

# **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 Manorath Ital 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**

Name of the Course : MAD & PWA Lab

Course Code : ITL604

<b>Project Title:</b>	<b>Roll No.</b>
-----------------------	-----------------

**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](mailto: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.

<b>Project Title:</b>	<b>Roll No.</b>
PO11) Project Management and Finance: Demonstrate knowledge and understanding of computer engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	

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

#### **Program specific Outcomes**

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

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

**Project Title:****Roll No.****Lab Objectives:**

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

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

# Index

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

**Project Title:**

**Roll No.**

## MAD & PWA Lab Journal

Experiment No.	01
Experiment Title.	To install and configure the Flutter Environment
Roll No.	19
Name	Manorath Ital
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO1: Understand cross platform mobile application development using Flutter framework
Grade:	

**Project Title:**

**Roll No.**

# **EXPERIMENT -1**

**NAME:** MANORATH ITAL

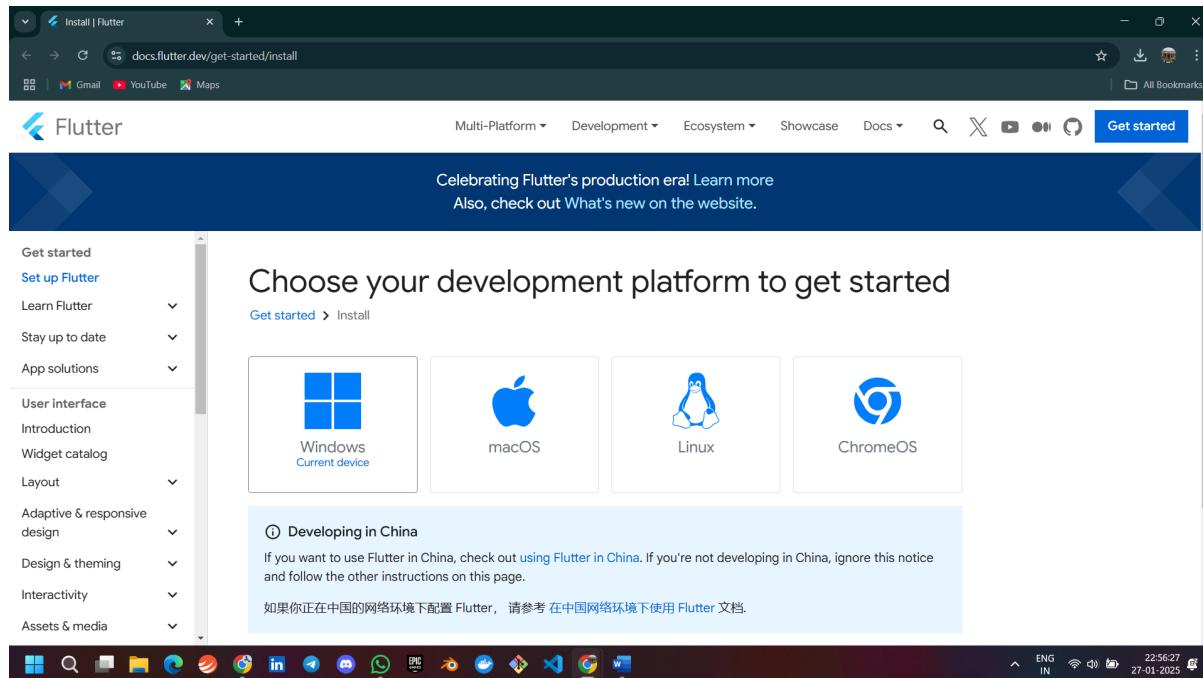
**CLASS:** D15A

**ROLLNO:** 19

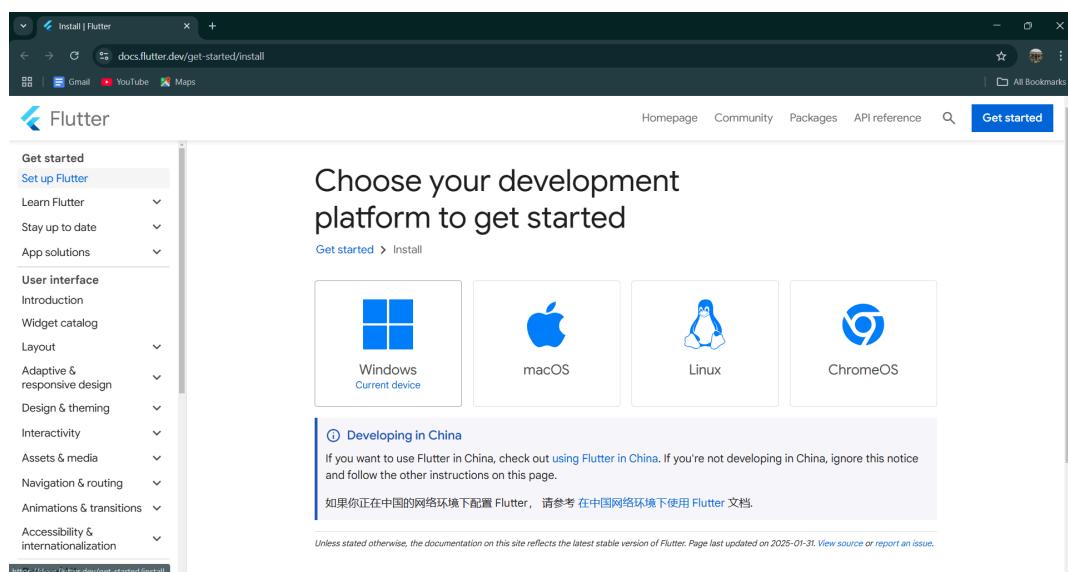
**AIM:** Installation and Configuration of Flutter Environment.

---

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



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



**Project Title:**

**Roll No.**

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

The screenshot shows a web browser window with the URL [docs.flutter.dev/get-started/install/windows/mobile](https://docs.flutter.dev/get-started/install/windows/mobile). The page is titled "Flutter" and has a sidebar on the left with sections like "Get started", "Set up Flutter", "Learn Flutter", etc. The main content area is titled "Download then install Flutter". It instructs users to download the latest stable release of the Flutter SDK, specifically pointing to the file [flutter\\_windows\\_3.27.3-stable.zip](#). A warning section notes that Flutter should be installed to the Windows default download directory (%USERPROFILE%\Downloads) or at %USERPROFILE% (C:\Users\{username}) or %LOCALAPPDATA% (C:\Users\{username}\AppData\Local). It also lists several configuration steps on the right side of the page.

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

The screenshot shows the AukZip application window. The left sidebar has buttons for "Extract", "Compress", and "Upgrade". The main area shows a list of files with the path "C:\Users\manor\Downloads" and a selected item "flutter\_windows\_3.27.3-stable.zip". Below this, there's an "Output Path" field set to "C:\Flutter" and two buttons: "Extract All" and "Extract Selected Items". On the right, there are sections for "Preview" (which says "Select an item to preview") and "Archive Comments".

**Project Title:**

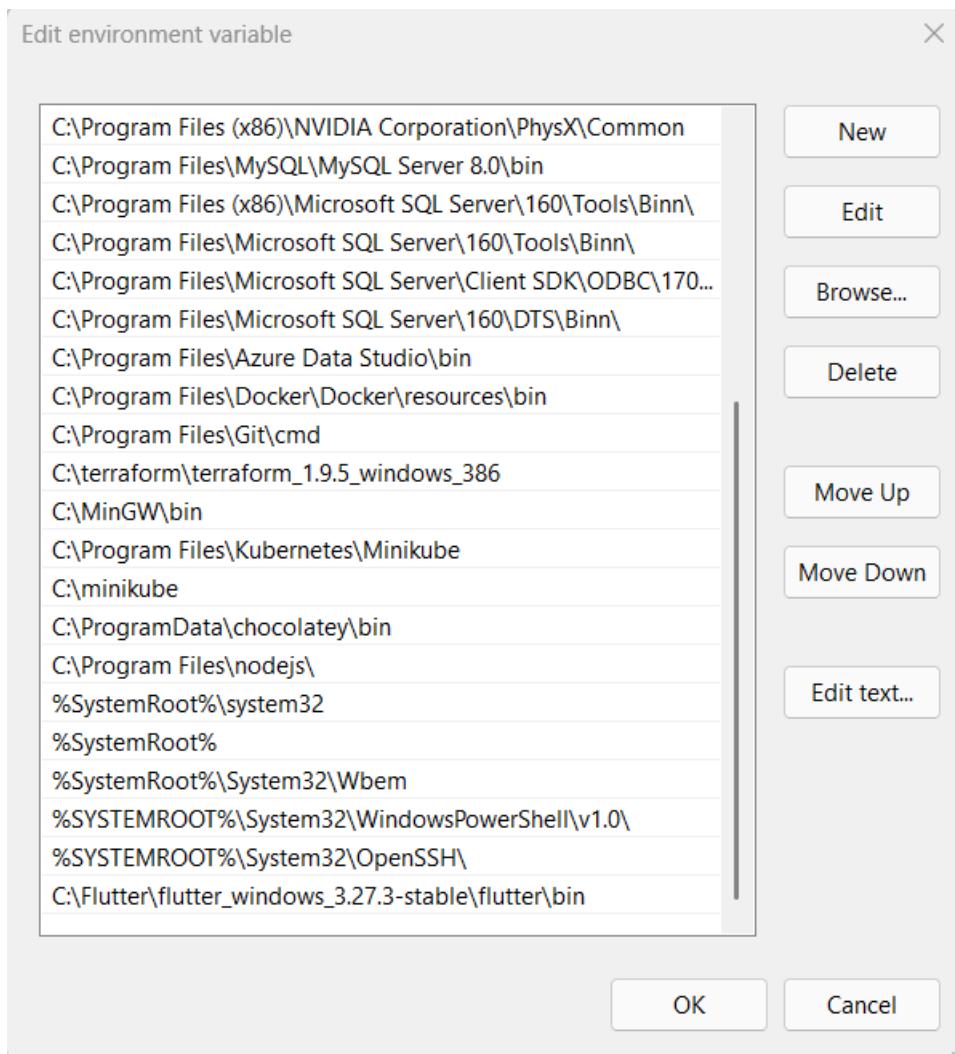
**Roll No.**

#### Step 5 :- Add Flutter to System PATH

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

Under System Variables, find Path and click Edit.

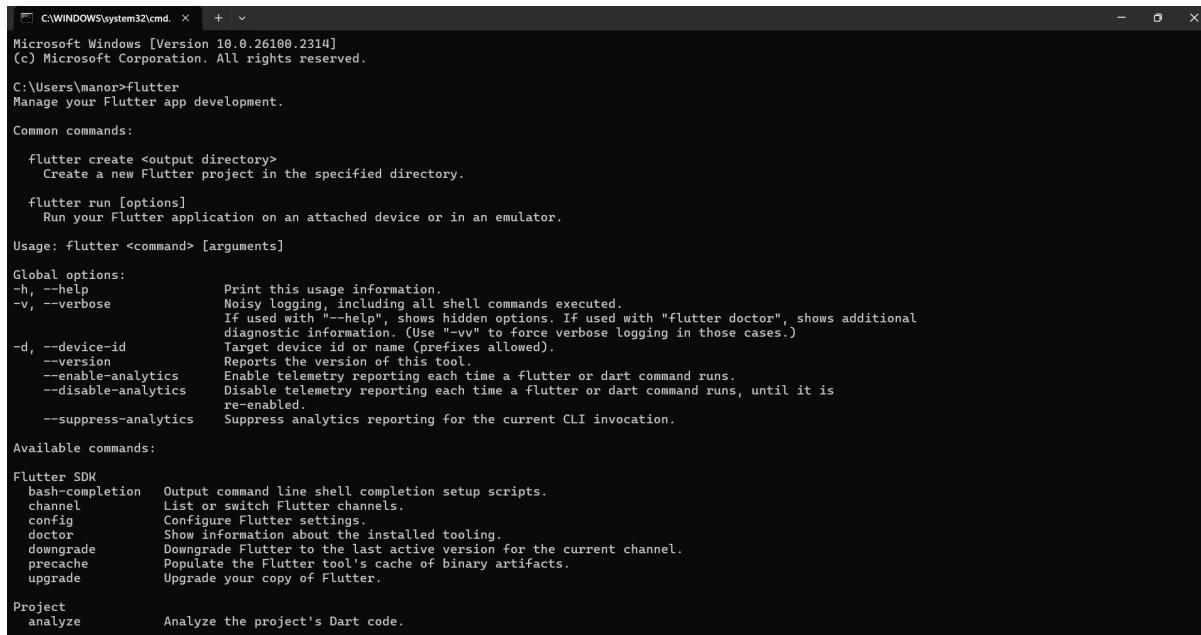
Add the full path to the flutter/bin directory (e.g., C:\flutter\bin).



