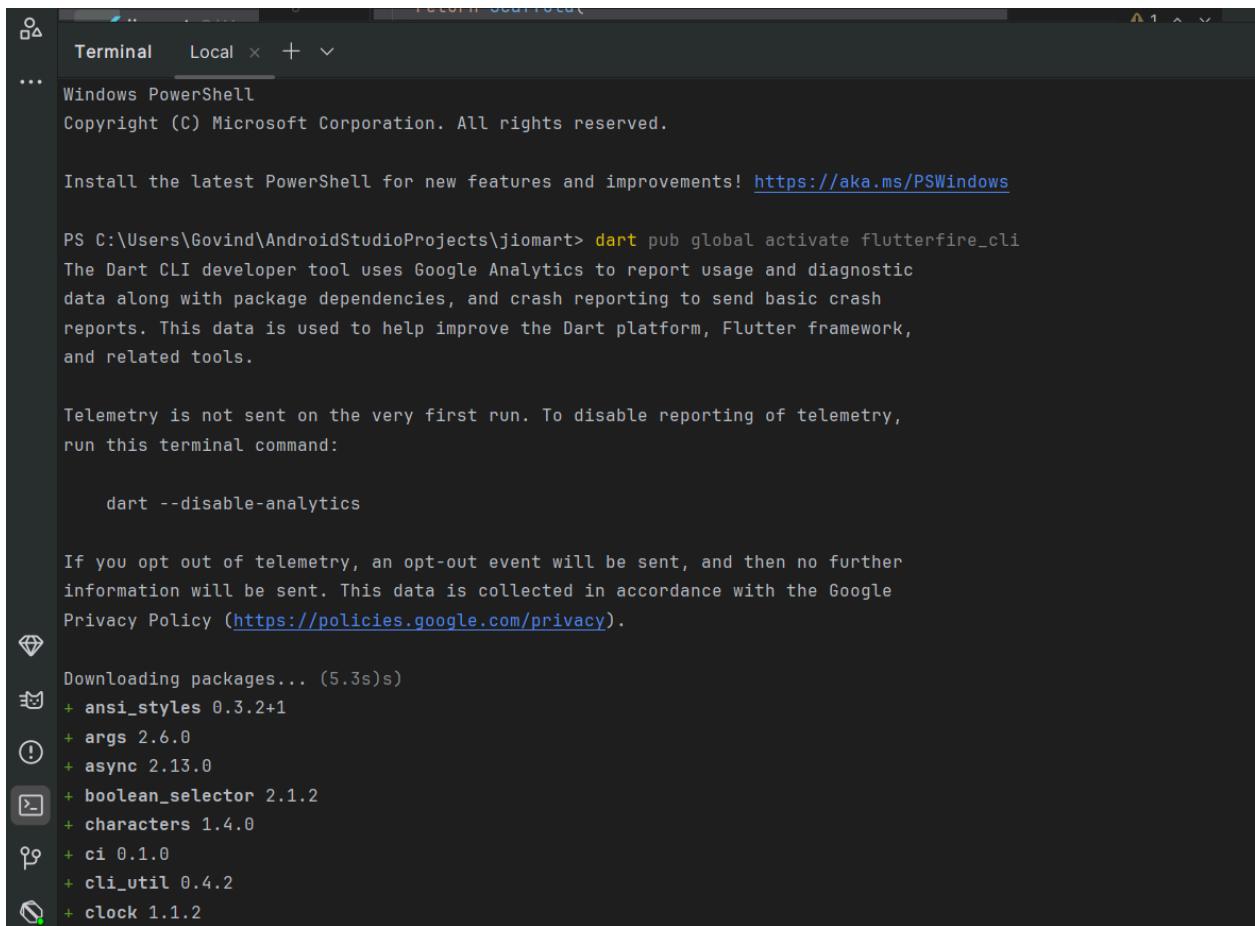
PS C:\Users\Govind\AndroidStudioProjects\jiomart> dart pub global activate flutterfire_cli
The Dart CLI developer tool uses Google Analytics to report usage and diagnostic
data along with package dependencies, and crash reporting to send basic crash
reports. This data is used to help improve the Dart 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).

Diamond: Downloading packages... (5.3s)
  + ansi_styles 0.3.2+1
  + args 2.6.0
  + async 2.13.0
  + boolean_selector 2.1.2
  + characters 1.4.0
  + ci 0.1.0
  + cli_util 0.4.2
  + clock 1.1.2
```



The screenshot shows a Windows Terminal window with a single tab labeled "Terminal". The terminal is running a Windows PowerShell session. The output of the command "dart pub global activate flutterfire_cli" is displayed. The output includes instructions about telemetry, how to disable it, and a list of packages being downloaded. The terminal interface has a dark theme with light-colored text.

```
Terminal Local + ▾
...
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\Govind\AndroidStudioProjects\jiomart> 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).

