# 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 **Bhagyesh Bharat Patil** of **D15A** 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: Rapido** **Roll No. 35**

**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: Rapido** **Roll No. 35**

**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: Rapido** **Roll No.35**

# Index

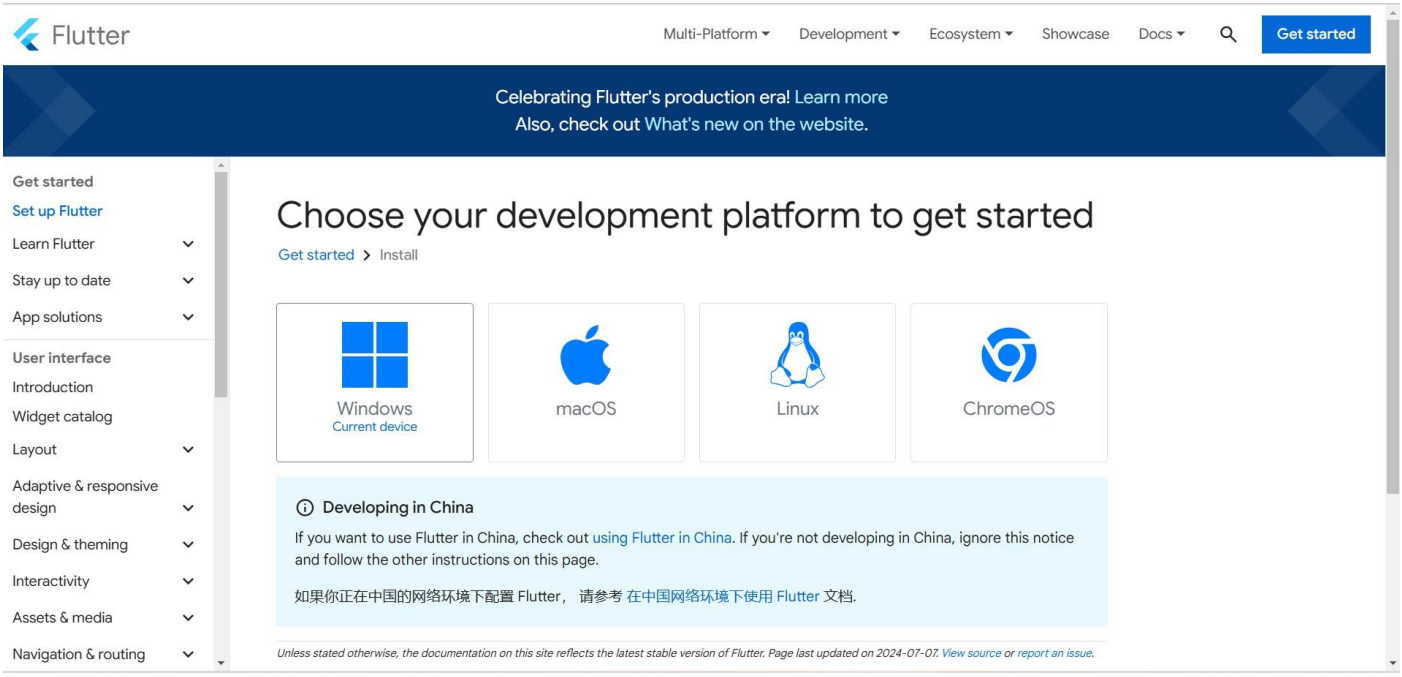
| Sr. No | Experiment Title | LO | DOP | DOS | Grade |
|---|---|---|---|---|---|
| **1.** | To install and configure the Flutter Environment | LO1 | | | |
| **2.** | To design Flutter UI by including common widgets. | LO2 | | | |
| **3.** | To include icons, images, fonts in Flutter app | LO2 | | | |
| **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 | | | |

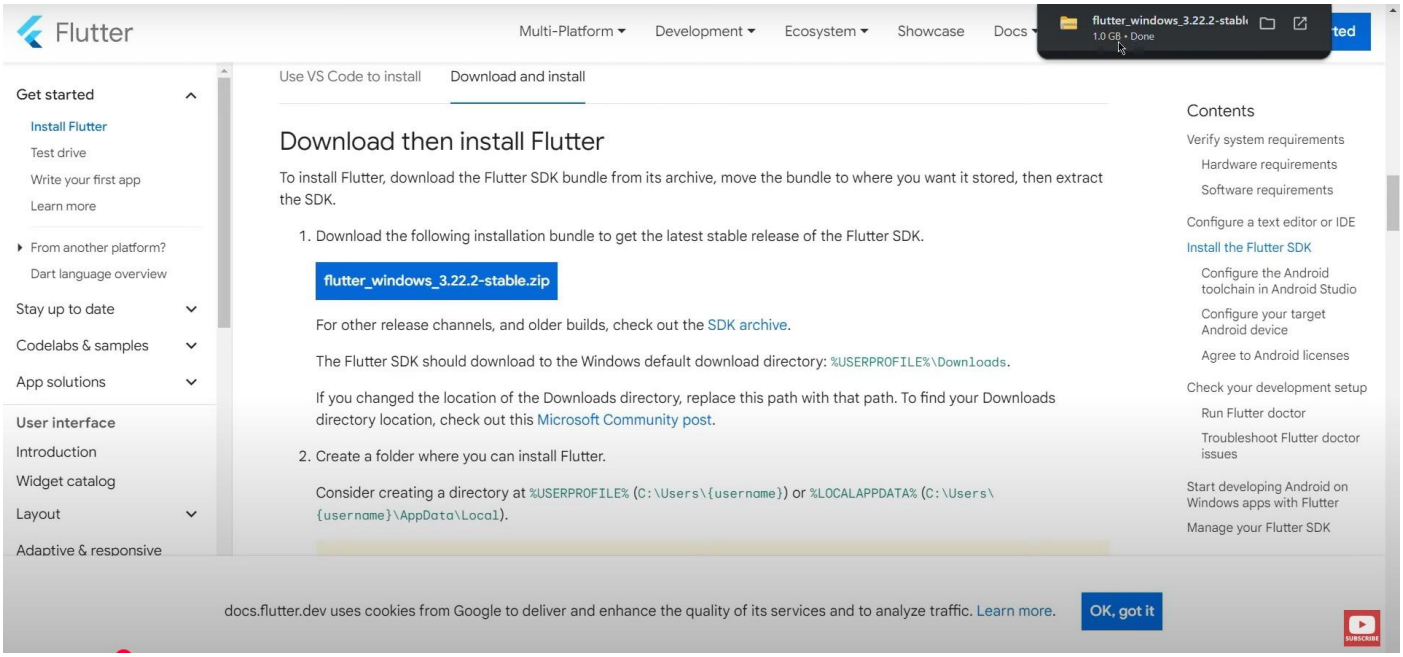
# MAD & PWA Lab

# Journal

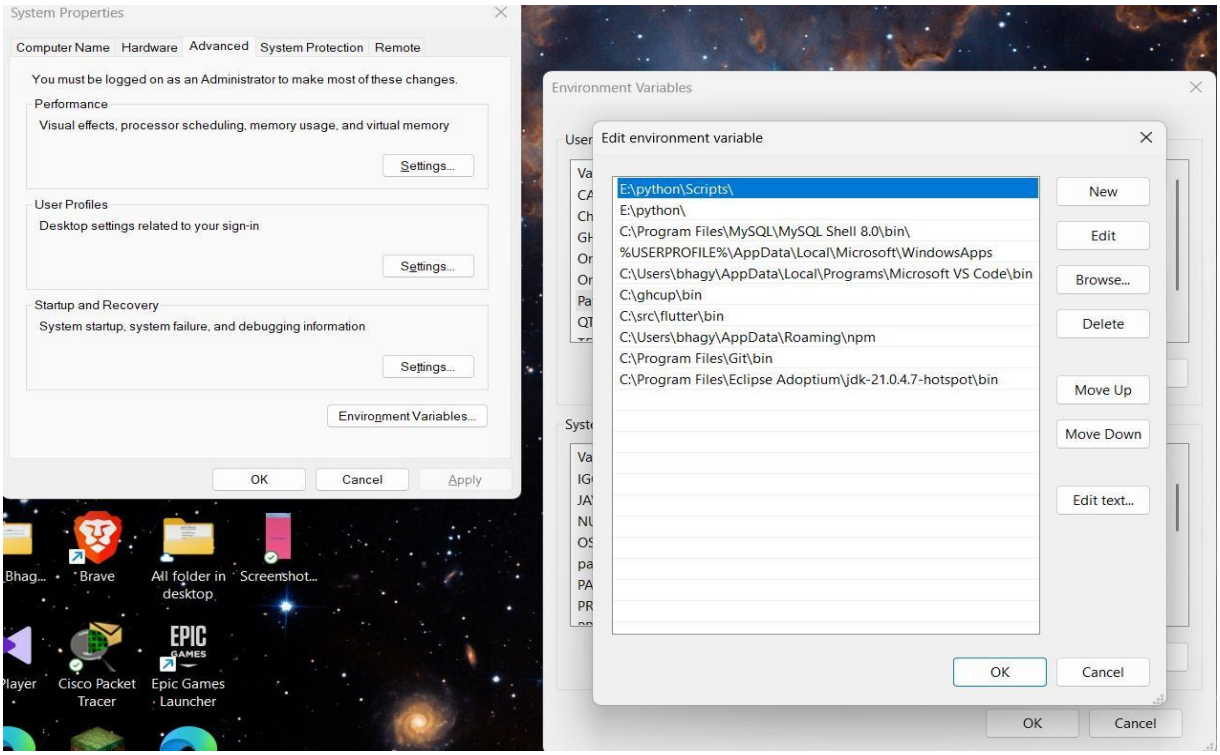
| | |
|---|---|
| Experiment No. | 01 |
| Experiment Title. | To install and configure the Flutter Environment |
| Roll No. | 35 |
| Name | Bhagyesh Patil |
| Class | D15A/D15B |
| Subject | MAD & PWA Lab |
| Lab Outcome | LO1: Understand cross platform mobile application development using Flutter framework |
| Grade: | |

# Experiment no 1



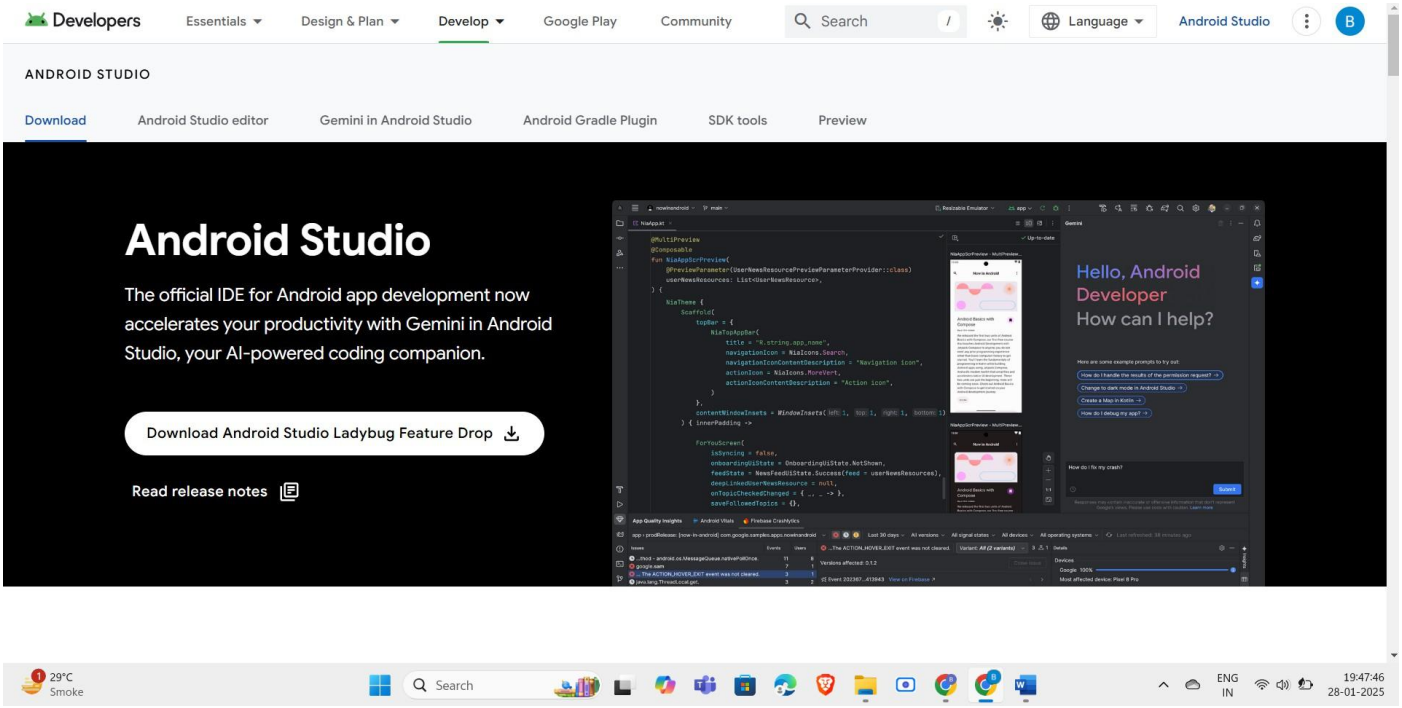
- Set Environment variables and paste the flutter bin path in path option




