Vivekanand Education Society's Institute of Technology

An Autonomous Institute Affiliated to University of Mumbai

Hashu Advani Memorial Complex, Collector Colony, Chembur East, Mumbai - 400074.

Department of Information Technology

CERTIFICATE

This is to certify that Snehal Anilkumar Patil of **D15A/D15B** semester **VI,** have successfully completed necessary experiments in the **MAD & PWA Lab** under my supervision in **VES Institute of Technology** during the academic year **2024-2025.**

Lab Assistant

Subject Teacher

Mrs. Kajal Joseph

Principal

Head of Department

Dr. Mrs. Shalu Chopra

Project Title: **Roll No.**

Name of the Course : MAD & PWA Lab **Course Code :** ITL604

Year/Sem/Class : D15A/D15B **A.Y.:** 24-25

Faculty Incharge : Mrs. Kajal Joseph.

Lab Teachers : Mrs. Kajal Joseph.

Email : kajal.jewani@ves.ac.in

Programme Outcomes: The graduate will be able to:

PO1) Basic Engineering knowledge: An ability to apply the fundamental knowledge in mathematics, science and engineering to solve problems in Computer engineering.

PO2) Problem Analysis: Identify, formulate, research literature and analyze computer engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and computer engineering and sciences.

PO3) Design/ Development of Solutions: Design solutions for complex computer engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal and environmental considerations.

PO4) Conduct investigations of complex engineering problems using research-based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of information to provide valid conclusions.

PO5) Modern Tool Usage: Create, select and apply appropriate techniques, resources and modern computer engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6) The Engineer and Society: Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to computer engineering practice.

PO7) Environment and Sustainability: Understand the impact of professional computer engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

PO8) Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of computer engineering practice.

PO9) Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.

PO10) Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.

PO11) Project Management and Finance: Demonstrate knowledge and understanding of computer engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12) Life-long Learning: Recognize the need for and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Program specific Outcomes

PSO1) An ability to manage and analyze data / information effectively for making better decisions.

PSO2) Demonstrate the ability to use state of the art technologies and tools including Free and Open Source Software (FOSS) tools in developing software.

Project Title: **Roll No.**

Lab Objectives:

Sr. No.	Lab Objectives
The Lab experiments aims:	
1	Learn the basics of the Flutter framework.
2	Develop the App UI by incorporating widgets, layouts, gestures and animation
3	Create a production ready Flutter App by including files and firebase backend service.
4	Learn the Essential technologies, and Concepts of PWAs to get started as quickly and efficiently as possible
5	Develop responsive web applications by combining AJAX development techniques with the jQuery JavaScript library.
6	Understand how service workers operate and also learn to Test and Deploy PWA.

Lab Outcomes:

Sr. No.	Lab Outcomes	Cognitive levels of attainment as per Bloom's Taxonomy
On Completion of the course the learner/student should be able to:		
1	Understand cross platform mobile application development using Flutter framework	L1, L2
2	Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation	L3
3	Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS	L3, L4
4	Understand various PWA frameworks and their requirements	L1, L2
5	Design and Develop a responsive User Interface by applying PWA Design techniques	L3
6	Develop and Analyse PWA Features and deploy it over app hosting solutions	L3, L4

Project Title: **Roll No.**

Index

Sr. No	Experiment Title	LO	DOP	DOS	Grade
1.	To install and configure the Flutter Environment	LO1			
2.	To design Flutter UI by including common widgets.	LO2			
3.	To include icons, images, fonts in Flutter app	LO2			
4.	To create an interactive Form using form widget	LO2			
5.	To apply navigation, routing and gestures in Flutter App	LO2			
6.	To Connect Flutter UI with fireBase database	LO3			
7.	To write meta data of your Ecommerce PWA in a Web app manifest file to enable “add to homescreen feature”.	LO4			
8.	To code and register a service worker, and complete the install and activation process for a new service worker for the E-commerce PWA	LO5			
9.	To implement Service worker events like fetch, sync and push for E-commerce PWA	LO5			
10.	To study and implement deployment of Ecommerce PWA to GitHub Pages.	LO5			
11.	To use google Lighthouse PWA Analysis Tool to test the PWA functioning.	LO6			
12.	Assignment-1	LO1,LO2 ,LO3			
13.	Assignment-2	LO4,LO5 ,LO6			

Project Title: **Roll No.**

MAD & PWA Lab
Journal

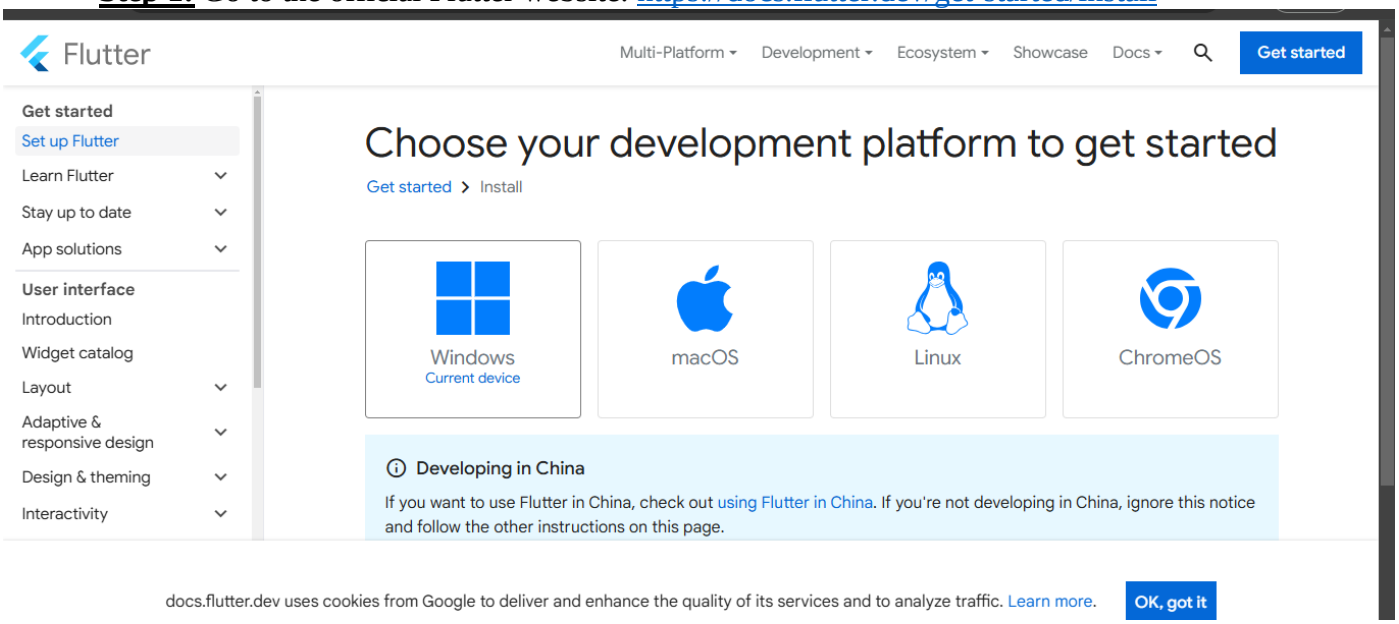
Experiment No.	01
Experiment Title.	To install and configure the Flutter Environment
Roll No.	38
Name	Snehal Anilkumar Patil
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO1: Understand cross platform mobile application development using Flutter framework
Grade:	

EXPERIMENT NO: - 01

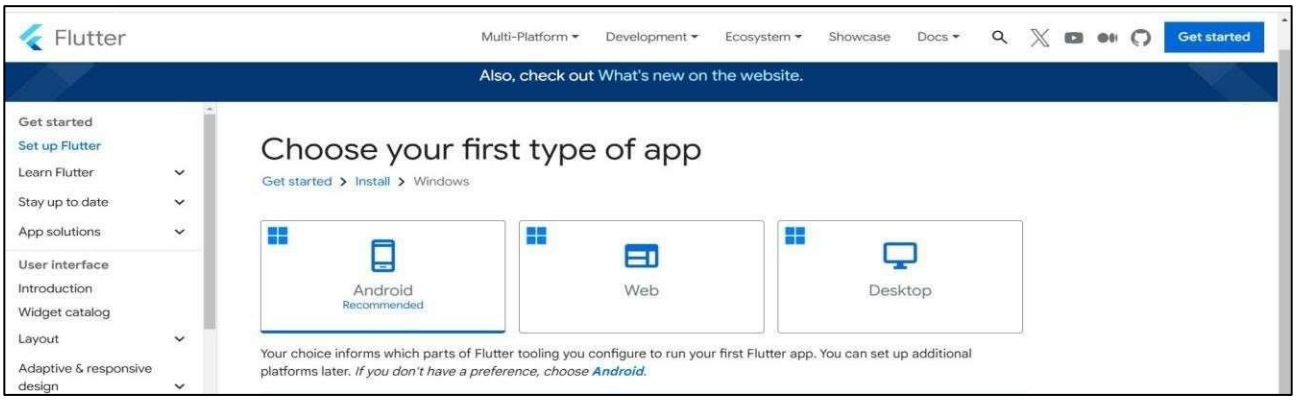
Name:- Snehal Patil **Class:-** D15A **Roll:No: -** 38

AIM: - Installation and Configuration of Flutter Environment.

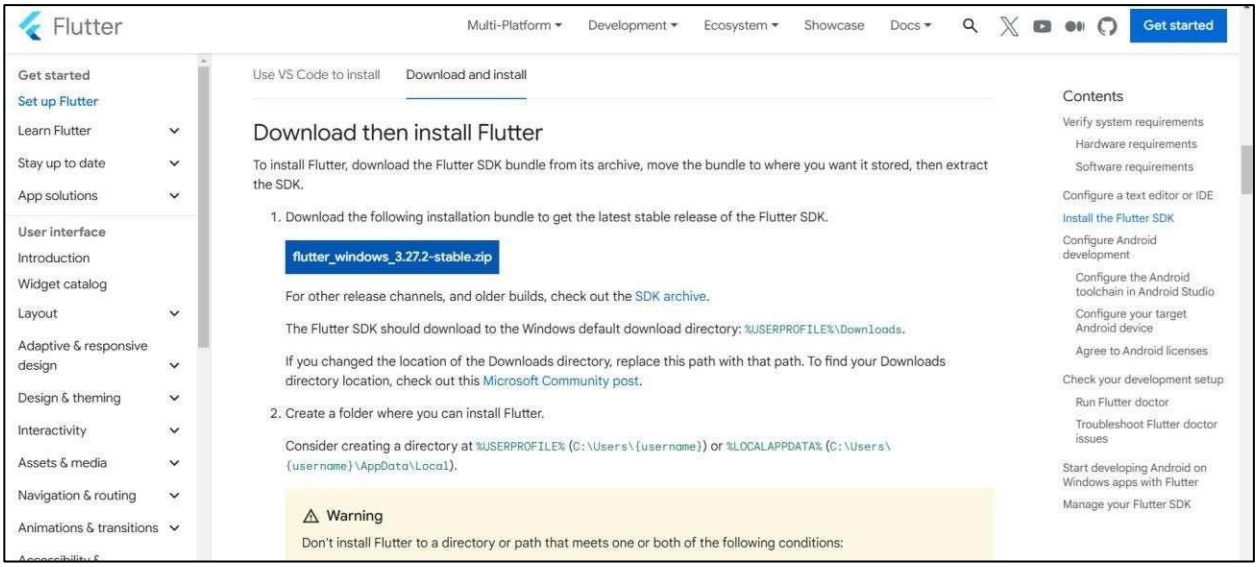
Step 1: Go to the official Flutter website: https://docs.flutter.dev/get-started/install



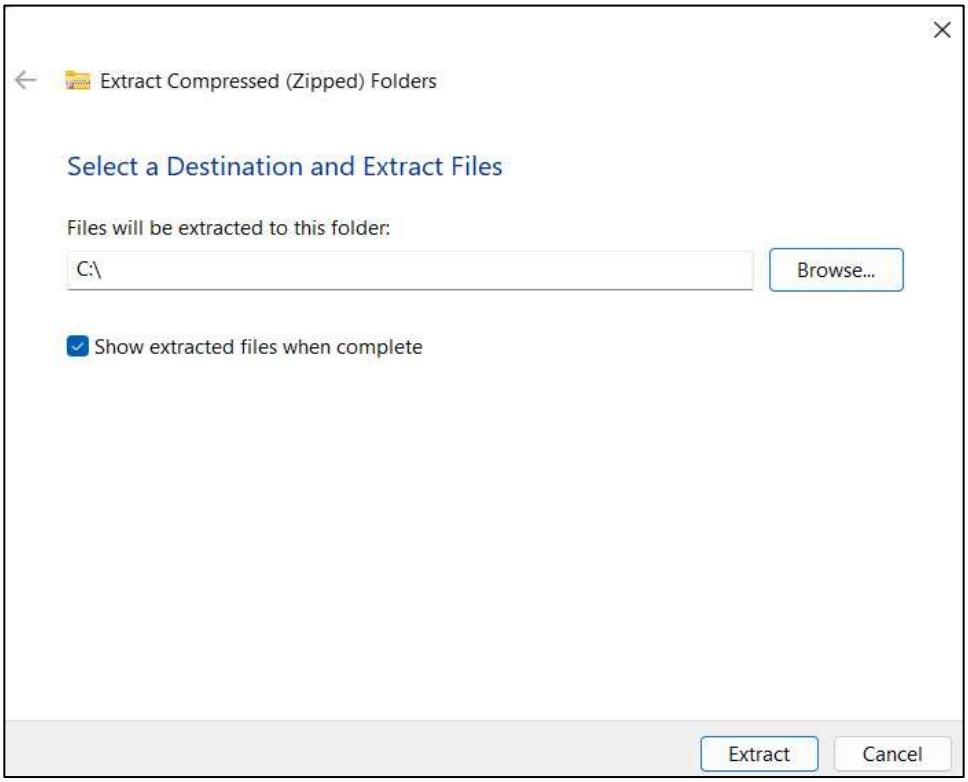
Step 2: To download the latest Flutter SDK, click on the Windows icon > Android



Step 3: For Windows, download the stable release (a .zip file).



Step 4: Extract the ZIP file to a folder (e.g., C:\flutter).

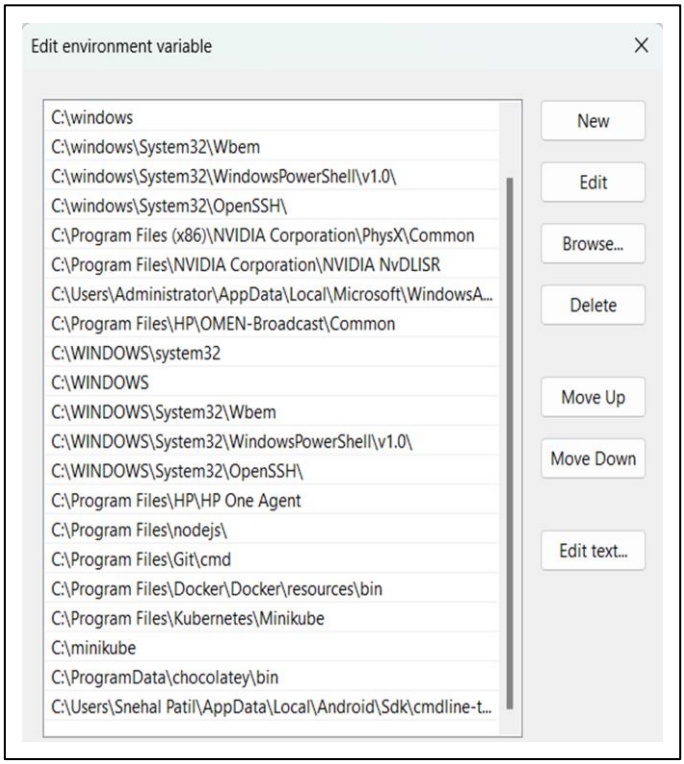


Step 5 :- Add Flutter to System PATH

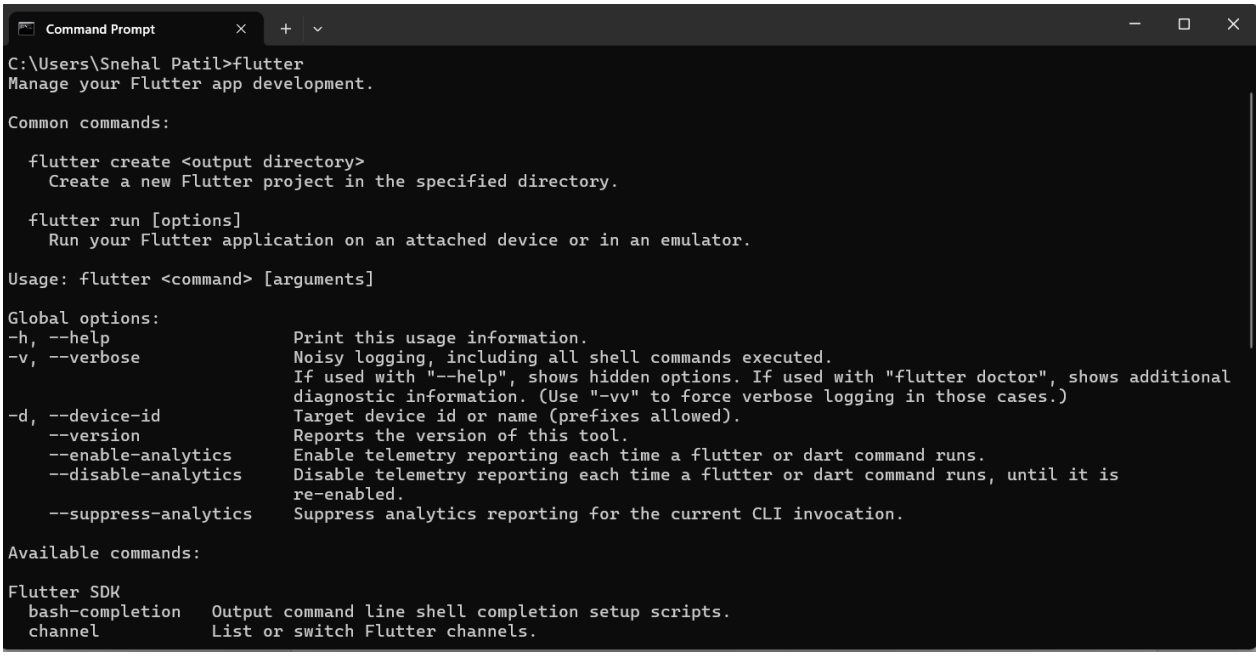
Right-click on the Start Menu > System > Advanced system settings > Environment Variables.

Under System Variables, find Path and click Edit.

Add the full path to the flutter/bin directory (e.g., C:\flutter\bin).



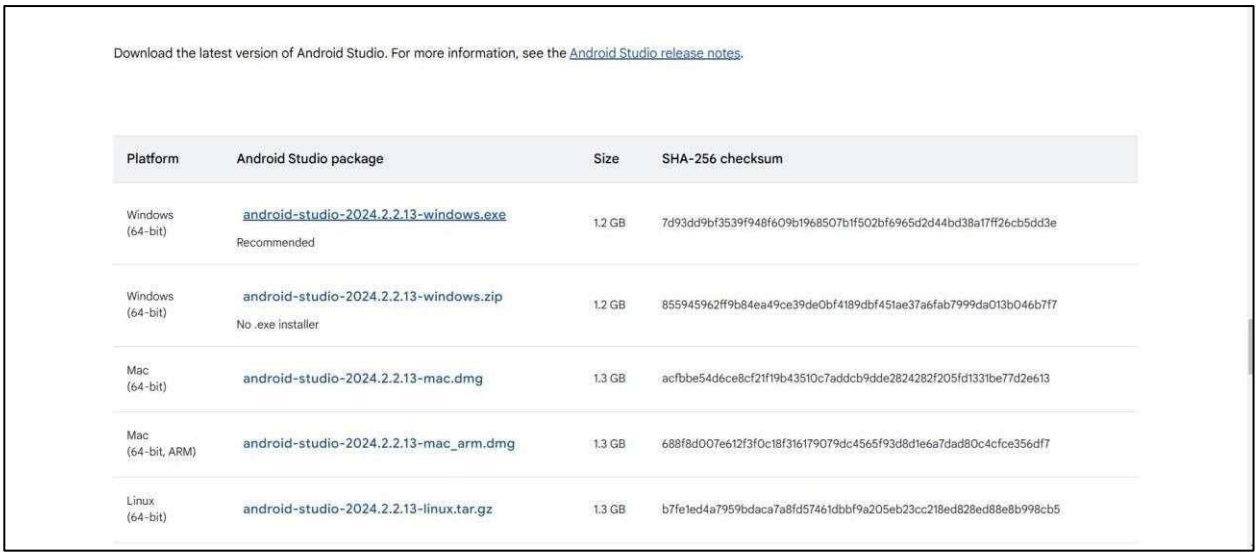
Step 6 : - Now, run the $ flutter command in command prompt.



Step 7:- Run the $ flutter doctor command. This command checks for all the requirements of Flutter app development and displays a report of the status of your Flutter installation

```
C:\Users\Snehal Patil>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, 3.27.3, on Microsoft Windows [Version 10.0.22631.4751], locale en-IN)
[√] Windows Version (Installed version of Windows is version 10 or higher)
[√] Android toolchain - develop for Android devices (Android SDK version 35.0.1)
[√] Chrome - develop for the web
[X] 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 2024.2)
[√] VS Code (version 1.96.4)
[√] Connected device (3 available)
[√] Network resources

! Doctor found issues in 1 category.

C:\Users\Snehal Patil>
```

Step 8 : - Go to Android Studio and download the installer.

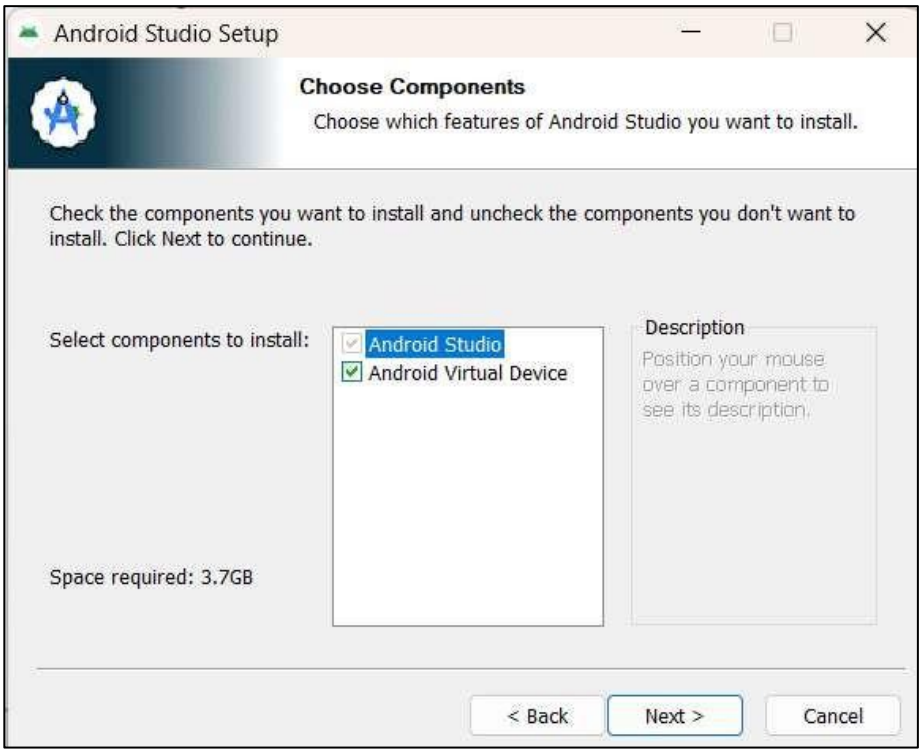


Download the latest version of Android Studio. For more information, see the Android Studio release notes.

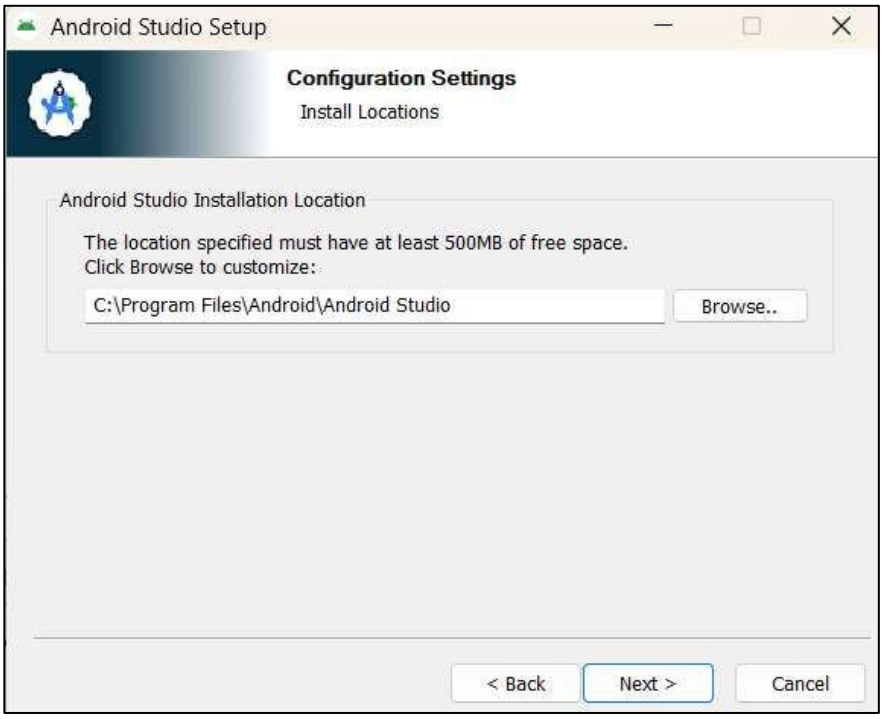
Platform	Android Studio package	Size	SHA-256 checksum
Windows (64-bit)	android-studio-2024.2.2.13-windows.exe Recommended	1.2 GB	7d93dd9bf3539f948f609b1968507b1f502bf6965d2d44bd38a17ff26cb5dd3e
Windows (64-bit)	android-studio-2024.2.2.13-windows.zip No .exe installer	1.2 GB	855945962ff9b84ea49ce39de0bf4189dbf451ae37a6fab7999da013b046b7f7
Mac (64-bit)	android-studio-2024.2.2.13-mac.dmg	1.3 GB	acfbbe54d6ce8cf21f19b43510c7addcb9dde2824282f205fd1331be77d2e613
Mac (64-bit, ARM)	android-studio-2024.2.2.13-mac_arm.dmg	1.3 GB	688f8d007e612f3f0c18f316179079dc4565f93d8d1e6a7dad80c4cfce356df7
Linux (64-bit)	android-studio-2024.2.2.13-linux.tar.gz	1.3 GB	b7fe1ed4a7959bdaca7a8fd57461dbbf9a205eb23cc218ed828ed88e8b998cb5

Step 8.1: - When the download is complete, open the .exe file and run it. You will get the following dialog box

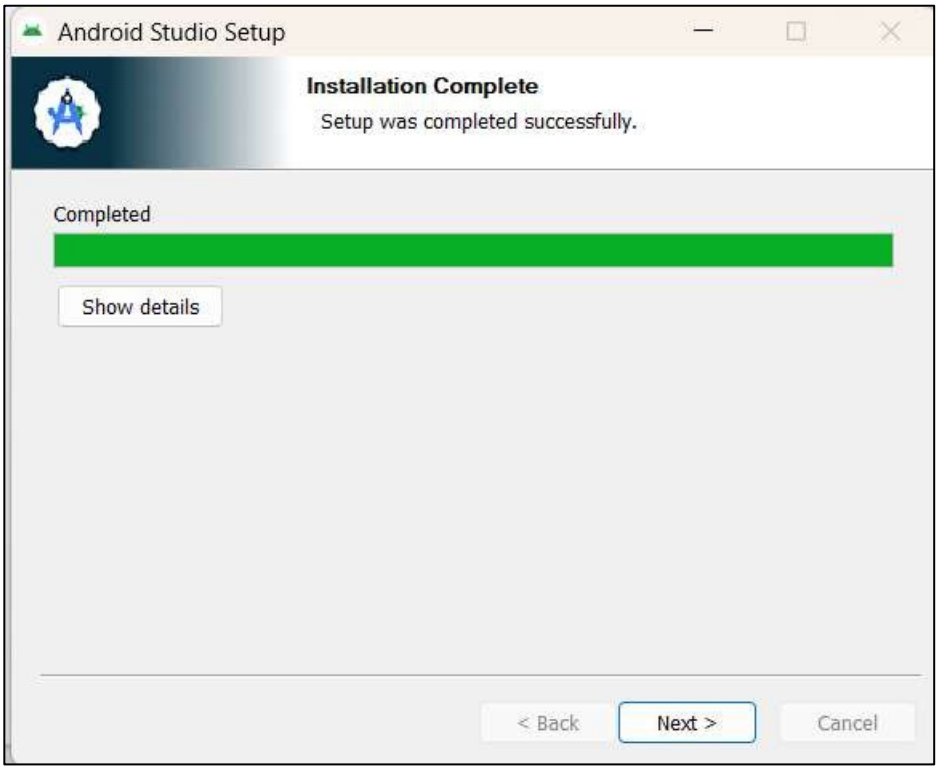
Step 8.2: - Select all the Checkboxes and Click on 'Next' Button.

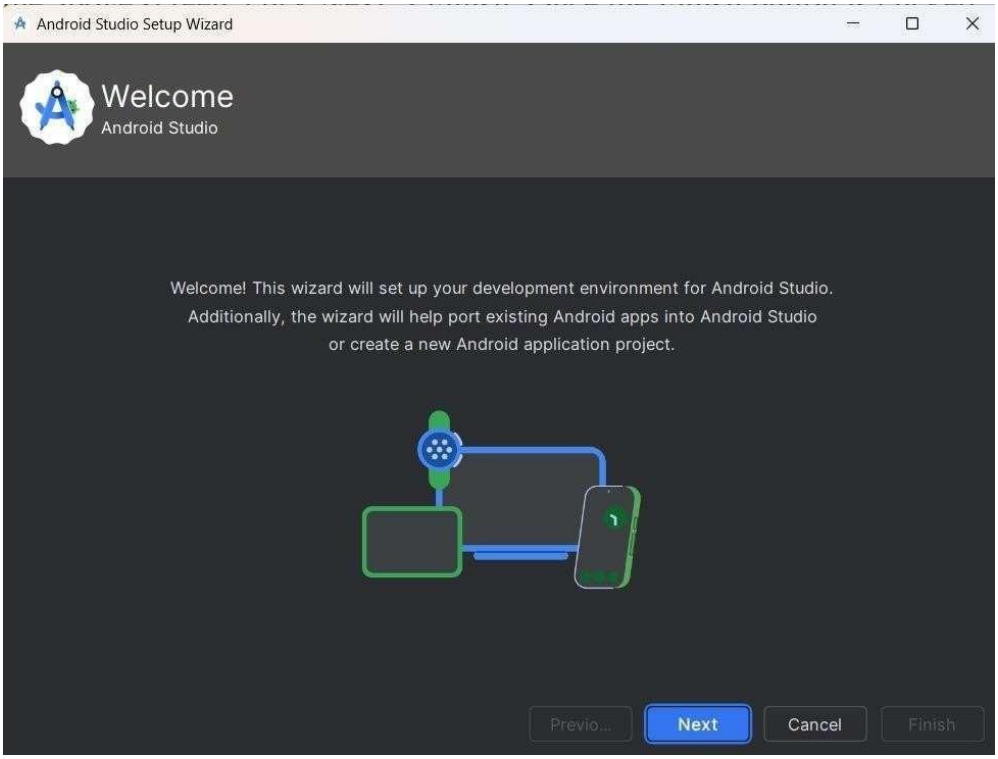


Step 8.3: - Change the destination as per your convenience and click on ‘Next’ Button.