Diamond: Downloading packages... (5.3s)
Email: + ansi_styles 0.3.2+1
Error: + args 2.6.0
Sync: + async 2.13.0
Copy: + boolean_selector 2.1.2
Copy: + characters 1.4.0
Copy: + ci 0.1.0
Copy: + cli_util 0.4.2
Clock: + clock 1.1.2
```

Step 5:

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

```
flutter pub add firebase_core
```

```
Learn more about using this file and next steps from the documentation:  
> https://firebase.google.com/docs/flutter/setup  
PS C:\Users\Govind\AndroidStudioProjects\jiomart> flutter pub add firebase_core  
Resolving dependencies...  
Downloading packages... (2.7s)  
  async 2.11.0 (2.13.0 available)  
  boolean_selector 2.1.1 (2.1.2 available)  
  characters 1.3.0 (1.4.0 available)  
  clock 1.1.1 (1.1.2 available)  
  fake_async 1.3.1 (1.3.3 available)  
+ firebase_core 3.11.0  
+ firebase_core_platform_interface 5.4.0  
+ firebase_core_web 2.20.0  
+ 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)  
meta 1.15.0 (1.16.0 available)  
path 1.9.0 (1.9.1 available)  
+ plugin_platform_interface 2.1.8  
source_span 1.10.0 (1.10.1 available)  
stack_trace 1.12.0 (1.12.1 available)  
stream_channel 2.1.2 (2.1.4 available)  
string_scanner 1.3.0 (1.4.1 available)  
term_glyph 1.2.1 (1.2.2 available)  
jiomart > lib > widgets > product_grid.dart  
9:25
```

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.

Authentication

Users

Sign-in method

Templates

Usage

Settings

Extensions

i 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.



Search by email address, phone number, or user UID

Add user



Identifier	Providers	Created	Signed In	User UID
patrasneha...	✉	Mar 1, ...	Mar 1, ...	LY2ICtrJtehz63K...
2022.sneha...	✉	Jan 29, ...	Mar 1, ...	ZdG6m3PSScRL...

Rows per page:

50



1 – 2 of 2



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 with the terminal tab selected. The terminal window displays the output of a PowerShell command, which shows the successful configuration of a Flutter project named 'jiomart' with Firebase. The output includes details about registered platforms (Android, iOS, macOS, Web, Windows) and their corresponding project IDs.

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

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

✓ Which platforms should your configuration support (use arrow keys & space to select)? - android, ios, macos, web, windows
i Firebase android app com.example.jiomart is not registered on Firebase project jiomart-7648d.
i Registered a new Firebase android app on Firebase project jiomart-7648d.
i Firebase ios app com.example.jiomart is not registered on Firebase project jiomart-7648d.
i Registered a new Firebase ios app on Firebase project jiomart-7648d.
i Firebase macos app com.example.jiomart registered.
i Firebase web app jiomart (web) is not registered on Firebase project jiomart-7648d.
i Registered a new Firebase web app on Firebase project jiomart-7648d.
i Firebase windows app jiomart (windows) is not registered on Firebase project jiomart-7648d.
i Registered a new Firebase windows app on Firebase project jiomart-7648d.

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

Platform  Firebase App Id
web      1:1048097925569:web:d26f6745d8efa0c65ba740
android  1:1048097925569:android:17ff95351792b9a6b5ba740
ios       1:1048097925569:ios:da6edde1e2e0030be5ba740
macos    1:1048097925569:ios:da6edde1e2e0030be5ba740
windows  1:1048097925569:web:8b12dfab7dd515605ba740

Learn more about using this file and next steps from the documentation:
> https://firebase.google.com/docs/flutter/setup
PS C:\Users\Govind\AndroidStudioProjects\jiomart>
```

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.