- In Cmd type flutter doctor

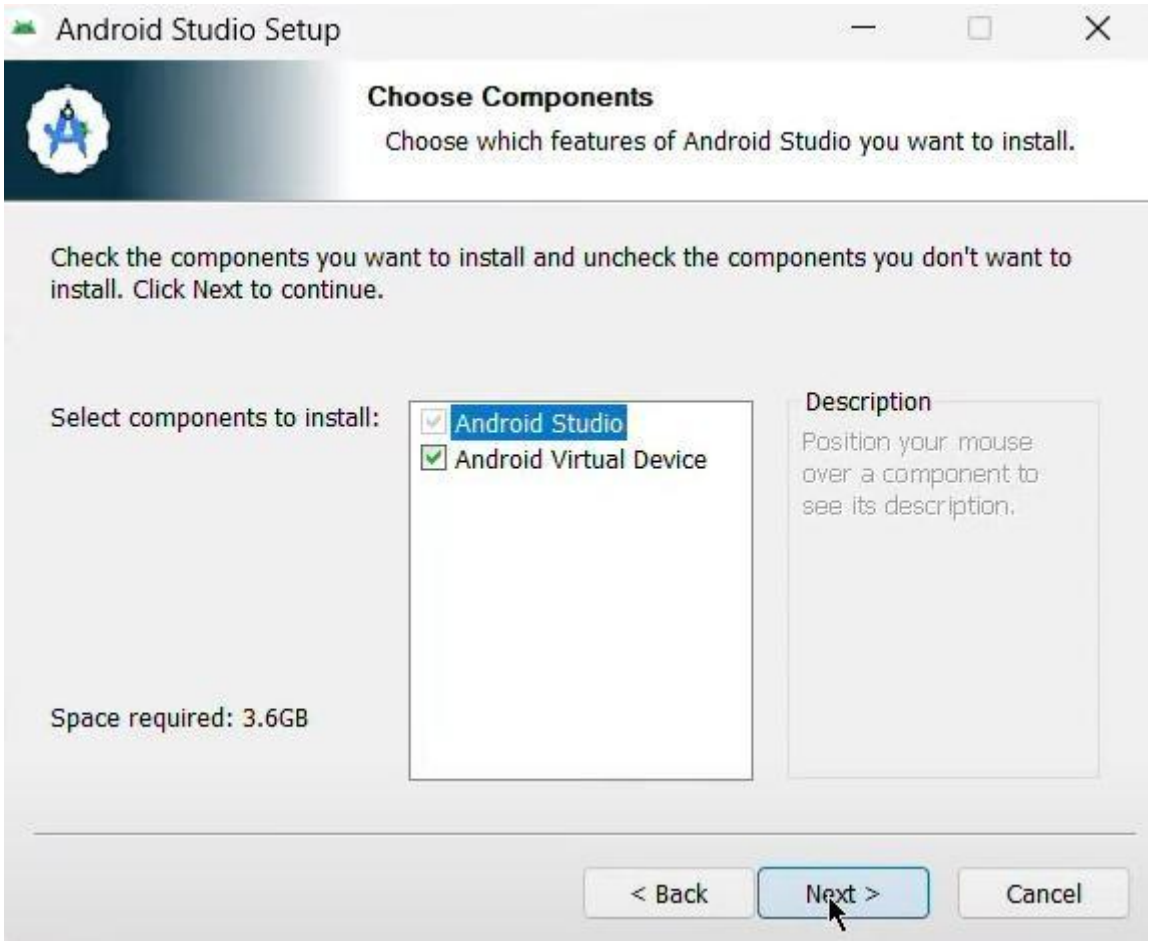
```
                Welcome to Flutter! - https://flutter.dev

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

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

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

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

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

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

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

- Download the android studio go the official website

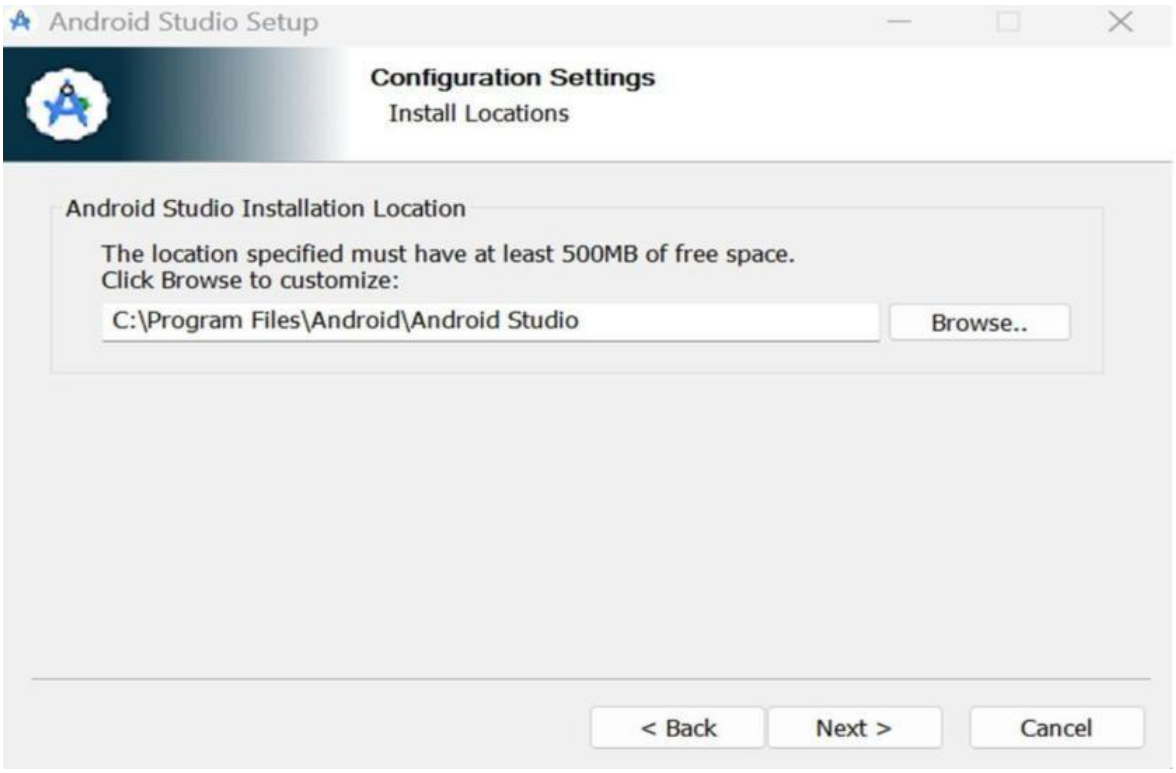


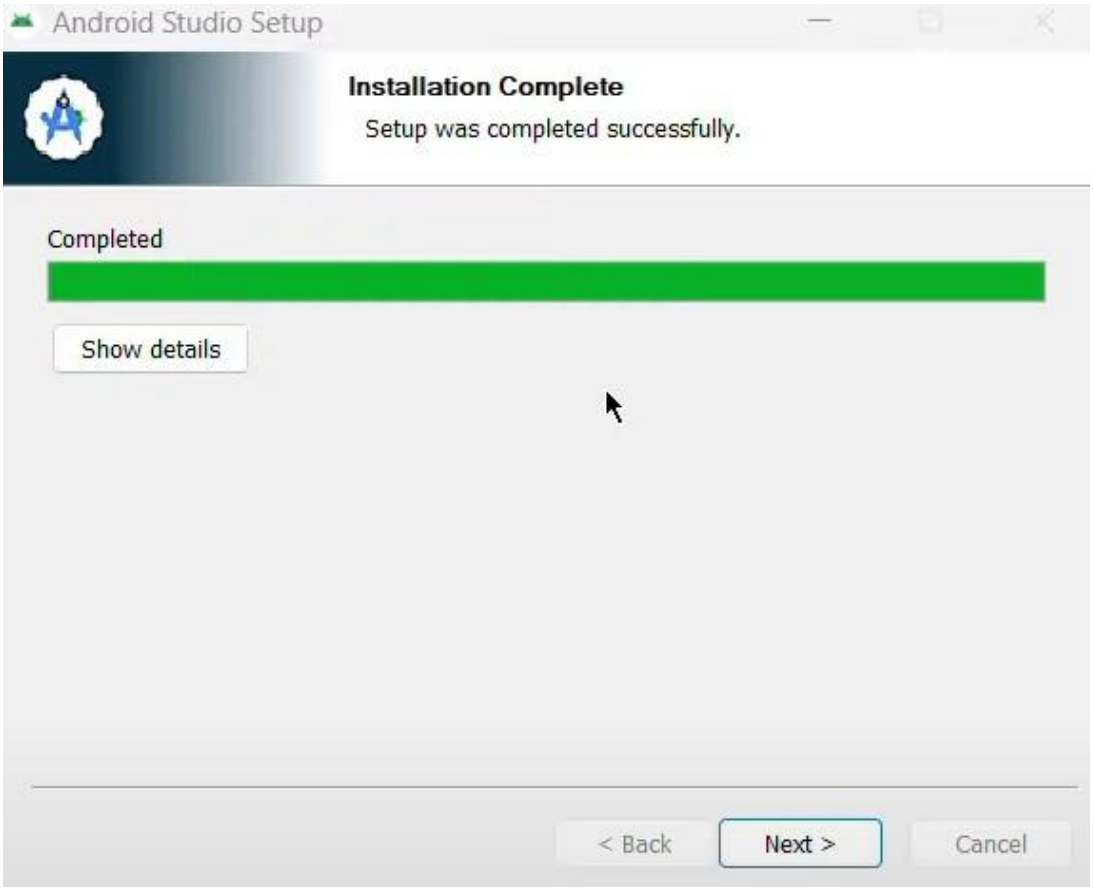




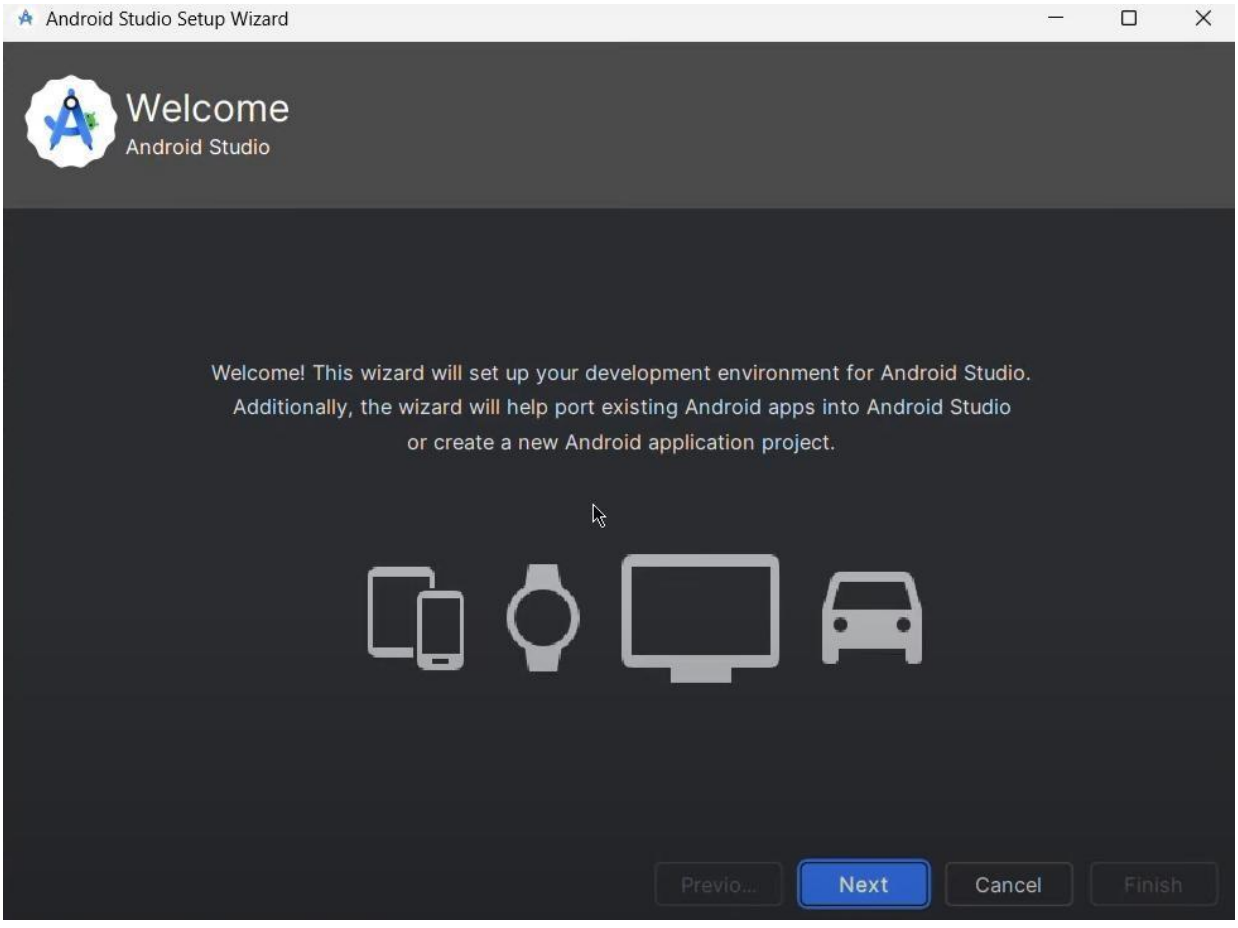



- Downloading Android SDK

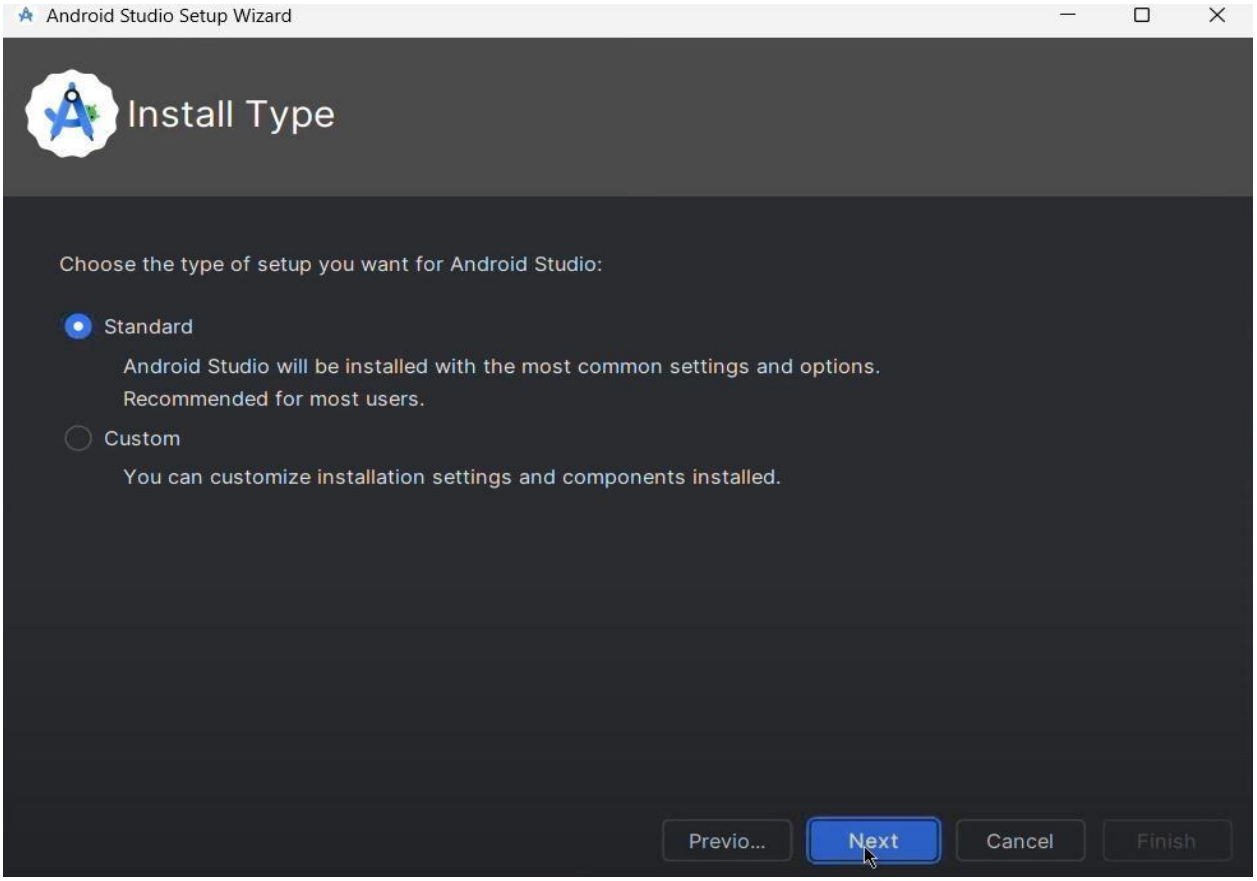




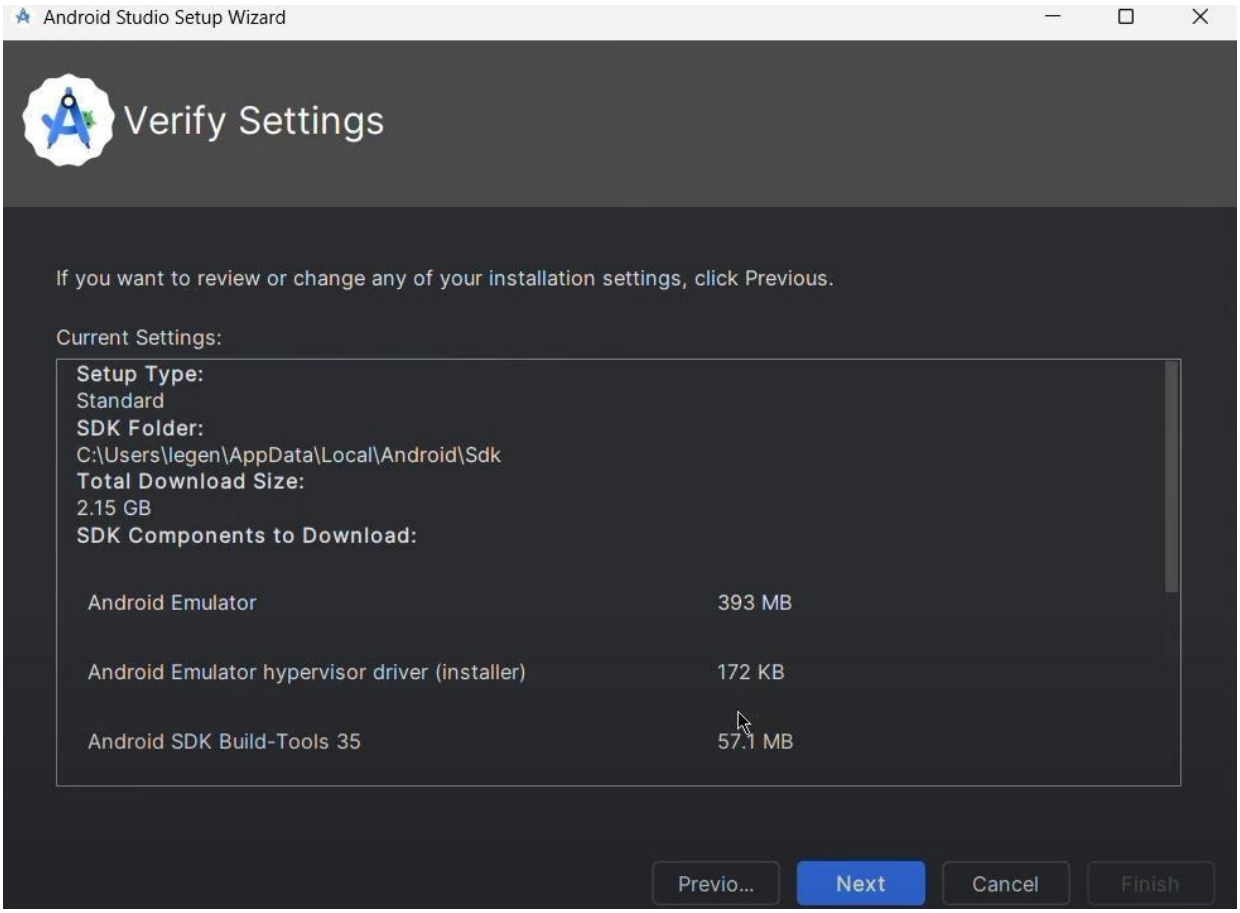

- After downloading the android studio this page come



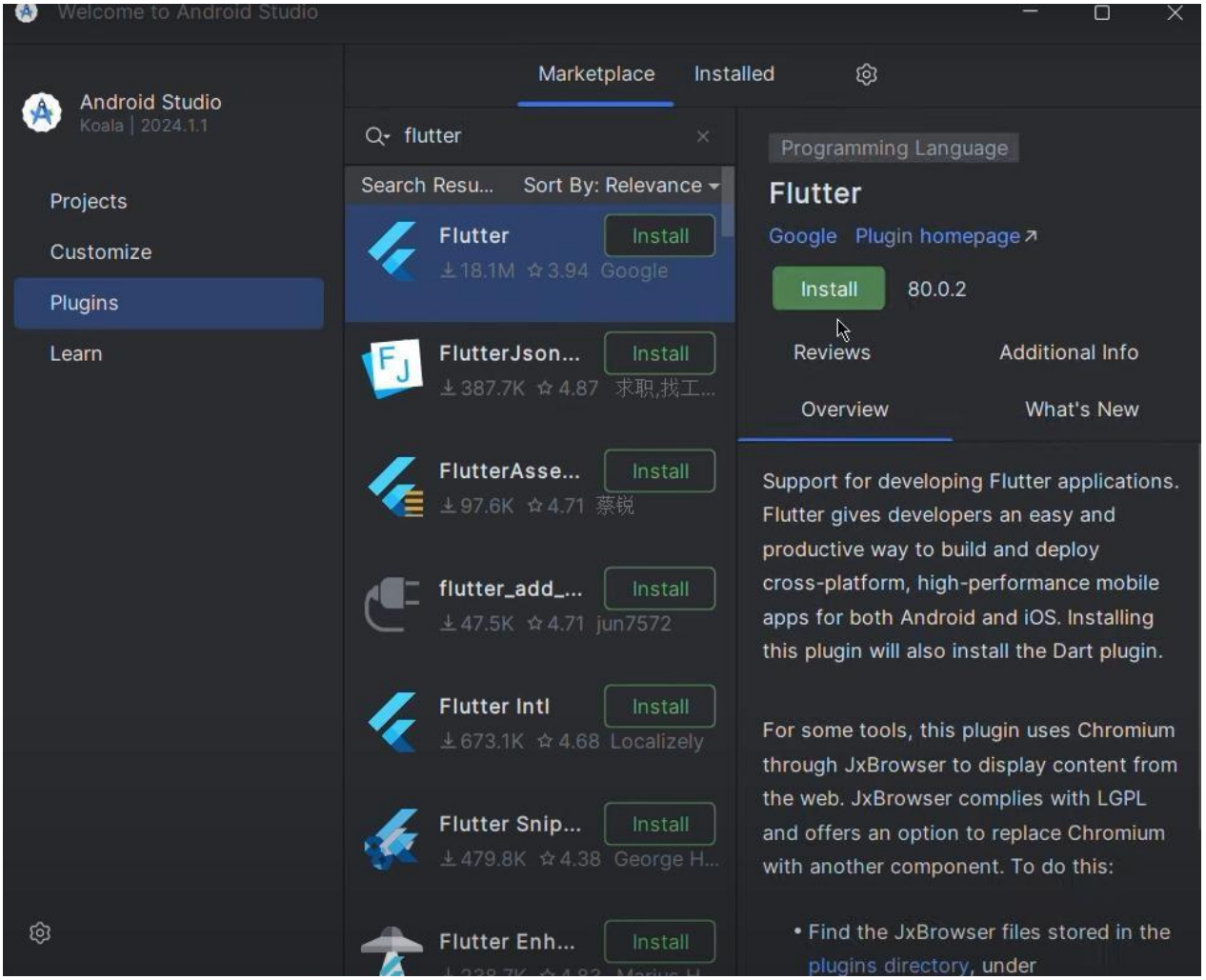
- Select Standard and click next



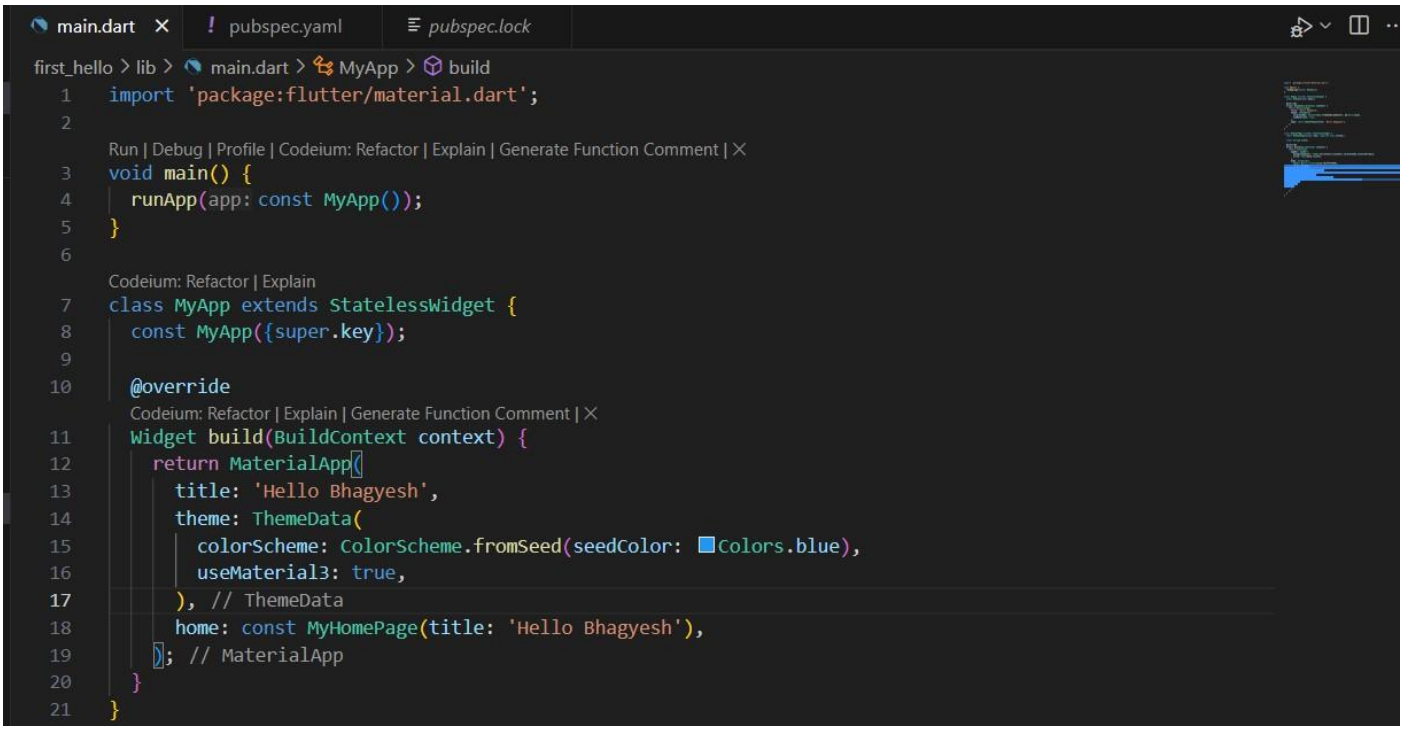

- Download the SDK and android emulator



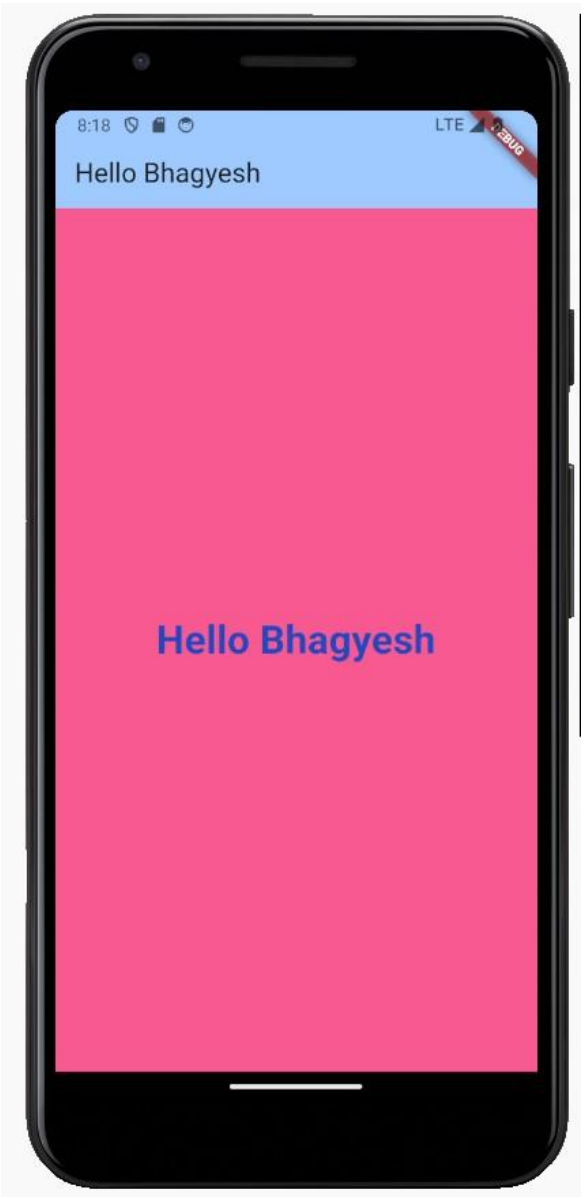
- Install Plugin in android studio




## Code:-



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

  final String title;

  @override
  Codeium: Refactor | Explain | Generate Function Comment | ×
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Theme.of(context: context).colorScheme.inversePrimary,
        title: Text(data: title),
      ), // AppBar
      body: Container(
        color: const Color(value: 0xFFF75990),
        child: Center(
          child: Text(
            data: 'Hello Bhagyesh',
            style: TextStyle(
              fontSize: 32,
              fontWeight: FontWeight.bold,
              color: Color.fromARGB(a: 255, r: 32, g: 73, b: 194),
            ), // TextStyle
          ), // Text
        ), // Center
      ), // Container
    ); // Scaffold
  }
}
```



# MAD & PWA Lab
# Journal

| | |
|---|---|
| Experiment No. | 02 |
| Experiment Title. | To design Flutter UI by including common widgets. |
| Roll No. | 35 |
| Name | Bhagyesh 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: | |

**MPL EXPERIMENT 2**

**NAME-Bhagyesh Patil**

**CLASS-D15A**

**ROLL NO-35**

**AIM: To design Flutter UI by including common widgets**

## 1,Introduction

Flutter is an open-source UI framework by Google that enables developers to build cross-platform applications with a single codebase. The UI in Flutter is built using widgets, which are the fundamental building blocks of the application. Widgets in Flutter can be classified into various categories, such as structural widgets, layout widgets, interactive widgets, and styling widgets. These widgets help in designing complex UI components efficiently and responsively.

## 2.Types of widgets

1. **StatefulWidget**
   These widgets have mutable state that can change during the lifetime of the widget. The `SplashScreen` class is a StatefulWidget because its state (like the transition to the next screen) changes over time.

   - **SplashScreen**: The main widget of the splash screen that transitions to the main screen after a delay.

2. **Scaffold**
   A structural widget that provides the basic structure for the screen, including the app bar, body, and other elements.

   - **Scaffold**: The root structure of the screen with a background color and a child widget.

3. **Center**
   A layout widget that centers its child widget within itself.

   - **Center**: Centers the child (the image in this case) within the available space.

4. **Image**
   A widget used to display an image.

   - **Image.asset**: Displays an image from the assets directory. This is used to show the splash screen image.

5. **Navigator**
   A navigation widget that manages a stack of pages and provides methods to navigate between them.

- **Navigator.pushReplacement**: Used to navigate to the `MainScreen` after a delay, replacing the current splash screen.

### 3. Interactive Widget

- **ElevatedButton**: A button that raises when pressed, used for primary actions.
- **TextButton**: A flat button that displays text and responds to taps.
- **IconButton**: A button that displays an icon and triggers actions on tap.
- **GestureDetector**: A widget that detects gestures like taps, long presses, and swipes.
- **TextField**: A widget for accepting user text input, often used in forms.
- **Switch**: A toggle widget that switches between two states, like on/off.
- **Slider**: A widget for selecting a value from a range by dragging a thumb.
- **Checkbox**: A widget for selecting or deselecting an option with a checkmark.
- **Radio**: A button group where users can select a single option from multiple choices.
- **InkWell**: A material design widget that adds a ripple effect when tapped.
- **Draggable**: A widget that allows users to drag and move elements around the screen.
- **Form**: A container widget for managing and validating form fields.
- **CupertinoSwitch**: An iOS-style switch widget for toggling values.
- **ListView**: A scrollable list of items, often with interactive elements.
- **DropdownButton**: A widget that displays a dropdown list for selecting an option.

### 4. Styling & Display Widgets

These widgets help in styling and displaying content.

- **Text**: Displays text content.
- **Image**: Displays an image from assets or a URL.
- **Icon:** Displays an icon from Material or custom icons.
- **Card**: A box with elevation and rounded corners for content display.
- **Chip**: A small, compact element that represents information.

**5. Scrolling & List Widgets**

These widgets handle large lists and scrolling content.

● **ListView:** Displays a scrollable list of items.

● **GridView:** Displays items in a grid format.

● **SingleChildScrollView:** Makes a child widget scrollable.

● **PageView:** Creates swipeable pages.