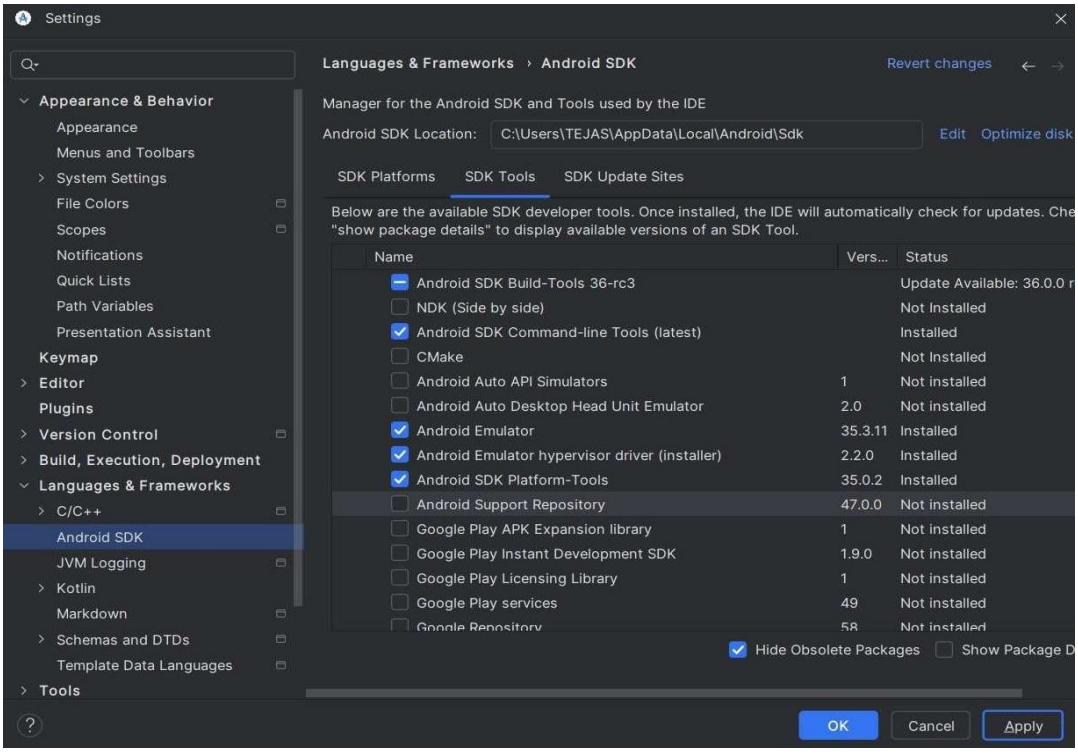


Step 8.4: - Follow the steps of the installation wizard. Once the installation wizard completes, you will get the following screen.

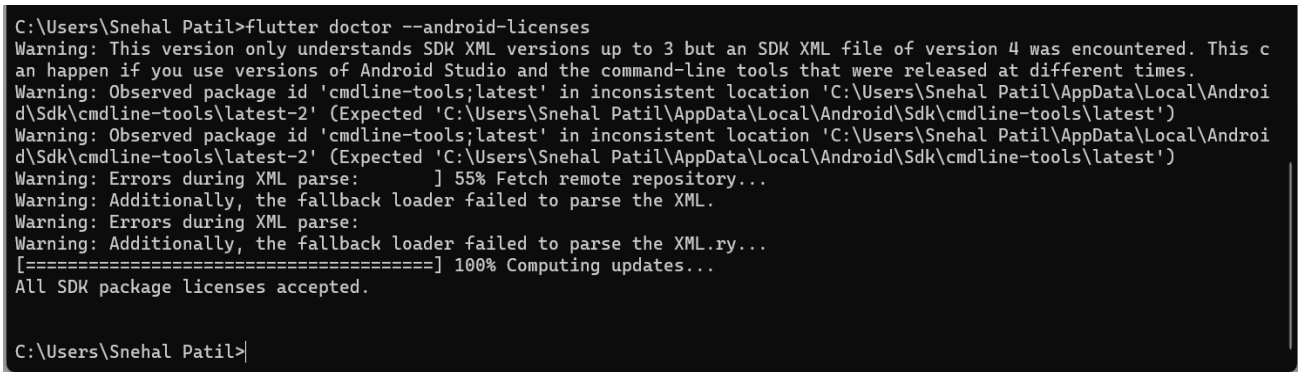




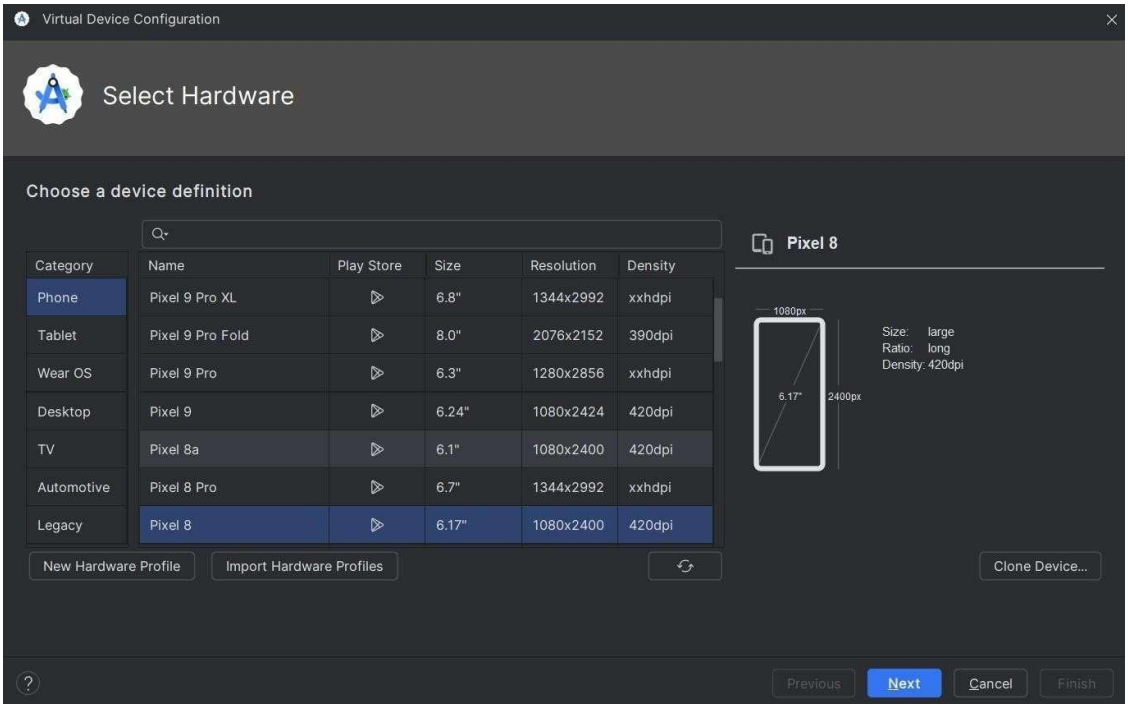
Step 8.5: - Go to Preferences > Appearance & Behavior > System Settings > Android SDK. Select the SDK Tools tab and check Android SDK Command-line Tools and Install it.



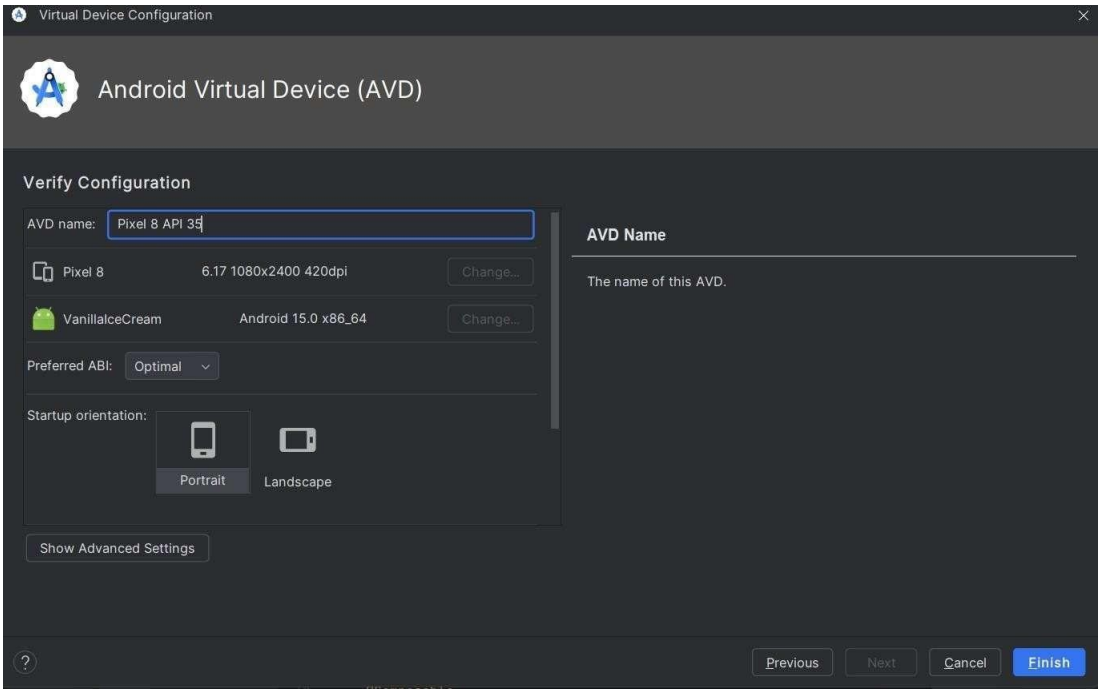
Step 9: - Open a terminal and run the following command



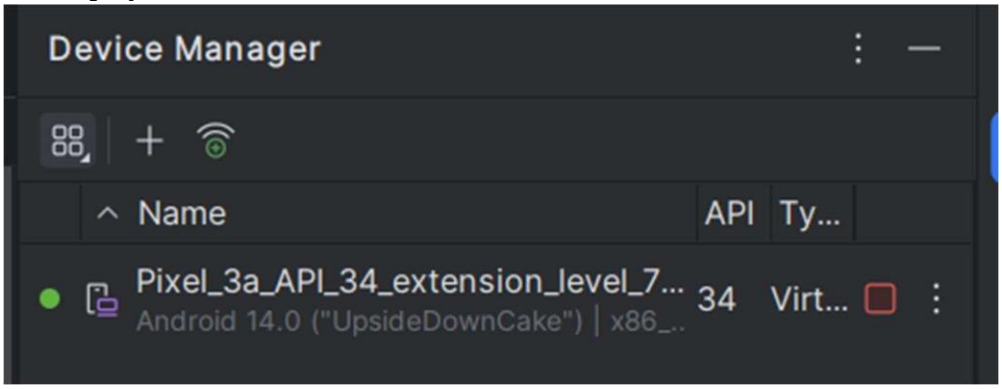
Step 10: - Next, you need to set up an Android emulator. It is responsible for running and testing the Flutter application



Step 10.1: - Open Android Studio and go to Tools > AVD Manager. Create a new virtual device.

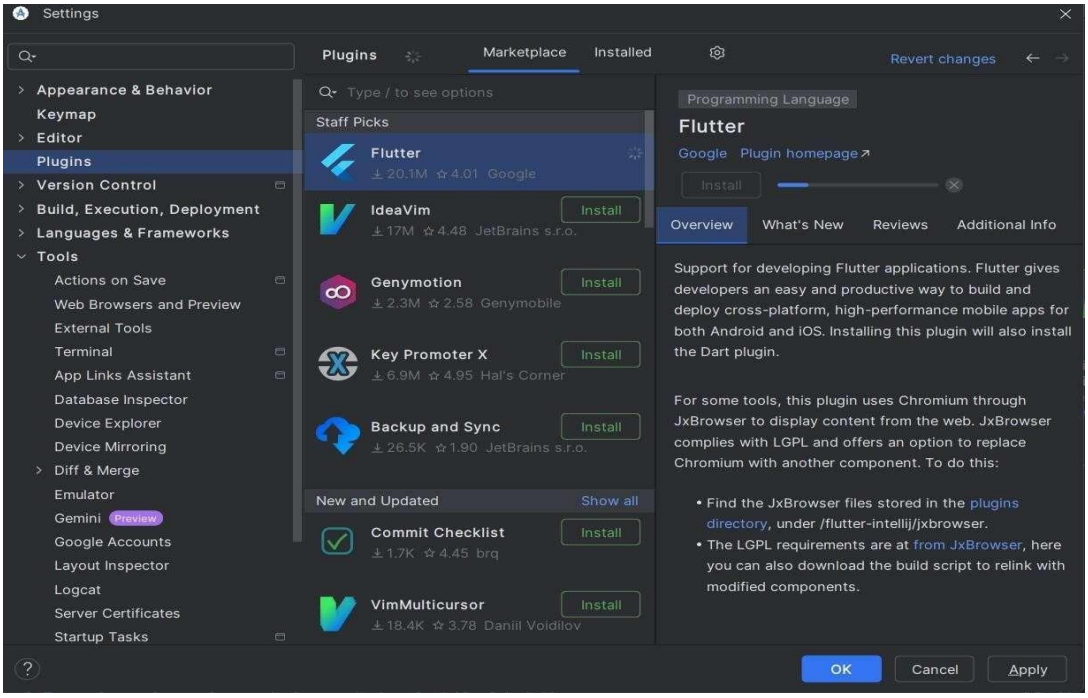


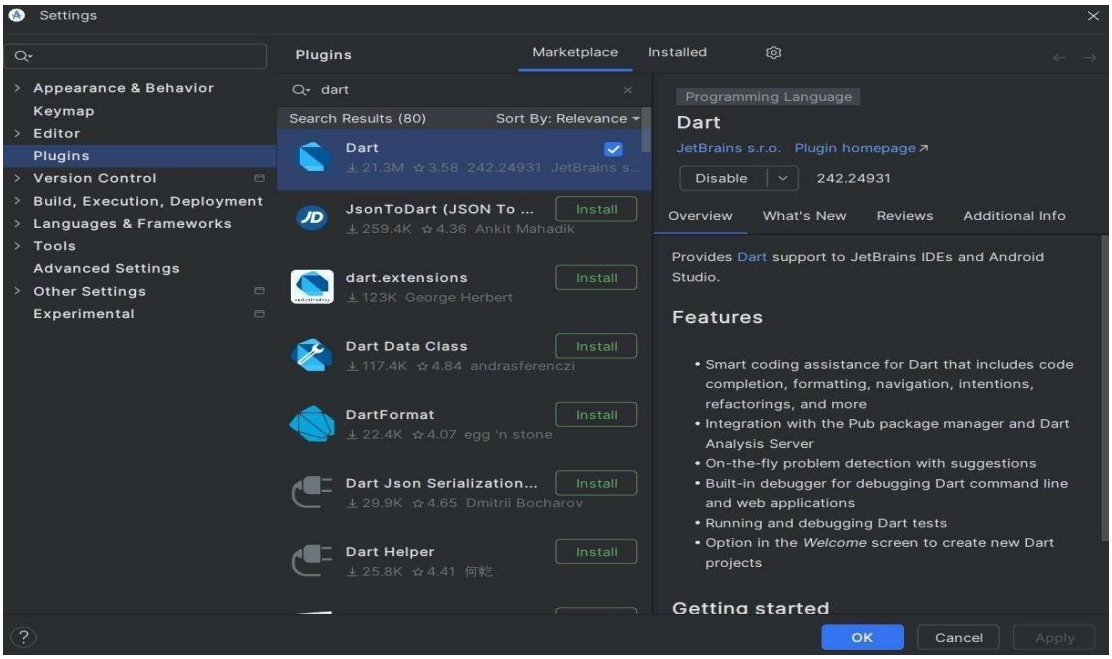
Step 10.2: - Click on the icon pointed into the red color rectangle. The Android emulator displayed as below screen



Step 11: - Now, install Flutter and Dart plugin for building Flutter application in Android Studio. These plugins provide a template to create a Flutter application, give an option to run and debug Flutter application in the Android Studio itself

Step 11.1: - Open the Android Studio and then go to File->Settings->Plugins. Now, search the Flutter plugin. If found, select Flutter plugin and click install





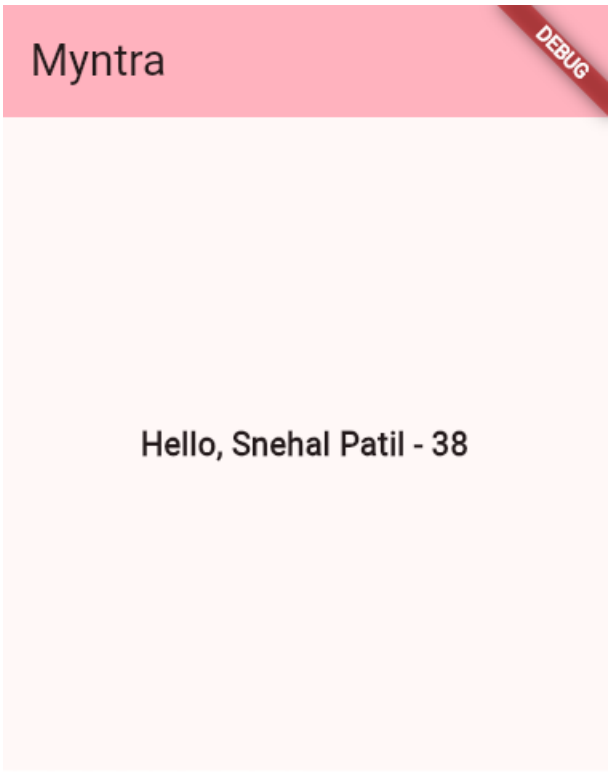
Step 11.2: - Restart the Android Studio

Step 12: - Go to File > New Project > Create Flutter Project, then select the project name and location, and click Next to proceed

Code : -

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.pink),
        useMaterial3: true,
      ),
      home: const MyHomePage(title: 'Myntra'), // Title remains 'Myntra'
    );
  }
}

class MyHomePage extends StatelessWidget {
  const MyHomePage({super.key, required this.title});
  final String title;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: Text(title),
      ),
      body: const Center(
        child: Text(
          'Hello, Snehal Patil - 38',
          style: TextStyle(fontSize: 16.0, fontWeight: FontWeight.bold),
        ),
      ),
    );
  }
}
```

OUTPUT:



Project Title: **Roll No.**

MAD & PWA Lab
Journal

Experiment No.	02
Experiment Title.	To design Flutter UI by including common widgets.
Roll No.	38
Name	Snehal Anilkumar Patil
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

Snehal Patil D15A 38

Experiment 2: Designing the UI for Myntra App in Flutter

Aim:
To design a Flutter UI for the Myntra App, including a **Login Page, Sign-up Page, Home Page, and Categories Page**, using common Flutter widgets.

Theory:
Flutter provides a wide range of widgets to build UI components. The key widgets used in this Myntra App include:

- **Scaffold**: Provides the basic layout structure.
- **SafeArea**: Ensures content is within safe display boundaries.
- **Container**: Used for styling elements with colors, images, or gradients.
- **Column & Row**: Used for structuring UI components in vertical and horizontal arrangements.
- **TextField**: Allows user input.
- **ElevatedButton**: Used for clickable buttons.
- **GridView**: Used for displaying lists of categories and products in a grid format.
- **BottomNavigationBar**: Provides navigation at the bottom.
- **AppBar**: Displays the top navigation bar.
- **ClipRRect**: Rounds the corners of images.

Common Widgets in Each Page

1. **Login Page**

Code:

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

class LoginPage extends StatelessWidget {
  final TextEditingController mobileController = TextEditingController();

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

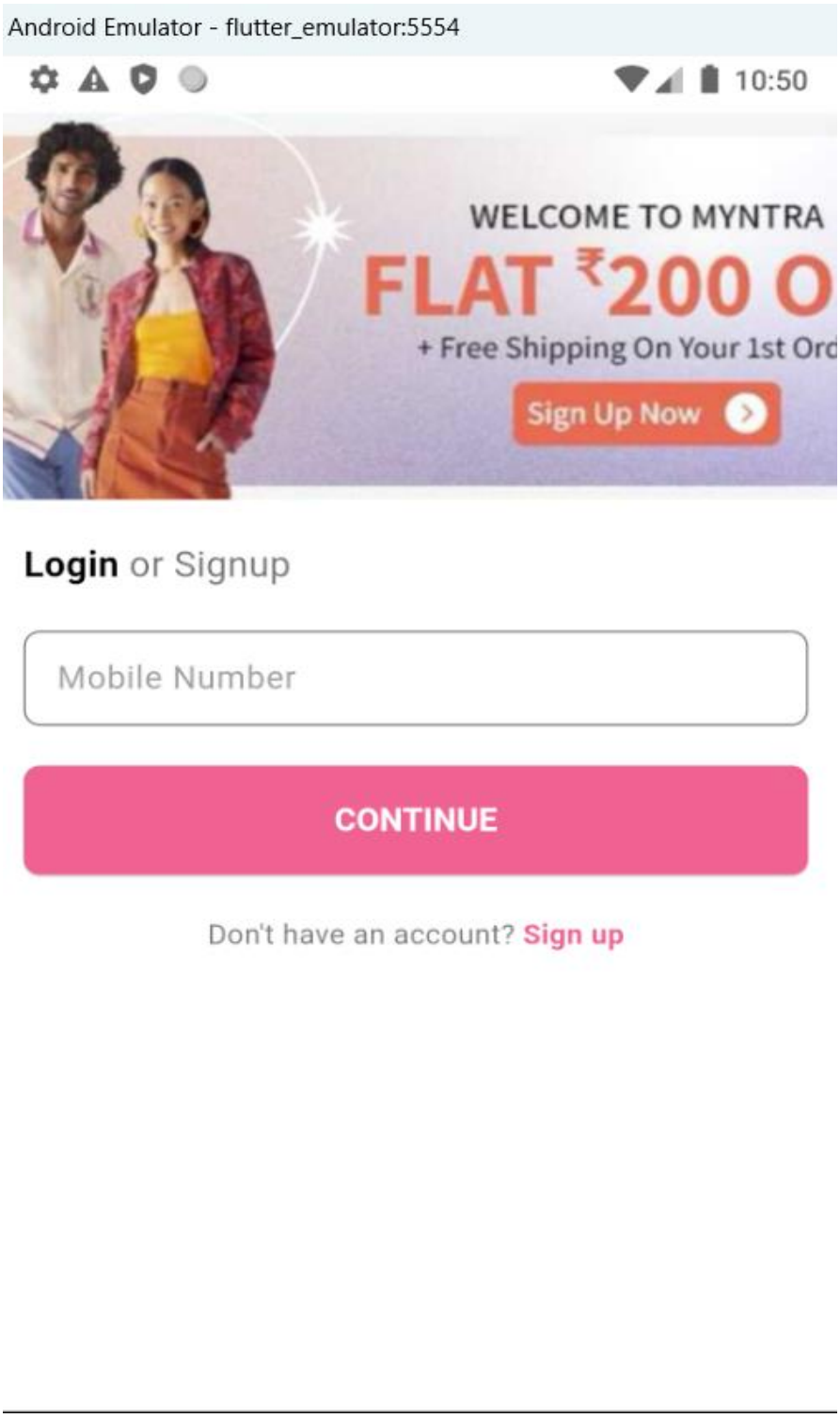
```
body: SafeArea(
  child: Column(
    children: [
      // Full Banner Section with Image
      Container(
        width: double.infinity,
        height: 200,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [Colors.yellow.shade100, Colors.pink.shade100],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
        ),
        child: Image.asset(
          'assets/images/shopping_cart.png', // Replace with your image
          fit: BoxFit.cover, // Ensures the image covers the entire container
        ),
      ),

      const SizedBox(height: 20),

      // Login or Signup Text
      Padding(
        padding: const EdgeInsets.symmetric(horizontal: 16.0),
        child: Row(
          children: [
            Text(
              "Login ",
              style: TextStyle(
                fontSize: 18,
                fontWeight: FontWeight.bold,
                color: Colors.black,
              ),
            ),
            Text(
```

```
        "or Signup",
        style: TextStyle(
          fontSize: 18,
          color: Colors.black54,
        ),
      ),
    ],
  ),
),

const SizedBox(height: 20),

// Input Field
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: TextField(
    controller: mobileController,
    keyboardType: TextInputType.phone,
    decoration: InputDecoration(
      hintText: "Mobile Number",
      hintStyle: TextStyle(color: Colors.grey),
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
      ),
      focusedBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
        borderSide: BorderSide(
          color: Colors.pink.shade300,
          width: 2,
        ),
      ),
      contentPadding: const EdgeInsets.symmetric(horizontal: 16),
    ),
  ),
),
```

```
const SizedBox(height: 20),

// Continue Button
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: SizedBox(
    width: double.infinity,
    child: ElevatedButton(
      onPressed: () {
        // Navigate to Home Page
        Navigator.pushNamed(context, '/home');
      },
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.pink.shade300,
        padding: const EdgeInsets.symmetric(vertical: 16),
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(8),
        ),
      ),
      child: Text(
        "CONTINUE",
        style: TextStyle(
          fontSize: 16,
          color: Colors.white,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
  ),
),

const SizedBox(height: 20),

// Don't have an account? Sign up
Row(
  mainAxisAlignment: MainAxisAlignment.center,
```

```
          children: [
            Text(
              "Don't have an account? ",
              style: TextStyle(color: Colors.black54),
            ),
            GestureDetector(
              onTap: () {
                // Navigate to Sign Up Page
                Navigator.pushNamed(context, '/signup');
              },
              child: Text(
                "Sign up",
                style: TextStyle(
                  color: Colors.pink.shade300,
                  fontWeight: FontWeight.bold,
                ),
              ),
            ),
          ],
        ),
      ],
    ),
  ),
 );
 }
}
```

- **Container**: Used for the background gradient and image.
- **Column**: Structures the UI.
- **Row**: Arranges "Login or Signup" text horizontally.
- **TextField**: Captures mobile number input.
- **ElevatedButton**: Used for the "CONTINUE" button.
- **GestureDetector**: Allows user interaction for the "Sign up" text.


Android Emulator - flutter_emulator:5554
10:50
WELCOME TO MYNTRA
FLAT ₹200 O
+ Free Shipping On Your 1st Orc
Sign Up Now
Login or Signup
Mobile Number
CONTINUE
Don't have an account? Sign up

2. **Sign-Up Page**

Code:

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

class SignUpPage extends StatelessWidget {
 final TextEditingController nameController = TextEditingController();
 final TextEditingController emailController = TextEditingController();
 final TextEditingController mobileController = TextEditingController();
 final TextEditingController passwordController = TextEditingController();

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   backgroundColor: Colors.white,
   body: SafeArea(
    child: Column(
     children: [
      // Full Banner Section with Image
      Container(
       width: double.infinity,
       height: 200,
       decoration: BoxDecoration(
        gradient: LinearGradient(
         colors: [Colors.yellow.shade100, Colors.pink.shade100],
         begin: Alignment.topLeft,
         end: Alignment.bottomRight,
        ),
       ),
       child: Image.asset(
        'assets/images/shopping_cart.png', // Replace with your image
        fit: BoxFit.cover,
       ),
      ),

      const SizedBox(height: 20),
```

```
// Sign Up Text
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: Row(
    children: [
      Text(
        "Create ",
        style: TextStyle(
          fontSize: 18,
          fontWeight: FontWeight.bold,
          color: Colors.black,
        ),
      ),
      Text(
        "Account",
        style: TextStyle(
          fontSize: 18,
          color: Colors.black54,
        ),
      ),
    ],
  ),
),

const SizedBox(height: 20),

// Name Field
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: TextField(
    controller: nameController,
    decoration: InputDecoration(
      hintText: "Full Name",
      hintStyle: TextStyle(color: Colors.grey),
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
```

```
      ),
      focusedBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
        borderSide: BorderSide(
          color: Colors.pink.shade300,
          width: 2,
        ),
      ),
      contentPadding: const EdgeInsets.symmetric(horizontal: 16),
    ),
  ),
),

const SizedBox(height: 12),

// Email Field
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: TextField(
    controller: emailController,
    keyboardType: TextInputType.emailAddress,
    decoration: InputDecoration(
      hintText: "Email Address",
      hintStyle: TextStyle(color: Colors.grey),
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
      ),
      focusedBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
        borderSide: BorderSide(
          color: Colors.pink.shade300,
          width: 2,
        ),
      ),
      contentPadding: const EdgeInsets.symmetric(horizontal: 16),
    ),
```

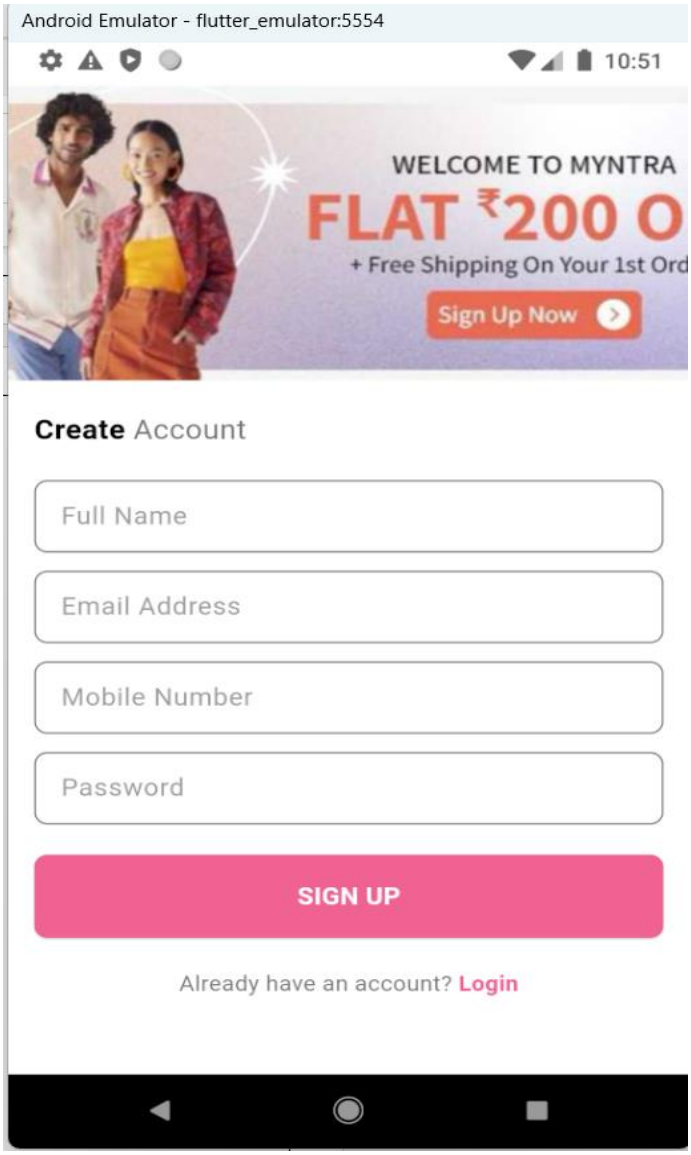
```
  ),
),

const SizedBox(height: 12),

// Mobile Number Field
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: TextField(
    controller: mobileController,
    keyboardType: TextInputType.phone,
    decoration: InputDecoration(
      hintText: "Mobile Number",
      hintStyle: TextStyle(color: Colors.grey),
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
      ),
      focusedBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
        borderSide: BorderSide(
          color: Colors.pink.shade300,
          width: 2,
        ),
      ),
      contentPadding: const EdgeInsets.symmetric(horizontal: 16),
    ),
  ),
),

const SizedBox(height: 12),

// Password Field
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: TextField(
    controller: passwordController,
```