**Project Title:**

**Roll No.**

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



```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.26100.2314]
(c) Microsoft Corporation. All rights reserved.

C:\Users\manor>flutter
Manage your Flutter app development.

Common commands:

  flutter create <output directory>
    Create a new Flutter project in the specified directory.

  flutter run [<options>]
    Run your Flutter application on an attached device or in an emulator.

Usage: flutter <command> [<arguments>]

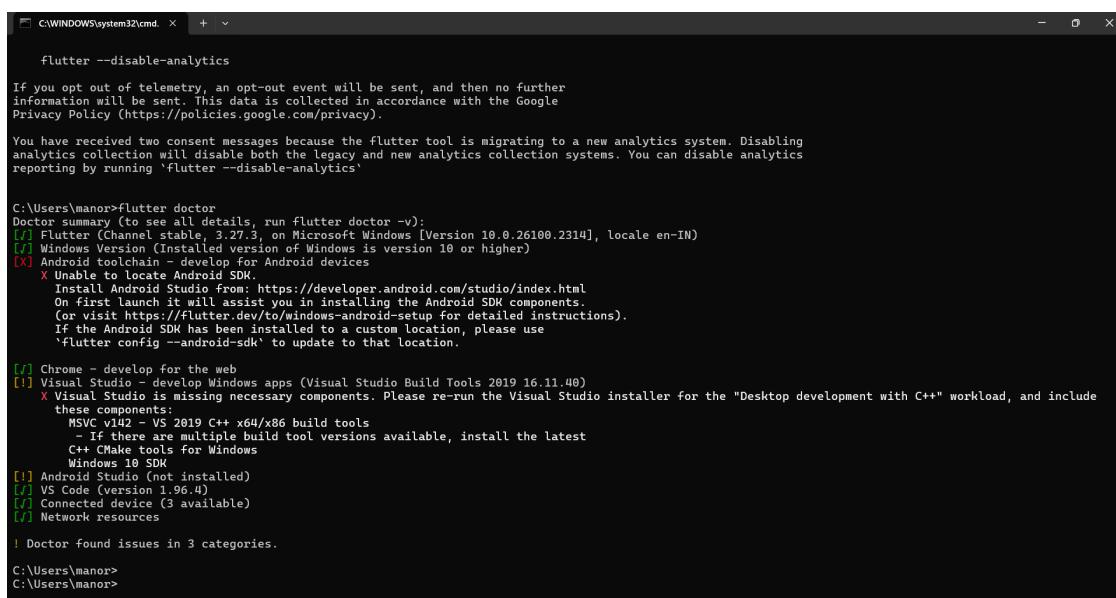
Global options:
  -h, --help           Print this usage information.
  -v, --verbose        Noisy logging, including all shell commands executed.
  If used with "--help", shows hidden options. If used with "flutter doctor", shows additional
  diagnostic information. (Use "-vv" to force verbose logging in those cases.)
  -d, --device-id     Target device id or name (prefixes allowed).
  --version          Reports the version of this tool.
  --enable-analytics Enable telemetry reporting each time a flutter or dart command runs.
  --disable-analytics Disable telemetry reporting each time a flutter or dart command runs, until it is
  re-enabled.
  --suppress-analytics Suppress analytics reporting for the current CLI invocation.

Available commands:

Flutter SDK
  bash-completion   Output command line shell completion setup scripts.
  channel           List or switch Flutter channels.
  config            Configure Flutter settings.
  doctor             Show information about the installed tooling.
  downgrade          Downgrade Flutter to the last active version for the current channel.
  precache           Populate the Flutter tool's cache of binary artifacts.
  upgrade            Upgrade your copy of Flutter.

Project
  analyze           Analyze the project's Dart code.
```

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



```
C:\WINDOWS\system32\cmd. x + v
Flutter --disable-analytics

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

You have received two consent messages because the flutter tool is migrating to a new analytics system. Disabling
analytics collection will disable both the legacy and new analytics collection systems. You can disable analytics
reporting by running 'Flutter --disable-analytics'

C:\Users\manor>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 3.27.3, on Microsoft Windows [Version 10.0.26100.2314], locale en-IN)
[!] Windows Version (Installed version of Windows is version 10 or higher)
[X] Android toolchain - develop for Android devices
  X Unable to locate Android SDK.
    Install Android Studio from: https://developer.android.com/studio/index.html
    On first launch it will assist you in installing the Android SDK components.
    (or visit https://flutter.dev/to/windows-android-setup for detailed instructions).
    If the Android SDK has been installed to a custom location, please use
      'flutter config --android-sdk' to update to that location.

[!] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Build Tools 2019 16.11.40)
  X Visual Studio is missing necessary components. Please re-run the Visual Studio installer for the "Desktop development with C++" workload, and include
    these components:
      MSVC v142 - VS 2019 C++ x64/x86 build tools
        - If there are multiple build tool versions available, install the latest
          C++ CMake tools for Windows
          Windows 10 SDK
[!] Android Studio (Not installed)
[!] VS Code (version 1.96.4)
[!] Connected device (3 available)
[!] Network resources

! Doctor found issues in 3 categories.

C:\Users\manor>
C:\Users\manor>
```

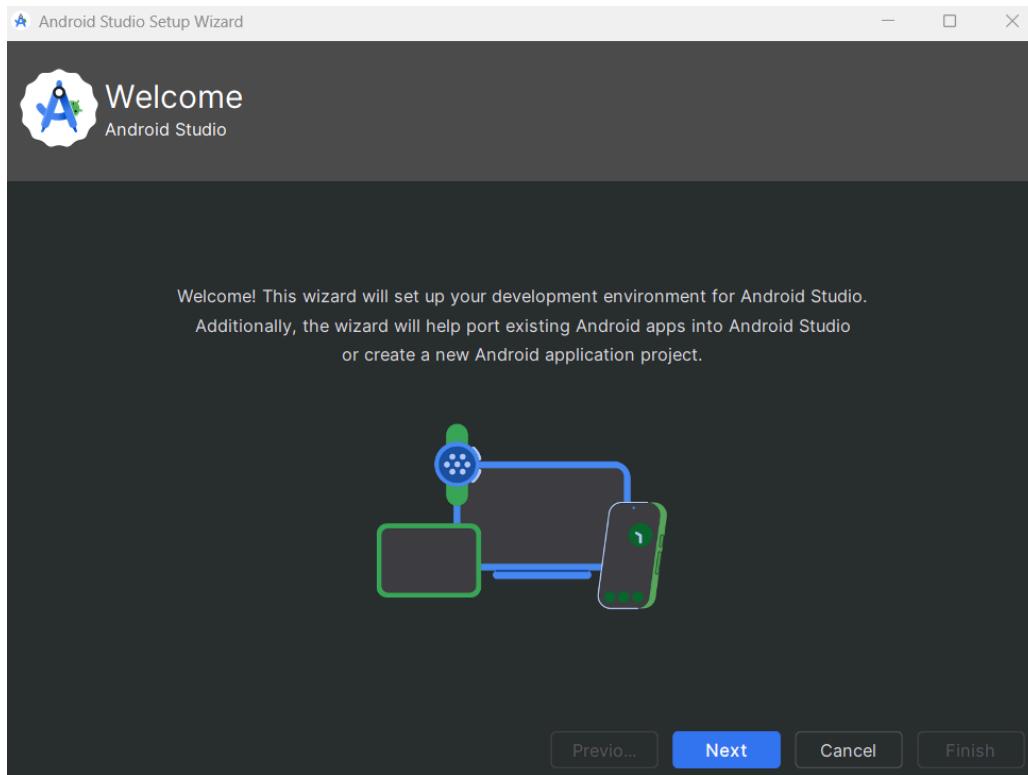
**Project Title:**

**Roll No.**

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

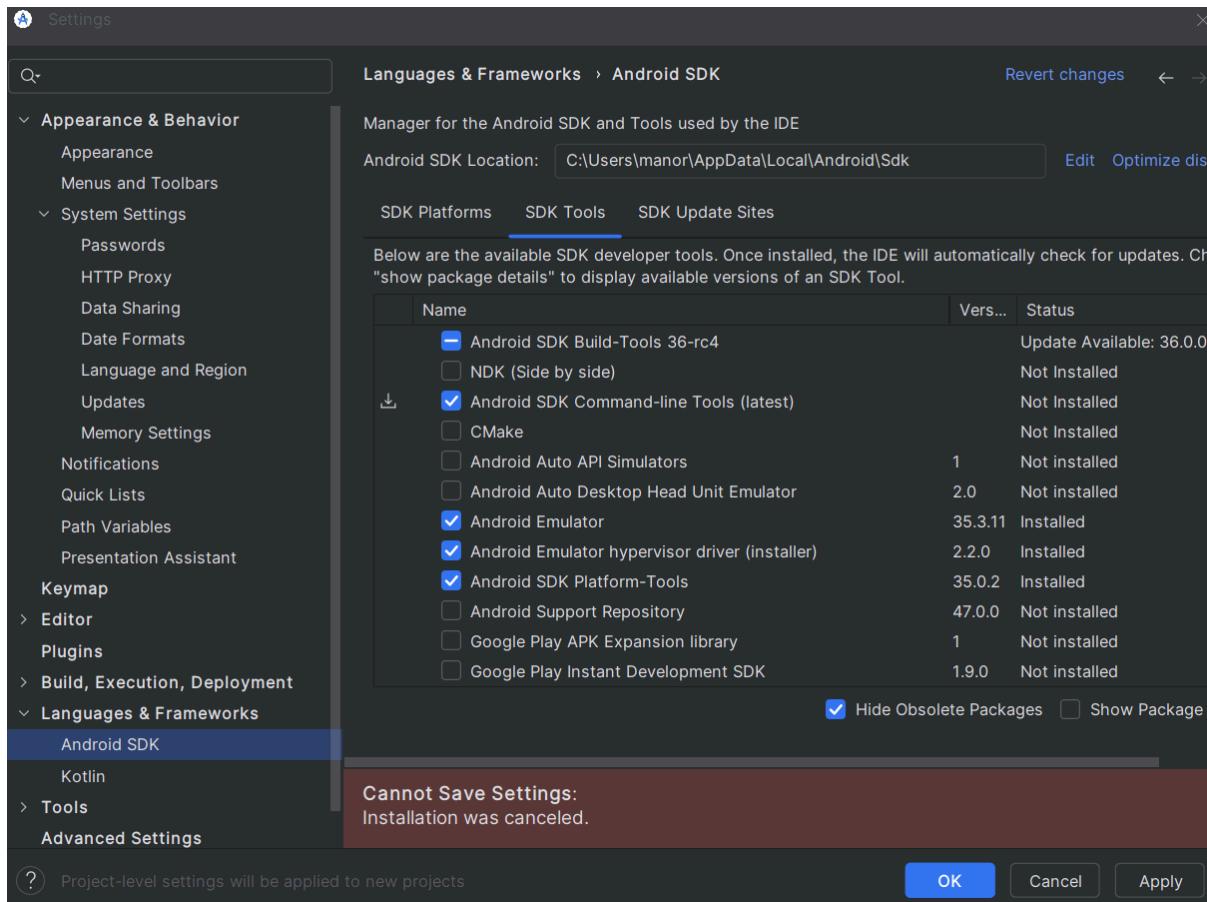
Platform	Android Studio package	Size	SHA-256 checksum
Windows (64-bit)	<a href="#">android-studio-2024.2.2.13-windows.exe</a> Recommended	1.2 GB	7d93dd9bf3539f948f609b1968507bf502bf6965d2d44bd38a17ff26cb5dd3e
Windows (64-bit)	<a href="#">android-studio-2024.2.2.13-windows.zip</a> No .exe installer	1.2 GB	855945962ff9b84ea49ce39de0bf4189dbf451ae37a6fab7999da013b046b7f7
Mac (64-bit)	<a href="#">android-studio-2024.2.2.13-mac.dmg</a>	1.3 GB	acfbbe54d6ce8cf2ff19b43510c7addcb9dde2824282f205fd1331be77d2e613

Step 9: - When the download is complete, open the .exe file and run it. Follow the steps of the installation wizard. Once the installation wizard completes, you will get the following screen.