❖ **Code:-**

**Homepage.dart**

```
import 'package:carousel_slider/carousel_slider.dart';
import 'package:flutter/material.dart';
import 'package:rapido_app_clone/screen/search_screen.dart';
import 'package:rapido_app_clone/utils/custom_text_style.dart';
import 'package:rapido_app_clone/widgets/myExploreCard.dart';
import 'package:rapido_app_clone/widgets/my_save_more_card.dart';
class HomeScreen extends StatefulWidget {
  const HomeScreen({super.key});
  @override
  State<HomeScreen> createState() => _HomeScreenState();
}
class _HomeScreenState extends State<HomeScreen> {
  /// here we create list image to shoe auto image slider
  List<Widget> bannerImage = [
    Container(
      height: 50,
```

```
  width: double.infinity,
  decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(10),
      border: Border.all(width: 1, color: Colors.blue),
      image: const DecorationImage(
          image: AssetImage(
            "assets/image/rapido banner 2.png",
          ),
          fit: BoxFit.cover)),
),
Container(
  height: 50,
  width: double.infinity,
  decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(10),
      border: Border.all(width: 1, color: Colors.blue),
      image: const DecorationImage(
          image: AssetImage(
            "assets/image/rapido banner 3.png",
          ),
          fit: BoxFit.cover)),
),
Container(
  height: 50,
  width: double.infinity,
  decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(10),
      border: Border.all(width: 1, color: Colors.blue),
```

```
        image: const DecorationImage(
            image: AssetImage(
              "assets/image/rapido poster 1.png",
            ),
            fit: BoxFit.cover)),
  ),
];
List<Map<String, dynamic>> data = [
  {"image": "assets/image/parcel.png", "title": "Parcel"},
  {"image": "assets/image/auto.png", "title": "Auto"},
  {"image": "assets/image/cab economy.png", "title": "Cab Economy"},
  {"image": "assets/image/motorbike.png", "title": "Bike"},
  {"image": "assets/image/lite bike.png", "title": "Lite Bike"},
  {"image": "assets/image/cab premium.png", "title": "Cab Premium"}
];
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: Colors.white,
      title: InkWell(
        /// when click on search box open search screen
        onTap: () {
          Navigator.push(context,
              MaterialPageRoute(builder: (context) => SearchScreen()));
        },
        child: Container(
            height: 40,
```

```
        width: 280,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(20),
          color: Colors.black12.withOpacity(0.05)),
        child: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 10.0),
          child: Row(
            children: [
              const Icon(Icons.search_rounded),
              const SizedBox(
                width: 10,
              ),
              Hero(
                tag: "anim",
                child: Text(
                  "Where are you going?",
                  style: myTextStyle20(fontWeight: FontWeight.bold),
                ),
              ),
            ],
          ),
        )),
  ),
),
backgroundColor: Colors.white,
body: SingleChildScrollView(
child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
```

```
children: [
  Padding(
    padding: const EdgeInsets.symmetric(horizontal: 10.0),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Text(
          "Explore",
          style: myTextStyle18(fontWeight: FontWeight.bold),
        ),

        /// View All Button , click on onPress bottom Sheet
        TextButton(
            onPressed: () {
              showModalBottomSheet(
              showDragHandle: true,
              context: context,
                builder: (context) {
                  return  Container(
                  height: 300,
                    width: double.infinity,
                    child: Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        Padding(
                          padding: const EdgeInsets.symmetric(
                              horizontal: 10.0),
                          child: Text(
```

```
    "All Services ",
    style: myTextStyle22(
        fontWeight: FontWeight.bold),
  ),
),
const SizedBox(
  height: 10,
),
Expanded(
  child: GridView.builder(
    itemCount: data.length,
    gridDelegate:
        const SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 4),
    itemBuilder: (context, index) {
      /// here we call my explore card
      return InkWell(
        onTap: () {
          Navigator.pop(context);
        },
        child: MyExploreCard(
          imagePath: data[index]['image'],
          title: data[index]['title'],
        ),
      );
    },
  ),
)
```

```
                ],
              ),
            );
          },
        );
      },
      child: Text(
        "View All",
        style: myTextStyle15(
            textColor: Colors.blueAccent,
            fontWeight: FontWeight.bold),
      ))
  ],
 ),
),

/// Here we show Rapido servie
/// Create custom Card ExploreCard
/// Here we call
Padding(
 padding: const EdgeInsets.symmetric(horizontal: 10.0),
 child: Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
   MyExploreCard(
    imagePath: "assets/image/parcel.png",
    title: "Parcel",
   ),
```

```
      MyExploreCard(
        imagePath: "assets/image/auto.png",
        title: "Auto",
      ),
      MyExploreCard(
        imagePath: "assets/image/cab economy.png",
        title: "Cab Economy",
      ),
      MyExploreCard(
        imagePath: "assets/image/motorbike.png",
        title: "Bike",
      ),
    ],
  ),
),
const SizedBox(
  height: 20,
),

/// here we show auto image slider for banner showing

Padding(
  padding: const EdgeInsets.all(8.0),
  child: CarouselSlider(
      items: bannerImage,
      options: CarouselOptions(
      height: 100,
        aspectRatio: 16 / 3,
```

```
        viewportFraction: 1,
        initialPage: 0,
        enableInfiniteScroll: true,
        reverse: true,
        autoPlay: true,
        autoPlayInterval: const Duration(seconds: 3),
        autoPlayAnimationDuration:
            const Duration(milliseconds: 900),
        autoPlayCurve: Curves.fastOutSlowIn,
        enlargeCenterPage: true,
        enlargeFactor: 0.3,
        scrollDirection: Axis.horizontal,
      )),
),
const SizedBox(
  height: 20,
),

/// SAVE MORE WITH RAPIDO
Padding(
  padding: const EdgeInsets.only(left: 10.0),
  child: Text(
    "Save More With Rapido",
    style: myTextStyle20(textColor: Colors.black87),
  ),
),

const SizedBox(
```

```
  height: 10,
),
SingleChildScrollView(
  scrollDirection: Axis.horizontal,
  child: Padding(
    padding: const EdgeInsets.symmetric(horizontal: 10.0),
    child: Row(
      children: [
        /// Airport
        MySaveMoreCard(
            title: "Airport",
            imagePath: "assets/image/AirPort-icon.png"),
        const SizedBox(
          width: 15,
        ),

        /// Railway Station
        MySaveMoreCard(
            title: "Railway station",
            imagePath: "assets/image/railway.png"),
        const SizedBox(
          width: 15,
        ),

        /// Bus stand
        MySaveMoreCard(
            title: "Bus Stand",
            imagePath: "assets/image/bus-removebg-preview.png"),
```

```
      ],
    ),
  ),
),

/// here we show single poster
Padding(
  padding: const EdgeInsets.all(10.0),
  child: Container(
    height: 150,
    decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(15),
        image: const DecorationImage(
            image: AssetImage(
              "assets/image/rapiod poster 2.png",
            ),
            fit: BoxFit.cover)),
  ),
),
const SizedBox(
  height: 20,
),

/// last background poster show
Stack(children: [
  Image.asset(
    "assets/image/childrens-pattern-600nw-367289369-removebg-preview.png",
    color: Colors.black38,
```

```
),
Positioned(
  left: 0,
  top: 200,
  child: Container(
   color: Colors.white70.withOpacity(0.9),
   child: Padding(
    padding: const EdgeInsets.symmetric(horizontal: 10.0),
    child: Column(
     mainAxisAlignment:  MainAxisAlignment.start,
     crossAxisAlignment: CrossAxisAlignment.start,
     children: [
      Text(
       "#goRapido",
       style: myTextStyle40(
          fontWeight: FontWeight.bold,
          textColor: Colors.black54),
      ),
      Row(
       mainAxisAlignment: MainAxisAlignment.start,
       children: [
        Image.asset(
         "assets/icon/india.png",
         height: 20,
         width: 20,
        ),
        const SizedBox(
         width: 5,
```

```
            ),
            Text(
              "Made For India",
              style: myTextStyle18(),
            )
          ],
        ),
        Row(
          mainAxisAlignment: MainAxisAlignment.start,
          children: [
            Image.asset(
              "assets/icon/love (1).png",
              height: 20,
              width: 20,
            ),
            const SizedBox(
              width: 5,
            ),
            Text(
              "Crafted in Bengaluru",
              style: myTextStyle18(),
            )
          ],
        ),
      ],
    ),
  ),
))
```

```
        ])
      ],
    ),
  ),      drawer: Drawer(
    width: double.infinity,
    backgroundColor: Colors.white,
    child: ListView(
      children: [
        DrawerHeader(
          child: Column(
        children: [
        Row(
            children: [
              IconButton(
                  onPressed: () {
                    Navigator.pop(context);
                  },
                  icon: const Icon(Icons.arrow_back_rounded)),
              Text(
                "Menu",
                style: myTextStyle18(fontWeight: FontWeight.bold),
              )
            ],
          ),
          Container(
            height: 80,
            decoration: BoxDecoration(
                boxShadow: const [
```

```
          BoxShadow(
              color: Colors.black,
              blurRadius: 1,
              spreadRadius: 1),
        ],
        borderRadius: BorderRadius.circular(10),
        color: Colors.white,
        border: Border.all(width: 1, color: Colors.black26)),
    child: Column(
      children: [
        ListTile(
          leading: Image.asset(
            "assets/icon/user (2).png",
            height: 40,
          ),
          title: Text(
            "App Creator",
            style: myTextStyle15(fontWeight: FontWeight.bold),
          ),
          subtitle: Text(
            "1234567890",
            style: myTextStyle15(),
          ),
          trailing: const Icon(Icons.navigate_next_rounded),
        )
      ],
    ),
  )
```

```
  ],
)),
ListTile(
  title: Text(
  "Help",
    style: myTextStyle18(fontWeight: FontWeight.bold),
  ),
  leading: Image.asset(
    "assets/icon/handshake.png",
    height: 30,
  ),
  trailing: const Icon(Icons.navigate_next_outlined),
),
const Divider(),

ListTile(
  title: Text(
    "Parcel - Send Items",
    style: myTextStyle18(fontWeight: FontWeight.bold),
  ),
  leading: Image.asset(
    "assets/icon/trolley (1).png",
    height: 30,
  ),
  trailing: const Icon(Icons.navigate_next_outlined),
),
const Divider(),
```

```
ListTile(
  title: Text(
  "Payment",
    style: myTextStyle18(fontWeight: FontWeight.bold),
  ),
  leading: Image.asset(
    "assets/icon/credit-card.png",
    height: 30,
  ),
  trailing: const Icon(Icons.navigate_next_outlined),
),
const Divider(),

ListTile(
  title: Text(
    "My Rides",
    style: myTextStyle18(fontWeight: FontWeight.bold),
  ),
  leading: Image.asset(
    "assets/icon/refresh.png",
    height: 30,
  ),
  trailing: const Icon(Icons.navigate_next_outlined),
),
const Divider(),
ListTile(
  title: Text(
    "Safety",
```

```
    style: myTextStyle18(fontWeight: FontWeight.bold),
  ),
  leading: Image.asset(
    "assets/icon/verified.png",
    height: 30,
  ),
  trailing: const Icon(Icons.navigate_next_outlined),
),
const Divider(),

ListTile(
  title: Text(
    "Refer and Earn",
    style: myTextStyle18(fontWeight: FontWeight.bold),
  ),
  subtitle: Text(
    "Get 50",
    style: myTextStyle15(),
  ),
  leading: Image.asset(
    "assets/icon/cost.png",
    height: 30,
  ),
  trailing: const Icon(Icons.navigate_next_outlined),
),
const Divider(),

ListTile(
```

```
  title: Text(
    "My Reward",
    style: myTextStyle18(fontWeight: FontWeight.bold),
  ),
  leading: Image.asset(
    "assets/icon/medal.png",
    height: 30,
  ),
  trailing: const Icon(Icons.navigate_next_outlined),
),
const Divider(),

ListTile(
  title: Text(
    "Power Pass",
    style: myTextStyle18(fontWeight: FontWeight.bold),
  ),
  leading: Image.asset(
    "assets/icon/id-card.png",
    height: 30,
  ),
  trailing: const Icon(Icons.navigate_next_outlined),
),
const Divider(),
ListTile(
  title: Text(
    "Rapido Coin",
    style: myTextStyle18(fontWeight: FontWeight.bold),
```

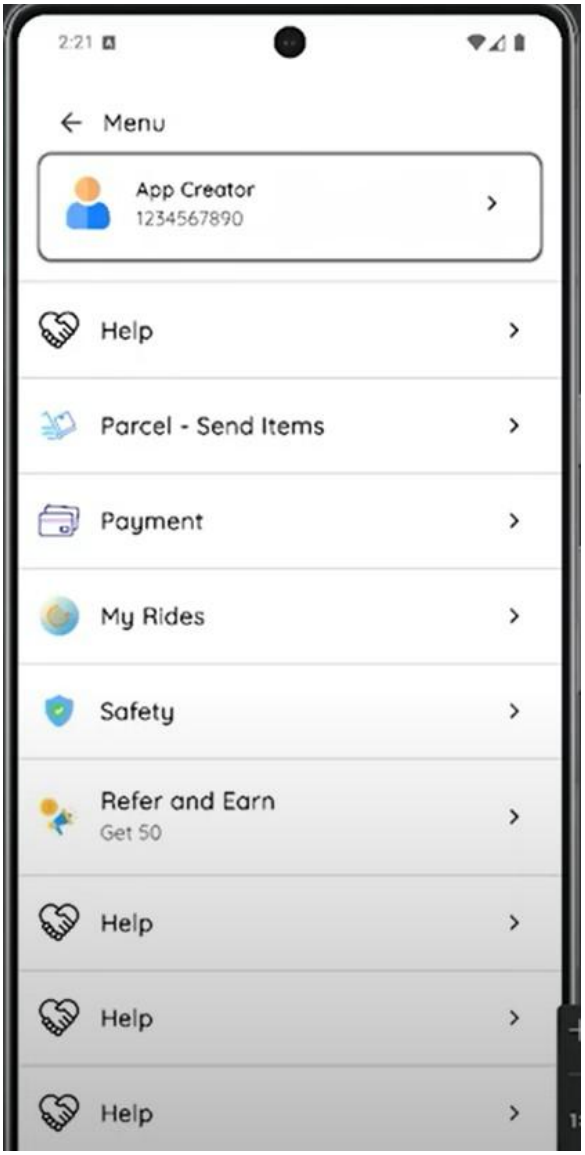
```
  ),
  leading: Image.asset(
    "assets/icon/dollar.png",
    height: 30,
  ),
  trailing: const Icon(Icons.navigate_next_outlined),
),
const Divider(),
ListTile(
  title: Text(
    "Notifications",
    style: myTextStyle18(fontWeight: FontWeight.bold),
  ),
  leading: Image.asset(
    "assets/icon/notification.png",
    height: 30,
  ),
  trailing: const Icon(Icons.navigate_next_outlined),
),
const Divider(),
ListTile(
  title: Text(
    "Claims",
    style: myTextStyle18(fontWeight: FontWeight.bold),
  ),
  leading: Image.asset(
    "assets/icon/security.png",
    height: 30,
```

```
  ),
  trailing: const Icon(Icons.navigate_next_outlined),
),
const Divider(),

ListTile(
  title: Text(
  "Setting",
    style: myTextStyle18(fontWeight: FontWeight.bold),
  ),
  leading: Image.asset(
    "assets/icon/setting.png",
    height: 30,
  ),
  trailing: const Icon(Icons.navigate_next_outlined),
),
const Divider(),

Padding(
  padding: const EdgeInsets.all(8.0),
  child: Container(
    height: 100,
    decoration: BoxDecoration(
        color: const Color(0x66ffcc75),
        borderRadius: BorderRadius.circular(16)),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
```

```
          Padding(
           padding: const EdgeInsets.only(left: 8.0),
           child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
             Text(
              "Earn money with Rapido",
              style: myTextStyle15(fontWeight: FontWeight.bold),
             ),
             Text(
              "Become a Captain!",
              style: myTextStyle15(),
             ),
            ],
           ),
          ),


                image: AssetImage("assets/image/Rapido_boy.png"),
                fit: BoxFit.cover)),
         )
        ],
       ),
      ),
     )
```

❖ **Output :-**


1:54
Where are you going?
Explore
View All
Parcel
Auto
Cab Economy
Bike
BIKE
WALI TAXI
sabse
saxi
Save More With Rapido
Airport
Railway station
Rapido
ka vaada


2:21
Menu
App Creator
1234567890
Help
Parcel - Send Items
Payment
My Rides
Safety
Refer and Earn
Get 50
Help
Help
Help

# MAD & PWA Lab

# Journal

| | |
|---|---|
| Experiment No. | 03 |
| Experiment Title. | To include icons, images, fonts in Flutter app |
| Roll No. | 35 |
| Name | Bhagyesh 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: | |

# EXPERIMENT NO 3

**Aim :- To include icon,image,fonts in Flutter app**

**Theory :-**

In Flutter, images play an essential role in enhancing the UI and user experience. Images can be loaded from different sources, including assets, networks, memory, and files.

Flutter provides the Image widget, which supports multiple ways of displaying images in an application. The most common ways to add images in Flutter are:

- **Asset Images:** Images stored locally in the project directory.
- **Network Images:** Images loaded from an online URL.
- **File Images:** Images selected from a device's storage.
- **Memory Images:** Images loaded from raw byte data.

Flutter makes it easy to display images efficiently using caching and other optimization techniques.

➢ **Adding Images in Flutter**

Displaying images in a Flutter application involves different methods based on the image source. Below are the common ways to handle images in Flutter:

**1. Using Asset Images (Local Images)**

Asset images are stored within the app's project directory and bundled with the application.

**Steps to Add Asset Images:**

1. **Place the Image in the Assets Folder**
    - Create an assets/images folder inside the project.
    - Add the image file inside the folder (e.g., assets/images/my_image.png).
2. **Declare the Image in pubspec.yaml**
    - Register the image in the pubspec.yaml file:

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

    - Run flutter pub get to update dependencies.
3. **Use the Image.asset Widget in Your Code**

```
Image.asset('assets/images/my_image.png')
```

**2. Using Network Images (Online Images)**

Network images are loaded from an external URL and require an internet connection.

**Displaying a Network Image**

Image.network('https://example.com/my_image.png')

**3. Using File Images (From Device Storage)**

File images allow users to pick an image from their device and display it in the app. The image_picker package is commonly used for this purpose.

**Example Usage with image_picker**

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

class ImagePickerExample extends StatefulWidget {
  @override
  _ImagePickerExampleState createState() => _ImagePickerExampleState();
}

class _ImagePickerExampleState extends State<ImagePickerExample> {
  File? _image;

  Future<void> pickImage() async {
    final pickedFile = await ImagePicker().pickImage(source: ImageSource.gallery);
    if (pickedFile != null) {
      setState(() {
        _image = File(pickedFile.path);
      });
    }
  }

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

```
    return Scaffold(
      appBar: AppBar(title: Text('Pick an Image')),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            _image != null ? Image.file(_image!) : Text('No image selected'),
            ElevatedButton(onPressed: pickImage, child: Text('Pick Image')),
          ],
        ),
      ),
    );
  }
}
```

**4. Using Memory Images (Raw Byte Data)**

Sometimes, images may be loaded from raw binary data using MemoryImage:

dart

CopyEdit

```
Uint8List imageData = ...; // Some image data in bytes
Image.memory(imageData)
```

**Image Properties**

All image types (Image.asset, Image.network, Image.file, Image.memory) support various properties like:

```
Image.asset(
  'assets/images/my_image.png',
  width: 200,
  height: 200,
  fit: BoxFit.cover,
)
```

**BoxFit Options:**

- BoxFit.cover – Covers the container, cropping if necessary.
- BoxFit.contain – Fits the image inside the container without cropping.
- BoxFit.fill – Stretches the image to fill the container.
- BoxFit.fitWidth – Scales the image to match the width.
- BoxFit.fitHeight – Scales the image to match the height.

Code :-

```
import 'package:carousel_slider/carousel_slider.dart';
import 'package:flutter/material.dart';
import 'package:rapido_app_clone/screen/search_screen.dart';
import 'package:rapido_app_clone/utils/custom_text_style.dart';
import 'package:rapido_app_clone/widgets/myExploreCard.dart';
import 'package:rapido_app_clone/widgets/my_save_more_card.dart';


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


  @override
  State<HomeScreen> createState() => _HomeScreenState();
}


class _HomeScreenState extends State<HomeScreen> {
  /// here we create list image to shoe auto image slider
  List<Widget> bannerImage = [
    Container(
      height: 50,
      width: double.infinity,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(10),
        border: Border.all(width: 1, color: Colors.blue),
        image: const DecorationImage(
          image: AssetImage(
            "assets/image/rapido banner 2.png",
          ),
          fit: BoxFit.cover)),
    ),
    Container(
      height: 50,
      width: double.infinity,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(10),
```

```
        border: Border.all(width: 1, color:
Colors.blue),
        image: const DecorationImage(
          image: AssetImage(
            "assets/image/rapido banner 3.png",
          ),
          fit: BoxFit.cover)),
    ),
    Container(
      height: 50,
      width: double.infinity,
      decoration: BoxDecoration(
        borderRadius:
BorderRadius.circular(10),
        border: Border.all(width: 1, color:
Colors.blue),
        image: const DecorationImage(
          image: AssetImage(
            "assets/image/rapido poster 1.png",
          ),
          fit: BoxFit.cover)),
    ),
  ];

  /// Image and title list
  List<Map<String, dynamic>> data = [
    {"image": "assets/image/parcel.png",
"title": "Parcel"},
    {"image": "assets/image/auto.png", "title":
"Auto"},
    {"image": "assets/image/cab
economy.png", "title": "Cab Economy"},
    {"image": "assets/image/motorbike.png",
"title": "Bike"},
    {"image": "assets/image/lite bike.png",
"title": "Lite Bike"},
    {"image": "assets/image/cab
premium.png", "title": "Cab Premium"}
  ];
  @override
  Widget build(BuildContext context) {
    return Scaffold(

      appBar: AppBar(
        backgroundColor: Colors.white,
        title: InkWell(
          /// when click on search box open
search screen
          onTap: () {
            Navigator.push(context,
                MaterialPageRoute(builder: (context)
=> SearchScreen()));
          },

          /// here we apply hero animation
          child: Container(
            height: 40,
            width: 280,
            decoration: BoxDecoration(
                borderRadius:
BorderRadius.circular(20),
                color:
Colors.black12.withOpacity(0.05)),
            child: Padding(
```

```
            padding: const
EdgeInsets.symmetric(horizontal: 10.0),
            child: Row(
              children: [
                const Icon(Icons.search_rounded),
                const SizedBox(
                  width: 10,
                ),
                Hero(
                  tag: "anim",
                  child: Text(
                    "Where are you going?",
                    style:
myTextStyle20(fontWeight: FontWeight.bold),
                  ),
                ),
              ],
            ),
          )),
      ),
    ),
    backgroundColor: Colors.white,

    ///
_____BODY_____
///
    body: SingleChildScrollView(
      child: Column(
        crossAxisAlignment:
CrossAxisAlignment.start,
        children: [
          Padding(
            padding: const
EdgeInsets.symmetric(horizontal: 10.0),
            child: Row(
              mainAxisAlignment:
MainAxisAlignment.spaceBetween,
              children: [
                Text(
                  "Explore",
                  style: myTextStyle18(fontWeight:
FontWeight.bold),
                ),


                /// View All Button , click on
onPress bottom Sheet
                ///
_____VIEW
ALL_____///
                TextButton(
                  onPressed: () {
                    showModalBottomSheet(
                      showDragHandle: true,
                      context: context,
                      builder: (context) {
                        return Container(
                          height: 300,
                          width: double.infinity,
                          child: Column(
                            crossAxisAlignment:
CrossAxisAlignment.start,
                            children: [
                              Padding(
                                padding: const
EdgeInsets.symmetric(
```

```
                    horizontal: 10.0),
                child: Text(
                  "All Services ",
                  style: myTextStyle22(
                      fontWeight:
FontWeight.bold),
                ),
              ),
              const SizedBox(
                height: 10,
              ),
              Expanded(
                child: GridView.builder(
                  itemCount: data.length,
                  gridDelegate:
                      const
SliverGridDelegateWithFixedCrossAxisCount(
                          crossAxisCount: 4),
                  itemBuilder: (context,
index) {
                    /// here we call my
explore card
                    return InkWell(
                      onTap: () {

Navigator.pop(context);
                      },
                      child: MyExploreCard(
                        imagePath:
data[index]['image'],
                        title:
data[index]['title'],
                      ),
                    );
                  },
                ),
              )
            ],
          ),
        );
      },
      child: Text(
        "View All",
        style: myTextStyle15(
            textColor: Colors.blueAccent,
            fontWeight: FontWeight.bold),
      ))
  ],
 ),
),

/// Here we show Rapido servie
/// Create custom Card ExploreCard
/// Here we call
Padding(
  padding: const
EdgeInsets.symmetric(horizontal: 10.0),
  child: Row(
    mainAxisAlignment:
MainAxisAlignment.spaceBetween,
    children: [
      MyExploreCard(
```

```
            imagePath:
"assets/image/parcel.png",
            title: "Parcel",
          ),
          MyExploreCard(
            imagePath:
"assets/image/auto.png",
            title: "Auto",
          ),
          MyExploreCard(
            imagePath: "assets/image/cab
economy.png",
            title: "Cab Economy",
          ),
          MyExploreCard(
            imagePath:
"assets/image/motorbike.png",
            title: "Bike",
          ),
        ],
      ),
    ),
    const SizedBox(
      height: 20,
    ),

    /// here we show auto image slider for
banner showing

    Padding(
      padding: const EdgeInsets.all(8.0),
      child: CarouselSlider(
        items: bannerImage,
        options: CarouselOptions(
          height: 100,
          aspectRatio: 16 / 3,
          viewportFraction: 1,
          initialPage: 0,
          enableInfiniteScroll: true,
          reverse: true,
          autoPlay: true,
          autoPlayInterval: const
Duration(seconds: 3),
          autoPlayAnimationDuration:
              const Duration(milliseconds:
900),
          autoPlayCurve:
Curves.fastOutSlowIn,
          enlargeCenterPage: true,
          enlargeFactor: 0.3,
          scrollDirection: Axis.horizontal,
        )),
    ),
    const SizedBox(
      height: 20,
    ),

    /// SAVE MORE WITH RAPIDO
    Padding(
      padding: const EdgeInsets.only(left:
10.0),
      child: Text(
        "Save More With Rapido",
```

```
          style: myTextStyle20(textColor:
Colors.black87),
          ),
        ),

        /// here we create custom card
        /// here we call my save card

        const SizedBox(
          height: 10,
        ),
        SingleChildScrollView(
          scrollDirection: Axis.horizontal,
          child: Padding(
            padding: const
EdgeInsets.symmetric(horizontal: 10.0),
            child: Row(
              children: [
                /// Airport
                MySaveMoreCard(
                    title: "Airport",
                    imagePath:
"assets/image/AirPort-icon.png"),
                const SizedBox(
                  width: 15,
                ),

                /// Railway Station
                MySaveMoreCard(
                    title: "Railway station",
                    imagePath:
"assets/image/railway.png"),
                const SizedBox(
                  width: 15,
                ),

                /// Bus stand
                MySaveMoreCard(
                    title: "Bus Stand",
                    imagePath: "assets/image/bus-
removebg-preview.png"),
              ],
            ),
          ),
        ),

        /// here we show single poster
        Padding(
          padding: const EdgeInsets.all(10.0),
          child: Container(
            height: 150,
            decoration: BoxDecoration(
                borderRadius:
BorderRadius.circular(15),
                image: const DecorationImage(
                    image: AssetImage(
                      "assets/image/rapiod poster
2.png",
                    ),
                    fit: BoxFit.cover)),
          ),
        ),
        const SizedBox(
```

```
          height: 20,
        ),

        /// last background poster show
        Stack(children: [
          Image.asset(
            "assets/image/childrens-pattern-600nw-367289369-removebg-preview.png",
            color: Colors.black38,
          ),
          Positioned(
            left: 0,
            top: 200,
            child: Container(
              color: Colors.white70.withOpacity(0.9),
              child: Padding(
                padding: const EdgeInsets.symmetric(horizontal: 10.0),
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.start,
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text(
                      "#goRapido",
                      style: myTextStyle40(
                          fontWeight: FontWeight.bold,
                          textColor: Colors.black54),
                    ),
                    Row(
                      mainAxisAlignment: MainAxisAlignment.start,
                      children: [
                        Image.asset(
                          "assets/icon/india.png",
                          height: 20,
                          width: 20,
                        ),
                        const SizedBox(
                          width: 5,
                        ),
                        Text(
                          "Made For India",
                          style: myTextStyle18(),
                        )
                      ],
                    ),
                    Row(
                      mainAxisAlignment: MainAxisAlignment.start,
                      children: [
                        Image.asset(
                          "assets/icon/love (1).png",
                          height: 20,
                          width: 20,
                        ),
                        const SizedBox(
                          width: 5,
                        ),
                        Text(
```

```
                    "Crafted in Bengaluru",
                    style: myTextStyle18(),
                  )
                ],
              ),
            ],
          ),
        ),
      ))
    ])
  ],
 ),
),


drawer: Drawer(
 width: double.infinity,
 backgroundColor: Colors.white,
 child: ListView(
  children: [
   DrawerHeader(
    child: Column(
    children: [
    Row(
      children: [
       IconButton(
        onPressed: () {
         Navigator.pop(context);
        },
        icon: const
Icon(Icons.arrow_back_rounded)),
        Text(
         "Menu",
         style: myTextStyle18(fontWeight:
FontWeight.bold),
        )
       ],
      ),


     /// back icon and manu text
     Container(
      height: 80,
      decoration: BoxDecoration(
        boxShadow: const [
        BoxShadow(
          color: Colors.black,
          blurRadius: 1,
          spreadRadius: 1),
        ],
        borderRadius:
BorderRadius.circular(10),
        color: Colors.white,
        border: Border.all(width: 1, color:
Colors.black26)),
       child: Column(
        children: [
        ListTile(
         leading: Image.asset(
          "assets/icon/user (2).png",
          height: 40,
         ),
         title: Text(
```

```
                "App Creator",
                style:
myTextStyle15(fontWeight: FontWeight.bold),
              ),
              subtitle: Text(
                "1234567890",
                style: myTextStyle15(),
              ),
              trailing: const
Icon(Icons.navigate_next_rounded),
            )
          ],
        ),
      )
    ],
  )),

  /// Drawer Body

  ListTile(
    title: Text(
      "Help",
      style: myTextStyle18(fontWeight:
FontWeight.bold),
    ),
    leading: Image.asset(
      "assets/icon/handshake.png",
      height: 30,
    ),
    trailing: const
Icon(Icons.navigate_next_outlined),
  ),
  const Divider(),

  ListTile(
    title: Text(
      "Parcel - Send Items",
      style: myTextStyle18(fontWeight:
FontWeight.bold),
    ),
    leading: Image.asset(
      "assets/icon/trolley (1).png",
      height: 30,
    ),
    trailing: const
Icon(Icons.navigate_next_outlined),
  ),
  const Divider(),

  ListTile(
    title: Text(
      "Payment",
      style: myTextStyle18(fontWeight:
FontWeight.bold),
    ),
    leading: Image.asset(
      "assets/icon/credit-card.png",
      height: 30,
    ),
    trailing: const
Icon(Icons.navigate_next_outlined),
  ),
  const Divider(),
```