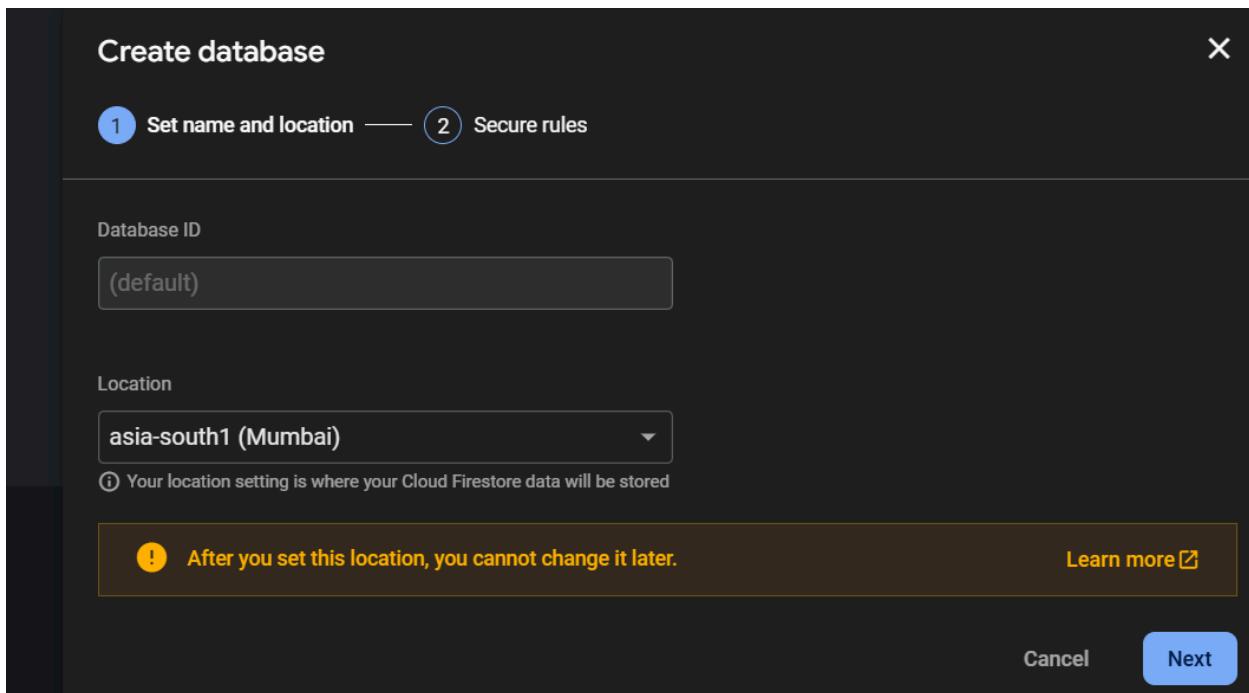
Firestore Database Setup

Step 1:

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

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

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



Create database

Set name and location — **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';

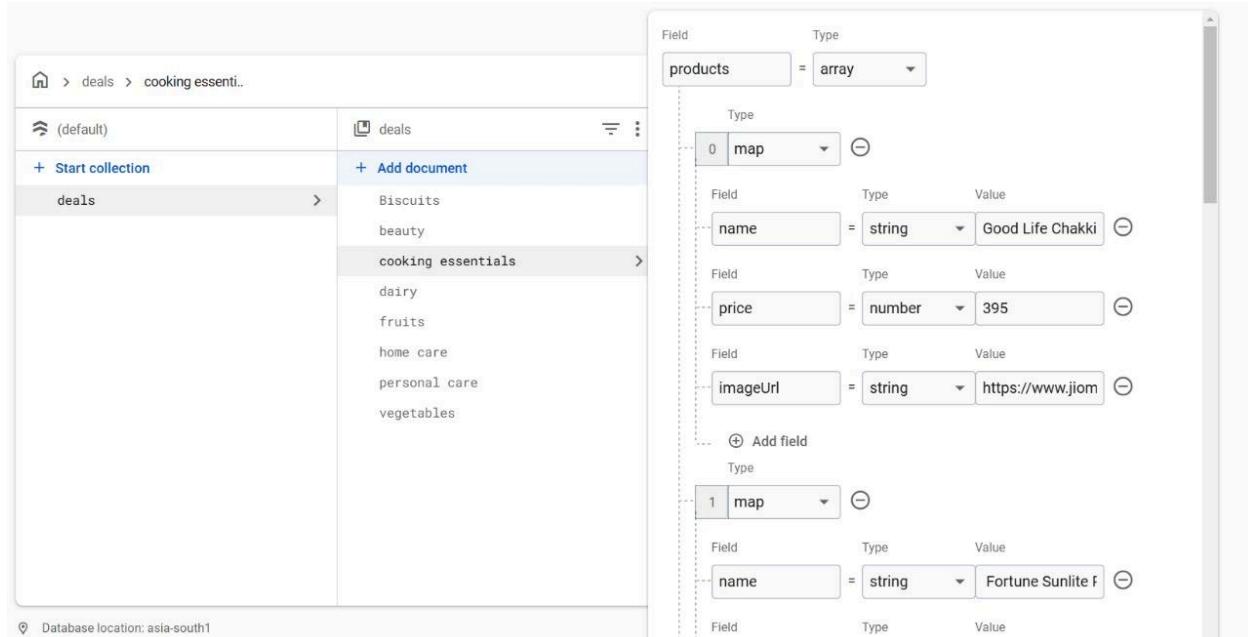
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
        request.time < timestamp.date(2025, 3, 10);
    }
  }
}
```

! 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.



The screenshot shows the Firebase Realtime Database interface. On the left, there's a navigation tree with a root node 'deals' containing sub-nodes like 'Biscuits', 'beauty', 'cooking_essentials', 'dairy', 'fruits', 'home care', 'personal care', and 'vegetables'. A modal window is open on the right, showing the structure of a new document being created under 'cooking_essentials'. The document has a single field named 'products' of type array. Inside the array, there are two map objects. The first map object has fields 'name' (string, value 'Good Life Chakki') and 'price' (number, value 395). The second map object has a field 'name' (string, value 'Fortune Sunlite F'). There are also '+' buttons to add more fields and maps.

```

{
  "products": [
    {
      "name": "Oreo",
      "price": 30,
      "imageUrl": "https://www.jiomart.com/images/product/original/49248951-choco-creme-biscuits-by-cadbury-459-25-g-product-images-0492489516-p590812455-0-202501191721.jpg?im=Resize=(360,360)"
    },
    {
      "name": "Parle-G",
      "price": 25,
      "imageUrl": "https://www.jiomart.com/images/product/original/49000873-g-original-glucose-biscuits-800-g-product-images-0490008739-p490008739-0-202203170454.jpg?im=Resize=(360,360)"
    }
  ]
}

```

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 /deals/{categoryId} { // Matches category documents like
      "biscuits", "dairy"
        allow read, write: if true;

      match /products/{productId} { // Matches products inside each
        category
          allow read, write: if true;
      }
    }
  }
}