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

const SizedBox(height: 20),

// Sign Up Button
Padding(
 padding: const EdgeInsets.symmetric(horizontal: 16.0),
 child: SizedBox(
  width: double.infinity,
  child: ElevatedButton(
   onPressed: () {
    // Navigate to Home Page
    Navigator.pushNamed(context, '/home');
   },
   style: ElevatedButton.styleFrom(
    backgroundColor: Colors.pink.shade300,
    padding: const EdgeInsets.symmetric(vertical: 16),
    shape: RoundedRectangleBorder(
     borderRadius: BorderRadius.circular(8),
```

```
        ),
      ),
      child: Text(
        "SIGN UP",
        style: TextStyle(
          fontSize: 16,
          color: Colors.white,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
  ),
),

const SizedBox(height: 20),

// Already have an account? Login
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Text(
      "Already have an account? ",
      style: TextStyle(color: Colors.black54),
    ),
    GestureDetector(
      onTap: () {
        Navigator.pop(context); // Go back to Login Page
      },
      child: Text(
        "Login",
        style: TextStyle(
          color: Colors.pink.shade300,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
```

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

- **Container**: Background image section.
- **Column**: Arranges the elements vertically.
- **TextField**: Captures name, email, mobile number, and password input.
- **ElevatedButton**: "SIGN UP" button.



3. **Home Page**

Code:

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

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Colors.white,
        elevation: 0,
        title: Row(
          children: [
            Text(
              "Myntra",
              style: TextStyle(
                fontWeight: FontWeight.bold,
                color: Colors.black,
              ),
            ),
            Spacer(),
            Text(
              "BECOME INSIDER >",
              style: TextStyle(
                color: Colors.orange,
                fontWeight: FontWeight.bold,
                fontSize: 12,
              ),
            ),
          ],
        ),
        actions: [
          Icon(Icons.notifications_none, color: Colors.black),
```

```
    SizedBox(width: 16),
    Icon(Icons.favorite_border, color: Colors.black),
    SizedBox(width: 16),
    Icon(Icons.account_circle, color: Colors.black),
    SizedBox(width: 16),
  ],
),
body: SingleChildScrollView(
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      // Search Bar
      Padding(
        padding: const EdgeInsets.all(16.0),
        child: TextField(
          decoration: InputDecoration(
            hintText: "Search for brands and products",
            prefixIcon: Icon(Icons.search),
            border: OutlineInputBorder(
              borderRadius: BorderRadius.circular(8),
            ),
          ),
        ),
      ),

      // HIM / HER Banner
      Padding(
        padding: const EdgeInsets.symmetric(horizontal: 16.0),
        child: ClipRRect(
          borderRadius: BorderRadius.circular(10),
          child: Image.asset(
            "assets/images/1.png",
            width: double.infinity, // Full width
            height: 200, // Fixed height
            fit: BoxFit.cover, // Ensures proper coverage
          ),
```

```
        ),
      ),

      SizedBox(height: 16),

      // Price Store Section
      _buildPriceStore(),

      // All-Time Faves Section
      Padding(
        padding: const EdgeInsets.all(16.0),
        child: Text(
          "ALL-TIME FAVES\nCurated For You",
          style: TextStyle(
            fontWeight: FontWeight.bold,
            fontSize: 16,
          ),
        ),
      ),
      _buildFavesGrid(),
    ],
  ),
),

// Bottom Navigation Bar
bottomNavigationBar: BottomNavigationBar(
  type: BottomNavigationBarType.fixed,
  selectedItemColor: Colors.pink,
  unselectedItemColor: Colors.grey,
  items: [
    BottomNavigationBarItem(
      icon: Icon(Icons.home),
      label: "Home",
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.category),
```

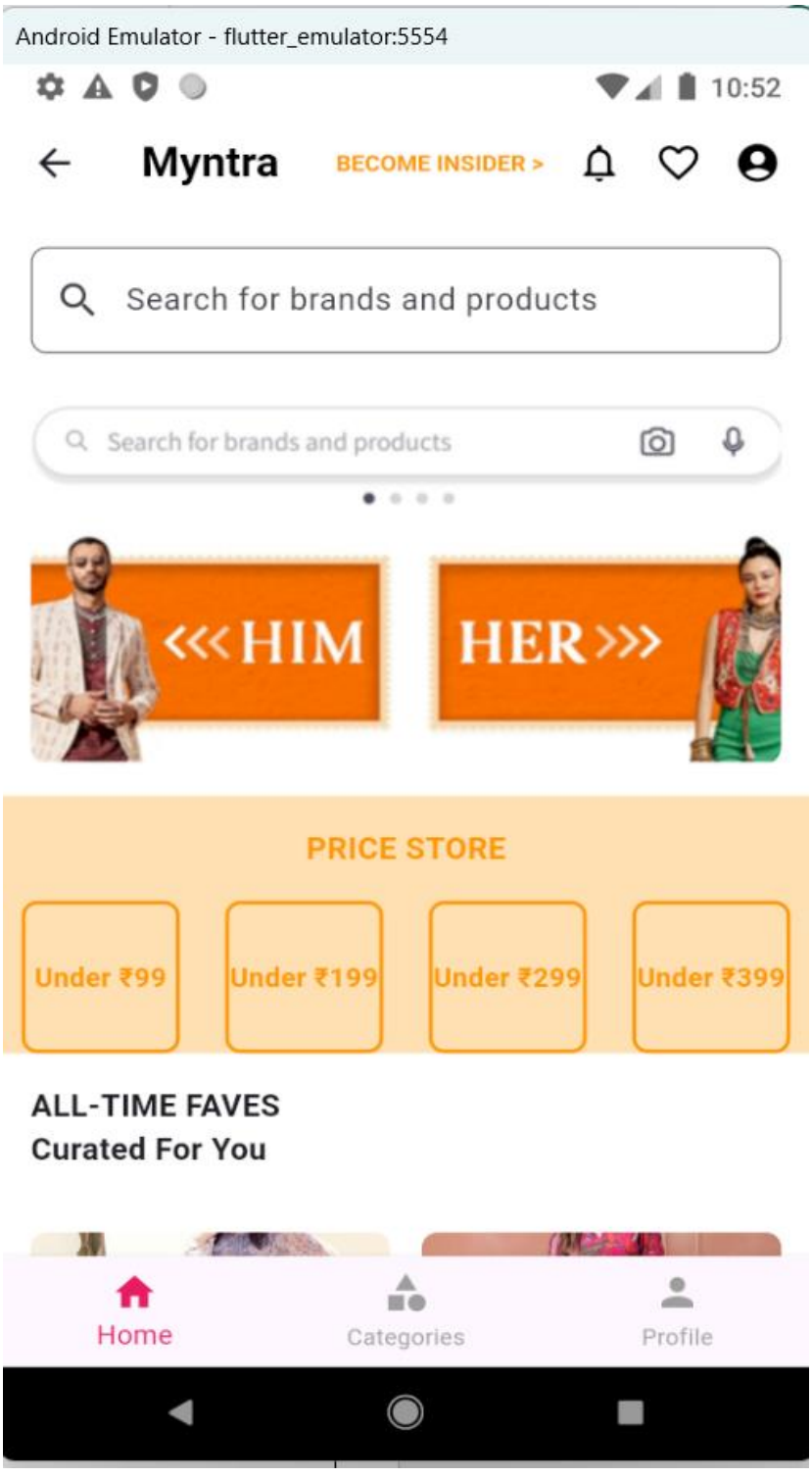
```
        label: "Categories",
       ),
       BottomNavigationBarItem(
        icon: Icon(Icons.person),
        label: "Profile",
       ),
      ],
      onTap: (index) {
       String currentRoute = ModalRoute.of(context)?.settings.name ?? "";
       String newRoute = "";

       switch (index) {
        case 0:
         newRoute = '/home';
         break;
        case 1:
         newRoute = '/categories';
         break;
        case 2:
         newRoute = '/profile';
         break;
       }

       if (currentRoute != newRoute) {
        Navigator.pushNamed(context, newRoute);
       }
      },
    ),
   );
}

Widget _buildPriceStore() {
  return Container(
    color: Colors.orange.shade100,
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.center,
```

```
      children: [
        Padding(
          padding: const EdgeInsets.all(16.0),
          child: Text(
            "PRICE STORE",
            style: TextStyle(
              fontWeight: FontWeight.bold,
              color: Colors.orange,
              fontSize: 16,
            ),
          ),
        ),
        Row(
          mainAxisAlignment: MainAxisAlignment.spaceAround,
          children: [
            _buildPriceTile("Under ₹99"),
            _buildPriceTile("Under ₹199"),
            _buildPriceTile("Under ₹299"),
            _buildPriceTile("Under ₹399"),
          ],
        ),
      ],
    ),
  );
}

Widget _buildPriceTile(String text) {
  return Container(
    height: 80,
    width: 80,
    decoration: BoxDecoration(
      border: Border.all(color: Colors.orange, width: 2),
      borderRadius: BorderRadius.circular(8),
    ),
    child: Center(
      child: Text(
```

```
        text,
        textAlign: TextAlign.center,
        style: TextStyle(
          fontWeight: FontWeight.bold,
          color: Colors.orange,
        ),
      ),
    ),
  );
}

Widget _buildFavesGrid() {
  return Padding(
    padding: const EdgeInsets.all(16.0),
    child: GridView.count(
      shrinkWrap: true,
      crossAxisCount: 2,
      crossAxisSpacing: 16,
      mainAxisSpacing: 16,
      childAspectRatio: 0.75, // Adjusted for better image fit
      physics: NeverScrollableScrollPhysics(),
      children: [
        _buildFaveItem("Under ₹799", "assets/images/item1.png"),
        _buildFaveItem("Under ₹699", "assets/images/item2.png"),
        _buildFaveItem("Under ₹599", "assets/images/item3.png"),
        _buildFaveItem("Under ₹499", "assets/images/item4.png"),
      ],
    ),
  );
}

Widget _buildFaveItem(String price, String imagePath) {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      ClipRRect(
```

```
        borderRadius: BorderRadius.circular(8),
        child: Image.asset(
          imagePath,
          height: 140, // Increased height for better visibility
          width: double.infinity,
          fit: BoxFit.cover, // Ensures the image fits well
        ),
      ),
      SizedBox(height: 8),
      Text(
        price,
        style: TextStyle(
          fontWeight: FontWeight.bold,
        ),
      ),
    ],
  );
 }
}
```

- **AppBar**: Displays Myntra logo, user account icon, and notification icon.
- **TextField**: Search bar for product searches.
- **ClipRRect**: Rounds the corners of images for banners and product thumbnails.
- **GridView**: Displays product collections.
- **BottomNavigationBar**: Provides navigation for Home, Categories, and Profile.


Android Emulator - flutter_emulator:5554
10:52
Myntra
BECOME INSIDER >
Search for brands and products
Search for brands and products
HIM
HER
PRICE STORE
Under ₹99
Under ₹199
Under ₹299
Under ₹399
ALL-TIME FAVES
Curated For You
Home
Categories
Profile

4. Categories Page

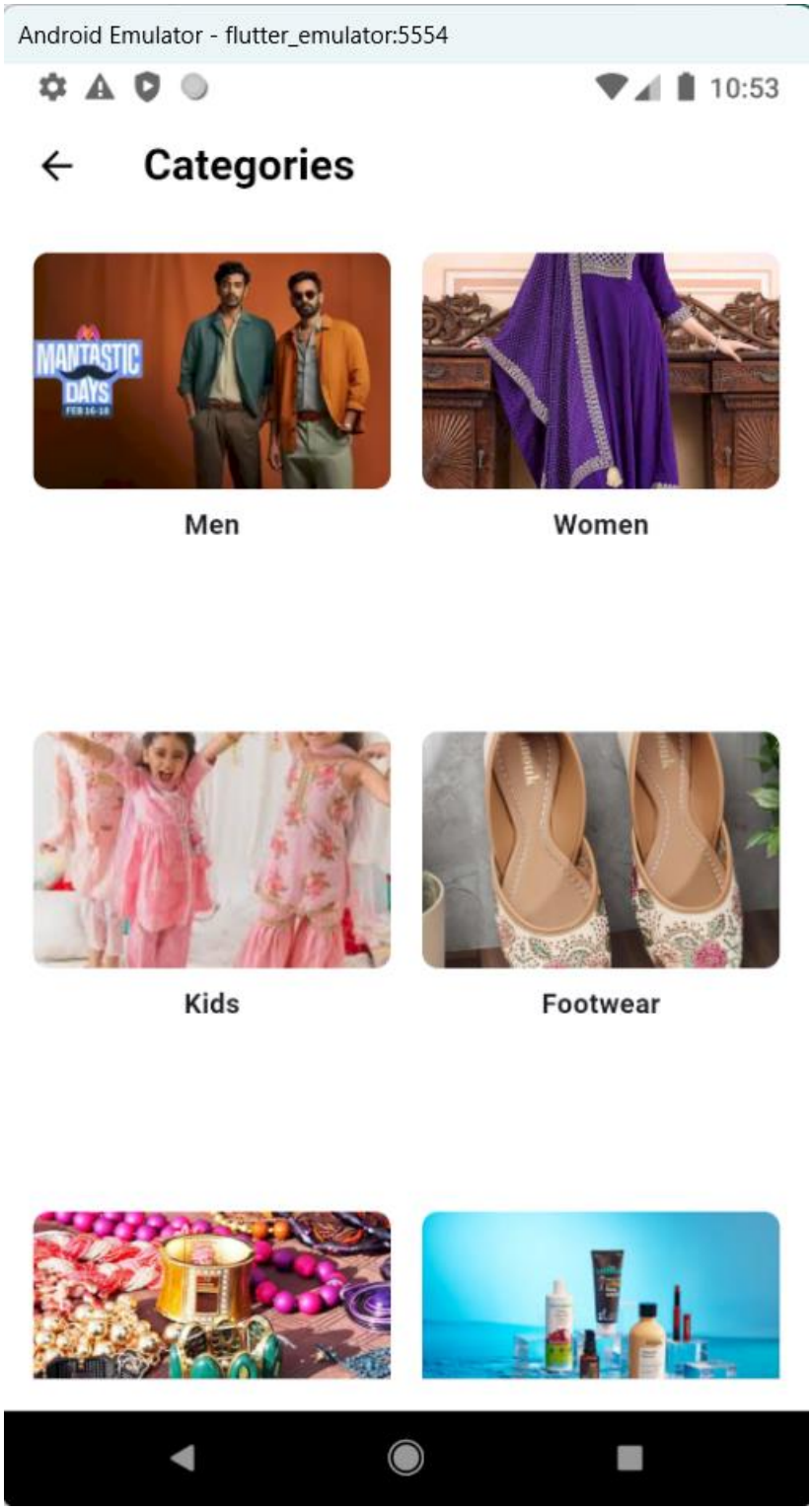
Code:

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

class CategoriesPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Colors.white,
        elevation: 0,
        title: Text(
          "Categories",
          style: TextStyle(
            fontWeight: FontWeight.bold,
            color: Colors.black,
          ),
        ),
        leading: IconButton(
          icon: Icon(Icons.arrow_back, color: Colors.black),
          onPressed: () {
            Navigator.pop(context);
          },
        ),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: GridView.builder(
          gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 2,
            crossAxisSpacing: 16,
            mainAxisSpacing: 16,
            childAspectRatio: 0.8,
          ),
          itemCount: categories.length,
```

```
        itemBuilder: (context, index) {
          return _buildCategoryTile(
            categories[index]['name']!,
            categories[index]['image']!,
          );
        },
      ),
    ),
  );
}

Widget _buildCategoryTile(String name, String imagePath) {
  return GestureDetector(
    onTap: () {
      print("$name category clicked");
    },
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.center,
      children: [
        ClipRRect(
          borderRadius: BorderRadius.circular(8),
          child: Image.asset(
            imagePath,
            height: 120,
            width: double.infinity,
            fit: BoxFit.cover,
            errorBuilder: (context, error, stackTrace) {
              return Container(
                height: 120,
                width: double.infinity,
                color: Colors.grey[300],
                child: Icon(Icons.image, size: 50, color: Colors.grey[600]),
              );
            },
          ),
        ),
```

```
        SizedBox(height: 8),
        Text(
          name,
          style: TextStyle(
            fontWeight: FontWeight.bold,
          ),
        ),
      ],
    ),
  );
 }
}

// Category List with Names and Image Paths
final List<Map<String, String>> categories = [
 {"name": "Men", "image": "assets/images/men.png"},
 {"name": "Women", "image": "assets/images/women.png"},
 {"name": "Kids", "image": "assets/images/kids.png"},
 {"name": "Footwear", "image": "assets/images/footwear.png"},
 {"name": "Accessories", "image": "assets/images/accessories.png"},
 {"name": "Beauty", "image": "assets/images/beauty.png"},
];
```

- **GridView.builder**: Displays categories dynamically.
- **ClipRRect**: Rounds category images.


Android Emulator - flutter_emulator:5554
10:53
Categories
MANTASTIC DAYS
Men
Women
Kids
Footwear

Flutter Syntax for Common Widgets:

1.Scaffold

```
dart
CopyEdit
Scaffold(
  appBar: AppBar(
    title: Text("Myntra"),
  ),
  body: Center(child: Text("Hello, Myntra!")),
);
```

2.Container

```
dart
CopyEdit
Container(
  width: 100,
  height: 100,
  decoration: BoxDecoration(
    color: Colors.pink,
    borderRadius: BorderRadius.circular(10),
  ),
);
```

3.Column & Row

```
dart
CopyEdit
Column(
  children: [
    Text("Login"),
    ElevatedButton(onPressed: () {}, child: Text("Continue")),
  ],
);
Row(
  mainAxisAlignment: MainAxisAlignment.center,
```

```
  children: [
    Text("Already have an account? "),
    GestureDetector(
      onTap: () {},
      child: Text("Login", style: TextStyle(color: Colors.blue)),
    ),
  ],
);
```

4.TextField

```
dart
CopyEdit
TextField(
  decoration: InputDecoration(
    hintText: "Enter Mobile Number",
    border: OutlineInputBorder(),
  ),
);
```

5.GridView

```
dart
CopyEdit
GridView.count(
  crossAxisCount: 2,
  children: [
    Image.asset("assets/images/men.png"),
    Image.asset("assets/images/women.png"),
  ],
);
```

Project Title: **Roll No.**

MAD & PWA Lab
Journal

Experiment No.	03
Experiment Title.	To include icons, images, fonts in Flutter app
Roll No.	38
Name	Snehal Anilkumar Patil
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

MAD & PWA Lab Journal

Experiment No.	03
Experiment Title.	To include icons, images, fonts in Flutter app
Roll No.	38
Name	Snehal Patil
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

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

Theory:

Using Icons in Flutter

Icons in Flutter can be added using the built-in Material Icons or custom icon packs.

(a) Material Icons

Flutter provides a collection of built-in Material Icons, which can be used with the Icon widget.

Eg: Icon(Icons.home, size: 30, color: Colors.blue)

(b) Custom Icons

If you need icons that are not available in the Material Icons set, you can use external icon packs like:

- Font Awesome (font_awesome_flutter package)
- Custom SVG Icons (flutter_svg package)

  Eg in pubspec.yaml file -

  dependencies:
    font_awesome_flutter: ^10.5.0

In code -

```
import 'package:font_awesome_flutter/font_awesome_flutter.dart';

IconButton(
  icon: FaIcon(FontAwesomeIcons.heart, color: Colors.red),
  onPressed: () {},
)
```

Adding Images in Flutter

Images can be loaded in Flutter from different sources like assets, network, or memory.

(a) Using Network Images

Network images are loaded from an online URL. Example:

Eg: Image.network("https://example.com/sample.jpg", width: 200, height: 150)

(b) Using Asset Images

To use images from the local project folder (assets/), follow these steps:

1. Place the image inside the assets/images/ folder.
2. Declare the image in pubspec.yaml:

```
flutter:
  assets:
    - assets/images/sample.png
```

In code: Image.asset("assets/images/sample.png", width: 200, height: 150)

Adding Custom Fonts in Flutter

Custom fonts improve the visual identity of an app. Steps to

Add a Custom Font:

1. Download the font and place it inside the assets/fonts/ folder.
2. Declare the font in pubspec.yaml:

```
flutter:
  fonts:
    - family: CustomFont fonts:
        - asset: assets/fonts/CustomFont-Regular.ttf
        - asset: assets/fonts/CustomFont-Bold.ttf weight:
          700
```

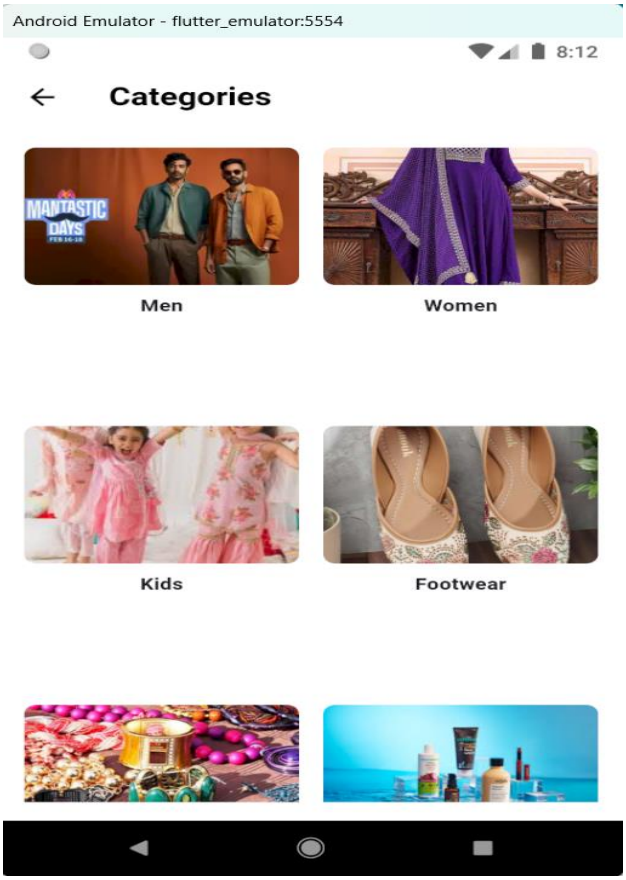
In Code:

```
 (
  "Hello, Flutter!",
  style: TextStyle(fontFamily: "CustomFont", fontSize: 20, fontWeight: FontWeight.bold),
)
```

Categories Page:

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

class CategoriesPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Colors.white,
        elevation: 0,
        title: Text(
          "Categories",
          style: TextStyle(
            fontWeight: FontWeight.bold,
            color: Colors.black,
          ),
        ),
        leading: IconButton(
          icon: Icon(Icons.arrow_back, color:
Colors.black),
          onPressed: () {
            Navigator.pop(context);
          },
        ),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: GridView.builder(
          gridDelegate:
SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 2,
            crossAxisSpacing: 16,
            mainAxisSpacing: 16,
            childAspectRatio: 0.8,
          ),
          itemCount: categories.length,
          itemBuilder: (context, index) {
            return _buildCategoryTile(
              categories[index]['name']!,
              categories[index]['image']!,
            );
          },
        ),
      ),
    );
  }

  Widget _buildCategoryTile(String name, String
imagePath) {
    return GestureDetector(
      onTap: () {
        print("$name category clicked");
      },
      child: Column(
        crossAxisAlignment:
CrossAxisAlignment.center,
        children: [
          ClipRRect(
            borderRadius: BorderRadius.circular(8),
            child: Image.asset(
              imagePath,
              height: 120,
              width: double.infinity,
              fit: BoxFit.cover,
              errorBuilder: (context, error, stackTrace) {
                return Container(
                  height: 120,
                  width: double.infinity,
                  color: Colors.grey[300],
                  child: Icon(Icons.image, size: 50, color: Colors.grey[600]),
                );
              },
            ),
          ),
          SizedBox(height: 8),
          Text(
            name,
            style: TextStyle(
              fontWeight: FontWeight.bold,
            ),
          ),
        ],
      ),
    );
  }
}

// Category List with Names and Image Paths
final List<Map<String, String>> categories = [
```

```
  {"name": "Men", "image": "assets/images/men.png"},
  {"name": "Women", "image": "assets/images/women.png"},
  {"name": "Kids", "image": "assets/images/kids.png"},
  {"name": "Footwear", "image": "assets/images/footwear.png"},
  {"name": "Accessories", "image": "assets/images/accessories.png"},
  {"name": "Beauty", "image": "assets/images/beauty.png"},
];
```



Home Page:

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

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Colors.white,
        elevation: 0,
        title: Row(
          children: [
            Text(
              "Myntra",
              style: TextStyle(
                fontWeight: FontWeight.bold,
                color: Colors.black,
              ),
            ),
            Spacer(),
            Text(
              "BECOME INSIDER >",
```

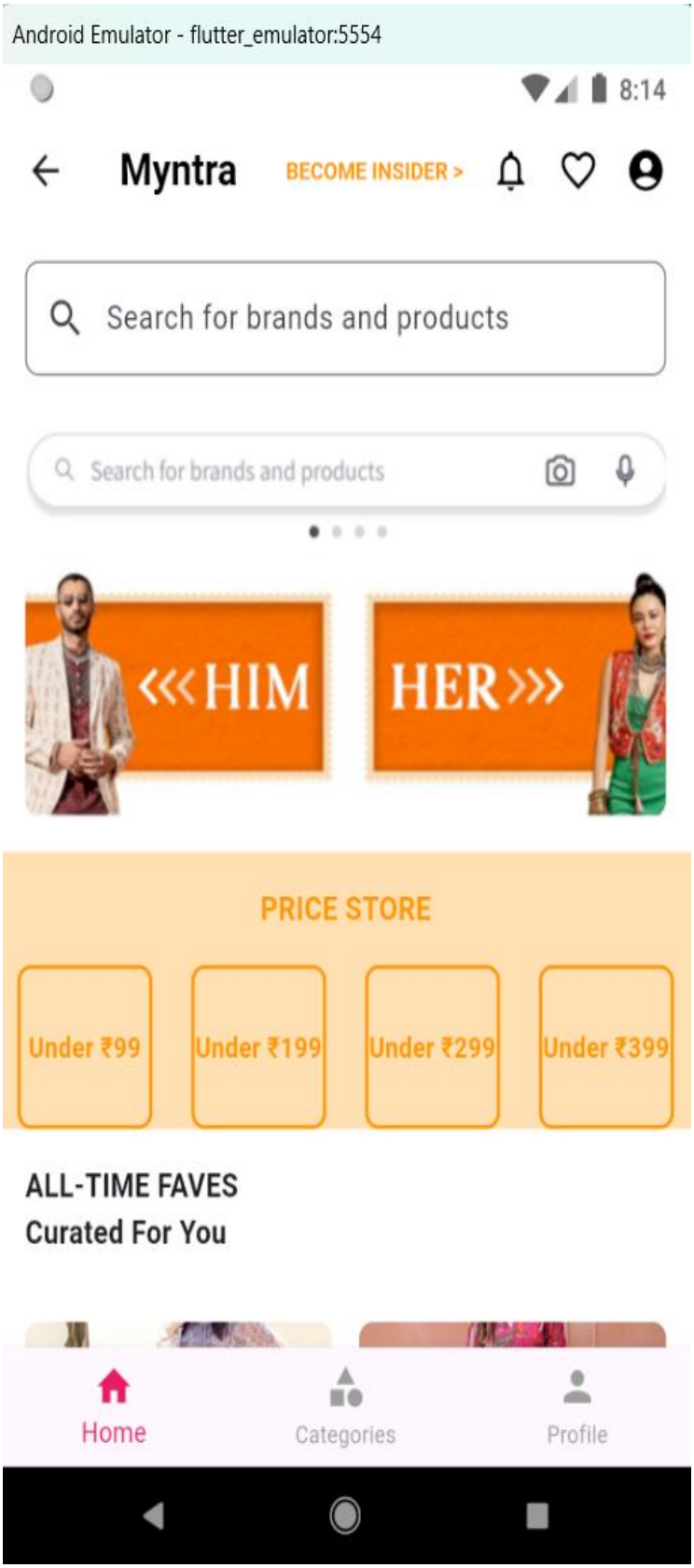
```
          style: TextStyle(
            color: Colors.orange,
            fontWeight: FontWeight.bold,
            fontSize: 12,
          ),
        ),
      ],
    ),
    actions: [
      Icon(Icons.notifications_none, color: Colors.black),
      SizedBox(width: 16),
      Icon(Icons.favorite_border, color: Colors.black),
      SizedBox(width: 16),
      Icon(Icons.account_circle, color: Colors.black),
      SizedBox(width: 16),
    ],
  ),
  body: SingleChildScrollView(
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        // Search Bar
        Padding(
          padding: const EdgeInsets.all(16.0),
          child: TextField(
            decoration: InputDecoration(
              hintText: "Search for brands and products",
              prefixIcon: Icon(Icons.search),
              border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(8),
              ),
            ),
          ),
        ),

        // HIM / HER Banner
        Padding(
          padding: const EdgeInsets.symmetric(horizontal: 16.0),
          child: ClipRRect(
            borderRadius: BorderRadius.circular(10),
            child: Image.asset(
              "assets/images/1.png",
              width: double.infinity, // Full width
              height: 200, // Fixed height
              fit: BoxFit.cover, // Ensures proper coverage
            ),
          ),
        ),

        SizedBox(height: 16),

        // Price Store Section
        _buildPriceStore(),

        // All-Time Faves Section
        Padding(
          padding: const EdgeInsets.all(16.0),
          child: Text(
            "ALL-TIME FAVES\nCurated For You",
            style: TextStyle(
```

```
                fontWeight: FontWeight.bold,
                fontSize: 16,
              ),
            ),
          ),
          _buildFavesGrid(),
        ],
      ),
    ),

    // Bottom Navigation Bar
    bottomNavigationBar: BottomNavigationBar(
      type: BottomNavigationBarType.fixed,
      selectedItemColor: Colors.pink,
      unselectedItemColor: Colors.grey,
      items: [
        BottomNavigationBarItem(
          icon: Icon(Icons.home),
          label: "Home",
        ),
        BottomNavigationBarItem(
          icon: Icon(Icons.category),
          label: "Categories",
        ),
        BottomNavigationBarItem(
          icon: Icon(Icons.person),
          label: "Profile",
        ),
      ],
      onTap: (index) {
        String currentRoute = ModalRoute.of(context)?.settings.name ?? "";
        String newRoute = "";

        switch (index) {
          case 0:
            newRoute = '/home';
            break;
          case 1:
            newRoute = '/categories';
            break;
          case 2:
            newRoute = '/profile';
            break;
        }

        if (currentRoute != newRoute) {
          Navigator.pushNamed(context, newRoute);
        }
      },
    ),
  );
}

Widget _buildPriceStore() {
  return Container(
    color: Colors.orange.shade100,
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.center,
      children: [
        Padding(
```

```
          padding: const EdgeInsets.all(16.0),
          child: Text(
            "PRICE STORE",
            style: TextStyle(
              fontWeight: FontWeight.bold,
              color: Colors.orange,
              fontSize: 16,
            ),
          ),
        ),
        Row(
          mainAxisAlignment:
MainAxisAlignment.spaceAround,
          children: [
            _buildPriceTile("Under ₹99"),
            _buildPriceTile("Under ₹199"),
            _buildPriceTile("Under ₹299"),
            _buildPriceTile("Under ₹399"),
          ],
        ),
      ],
    ),
  );
 }

 Widget _buildPriceTile(String text) {
  return Container(
    height: 80,
    width: 80,
    decoration: BoxDecoration(
      border: Border.all(color: Colors.orange,
width: 2),
      borderRadius: BorderRadius.circular(8),
    ),
      ],
    ),
  );
 }

 Widget _buildFaveItem(String price, String imagePath) {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      ClipRRect(
        borderRadius: BorderRadius.circular(8),
        child: Image.asset(
          imagePath,
          height: 140, // Increased height for better visibility
          width: double.infinity,
          fit: BoxFit.cover, // Ensures the image fits well
        ),
      ),
      SizedBox(height: 8),
      Text(
        price,
        style: TextStyle(
          fontWeight: FontWeight.bold,
        ),
      ),
    ],
```