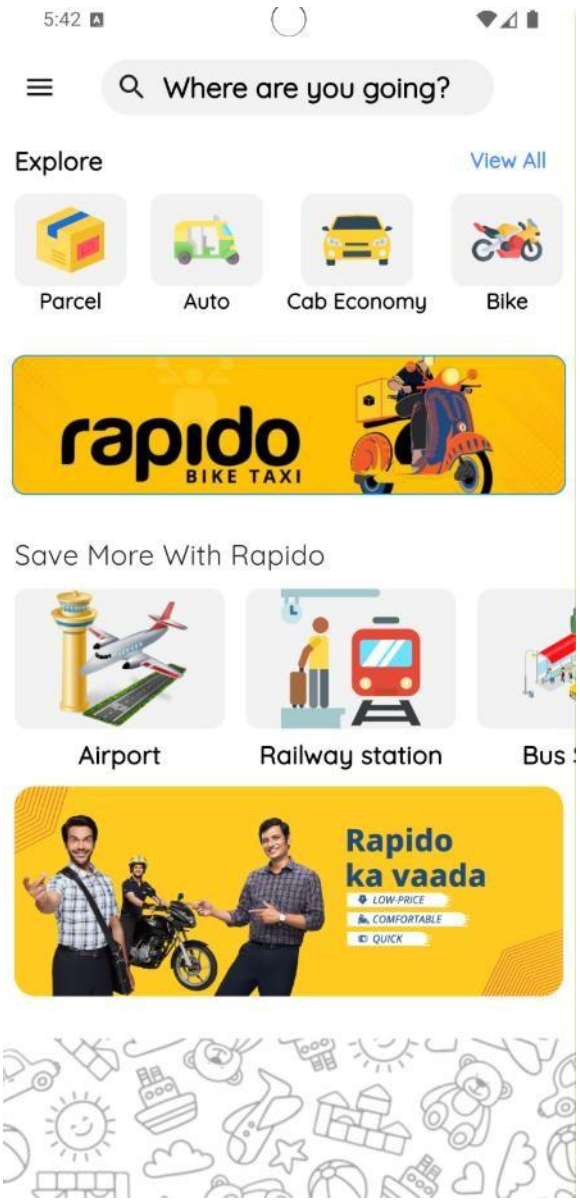
```
        ListTile(
          title: Text(
            "My Rides",
            style: myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/refresh.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),
        ListTile(
          title: Text(
            "Safety",
            style: myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/verified.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),

        ListTile(
          title: Text(
            "Refer and Earn",
            style: myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          subtitle: Text(
            "Get 50",
            style: myTextStyle15(),
          ),
          leading: Image.asset(
            "assets/icon/cost.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),

        ListTile(
          title: Text(
            "My Reward",
            style: myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/medal.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),

        ListTile(
```

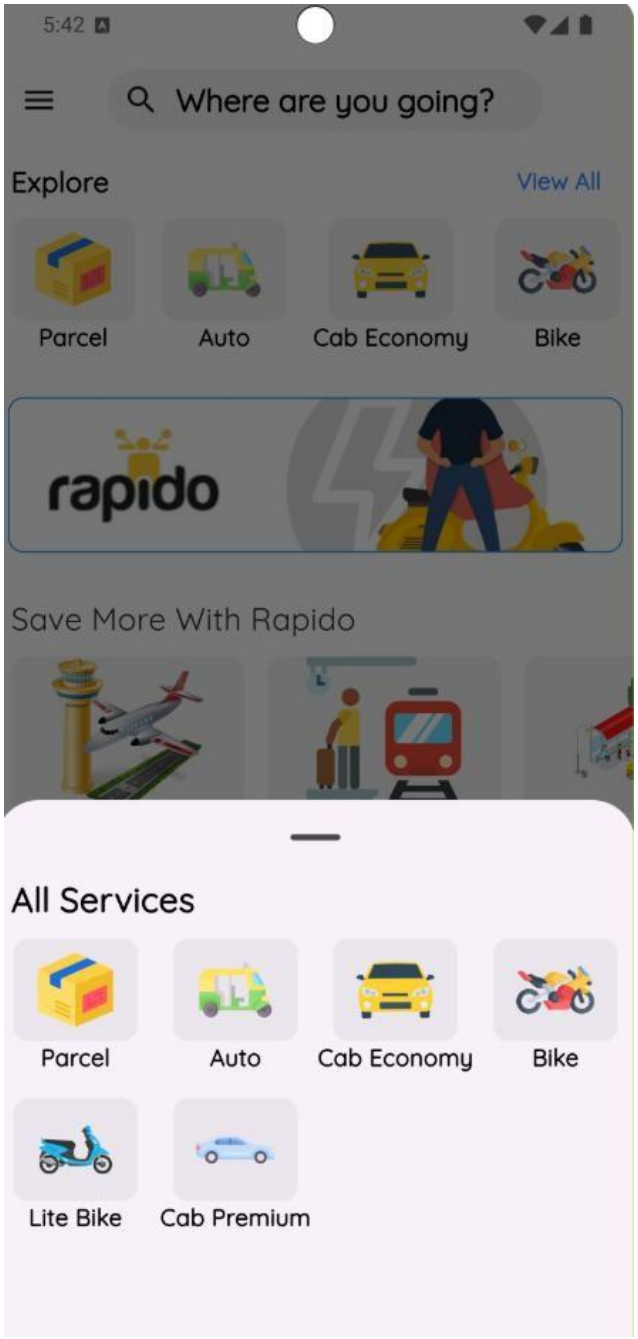
```
          title: Text(
            "Power Pass",
            style: myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/id-card.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),
        ListTile(
          title: Text(
            "Rapido Coin",
            style: myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/dollar.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),
        ListTile(
          title: Text(
            "Notifications",
            style: myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/notification.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),
        ListTile(
          title: Text(
            "Claims",
            style: myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/security.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),


        ListTile(
          title: Text(
            "Setting",
            style: myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/setting.png",
```

```
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),

        Padding(
          padding: const EdgeInsets.all(8.0),
          child: Container(
            height: 100,
            decoration: BoxDecoration(
              color: const Color(0x66ffcc75),
              borderRadius:
BorderRadius.circular(16)),
            child: Row(
              mainAxisAlignment:
MainAxisAlignment.spaceBetween,
              children: [
                Padding(
                  padding: const
EdgeInsets.only(left: 8.0),
                  child: Column(
                    mainAxisAlignment:
MainAxisAlignment.center,
                    crossAxisAlignment:
CrossAxisAlignment.start,
                    children: [
                      Text(
                        "Earn money with Rapido",
                        style:
myTextStyle15(fontWeight: FontWeight.bold),
                      ),
                      Text(
                        "Become a Captain!",
                        style: myTextStyle15(),
                      ),
                    ],
                  ),
                ),
                Container(
                  width: 200,
                  decoration: const BoxDecoration(
                    borderRadius:
BorderRadius.only(
                      topRight: Radius.circular(10),
                      bottomRight:
Radius.circular(10)),
                    image: DecorationImage(
                      image:
AssetImage("assets/image/Rapido_boy.png"),
                      fit: BoxFit.cover)),
                )
              ],
            ),
          ),
        )
      ],
    ),
  );
 }
}
```

**Output**


5:41


5:42
Where are you going?
Explore
View All
Parcel
Auto
Cab Economy
Bike
rapido
BIKE TAXI
Save More With Rapido
Airport
Railway station
Rapido
ka vaada
LOW-PRICE
COMFORTABLE
QUICK


5:42
Where are you going?
Explore
View All
Parcel
Auto
Cab Economy
Bike
rapido
Save More With Rapido
All Services
Parcel
Auto
Cab Economy
Bike
Lite Bike
Cab Premium

# MAD & PWA Lab

# Journal

| | |
|---|---|
| Experiment No. | 04 |
| Experiment Title. | To create an interactive Form using form widget |
| Roll No. | 35 |
| Name | Bhagyesh 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: | |

**MPL EXPERIMENT 4**

**NAME-Bhagyesh Patil**

**CLASS-D15A**

**ROLL NO-35**

**AIM:- To Create an interactive Form using form widget**

## 1. Introduction

Forms are a crucial component in mobile applications for collecting user input. Flutter provides the Form widget, which helps manage multiple input fields efficiently and allows validation before submission. The Form widget works in combination with TextFormField and a GlobalKey to track the form state, validate inputs, and process the collected data.

## 2. Form Widget Overview

The Form widget in Flutter acts as a container for form fields and manages their state. It provides built-in validation and submission handling.

### ❖ Key Components of a Form:

### 1. Scaffold

- Acts as the **main container** for the screen.
- Provides a structured layout with a **background color** and a dedicated **body section** for arranging widgets.

### 2. Column

- Used for arranging elements **vertically** (title, input fields, buttons).
- Ensures content is properly **centered** and spaced for better readability.

### 3. Text

- Displays headings and descriptions like **"Enter OTP"** and **"Enter the code sent to your number"**.
- Uses **bold styling** for important text and **lighter colors** for instructions.

### 4. TextField (OTP Input Fields)

- Allows users to enter the OTP.
- Can be implemented as:
  - **A single input field** for the full OTP.
  - **Multiple input boxes** for individual OTP digits.
- Accepts **numeric input** only and has a **fixed length** of 6 digits.
- Uses a **bordered container** for a clean and structured look.

### 5. GestureDetector / TextButton (Resend OTP)

- Provides a **clickable text button** for resending OTP.
- Uses a **blue color** to indicate interactivity.
- Can include a **countdown timer** to prevent immediate resends.

### 6. ElevatedButton (Verify Button)

- Initially **disabled**, gets enabled when the OTP is correctly entered.
- Uses **contrast colors** (e.g., black background, white text) for visibility.
- On press, triggers OTP verification and navigates to the next screen.

### 7. CircularProgressIndicator (Loading Spinner - Optional)

- Displays a **loading animation** after clicking "Verify."
- Helps users know that the OTP is being processed.

### 8. SnackBar / AlertDialog (Feedback Messages - Optional)

- Shows a **success message** if the OTP is correct.
- Displays an **error message** if verification fails.
- Uses **different colors** for success (green) and error (red).

### 9. StatefulWidget

- The SplashScreen is a **stateful widget** because it needs to handle a **time delay** before navigating to the next screen.
- The _SplashScreenState class manages the stat

### 10. Image.asset()

- Displays an **image logo** from the assets/images/ directory.
- The height is adjusted to **150 pixels** to fit well in the layout.

### 11. _validatePhoneNumber() (Phone Number Validation)

- A function that **enables/disables** the "Next" button based on whether the user has entered a **10-digit** phone number.
- Uses setState() to update the UI dynamically.

❖ **Code:-**

**Main_screen.dart**

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

class MainScreen extends StatefulWidget {
  @override
  _MainScreenState createState() => _MainScreenState();
}

class _MainScreenState extends State<MainScreen> {
  TextEditingController _phoneController = TextEditingController();
  bool isButtonEnabled = false;

  void _validatePhoneNumber(String value) {
    setState(() {
      isButtonEnabled = value.length == 10;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 20),
        child: Column(
          mainAxisAlignment:  MainAxisAlignment.center,
          crossAxisAlignment:  CrossAxisAlignment.center,
          children: [
            Image.asset(
```

```
  'assets/images/rapido12.jpeg',
  height: 150,
),
const SizedBox(height: 20),
const Text(
  "What's your number?",
  style: TextStyle(
  fontSize: 16,
    fontWeight: FontWeight.bold,
  ),
),
const SizedBox(height: 5),
const Text(
  "Enter your phone number to proceed",
  style: TextStyle(
    fontSize: 16,
    color: Colors.grey,
  ),
),
const SizedBox(height: 20),
Container(
  padding: EdgeInsets.symmetric(horizontal: 12),
  decoration: BoxDecoration(
    border: Border.all(color: Colors.grey),
    borderRadius: BorderRadius.circular(8),
  ),
  child: Row(
    children: [
    const Text(
        "IN +91",
        style: TextStyle(
          fontSize: 16,
```

```
            fontWeight: FontWeight.bold,
          ),
        ),
        const SizedBox(width: 10),
        Expanded(
          child: TextField(
            controller: _phoneController,
            keyboardType: TextInputType.number,
            maxLength: 10,
            onChanged: _validatePhoneNumber,
            decoration: InputDecoration(
            hintText: "0000000000",
            counterText: "",
              border: InputBorder.none,
            ),
          ),
        ),
      ],
    ),
  ),
const SizedBox(height: 20),
ElevatedButton(
  onPressed: isButtonEnabled
      ? () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => OneLastStepScreen(),
            ),
          );
        }
      : null,
```

```
            style: ElevatedButton.styleFrom(
              backgroundColor: Colors.black,
              foregroundColor: Colors.white,
              padding: EdgeInsets.symmetric(vertical: 14, horizontal: 120),
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(8),
              ),
            ),
            child: Text(
              "Next",
              style: TextStyle(
                fontSize: 16,
                fontWeight: FontWeight.bold,
              ),
            ),
          ),
        ],
      ),
    ),
  );
 }
}
```

**One_last_step_screen.dart**

```
import 'package:flutter/material.dart';
class OneLastStepScreen extends StatefulWidget {
 @override
 _OneLastStepScreenState createState() => _OneLastStepScreenState();
}
class _OneLastStepScreenState extends State<OneLastStepScreen> {
 TextEditingController _nameController = TextEditingController();
 TextEditingController _referralController = TextEditingController();
 bool receiveWhatsAppUpdates = false;
```

```
String? selectedGender;
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Complete Your Profile"),
    ),
    body: Padding(
      padding: const EdgeInsets.all(20.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text("Your Name", style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold)),
          SizedBox(height: 5),
          TextField(
            controller: _nameController,
            decoration: InputDecoration(
            hintText: "Enter your name",
            border: OutlineInputBorder(),
              prefixIcon: Icon(Icons.person, color: Colors.blue),
            ),
          ),
          SizedBox(height: 15),
          Text("Gender", style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold)),
          SizedBox(height: 5),
          Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
              _genderButton("Male", Icons.male, Colors.blue),
              _genderButton("Female", Icons.female, Colors.pink),
              _genderButton("Other", Icons.transgender, Colors.purple),
            ],
```

```
        ),
        SizedBox(height: 15),
        Row(
          children: [
            Image.asset("assets/images/wahstapp12.png", height: 25),
            SizedBox(width: 10),
            Expanded(
              child: Text(
                "Receive updates on WhatsApp",
                style: TextStyle(fontSize: 14),
              ),
            ),
            Switch(
              value: receiveWhatsAppUpdates,
              onChanged: (bool value) {
              setState(() {
                  receiveWhatsAppUpdates = value;
                });
              },
            ),
          ],
        ),
        SizedBox(height: 15),
        Text("Have a referral code?", style: TextStyle(fontSize: 16, fontWeight:
FontWeight.bold)),
        SizedBox(height: 5),
        TextField(
          controller: _referralController,
          decoration: InputDecoration(
          hintText: "Enter referral code",
          border: OutlineInputBorder(),
          ),
```

```
          ),
          SizedBox(height: 20),
          Center(
            child: ElevatedButton(
              onPressed: () {
                print("Form Submitted: Name - \${_nameController.text}, Gender - \$selectedGender, WhatsApp Updates - \$receiveWhatsAppUpdates, Referral Code - \${_referralController.text}");
              },
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.black,
                foregroundColor: Colors.white,
                padding: EdgeInsets.symmetric(vertical: 14, horizontal: 120),
                shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(8),
                ),
              ),
              child: Text("Next"),
            ),
          ),
        ],
      ),
    ),
  );
}

Widget _genderButton(String gender, IconData icon, Color color) {
  return GestureDetector(
    onTap: () {
      setState(() {
        selectedGender = gender;
      });
```

```
      },
      child: Container(
        width: 100,
        padding: EdgeInsets.symmetric(vertical: 10),
        decoration: BoxDecoration(
          color: selectedGender == gender ? color.withOpacity(0.2) : Colors.white,
          border: Border.all(color: color),
          borderRadius: BorderRadius.circular(10),
        ),
        child: Column(
          children: [
            Icon(icon, color: color),
            Text(
              gender,
              style: TextStyle(fontWeight: FontWeight.bold, color: color),
            ),
          ],
        ),
      ),
    );
  }
}
```

**Splash_screen.dart**

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

class SplashScreen extends StatefulWidget {
  @override
  _SplashScreenState createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {
```

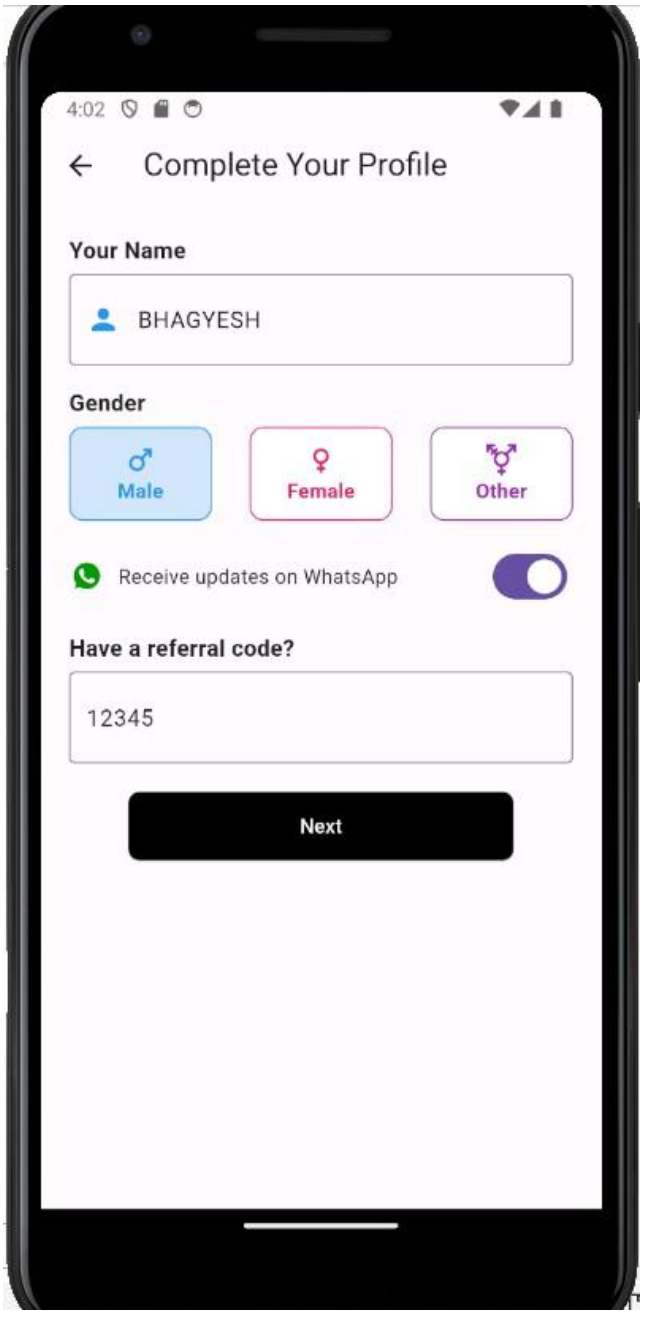
```
  @override
  void initState() {
    super.initState();
    Future.delayed(Duration(seconds: 3), () {
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => MainScreen()),
      );
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: Center(
        child: Image.asset(
          'assets/images/rapido12.jpeg',
          height: 150,
        ),
      ),
    );
  }
}
```

❖ **OUTPUT**


4:22
rapido
BIKE TAXI


4:01
rapido
BIKE TAXI
What's your number?
Enter your phone number to proceed
+91 0000000000
Next


4:02
rapido
BIKE TAXI
What's your number?
Enter your phone number to proceed
+91 1234567891
Next
1 2 3 –
4 5 6
7 8 9
, 0 .


4:02
Complete Your Profile
Your Name
BHAGYESH
Gender
Male
Female
Other
Receive updates on WhatsApp
Have a referral code?
12345
Next

# MAD & PWA Lab

# Journal

| | |
|---|---|
| Experiment No. | 05 |
| Experiment Title. | To apply navigation, routing and gestures in Flutter App |
| Roll No. | 35 |
| Name | Bhagyesh 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: | |

**Experiment no 5**

**Aim :- To apply navigation routing gestures in flutter app**

**Theory :-**

In Flutter, navigation refers to the process of moving between different screens (routes) in an application. A route is a widget that represents a screen, and Flutter provides multiple ways to manage navigation efficiently.

➢ **Navigation in Flutter**

Navigation is essential for defining the workflow of an application. Flutter manages navigation using the Navigator widget, which maintains a stack of routes.

Types of Navigation in Flutter:

1. Stack-Based Navigation (Using Navigator.push() & Navigator.pop())
2. Named Routes (For Structured Navigation)
3. Gesture-Based Navigation (Swipe, Tap, etc.)

1. Stack-Based Navigation

Flutter follows a stack-based approach where new screens are pushed onto the stack and removed when they are no longer needed.