```

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

The screenshot shows the Cloud Firestore Rules editor interface. On the left, there's a sidebar with various icons. The main area has tabs for Data, Rules (which is selected), Indexes, Disaster Recovery (NEW), Usage, and Extensions. A button labeled "Develop & Test" is at the top right. Below these, there's a timestamped log entry: "Feb 12, 2025 · 10:45 AM" followed by another entry "Feb 12, 2025 · 10:07 AM". The code editor contains the following Firebase rules:

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /deals/{categoryId} { // Matches category documents like "biscuits", "dairy"
      allow read, write: if true;
    }
    match /products/{productId} { // Matches products inside each category
      allow read, write: if true;
    }
  }
}
```

Code:

```
//category_list.dart
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import './Screens/product_grid.dart';

class CategoryList extends StatelessWidget {
  final List<Map<String, String>> categories = [
    {"name": "Biscuits", "icon": "🍪"},
    {"name": "dairy", "icon": "🥛"},
    {"name": "fruits", "icon": "🍎"},
    {"name": "vegetables", "icon": "🥦"},
    {"name": "cooking essentials", "icon": "🧂"},
    {"name": "babycare", "icon": "👶"},
    {"name": "stationery", "icon": "📝"},
    {"name": "kitchenware", "icon": "🍽️"},
    {"name": "disposables", "icon": "📦"},
  ];
}

// Function to fetch products from Firestore
Future<List<Map<String, dynamic>>> fetchProductsForCategory(String category, {bool limitToThree = false}) async {
  List<Map<String, dynamic>> products = [];
  try {
    DocumentSnapshot categorySnapshot =

```

```
await FirebaseFirestore.instance.collection("deals").doc(category).get();

if (categorySnapshot.exists) {
    var categoryData = categorySnapshot.data() as Map<String, dynamic>;
    if (categoryData.containsKey('products')) {
        var productList = categoryData['products'] as List<dynamic>;
        int takeCount = limitToThree ? 3 : productList.length;

        for (var product in productList.take(takeCount)) {
            products.add({
                'name': product['name'],
                'price': product['price'],
                'imageUrl': product['imageUrl'],
            });
        }
    }
} catch (e) {
    print("Error fetching category products: $e");
}
return products;
}

// Quick Peek Modal (Shows 3 products in bottom sheet)
void showQuickPeek(BuildContext context, String category) async {
    List<Map<String, dynamic>> products = await fetchProductsForCategory(category,
limitToThree: true);

    if (products.isEmpty) {
        ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(content: Text("No products available for $category")),
        );
        return;
    }

    showModalBottomSheet(
        context: context,
        shape: RoundedRectangleBorder(borderRadius: BorderRadius.vertical(top:
Radius.circular(20))),
        builder: (context) {
            return Container(
                padding: EdgeInsets.all(16),
                height: 280,
                child: Column(
```

```
crossAxisAlignment: CrossAxisAlignment.start,
children: [
  Text(
    "Top Picks in $category",
    style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
  ),
  SizedBox(height: 10),
  Expanded(
    child: ListView.separated(
      itemCount: products.length,
      separatorBuilder: (_, __) => Divider(),
      itemBuilder: (context, index) {
        var product = products[index];
        return ListTile(
          leading: Image.network(product['imageUrl'], width: 50, height: 50, fit:
BoxFit.cover),
          title: Text(product['name'], style: TextStyle(fontSize: 16)),
          subtitle: Text("₹${product['price']}"), style: TextStyle(color: Colors.green,
fontWeight: FontWeight.bold)),
        );
      },
    ),
  ),
],
);
},
);
},
);
},
);
},
);
}
}

@Override
Widget build(BuildContext context) {
return Expanded(
child: ListView.builder(
itemCount: categories.length,
itemBuilder: (context, index) {
return GestureDetector(
onTap: () async {
List<Map<String, dynamic>> products = await
fetchProductsForCategory(categories[index]["name"]!);
Navigator.push(
context,
MaterialPageRoute(
builder: (context) => ProductGrid(products: products),

```

```
        ),
    );
},
onLongPress: () {
    showQuickPeek(context, categories[index]["name"]!);
},
child: Container(
    margin: EdgeInsets.symmetric(horizontal: 12, vertical: 8),
    padding: EdgeInsets.all(16),
    decoration: BoxDecoration(
        gradient: LinearGradient(
            colors: [Colors.blueAccent, Colors.lightBlue.shade200],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
        ),
        borderRadius: BorderRadius.circular(15),
        boxShadow: [
            BoxShadow(
                color: Colors.black26,
                blurRadius: 6,
                spreadRadius: 2,
                offset: Offset(2, 4),
            ),
        ],
    ),
),
child: Row(
    children: [
        Text(
            categories[index]["icon"]!,
            style: TextStyle(fontSize: 30),
        ),
        SizedBox(width: 15),
        Expanded(
            child: Text(
                categories[index]["name"]!.toUpperCase(),
                style: TextStyle(
                    color: Colors.white,
                    fontSize: 18,
                    fontWeight: FontWeight.bold,
                ),
            ),
        ),
        Icon(Icons.arrow_forward_ios, color: Colors.white, size: 20),
    ],
),
```

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

//product_grid.dart

import 'package:flutter/material.dart';
import 'cart_page.dart'; // Import cart_page.dart

class ProductGrid extends StatefulWidget {
  final List<Map<String, dynamic>> products;

  ProductGrid({required this.products});

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

class _ProductGridState extends State<ProductGrid> {
  Map<String, int> cartQuantities = {};

  @override
  void initState() {
    super.initState();
    // Initialize quantities from existing cart items
    for (var item in cartItems) {
      cartQuantities[item['name']] = item['quantity'];
    }
  }

  // Function to add/update product in cart
  void _updateCart(Map<String, dynamic> product, int quantity) {
```

```
if (quantity <= 0) {
    // Remove product from cart if quantity is 0
    setState(() {
        cartItems.removeWhere((item) => item['name'] == product['name']);
        cartQuantities[product['name']] = 0;
    });
    return;
}

// Check if product already exists in cart
int existingIndex = cartItems.indexWhere((item) => item['name'] == product['name']);

setState(() {
    if (existingIndex >= 0) {
        // Update existing product quantity
        cartItems[existingIndex]['quantity'] = quantity;
    } else {
        // Add new product to cart
        cartItems.add({
            'name': product['name'],
            'price': product['price'],
            'imageUrl': product['imageUrl'],
            'quantity': quantity,
        });
    }
    // Update local quantity tracker
    cartQuantities[product['name']] = quantity;
});
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text("Products"),
            backgroundColor: Colors.blueAccent,
            actions: [
                Stack(
                    alignment: Alignment.center,
                    children: [
                        IconButton(