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

Step 10: - Go to Preferences > Appearance & Behavior > System Settings > Android SDK. Select the SDK Tools tab and check Android SDK Command-line Tools and Install it.



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

1) flutter doctor --android-licenses

```
C:\WINDOWS\system32\cmd. x + v
C:\Users\manor>flutter doctor --android-licenses
Warning: Additionally, the fallback loader failed to parse the XML.
Warning: Errors during XML parse: [ 62% Fetch remote repository...
Warning: Additionally, the fallback loader failed to parse the XML...[=====
[=====] 100% Computing updates...
8 of 7 SDK package licenses have not been accepted.
Review licenses that have not been accepted (y/N)? y
1/6: License android-googletv-license:
-----
Terms and Conditions
This is the Google TV Add-on for the Android Software Development Kit License Agreement.
1. Introduction
1.1 The Google TV Add-on for the Android Software Development Kit (referred to in this License Agreement as the "Google TV Add-on" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of this License Agreement. This License Agreement forms a legally binding contract between you and Google in relation to your use of the Google TV Add-on.
1.2 "Google" means Google Inc., a Delaware corporation with principal place of business at 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States.
2. Accepting this License Agreement
2.1 In order to use the Google TV Add-on, you must first agree to this License Agreement. You may not use the Google TV Add-on if you do not accept this License Agreement.
2.2 You can accept this License Agreement by:
(A) clicking to accept or agree to this License Agreement, where this option is made available to you; or
(B) by actually using the Google TV Add-on. In this case, you agree that use of the Google TV Add-on constitutes acceptance of the License Agreement from this point onwards.
2.3 You may not use the Google TV Add-on and may not accept the Licensing Agreement if you are a person barred from receiving the Google TV Add-on under the laws of the United States or other countries including the country in which you are resident or from which you use the Google TV Add-on.
2.4 If you are agreeing to be bound by this License Agreement on behalf of your employer or other entity, you represent and warrant that you have full legal
```

**Project Title:**

**Roll No.**

Accept all the licenses

```
C:\WINDOWS\system32\cmd. x + v

en agreements previously existing between Recipient and MIPS with respect to the subject matter hereof. This Agreement may only be amended or supplemented b
y a writing that refers explicitly to this Agreement and that is signed or otherwise accepted by duly authorized representatives of Recipient and MIPS.

10.5 Severability. In the event that any provision of this Agreement is finally adjudicated to be unenforceable or invalid under any applicable law, such un
enforceability or invalidity shall not render this Agreement unenforceable or invalid as a whole, and, in such event, such unenforceable or invalid provisio
n shall be interpreted so as to best accomplish the objectives of such provision within the limits of applicable law or applicable court decisions.

10.6 Export Regulations / Export Control. Recipient shall not export, either directly or indirectly, any product, service or technical data or system incorp
orating the Evaluation Materials without first obtaining any required license or other necessary approval from the U.S. Department of Commerce or any other
governing agency or department of the United States Government. In the event any product is exported from the United States or re-exported from a foreign de
stination by Recipient, Recipient shall ensure that the distribution and export/re-export or import of the product is in compliance with all applicable laws
, regulations, orders, or other restrictions of the U.S. Export Administration Regulations and the appropriate foreign government. Recipient agrees that nei
ther it nor any of its subsidiaries will export/re-export any technical data, process, product, or service, directly or indirectly, to any country for which
the United States government or any agency thereof or the foreign government from where it is shipping requires an export license, or other governmental ap
proval, without first obtaining such license or approval. Recipient also agrees to implement measures to ensure that foreign national employees are authoriz
ed to receive any information controlled by U.S. export control laws. An export is "deemed" to take place when information is released to a foreign national
wherever located.

10.7 Special Terms for Pre-Release Materials. If so indicated in the description of the Evaluation Software, the Evaluation Software may contain Pre-Release
Materials. Recipient hereby understands, acknowledges and agrees that: (i) Pre-Release Materials may not be fully tested and may contain bugs or errors; (ii)
Pre-Release materials are not suitable for commercial release in their current state; (iii) regulatory approvals for Pre-Release Materials (such as UL or
FCC) have not been obtained, and Pre-Release Materials may therefore not be certified for use in certain countries or environments or may not be suitable f
or certain applications and (iv) MIPS can provide no assurance that it will ever produce or make generally available a production version of the Pre-Release
Materials . MIPS is not under any obligation to develop and/or release or offer for sale or license a final product based upon the Pre-Release Materials an
d may unilaterally elect to abandon the Pre-Release Materials or any such development platform at any time and without any obligation or liability whatsoeve
r to Recipient or any other person.

ANY PRE-RELEASE MATERIALS ARE NON-QUALIFIED AND, AS SUCH, ARE PROVIDED *AS IS* AND *AS AVAILABLE*, POSSIBLY WITH FAULTS, AND WITHOUT REPRESENTATION OR WARRA
NTY OF ANY KIND.

10.8 Open Source Software. In the event Open Source software is included with Evaluation Software, such Open Source software is licensed pursuant to the app
licable Open Source software license agreement identified in the Open Source software comments in the applicable source code file(s) and/or file header as i
ndicated in the Evaluation Software. Additional detail may be available (where applicable) in the accompanying on-line documentation. With respect to the Op
en Source software, nothing in this Agreement limits any rights under, or grants rights that supersede, the terms of any applicable Open Source software lic
ense agreement.

-----
Accept? (y/N): y
All SDK package licenses accepted

C:\Users\manor>
```

2)flutter doctor

```
C:\WINDOWS\system32\cmd. x + v

i) Pre-Release materials are not suitable for commercial release in their current state; (iii) regulatory approvals for Pre-Release Materials (such as UL or
FCC) have not been obtained, and Pre-Release Materials may therefore not be certified for use in certain countries or environments or may not be suitable f
or certain applications and (iv) MIPS can provide no assurance that it will ever produce or make generally available a production version of the Pre-Release
Materials . MIPS is not under any obligation to develop and/or release or offer for sale or license a final product based upon the Pre-Release Materials an
d may unilaterally elect to abandon the Pre-Release Materials or any such development platform at any time and without any obligation or liability whatsoeve
r to Recipient or any other person.

ANY PRE-RELEASE MATERIALS ARE NON-QUALIFIED AND, AS SUCH, ARE PROVIDED *AS IS* AND *AS AVAILABLE*, POSSIBLY WITH FAULTS, AND WITHOUT REPRESENTATION OR WARRA
NTY OF ANY KIND.

10.8 Open Source Software. In the event Open Source software is included with Evaluation Software, such Open Source software is licensed pursuant to the app
licable Open Source software license agreement identified in the Open Source software comments in the applicable source code file(s) and/or file header as i
ndicated in the Evaluation Software. Additional detail may be available (where applicable) in the accompanying on-line documentation. With respect to the Op
en Source software, nothing in this Agreement limits any rights under, or grants rights that supersede, the terms of any applicable Open Source software lic
ense agreement.

-----
Accept? (y/N): y
All SDK package licenses accepted

C:Users\manor>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 3.27.3, on Microsoft Windows [Version 10.0_26100_2314], locale en-IN)
[!] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices (Android SDK version 35.0.1)
[!] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Build Tools 2019 16.11.40)
  X Visual Studio is missing necessary components. Please re-run the Visual Studio installer for the "Desktop development with C++" workload, and include
    these components:
      MSVC v142 - VS 2019 C++ x64/x86 build tools
        - If there are multiple build tool versions available, install the latest
          C++ CMake tools for Windows
          Windows 10 SDK
[!] Android Studio (version 2024.2)
[!] VS Code (version 1.96.4)
[!] Connected device (3 available)
[!] Network resources

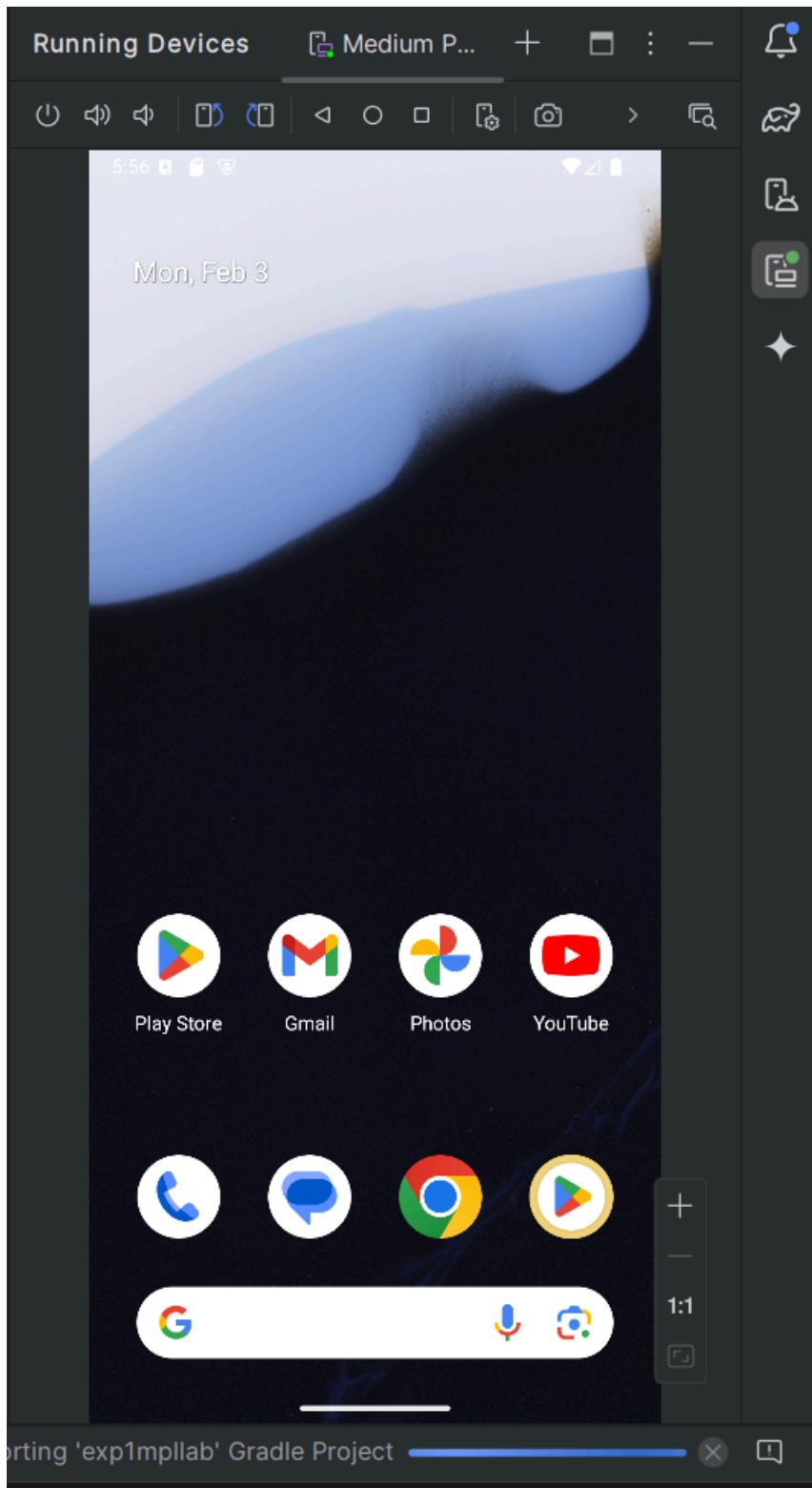
! Doctor found issues in 1 category.

C:\Users\manor>
```