- Navigator.push() – Adds a new screen (route) to the navigation stack.
- Navigator.pop() – Removes the current screen and returns to the previous one.

This approach is useful for small apps or simple screen transitions.

2. Named Routes

Named routes provide a more structured way to manage navigation by defining route names inside the MaterialApp widget.

Advantages of Named Routes:

Makes navigation cleaner and more manageable.
Ideal for large applications with multiple screens.
Avoids code duplication for repeated navigation logic.

Named routes help in organizing navigation paths systematically.

3. Gesture-Based Navigation

Flutter provides the GestureDetector widget, which enables interaction-based navigation using gestures such as:

- Tap Gesture – Navigate to a new screen when the user taps.
- Swipe Gesture – Navigate back when the user swipes right.
- Drag Gesture – Move between screens using drag gestures.

Gestures improve user experience by making the app more interactive and intuitive.

Code :-

```
appBar: AppBar(
      backgroundColor: Colors.white,
      title: InkWell(
       /// when click on search box open search screen
        onTap: () {
         Navigator.push(context,
            MaterialPageRoute(builder: (context) => SearchScreen()));
        },

        /// here we apply hero animation
        child: Container(
          height: 40,
          width: 280,
          decoration: BoxDecoration(
              borderRadius: BorderRadius.circular(20),
              color: Colors.black12.withOpacity(0.05)),
          child: Padding(
              padding: const EdgeInsets.symmetric(horizontal: 10.0),
              child: Row(
                children: [
                  const Icon(Icons.search_rounded),
                  const SizedBox(
                    width: 10,
                  ),
                  Hero(
                    tag: "anim",
                    child: Text(
                      "Where are you going?",
                      style: myTextStyle20(fontWeight: FontWeight.bold),
                    ),
                  ),
                ],
              ),
            )),
        ),
      ),
```

```
    backgroundColor: Colors.white,


    ///
_____BODY_____
_____///
    body: SingleChildScrollView(
      child: Column(
        crossAxisAlignment:
CrossAxisAlignment.start,
        children: [
          Padding(
            padding: const
EdgeInsets.symmetric(horizontal: 10.0),
            child: Row(
              mainAxisAlignment:
MainAxisAlignment.spaceBetween,
              children: [
                Text(
                  "Explore",
                  style:
myTextStyle18(fontWeight:
FontWeight.bold),
                ),


                /// View All Button , click on
onPress bottom Sheet
                ///
_____VIEW
ALL_____
///
                TextButton(
                    onPressed: () {
                      showModalBottomSheet(
                        showDragHandle: true,
                        context: context,
                        builder: (context) {
                          return Container(
                            height: 300,
                            width: double.infinity,
                            child:
Column(
                              crossAxisAlignment:
CrossAxisAlignment.start,
                              children: [
                                Padding(
                                  padding: const
EdgeInsets.symmetric(
                                      horizontal: 10.0),
                                  child: Text(
                                    "All Services ",
                                    style:
myTextStyle22(
                                        fontWeight:
FontWeight.bold),
                                  ),
                                ),
                                const SizedBox(
                                  height: 10,
                                ),
                                Expanded(
                                  child:
GridView.builder(
                                    itemCount:
data.length,
```

```
                    gridDelegate:
                        const
SliverGridDelegateWithFixedCrossAxisC
ount(
crossAxisCount: 4),
                    itemBuilder:
(context, index) {
                      /// here we call
my explore card
                      return InkWell(
                        onTap: () {
Navigator.pop(context);
                        },
                        child:
MyExploreCard(
                          imagePath:
data[index]['image'],
                          title:
data[index]['title'],
                        ),
                      );
                    },
                  ),
                )
              ],
            ),
          );
        },
      );
    },
                  child: Text(
                    "View All",
                    style: myTextStyle15(
                      textColor:
Colors.blueAccent,
                      fontWeight:
FontWeight.bold),
                  ))
            ],
          ),
        ),

        /// Here we show Rapido servie
        /// Create custom Card
ExploreCard
        /// Here we call
        Padding(
          padding: const
EdgeInsets.symmetric(horizontal: 10.0),
          child: Row(
            mainAxisAlignment:
MainAxisAlignment.spaceBetween,
            children: [
              MyExploreCard(
                imagePath:
"assets/image/parcel.png",
                title: "Parcel",
              ),
              MyExploreCard(
                imagePath:
"assets/image/auto.png",
```

```
            title: "Auto",
          ),
          MyExploreCard(
            imagePath:
"assets/image/cab economy.png",
            title: "Cab Economy",
          ),
          MyExploreCard(
            imagePath:
"assets/image/motorbike.png",
            title: "Bike",
          ),
        ],
      ),
    ),
    const SizedBox(
      height: 20,
    ),

    /// here we show auto image
slider for banner showing

    Padding(
      padding: const
EdgeInsets.all(8.0),
      child: CarouselSlider(
        items: bannerImage,
        options: CarouselOptions(
          height: 100,
          aspectRatio: 16 / 3,
          viewportFraction: 1,
          initialPage: 0,
          enableInfiniteScroll: true,
          reverse: true,
          autoPlay: true,
          autoPlayInterval: const
Duration(seconds: 3),
          autoPlayAnimationDuration:
              const
Duration(milliseconds: 900),
          autoPlayCurve:
Curves.fastOutSlowIn,
          enlargeCenterPage: true,
          enlargeFactor: 0.3,
          scrollDirection:
Axis.horizontal,
        )),
    ),
    const SizedBox(
      height: 20,
    ),

    /// SAVE MORE WITH RAPIDO
    Padding(
      padding: const
EdgeInsets.only(left: 10.0),
      child: Text(
        "Save More With Rapido",
        style:
myTextStyle20(textColor:
Colors.black87),
```

```
          ),
        ),

        /// here we create custom card
        /// here we call my save card

        const SizedBox(
          height: 10,
        ),
        SingleChildScrollView(
          scrollDirection: Axis.horizontal,
          child: Padding(
            padding: const
EdgeInsets.symmetric(horizontal: 10.0),
            child: Row(
              children: [
                /// Airport
                MySaveMoreCard(
                    title: "Airport",
                    imagePath:
"assets/image/AirPort-icon.png"),
                const SizedBox(
                  width: 15,
                ),

                /// Railway Station
                MySaveMoreCard(
                    title: "Railway station",
                    imagePath:
"assets/image/railway.png"),
                const SizedBox(
                  width: 15,
                ),

                /// Bus stand
                MySaveMoreCard(
                    title: "Bus Stand",
                    imagePath:
"assets/image/bus-removebg-
preview.png"),
              ],
            ),
          ),
        ),

        /// here we show single poster
        Padding(
          padding: const
EdgeInsets.all(10.0),
          child: Container(
            height: 150,
            decoration: BoxDecoration(
                borderRadius:
BorderRadius.circular(15),
                image: const
DecorationImage(
                    image: AssetImage(
                      "assets/image/rapiod
poster 2.png",
                    ),
                    fit: BoxFit.cover)),
```

```
          ),
        ),
        const SizedBox(
          height: 20,
        ),

        /// last background poster show
        Stack(children: [
          Image.asset(
            "assets/image/childrens-pattern-600nw-367289369-removebg-preview.png",
            color: Colors.black38,
          ),
          Positioned(
            left: 0,
            top: 200,
            child: Container(
              color: Colors.white70.withOpacity(0.9),
              child: Padding(
                padding: const EdgeInsets.symmetric(horizontal: 10.0),
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.start,
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text(
                      "#goRapido",
                      style: myTextStyle40(
                        fontWeight: FontWeight.bold,
                        textColor: Colors.black54),
                    ),
                    Row(
                      mainAxisAlignment: MainAxisAlignment.start,
                      children: [
                        Image.asset(
                          "assets/icon/india.png",
                          height: 20,
                          width: 20,
                        ),
                        const SizedBox(
                          width: 5,
                        ),
                        Text(
                          "Made For India",
                          style: myTextStyle18(),
                        )
                      ],
                    ),
                    Row(
                      mainAxisAlignment: MainAxisAlignment.start,
                      children: [
                        Image.asset(
```

```
                    "assets/icon/love
(1).png",
                    height: 20,
                    width: 20,
                  ),
                  const SizedBox(
                    width: 5,
                  ),
                  Text(
                    "Crafted in
Bengaluru",
                    style:
myTextStyle18(),
                  )
                ],
              ),
            ],
          ),
        ),
      ))
    ])
  ],
 ),
),

/// _____DRAWER
IMPLEMENT
HERE_____///
drawer: Drawer(
 width: double.infinity,
 backgroundColor: Colors.white,
 child: ListView(
  children: [

///_____HEADER
PART_____///
   DrawerHeader(
    child: Column(
    children: [
    Row(
      children: [
       IconButton(
         onPressed: () {
          Navigator.pop(context);
         },
         icon: const
Icon(Icons.arrow_back_rounded)),
        Text(
         "Menu",
         style:
myTextStyle18(fontWeight:
FontWeight.bold),
        )
      ],
     ),

     /// back icon and manu text
     Container(
      height: 80,
      decoration: BoxDecoration(
```

```
                boxShadow: const [
                  BoxShadow(
                    color: Colors.black,
                    blurRadius: 1,
                    spreadRadius: 1),
                ],
                borderRadius:
BorderRadius.circular(10),
                color: Colors.white,
                border: Border.all(width: 1,
color: Colors.black26)),
              child: Column(
               children: [
               ListTile(
                  leading: Image.asset(
                    "assets/icon/user
(2).png",
                    height: 40,
                  ),
                  title: Text(
                    "App Creator",
                    style:
myTextStyle15(fontWeight:
FontWeight.bold),
                  ),
                  subtitle: Text(
                    "1234567890",
                    style: myTextStyle15(),
                  ),
                  trailing: const
Icon(Icons.navigate_next_rounded),
                )
              ],
            )),

            /// Drawer Body

            ListTile(
              title: Text(
                "Help",
                style:
myTextStyle18(fontWeight:
FontWeight.bold),
              ),
              leading: Image.asset(
                "assets/icon/handshake.png",
                height: 30,
              ),
              trailing: const
Icon(Icons.navigate_next_outlined),
            ),
            const Divider(),

            ListTile(
              title: Text(
                "Parcel - Send Items",
```

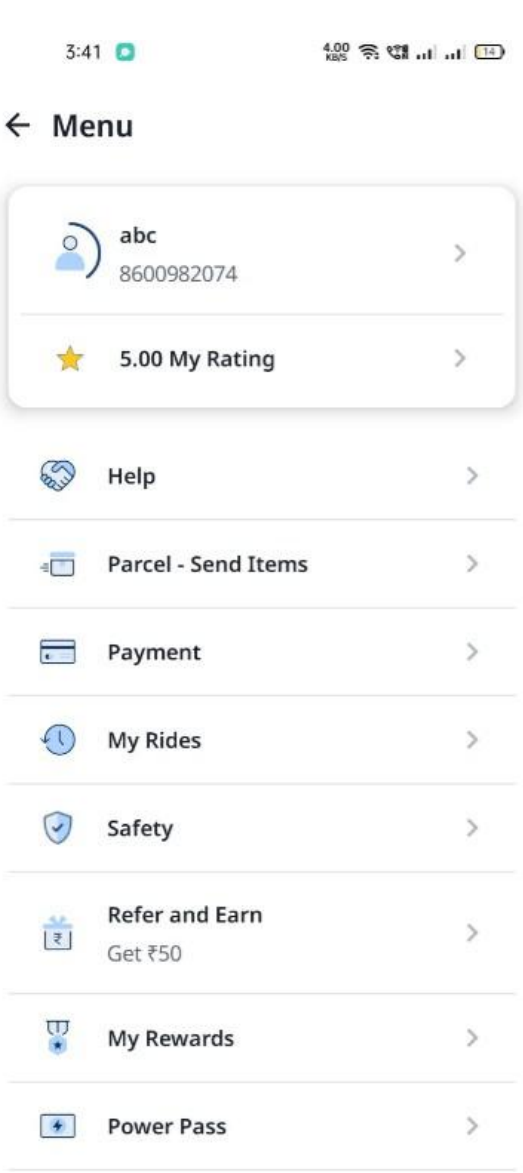
```
            style:
myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/trolley (1).png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),

        ListTile(
          title: Text(
            "Payment",
            style:
myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/credit-card.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),

        ListTile(
          title: Text(
            "My Rides",
            style:
myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/refresh.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),
        ListTile(
          title: Text(
            "Safety",
            style:
myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/verified.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),

        ListTile(
          title: Text(
```

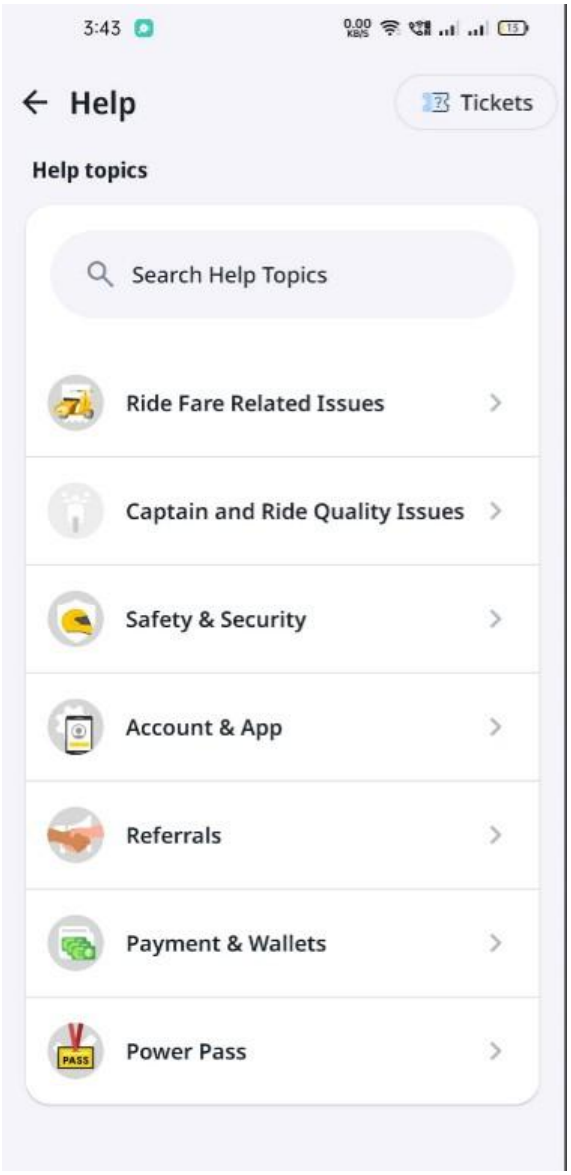
```
            "Refer and Earn",
            style:
myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          subtitle: Text(
            "Get 50",
            style: myTextStyle15(),
          ),
          leading: Image.asset(
            "assets/icon/cost.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),

        ListTile(
          title: Text(
            "My Reward",
            style:
myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/medal.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),

        ListTile(
          title: Text(
            "Power Pass",
            style:
myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/id-card.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),
        ListTile(
          title: Text(
            "Rapido Coin",
            style:
myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/dollar.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
```

```
        ),
        const Divider(),
        ListTile(
          title: Text(
            "Notifications",
            style:
myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/notification.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),
        ListTile(
          title: Text(
            "Claims",
            style:
myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/security.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),

        ListTile(
          title: Text(
            "Setting",
            style:
myTextStyle18(fontWeight:
FontWeight.bold),
          ),
          leading: Image.asset(
            "assets/icon/setting.png",
            height: 30,
          ),
          trailing: const
Icon(Icons.navigate_next_outlined),
        ),
        const Divider(),

        Padding(
          padding: const
EdgeInsets.all(8.0),
          child: Container(
            height: 100,
            decoration: BoxDecoration(
              color: const
Color(0x66ffcc75),
              borderRadius:
BorderRadius.circular(16)),
            child: Row(
              mainAxisAlignment:
MainAxisAlignment.spaceBetween,
```

```
            children: [
              Padding(
                padding: const EdgeInsets.only(left: 8.0),
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.center,
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text(
                      "Earn money with Rapido",
                      style: myTextStyle15(fontWeight: FontWeight.bold),
                    ),
                    Text(
                      "Become a Captain!",
                      style: myTextStyle15(),
                    ),
                  ],
                ),
              ),
              Container(
                width: 200,
                decoration: const BoxDecoration(
                  borderRadius: BorderRadius.only(
                    topRight: Radius.circular(10),
                    bottomRight: Radius.circular(10)),
                  image: DecorationImage(
                    image: AssetImage("assets/image/Rapido_boy.png"),
                    fit: BoxFit.cover)),
              )
            ],
```

**Output**


3:41
Menu
abc
8600982074
5.00 My Rating
Help
Parcel - Send Items
Payment
My Rides
Safety
Refer and Earn
Get ₹50
My Rewards
Power Pass


3:43
Help
Tickets
Help topics
Search Help Topics
Ride Fare Related Issues
Captain and Ride Quality Issues
Safety & Security
Account & App
Referrals
Payment & Wallets
Power Pass

# MAD & PWA Lab

# Journal

| | |
|---|---|
| Experiment No. | 06 |
| Experiment Title. | To Connect Flutter UI with fireBase database |
| Roll No. | 35 |
| Name | Bhagyesh 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:- 6**

**Name:-Bhagyesh Patil D15A Roll-no:-35**

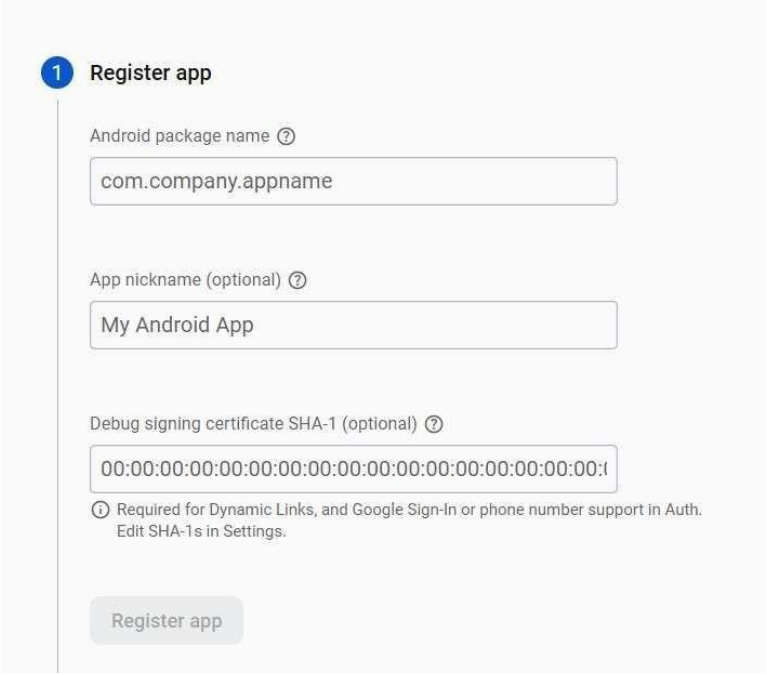
Aim:- To Connect flutter UI with firebase database

---

**Creating a New Firebase Project**

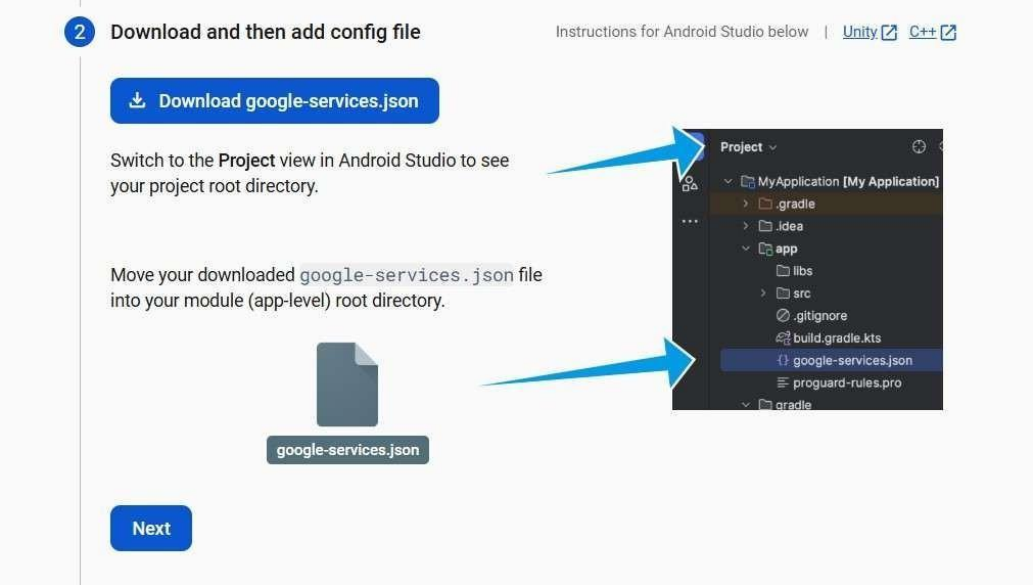

First, log in with your Google account to manage your Firebase projects. From within the Firebase dashboard, select the Create new project button and give it a name

In order to add Android support to our Flutter application, select the Android logo from the dashboard. This brings us to the following screen:



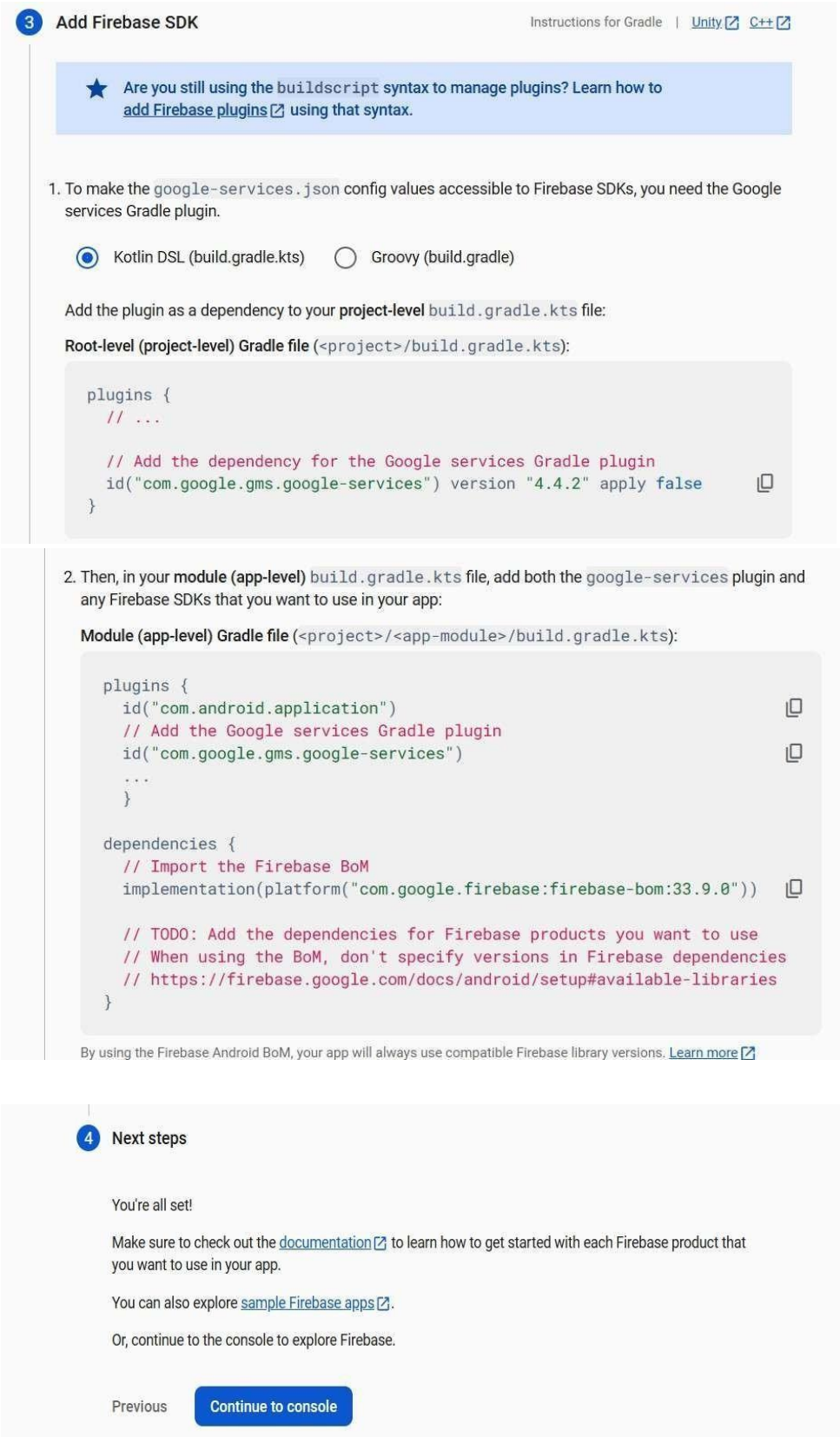

The most important thing here is to match up the Android package name that you choose here with the one inside of our application.