```

```
icon: Icon(Icons.shopping_cart, color: Colors.white),
onPressed: () {
    Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => CartPage()),
    ).then((_) {
        // Refresh quantities when returning from cart page
        setState(() {
            for (var product in widget.products) {
                String productName = product['name'];
                int cartIndex = cartItems.indexWhere((item) => item['name'] ==
productName);
                if (cartIndex >= 0) {
                    cartQuantities[productName] = cartItems[cartIndex]['quantity'];
                } else {
                    cartQuantities[productName] = 0;
                }
            }
        });
    },
),
),
if (cartItems.isNotEmpty)
    Positioned(
        right: 8,
        top: 8,
        child: Container(
            padding: EdgeInsets.all(2),
            decoration: BoxDecoration(
                color: Colors.red,
                borderRadius: BorderRadius.circular(10),
            ),
            constraints: BoxConstraints(
                minWidth: 16,
                minHeight: 16,
            ),
            child: Text(
                '${cartItems.length}',
                style: TextStyle(
                    color: Colors.white,
                    fontSize: 10,
                ),
                textAlign: TextAlign.center,
            ),
        ),
    ),
);
```

```
        ),
      ),
    ],
  ),
),
),
body: Padding(
  padding: const EdgeInsets.all(8.0),
  child: GridView.builder(
    gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
      crossAxisCount: 2,
      childAspectRatio: 0.7,
      crossAxisSpacing: 10,
      mainAxisSpacing: 10,
    ),
    itemCount: widget.products.length,
    itemBuilder: (context, index) {
      var product = widget.products[index];
      String productName = product['name'];
      cartQuantities.putIfAbsent(productName, () => 0); // Default quantity 0

      return Card(
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(10),
        ),
        elevation: 3,
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Expanded(
              child: Padding(
                padding: const EdgeInsets.all(8.0),
                child: Image.network(
                  product['imageUrl'],
                  fit: BoxFit.cover,
                ),
              ),
            ),
            Text(
              productName,
              style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
            ),
            Text(

```

```
"₹${product['price']}",
style: TextStyle(fontSize: 14, color: Colors.green),
),
SizedBox(height: 5),

// Quantity Controller (Increase & Decrease Buttons)
Row(
mainAxisAlignment: MainAxisAlignment.center,
children: [
IconButton(
icon: Icon(Icons.remove, color: Colors.red),
onPressed: () {
if (cartQuantities[productName]! > 0) {
_updateCart(product, cartQuantities[productName]! - 1);
}
},
),
Text(
"${cartQuantities[productName]}",
style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
),
IconButton(
icon: Icon(Icons.add, color: Colors.blue),
onPressed: () {
_updateCart(product, cartQuantities[productName]! + 1);
},
),
],
),
],
);
},
),
);
});
```

```
import 'package:flutter/material.dart';
import './screens/CheckoutPage.dart'; // Import CheckoutPage
```

```
// Global cart list to store cart items
List<Map<String, dynamic>> cartItems = [];

class CartPage extends StatefulWidget {
  const CartPage({Key? key}) : super(key: key);

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

class _CartPageState extends State<CartPage> {
  // Function to remove an item from cart
  void _removeItem(int index) {
    setState(() {
      cartItems.removeAt(index);
    });
  }

  // Function to update quantity
  void _updateQuantity(int index, int change) {
    setState(() {
      cartItems[index]['quantity'] += change;
      if (cartItems[index]['quantity'] <= 0) {
        cartItems.removeAt(index);
      }
    });
  }

  // Function to clear cart
  void _clearCart() {
    setState(() {
      cartItems.clear();
    });
  }

  // Function to calculate total price
  double _calculateTotal() {
    return cartItems.fold(0, (sum, item) => sum + (item['price'] * item['quantity']));
  }

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