**Project Title:**

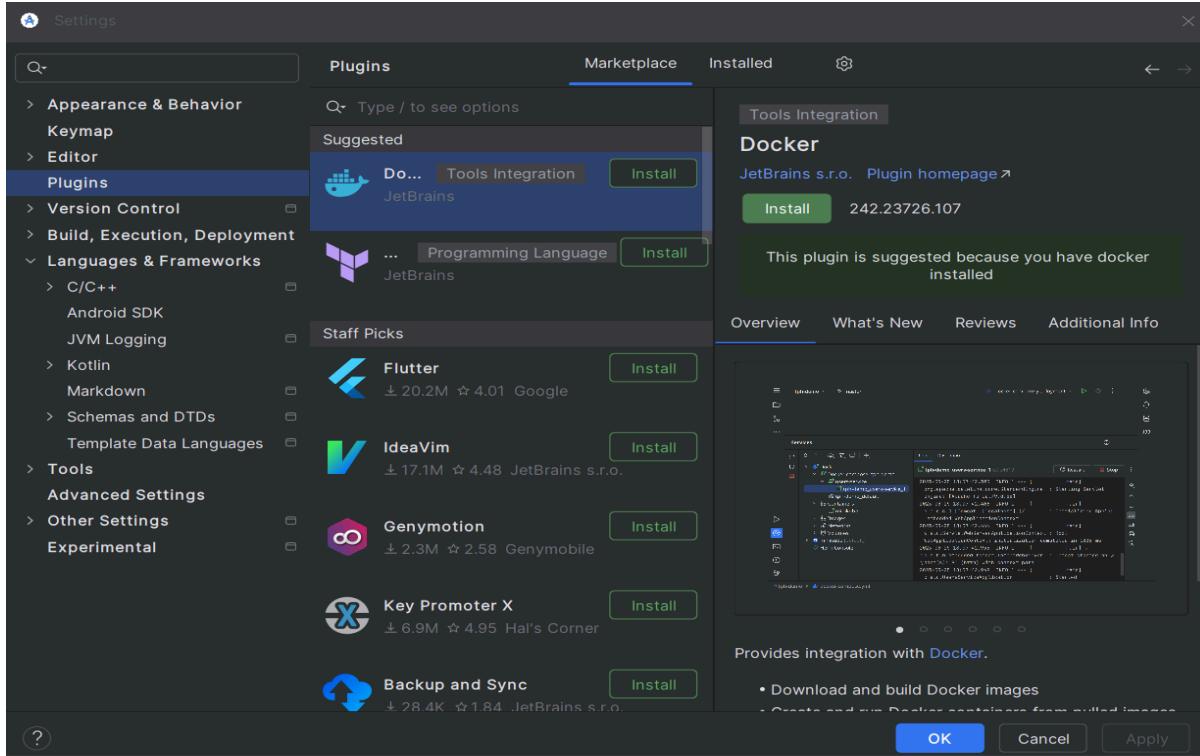
**Roll No.**

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



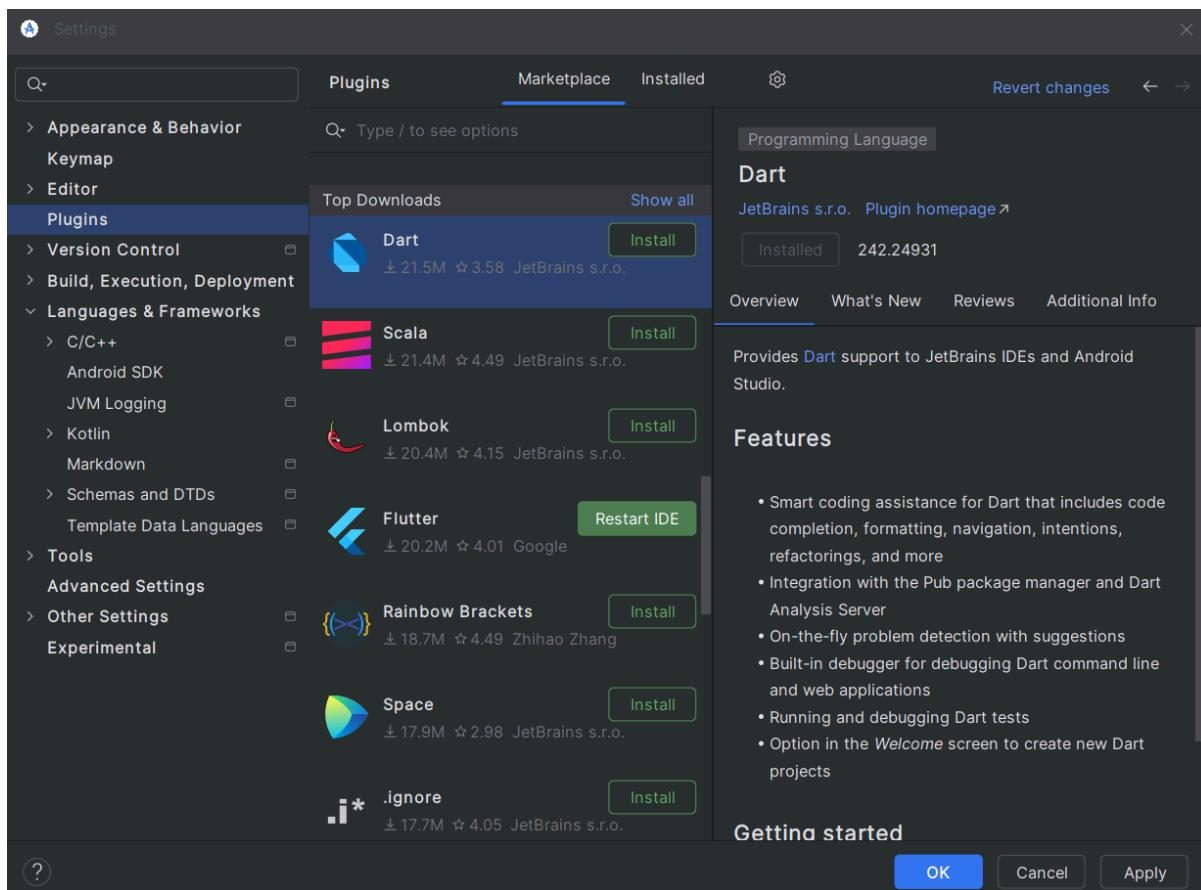
**Project Title:****Roll No.**

Step 11: - Now, install Flutter and Dart plugin for building Flutter application in Android Studio. These plugins provide a template to create a Flutter application, give an option to run and debug Flutter application in the Android Studio itself. Open the Android Studio and then go to File->Settings->Plugins. Now, search the Flutter plugin. If found, select Flutter plugin and click install

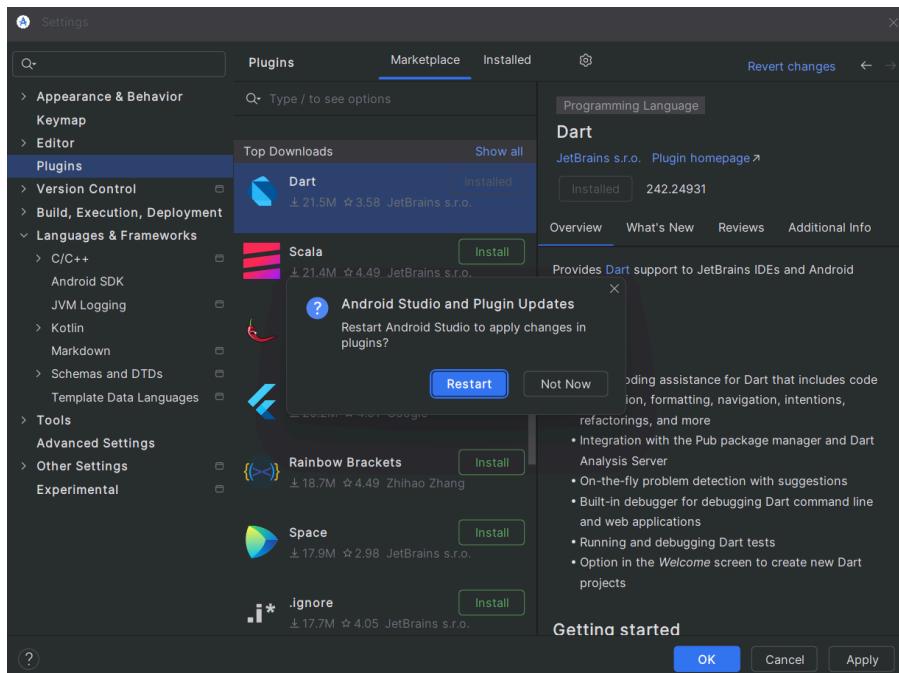


**Project Title:**

**Roll No.**



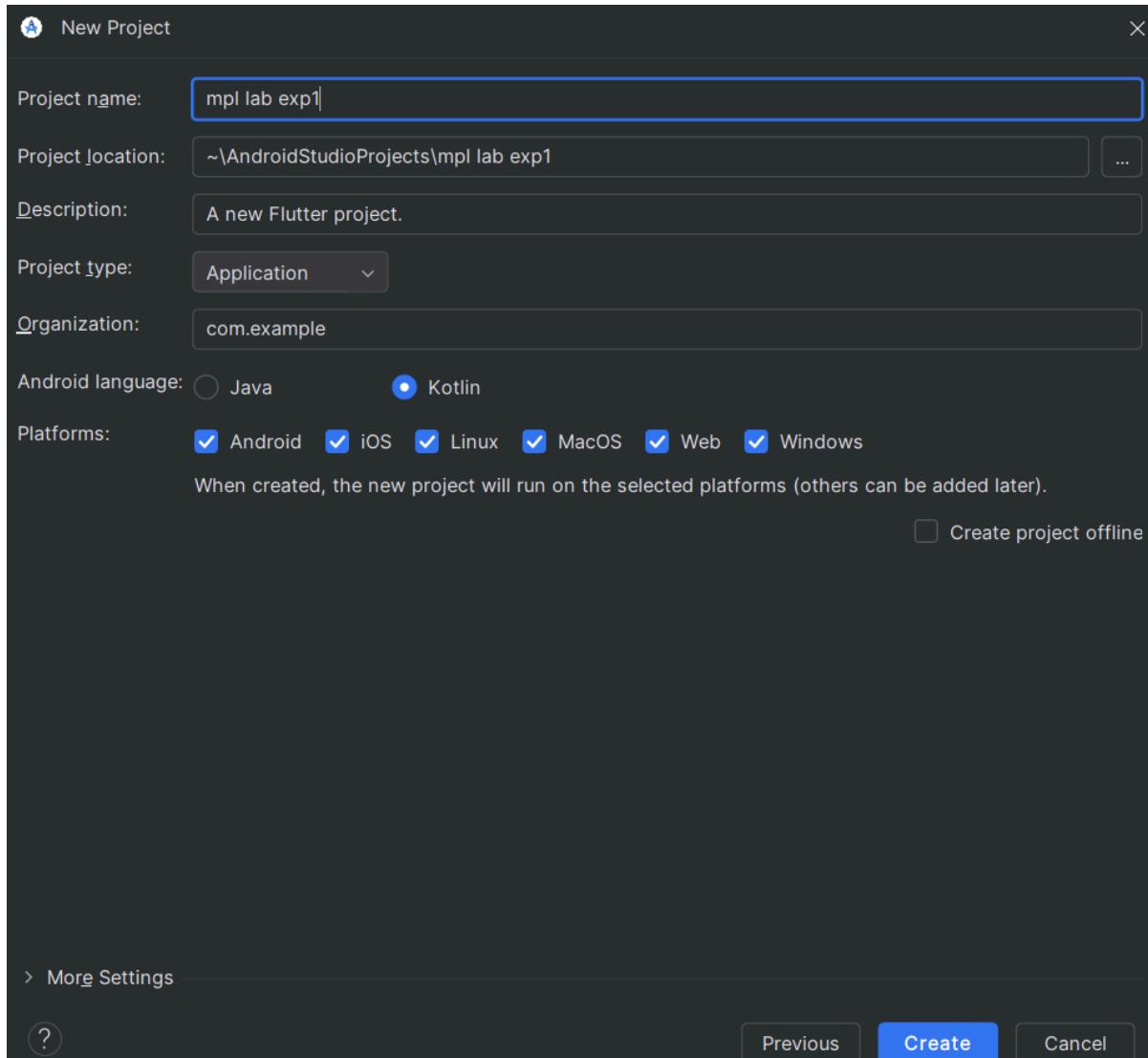
## Step 12: - Restart the Android Studio