```
      child: Center(
        child: Text(
          text,
          textAlign: TextAlign.center,
          style: TextStyle(
            fontWeight: FontWeight.bold,
            color: Colors.orange,
          ),
        ),
      ),
    );
  }

  Widget _buildFavesGrid() {
    return Padding(
      padding: const EdgeInsets.all(16.0),
      child: GridView.count(
        shrinkWrap: true,
        crossAxisCount: 2,
        crossAxisSpacing: 16,
        mainAxisSpacing: 16,
        childAspectRatio: 0.75, // Adjusted for better
image fit
        physics: NeverScrollableScrollPhysics(),
        children: [
          _buildFaveItem("Under ₹799",
"assets/images/item1.png"),
          _buildFaveItem("Under ₹699",
"assets/images/item2.png"),
          _buildFaveItem("Under ₹599",
"assets/images/item3.png"),
          _buildFaveItem("Under ₹499",
"assets/images/item4.png"),
```

```
    );
  }
}
```

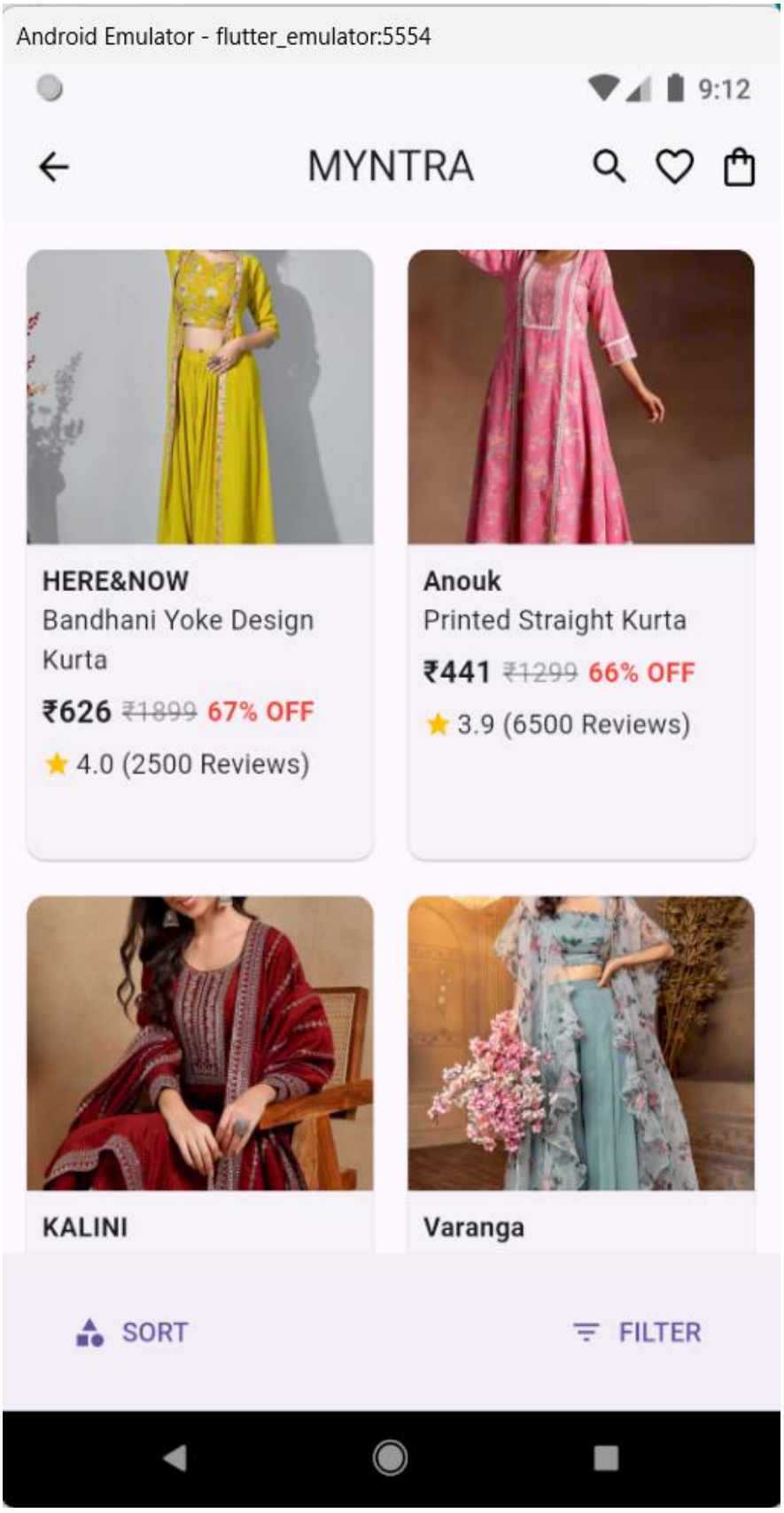

Android Emulator - flutter_emulator:5554
8:14
Myntra
BECOME INSIDER >
Search for brands and products
Search for brands and products
«HIM
HER»
PRICE STORE
Under ₹99
Under ₹199
Under ₹299
Under ₹399
ALL-TIME FAVES
Curated For You
Home
Categories
Profile

Products Page:

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

void main() {
  runApp(MaterialApp(
    debugShowCheckedModeBanner: false,
    home: ProductPage(),
  ));
}

class ProductPage extends StatelessWidget {
  final List<Map<String, dynamic>> products = [
    {
      "image": "assets/images/product1.png", // Replace with actual image URL
      "brand": "HERE&NOW",
      "name": "Bandhani Yoke Design Kurta",
      "price": 626,
      "originalPrice": 1899,
      "discount": "67% OFF",
      "rating": 4.0,
      "reviews": 2500,
    },
    {
      "image": "assets/images/product2.png",
      "brand": "Anouk",
      "name": "Printed Straight Kurta",
      "price": 441,
      "originalPrice": 1299,
      "discount": "66% OFF",
      "rating": 3.9,
      "reviews": 6500,
    },
    {
      "image": "assets/images/product3.png",
      "brand": "KALINI",
      "name": "High-Slit Kurta",
      "price": 599,
      "originalPrice": 2499,
      "discount": "76% OFF",
      "rating": 4.3,
      "reviews": 5500,
    },
    {
      "image": "assets/images/product4.png",
      "brand": "Varanga",
      "name": "Women Printed Kurta",
      "price": 599,
      "originalPrice": 1499,
      "discount": "60% OFF",
      "rating": 4.1,
      "reviews": 2600,
    },
  ];

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

```
  appBar: AppBar(
    title: Text("MYNTRA"),
    centerTitle: true,
    backgroundColor: Colors.white,
    elevation: 1,
    iconTheme: IconThemeData(color: Colors.black),
    actions: [
      Icon(Icons.search, color: Colors.black),
      SizedBox(width: 10),
      Icon(Icons.favorite_border, color: Colors.black),
      SizedBox(width: 10),
      Icon(Icons.shopping_bag_outlined, color: Colors.black),
      SizedBox(width: 10),
    ],
  ),
  body: GridView.builder(
    padding: EdgeInsets.all(10),
    gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
      crossAxisCount: 2,
      childAspectRatio: 0.6,
      crossAxisSpacing: 10,
      mainAxisSpacing: 10,
    ),
    itemCount: products.length,
    itemBuilder: (context, index) {
      final product = products[index];
      return Card(
        shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(10)),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            ClipRRect(
              borderRadius: BorderRadius.vertical(top: Radius.circular(10)),
              child: Image.asset(
                product['image'],
                height: 150,
                width: double.infinity,
                fit: BoxFit.cover,
              ),
            ),
            Padding(
              padding: const EdgeInsets.all(8.0),
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Text(
                    product['brand'],
                    style: TextStyle(fontWeight: FontWeight.bold),
                  ),
                  Text(
                    product['name'],
                    maxLines: 2,
                    overflow: TextOverflow.ellipsis,
                  ),
                  SizedBox(height: 5),
                  Row(
                    children: [
                      Text(
                        "₹${product['price']}",
                        style: TextStyle(fontWeight: FontWeight.bold, fontSize: 16),
```

```
                    ),
                    SizedBox(width: 5),
                    Text(
                      "₹${product['originalPrice']}",
                      style: TextStyle(
                        decoration: TextDecoration.lineThrough,
                        color: Colors.grey,
                      ),
                    ),
                    SizedBox(width: 5),
                    Text(
                      product['discount'],
                      style: TextStyle(color: Colors.red, fontWeight: FontWeight.bold),
                    ),
                  ],
                ),
                SizedBox(height: 5),
                Row(
                  children: [
                    Icon(Icons.star, color: Colors.amber, size: 16),
                    SizedBox(width: 2),
                    Text("${product['rating']} (${product['reviews']} Reviews)"),
                  ],
                ),
              ],
            ),
          ),
        ],
      ),
    );
  },
),
bottomNavigationBar: BottomAppBar(
  child: Padding(
    padding: const EdgeInsets.all(10.0),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        TextButton.icon(
          onPressed: () {},
          icon: Icon(Icons.category),
          label: Text("SORT"),
        ),
        TextButton.icon(
          onPressed: () {},
          icon: Icon(Icons.filter_list),
          label: Text("FILTER"),
        ),
      ],
    ),
  ),
),
);
}
}
```


Android Emulator - flutter_emulator:5554
9:12
MYNTRA
HERE&NOW
Bandhani Yoke Design Kurta
₹626 ₹1899 67% OFF
4.0 (2500 Reviews)
Anouk
Printed Straight Kurta
₹441 ₹1299 66% OFF
3.9 (6500 Reviews)
KALINI
Varanga
SORT
FILTER

Project Title: **Roll No.**

MAD & PWA Lab
Journal

Experiment No.	04
Experiment Title.	To create an interactive Form using form widget
Roll No.	38
Name	Snehal Anilkumar Patil
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

Snehal Patil D15A 38

EXPERIMENT NO: 04

Aim:

To create an interactive **Login** and **Sign-Up** form using the **Form widget** in Flutter.

Theory: Understanding Forms in Flutter

A **Form** in Flutter is a structured widget used to collect user input through various fields like **TextFields**, **Dropdowns**, and **Buttons**. Forms are essential for user authentication, data entry, and submissions in applications.

Key Features of Forms in Flutter:

1. **Validation** – Ensures user input meets required criteria before submission.
2. **State Management** – Manages user input efficiently.
3. **Error Handling** – Displays validation errors when needed.

Components of Login & Sign-Up Forms in Flutter:

1. Form Widget:

- Acts as a container to group multiple **TextFormFields** and manage their validation.

2. TextFormField for User Input:

- Used for entering **mobile numbers, names, emails, and passwords**.

3. Input Decoration:

- **Custom styling**, including labels, icons, borders, and hint text.

4. GlobalKey for Identification:

- **GlobalKey<FormState>** helps manage validation and state.

5. Form Validation:

- Ensures correct format for phone numbers, emails, and passwords.

6. Submit Button:

- Triggers validation and submits data.

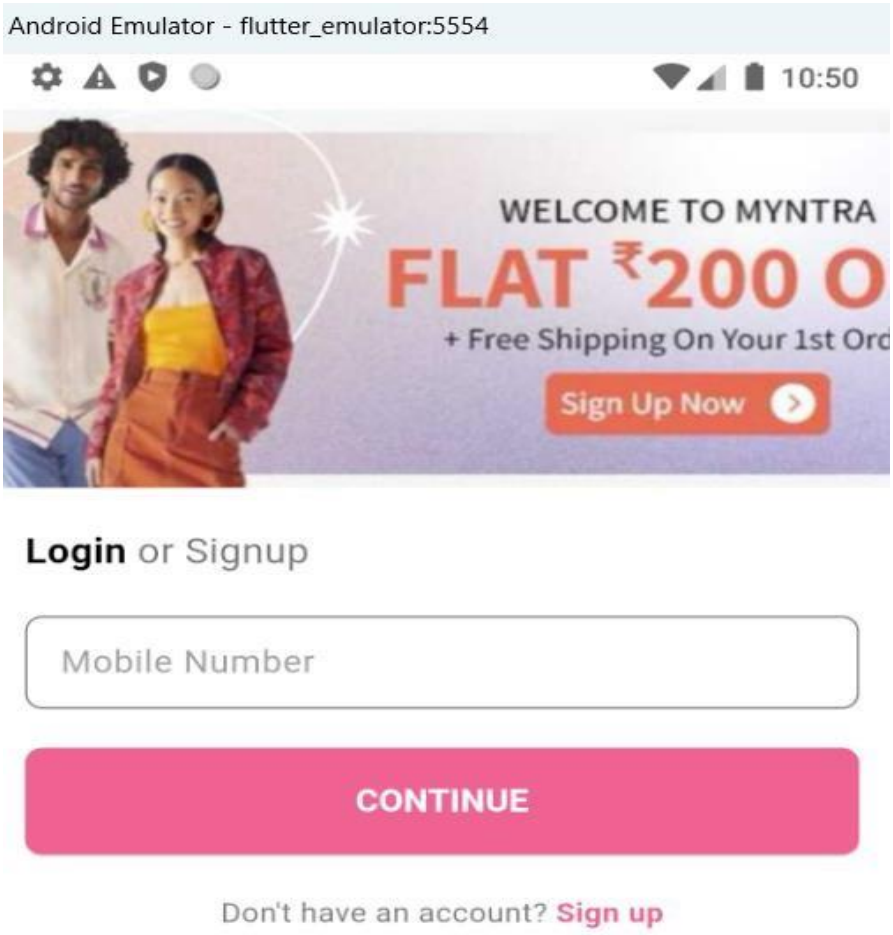
Login Page :

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

class LoginPage extends StatelessWidget {
  final TextEditingController mobileController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: SafeArea(
        child: Column(
          children: [
            // Banner with Shopping Cart Image
            Container(
              width: double.infinity,
              height: 200,
              decoration: BoxDecoration(
                gradient: LinearGradient(
                  colors: [Colors.yellow.shade100, Colors.pink.shade100],
                  begin: Alignment.topLeft,
                  end: Alignment.bottomRight,
                ),
              ),
              child: Image.asset(
                'assets/images/shopping_cart.png',
                fit: BoxFit.cover,
              ),
            ),

            const SizedBox(height: 20),

            // Login or Signup Text
            Padding(
```

```
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: Row(
    children: [
      Text("Login ", style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold, color: Colors.black)),
      Text("or Signup", style: TextStyle(fontSize: 18, color: Colors.black54)),
    ],
  ),
),

const SizedBox(height: 20),

// Mobile Number Input Field
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: TextField(
    controller: mobileController,
    keyboardType: TextInputType.phone,
    decoration: InputDecoration(
      hintText: "Mobile Number",
      border: OutlineInputBorder(borderRadius: BorderRadius.circular(8)),
      focusedBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
        borderSide: BorderSide(color: Colors.pink.shade300, width: 2),
      ),
    ),
  ),
),

const SizedBox(height: 20),

// Continue Button
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: SizedBox(
    width: double.infinity,
```

```
            child: ElevatedButton(
              onPressed: () {
                Navigator.pushNamed(context, '/home');
              },
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.pink.shade300,
                padding: const EdgeInsets.symmetric(vertical: 16),
                shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(8)),
              ),
              child: Text("CONTINUE", style: TextStyle(fontSize: 16, color: Colors.white, fontWeight: FontWeight.bold)),
            ),
          ),
        ),

        const SizedBox(height: 20),

        // Sign Up Link
        Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text("Don't have an account? ", style: TextStyle(color: Colors.black54)),
            GestureDetector(
              onTap: () {
                Navigator.pushNamed(context, '/signup');
              },
              child: Text("Sign up", style: TextStyle(color: Colors.pink.shade300, fontWeight: FontWeight.bold)),
            ),
          ],
        ),
      ],
    ),
  ),
);
}
```

}


Android Emulator - flutter_emulator:5554
10:50
WELCOME TO MYNTRA
FLAT ₹200 O
+ Free Shipping On Your 1st Ord
Sign Up Now
Login or Signup
Mobile Number
CONTINUE
Don't have an account? Sign up

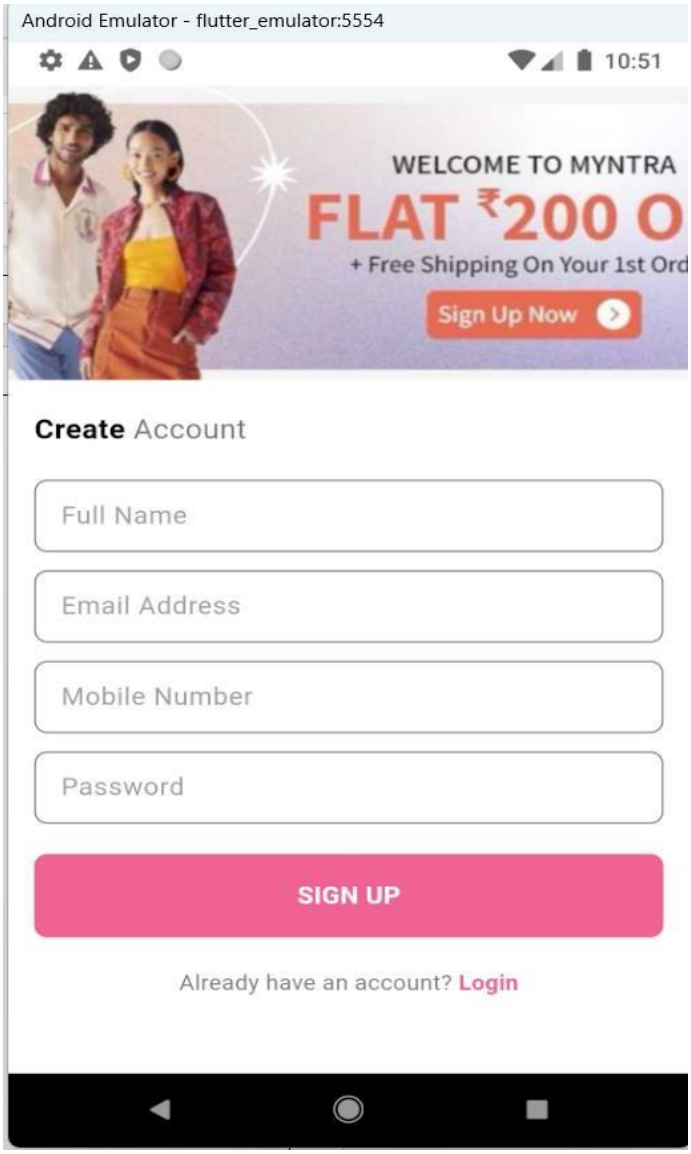
Sign-Up Page Implementation (SignUpPage.dart)

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

class SignUpPage extends StatelessWidget {
  final TextEditingController nameController = TextEditingController();
  final TextEditingController emailController = TextEditingController();
  final TextEditingController mobileController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: SafeArea(
        child: Column(
          children: [
            // Banner Section
            Container(
              width: double.infinity,
              height: 200,
              decoration: BoxDecoration(
                gradient: LinearGradient(
                  colors: [Colors.yellow.shade100, Colors.pink.shade100],
                  begin: Alignment.topLeft,
                  end: Alignment.bottomRight,
                ),
              ),
              child: Image.asset(
                'assets/images/shopping_cart.png',
                fit: BoxFit.cover,
              ),
            ),

            const SizedBox(height: 20),
```

```
// Create Account Text
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: Row(
    children: [
      Text("Create ", style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold, color: Colors.black)),
      Text("Account", style: TextStyle(fontSize: 18, color: Colors.black54)),
    ],
  ),
),

const SizedBox(height: 20),

// Name Input
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: TextField(
    controller: nameController,
    decoration: InputDecoration(
      hintText: "Full Name",
      border: OutlineInputBorder(borderRadius: BorderRadius.circular(8)),
    ),
  ),
),

const SizedBox(height: 12),

// Email Input
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: TextField(
    controller: emailController,
    keyboardType: TextInputType.emailAddress,
    decoration: InputDecoration(
```

```
      hintText: "Email Address",
      border: OutlineInputBorder(borderRadius: BorderRadius.circular(8)),
    ),
  ),
),

const SizedBox(height: 12),

// Mobile Number Input
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: TextField(
    controller: mobileController,
    keyboardType: TextInputType.phone,
    decoration: InputDecoration(
      hintText: "Mobile Number",
      border: OutlineInputBorder(borderRadius: BorderRadius.circular(8)),
    ),
  ),
),

const SizedBox(height: 12),

// Password Input
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: TextField(
    controller: passwordController,
    obscureText: true,
    decoration: InputDecoration(
      hintText: "Password",
      border: OutlineInputBorder(borderRadius: BorderRadius.circular(8)),
    ),
  ),
),
```

```
          const SizedBox(height: 20),


          // Sign Up Button
          Padding(
            padding: const EdgeInsets.symmetric(horizontal: 16.0),
            child: SizedBox(
              width: double.infinity,
              child: ElevatedButton(
                onPressed: () {
                  Navigator.pushNamed(context, '/home');
                },
                style: ElevatedButton.styleFrom(
                  backgroundColor: Colors.pink.shade300,
                  padding: const EdgeInsets.symmetric(vertical: 16),
                ),
                child: Text("SIGN UP", style: TextStyle(fontSize: 16, color: Colors.white, fontWeight:
FontWeight.bold)),
              ),
            ),
          ),
        ],
      ),
    ),
  );
 }
}
```


Android Emulator - flutter_emulator:5554
10:51
WELCOME TO MYNTRA
FLAT ₹200 O
+ Free Shipping On Your 1st Ord
Sign Up Now
Create Account
Full Name
Email Address
Mobile Number
Password
SIGN UP
Already have an account? Login

Project Title: **Roll No.**

MAD & PWA Lab
Journal

Experiment No.	05
Experiment Title.	To apply navigation, routing and gestures in Flutter App
Roll No.	38
Name	Snehal Anilkumar Patil
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

Snehal Patil D15A 38

EXPERIMENT NO:05

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

Theory:

Navigation, Routing, and Gestures in Flutter App Development

Flutter provides a powerful and flexible way to manage navigation, routing, and user gestures in mobile applications. Understanding these concepts is crucial for creating smooth, user-friendly, and efficient applications. Below is a detailed explanation of each aspec

Navigation in Flutter

Navigation in Flutter refers to moving from one screen (or page) to another within the application. Flutter uses the concept of a **Navigator** and **Routes** to handle navigation.

Navigator Class

Flutter provides various methods for navigation:

The Navigator class manages a stack of routes (screens) and allows transitioning between them. Flutter uses a stack-based approach where each new screen is pushed onto the stack, and when you go back, it is popped from the stack.

Basic Navigation Methods

- **push()**: Adds a new route to the navigation stack.
- **pop()**: Removes the current route and returns to the previous screen.
- **pushReplacement()**: Replaces the current route with a new route.
- **pushAndRemoveUntil()**: Pushes a new route and removes all previous routes until a condition is met.

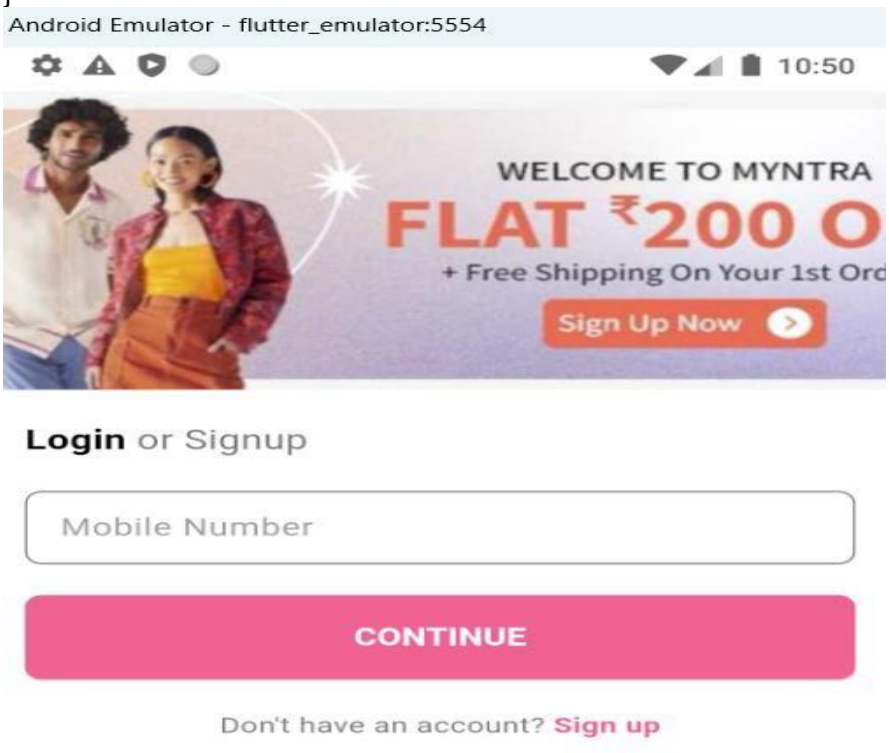
Code :

Login page:

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

class LoginPage extends StatelessWidget {
  final TextEditingController mobileController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: SafeArea(
        child: Column(
          children: [
            // Full Banner Section with Image
            Container(
              width: double.infinity,
              height: 200,
              decoration: BoxDecoration(
                gradient: LinearGradient(
                  colors: [Colors.yellow.shade100, Colors.pink.shade100],
                  begin: Alignment.topLeft,
                  end: Alignment.bottomRight,
                ),
              ),
              child: Image.asset(
                'assets/images/shopping_cart.png', // Replace with your image
                fit: BoxFit.cover, // Ensures the image covers the entire container
              ),
            ),

            const SizedBox(height: 20),

            // Login or Signup Text
            Padding(
              padding: const EdgeInsets.symmetric(horizontal: 16.0),
              child: Row(
                children: [
                  Text(
                    "Login ",
                    style: TextStyle(
                      fontSize: 18,
                      fontWeight: FontWeight.bold,
                      color: Colors.black,
                    ),
                  ),
                  Text(
                    "or Signup",
                    style: TextStyle(
                      fontSize: 18,
                      color: Colors.black54,
                    ),
                  ),
                ],
              ),
            ),
```

```
const SizedBox(height: 20),

// Input Field
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: TextField(
    controller: mobileController,
    keyboardType: TextInputType.phone,
    decoration: InputDecoration(
      hintText: "Mobile Number",
      hintStyle: TextStyle(color: Colors.grey),
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
      ),
      focusedBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
        borderSide: BorderSide(
          color: Colors.pink.shade300,
          width: 2,
        ),
      ),
      contentPadding: const EdgeInsets.symmetric(horizontal: 16),
    ),
  ),
),

const SizedBox(height: 20),

// Continue Button
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: SizedBox(
    width: double.infinity,
    child: ElevatedButton(
      onPressed: () {
        // Navigate to Home Page
        Navigator.pushNamed(context, '/home');
      },
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.pink.shade300,
        padding: const EdgeInsets.symmetric(vertical: 16),
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(8),
        ),
      ),
      child: Text(
        "CONTINUE",
        style: TextStyle(
          fontSize: 16,
          color: Colors.white,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
  ),
),

const SizedBox(height: 20),

// Don't have an account? Sign up
```

```
          Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text(
                "Don't have an account? ",
                style: TextStyle(color: Colors.black54),
              ),
              GestureDetector(
                onTap: () {
                  // Navigate to Sign Up Page
                  Navigator.pushNamed(context, '/signup');
                },
                child: Text(
                  "Sign up",
                  style: TextStyle(
                    color: Colors.pink.shade300,
                    fontWeight: FontWeight.bold,
                  ),
                ),
              ),
            ],
          ),
        ],
      ),
    ),
  );
 }
}
```



On click on login button it navigates to Home Page

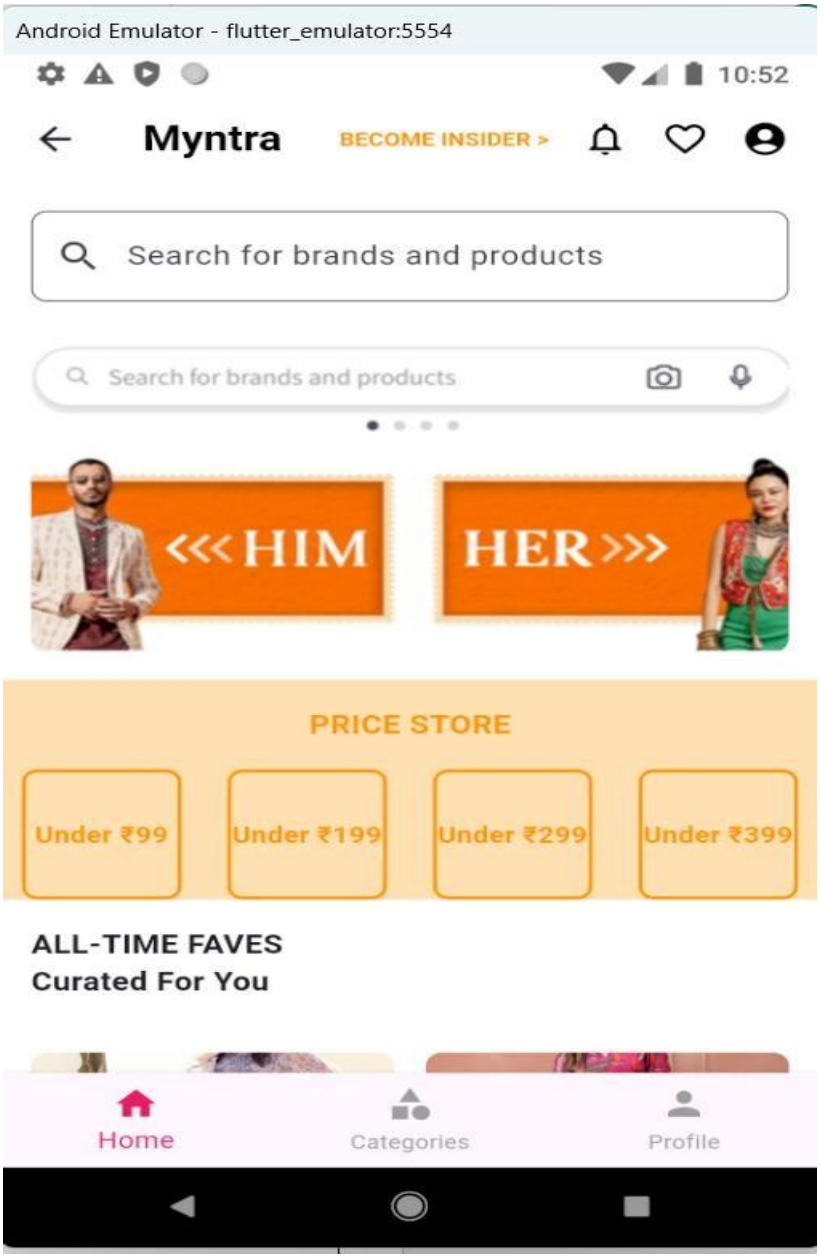
Home Page:

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

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Colors.white,
        elevation: 0,
        title: Row(
          children: [
            Text(
              "Myntra",
              style: TextStyle(
                fontWeight: FontWeight.bold,
                color: Colors.black,
              ),
            ),
            Spacer(),
            Text(
              "BECOME INSIDER >",
              style: TextStyle(
                color: Colors.orange,
                fontWeight: FontWeight.bold,
                fontSize: 12,
              ),
            ),
          ],
        ),
        actions: [
          Icon(Icons.notifications_none, color: Colors.black),
          SizedBox(width: 16),
          Icon(Icons.favorite_border, color: Colors.black),
          SizedBox(width: 16),
          Icon(Icons.account_circle, color: Colors.black),
          SizedBox(width: 16),
        ],
      ),
      body: SingleChildScrollView(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            // Search Bar
            Padding(
              padding: const EdgeInsets.all(16.0),
              child: TextField(
                decoration: InputDecoration(
                  hintText: "Search for brands and products",
                  prefixIcon: Icon(Icons.search),
                  border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(8),
                  ),
                ),
```