```
appBar: AppBar(
  backgroundColor: Colors.blueAccent,
  title: const Text(
    "My Cart",
    style: TextStyle(color: Colors.black, fontWeight: FontWeight.bold),
  ),
  iconTheme: const IconThemeData(color: Colors.black),
  actions: [
    if (cartItems.isNotEmpty)
      IconButton(
        icon: const Icon(Icons.delete, color: Colors.black),
        onPressed: _clearCart,
      ),
  ],
),
body: cartItems.isEmpty
  ? const Center(
    child: Text("Your cart is empty!", style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold)),
  )
  : Column(
    children: [
      Expanded(
        child: ListView.builder(
          itemCount: cartItems.length,
          itemBuilder: (context, index) {
            return _buildCartItem(index);
          },
        ),
      ),
      _buildTotalSection(),
    ],
  );
);

}

Widget _buildCartItem(int index) {
  var item = cartItems[index];
  return Padding(
    padding: const EdgeInsets.symmetric(horizontal: 10, vertical: 5),
    child: Card(
      shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(10)),
      elevation: 2,
      child: Padding(
```

```
padding: const EdgeInsets.all(8.0),
child: Row(
  children: [
    ClipRRect(
      borderRadius: BorderRadius.circular(10),
      child: Image.network(
        item['imageUrl'],
        width: 80,
        height: 80,
        fit: BoxFit.cover,
        errorBuilder: (context, error, stackTrace) =>
            Image.asset("assets/fallback_image.png", width: 80, height: 80),
      ),
    ),
    const SizedBox(width: 10),
    Expanded(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(
            item['name'],
            style: const TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
            overflow: TextOverflow.ellipsis,
          ),
          const SizedBox(height: 5),
          Text(
            "₹${item['price']}",
            style: const TextStyle(fontSize: 14, color: Colors.green),
          ),
          const SizedBox(height: 5),
        ],
      ),
    ),
    Row(
      children: [
        IconButton(
          icon: const Icon(Icons.remove_circle, color: Colors.red),
          onPressed: () => _updateQuantity(index, -1),
        ),
        Text(
          "${item['quantity']}",
          style: const TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
        ),
        IconButton(
          icon: const Icon(Icons.add_circle, color: Colors.green),
          onPressed: () => _updateQuantity(index, 1),
        ),
      ],
    ),
  ],
),
```

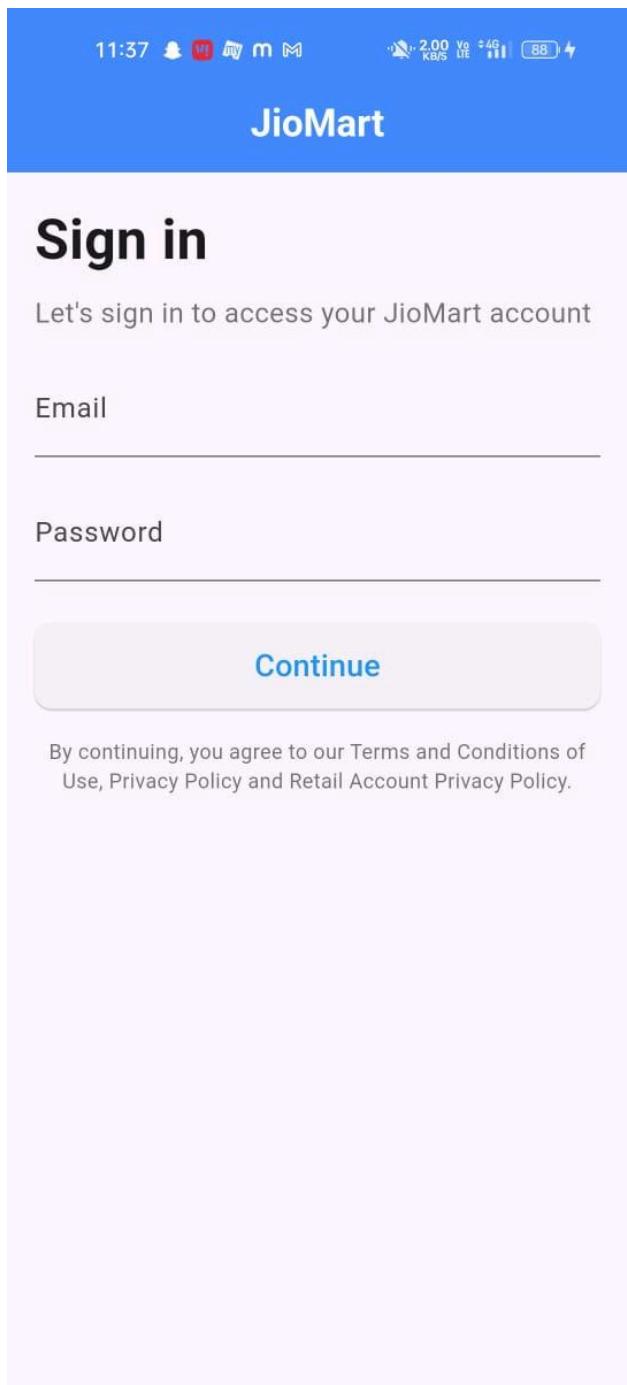
```
        ],
        ),
        ],
        ),
        ),
        IconButton(
        icon: const Icon(Icons.delete, color: Colors.red),
        onPressed: () => _removeItem(index),
        ),
        ],
        ),
        ),
        );
    );
}