**Project Title:**

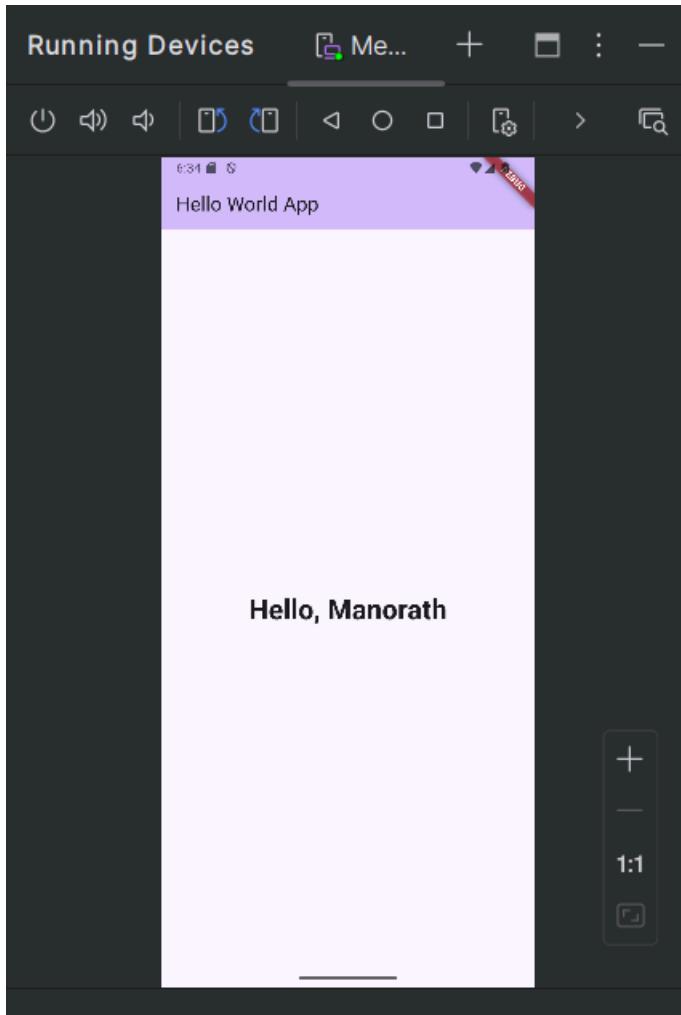
**Roll No.**

Step 13: - Go to File > New Project > Create Flutter Project, then select the project name and location, and click Next to proceed.



**Project Title:**

**Roll No.**



**Project Title:**

**Roll No.**

## MAD & PWA Lab Journal

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

**Project Title:**

**Roll No.**

**Aim:**

To design Flutter UI by including common widgets.

**Theory:**

Flutter follows a widget-based approach where everything in the UI is a widget. Widgets can be classified into two main types:

- **Stateless Widgets:** Do not change their state once built (e.g., Text, Container).
- **Stateful Widgets:** Can update dynamically based on user interaction (e.g., TextField, Checkbox).

## Commonly Used Widgets in Flutter

### (a) Scaffold Widget

The **Scaffold** widget provides the basic structure for a Flutter app, including an **AppBar**, **Drawer**, **FloatingActionButton**, and **BottomNavigationBar**. It is a fundamental widget used to create a standard screen layout in Flutter.

### (b) Container Widget

A **Container** is a box model widget that can hold other widgets. It is commonly used for adding padding, margins, borders, and background decorations.

### (c) Row and Column Widgets

- **Row:** Arranges widgets horizontally.
  - **Column:** Arranges widgets vertically.
- These two widgets are fundamental for designing layouts in Flutter.

### (d) ListView Widget

The **ListView** widget is used for displaying a scrollable list of items. It is useful for showing large amounts of data dynamically.

### (e) Stack Widget

The **Stack** widget is used to place widgets on top of each other. This is useful for creating overlapping UI elements such as banners, profile images, or layered designs.

### (f) ElevatedButton Widget

The **ElevatedButton** widget is used for clickable buttons with a raised effect. It is a commonly used button in Flutter applications.

**Project Title:**

**Roll No.**

### **(g) TextField Widget**

The **TextField** widget is used to take user input, such as entering a name, email, or password. It is commonly used in forms and authentication screens.

### **Home Page**

This home page is designed using Flutter and consists of various widgets to create an engaging UI for a movie app called **FilmyFun**.

### **Widgets Used:**

1. **Scaffold Widget** – Provides the basic structure of the app.
2. **Container Widget** – Used for layout styling and decorations.
3. **Row & Column Widgets** – Organize UI elements horizontally and vertically.
4. **Text Widget** – Displays textual content.
5. **Image.asset Widget** – Loads images from local assets.
6. **CarouselSlider Widget** – Implements an image slider for featured movie posters.
7. **ListView Widget** – Displays a scrollable list of trending movies.
8. **Stack Widget** – Used for overlaying text and effects on movie images.

### **UI Features:**

#### **1. Header Section:**

- Displays a welcome message: "*Hello, Manorath*".
- Includes a wave emoji icon and a user profile picture.

#### **2. App Title:**

- The app name is displayed in a stylish way:
  - "Filmy" (White text)
  - "Fun" (Golden yellow text)

#### **3. Carousel Image Slider:**

- Showcases featured movie posters in a **carousel slider**.
- Users can swipe through movie images.

#### **4. Trending Movies Section:**

- Displays popular movies in a horizontal scrollable **ListView**.
- Each movie has:
  - A movie poster.
  - A title (e.g., *Infinity Wars*).
  - A genre (e.g., *Action, Adventure*).

**Project Title:**

**Roll No.**

- A semi-transparent overlay for better readability.

**Color Scheme:**

- **Background:** Black for a cinematic feel.
- **Text Colors:** White and golden yellow for contrast.
- **Overlay Effects:** Black with transparency for better readability.

**home.dart**

**Code:**

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

import 'package:flutter/material.dart';

class home extends StatefulWidget {

const home({super.key});

@Override

State<home> createState() => _homeState();

}

class _homeState extends State<home> {

final List<String>imageUrls = [

"images/infinity.jpg",

"images/salman.jpg",

"images/shahrukhmovies.png",

];

@Override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Colors.black,

body: Container(


margin: EdgeInsets.only(top: 40.0,left: 20.0),
```

**Project Title:**

**Roll No.**

```
child: Column(  
  crossAxisAlignment: CrossAxisAlignment.start,  
  children: [  
    Row(children: [  
      Image.asset("images/wave.png", height: 40, width: 40, fit: BoxFit.cover,),  
      SizedBox(width: 10.0,),  
      Text("Hello, Manorath",  
        style: TextStyle(  
          color: Colors.white,  
          fontSize: 22.0,  
          fontWeight: FontWeight.bold),  
      ),  
      Spacer(),  
      Padding(  
        padding: const EdgeInsets.only(right: 20.0),  
        child: ClipRRect(  
          borderRadius: BorderRadius.circular(60),  
          child: Image.asset(  
            "images/Screenshot (55).png",  
            height: 60,  
            width: 60,  
            fit: BoxFit.cover,),  
      ),  
    ]  
  ]  
)
```

**Project Title:**

**Roll No.**

```
],  
,  
SizedBox(height: 10.0),  
Text("Welcome To",  
style: TextStyle(  
color: Color.fromARGB(186, 255, 255, 255),  
fontSize: 19.0,  
fontWeight: FontWeight.bold),),  
Row(children: [  
Text("Filmy",  
style: TextStyle(  
color: Colors.white,  
fontSize: 36.0,  
fontWeight: FontWeight.bold),),  
Text("Fun",  
style: TextStyle(  
color: Color(0xffffdb41d),  
fontSize: 36.0,  
fontWeight: FontWeight.bold),)  
],  
,  
SizedBox(height: 20.0,),  
Center(child:  
CarouselSlider(  

```

**Project Title:**

**Roll No.**

```
items: imageUrl.map((url) {  
    return Builder(builder: (context) {  
        return Container(  
            width: MediaQuery.of(context).size.width, // Fixed incorrect comma  
            child: ClipRRect(  
                borderRadius: BorderRadius.circular(10),  
                child: Image.asset(url, fit: BoxFit.cover),  
            ),  
        );  
    });  
}).toList(), options: CarouselOptions(  
    height: 250,  
    autoPlay: false,  
    enlargeCenterPage: true,  
    aspectRatio: 16/9,  
    viewportFraction: 1.0  
,// Fixed missing .toList()  
)  
,  
SizedBox(height: 25.0,),  
Text("Top Trending Movies",  
    style: TextStyle(  
        color: Colors.white,  
        fontSize: 24.0,
```

**Project Title:**

**Roll No.**

```
fontWeight: FontWeight.bold),),  
SizedBox(height: 20.0,),  
Container(  
height: 250,  
child: ListView(  
scrollDirection: Axis.horizontal,  
children: [  
Container(  
decoration: BoxDecoration(border: Border.all(color: Colors.white),borderRadius:  
BorderRadius.circular(20)),  
child: Stack(  
children: [  
ClipRRect(  
borderRadius: BorderRadius.circular(20),  
child: Image.asset("images/infinity.jpg",  
height: 220,  
width: 180,  
fit: BoxFit.cover,  
),  
),  
Container(  
padding: EdgeInsets.only(left: 10.0),  
margin: EdgeInsets.only(top: 180),  
height: 220,  
width: 180,
```

<b>Project Title:</b>	<b>Roll No.</b>
-----------------------	-----------------

decoration:

```
BoxDecoration(color:  
Colors.black45,  
borderRadius:  
BorderRadius.only(  
bottomRight: Radius.circular(20),  
bottomLeft: Radius.circular(20))),  
child: Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: [  
Text("Infinity Wars", style: TextStyle(color: Colors.white, fontSize: 25.0, fontWeight:  
FontWeight.bold),),  
Text("Action, Adventure", style: TextStyle(color: Color.fromARGB(173, 255, 255, 255), fontSize:  
16.0, fontWeight: FontWeight.bold),),  
],  
),  
,  
],  
,  
),  
),  
),  
SizedBox(width: 20.0),  
Container(  
decoration: BoxDecoration(border: Border.all(color: Colors.white),borderRadius:  
BorderRadius.circular(20)),  
child: Stack(  
children: [
```

**Project Title:**

**Roll No.**

```
ClipRRect(  
    borderRadius: BorderRadius.circular(20),  
    child: Image.asset("images/pushpa.jpg",  
        height: 220,  
        width: 180,  
        fit: BoxFit.cover,  
,  
,
```

```
Container(  
    padding: EdgeInsets.only(left: 10.0),  
    margin: EdgeInsets.only(top: 180),  
    height: 220,  
    width: 180,
```

decoration:  
BoxDecoration(color:

```
Colors.black45,  
borderRadius:  
BorderRadius.only(  
    bottomRight: Radius.circular(20),  
    bottomLeft: Radius.circular(20))),  
child: Column(  
    crossAxisAlignment: CrossAxisAlignment.start,
```

children: [  
 Text("Pushpa 2", style: TextStyle(color: Colors.white, fontSize: 25.0, fontWeight: FontWeight.bold)),

**Project Title:**

**Roll No.**

```
Text("Action, Thriller,", style: TextStyle(color: Color.fromARGB(173, 255, 255, 255), fontSize: 16.0, fontWeight: FontWeight.bold),),  
],  
,  
,  
],  
,  
,  
),  
),  
SizedBox(width: 20.0),  
Container(  
decoration: BoxDecoration(border: Border.all(color: Colors.white),borderRadius: BorderRadius.circular(20)),  
child: Stack(  
children: [  
ClipRRect(  
borderRadius: BorderRadius.circular(20),  
child: Image.asset("images/salman.jpg",  
height: 220,  
width: 180,  
fit: BoxFit.cover,  
),  
),  
Container(  
padding: EdgeInsets.only(left: 10.0),  
margin: EdgeInsets.only(top: 180),
```

**Project Title:**

**Roll No.**