```
          ),
        ),

        // HIM / HER Banner
        Padding(
          padding: const EdgeInsets.symmetric(horizontal: 16.0),
          child: ClipRRect(
            borderRadius: BorderRadius.circular(10),
            child: Image.asset(
              "assets/images/1.png",
              width: double.infinity, // Full width
              height: 200, // Fixed height
              fit: BoxFit.cover, // Ensures proper coverage
            ),
          ),
        ),

        SizedBox(height: 16),

        // Price Store Section
        _buildPriceStore(),

        // All-Time Faves Section
        Padding(
          padding: const EdgeInsets.all(16.0),
          child: Text(
            "ALL-TIME FAVES\nCurated For You",
            style: TextStyle(
              fontWeight: FontWeight.bold,
              fontSize: 16,
            ),
          ),
        ),
        _buildFavesGrid(),
      ],
    ),
  ),

  // Bottom Navigation Bar
  bottomNavigationBar: BottomNavigationBar(
    type: BottomNavigationBarType.fixed,
    selectedItemColor: Colors.pink,
    unselectedItemColor: Colors.grey,
    items: [
      BottomNavigationBarItem(
        icon: Icon(Icons.home),
        label: "Home",
      ),
      BottomNavigationBarItem(
        icon: Icon(Icons.category),
        label: "Categories",
      ),
      BottomNavigationBarItem(
        icon: Icon(Icons.person),
        label: "Profile",
      ),
    ],
    onTap: (index) {
      String currentRoute = ModalRoute.of(context)?.settings.name ?? "";
      String newRoute = "";
```

```
        switch (index) {
          case 0:
            newRoute = '/home';
            break;
          case 1:
            newRoute = '/categories';
            break;
          case 2:
            newRoute = '/profile';
            break;
        }

        if (currentRoute != newRoute) {
          Navigator.pushNamed(context, newRoute);
        }
      },
    ),
  );
}

Widget _buildPriceStore() {
  return Container(
    color: Colors.orange.shade100,
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.center,
      children: [
        Padding(
          padding: const EdgeInsets.all(16.0),
          child: Text(
            "PRICE STORE",
            style: TextStyle(
              fontWeight: FontWeight.bold,
              color: Colors.orange,
              fontSize: 16,
            ),
          ),
        ),
        Row(
          mainAxisAlignment: MainAxisAlignment.spaceAround,
          children: [
            _buildPriceTile("Under ₹99"),
            _buildPriceTile("Under ₹199"),
            _buildPriceTile("Under ₹299"),
            _buildPriceTile("Under ₹399"),
          ],
        ),
      ],
    ),
  );
}

Widget _buildPriceTile(String text) {
  return Container(
    height: 80,
    width: 80,
    decoration: BoxDecoration(
      border: Border.all(color: Colors.orange, width: 2),
      borderRadius: BorderRadius.circular(8),
    ),
```

```
      child: Center(
        child: Text(
          text,
          textAlign: TextAlign.center,
          style: TextStyle(
            fontWeight: FontWeight.bold,
            color: Colors.orange,
          ),
        ),
      ),
    );
  }

  Widget _buildFavesGrid() {
    return Padding(
      padding: const EdgeInsets.all(16.0),
      child: GridView.count(
        shrinkWrap: true,
        crossAxisCount: 2,
        crossAxisSpacing: 16,
        mainAxisSpacing: 16,
        childAspectRatio: 0.75, // Adjusted for better image fit
        physics: NeverScrollableScrollPhysics(),
        children: [
          _buildFaveItem("Under ₹799", "assets/images/item1.png"),
          _buildFaveItem("Under ₹699", "assets/images/item2.png"),
          _buildFaveItem("Under ₹599", "assets/images/item3.png"),
          _buildFaveItem("Under ₹499", "assets/images/item4.png"),
        ],
      ),
    );
  }

  Widget _buildFaveItem(String price, String imagePath) {
    return Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        ClipRRect(
          borderRadius: BorderRadius.circular(8),
          child: Image.asset(
            imagePath,
            height: 140, // Increased height for better visibility
            width: double.infinity,
            fit: BoxFit.cover, // Ensures the image fits well
          ),
        ),
        SizedBox(height: 8),
        Text(
          price,
          style: TextStyle(
            fontWeight: FontWeight.bold,
          ),
        ),
      ],
    );
  }
}
```

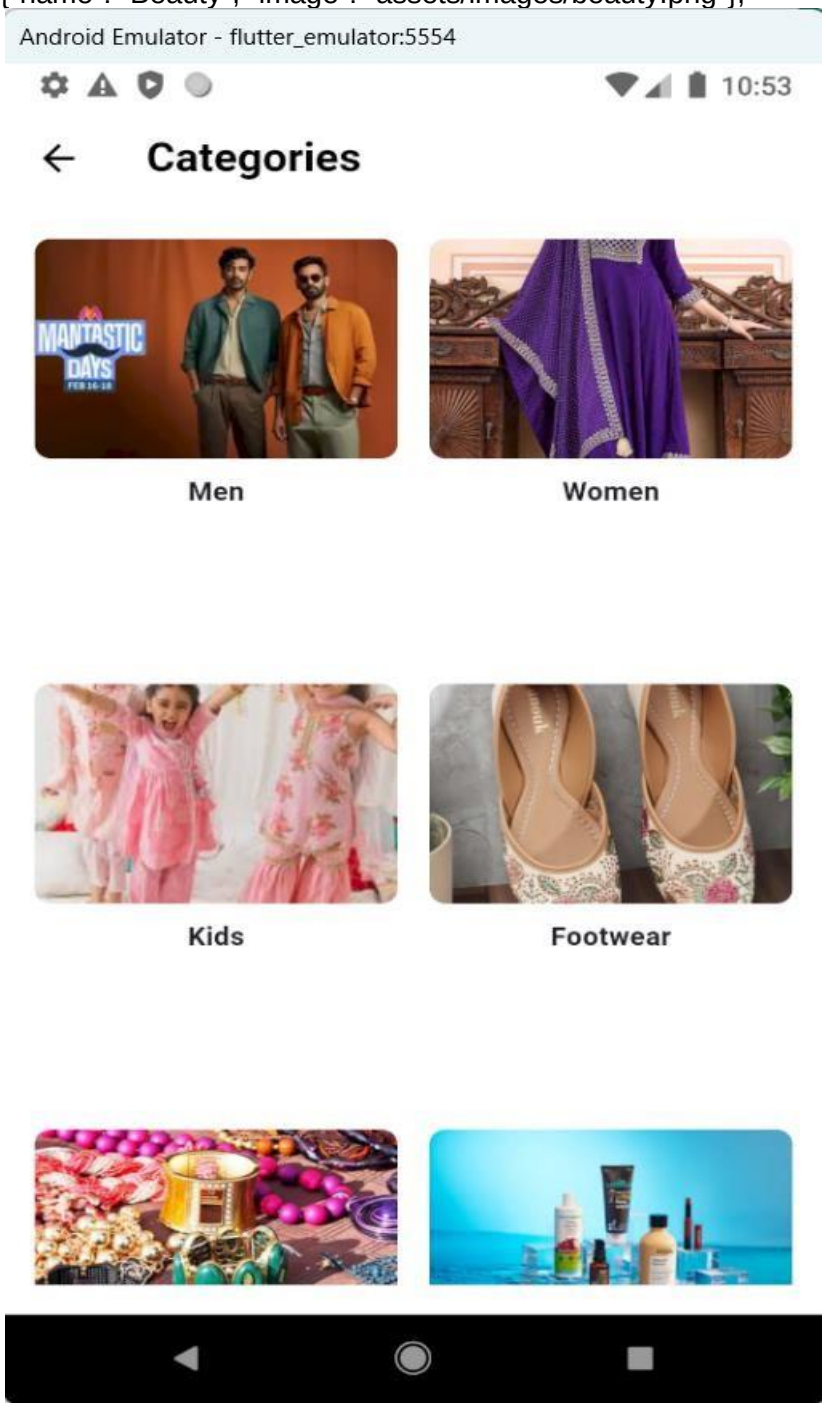


On click on categories at bottom it navigates to categories page:

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

class CategoriesPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Colors.white,
        elevation: 0,
        title: Text(
          "Categories",
          style: TextStyle(
            fontWeight: FontWeight.bold,
            color: Colors.black,
          ),
        ),
        leading: IconButton(
```

```
        icon: Icon(Icons.arrow_back, color: Colors.black),
        onPressed: () {
          Navigator.pop(context);
        },
      ),
    ),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: GridView.builder(
        gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
          crossAxisCount: 2,
          crossAxisSpacing: 16,
          mainAxisSpacing: 16,
          childAspectRatio: 0.8,
        ),
        itemCount: categories.length,
        itemBuilder: (context, index) {
          return _buildCategoryTile(
            categories[index]['name']!,
            categories[index]['image']!,
          );
        },
      ),
    ),
  );
}

Widget _buildCategoryTile(String name, String imagePath) {
  return GestureDetector(
    onTap: () {
      print("$name category clicked");
    },
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.center,
      children: [
        ClipRRect(
          borderRadius: BorderRadius.circular(8),
          child: Image.asset(
            imagePath,
            height: 120,
            width: double.infinity,
            fit: BoxFit.cover,
            errorBuilder: (context, error, stackTrace) {
              return Container(
                height: 120,
                width: double.infinity,
                color: Colors.grey[300],
                child: Icon(Icons.image, size: 50, color: Colors.grey[600]),
              );
            },
          ),
        ),
        SizedBox(height: 8),
        Text(
          name,
          style: TextStyle(
            fontWeight: FontWeight.bold,
          ),
        ),
      ],
```

```
      ),
    );
  }
}

// Category List with Names and Image Paths
final List<Map<String, String>> categories = [
  {"name": "Men", "image": "assets/images/men.png"},
  {"name": "Women", "image": "assets/images/women.png"},
  {"name": "Kids", "image": "assets/images/kids.png"},
  {"name": "Footwear", "image": "assets/images/footwear.png"},
  {"name": "Accessories", "image": "assets/images/accessories.png"},
  {"name": "Beauty", "image": "assets/images/beauty.png"},
```




```
];
```

Project Title: **Roll No.**

MAD & PWA Lab
Journal

Experiment No.	06
Experiment Title.	To Connect Flutter UI with fireBase database
Roll No.	38
Name	Snehal Anilkumar Patil
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO3: Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS
Grade:	

EXPERIMENT NO: - 06

Name:-Snehal Patil **Class:-** D15A **Roll:No: -** 38

AIM: - To connect Flutter UI with Firebase database.

Theory: -

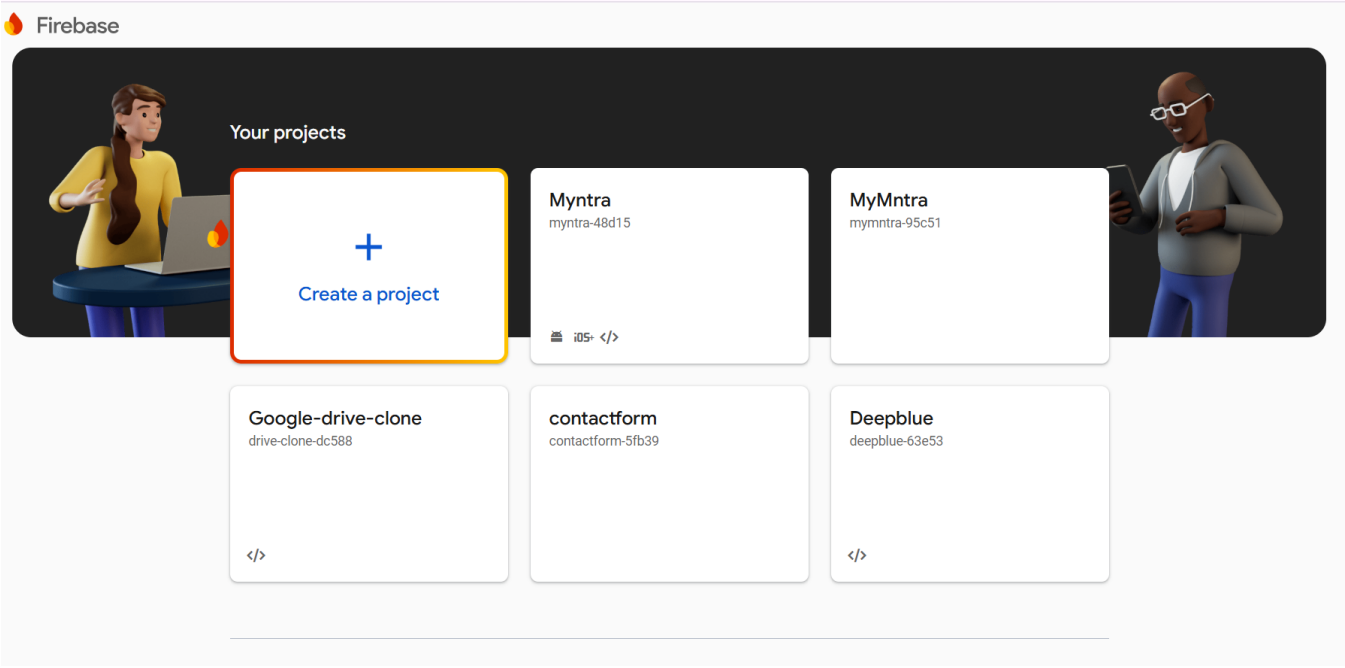
Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. Firebase, a Backend-as-a-Service (BaaS) platform, provides real-time database, authentication, and cloud storage services, making it a powerful backend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

By integrating Firebase with Flutter, developers can store and retrieve data in real time, authenticate users, and manage cloud-based data efficiently. This is particularly useful for applications requiring dynamic content updates and user interactions.

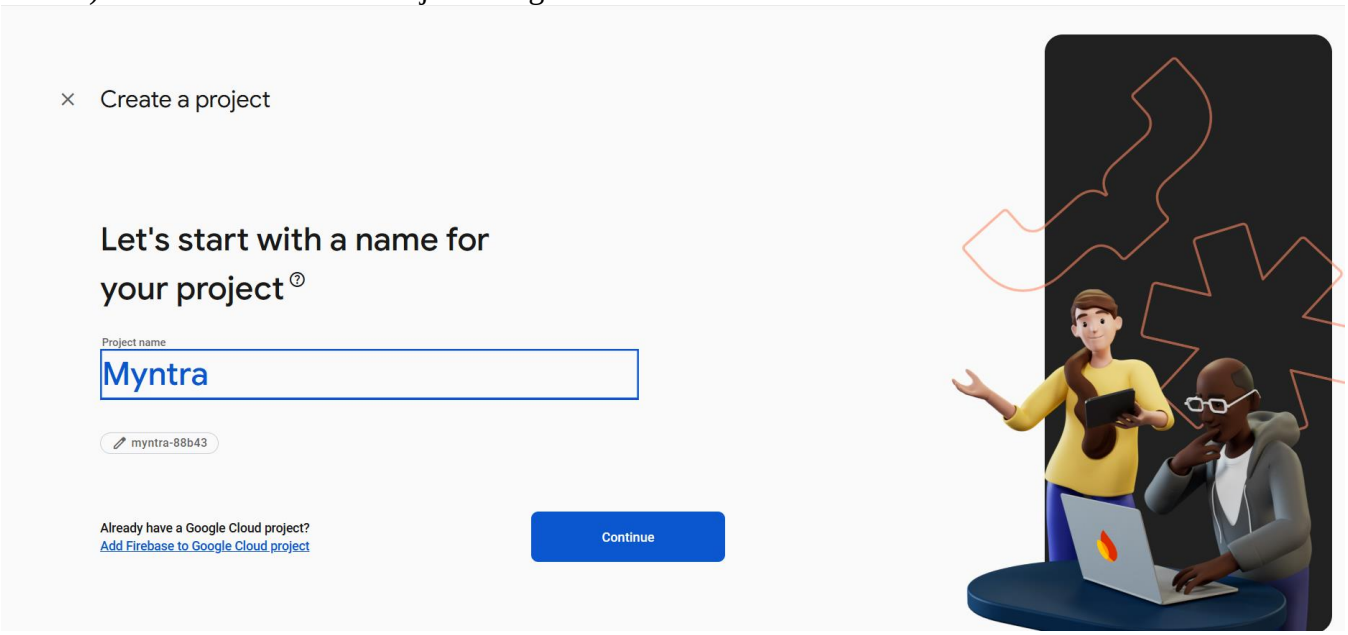
➢ Steps to Connect Flutter UI with Firebase Database

Step 1:

1.1) Go to Firebase Console and Create a Firebase Project

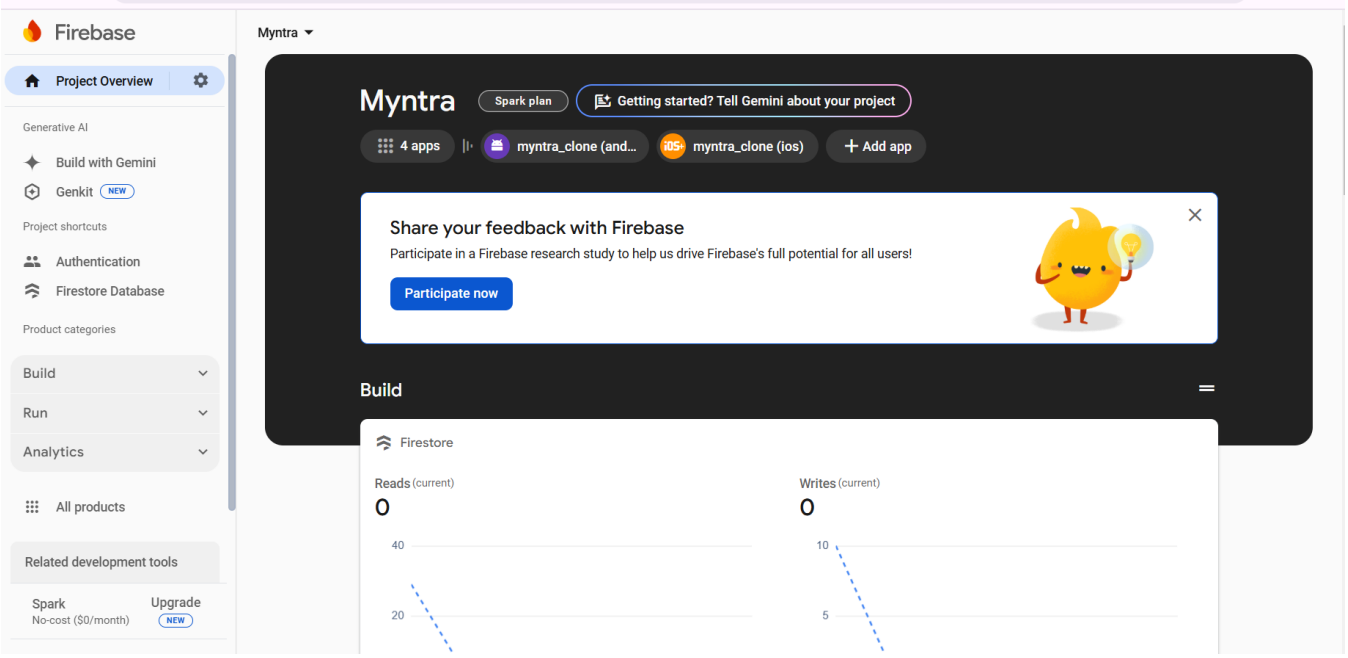


1.2) Click on Create a Project and give it a suitable name.



Step 2:- Add Firebase to Your Flutter App

2.1) Click on Android/iOS/Web based on your Flutter application

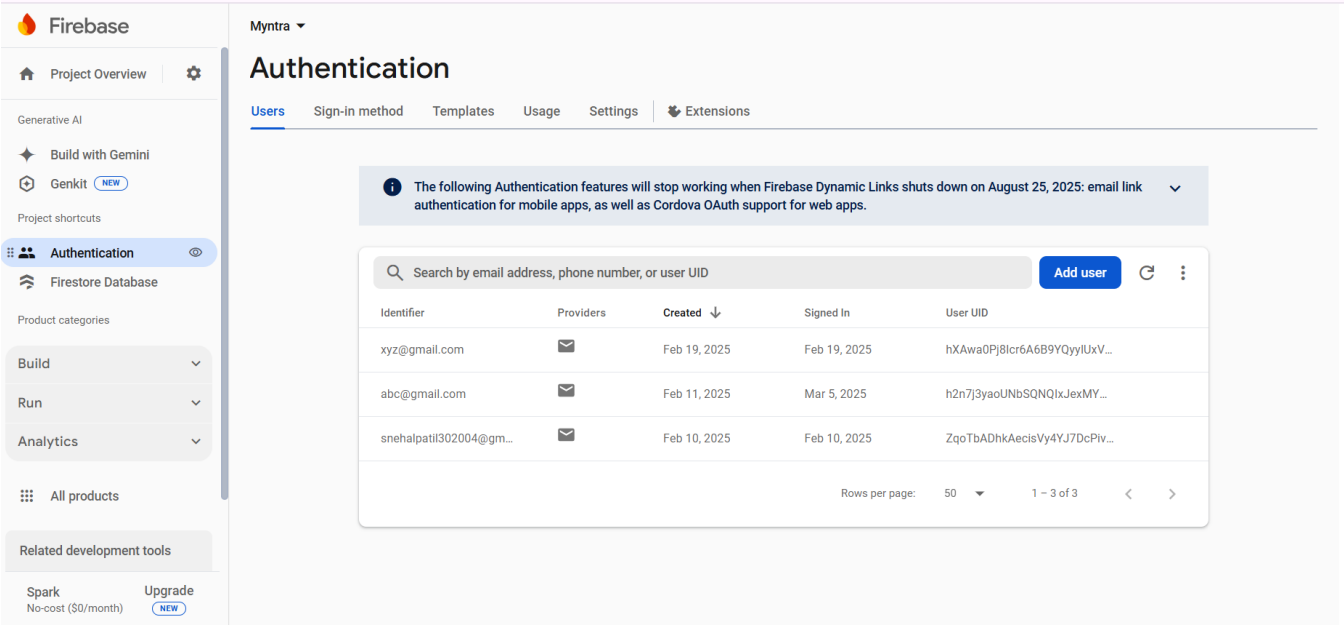


Step 3: - Add Firebase Authentication to Your App

3.1) Add Firebase Authentication Dependencies

```yaml
dependencies:
  flutter:
    sdk: flutter
  firebase_core: ^3.11.0
  firebase_auth: ^5.4.2  # For authentication
  cloud_firestore: ^5.6.3  # For Firestore, if you need it
  firebase_messaging: ^15.2.2
  http: ^0.13.3
  image_picker: ^1.0.4
  tflite_flutter: ^0.11.0
  image: ^3.2.0
  url_launcher: ^6.1.14
```

3.2) Enable Authentication in Firebase Console
Go to **Firebase Console → Authentication**.
Click on **Sign-in method** and enable **Email/Password** (or any other method like Google).
Click Save



3.3) Implement Authentication in Flutter
Modify main.dart

```
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_auth/firebase_auth.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(MyApp());
}
```

Step 4: -Configure Firebase Realtime Database

4.1) Go to Firebase Console → Realtime Database.
4.2) Click **Create Database** → Choose location → Set rules (for development, set read/write to true).
4.3) Click **Publish**.

Code:-

Sign_up_page.dart

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

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

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

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

                const SizedBox(height: 20),

                // Sign Up Text
                Row(
                  children: [
                    Text(
                      "Create ",
                      style: TextStyle(
                        fontSize: 18,
                        fontWeight: FontWeight.bold,
                        color: Colors.black,
                      ),
                    ),
                    Text(
                      "Account",
                      style: TextStyle(
                        fontSize: 18,
                        color: Colors.black54,
                      ),
                    ),
                  ],
                ),

                const SizedBox(height: 20),

                // Name Field
                TextField(
```

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

          const SizedBox(height: 12),

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

          const SizedBox(height: 12),

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

          const SizedBox(height: 12),

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

```
EdgeInsets.symmetric(horizontal: 16),
                ),
              ),

              const SizedBox(height: 20),

              // Sign Up Button
              SizedBox(
                width: double.infinity,
                child: ElevatedButton(
                  onPressed: _isLoading ? null :
_signUpUser,
                  style: ElevatedButton.styleFrom(
                    backgroundColor:
Colors.pink.shade300,
                    padding: const
EdgeInsets.symmetric(vertical: 16),
                    shape: RoundedRectangleBorder(
                      borderRadius:
BorderRadius.circular(8),
                    ),
                  ),
                  child: _isLoading
                      ?
CircularProgressIndicator(color: Colors.white)
                      : Text(
                          "SIGN UP",
                          style: TextStyle(
                            fontSize: 16,
                            color: Colors.white,
                            fontWeight:
FontWeight.bold,
                          ),
                        ),
                ),
              ),

              const SizedBox(height: 20),

              // Already have an account? Login
              Row(
                mainAxisAlignment:
MainAxisAlignment.center,
                children: [
                  Text(
                    "Already have an account? ",
                    style: TextStyle(color:
Colors.black54),
                  ),
                  GestureDetector(
                    onTap: () {
                      Navigator.pop(context);
                    },
                    child: Text(
                      "Login",
                      style: TextStyle(
                        color: Colors.pink.shade300,
                        fontWeight: FontWeight.bold,
                      ),
                    ),
                  ),
                ],
              ),
            ],
          ),
        ),
      ),
    );
  }

  // Function to handle user sign up
  void _signUpUser() async {
    setState(() {
      _isLoading = true;
    });

    String name = nameController.text.trim();
    String email = emailController.text.trim();
    String mobile = mobileController.text.trim();
    String password =
passwordController.text.trim();

    if (name.isEmpty || email.isEmpty ||
mobile.isEmpty || password.isEmpty) {
      Fluttertoast.showToast(msg: "All fields are
required!");
      setState(() {
        _isLoading = false;
      });
      return;
    }

    if (!RegExp(r"^[a-zA-Z]+$").hasMatch(name)) {
      Fluttertoast.showToast(msg: "Enter a valid
name");
      setState(() {
        _isLoading = false;
      });
      return;
    }
```

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

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

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

Sign In Page:

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import
'package:cloud_firestore/cloud_firestore.dart'; //  🔹
Firestore Import
import 'package:fluttertoast/fluttertoast.dart';
import
'package:myntra_clone/pages/home_page.dart';

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

class _LoginPageState extends State<LoginPage> {
  final TextEditingController emailController =
TextEditingController();
  final TextEditingController passwordController =
TextEditingController();
  final FirebaseAuth _auth = FirebaseAuth.instance;
  final FirebaseFirestore _firestore =
FirebaseFirestore.instance; //  🔹  Firestore Instance
  bool _isLoading = false;

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

```
            ),
            child: Image.asset(
              'assets/images/shopping_cart.png',
              fit: BoxFit.cover,
            ),
          ),

          const SizedBox(height: 20),

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

          const SizedBox(height: 20),

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

          const SizedBox(height: 12),

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

          const SizedBox(height: 20),

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

```
                ),
              ),
              child: _isLoading
                  ?
CircularProgressIndicator(color: Colors.white)
                  : Text(
                      "LOGIN",
                      style: TextStyle(
                        fontSize: 16,
                        color: Colors.white,
                        fontWeight:
FontWeight.bold,
                      ),
                    ),
            ),
          ),

          const SizedBox(height: 20),

          // Don't have an account? Sign up
          Row(
            mainAxisAlignment:
MainAxisAlignment.center,
            children: [
              Text(
                "Don't have an account? ",
                style: TextStyle(color:
Colors.black54),
              ),
              GestureDetector(
                onTap: () {
                  Navigator.pushNamed(context,
'/signup');
                },
                child: Text(
                  "Sign up",
                  style: TextStyle(
                    color: Colors.pink.shade300,
                    fontWeight: FontWeight.bold,
                  ),
                ),
              ),
            ],
          ),
        ],
      ),
    ),
  );
 }
```

```
 // Function to handle user login
 void _loginUser() async {
  setState(() {
   _isLoading = true;
  });

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

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

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

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

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

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

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

```
    DocumentReference userRef =
_firestore.collection('users').doc(user.uid);
    DocumentSnapshot userDoc = await
userRef.get();

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

Main.dart

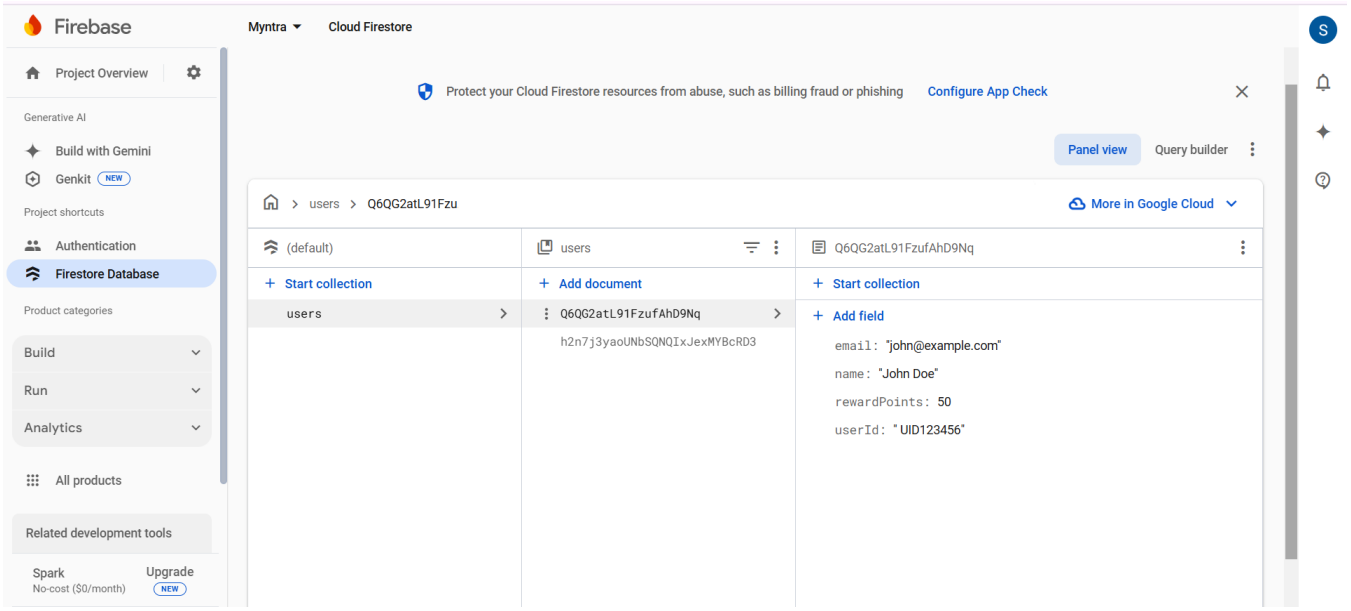
```
import 'package:flutter/material.dart';
import
'package:firebase_core/firebase_core.dart';
import
'package:myntra_clone/pages/login_page.dart';
// import
'package:myntra_clone/pages/productDetails_pa
ge.dart';
import
'package:myntra_clone/pages/profile_page.dart';
import
'package:myntra_clone/pages/signup_page.dart';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

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

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

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

      },
    );
  }
}
```