Widget _buildTotalSection() {
    return Container(
        padding: const EdgeInsets.all(16.0),
        decoration: BoxDecoration(
            color: Colors.white,
            boxShadow: [
                BoxShadow(color: Colors.black12, blurRadius: 5, spreadRadius: 2),
            ],
        ),
        child: Column(
            children: [
                Row(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,
                    children: [
                        const Text(
                            "Total:",
                            style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
                        ),
                        Text(
                            "₹${_calculateTotal().toStringAsFixed(2)}",
                            style: const TextStyle(fontSize: 18, fontWeight: FontWeight.bold, color: Colors.green),
                        ),
                    ],
                ),
                const SizedBox(height: 10),
                SizedBox(
                    width: double.infinity,
```

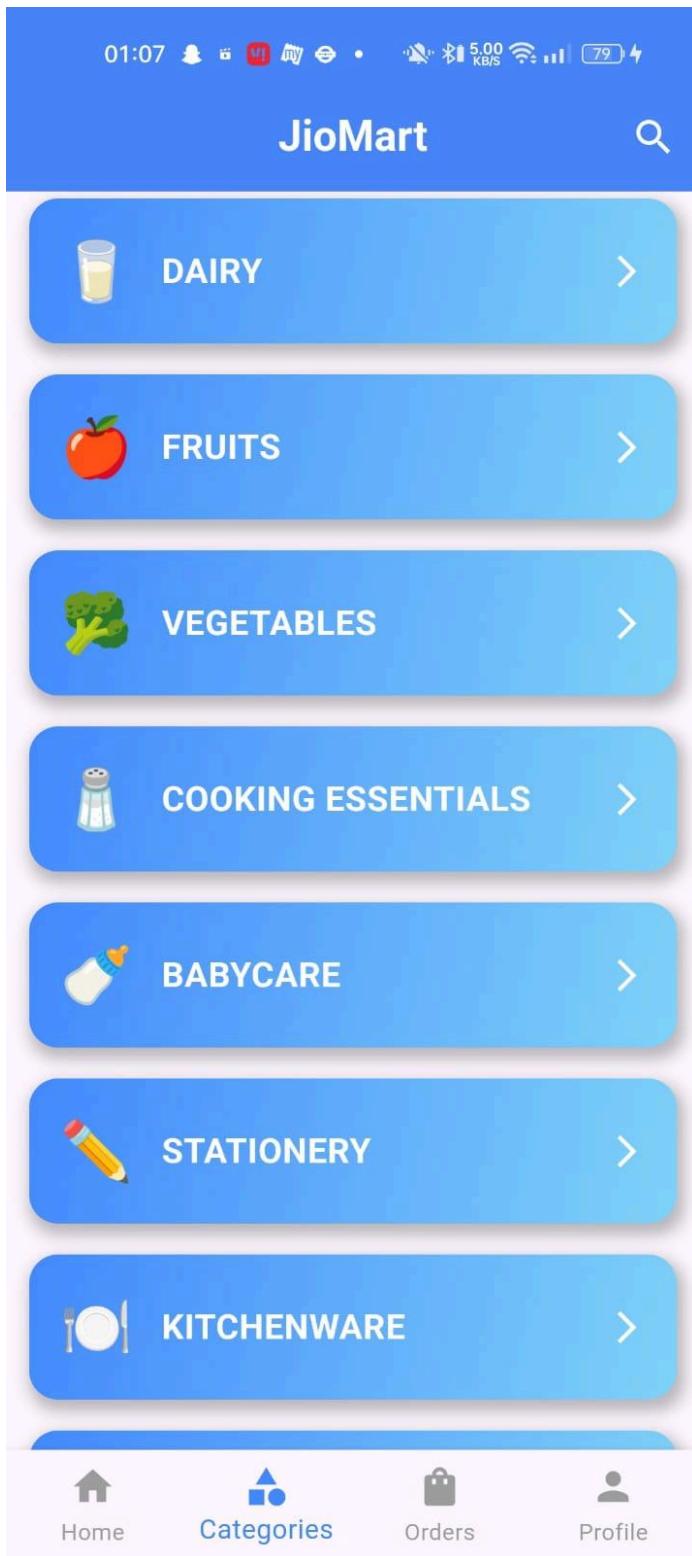
```
        child: ElevatedButton(
            style: ElevatedButton.styleFrom(
                backgroundColor: Colors.blue,
                shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(30)),
            ),
            onPressed: () {
                // Navigate to CheckoutPage
                Navigator.push(
                    context,
                    MaterialPageRoute(builder: (context) => const CheckoutPage()),
                );
            },
            child: const Padding(
                padding: EdgeInsets.symmetric(vertical: 12),
                child: Text(
                    "Proceed to Checkout",
                    style: TextStyle(fontSize: 16, color: Colors.white),