```
height: 220,  
width: 180,  
decoration:  
BoxDecoration(color:  
Colors.black45,  
borderRadius:  
BorderRadius.only(  
bottomRight: Radius.circular(20),  
bottomLeft: Radius.circular(20))),  
child: Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: [  
Text("Pushpa 2", style: TextStyle(color: Colors.white, fontSize: 25.0, fontWeight:  
FontWeight.bold),),  
Text("Action, Thriller,", style: TextStyle(color: Color.fromARGB(173, 255, 255, 255), fontSize:  
16.0, fontWeight: FontWeight.bold),),  
],  
,  
,  
],  
,  
)  
,  
)
```

**Project Title:**

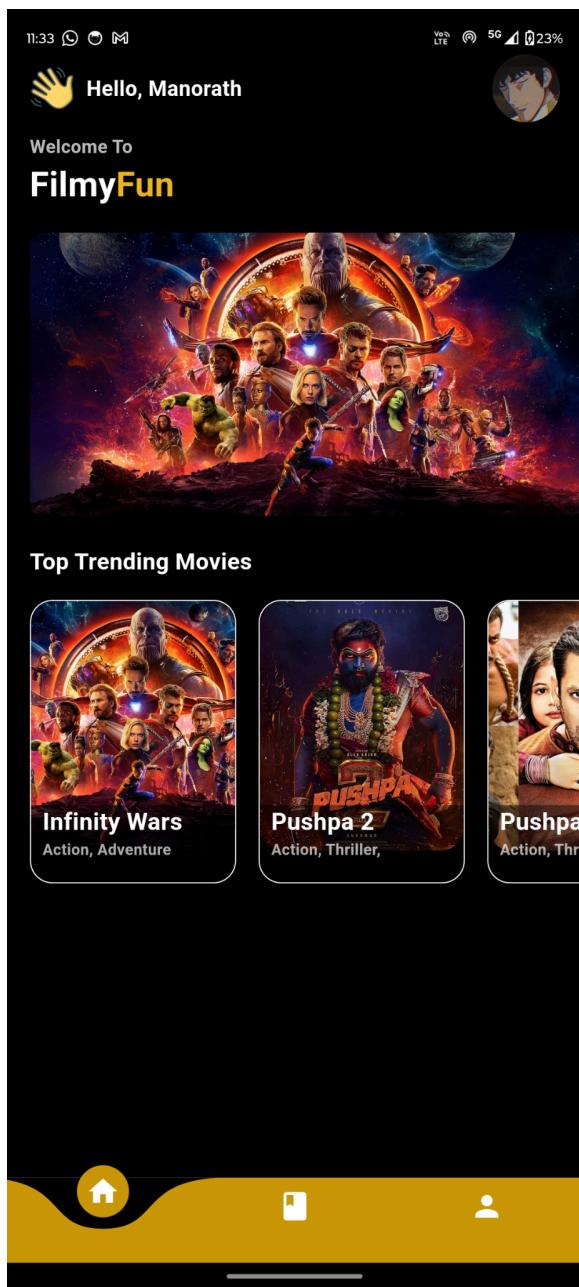
**Roll No.**

]),

));

}

}



**Project Title:**

**Roll No.**

## MAD & PWA Lab Journal

Experiment No.	03
Experiment Title.	To include icons, images, fonts in Flutter app
Roll No.	19
Name	Manorath Ital
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: - 03**

Manorath Ital

D15A/19

**AIM:** - To include icons, images, fonts in Flutter app.

---

### **Theory: -**

#### **Incorporating Icons, Images, and Custom Fonts into FilmyFun App**

**FilmyFun** is a user-friendly movie ticket booking app that allows users to easily select showtimes and reserve tickets. A crucial aspect of creating an engaging and visually appealing app is incorporating various visual elements, such as icons, images, and custom fonts. These elements enhance the overall user experience and help the app stand out.

Like most modern mobile applications, FilmyFun uses resources such as assets and code to provide a dynamic user experience. Assets, such as images, icons, and custom fonts, are integral in delivering a rich and intuitive interface. These resources are bundled with the app and are made available during runtime.

#### **Visual Elements and Their Role in the App**

Visual elements—icons, images, and fonts—play an essential role in app development. These elements are used to improve usability, convey information quickly, and enhance branding.

1. **Enhanced User Experience:** Icons and images make the app visually attractive and easier to navigate.
2. **Information Conveyance:** Icons help reduce text and convey actions intuitively.
3. **Branding:** Custom icons and images align with FilmyFun's branding, making it memorable and recognizable.

#### **Incorporating Icons in FilmyFun**

Icons serve as intuitive visual shortcuts, providing a quick way for users to identify actions and navigate the app. Flutter provides a wide variety of built-in icons through the Icons class, and custom icons can be added via third-party packages like flutter\_launcher\_icons or font\_awesome\_flutter.

#### **Example of adding an icon:**

Icon(

**Project Title:**

**Roll No.**

```
Icons.movie,  
size: 40,  
);
```

### **Adding Images in FilmyFun**

Images are crucial for enhancing the visual experience in FilmyFun. Flutter allows for images to be added from multiple sources:

#### **1. Assets (Stored Locally)**

- To add an image stored locally in the project, place the image in the assets/images/ folder.
- Declare the image path in the pubspec.yaml file.

flutter:

assets:

```
- assets/images/movie_poster.png
```

- Display the image in the app using the Image.asset method:

```
Image.asset('assets/images/movie_poster.png');
```

### **Incorporating Custom Fonts in FilmyFun**

While Flutter's default font is **Roboto**, you may want to add custom fonts to align with your app's branding and overall design aesthetic. Adding custom fonts is simple:

1. **Download the font** and place it in the assets/fonts/ folder of the project.
2. **Declare the font** in the pubspec.yaml file.

flutter:

fonts:

```
- family: CustomFont
```

fonts:

```
- asset: assets/fonts/CustomFont-Regular.ttf
```

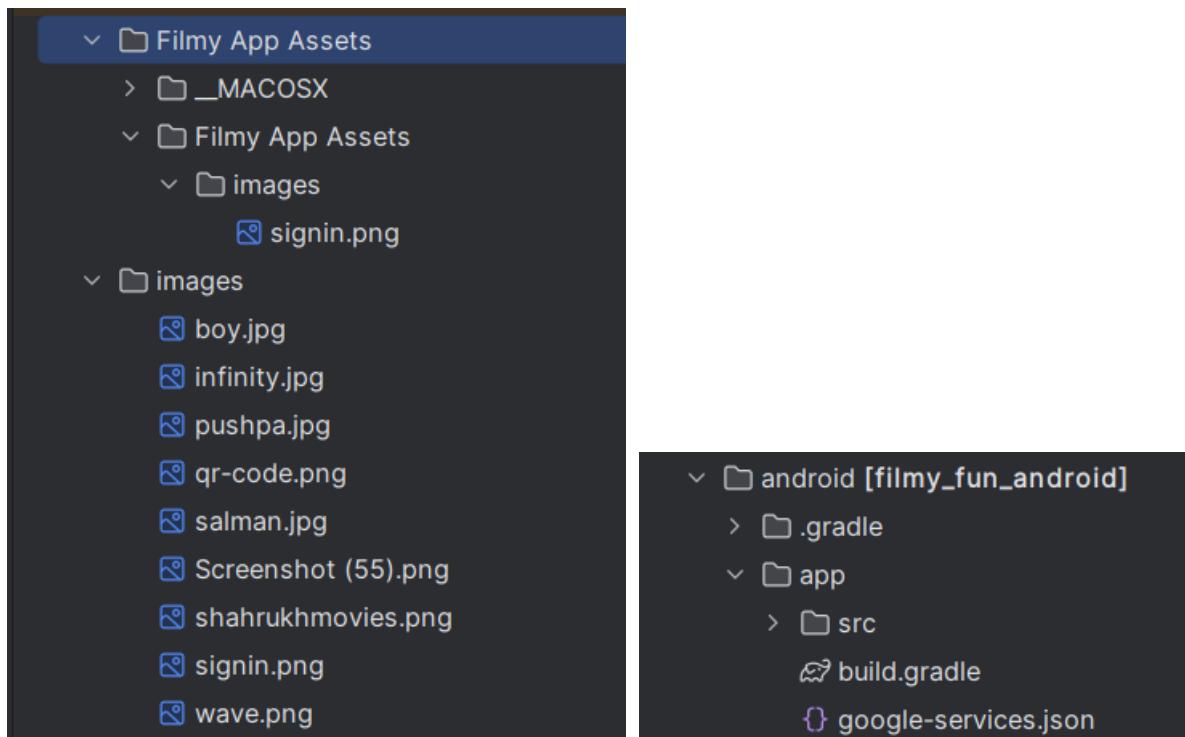
#### **3. Use the custom font** in your app:

```
Text(  
'Welcome to FilmyFun!',  
style: TextStyle(fontFamily: 'CustomFont', fontSize: 24),
```

**Project Title:**

**Roll No.**

);



Pubspec.yaml

```
name: filmy_fun
description: "FilmyFun-Movie Ticket Booking App"
# The following line prevents the package from being accidentally published to
# pub.dev using `flutter pub publish`. This is preferred for private packages.
publish_to: 'none' # Remove this line if you wish to publish to pub.dev

# The following defines the version and build number for your application.
# A version number is three numbers separated by dots, like 1.2.43
# followed by an optional build number separated by a +.
# Both the version and the builder number may be overridden in flutter
# build by specifying --build-name and --build-number, respectively.
# In Android, build-name is used as versionName while build-number used as versionCode.
# Read more about Android versioning at https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString while build-number is used as
# CFBundleVersion.
```

<b>Project Title:</b>	<b>Roll No.</b>
# Read more about iOS versioning at # <a href="https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html">https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html</a> # In Windows, build-name is used as the major, minor, and patch parts # of the product and file versions while build-number is used as the build suffix. version: 1.0.0+1	
environment: sdk: ^3.6.2	
# Dependencies specify other packages that your package needs in order to work. # To automatically upgrade your package dependencies to the latest versions # consider running `flutter pub upgrade --major-versions`. Alternatively, # dependencies can be manually updated by changing the version numbers below to # the latest version available on pub.dev. To see which dependencies have newer # versions available, run `flutter pub outdated`.	
dependencies: flutter: sdk: flutter carousel_slider: ^5.0.0 curved_navigation_bar: ^1.0.3 # Use the latest version intl: ^0.18.0 firebase_auth: ^5.4.2 cloud_firestore: ^5.6.3 random_string: ^2.3.1 firebase_core: ^3.11.0 shared_preferences: flutter_stripe: http:  # The following adds the Cupertino Icons font to your application. # Use with the CupertinoIcons class for iOS style icons. cupertino_icons: ^1.0.8	
dev_dependencies: flutter_test: sdk: flutter	
# The "flutter_lints" package below contains a set of recommended lints to # encourage good coding practices. The lint set provided by the package is # activated in the `analysis_options.yaml` file located at the root of your # package. See that file for information about deactivating specific lint	

**Project Title:**

**Roll No.**

```
# rules and activating additional ones.  
flutter_lints: ^5.0.0  
  
# For information on the generic Dart part of this file, see the  
# following page: https://dart.dev/tools/pub/pubspec  
  
# The following section is specific to Flutter packages.  
flutter:  
  
# The following line ensures that the Material Icons font is  
# included with your application, so that you can use the icons in  
# the material Icons class.  
uses-material-design: true  
  
# To add assets to your application, add an assets section, like this:  
assets:  
  - images/  
    # - images/a_dot_ham.jpeg  
  
# An image asset can refer to one or more resolution-specific "variants", see  
# https://flutter.dev/to/resolution-aware-images  
  
# For details regarding adding assets from package dependencies, see  
# https://flutter.dev/to/asset-from-package  
  
# To add custom fonts to your application, add a fonts section here,  
# in this "flutter" section. Each entry in this list should have a  
# "family" key with the font family name, and a "fonts" key with a  
# list giving the asset and other descriptors for the font. For  
# example:  
# fonts:  
#   - family: Schyler  
#     fonts:  
#       - asset: fonts/Schyler-Regular.ttf  
#       - asset: fonts/Schyler-Italic.ttf  
#         style: italic  
#   - family: Trajan Pro  
#     fonts:  
#       - asset: fonts/TrajanPro.ttf  
#       - asset: fonts/TrajanPro_Bold.ttf  
#         weight: 700  
#  
# For details regarding fonts from package dependencies,  
# see https://flutter.dev/to/font-from-package
```

**Project Title:**

**Roll No.**

Home.dart