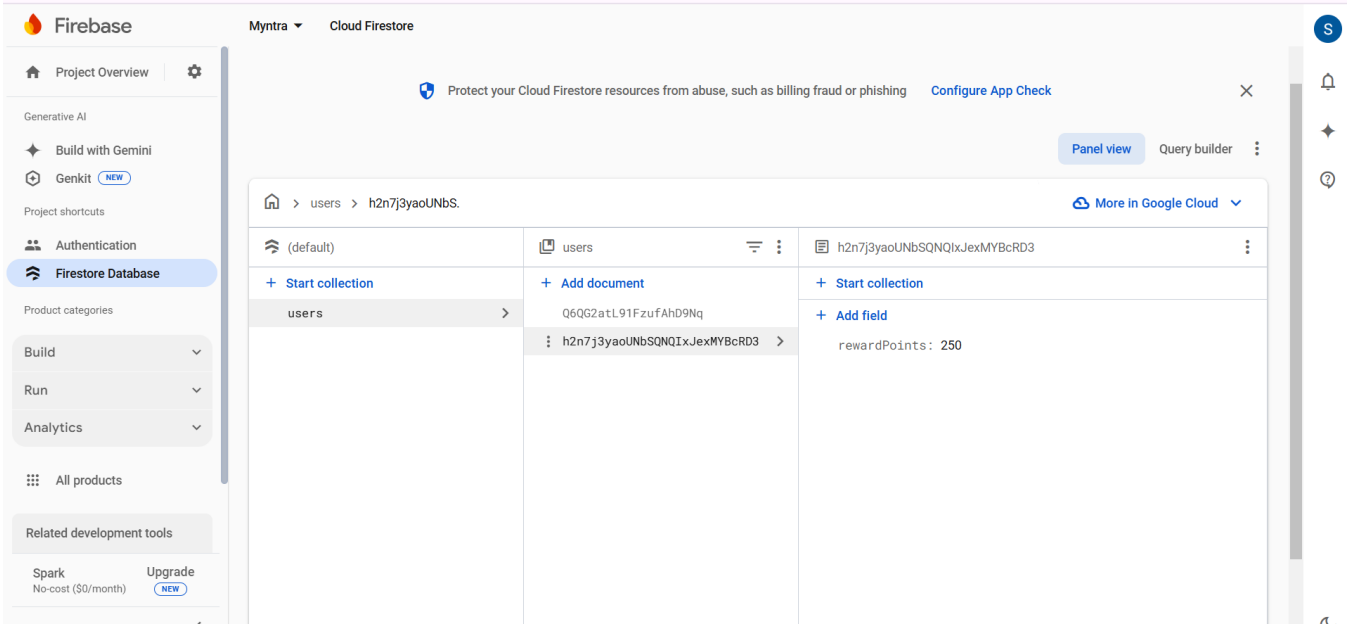
Checkout page:

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

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

  CheckoutPage({required this.product});

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

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

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

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

  /// Fetches the user's current reward points
from Firestore
  void _fetchUserPoints() async {
    final user =
FirebaseAuth.instance.currentUser;
    if (user != null) {
      final userDoc = await
FirebaseFirestore.instance.collection('users').doc
(user.uid).get();
      setState(() {
        userPoints = userDoc.exists ?
(userDoc.data()?['rewardPoints'] ?? 0) : 0;
      });
    }
  }

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

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

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

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

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

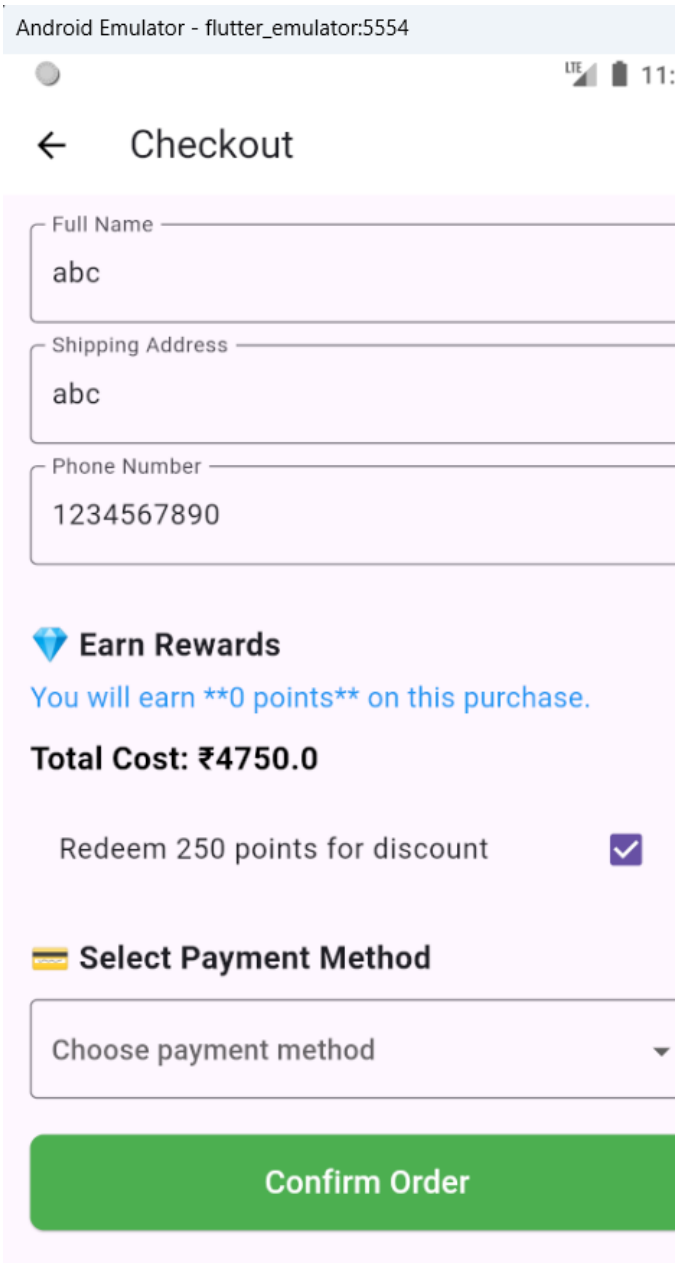
```
method.");
      }

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

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

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

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

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

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Checkout"),
        backgroundColor: Colors.white,
        elevation: 0,
        leading: IconButton(
          icon: Icon(Icons.arrow_back, color:
Colors.black),
          onPressed: () => Navigator.pop(context),
        ),
      ),
      body: SingleChildScrollView(
        padding: EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment:
CrossAxisAlignment.start,
          children: [
            _buildInputField("Full Name",
nameController),
            _buildInputField("Shipping Address",
addressController),
            _buildInputField("Phone Number",
phoneController, keyboardType:
TextInputType.phone),
            SizedBox(height: 20),

            // Earned Rewards Section
            Text("💎 Earn Rewards", style:
TextStyle(fontSize: 18, fontWeight:
FontWeight.bold)),
            SizedBox(height: 5),
            Text("You will earn **$rewardPoints
points** on this purchase.", style:
TextStyle(fontSize: 16, color: Colors.blue)),
            SizedBox(height: 10),

            // Display Total Cost
            Text(
              "Total Cost: ₹$totalCost",
              style: TextStyle(fontSize: 18,
fontWeight: FontWeight.bold, color:
Colors.black),
            ),
            SizedBox(height: 10),

            // Checkbox for redeeming points
            CheckboxListTile(
              title: Text("Redeem $userPoints points
for discount"),
              value: useRewards,
              onChanged: (bool? value) {
                setState(() {
                  useRewards = value ?? false;
```

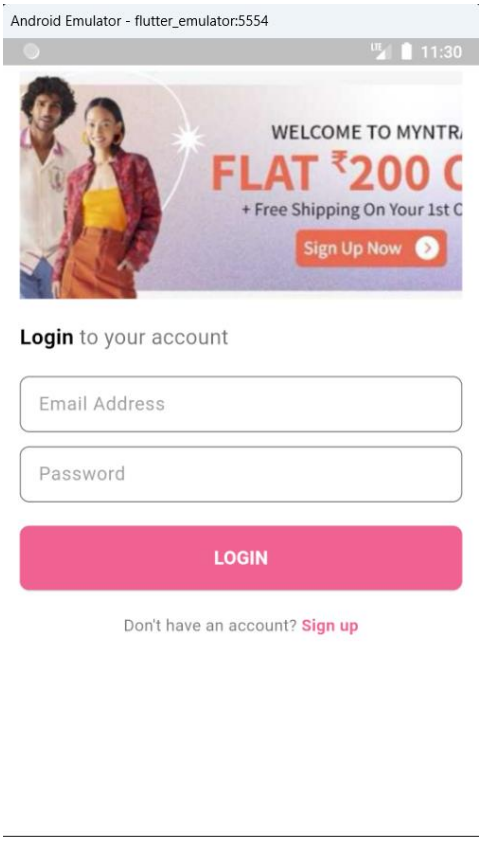
```
            totalCost = useRewards
                ? (widget.product['price'] -
userPoints).clamp(0,
widget.product['price']).toDouble()
                :
widget.product['price'].toDouble();
            });
          },
        ),

        SizedBox(height: 20),

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

        // Confirm Order Button
        SizedBox(
          width: double.infinity,
          child: ElevatedButton(
            style: ElevatedButton.styleFrom(
              padding:
EdgeInsets.symmetric(vertical: 14),
              backgroundColor: Colors.green,
              shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(8)),
            ),
            onPressed: _placeOrder,
            child: Text("Confirm Order", style:
TextStyle(fontSize: 18, color: Colors.white)),
          ),
        ),
      ],
    ),
  ),
);
}

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

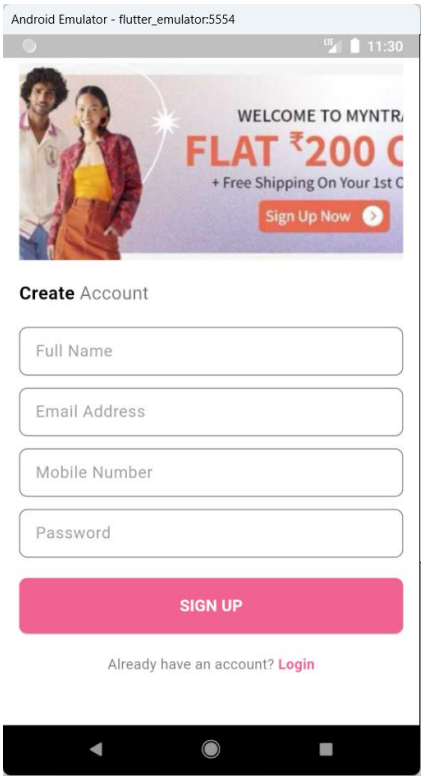

Firebase
Myntra
Cloud Firestore
Project Overview
Generative AI
Build with Gemini
Genkit NEW
Project shortcuts
Authentication
Firestore Database
Product categories
Build
Run
Analytics
All products
Related development tools
Spark
No-cost ($0/month)
Upgrade
NEW
Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing
Configure App Check
Panel view
Query builder
users > Q6QG2atL91Fzu
More in Google Cloud
(default)
users
Q6QG2atL91FzufAhD9Nq
Start collection
Add document
Add field
users
Q6QG2atL91FzufAhD9Nq
h2n7j3yaoUNbSQNQIxJexMYBcRD3
email: "john@example.com"
name: "John Doe"
rewardPoints: 50
userId: "UID123456"


Firebase
Myntra
Cloud Firestore
Project Overview
Generative AI
Build with Gemini
Genkit NEW
Project shortcuts
Authentication
Firestore Database
Product categories
Build
Run
Analytics
All products
Related development tools
Spark
No-cost ($0/month)
Upgrade
NEW
Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing
Configure App Check
Panel view
Query builder
users > h2n7j3yaoUNbS.
More in Google Cloud
(default)
users
h2n7j3yaoUNbSQNQIxJexMYBcRD3
Start collection
Add document
Add field
users
Q6QG2atL91FzufAhD9Nq
h2n7j3yaoUNbSQNQIxJexMYBcRD3
rewardPoints: 250

Checkout page:







Project Title: **Roll No.**

MAD & PWA Lab
Journal

Experiment No.	07
Experiment Title.	To write meta data of your Ecommerce PWA in a Web app manifest file to enable “add to homescreen feature”.
Roll No.	38
Name	Snehal Anilkumar Patil
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO4: Understand various PWA frameworks and their requirements
Grade:	

Experiment 7: Adding Metadata to Web App Manifest for PWA

Ronak Katariya 23

Prajjwal Pandey 32

Snehal Patil 38

Objective:

To write metadata for an eCommerce Progressive Web App (PWA) in a Web App Manifest file to enable the "Add to Home Screen" feature.

Theory:

Regular Web App
A regular web app is a website that is designed to be accessible on all mobile devices such that the content gets fit as per the device screen. It is designed using a web technology stack (HTML, CSS, JavaScript, Ruby, etc.) and operates via a browser. They offer various native-device features and functionalities. However, it entirely depends on the browser the user is using. In other words, it might be possible that you can access a native-device feature on Chrome but not on Safari or Mozilla Firefox because the browsers are incompatible with that feature.

Progressive Web App
Progressive Web App (PWA) is a regular web app, but some extras enable it to deliver an excellent user experience. It is a perfect blend of desktop and mobile application experience to give both platforms to the end-users.

Difference between PWAs vs. Regular Web Apps:
A Progressive Web is different and better than a Regular Web app with features like:
1. Native Experience
Though a PWA runs on web technologies (HTML, CSS, JavaScript) like a Regular web app, it gives user experience like a native mobile application. It can use most native device features, including push notifications, without relying on the browser or any other entity. It offers a seamless and integrated user experience that it is quite tough for one to differentiate between a PWA and a Native application by considering its look and feel.

2. Ease of Access
Unlike other mobile apps, PWAs do not demand longer download time and make memory space available for installing the applications. The PWAs can be shared and installed by a link, which cuts down the number of steps to install and use. These applications can easily keep an app

icon on the user's home screen, making the app easily accessible to the users and helps the brands remain in the users' minds, and improving the chances of interaction.

3. Faster Services
PWAs can cache the data and serve the user with text stylesheets, images, and other web content even before the page loads completely. This lowers the waiting time for the end-users and helps the brands improve the user engagement and retention rate, which eventually adds value to their business.

4. Engaging Approach
As already shared, the PWAs can employ push notifications and other native device features more efficiently. Their interaction does not depend on the browser user uses. This eventually improves the chances of notifying the user regarding your services, offers, and other options related to your brand and keeping them hooked to your brand. In simpler words, PWAs let you maintain the user engagement and retention rate.

5. Updated Real-Time Data Access
Another plus point of PWAs is that these apps get updated on their own. They do not demand the end-users to go to the App Store or other such platforms to download the update and wait until installed.
In this app type, the web app developers can push the live update from the server, which reaches the apps residing on the user's devices automatically. Therefore, it is easier for the mobile app developer to provide the best of the updated functionalities and services to the end-users without forcing them to update their app.
6. Discoverable
PWAs reside in web browsers. This implies higher chances of optimizing them as per the Search Engine Optimization (SEO) criteria and improving the Google rankings like that in websites and other web apps.

7. Lower Development Cost
Progressive web apps can be installed on the user device like a native device, but it does not demand submission on an App Store. This makes it far more cost-effective than native mobile applications while offering the same set of functionalities.

Pros and cons of the Progressive Web App
The main features are:

Progressive — They work for every user, regardless of the browser chosen because they are built at the base with progressive improvement principles.

Responsive — They adapt to the various screen sizes: desktop, mobile, tablet, or dimensions that can later become available.

App-like — They behave with the user as if they were native apps, in terms of interaction and navigation.

Updated — Information is always up-to-date thanks to the data update process offered by service workers.

Secure — Exposed over HTTPS protocol to prevent the connection from displaying information or altering the contents.

Searchable — They are identified as “applications” and are indexed by search engines.

Reactivable — Make it easy to reactivate the application thanks to capabilities such as web notifications.
Installable — They allow the user to “save” the apps that he considers most useful with the corresponding icon on the screen of his mobile terminal (home screen) without having to face all the steps and problems related to the use of the app store.

Linkable — Easily shared via URL without complex installations.

Offline — Once more it is about putting the user before everything, avoiding the usual error message in case of weak or no connection. The PWA are based on two particularities: first of all the ‘skeleton’ of the app, which recalls the page structure, even if its contents do not respond and its elements include the header, the page layout, as well as an illustration that signals that the page is loading.

Weaknesses refer to:
IOS support from version 11.3 onwards;
Greater use of the device battery;
Not all devices support the full range of PWA features (same speech for iOS and Android operating systems);
It is not possible to establish a strong re-engagement for iOS users (URL scheme, standard web notifications);
Support for offline execution is however limited;
Lack of presence on the stores (there is no possibility to acquire traffic from that channel);
There is no “body” of control (like the stores) and an approval process;
Limited access to some hardware components of the devices;
Little flexibility regarding “special” content for users (eg loyalty programs, loyalty, etc.).

Requirements:

- A code editor (e.g., VS Code, Sublime Text)
- A basic eCommerce web app
- A browser supporting PWA features (e.g., Chrome, Edge)

Procedure:

1. **Create the Manifest File:**
    - In the root directory of your eCommerce web app, create a file named `manifest.json`.
    - Add the following metadata:

```
{
  "name": "My eCommerce App",
  "short_name": "ShopApp",
  "description": "A fast and secure eCommerce PWA",
  "start_url": "/index.html",
  "display": "standalone",
  "background_color": "#ffffff",
  "theme_color": "#2196F3",
  "icons": [
    {
      "src": "/icons/icon-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "/icons/icon-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
}
```

2. **Link the Manifest File to HTML:**

- In the `<head>` section of `index.html`, add:

<link rel="manifest" href="/manifest.json">

4.
5. **Serve the App and Test Installation:**
    - Host the application using a local server (e.g., `Live Server` extension in VS Code or Node.js `http-server`).
    - Open the app in a PWA-compatible browser.
    - Check for the "Add to Home Screen" prompt or manually install it from the browser menu.
6. **Verify in DevTools:**
    - Open Chrome DevTools (`F12` or `Ctrl+Shift+I`).
    - Go to the "Application" tab → "Manifest".
    - Ensure all metadata is correctly loaded.

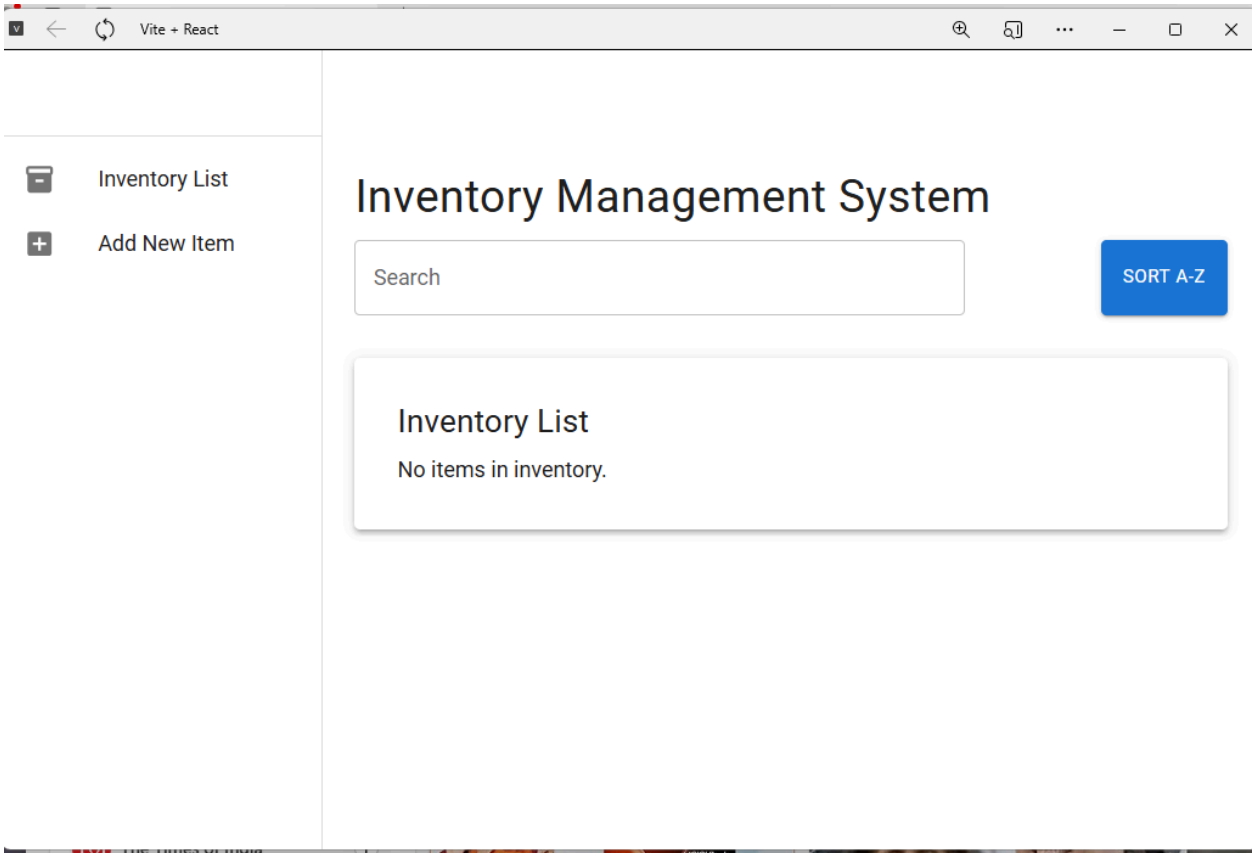
Observations:

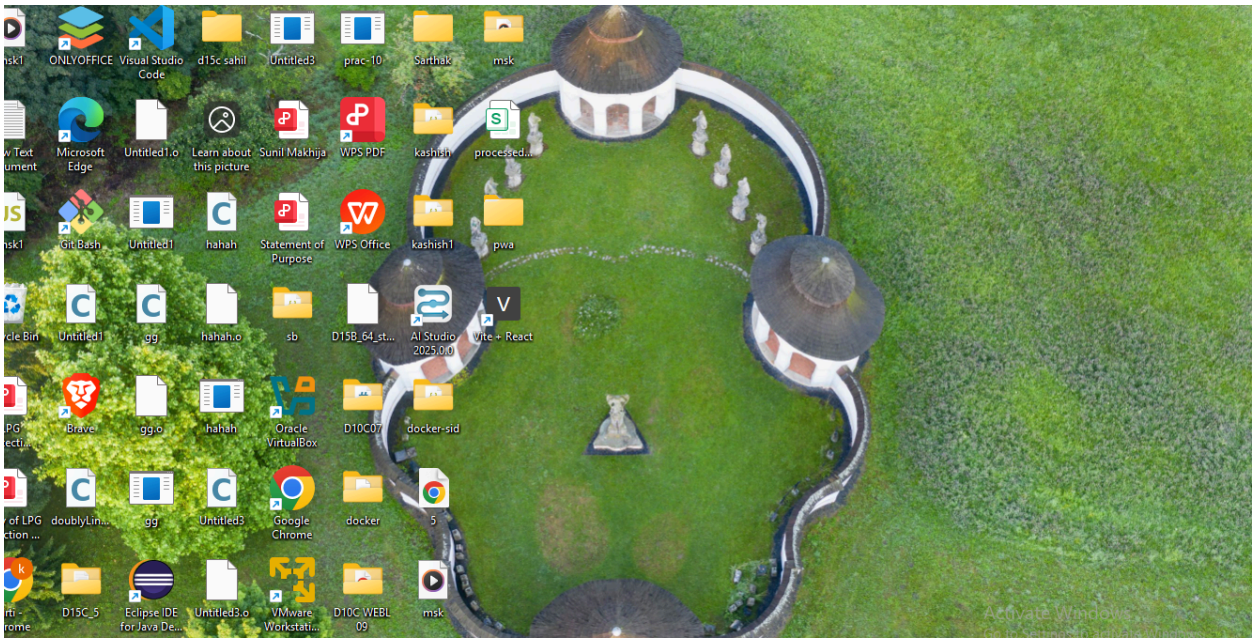
- The manifest file enables the PWA installation.
- The app runs in standalone mode without a browser UI.
- The defined icons and theme colors appear as expected.

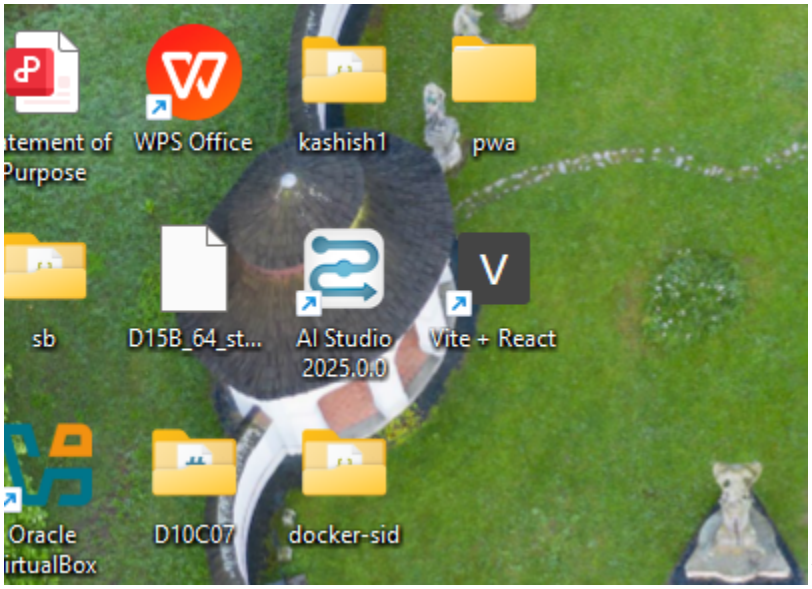
Conclusion:

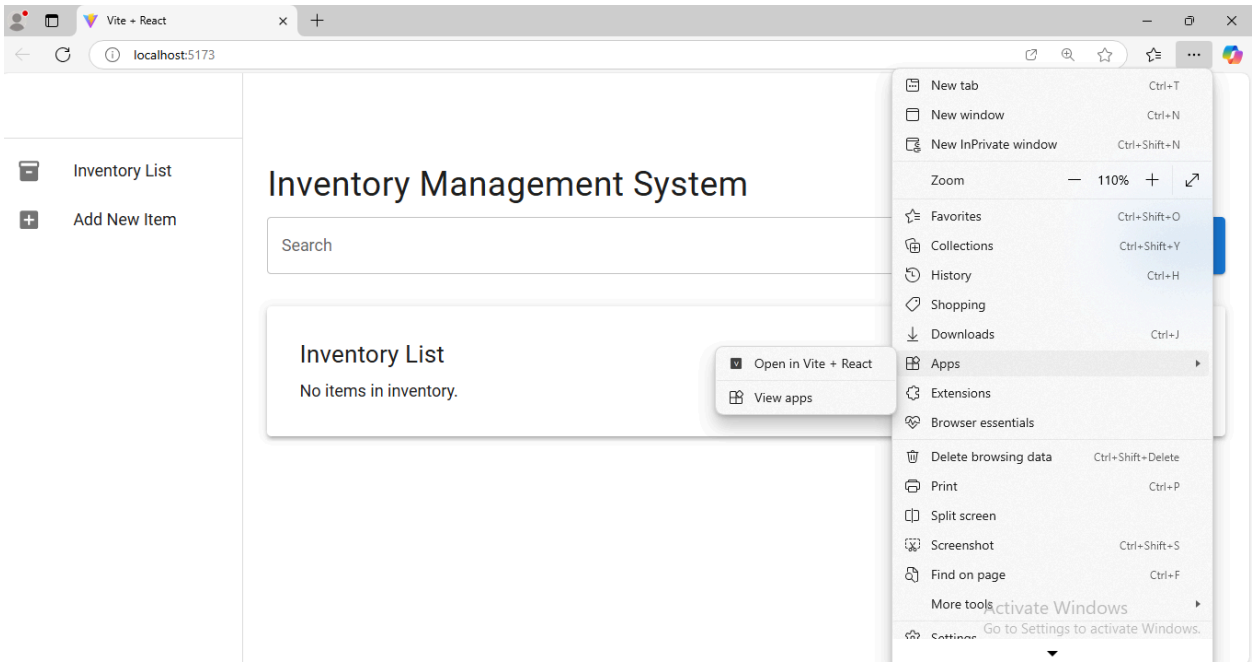
By adding a Web App Manifest file with proper metadata, the eCommerce PWA supports the "Add to Home Screen" feature, improving the user experience with an app-like interface.

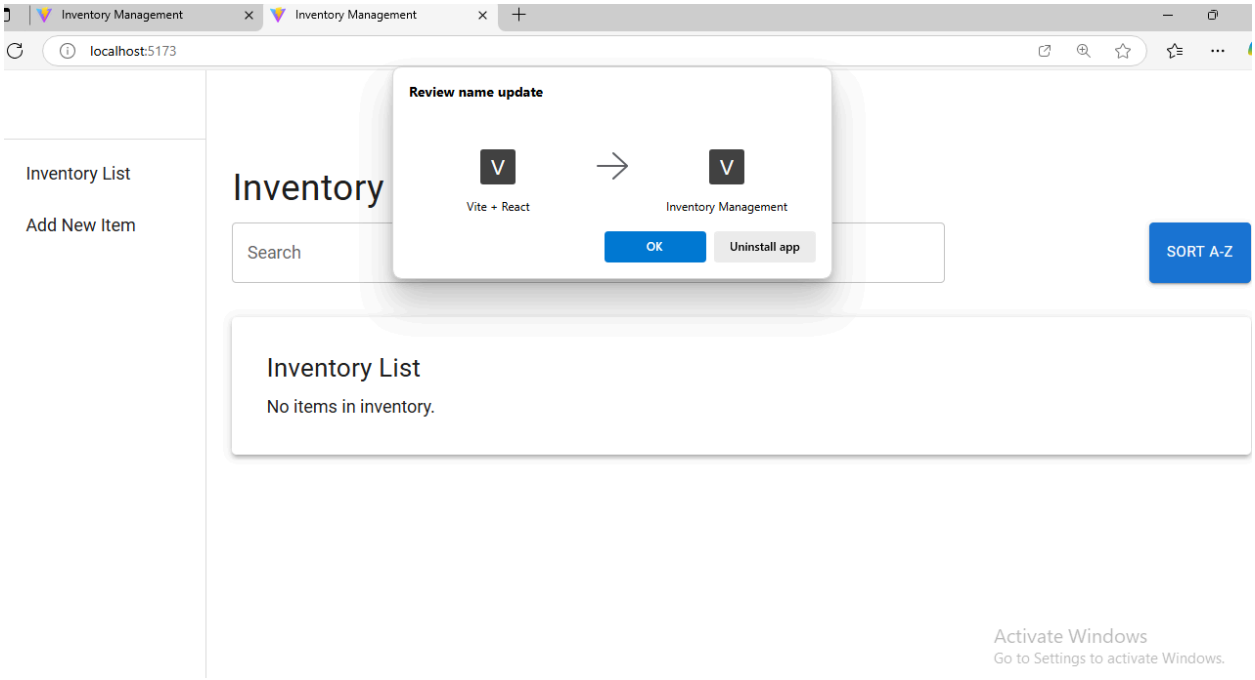
Screenshots:

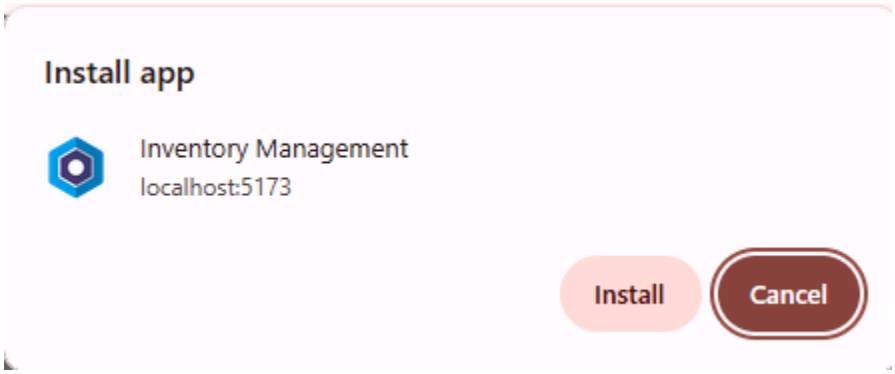


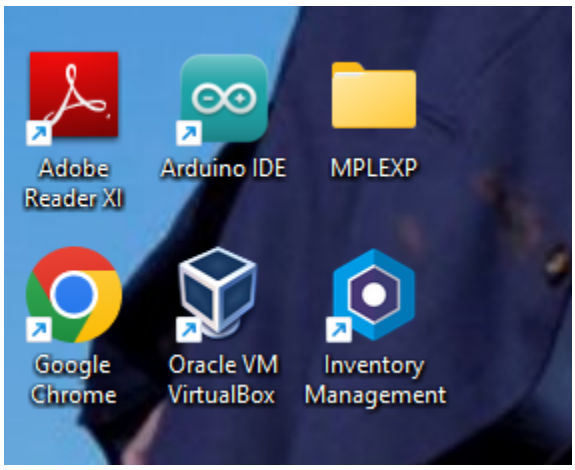



WPS Office
kashish1
pwa
sb
D15B_64_st...
AI Studio 2025.0.0
Vite + React
Oracle VirtualBox
D10C07
docker-sid
\


Vite + React
localhost:5173
Inventory List
Add New Item
Inventory Management System
Search
Inventory List
No items in inventory.
Open in Vite + React
View apps
New tab Ctrl+T
New window Ctrl+N
New InPrivate window Ctrl+Shift+N
Zoom 110%
Favorites Ctrl+Shift+O
Collections Ctrl+Shift+Y
History Ctrl+H
Shopping
Downloads Ctrl+J
Apps
Extensions
Browser essentials
Delete browsing data Ctrl+Shift+Delete
Print Ctrl+P
Split screen
Screenshot Ctrl+Shift+S
Find on page Ctrl+F
More tools
Activate Windows
Go to Settings to activate Windows.


Inventory Management
Inventory Management
localhost:5173
Review name update
Vite + React
Inventory Management
OK
Uninstall app
Inventory List
Add New Item
Inventory
Search
SORT A-Z
Inventory List
No items in inventory.
Activate Windows
Go to Settings to activate Windows.


Install app
Inventory Management
localhost:5173
Install
Cancel


Adobe Reader XI
Arduino IDE
MPLEXP
Google Chrome
Oracle VM VirtualBox
Inventory Management

Project Title: **Roll No.**

MAD & PWA Lab
Journal

Experiment No.	08
Experiment Title.	To code and register a service worker, and complete the install and activation process for a new service worker for the E-commerce PWA
Roll No.	38
Name	Snehal Anilkumar Patil
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

Ronak Katariya 23
Prajjwal Pandey 32
Snehal Patil 38

Experiment - 8 : Registering and Activating a Service Worker for E-commerce PWA

Objective:
To code and register a service worker and complete the install and activation process for an eCommerce Progressive Web App (PWA).

Theory:

Service Worker

Service Worker is a script that works on browser background without user interaction independently. Also, It resembles a proxy that works on the user side. With this script, you can track network traffic of the page, manage push notifications and develop “offline first” web applications with Cache API.

Things to note about Service Worker:

- A service worker is a programmable network proxy that lets you control how network requests from your page are handled.
- Service workers only run over HTTPS. Because service workers can intercept network requests and modify responses, "man-in-the-middle" attacks could be very bad.
- The service worker becomes idle when not in use and restarts when it's next needed. You cannot rely on a global state persisting between events. If there is information that you need to persist and reuse across restarts, you can use IndexedDB databases.

What can we do with Service Workers?

- You can dominate **Network Traffic**
  You can manage all network traffic of the page and do any manipulations. For example, when the page requests a CSS file, you can send plain text as a

response or when the page requests an HTML file, you can send a png file as a response. You can also send a true response too.

- You can **Cache**
  You can cache any request/response pair with Service Worker and Cache API and you can access these offline content anytime.

- You can manage **Push Notifications**
  You can manage push notifications with Service Worker and show any information message to the user.
- You can **Continue**
  Although Internet connection is broken, you can start any process with Background Sync of Service Worker.

What can't we do with Service Workers?

- You can't access the **Window**
  You can't access the window, therefore, You can't manipulate DOM elements. But, you can communicate to the window through post Message and manage processes that you want.

- You can't work it on **80 Port**

Service Worker just can work on HTTPS protocol. But you can work on localhost during development.

A service worker follows three main lifecycle phases:

1. **Installation** - The service worker is downloaded and installed.
2. **Activation** - The service worker takes control of the pages and manages caching.
3. **Fetching and Events Handling** - The service worker intercepts network requests and serves cached responses when necessary.

Requirements:

- A code editor (e.g., VS Code, Sublime Text)
- A basic eCommerce web app
- A browser that supports service workers (e.g., Chrome, Edge, Firefox)

Procedure:

1. Creating the Service Worker File

- In the root directory of the eCommerce PWA, create a file named `service-worker.js`.
- Add the following code to set up basic caching:

```
const CACHE_NAME = 'ecommerce-pwa-v1';
const urlsToCache = [
  '/',
  '/index.html',
  '/styles.css',
  '/script.js',
  '/icons/icon-192x192.png',
  '/icons/icon-512x512.png'
];

// Install event - Caching files
self.addEventListener('install', (event) => {
  event.waitUntil(
    caches.open(CACHE_NAME).then((cache) => {
      return cache.addAll(urlsToCache);
    })
  );
});

// Activate event - Cleanup old caches
self.addEventListener('activate', (event) => {
  event.waitUntil(
    caches.keys().then((cacheNames) => {
      return Promise.all(
        cacheNames.filter((cache) => cache !== CACHE_NAME).map((cache) =>
caches.delete(cache))
```

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

// Fetch event - Serve files from cache
self.addEventListener('fetch', (event) => {
  event.respondWith(
    caches.match(event.request).then((response) => {
      return response || fetch(event.request);
    })
  );
});
```

---

2. Registering the Service Worker

- In the `index.html` or `app.js` file, register the service worker with the following code:

```
if ('serviceWorker' in navigator) {
  window.addEventListener('load', () => {
    navigator.serviceWorker.register('/service-worker.js')
      .then((registration) => {
        console.log('Service Worker registered with scope:', registration.scope);
      })
      .catch((error) => {
        console.log('Service Worker registration failed:', error);
      });
  });
}
```

---

3. Testing the Service Worker

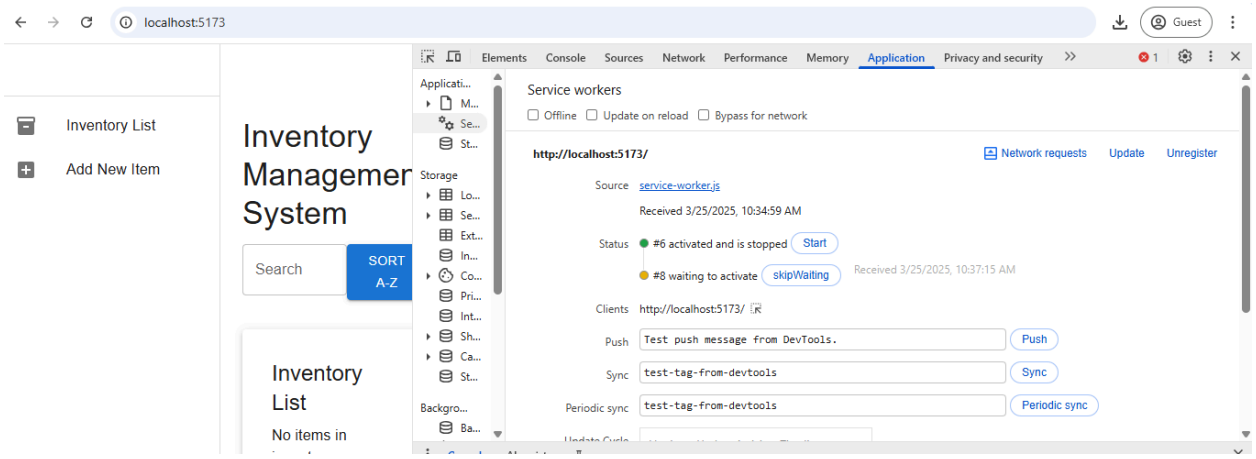
- Host the application on a local server (e.g., using `Live Server` in VS Code or `http-server` in Node.js).
- Open the web application in a browser.
- Open Chrome DevTools (`F12` or `Ctrl+Shift+I`).

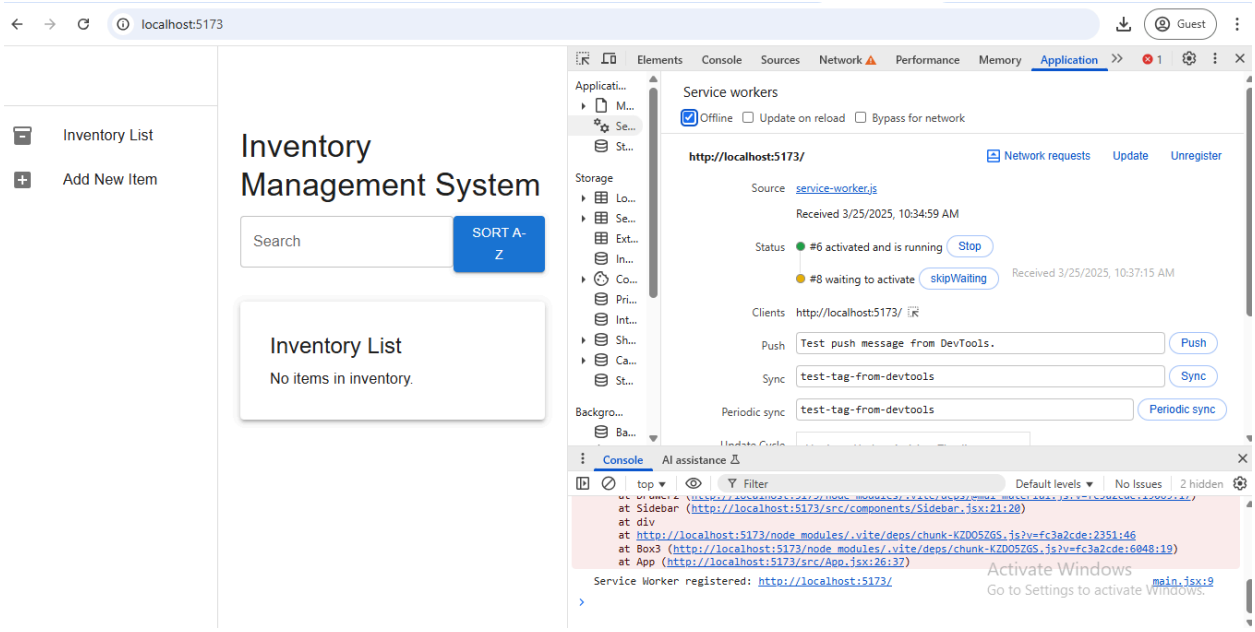
- Go to the "Application" tab → "Service Workers".
- Verify the service worker is installed and activated.

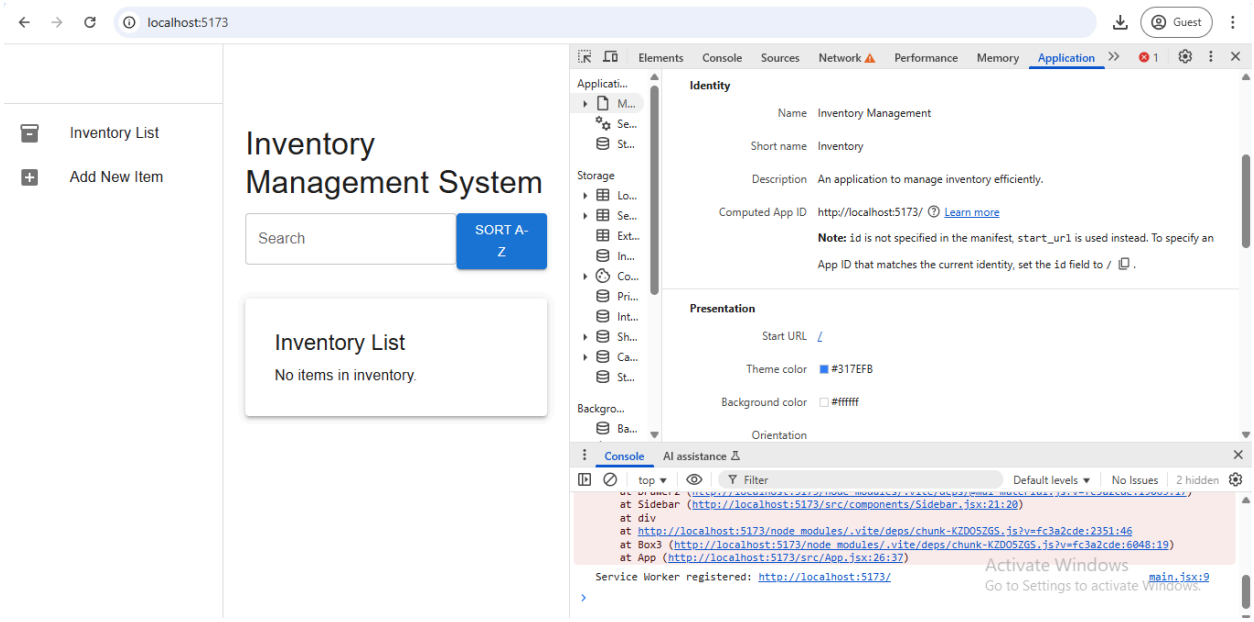
Observations:

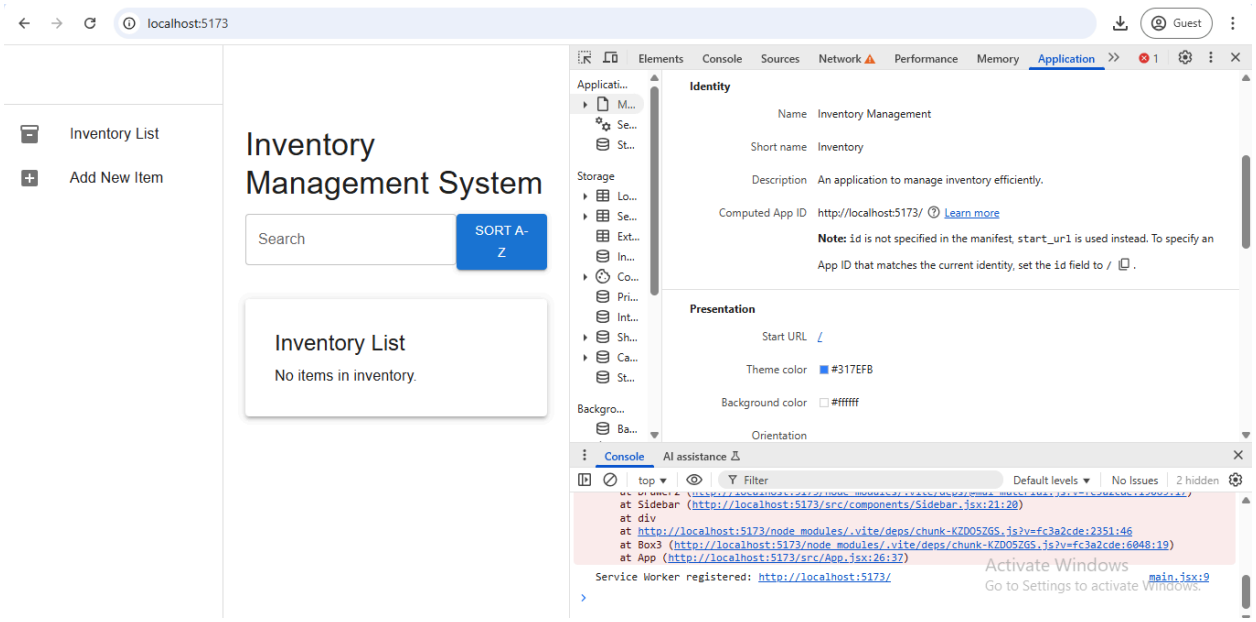
- The service worker successfully caches essential files.
- The app works offline using cached files.
- The activation process ensures that old caches are removed.

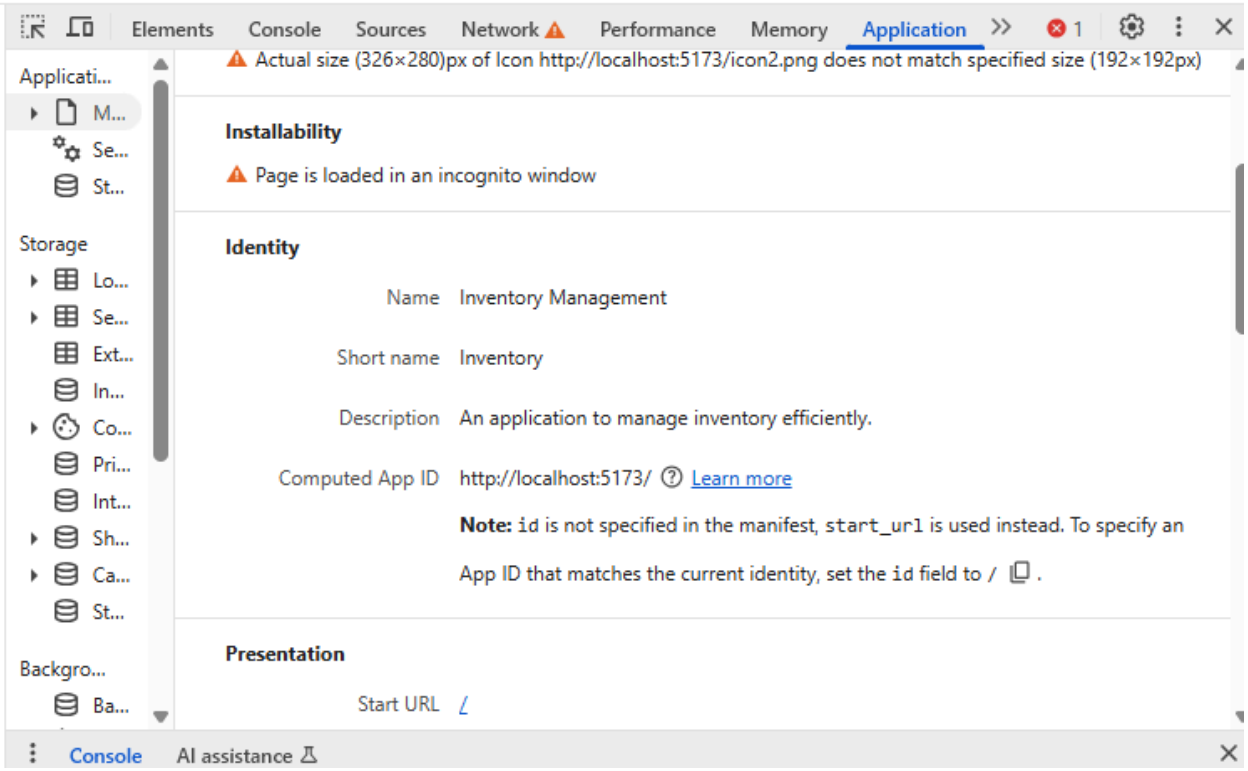
Conclusion:
By coding and registering a service worker, we successfully enabled caching and offline support for the eCommerce PWA, improving performance and reliability.

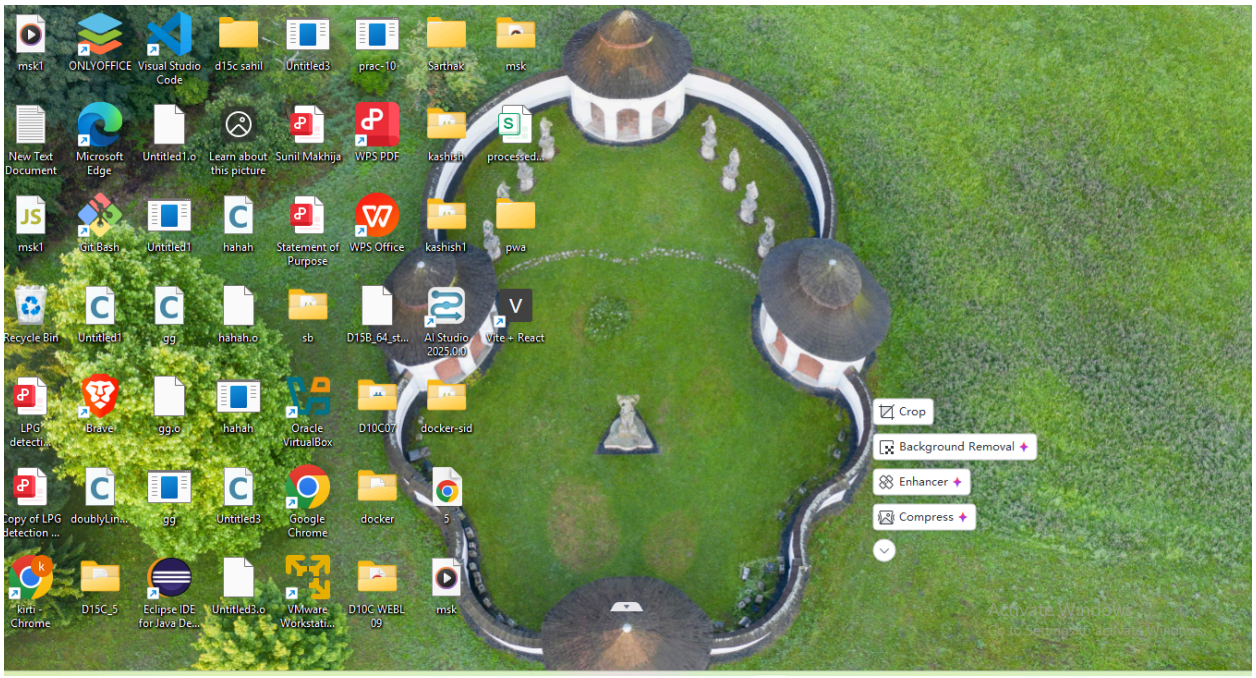



localhost:5173
Guest
Inventory List
Add New Item
Inventory Management System
Search
SORT A-Z
Inventory List
No items in inventory.
Elements
Console
Sources
Network
Performance
Memory
Application
Service workers
Offline
Update on reload
Bypass for network
http://localhost:5173/
Network requests
Update
Unregister
Source
service-worker.js
Received 3/25/2025, 10:34:59 AM
Status
#6 activated and is running
Stop
#8 waiting to activate
skipWaiting
Received 3/25/2025, 10:37:15 AM
Clients
http://localhost:5173/
Push
Test push message from DevTools.
Push
Sync
test-tag-from-devtools
Sync
Periodic sync
test-tag-from-devtools
Periodic sync
Console
AI assistance
Filter
Default levels
No Issues
2 hidden
at Sidebar (http://localhost:5173/src/components/Sidebar.jsx:21:20)
at div
at http://localhost:5173/node_modules/.vite/deps/chunk-KZDO5ZGS.js?v=fc3a2cde:2351:46
at Box3 (http://localhost:5173/node_modules/.vite/deps/chunk-KZDO5ZGS.js?v=fc3a2cde:6048:19)
at App (http://localhost:5173/src/App.jsx:26:37)
Service Worker registered: http://localhost:5173/
main.jsx:9
Activate Windows
Go to Settings to activate Windows.


localhost:5173
Guest
Inventory List
Add New Item
Inventory Management System
Search
SORT A-Z
Inventory List
No items in inventory.
Elements
Console
Sources
Network
Performance
Memory
Application
Identity
Name
Inventory Management
Short name
Inventory
Description
An application to manage inventory efficiently.
Computed App ID
http://localhost:5173/
Learn more
Note: id is not specified in the manifest, start_url is used instead. To specify an App ID that matches the current identity, set the id field to /.
Presentation
Start URL
/
Theme color
#317EFB
Background color
#ffffff
Orientation
Console
AI assistance
Filter
Default levels
No Issues
2 hidden
at Sidebar (http://localhost:5173/src/components/Sidebar.jsx:21:20)
at div
at http://localhost:5173/node_modules/.vite/deps/chunk-KZDO5ZGS.js?v=fc3a2cde:2351:46
at Box3 (http://localhost:5173/node_modules/.vite/deps/chunk-KZDO5ZGS.js?v=fc3a2cde:6048:19)
at App (http://localhost:5173/src/App.jsx:26:37)
Service Worker registered: http://localhost:5173/
main.jsx:9
Activate Windows
Go to Settings to activate Windows.


localhost:5173
Guest
Inventory List
Add New Item
Inventory Management System
Search
SORT A-Z
Inventory List
No items in inventory.
Elements
Console
Sources
Network
Performance
Memory
Application
Identity
Name Inventory Management
Short name Inventory
Description An application to manage inventory efficiently.
Computed App ID http://localhost:5173/ Learn more
Note: id is not specified in the manifest, start_url is used instead. To specify an App ID that matches the current identity, set the id field to /.
Presentation
Start URL /
Theme color #317EFB
Background color #ffffff
Orientation
Console
AI assistance
Service Worker registered: http://localhost:5173/
Activate Windows
Go to Settings to activate Windows.


Elements
Console
Sources
Network
Performance
Memory
Application
Actual size (326×280)px of Icon http://localhost:5173/icon2.png does not match specified size (192×192px)
Installability
Page is loaded in an incognito window
Identity
Name Inventory Management
Short name Inventory
Description An application to manage inventory efficiently.
Computed App ID http://localhost:5173/ Learn more
Note: id is not specified in the manifest, start_url is used instead. To specify an App ID that matches the current identity, set the id field to /.
Presentation
Start URL /
Storage
Backgro...
Console
AI assistance


msk1
ONLYOFFICE
Visual Studio Code
d15c sahil
Untitled3
prac-10
Sarthak
msk
New Text Document
Microsoft Edge
Untitled1.o
Learn about this picture
Sunil Makhija
WPS PDF
kashish
processed...
msk1
Git Bash
Untitled1
hahah
Statement of Purpose
WPS Office
kashish1
pwa
Recycle Bin
Untitled1
gg
hahah.o
sb
D15B_64_st...
AI Studio 2025.0.0
Vite + React
LPG detecti...
Brave
gg.o
hahah
Oracle VirtualBox
D10C07
docker-sid
Copy of LPG detection ...
doublyLin...
gg
Untitled3
Google Chrome
docker
5
kirti - Chrome
D15C_5
Eclipse IDE for Java De...
Untitled3.o
VMware Workstati...
D10C WEBL 09
msk
Crop
Background Removal
Enhancer
Compress
Activate Windows
Go to Settings to activate Windows.

Project Title: **Roll No.**

MAD & PWA Lab
Journal

Experiment No.	09
Experiment Title.	To implement Service worker events like fetch, sync and push for E-commerce PWA
Roll No.	38
Name	Snehal Anilkumar Patil
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

EXPERIMENT 9

Ronak Katariya 23
Prajjwal Pandey 32
Snehal Patil 38

Aim: To implement Service worker events like fetch, sync and push for E-commerce PWA.

Theory:
Service Worker

Service Worker is a script that works on browser background without user interaction independently. Also, It resembles a proxy that works on the user side. With this script, you can track network traffic of the page, manage push notifications and develop “offline first” web applications with Cache API.

Things to note about Service Worker:

- A service worker is a programmable network proxy that lets you control how network requests from your page are handled.
- Service workers only run over HTTPS. Because service workers can intercept network requests and modify responses, "man-in-the-middle" attacks could be very bad.
- The service worker becomes idle when not in use and restarts when it's next needed. You cannot rely on a global state persisting between events. If there is information that you need to persist and reuse across restarts, you can use IndexedDB databases.
- Service workers make extensive use of promises, so if you're new to promises, then you should stop reading this and check out Promises, an introduction.

Fetch Event

You can track and manage page network traffic with this event. You can check existing cache, manage “cache first” and “network first” requests and return a response that you want.

Of course, you can use many different methods but you can find in the following example a “cache first” and “network first” approach. In this example, if the request’s and current location’s origin are the same (Static content is requested.), this is called “cacheFirst” but if you request a targeted external URL, this is called “networkFirst”.

- **CacheFirst** - In this function, if the received request has cached before, the cached response is returned to the page. But if not, a new response requested from the network.
- **NetworkFirst** - In this function, firstly we can try getting an updated response from the network, if this process completed successfully, the new response will be cached and returned.

But if this process fails, we check whether the request has been cached before or not. If a cache exists, it is returned to the page, but if not, this is up to you. You can return dummy content or information messages to the page.

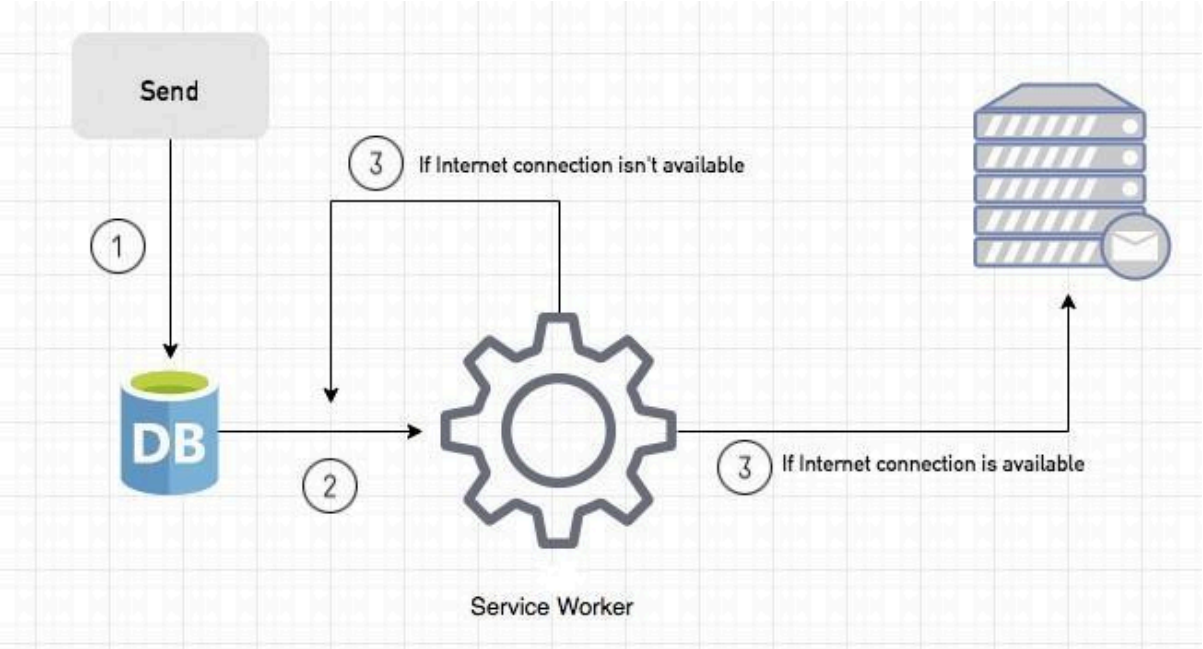
Sync Event

Background Sync is a Web API that is used to delay a process until the Internet connection is stable. We can adapt this definition to the real world; there is an e-mail client application that works on the browser and we want to send an email with this tool. Internet connection is broken while we are writing e-mail content and we didn't realize it. When completing the writing, we click the send button.

Here is a job for the Background Sync.

The following view shows the classical process of sending email to us. If the Internet Connection is broken, we can't send any content to Mail Server.

Here, you can create any scenario for yourself. A sample is in the following for this case.



1. When we click the "send" button, email content will be saved to IndexedDB.
2. Background Sync registration.
3. **If the Internet connection is available**, all email content will be read and sent to Mail Server.
   If the Internet connection is unavailable, the service worker waits until the connection is available even though the window is closed. When it is available, email content will be sent to Mail Server.

   You can see the working process within the following code block.

Event Listener for Background Sync Registration

```
document.querySelector("button").addEventListener("click", async () => {
    var swRegistration = await navigator.serviceWorker.register("sw.js");
    swRegistration.sync.register("helloSync").then(function () {
        console.log("helloSync success [main.js]");
    });
});
```

Event Listener for sw.js

```
self.addEventListener('sync', event => {
    if (event.tag == 'helloSync') {
        console.log("helloSync [sw.js]");
    }
});
```

Push Event

This is the event that handles push notifications that are received from the server. You can apply any method with received data.