Then download the google-services.json file, that you will get.




put that file in the android folder (root level)

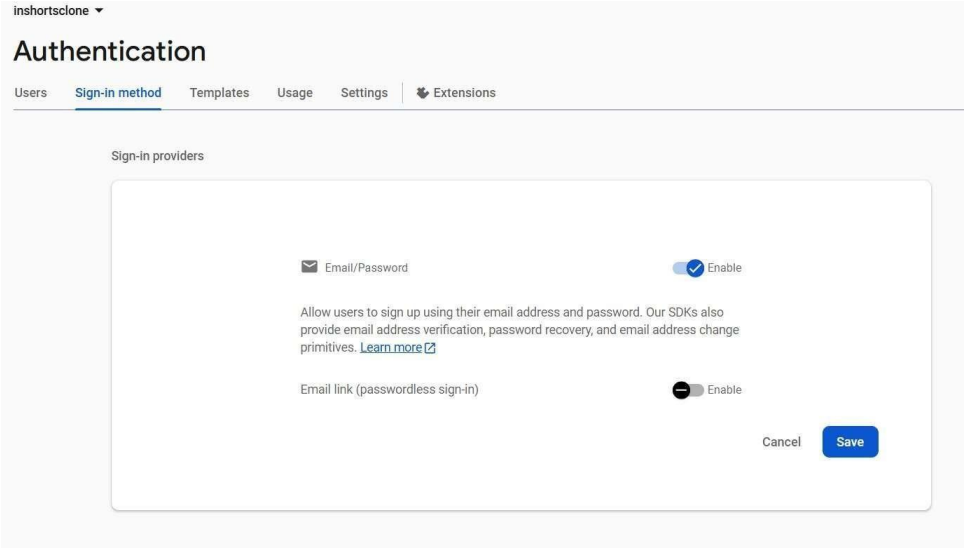

then select the build.gradle.kts (Kotlin DSL) part, and then follow the rest instructions




Generate the firebase_options.dart file, based on the google-services.json file

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

class DefaultFirebaseOptions {
  static FirebaseOptions get currentPlatform {
    return const FirebaseOptions(
      apiKey: "AIzaSyA_VIR7fj4d-b6tNZhW9qJ6GRRx5EXKqs0",
      appId: "1:388272292768:android:33180b2382688b18781ac5",
      messagingSenderId: "388272292768",
      projectId: "inshortsclone-848a9",
      storageBucket: "inshortsclone-848a9.firebasestorage.app",
      androidClientId: "1:388272292768:android:33180b2382688b18781ac5",
    );
  }
}
```

In your part select the sign-in method and enable it.



**Code:-**

**Authenction_logic**

```
import
'package:firebase_auth/firebase_auth.dart'
;
import
'package:google_sign_in/google_sign_in.d
art';


class AuthService {
  final FirebaseAuth _auth =
FirebaseAuth.instance;


  // Sign in with Email & Password
  Future<User?> signInWithEmail(String
email, String password) async {
    try {
      UserCredential userCredential = await
_auth.signInWithEmailAndPassword(
        email: email,
        password: password,
      );
      return userCredential.user;
    } catch (e) {
      print("Error: $e");
      return null;
    }
  }


  // Register with Email & Password
  Future<User?> registerWithEmail(String
email, String password) async {
    try {
      UserCredential userCredential = await
_auth.createUserWithEmailAndPassword(
        email: email,
        password: password,
      );
      return userCredential.user;
    } catch (e) {
      print("Error: $e");
      return null;
    }
  }


  // Google Sign-In
  Future<User?> signInWithGoogle()
async {
    try {
      final GoogleSignInAccount?
googleUser = await
GoogleSignIn().signIn();
      if (googleUser == null) return null;


      final GoogleSignInAuthentication
googleAuth = await
googleUser.authentication;
      final AuthCredential credential =
GoogleAuthProvider.credential(
        accessToken:
googleAuth.accessToken,
        idToken: googleAuth.idToken,
      );


      UserCredential userCredential = await
_auth.signInWithCredential(credential);
      return userCredential.user;
    } catch (e) {
      print("Google Sign-In Error: $e");
      return null;
    }
  }


  // Sign Out
  Future<void> signOut() async {
    await _auth.signOut();
    await GoogleSignIn().signOut();
  }


  // Get current user
```

```
User? getCurrentUser() {                    }
  return _auth.currentUser;                 }
```

**Login screen**

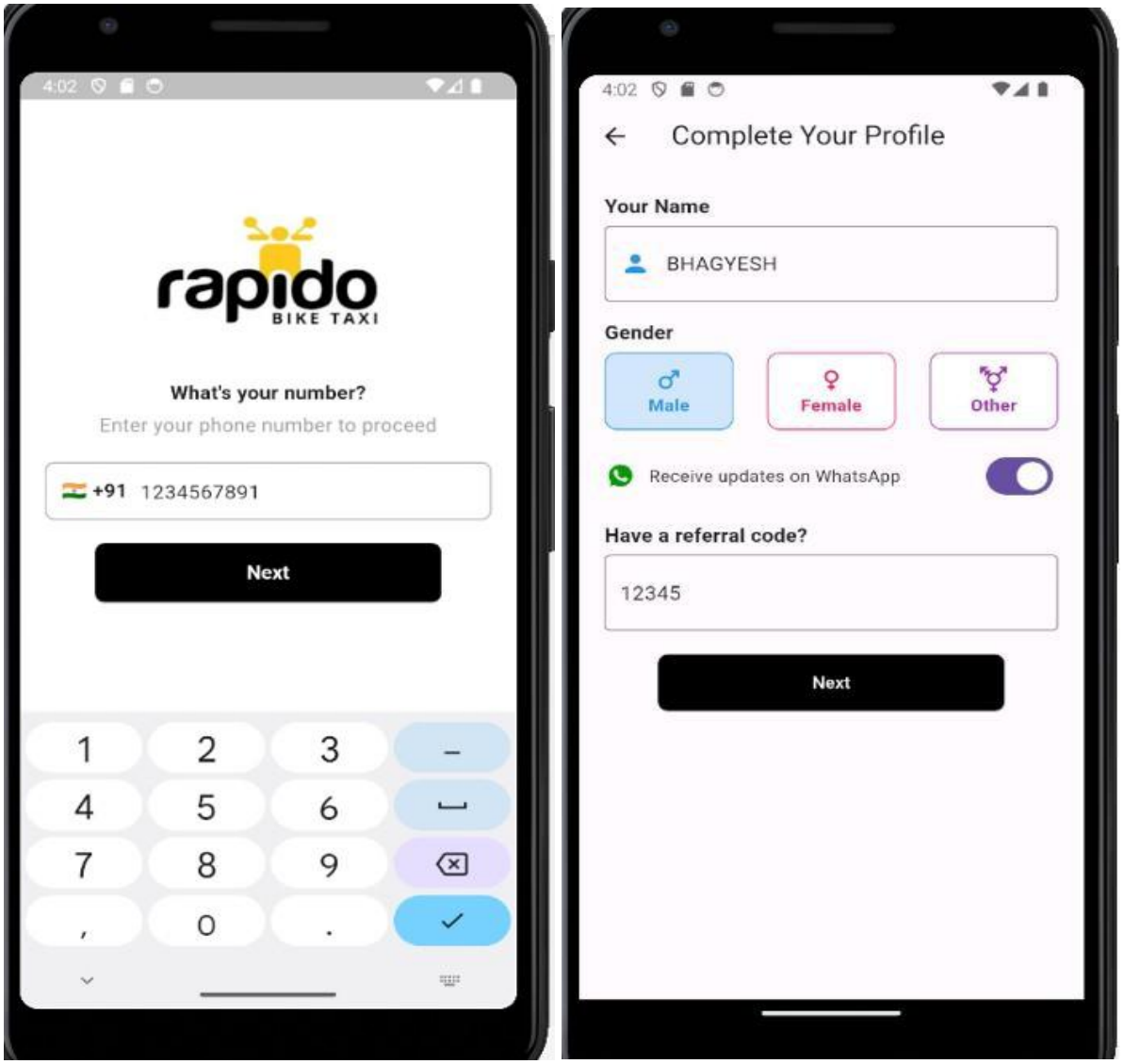
```
import 'package:flutter/material.dart';
import '../services/auth_service.dart';
import 'home_screen.dart';

class LoginScreen extends StatefulWidget {
  @override
  _LoginScreenState createState() =>
_LoginScreenState();
}

class _LoginScreenState extends
State<LoginScreen> {
  final TextEditingController emailController =
TextEditingController();
  final TextEditingController passwordController
= TextEditingController();
  final AuthService _authService =
AuthService();

  void _login() async {
    String email = emailController.text.trim();
    String password =
passwordController.text.trim();
    var user = await
_authService.signInWithEmail(email,
password);
    if (user != null) {
      Navigator.pushReplacement(context,
MaterialPageRoute(builder: (context) =>
HomeScreen()));
    } else {

ScaffoldMessenger.of(context).showSnackBar(
SnackBar(content: Text("Login failed!")));
    }
  }

  void _googleSignIn() async {
    var user = await
_authService.signInWithGoogle();
    if (user != null) {
      Navigator.pushReplacement(context,
MaterialPageRoute(builder: (context) =>
HomeScreen()));
    } else {

ScaffoldMessenger.of(context).showSnackBar(
SnackBar(content: Text("Google Sign-In
failed!")));
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Rapido Login')),
      body: Padding(
        padding: EdgeInsets.all(16.0),
        child: Column(


        mainAxisAlignment:
MainAxisAlignment.center,
        children: [
          TextField(controller: emailController,
decoration: InputDecoration(labelText:
'Email')),
          TextField(controller:
passwordController, decoration:
InputDecoration(labelText: 'Password'),
obscureText: true),
          SizedBox(height: 20),
          ElevatedButton(onPressed: _login,
child: Text('Login')),
          SizedBox(height: 10),
          ElevatedButton(onPressed:
_googleSignIn, child: Text('Sign in with
Google')),
        ],
      ),
    ),
  );
  }
}
```

**Home_Screen**

```
import 'package:flutter/material.dart';
import '../services/auth_service.dart';
import 'login_screen.dart';

class HomeScreen extends StatelessWidget {
final AuthService _authService =
AuthService();

  void _logout(BuildContext context) async {
    await _authService.signOut();
    Navigator.pushReplacement(context,
MaterialPageRoute(builder: (context) =>
LoginScreen()));
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Rapido
```

```
Home')),
      body: Center(
        child: Column(
          mainAxisAlignment:
MainAxisAlignment.center,
          children: [
            Text('Welcome to Rapido!'),
            SizedBox(height: 20),
            ElevatedButton(onPressed: () =>
_logout(context), child: Text('Logout')),
          ],
        ),
```

**OUTPUT** :


4:02
rapido
BIKE TAXI
What's your number?
Enter your phone number to proceed
+91 1234567891
Next
1 2 3
4 5 6
7 8 9
, 0 .
4:02
Complete Your Profile
Your Name
BHAGYESH
Gender
Male
Female
Other
Receive updates on WhatsApp
Have a referral code?
12345
Next

# 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. | 35 |
| Name | Bhagyesh Patil |
| Class | D15A/D15B |
| Subject | MAD & PWA Lab |
| Lab Outcome | LO4: Understand various PWA frameworks and their requirements |
| Grade: | |

# PWA Experiment -7

Bhagyesh Patil 35/D15A

❖ **AIM**

To write meta data of your Ecommerce PWA in a Web app manifest file to enable “add to homescreen feature”.

❖ **Theory**:-

## Regular Web Application

A regular web application is a website designed to be accessible on various devices, ensuring that content adjusts dynamically to different screen sizes. Built using web technologies such as HTML, CSS, JavaScript, and Ruby, these applications function through a browser. While they can leverage some native device features, their functionality is dependent on browser compatibility. For instance, a feature may work on Google Chrome but not on Safari or Mozilla Firefox due to browser limitations.

## Progressive Web Application (PWA)

A Progressive Web Application (PWA) is an advanced version of a regular web app, integrating additional features to enhance the user experience. PWAs combine the best elements of desktop and mobile applications, delivering a seamless experience across platforms.

## ❖ Key Differences Between PWAs and Regular Web Apps

### 1. Native-Like Experience

While both PWAs and regular web apps utilize standard web technologies like HTML, CSS, and JavaScript, PWAs offer a user experience similar to native mobile applications. PWAs can access device-specific features such as push notifications without relying on a particular browser, creating a smoother, more integrated experience.

## 2. Ease of Access

Unlike traditional mobile apps that require time-consuming downloads and storage space, PWAs can be installed directly via a URL. Users can add a PWA to their home screen with a simple link, eliminating installation complexities and ensuring easy access while keeping brand presence strong.

## 3. Enhanced Performance

PWAs utilize caching mechanisms to pre-load content, such as text, stylesheets, and images, allowing for faster loading times. This significantly improves user engagement and retention by reducing waiting periods, ultimately benefiting businesses by increasing interaction rates.

## 4. Improved User Engagement

PWAs efficiently leverage push notifications and native device features to keep users engaged. Unlike regular web apps, their functionality is not restricted by browser dependencies, enabling businesses to notify users about updates, offers, and promotions without disruptions.

## 5. Real-Time Updates

A major advantage of PWAs is their ability to update automatically without requiring users to download new versions from an app store. Developers can push updates directly from the server, ensuring that users always access the latest features and improvements instantly.

## 6. Search Engine Optimization (SEO) Benefits

Since PWAs function within web browsers, they can be indexed by search engines, improving their visibility in search results. This gives them a strategic advantage over native apps, which are limited to app store searches.

## 7. Cost-Effective Development

Unlike native mobile apps, PWAs do not require approval or submission to app stores, reducing development and maintenance costs.

## ❖ Advantages and Limitations of PWAs

### ➢ Advantages:

- **Progressive:** Compatible with all browsers and devices, following the principle of progressive enhancement.
- **Responsive:** Adapts to different screen sizes, including desktops, tablets, and smartphones.
- **App-Like Feel:** Mimics the experience of native applications in navigation and interaction.
- **Always Updated:** Service workers ensure real-time updates without requiring user intervention.
- **Secure:** Delivered over HTTPS, ensuring secure data transfer and protection against cyber threats.
- **SEO-Friendly:** Can be indexed by search engines, enhancing discoverability.
- **Re-Engagement:** Enables push notifications to encourage continued user interaction.
- **Installable:** Allows users to add the app to their home screen without app store downloads.
- **Offline Functionality:** Can function in low or no connectivity conditions using cached content.

### ➢ Limitations:

- **Higher Battery Consumption:** PWAs tend to consume more battery due to constant background processes.
- **Limited Hardware Access:** Some device features, such as advanced sensors and Bluetooth, may not be fully accessible.
- **Offline Mode Constraints:** Some offline capabilities remain limited, depending on browser support.
- **No App Store Presence:** PWAs cannot generate traffic from app store searches.
- **Lack of Centralized Control:** Unlike native apps, PWAs do not undergo an official approval process, potentially affecting credibility.

## CODE:

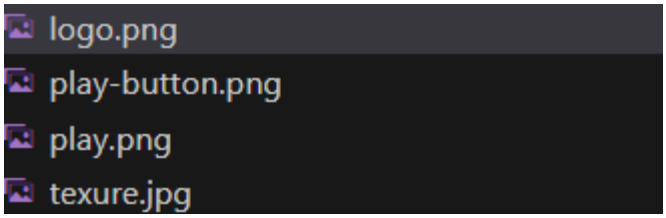
### Index.html :

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta name="description" content="Web site created using create-react-app" />

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

    <link  rel="apple-touch-icon"  href="%PUBLIC_URL%/apple-touch-icon.png">
    <link rel="apple-touch-icon" sizes="192x192" href="%PUBLIC_URL%/logo.png">
    <link rel="apple-touch-icon" sizes="512x512" href="%PUBLIC_URL%/logo.png">

    <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico" type="image/x-icon"
/>

    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.15.4/css/all.css"
        integrity="sha384-
DyZ88mC6Up2uqS4h/KRgHuoeGwBcD4Ng9SiP4dIRy0EXTlnuz47vAwmeGwVChigm
"
        crossorigin="anonymous" />

    <title>Streamo - Netflix</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
  </body>
</html>
```

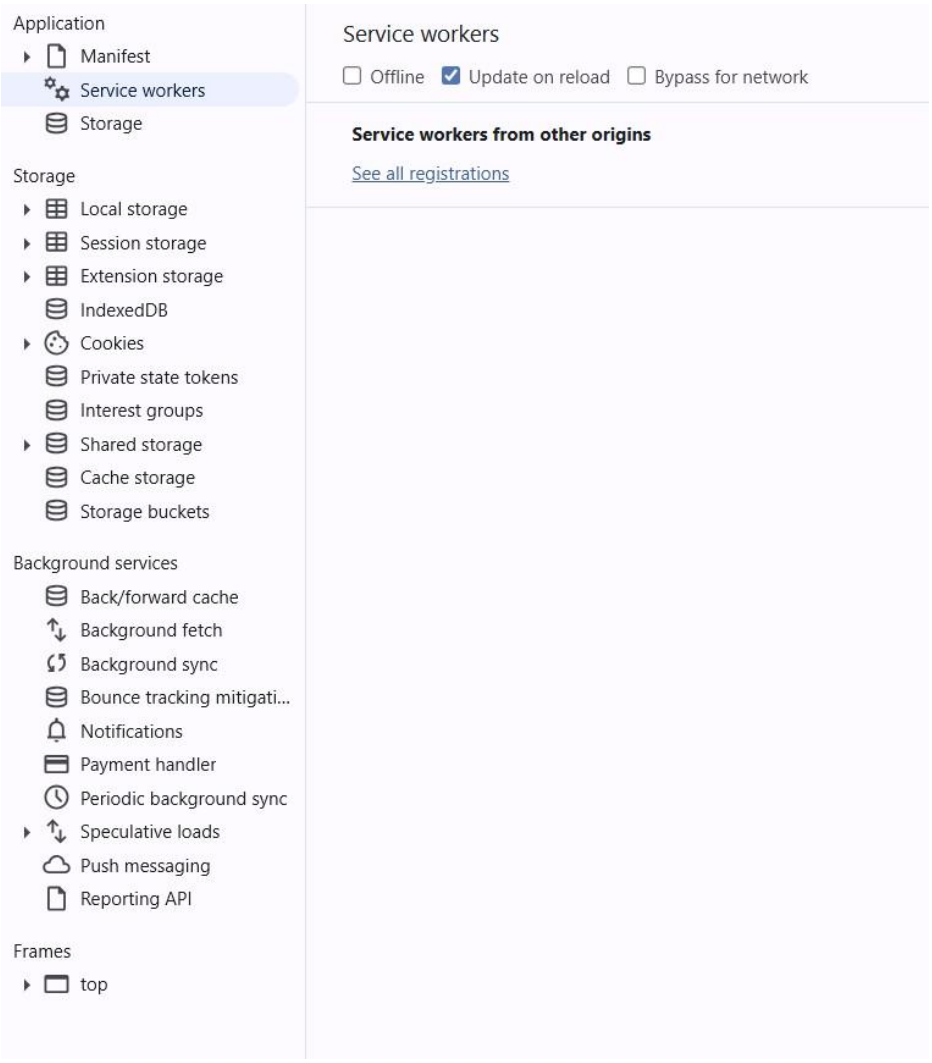
**Manifest.json**

```
{
  "name": "Streamo - Netflix",
  "short_name": "Streamo",
  "start_url": "/",
  "display": "standalone",
  "background_color": "#000000",
  "theme_color": "#000000",
  "description": "Watch unlimited movies & TV shows.",
  "icons": [
    {
      "src": "/logo.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "/logo.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ]
}
```