```
import 'package:carousel_slider/carousel_slider.dart';
import 'package:filmy_fun/pages/detail_page.dart';
import 'package:flutter/material.dart';

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

  @override
  State<home> createState() => _homeState();
}

class _homeState extends State<home> {
  final List<String> imageUrls = [
    "images/infinity.jpg",
    "images/salman.jpg",
    "images/shahrukhmovies.png",
  ];
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.black,
      body: Container(
        margin: EdgeInsets.only(top: 40.0, left: 20.0),
        child:
          Column(crossAxisAlignment: CrossAxisAlignment.start, children: [
            Row(
              children: [
                Image.asset(
                  "images/wave.png",
                  height: 40,
                  width: 40,
                  fit: BoxFit.cover,
                ),
                SizedBox(
                  width: 10.0,
                ),
                Text(
                  "Hello, Manorath",
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 22.0,
                    fontWeight: FontWeight.bold),
                ),
              ],
            ),
          ],
        ),
      ),
    );
  }
}
```

**Project Title:**

**Roll No.**

```
Spacer(),
Padding(
  padding: const EdgeInsets.only(right: 20.0),
  child: ClipRRect(
    borderRadius: BorderRadius.circular(60),
    child: Image.asset(
      "images/Screenshot (55).png",
      height: 60,
      width: 60,
      fit: BoxFit.cover,
    ),
  ),
),
],
),
SizedBox(height: 10.0),
Text(
  "Welcome To",
  style: TextStyle(
    color: Color.fromARGB(186, 255, 255, 255),
    fontSize: 19.0,
    fontWeight: FontWeight.bold),
),
Row(
  children: [
    Text(
      "Filmy",
      style: TextStyle(
        color: Colors.white,
        fontSize: 36.0,
        fontWeight: FontWeight.bold),
    ),
    Text(
      "Fun",
      style: TextStyle(
        color: Color(0xffedb41d),
        fontSize: 36.0,
        fontWeight: FontWeight.bold),
    )
  ],
),
SizedBox(
  height: 20.0,
),
```

**Project Title:**

**Roll No.**

```
Center(
  child: CarouselSlider(
    items: imageUrl.map((url) {
      return Builder(builder: (context) {
        return Container(
          width: MediaQuery.of(context)
            .size
            .width, // Fixed incorrect comma
          child: ClipRRect(
            borderRadius: BorderRadius.circular(10),
            child: Image.asset(url, fit: BoxFit.cover),
          ),
        );
      });
    }).toList(),
    options: CarouselOptions(
      height: 250,
      autoPlay: false,
      enlargeCenterPage: true,
      aspectRatio: 16 / 9,
      viewportFraction: 1.0), // Fixed missing .toList()
  )),  
SizedBox(
  height: 25.0,
),  
Text(
  "Top Trending Movies",
  style: TextStyle(
    color: Colors.white,
    fontSize: 24.0,
    fontWeight: FontWeight.bold),
),  
SizedBox(
  height: 20.0,
),  
Container(
  height: 250,
  child: ListView(
    scrollDirection: Axis.horizontal,
    children: [
      GestureDetector(
        onTap: () {
          Navigator.push(
            context,
```

**Project Title:**

**Roll No.**

```
MaterialPageRoute(  
    builder: (context) => DetailPage(  
        image: "images/infinity.jpg",  
        name: "Infinity Wars",  
        shortdetail: "Action, Adventure",  
        moviedetail:  
            "Avengers: Infinity War is a 2018 American superhero film based on the Marvel  
        Comics superhero team the Avengers Produced by Marvel Studios and distributed by Walt Disney  
        Studios Motion Pictures, it is the sequel to The Avengers (2012) and Avengers: Age of Ultron (2015), and  
        the 19th film in the Marvel Cinematic Universe (MCU)",  
        price: '50',  
    )),  
,  
child: Container(  
    decoration: BoxDecoration(  
        border: Border.all(color: Colors.white),  
        borderRadius: BorderRadius.circular(20)),  
    child: Stack(  
        children: [  
            ClipRRect(  
                borderRadius: BorderRadius.circular(20),  
                child: Image.asset(  
                    "images/infinity.jpg",  
                    height: 220,  
                    width: 180,  
                    fit: BoxFit.cover,  
                ),  
            ),  
        ],  
        Container(  
            padding: EdgeInsets.only(left: 10.0),  
            margin: EdgeInsets.only(top: 180),  
            height: 220,  
            width: 180,  
            decoration: BoxDecoration(  
                color: Colors.black45,  
                borderRadius: BorderRadius.only(  
                    bottomRight: Radius.circular(20),  
                    bottomLeft: Radius.circular(20))),  
            child: Column(  
                crossAxisAlignment: CrossAxisAlignment.start,  
                children: [  
                    Text(  
                        "Infinity Wars",  
                        style: TextStyle(  
                            color: Colors.white  
                        ),  
                ],  
            ),  
        ),  
    ),  
);
```

**Project Title:** \_\_\_\_\_ **Roll No.** \_\_\_\_\_

```
        color: Colors.white,  
        fontSize: 25.0,  
        fontWeight: FontWeight.bold),  
,  
Text(  
    "Action, Adventure",  
    style: TextStyle(  
        color: Color.fromARGB(173, 255, 255, 255),  
        fontSize: 16.0,  
        fontWeight: FontWeight.bold),  
,  
],  
,  
,  
],  
,  
,  
,  
),  
),  
),  
SizedBox(width: 20.0),  
Container(  
decoration: BoxDecoration(  
    border: Border.all(color: Colors.white),  
    borderRadius: BorderRadius.circular(20)),  
child: Stack(  
children: [  
ClipRRect(  
    borderRadius: BorderRadius.circular(20),  
    child: Image.asset(  
        "images/pushpa.jpg",  
        height: 220,  
        width: 180,  
        fit: BoxFit.cover,  
,  
),  
),  
Container(  
padding: EdgeInsets.only(left: 10.0),  
margin: EdgeInsets.only(top: 180),  
height: 220,  
width: 180,  
decoration: BoxDecoration(  
    color: Colors.black45,  
    borderRadius: BorderRadius.only(  
        bottomRight: Radius.circular(20),  
        bottomLeft: Radius.circular(20))),
```

**Project Title:**

**Roll No.**

```
child: Column(  
    crossAxisAlignment: CrossAxisAlignment.start,  
    children: [  
        Text(  
            "Pushpa 2",  
            style: TextStyle(  
                color: Colors.white,  
                fontSize: 25.0,  
                fontWeight: FontWeight.bold),  
        ),  
        Text(  
            "Action, Thriller,",  
            style: TextStyle(  
                color: Color.fromARGB(173, 255, 255, 255),  
                fontSize: 16.0,  
                fontWeight: FontWeight.bold),  
        ),  
    ],  
),  
],  
),  
],  
),  
],  
),  
),  
SizedBox(width: 20.0),  
Container(  
    decoration: BoxDecoration(  
        border: Border.all(color: Colors.white),  
        borderRadius: BorderRadius.circular(20)),  
    child: Stack(  
        children: [  
            ClipRRect(  
                borderRadius: BorderRadius.circular(20),  
                child: Image.asset(  
                    "images/salman.jpg",  
                    height: 220,  
                    width: 180,  
                    fit: BoxFit.cover,  
                ),  
            ),  
        ],  
    ),  
    Container(  
        padding: EdgeInsets.only(left: 10.0),  
        margin: EdgeInsets.only(top: 180),  
        height: 220,  
        width: 180,
```

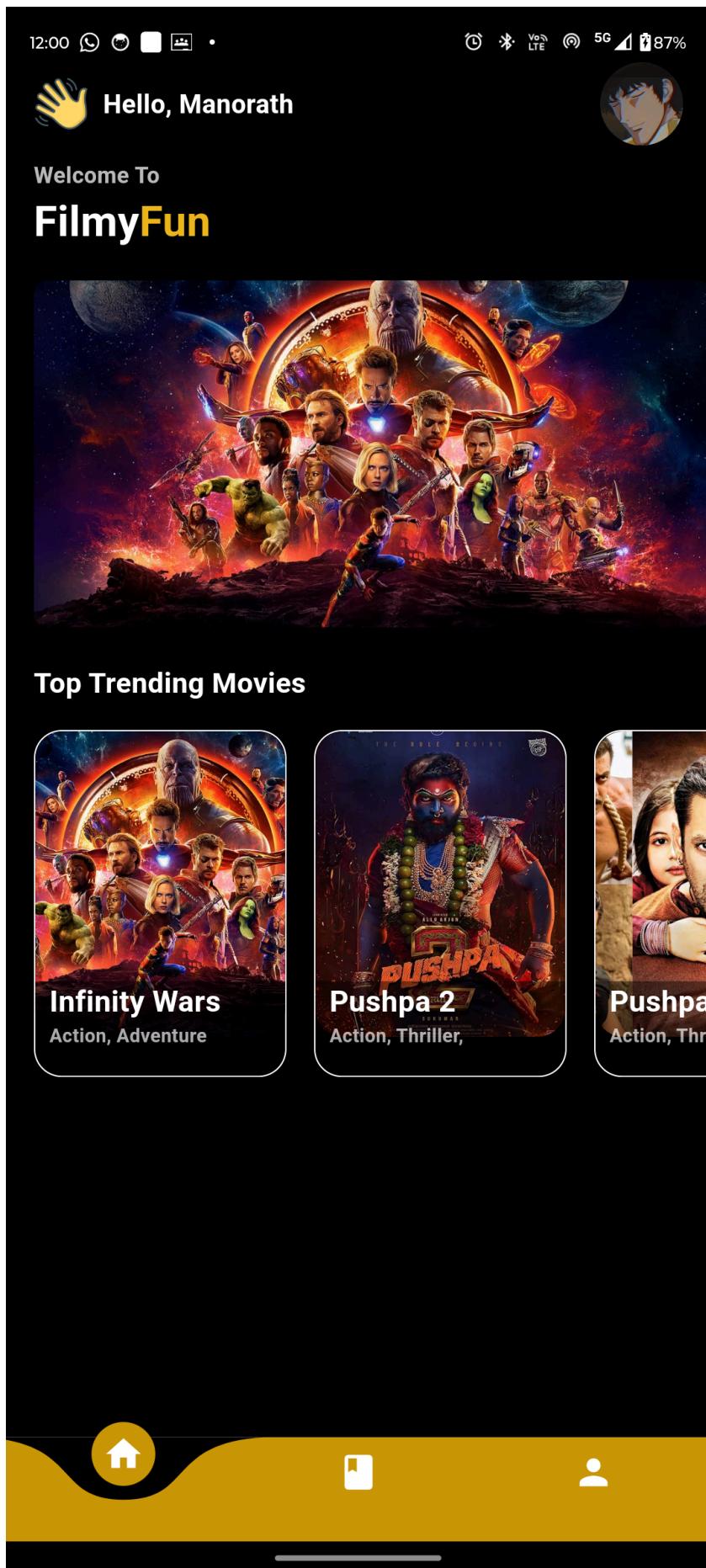
**Project Title:**

**Roll No.**