We can check in the following example.
“Notification.requestPermission();” is the necessary line to show notification to the user. If you don’t want to show any notification, you don’t need this line.

In the following code block is in sw.js file. You can handle push notifications with this event. In this example, I kept it simple. We send an object that has “method” and “message” properties. If the method value is “pushMessage”, we open the information notification with the “message” property.

```
self.addEventListener('push', event => {
    if (event && event.data) {
        var data = event.data.json();
        if (data.method === "pushMessage") {
            event.waitUntil(self.registration.showNotification("Test App", {
                body: data.message
            }));
        }
    }
});
```

CODE:

Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Stopwatch</title>
    <link rel="stylesheet" href="style.css">
    <link rel="manifest" href="/manifest.json">
  </head>
  <body>
    <div class="container">
      <h1>Stopwatch</h1>
      <p class="time">
        <span id="minutes">00</span>:<span id="seconds">00</span>:<span id="tens">00</span>
      </p>
      <ul id="lapList"></ul>
      <button id="start">Start</button>
      <button id="stop">Stop</button>
      <button id="reset">Reset</button>
      <button id="lap">Lap</button>
    </div>
    <script src="stopwatch.js"></script>
    <script>
      if ('serviceWorker' in navigator) {
        window.addEventListener('load', () => {
          navigator.serviceWorker.register('/service-worker.js')
            .then(registration => {
              console.log('ServiceWorker registered with scope: ', registration.scope);
            })
            .catch(error => {
              console.log('ServiceWorker registration failed: ', error);
            });
        });
      }

      // Request Notification Permission and Show Popup
      if ('Notification' in window) {
        Notification.requestPermission().then(permission => {
          if (permission === 'granted') {
            console.log('Notification permission granted');
            showNotification();
```

```
        }
      });
    }

    function showNotification() {
      if (Notification.permission === 'granted') {
        const options = {
          body: 'Welcome Snehal, Ronak, Prajjwal !',
          icon: '/images/icon.png',
          badge: '/images/icon.png'
        };

        // Display the notification
        new Notification('Welcome Snehal, Ronak, Prajjwal ', options);
      }
    }
  </script>
 </body>
</html>
```

Manifest.json

```
{
  "name": "PWA",
  "short_name": "RPS",
  "description": "A simple and elegant stopwatch application.",
  "start_url": "/index.html",
  "display": "standalone",
  "background_color": "#ffffff",
  "theme_color": "#000000",
  "icons": [
    {
      "src": "/images/icon2.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "/images/icon.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
 }
```

serviceworker.js

```
const CACHE_NAME = 'prodigy-stopwatch-cache-v1';
const urlsToCache = [
  '/',
  '/index.html',
  '/style.css',
  '/stopwatch.js',
  '/images/icon2.png',
  '/images/icon.png'
];

// Install Service Worker and Cache Files
self.addEventListener('install', event => {
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then(cache => {
        console.log('Opened cache');
        return cache.addAll(urlsToCache);
      })
  );
});

// Cache and Return Requests
self.addEventListener('fetch', event => {
  event.respondWith(
    caches.match(event.request).then(response => {
      return response || fetch(event.request)
        .then(networkResponse => {
          if (!networkResponse || networkResponse.status !== 200) {
            return networkResponse;
          }
          return caches.open(CACHE_NAME).then(cache => {
            cache.put(event.request, networkResponse.clone());
            return networkResponse;
          });
        }).catch(() => {
          if (event.request.mode === 'navigate') {
            return caches.match('/index.html');
          }
        });
    })
  );
});
```

```
// Activate & Remove Old Caches
self.addEventListener('activate', event => {
  const cacheWhitelist = [CACHE_NAME];
  event.waitUntil(
    caches.keys().then(cacheNames => {
      return Promise.all(
        cacheNames.map(cacheName => {
          if (!cacheWhitelist.includes(cacheName)) {
            return caches.delete(cacheName);
          }
        })
      );
    })
  );
});
```

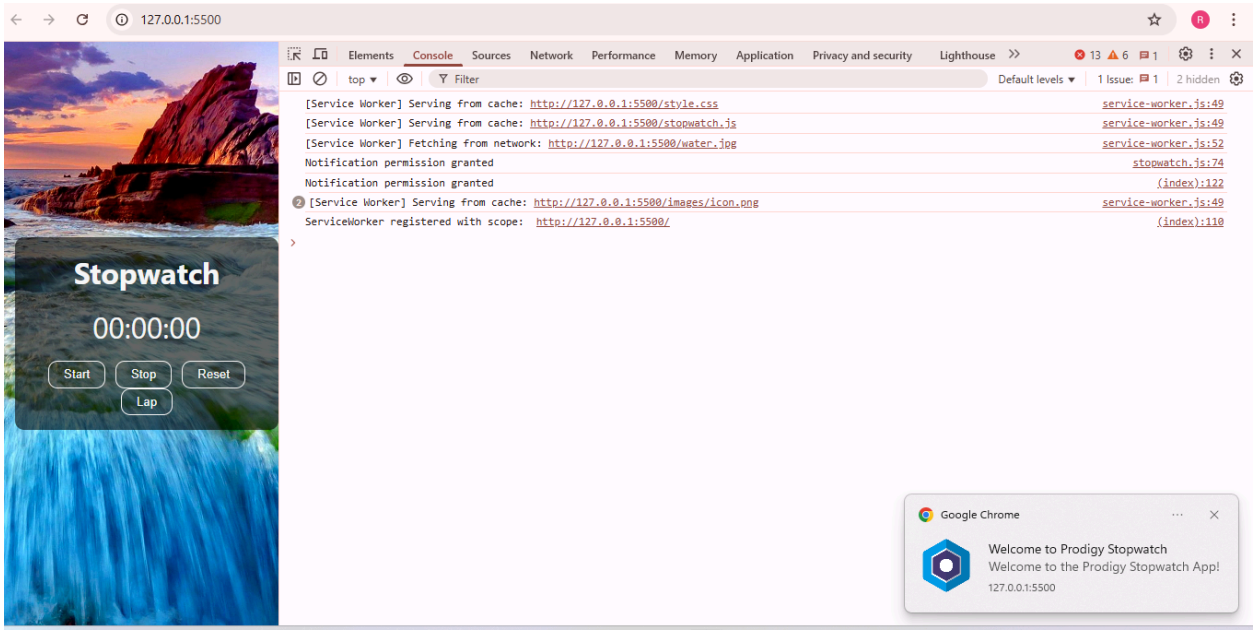
stopwatch.js

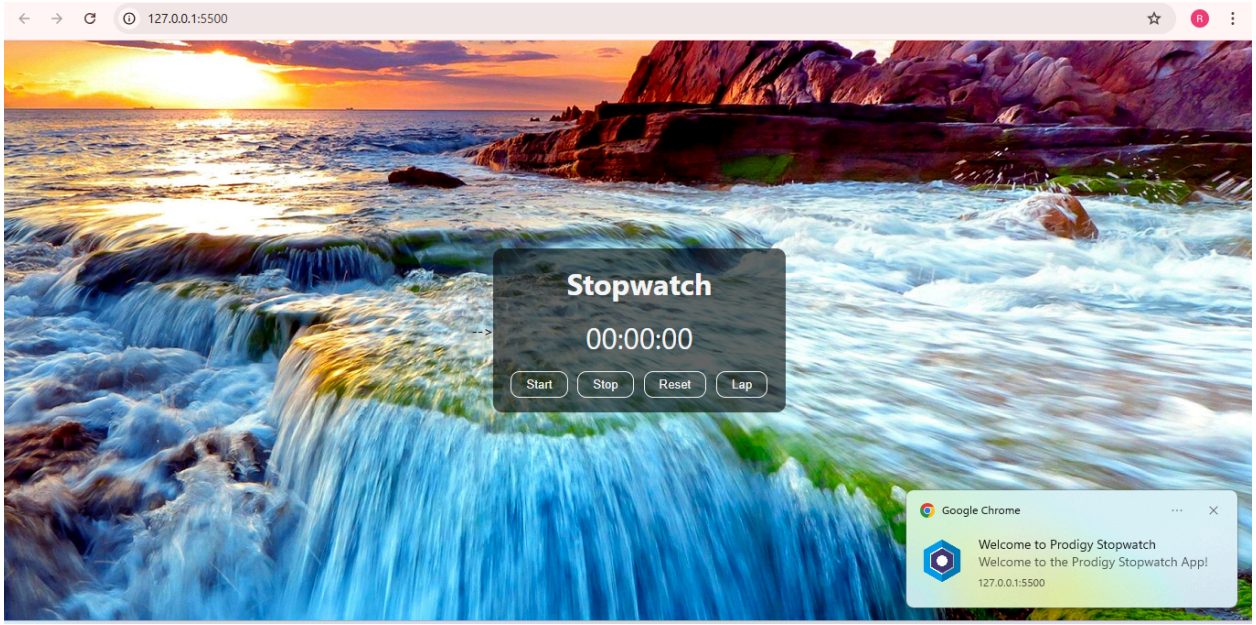
```
window.onload = function () {
    let minutes = 0;
    let seconds = 0;
    let tens = 0;
    let appendMinutes = document.querySelector('#minutes');
    let appendTens = document.querySelector('#tens');
    let appendSeconds = document.querySelector('#seconds');
    let startBtn = document.querySelector('#start');
    let stopBtn = document.querySelector('#stop');
    let resetBtn = document.querySelector('#reset');
    let lapBtn = document.querySelector('#lap');
    let lapList = document.querySelector('#lapList');
    let Interval;

    const startTimer = () => {
      tens++;
      if (tens <= 9) {
        appendTens.innerHTML = '0' + tens;
      }
      if (tens > 9) {
        appendTens.innerHTML = tens;
      }

      if (tens > 99) {
        seconds++;
        appendSeconds.innerHTML = '0' + seconds;
        tens = 0;
```

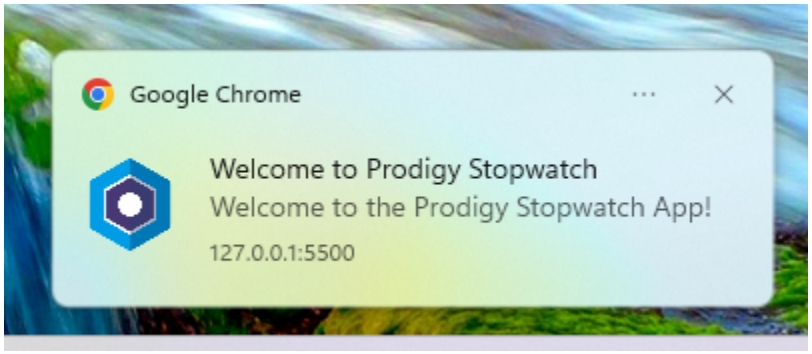
```
      appendTens.innerHTML = '0' + 0;
    }

    if (seconds > 9) {
      appendSeconds.innerHTML = seconds;
    }

    if (seconds > 59) {
      minutes++;
      appendMinutes.innerHTML = '0' + minutes;
      seconds = 0;
      appendSeconds.innerHTML = '0' + 0;
    }
  };

  startBtn.onclick = () => {
    clearInterval(Interval);
    Interval = setInterval(startTimer, 10);
  };

  stopBtn.onclick = () => {
    clearInterval(Interval);
  };

  resetBtn.onclick = () => {
    clearInterval(Interval);
    tens = '00';
    seconds = '00';
    minutes = '00';
    appendTens.innerHTML = tens;
    appendSeconds.innerHTML = seconds;
    appendMinutes.innerHTML = minutes;
    lapList.innerHTML = '';
  };

  lapBtn.onclick = () => {
    var li = document.createElement('li');
    li.innerHTML = appendMinutes.innerHTML + ':' + appendSeconds.innerHTML + '.' +
appendTens.innerHTML;
    lapList.appendChild(li);
  };
};

// Request Notification Permission when the app is loaded
```

```
if ('Notification' in window && 'serviceWorker' in navigator) {
  Notification.requestPermission().then(permission => {
    if (permission === 'granted') {
      console.log('Notification permission granted');
    }
  });
}
```

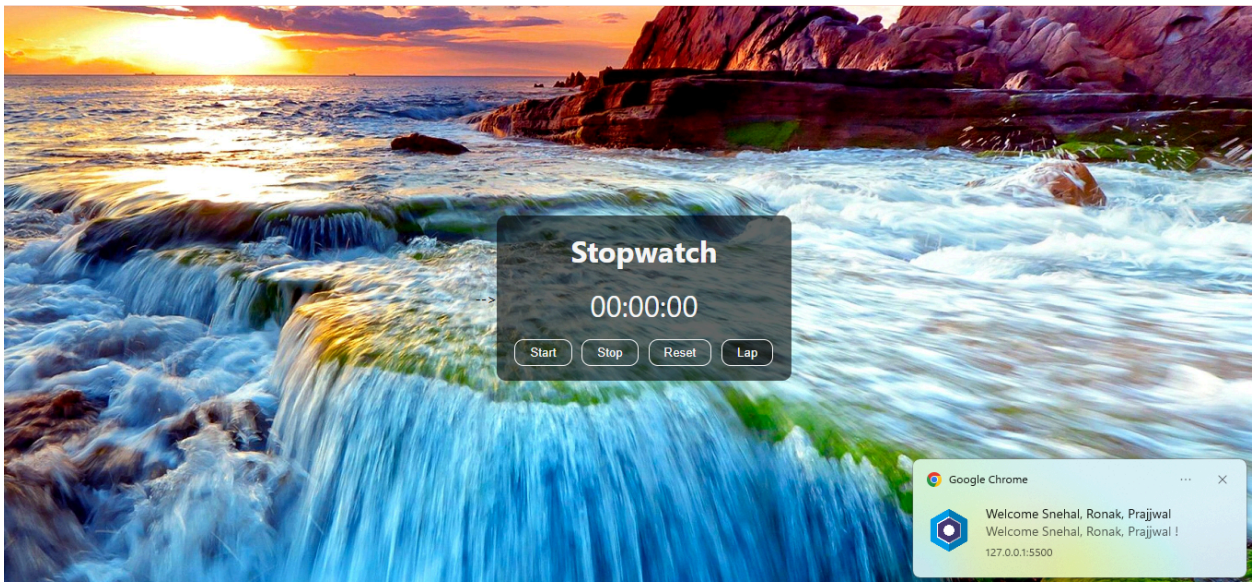
Screenshots:

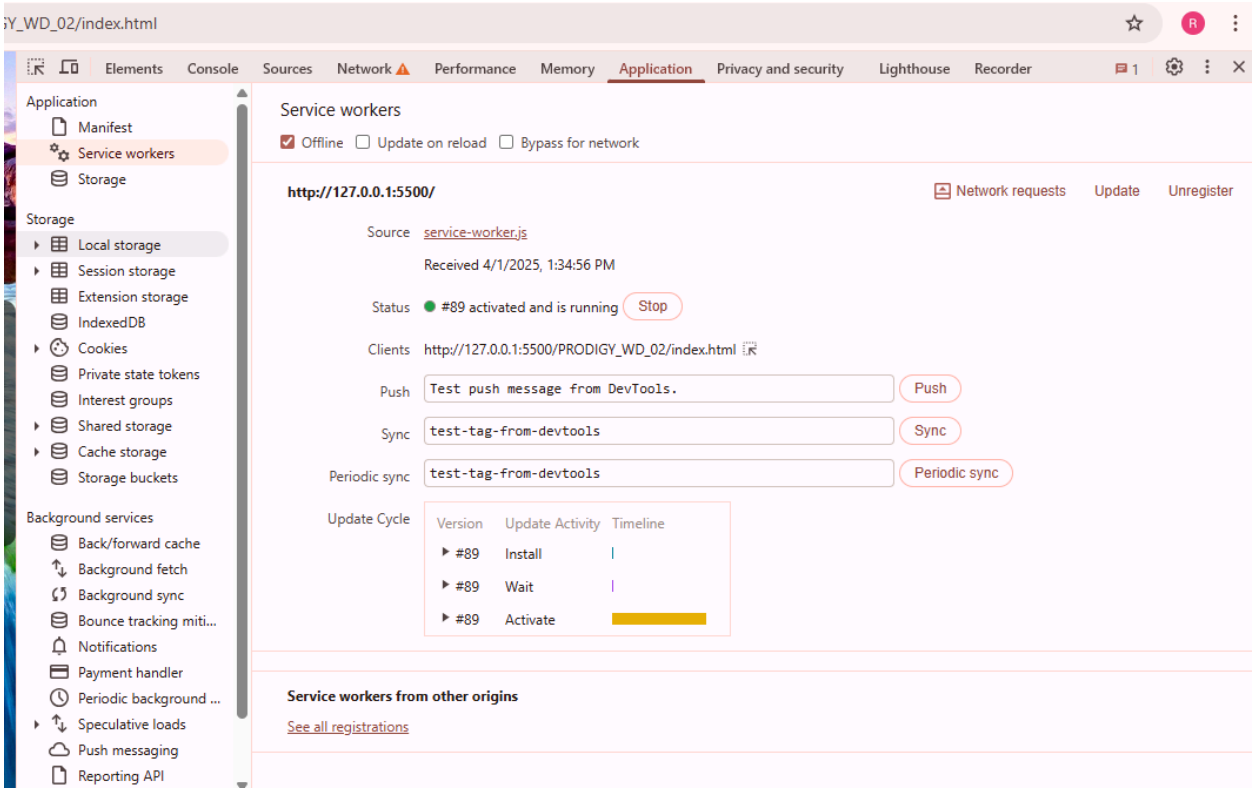


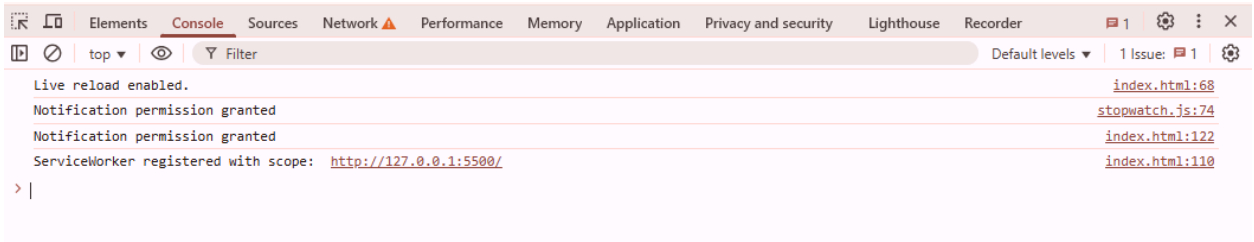





```
Notification permission granted                                  stopwatch.js:74
Notification permission granted                                  (index):122
ServiceWorker registered with scope:  http://127.0.0.1:5500/     (index):110
```




Y_WD_02/index.html
Elements
Console
Sources
Network
Performance
Memory
Application
Privacy and security
Lighthouse
Recorder
Application
Manifest
Service workers
Storage
Storage
Local storage
Session storage
Extension storage
IndexedDB
Cookies
Private state tokens
Interest groups
Shared storage
Cache storage
Storage buckets
Background services
Back/forward cache
Background fetch
Background sync
Bounce tracking miti...
Notifications
Payment handler
Periodic background ...
Speculative loads
Push messaging
Reporting API
Service workers
Offline
Update on reload
Bypass for network
http://127.0.0.1:5500/
Network requests
Update
Unregister
Source
service-worker.js
Received 4/1/2025, 1:34:56 PM
Status
#89 activated and is running
Stop
Clients
http://127.0.0.1:5500/PRODIGY_WD_02/index.html
Push
Test push message from DevTools.
Push
Sync
test-tag-from-devtools
Sync
Periodic sync
test-tag-from-devtools
Periodic sync
Update Cycle
Version
Update Activity
Timeline
#89
Install
#89
Wait
#89
Activate
Service workers from other origins
See all registrations


Elements
Console
Sources
Network
Performance
Memory
Application
Privacy and security
Lighthouse
Recorder
top
Filter
Default levels
1 Issue:
Live reload enabled.
index.html:68
Notification permission granted
stopwatch.js:74
Notification permission granted
index.html:122
ServiceWorker registered with scope:
http://127.0.0.1:5500/
index.html:110

Project Title: **Roll No.**

MAD & PWA Lab
Journal

Experiment No.	10
Experiment Title.	To study and implement deployment of Ecommerce PWA to GitHub Pages.
Roll No.	38
Name	Snehal Anilkumar Patil
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

Experiment No.
10 MAD & PWA
Lab

Ronak Katariya 23
Prajjwal Pandey 32
Snehal Patil 38

Aim:

To study and implement deployment of Ecommerce PWA to GitHub Pages.

Theory:

GitHub Pages

Public web pages are freely hosted and easily published. Public webpages hosted directly from your GitHub repository. Just edit, push, and your changes are live.

GitHub Pages provides the following key features:

1. Blogging with Jekyll
2. Custom URL
3. Automatic Page Generator

Reasons for favoring this over Firebase:

1. Free to use
2. Right out of github
3. Quick to set up

GitHub Pages is used by Lyft, CircleCI, and HubSpot.
GitHub Pages is listed in 775 company stacks and 4401 developer stacks.

Pros

1. Very familiar interface if you are already using GitHub for your projects.
2. Easy to set up. Just push your static website to the gh-pages branch and your website is ready.
3. Supports Jekyll out of the box.
4. Supports custom domains. Just add a file called CNAME to the root of your site, add an A record in the site's DNS configuration, and you are done.

Cons

1. The code of your website will be public, unless you pay for a private repository.
2. Currently, there is no support for HTTPS for custom domains. It's probably coming soon though.
3. Although Jekyll is supported, plug-in support is rather spotty.

Firebase

The Realtime App Platform. Firebase is a cloud service designed to power real-time, collaborative applications. Simply add the Firebase library to your application to gain access to a shared data structure; any changes you make to that data are automatically synchronized with the Firebase cloud and with other clients within milliseconds.

Some of the features offered by Firebase are:

1. Add the Firebase library to your app and get access to a shared data structure. Any changes made to that data are automatically synchronized with the Firebase cloud and with other clients within milliseconds.
2. Firebase apps can be written entirely with client-side code, update in real-time out-of-the-box, interoperate well with existing services, scale automatically, and provide strong data security.
3. Data Accessibility- Data is stored as JSON in Firebase. Every piece of data has its own URL which can be used in Firebase's client libraries and as a REST endpoint. These URLs can also be entered into a browser to view the data and watch it update in real-time.

Reasons for favoring over GitHub Pages:

1. Realtime backend made easy
2. Fast and responsive

Instacart, 9GAG, and Twitch are some of the popular companies that use Firebase
Firebase has a broader approval, being mentioned in 1215 company stacks & 4651 developers stacks

Pros

1. Hosted by Google. Enough said.
2. Authentication, Cloud Messaging, and a whole lot of other handy services will be available to you.
3. A real-time database will be available to you, which can store 1 GB of data.
4. You'll also have access to a blob store, which can store another 1 GB of data.
5. Support for HTTPS. A free certificate will be provisioned for your custom domain within 24 hours.

Cons

1. Only 10 GB of data transfer is allowed per month. But this is not really a big problem, if you use a CDN or AMP.
2. Command-line interface only.
3. No in-built support for any static site generator.

Link to our GitHub repository: https://github.com/ronak03rsk/PRODIGY_WD_02

Project Title: **Roll No.**

MAD & PWA Lab
Journal

Experiment No.	11
Experiment Title.	To use google Lighthouse PWA Analysis Tool to test the PWA functioning.
Roll No.	38
Name	Snehal Anilkumar Patil
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO6: Develop and Analyze PWA Features and deploy it over app hosting solution
Grade:	

Experiment: Using Google Lighthouse for PWA Analysis

Objective:
To use Google Lighthouse PWA Analysis Tool to test the Progressive Web App (PWA) functionality.

Theory:
Google Lighthouse is an open-source automated tool used to audit and analyze web applications for performance, accessibility, best practices, SEO, and Progressive Web App (PWA) capabilities. Lighthouse evaluates whether a web app meets PWA standards, such as:

- Service worker registration for offline support
- Web App Manifest inclusion
- HTTPS security compliance
- Fast and responsive performance
- Installability and app-like user experience

Lighthouse assigns a score based on these criteria, helping developers optimize their PWA.

Google Lighthouse :

> Google Lighthouse is a tool that lets you audit your web application based on a number of parameters including (but not limited to) performance, based on a number of metrics, mobile compatibility, Progressive Web App (PWA) implementations, etc. All you have to do is run it on a page or pass it a URL, sit back for a couple of minutes and get a very elaborate report, not much short of one that a professional auditor would have compiled in about a week.
>
> The best part is that you have to set up almost nothing to get started. Let's begin by looking at some of the top features and audit criteria used by Lighthouse.

Key Features and Audit Metrics

Google Lighthouse has the option of running the Audit for Desktop as well as mobile version of your page(s). The top metrics that will be measured in the Audit are:

1. **Performance:** This score is an aggregation of how the page fared in aspects such as (but not limited to) loading speed, time taken for loading for basic frame(s), displaying meaningful content to the user, etc. To a layman, this score is indicative of how decently the site performs, with a score of 100 meaning that you figure in the 98th percentile, 50 meaning that you figure in the 75th percentile and so on.
2. **PWA Score (Mobile):** Thanks to the rise of Service Workers, app manifests, etc., a lot of modern web applications are moving towards the PWA paradigm, where the objective is to make the application behave as close as possible to native mobile applications. Scoring points are based on the Baseline PWA checklist laid down by Google which includes Service Worker implementation(s), viewport handling, offline functionality, performance in script-disabled environments, etc.
3. **Accessibility:** As you might have guessed, this metric is a measure of how accessible your website is, across a plethora of accessibility features that can be implemented in your page (such as the ‘aria-’ attributes like aria-required, audio captions, button names, etc.). Unlike the other metrics though, Accessibility metrics score on a pass/fail basis i.e. if all possible elements of the page are not screen-reader friendly (HTML5 introduced features that would make pages easy to interpret for screen readers used by visually challenged people like tag names, tags such as <section>, <article>, etc.), you get a 0 on that score. The aggregate of these scores is your Accessibility metric score.
4. **Best Practices:** As any developer would know, there are a number

of practices that have been deemed ‘best’ based on empirical data. This metric is an aggregation of many such points, including but not limited to:Use of HTTPS
Avoiding the use of deprecated code elements like tags, directives, libraries, etc. Password input with paste-into disabled
Geo-Location and cookie usage alerts on load, etc.

Requirements:

- Google Chrome browser
- A PWA-enabled web application
- Chrome DevTools or Lighthouse CLI

Procedure:

1. Open Lighthouse in Chrome DevTools

- Launch the PWA in Google Chrome.
- Open DevTools using `F12` or `Ctrl+Shift+I`.
- Navigate to the **Lighthouse** tab.

2. Configure Lighthouse Audit

- Select the **Progressive Web App** category.
- Ensure other relevant categories like Performance and Best Practices are checked.
- Choose the mode:

    - **Mobile** (default) for testing mobile performance.
    - **Desktop** for desktop evaluation.
- Click **Generate Report**.

3. Analyze the Results

- Lighthouse generates a PWA score based on:
    - **Fast and reliable:** Checks service worker caching.
    - **Installable:** Verifies manifest and service worker.
    - **PWA Optimizations:** Ensures proper web app experience.
- Click on individual sections to view recommendations for improvements.

4. Running Lighthouse via CLI (Optional)

Install Lighthouse CLI using Node.js:

```
npm install -g lighthouse
```

-

Run an audit on a PWA URL:

```
lighthouse https://your-pwa-url.com --view
```

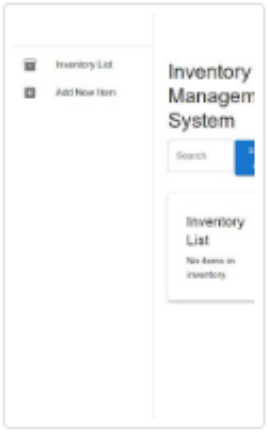
-

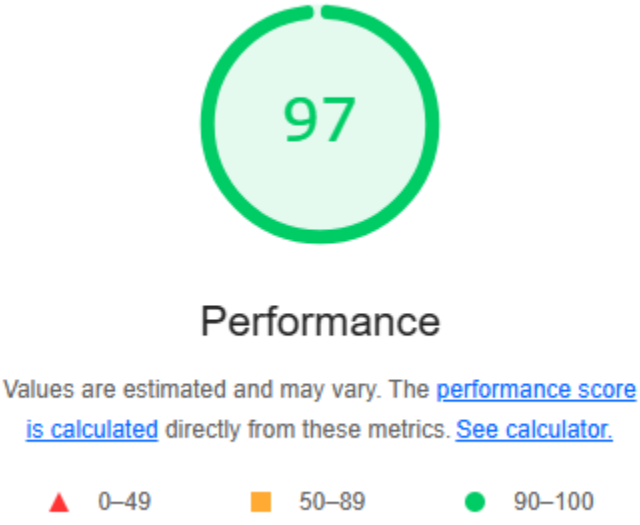
Observations:

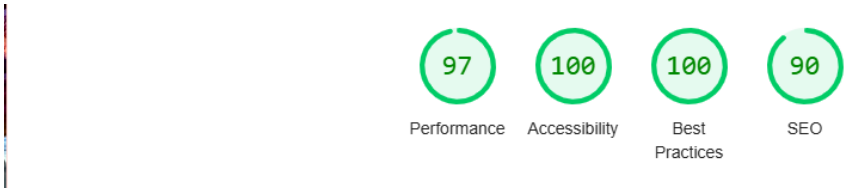
- The PWA is analyzed based on Lighthouse criteria.
- Identified areas needing improvement (e.g., caching strategies, manifest completeness, accessibility fixes).
- Scores indicate overall PWA compliance and optimization level.

Conclusion:
By using Google Lighthouse, we effectively evaluated the PWA's functionality, installability, and performance, enabling improvements for a better user experience.

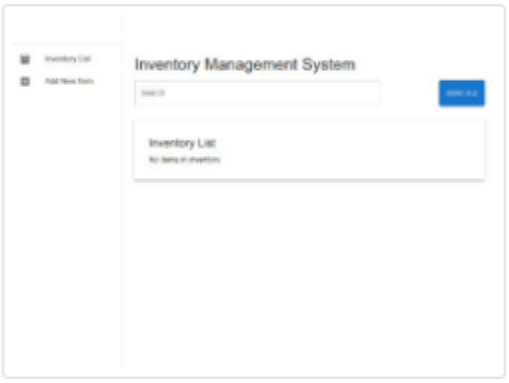
Mobile View:

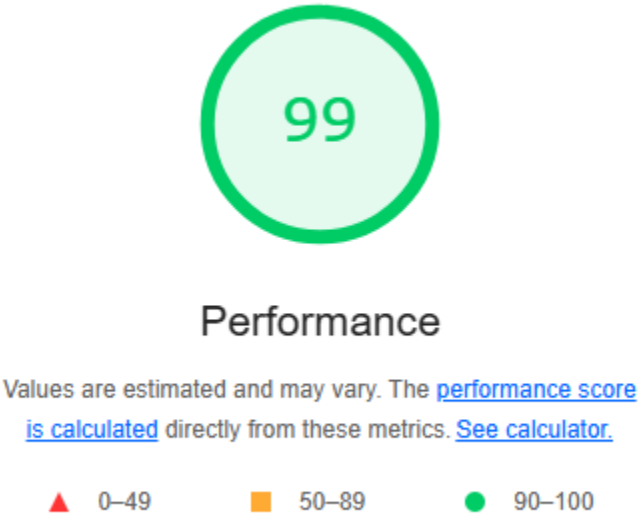


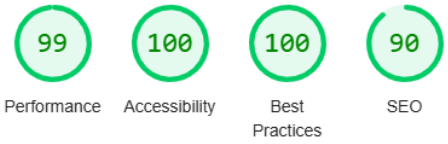




Desktop View:






99
Performance
100
Accessibility
100
Best Practices
90
SEO