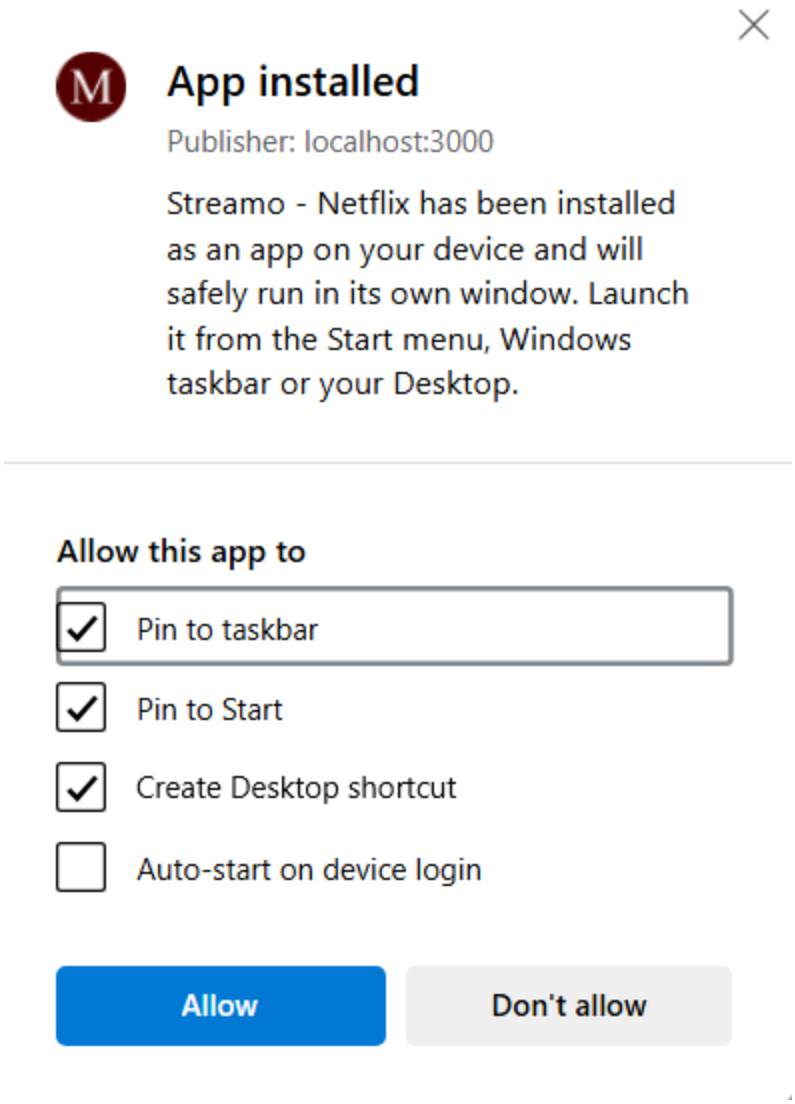
## Icons

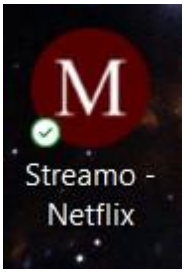


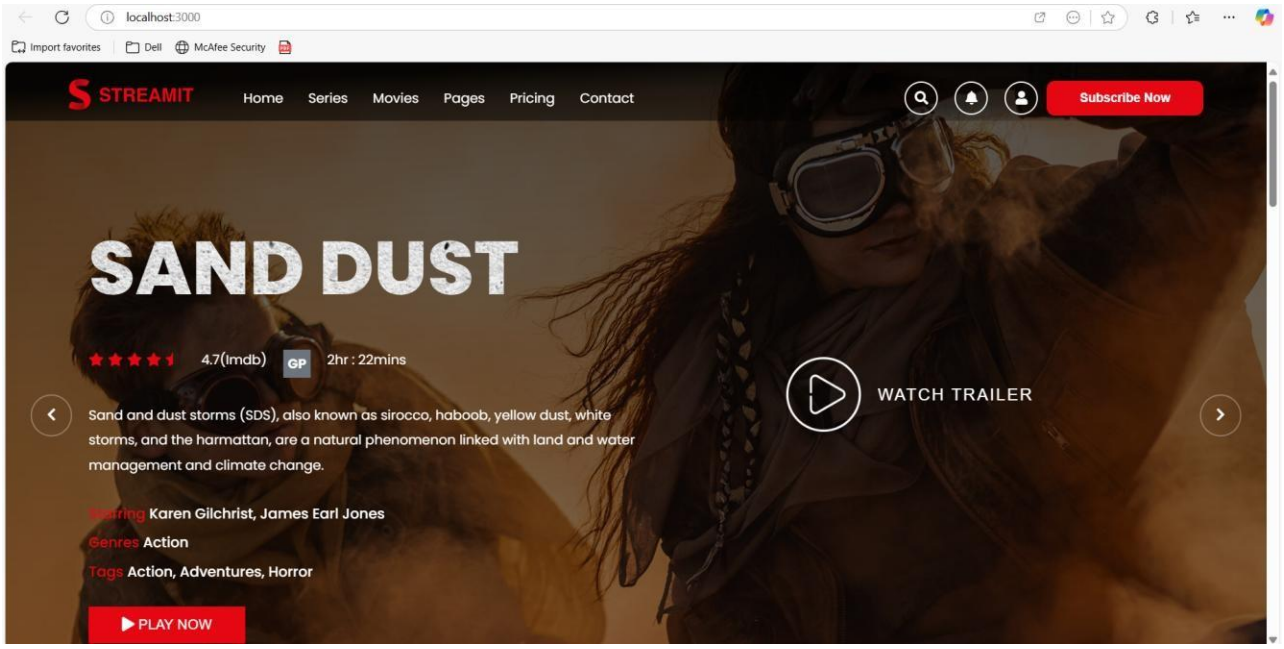

## Google Dev

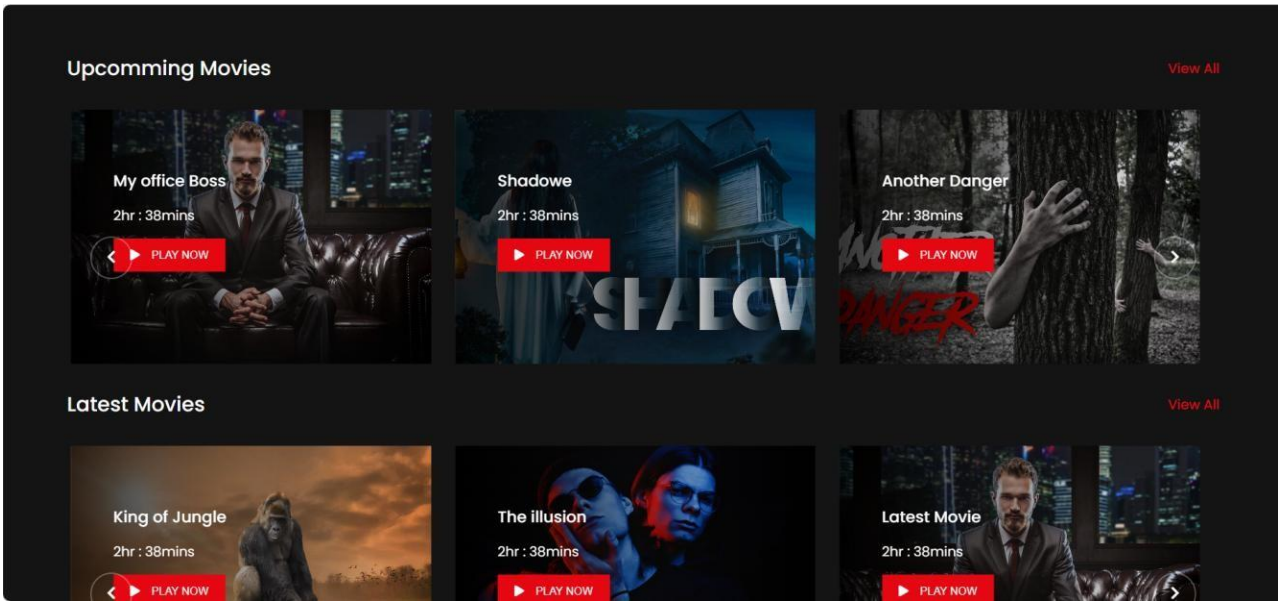


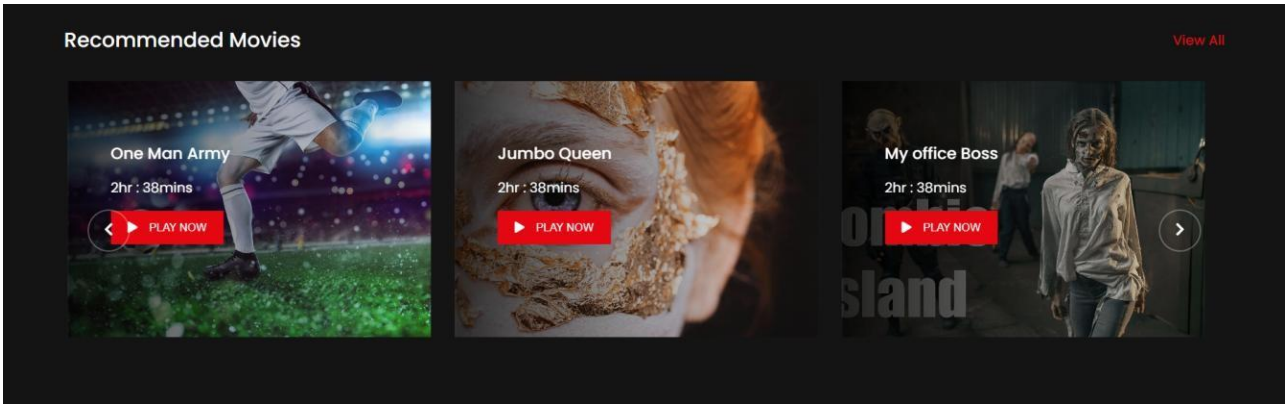
❖ **Output:-**


App installed
Publisher: localhost:3000
Streamo - Netflix has been installed as an app on your device and will safely run in its own window. Launch it from the Start menu, Windows taskbar or your Desktop.
Allow this app to
Pin to taskbar
Pin to Start
Create Desktop shortcut
Auto-start on device login
Allow
Don't allow


Streamo - Netflix


localhost:3000
Import favorites
Dell
McAfee Security
STREAMIT
Home
Series
Movies
Pages
Pricing
Contact
Subscribe Now
SAND DUST
4.7(Imdb)
GP
2hr : 22mins
Sand and dust storms (SDS), also known as sirocco, haboob, yellow dust, white storms, and the harmattan, are a natural phenomenon linked with land and water management and climate change.
Karen Gilchrist, James Earl Jones
Action
Action, Adventures, Horror
PLAY NOW
WATCH TRAILER


Upcomming Movies
View All
My office Boss
2hr : 38mins
PLAY NOW
Shadowe
2hr : 38mins
PLAY NOW
Another Danger
2hr : 38mins
PLAY NOW
Latest Movies
View All
King of Jungle
2hr : 38mins
PLAY NOW
The illusion
2hr : 38mins
PLAY NOW
Latest Movie
2hr : 38mins
PLAY NOW


Recommended Movies
View All
One Man Army
2hr : 38mins
PLAY NOW
Jumbo Queen
2hr : 38mins
PLAY NOW
My office Boss
2hr : 38mins
PLAY NOW

# 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. | 35 |
| Name | Bhagyesh Patil |
| Class | D15A/D15B |
| Subject | MAD & PWA Lab |
| Lab Outcome | LO5: Design and Develop a responsive User Interface by applying PWA Design techniques |
| Grade: | |

# MPL Experiment 8 (PWA)

**Name:** Bhagyesh patil **Class:** D15A **Roll no:** 35
**Aim:** To code and register a service worker, and complete the install and activation process for a new service worker for the 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.

**What can we do with Service Workers?**

- 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.

**Codes:**

```
//Serviceworker.js

// sw.js - Complete Service Worker for E-commerce PWA
const CACHE_NAME = 'ecommerce-pwa-v2';
const API_CACHE = 'ecommerce-api-v1';
const ASSETS_TO_CACHE = [
  '/',
  '/index.html',
  '/manifest.json',
  '/offline.html',
  '/css/main.min.css',
  '/js/app.min.js',
  '/icons/icon-192x192.png'
  ,
  '/icons/icon-512x512.png'
  ,
  '/images/placeholder-product.jpg'
];

// ======================
// Install Event
// ======================
self.addEventListener('install', (event) => {
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then((cache) => {
        console.log('[Service Worker] Cache opened');
        return cache.addAll(ASSETS_TO_CACHE);
      })
```

```
      .then(() => self.skipWaiting())
  );
});

// ======================
// Activate Event
// ======================
self.addEventListener('activate', (event) => {
  event.waitUntil(
    caches.keys().then((cacheNames) => {
      return Promise.all(
        cacheNames.map((cacheName) => {
          if (cacheName !== CACHE_NAME && cacheName !== API_CACHE) {
            console.log('[Service Worker] Deleting old cache:',
cacheName);                                return caches.delete(cacheName);

          }
        })
      );
    })

    .then(() => self.clients.claim())
  );
});

// ======================
// Fetch Event
// ======================
self.addEventListener('fetch', (event) =>
  { const { request } = event;
  const url = new URL(request.url);

  // 1. Skip non-GET requests and chrome-extension
  if (request.method !== 'GET' || url.protocol ===
'chrome-extension:') {
    return;
  }

  // 2. API Requests (Network First with Cache Fallback)
```

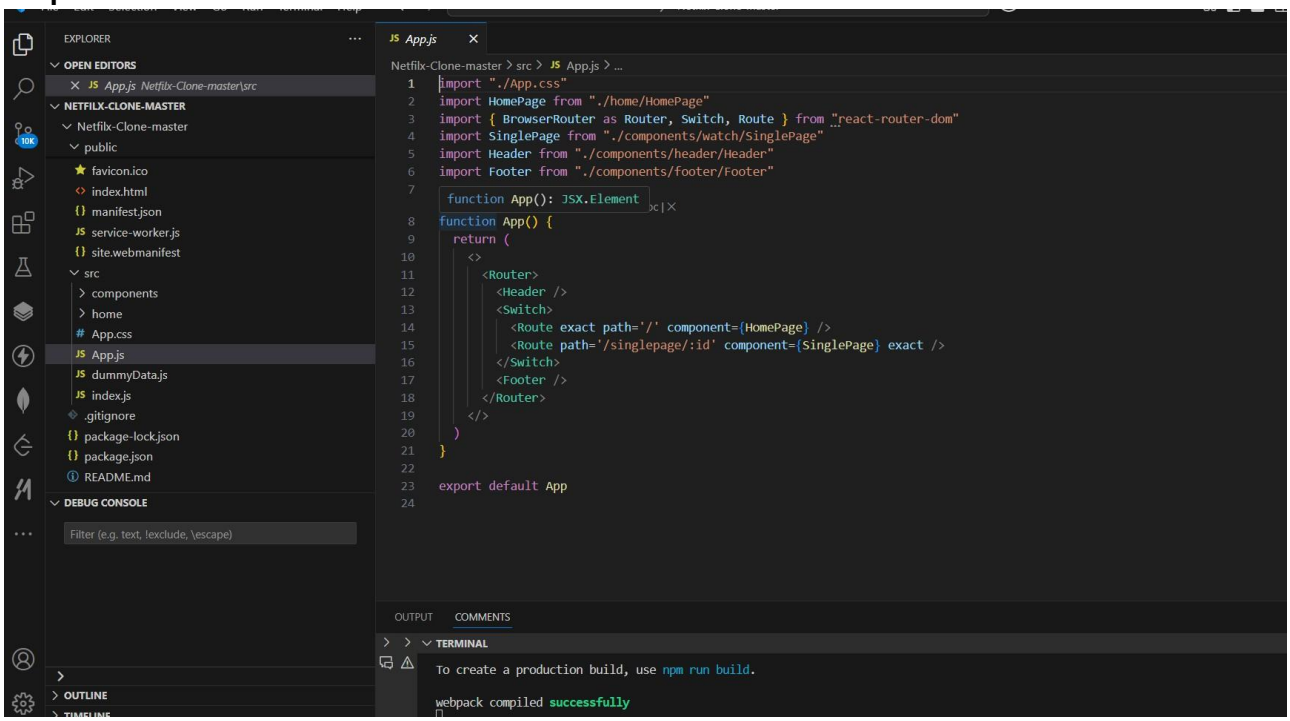
```
if (url.pathname.startsWith('/api/')) {
  event.respondWith(
    fetch(request)
      .then(networkResponse => {
        // Cache successful API
        responses if
        (networkResponse.ok) {
          const clone = networkResponse.clone();
          caches.open(API_CACHE)
            .then(cache => cache.put(request, clone));
        }
        return networkResponse;
      })
      .catch(() => {
        // Return cached version if available
        return caches.match(request)
          .then(cachedResponse => cachedResponse || Response.json(
            { error: 'Network error' },
            { status: 503 }
          ));
      })
  );
  return;
}

// 3. Static Assets (Cache First with Network Fallback)
event.respondWith(
  caches.match(request)
    .then(cachedResponse => {
      // Return cached version if
      found if (cachedResponse) {
        return cachedResponse;
      }

      // Otherwise fetch from network
      return fetch(request)
        .then(networkResponse => {
          // Cache successful
          responses if
          (networkResponse.ok) {
```

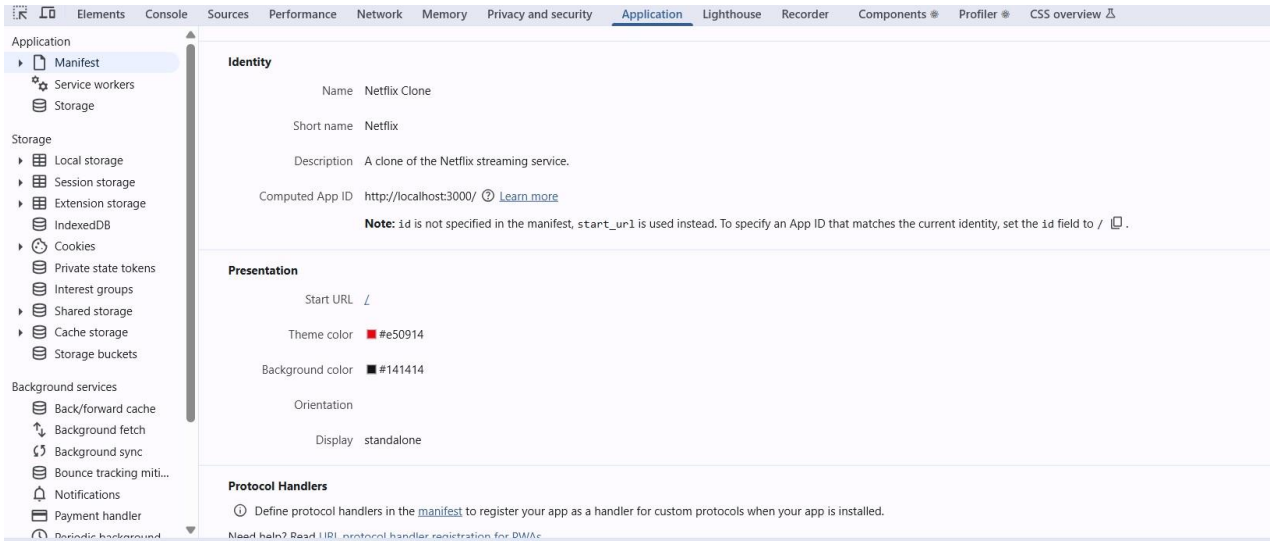
```
            const clone = networkResponse.clone();
            caches.open(CACHE_NAME)
              .then(cache => cache.put(request, clone));
          }
          return networkResponse;
        })
        .catch(() => {
          // Special handling for HTML pages
          if (request.headers.get('accept').includes('text/html')) {
            return caches.match('/offline.html');
          }
          // Return placeholder for images
          if (request.headers.get('accept').includes('image')) {
            return caches.match('/images/placeholder-product.jpg');
                                    }
                                    }
                                    )
                                    ;
  );                                })
});

// ======================
// Background Sync
// ======================
self.addEventListener('sync', (event) => {
  if (event.tag === 'sync-cart') {
    event.waitUntil(
      // Get cart data from IndexedDB
      getCartData()
        .then(cartItems => {
          return fetch('/api/cart/sync', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify(cartItems)
          });
        })
        .then(() => {
          return showNotification('Cart Synced', 'Your cart has been
updated');
```

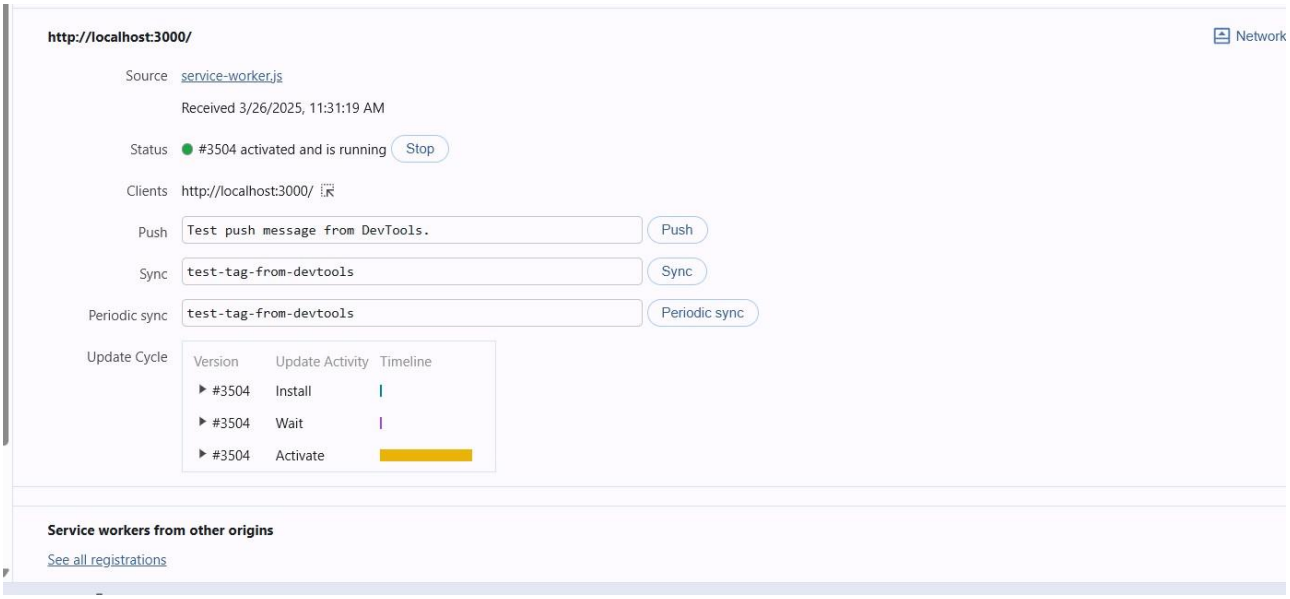
```
                                        })
                                        .catch(err => {
                                          console.error('Sync failed:',
                                          err);
                                        })
    );
                                    }
                                });


// ======================
// Push Notifications
// ======================
self.addEventListener('push', (event) => {
  let data = {};
  try {
    data = event.data.json();
  } catch (e) {
    data = {
      title: 'New Update',
      body: 'Check out our latest products!',
      icon: '/icons/icon-192x192.png',
      url: '/'
    };
  }

  const options = {
    body: data.body,
    icon: data.icon || '/icons/icon-192x192.png',
    badge: '/icons/icon-96x96.png',
    data: {
      url: data.url || '/'
    }
  };

  event.waitUntil(
    self.registration.showNotification(data.title, options)
  );
});
```

```
self.addEventListener('notificationclick', (event) => {
  event.notification.close();
  event.waitUntil(
    clients.matchAll({ type: 'window' })
      .then(clientList => {
        for (const client of clientList) {
          if (client.url === event.notification.data.url && 'focus' in
                          client) {   return client.focus();
                                    }


                                  }

        if (clients.openWindow) {
          return clients.openWindow(event.notification.data.url);
        }
      })
  );
});

// ======================
// Helper Functions
// ======================
async function getCartData() {
  // In a real app, you would use
  IndexedDB return new Promise(resolve =>
  {
    resolve([]);
  });
}

async function showNotification(title, body) {
  return self.registration.showNotification(title, { body });
}
```

## Output:


EXPLORER
OPEN EDITORS
App.js Netflix-Clone-master\src
NETFILX-CLONE-MASTER
Netflix-Clone-master
public
favicon.ico
index.html
manifest.json
service-worker.js
site.webmanifest
src
components
home
App.css
App.js
dummyData.js
index.js
.gitignore
package-lock.json
package.json
README.md
DEBUG CONSOLE
Filter (e.g. text, !exclude, \escape)
OUTLINE
TIMELINE
App.js
Netflix-Clone-master > src > App.js > ...
import "./App.css"
import HomePage from "./home/HomePage"
import { BrowserRouter as Router, Switch, Route } from "react-router-dom"
import SinglePage from "./components/watch/SinglePage"
import Header from "./components/header/Header"
import Footer from "./components/footer/Footer"
function App(): JSX.Element
function App() {
return (
<>
<Router>
<Header />
<Switch>
<Route exact path='/' component={HomePage} />
<Route path='/singlepage/:id' component={SinglePage} exact />
</Switch>
<Footer />
</Router>
</>
)
}
export default App
OUTPUT
COMMENTS
TERMINAL
To create a production build, use npm run build.
webpack compiled successfully


Elements
Console
Sources
Performance
Network
Memory
Privacy and security
Application
Lighthouse
Recorder
Components
Profiler
CSS overview
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
Identity
Name Netflix Clone
Short name Netflix
Description A clone of the Netflix streaming service.
Computed App ID http://localhost:3000/ 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 #e50914
Background color #141414
Orientation
Display standalone
Protocol Handlers
Define protocol handlers in the manifest to register your app as a handler for custom protocols when your app is installed.


http://localhost:3000/
Network
Source
service-worker.js
Received 3/26/2025, 11:31:19 AM
Status
#3504 activated and is running
Stop
Clients
http://localhost:3000/
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
#3504
Install
#3504
Wait
#3504
Activate
Service workers from other origins
See all registrations

# 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. | 35 |
| Name | Bhagyesh Patil |
| Class | D15A/D15B |
| Subject | MAD & PWA Lab |
| Lab Outcome | LO5: Design and Develop a responsive User Interface by applying PWA Design techniques |
| Grade: | |

# MPL Experiment 9 (PWA)

**Name:** Bhagyesh Patil **Class:** D15A **Roll no: 35**

**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.

**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.

**Code:**

```
//Serviceworker.js
// sw.js - Complete Service Worker for E-commerce PWA
const CACHE_NAME = 'ecommerce-pwa-v2';
const API_CACHE = 'ecommerce-api-v1';
const ASSETS_TO_CACHE = [
  '/',
  '/index.html',
  '/manifest.json',
  '/offline.html',
  '/css/main.min.css',
  '/js/app.min.js',
  '/icons/icon-192x192.png'
  ,
  '/icons/icon-512x512.png'
  ,
```

```
  '/images/placeholder-product.jpg'
];

// ======================
// Install Event
// ======================
self.addEventListener('install', (event) => {
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then((cache) => {
        console.log('[Service Worker] Cache opened');
        return cache.addAll(ASSETS_TO_CACHE);
      })
      .then(() => self.skipWaiting())
  );
});

// ======================
// Activate Event
// ======================
self.addEventListener('activate', (event) => {
  event.waitUntil(
    caches.keys().then((cacheNames)  =>
      { return Promise.all(
        cacheNames.map((cacheName) => {
          if (cacheName !== CACHE_NAME && cacheName !== API_CACHE) {
            console.log('[Service Worker] Deleting old cache:',
cacheName);                              return caches.delete(cacheName);

          }
        })
      );
    })

    .then(() => self.clients.claim())
  );
});

// ======================
```

```
// Fetch Event
// ======================
self.addEventListener('fetch', (event) =>
  { const { request } = event;
  const url = new URL(request.url);

  // 1. Skip non-GET requests and chrome-extension
  if (request.method !== 'GET' || url.protocol ===
'chrome-extension:') {
    return;
  }

  // 2. API Requests (Network First with Cache Fallback)
  if (url.pathname.startsWith('/api/')) {
    event.respondWith(
      fetch(request)
        .then(networkResponse => {
          // Cache successful API
          responses if
          (networkResponse.ok) {
            const clone = networkResponse.clone();
            caches.open(API_CACHE)
              .then(cache => cache.put(request, clone));
          }
          return networkResponse;
        })
        .catch(() => {
          // Return cached version if available
          return caches.match(request)
            .then(cachedResponse => cachedResponse || Response.json(
              { error: 'Network error' },
              { status: 503 }
            ));
        })
    );
    return;
  }

  // 3. Static Assets (Cache First with Network Fallback)
```