                ),
            ),
        ),
    ],
),
);
}
}
```

Output:

Login Page:



Category Page:



Product List Page:

01:17 66.0 KB/S 80

Products 2



French Beans 500 g
₹25

- 0 +



Potato 1 kg
₹32

- 0 +



Onion 1 kg
₹46

- 0 +



Green Capsicum 500 g
₹27

- 0 +



Tomato Country 1 kg (Pack)
₹32



Cauliflower 1 pc (Approx 600 g)
₹28

01:16



•



16.0



KB/S



80



Products



**Amul Pasteurised
Butter 100 g**

₹56

- 0 +



**Amul Taaza Toned
Milk 1 L**

₹71

- 0 +



**Hazelnut Chocolate
Flavoured Milk 80 ml**

₹10

- 0 +



**Amul Fresh Cream
250 ml**

₹64

- 0 +



**Amul Masti Spiced
Buttermilk 200 ml**

₹14



**Amul Masti Dahi 200
g**

₹22

01:17 • 53.0 KB/S 80

← Products 2

 Good Life Chakki Atta 10 kg ₹395 - 0 +	 Fortune Sunlite Refined Sunflower Oil 1 L ₹157 - 0 +
 Good Life Pure Crystal Sugar (M) 5 kg ₹239 - 0 +	 Fortune Suji / Semolina 500 g ₹26 - 0 +
 Good Life Free Flow Iodised Salt 1 kg ₹14 - 0 +	 Good Life Jeera 500 g ₹185 - 0 +

Cart Page:

01:08 Snapchat 2.00 KB/S 79%

← My Cart



Oreo Pokemon Cream...

₹30



- 1 +



Apple Red Delicious 4 ...

₹189



- 1 +

Total: ₹219.00

Proceed to Checkout

Checkout page:

01:08 3.00 KB/S 79%

← Checkout Page

Delivery Address

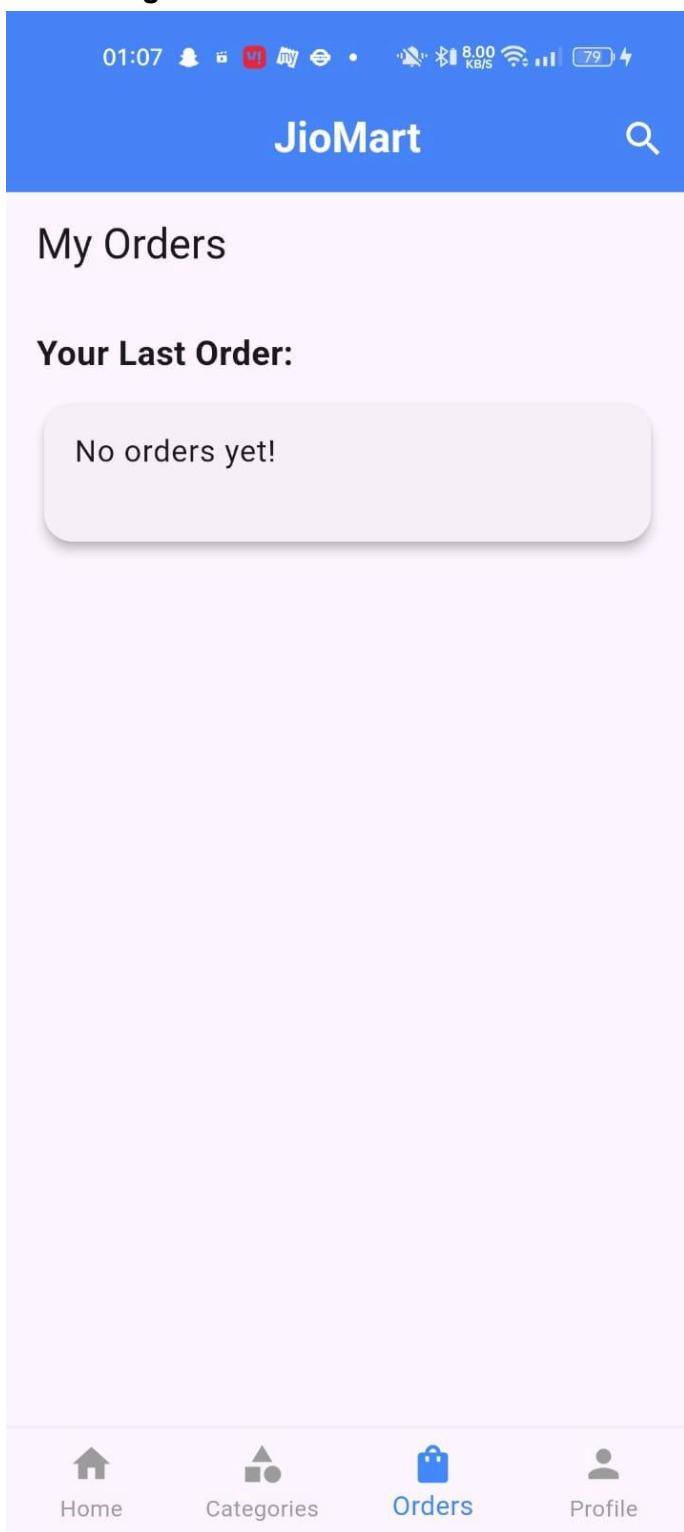
ion, Mumbai, Maharashtra 400022, India

Payment Method

Cash on Delivery ▾

Place Order

Orders Page:



Unique Feature:"Quick Peek" (Instant Product Preview)

Instead of navigating to a new screen, users can long-press a category to instantly preview a few top products in a small pop-up (bottom sheet).

How It Works?

- **Tap on a category** → Goes to the full product grid (as before).
- **Long-press on a category** → Opens a Quick Peek (bottom sheet) showing the top 3 products from that category without leaving the page.