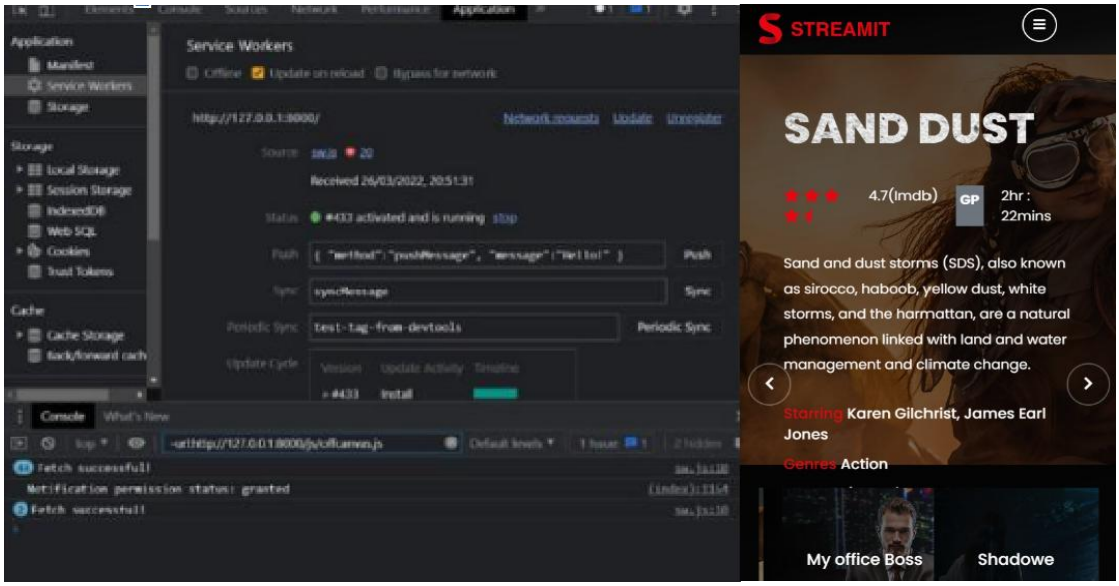
```
  event.respondWith(
    caches.match(request)
      .then(cachedResponse => {
        // Return cached version if found
        if (cachedResponse) {
          return cachedResponse;
        }

        // Otherwise fetch from network
        return fetch(request)
          .then(networkResponse => {
            // Cache successful responses
            if (networkResponse.ok) {
              const clone = networkResponse.clone();
              caches.open(CACHE_NAME)
                .then(cache => cache.put(request, clone));
            }
            return networkResponse;
          })
          .catch(() => {
            // Special handling for HTML pages
            if (request.headers.get('accept').includes('text/html')) {
              return caches.match('/offline.html');
            }
            // Return placeholder for images
            if (request.headers.get('accept').includes('image')) {
              return caches.match('/images/placeholder-product.jpg');
                                    }
                                    }
                                    )
                                    ;
  );                                })
});


// ======================
// Background Sync
// ======================
self.addEventListener('sync', (event) => {
  if (event.tag === 'sync-cart') {
```

```
    event.waitUntil(
      // Get cart data from IndexedDB
      getCartData()
        .then(cartItems => {
          return fetch('/api/cart/sync', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify(cartItems)
          });
        })
        .then(() => {
          return showNotification('Cart Synced', 'Your cart has been
updated');
        })
        .catch(err => {
          console.error('Sync failed:', err);
        })
    );
  }
 })
  ;

// ======================
// Push Notifications
// ======================
self.addEventListener('push', (event) => {
  let data = {};
  try {
    data = event.data.json();
  } catch (e) {
    data = {
      title: 'New Update',
      body: 'Check out our latest products!',
      icon: '/icons/icon-192x192.png',
      url: '/'
    };
  }

  const options = {
```

```
    body: data.body,
    icon: data.icon || '/icons/icon-192x192.png',
    badge: '/icons/icon-96x96.png',
    data: {
      url: data.url || '/'
    }
  };

  event.waitUntil(
    self.registration.showNotification(data.title, options)
  );
});

self.addEventListener('notificationclick', (event) => {
  event.notification.close();
  event.waitUntil(
    clients.matchAll({ type: 'window' })
      .then(clientList => {
        for (const client of clientList) {
          if (client.url === event.notification.data.url && 'focus' in
client) {
            return client.focus();
          }
        }
        if (clients.openWindow) {
          return clients.openWindow(event.notification.data.url);
        }
      })
  );
});

// ======================
// Helper Functions
// ======================
async function getCartData() {
  // In a real app, you would use
  IndexedDB return new Promise(resolve =>
  {
    resolve([]);
```

```
  });
}

async function showNotification(title, body) {
  return self.registration.showNotification(title, { body });
}
```

**Output:**




t

# MAD & PWA Lab

# Journal

| | |
|---|---|
| Experiment No. | 10 |
| Experiment Title. | To study and implement deployment of Ecommerce PWA to GitHub Pages. |
| Roll No. | 35 |
| Name | Bhagyesh Patil |
| Class | D15A/D15B |
| Subject | MAD & PWA Lab |
| Lab Outcome | LO5: Design and Develop a responsive User Interface by applying PWA Design techniques |
| Grade: | |

# MPL Experiment 10 (PWA)

**Name:** Bhagyesh Patil **Class:** D15A **Roll no:** 35

**Aim:** To study and implement deployment of Ecommerce PWA to GitHub Page.

**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:**

- Blogging with Jekyll
- Custom URL
- Automatic Page Generator

**Reasons for favoring this over Firebase:**

- Free to use
- Right out of github
- Quick to set up

**Companies Using GitHub Pages:**

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

**Pros**

- Very familiar interface if you are already using GitHub for your projects.
- Easy to set up. Just push your static website to the `gh-pages` branch and your website is ready.
- Supports Jekyll out of the box.
- 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**

- The code of your website will be public, unless you pay for a private repository.
- Currently, there is no support for HTTPS for custom domains. It's probably coming soon though.
- 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:**

- 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.
- 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.
- 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:**

- Realtime backend made easy
- Fast and responsive

**Companies Using Firebase:**

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**

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

**Cons**

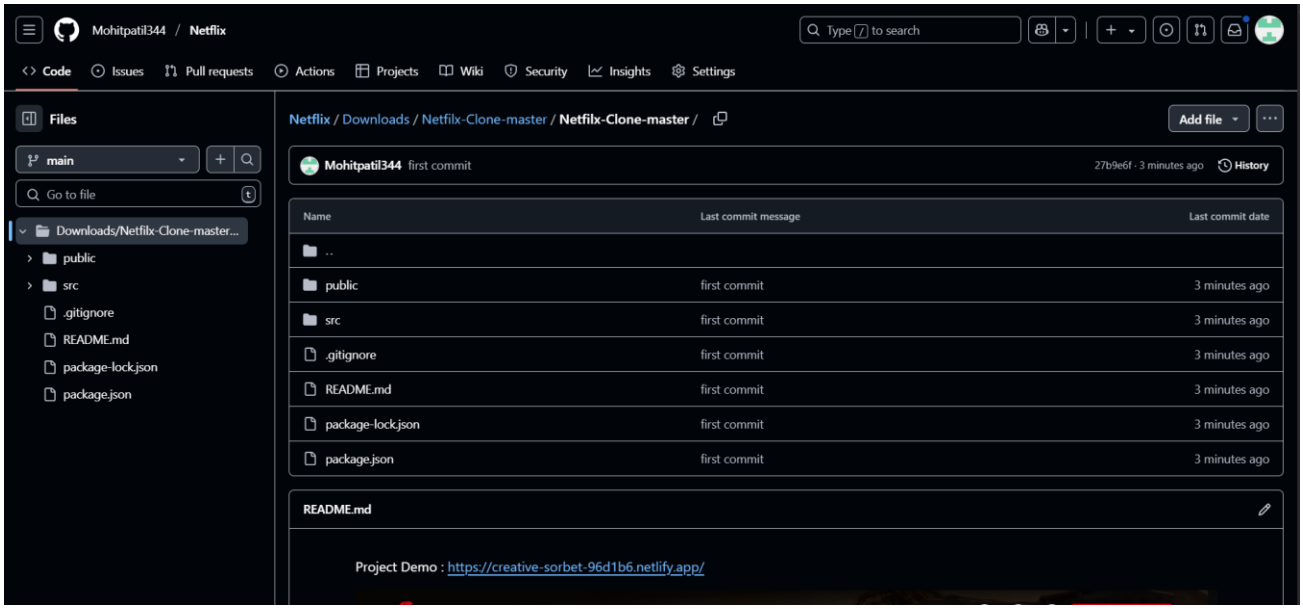
- Only 10 GB of data transfer is allowed per month. But this is not really a big problem, if you use a CDN or AMP.
- Command-line interface only.
- No in-built support for any static site generator.

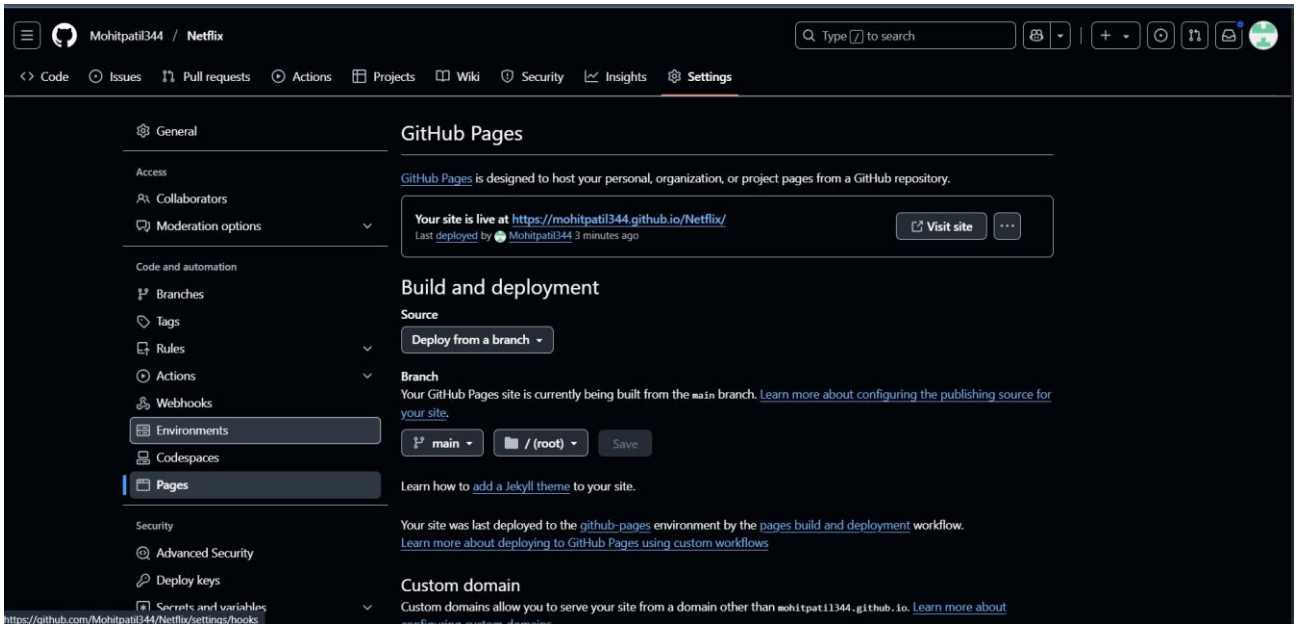
**Link to our GitHub repository:**

https://github.com/Mohitpatil344/Netflix

**Link to our Hosted website:**
https://creative-sorbet-96d1b6.netlify.app/

**Github Screenshot:**



# MAD & PWA Lab
# Journal

| | |
|---|---|
| Experiment No. | 11 |
| Experiment Title. | To use google Lighthouse PWA Analysis Tool to test the PWA functioning. |
| Roll No. | 35 |
| Name | Bhagyesh patil |
| Class | D15A/D15B |
| Subject | MAD & PWA Lab |
| Lab Outcome | LO6: Develop and Analyze PWA Features and deploy it over app hosting solution |
| Grade: | |

# PWA Experiment -11

Bhagyesh Patil 35/D15A

## ❖ Aim:

To use Google Lighthouse PWA Analysis Tool to test the PWA functioning.

## ❖ Theory:

**Google Lighthouse: Overview**

Google Lighthouse is an open-source automated tool developed by Google to audit web applications based on multiple parameters, including performance, accessibility, SEO, best practices, and Progressive Web App (PWA) implementations. It provides an in-depth analysis of a webpage by running different tests and generating a detailed report highlighting areas for improvement.

Lighthouse can be executed via **Chrome DevTools**, **Node.js command line**, or as a **browser extension**. It helps developers optimize their applications to enhance user experience, improve mobile responsiveness, and ensure compliance with best web development practices.

**Key Features of Google Lighthouse**

Lighthouse audits web pages for both **desktop** and **mobile** versions. The key metrics analyzed during an audit are as follows:

- **Performance:**

This metric evaluates how fast a webpage loads and becomes interactive for users. The score is based on various factors, including:

- **First Contentful Paint (FCP):** Measures the time taken to render the first visible content.
- **Largest Contentful Paint (LCP):** Measures the time taken for the largest visible element to load.
- **Time to Interactive (TTI):** Measures how long the page takes to become fully interactive.
- **Speed Index:** Indicates how quickly content is visually displayed.
- **Total Blocking Time (TBT):** Calculates the time a page remains unresponsive due to heavy JavaScript execution.

A **high performance score (closer to 100)** means the website loads quickly and delivers a smooth user experience.

- **Progressive Web App (PWA) Analysis:**

Google Lighthouse checks whether a web application follows the **Baseline PWA Checklist** set by Google. It verifies essential PWA components like:

- **Service Workers:** Ensuring offline functionality and background synchronization.
- **Web App Manifest:** Proper implementation of manifest.json for home screen installation.
- **Viewport Handling:** Ensuring mobile-friendliness with <meta name="viewport">.
- **HTTPS:** Ensuring a secure connection for user safety.
- **Responsive Design:** Optimizing layout and content for different screen sizes.
- **Offline Support:** Verifying if key resources are cached to enable offline access.

A high **PWA score** ensures that the application provides an **app-like experience** on mobile devices.

- **Accessibility:**

Accessibility measures how well a web page supports users with disabilities, including those using screen readers and assistive technologies. Lighthouse evaluates accessibility based on:

- **ARIA Attributes:** Proper use of aria-label, aria-required, etc. for better screen reader support.
- **Text Contrast:** Ensuring readable text against the background color.
- **Keyboard Navigation:** Ensuring all elements are accessible via keyboard (no mouse required).
- **Form Labels:** Ensuring form fields have proper labels and descriptions.
- **Alt Text for Images:** Checking if images have alt attributes for visually impaired users.

Accessibility scores are calculated based on pass/fail criteria. A **low score** means that the website is not user-friendly for people with disabilities.

- **Best Practices:**

Lighthouse evaluates whether a web application follows industry-recommended best practices to ensure security, efficiency, and maintainability. It checks for:

- **Use of HTTPS:** Ensuring a secure connection.
- **Deprecated Code:** Identifying outdated HTML tags, CSS styles, and JavaScript APIs.
- **Password Protection:** Verifying that users can securely input passwords (e.g., disabling "paste" for password fields).
- **Safe JavaScript Execution:** Identifying possible security risks and performance issues in JavaScript code.
- **Geo-Location and Cookie Alerts:** Ensuring compliance with privacy regulations like GDPR by displaying necessary permission prompts.

A **high Best Practices score** ensures the website is built using modern, secure, and efficient coding techniques.

- **SEO (Search Engine Optimization):**

SEO audits help determine how well a webpage is optimized for search engines. Lighthouse checks:

- **Meta Tags:** Ensuring title and meta description are properly set.
- **Mobile-Friendliness:** Verifying that the website is optimized for mobile devices.
- **Canonical URLs:** Preventing duplicate content issues.
- **Crawlability:** Ensuring search engines can index the website properly.

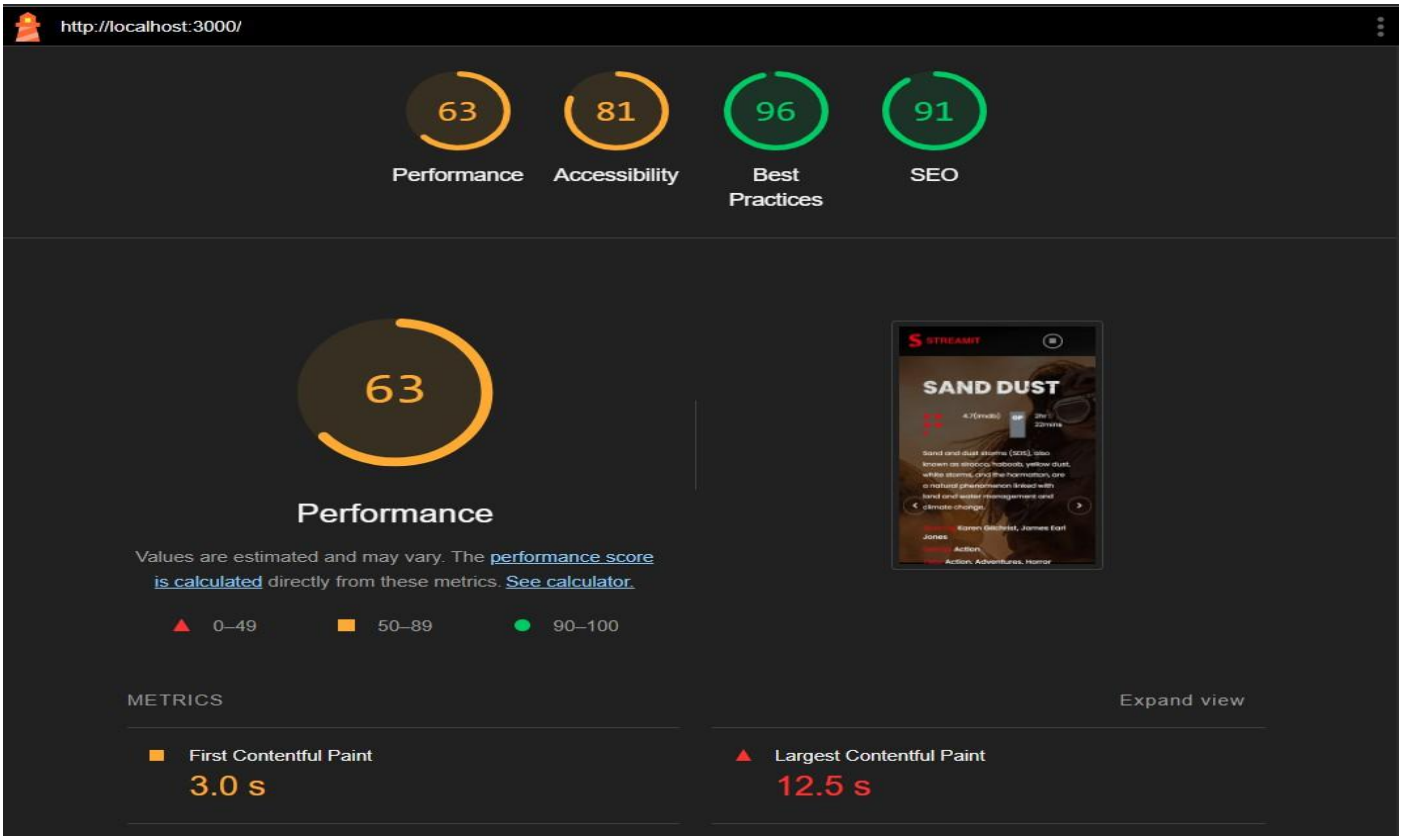
A high **SEO score** improves a website's ranking on search engines like Google.

**manifest.json**

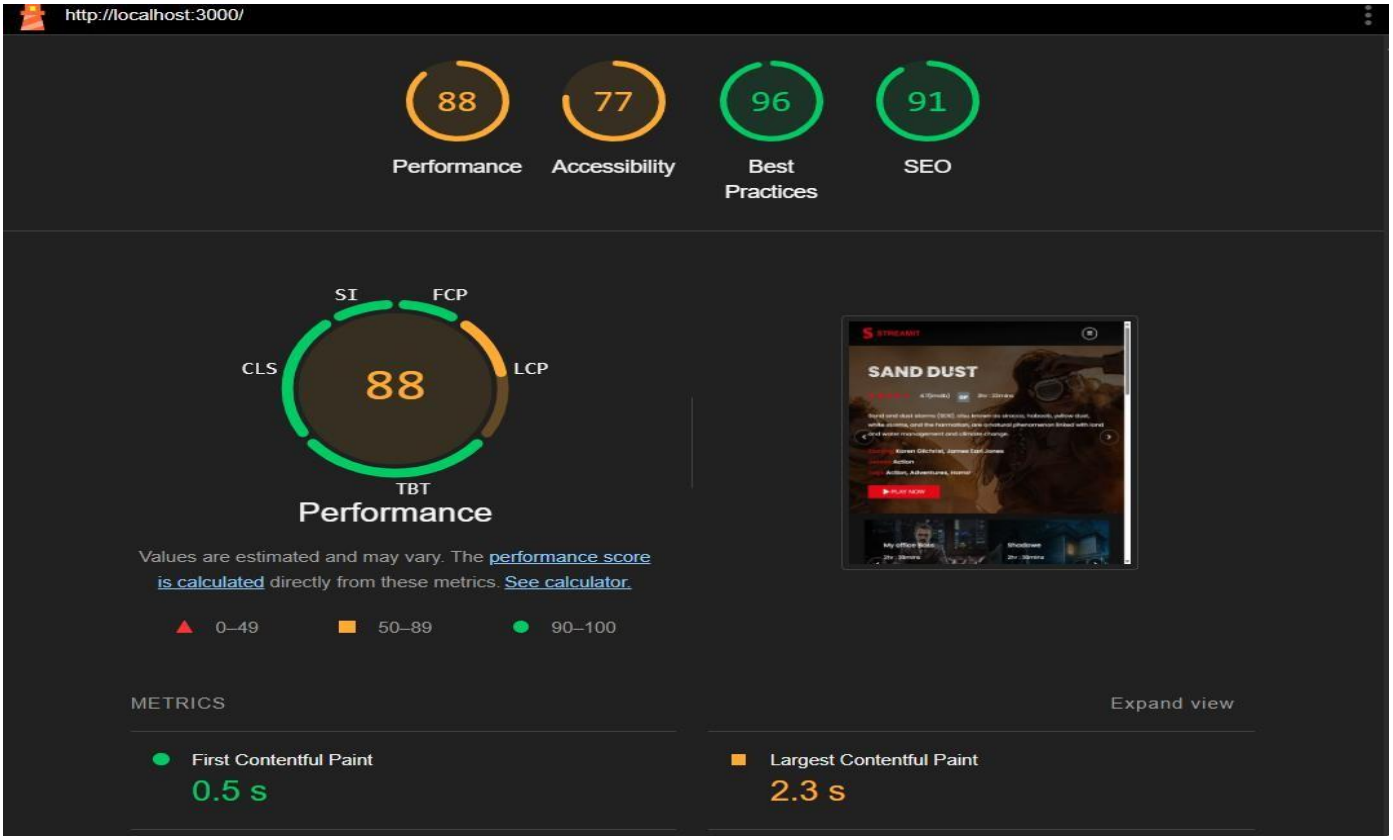

```
{
  "name": "Streamo - Netflix",
  "short_name": "Streamo",
  "start_url": "/",
  "display": "standalone",
  "background_color": "#000000",
  "theme_color": "#000000",
  "description": "Watch unlimited movies & TV shows.",
  "icons": [
    {
      "src": "/logo.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "/logo.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ]
```

## ❖ Output

- Before Code change




- After code change



**Project Title:** **Roll No.**

# MAD & PWA Lab
# Journal

| | |
|---|---|
| Experiment No. | Assignment-1 |
| Assignment 1 Questions | 1. Flutter Overview:Explain the key features and advantages of using Flutter for mobile app development. Discuss how the Flutter framework differs from traditional approaches and why it has gained popularity in the developer community.<br>2. Widget Tree and Composition: Describe the concept of the widget tree in Flutter. Explain how widget composition is used to build complex user interfaces. Provide examples of commonly used widgets and their roles in creating a widget tree.<br>3. State Management in Flutter: Discuss the importance of state management in Flutter applications. Compare and contrast the different state management approaches available in Flutter, such as setState, Provider, and Riverpod. Provide scenarios where each approach is suitable.<br>4. Firebase Integration in Flutter: Explain the process of integrating Firebase with a Flutter application. Discuss the benefits of using Firebase as a backend solution. Highlight the Firebase services commonly used in Flutter development and provide a brief overview of how data synchronization is achieved. |
| Roll No. | 35 |
| Name | Bhagyesh Patil |
| Class | D15A |
| Subject | MAD & PWA Lab |
| Lab Outcome | LO1: Understand cross platform mobile application development using Flutter framework<br>LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation<br>LO3: Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS |
| Grade: | |

**Project Title:** **Roll No.**

# MAD & PWA Lab
# Journal

| | |
|---|---|
| Experiment No. | Assignment-2 |
| Assignment 2 Questions | 1. Define Progressive Web App (PWA) and explain its significance in modern web development.Discuss the key characteristics that differentiate PWAs from traditional mobile apps<br>2. Define responsive web design and explain its importance in the context of Progressive Web Apps.Compare and contrast responsive, fluid, and adaptive web design approaches.<br>3. Describe the lifecycle of Service Workers, including registration, installation, and activation phases.<br>4. Explain the use of IndexedDB in the Service Worker for data storage. |
| Roll No. | 35 |
| Name | Bhagyesh Patil |
| Class | D15A |
| Subject | MAD & PWA Lab |
| Lab Outcome | LO4:Understand various PWA frameworks and their requirements<br>LO5: Design and Develop a responsive User Interface by applying PWA Design techniques<br>LO6:Develop and Analyze PWA Features and deploy it over app hosting solutions |
| Grade: | |