```
decoration: BoxDecoration(
    color: Colors.black45,
    borderRadius: BorderRadius.only(
        bottomRight: Radius.circular(20),
        bottomLeft: Radius.circular(20))),
child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
        Text(
            "Pushpa 2",
            style: TextStyle(
                color: Colors.white,
                fontSize: 25.0,
                fontWeight: FontWeight.bold),
        ),
        Text(
            "Action, Thriller",
            style: TextStyle(
                color: Color.fromARGB(173, 255, 255, 255),
                fontSize: 16.0,
                fontWeight: FontWeight.bold),
        ),
        ],
    ),
),
],
),
],
),
],
),
],
),
],
),
],
),
]);
));
}
}
```

Project Title:

Roll No.

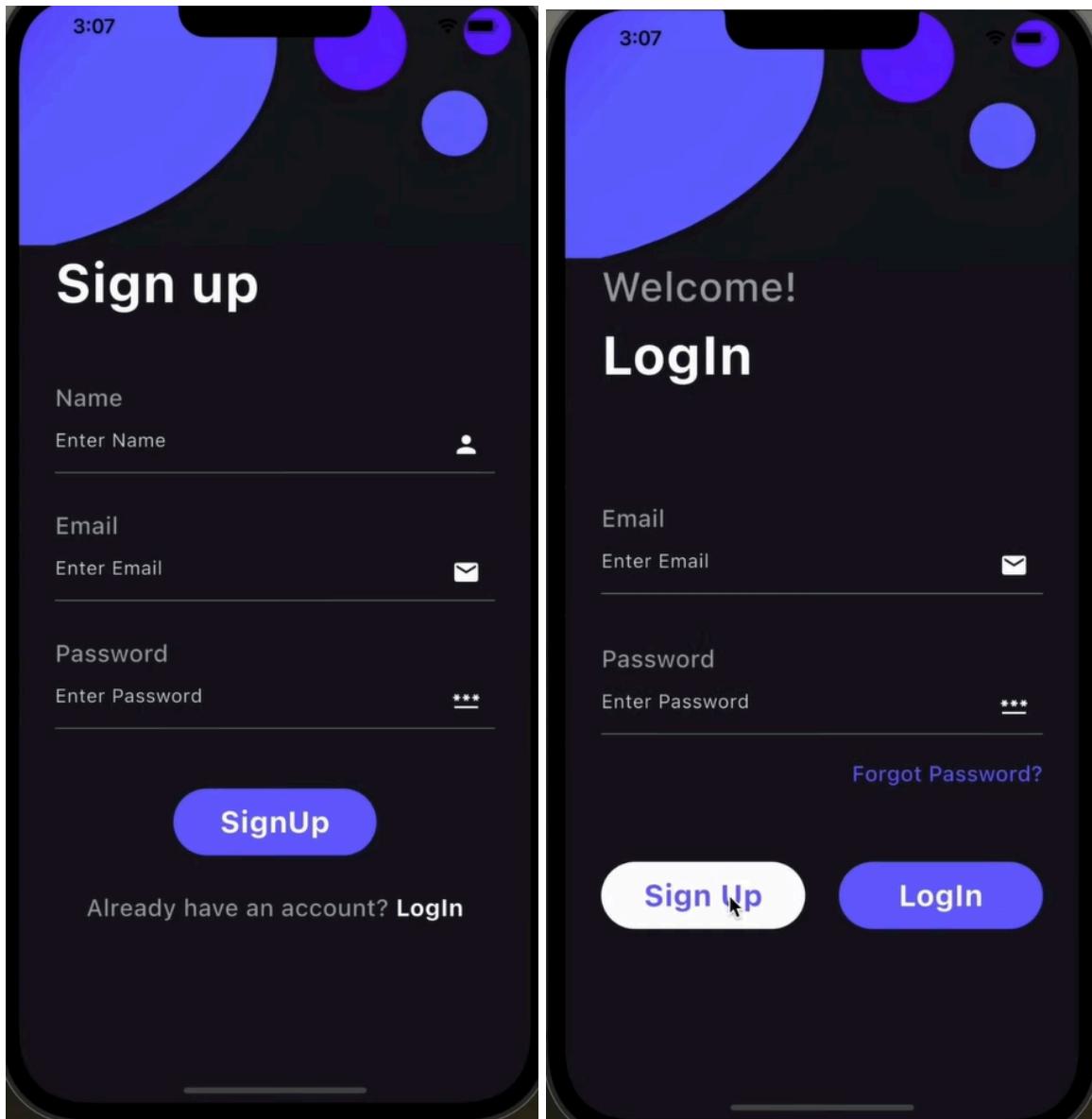


**Project Title:**

**Roll No.**

**ICONS:**

```
buildInputField("Name", "Enter Name", Icons.person, namecontroller),  
buildInputField("Email", "Enter Email", Icons.email, mailcontroller),  
buildInputField("Password", "Enter Password", Icons.lock, passwordcontroller, obscureText: true),  
const SizedBox(height: 30.0),
```



## Conclusion

By incorporating icons, images, and custom fonts into the FilmyFun app, you can create an intuitive, visually appealing, and engaging user experience. These visual elements contribute to the overall

**Project Title:**

**Roll No.**

aesthetic, enhance usability, and ensure your app stands out in a competitive market. Whether displaying movie posters, providing easy navigation with icons, or using custom fonts for unique branding, these resources will elevate the look and feel of your app.

**Project Title:**

**Roll No.**

## MAD & PWA Lab Journal

Experiment No.	04
Experiment Title.	To create an interactive Form using form widget
Roll No.	19
Name	Manorath Ital
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

**Project Title:**

**Roll No.**

**Aim:** To create interactive form

SignUp page

Code:

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

import 'package:filmy_fun/pages/login.dart';

class Signup extends StatefulWidget {

const Signup({super.key});

@Override

State<Signup> createState() => _SignupState();

}

class _SignupState extends State<Signup> {

@Override

Widget build(BuildContext context) {

return Scaffold(


backgroundColor: Colors.black,


body: Padding(


padding: const EdgeInsets.symmetric(horizontal: 20.0), // Adds padding to align left


child: Column(


crossAxisAlignment: CrossAxisAlignment.start, // Aligns content to left


children: [


Image.asset("images/signin.png"),


const SizedBox(height: 20.0), // Adds spacing


const Text(
```

**Project Title:**

**Roll No.**

```
"Welcome!",  
style: TextStyle(  
color: Color.fromARGB(157, 255, 255, 255),  
fontSize: 34.0,  
fontWeight: FontWeight.w500,  
,  
,  
const Text(  
"SignUp",  
style: TextStyle(  
color: Colors.white,  
fontSize: 45.0,  
fontWeight: FontWeight.bold,  
,  
,  
const SizedBox(height: 30.0), // Adjust spacing  
const Text(  
"Name",  
style: TextStyle(  
color: Colors.white,  
fontSize: 20.0,  
fontWeight: FontWeight.w500,  
,  
,
```

**Project Title:**

**Roll No.**

```
TextField(  
  decoration: InputDecoration(  
    hintText: "Enter Name",  
    hintStyle: TextStyle(  
      color: Colors.white,  
    ),  
    suffixIcon: Icon(Icons.password,  
      color: Colors.white,),  
  ),  
,  
  const SizedBox(height: 50.0), // Adjust spacing  
  const Text(  
    "Email",  
    style: TextStyle(  
      color: Colors.white,  
      fontSize: 20.0,  
      fontWeight: FontWeight.w500,  
    ),  
,  
  TextField(  
    decoration: InputDecoration(  
      hintText: "Enter Email",  
      hintStyle: TextStyle(  
        color: Colors.white,
```

**Project Title:**

**Roll No.**

```
),  
suffixIcon: Icon(Icons.email, color: Colors.white,),  
,  
,  
const SizedBox(height: 50.0), // Adjust spacing  
const Text(  
"Password",  
style: TextStyle(  
color: Colors.white,  
fontSize: 20.0,  
fontWeight: FontWeight.w500,  
,  
,  
TextField(  
decoration: InputDecoration(  
hintText: "Enter Password",  
hintStyle: TextStyle(  
color: Colors.white,  
,  
suffixIcon: Icon(Icons.password,  
color: Colors.white,),  
,  
,  
const SizedBox(height: 50.0), // Adjust spacing
```

**Project Title:**

**Roll No.**

```
const Text(  
  "Confirm Password",  
  style: TextStyle(  
    color: Colors.white,  
    fontSize: 20.0,  
    fontWeight: FontWeight.w500,  
,  
,  
  ),  
  ),  
  
TextField(  
  decoration: InputDecoration(  
    hintText: "Enter Password",  
    hintStyle: TextStyle(  
      color: Colors.white,  
,  
    ),  
    suffixIcon: Icon(Icons.password,  
    color: Colors.white,),  
,  
,  
  ),  
  
SizedBox(height: 50.0,),  
  
Center(  
  child: Column(  
    children: [  
      Container(  
        width: 170,
```

**Project Title:**

**Roll No.**

```
padding: EdgeInsets.all(10),  
decoration: BoxDecoration(  
color: Colors.white,  
borderRadius: BorderRadius.circular(30),  
,  
child: Text(  
"SignUp",  
style: TextStyle(  
color: Color(0xff6b63ff),  
fontSize: 25.0,  
fontWeight: FontWeight.bold,  
,  
textAlign: TextAlign.center,  
,  
,  
SizedBox(height: 20.0), // Space between button and text  
Row(  
mainAxisAlignment: MainAxisAlignment.center,  
children: [  
Text(  
"Already have an account?",  
style: TextStyle(  
color: Color.fromARGB(175, 255, 255, 255),  
fontSize: 18.0,
```

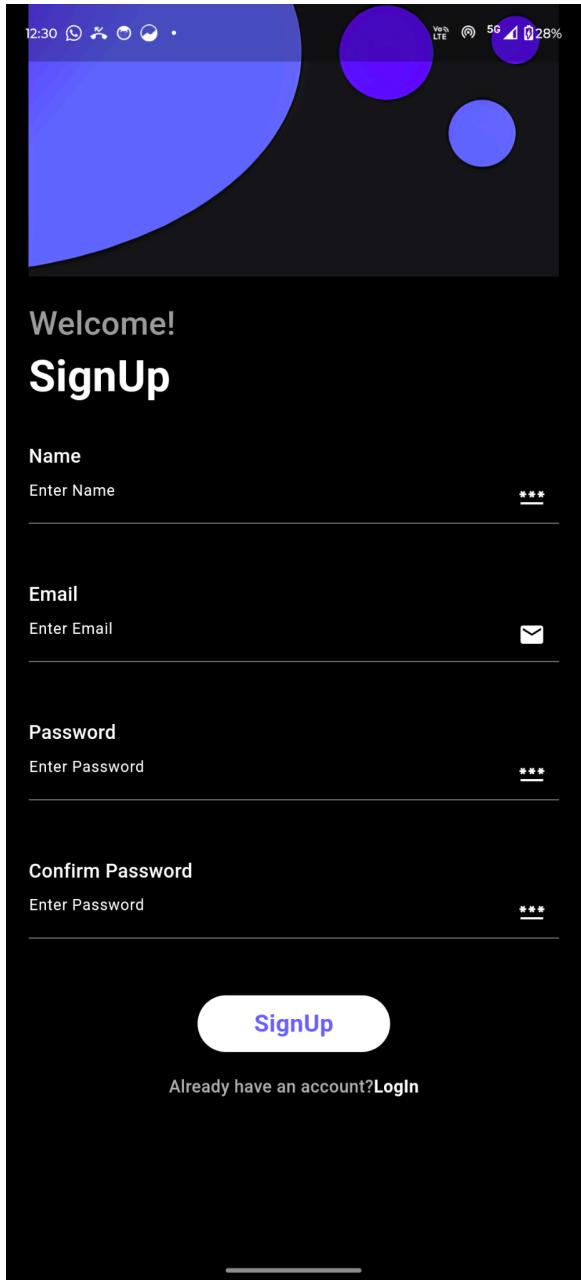
**Project Title:**

**Roll No.**

```
fontWeight: FontWeight.w500,  
,  
,  
GestureDetector(  
onTap:() {Navigator.push(context, MaterialPageRoute(builder: (context)=> Login()));},  
child: Text(  
"Login",  
style: TextStyle(  
color: Colors.white,  
fontSize: 18.0,  
fontWeight: FontWeight.bold,  
,),],),],),],),});})}
```

**Project Title:**

**Roll No.**



Login page

Code:

**Project Title:**

**Roll No.**

```
import 'package:filmy_fun/pages/bottomnav.dart';

import 'package:flutter/material.dart';

import 'package:filmy_fun/pages/signup.dart';

class Login extends StatefulWidget {

const Login({super.key});

@Override

State<Login> createState() => _LoginState();

}

class _LoginState extends State<Login> {

@Override

Widget build(BuildContext context) {

return Scaffold(


backgroundColor: Colors.black,


body: Padding(


padding: const EdgeInsets.symmetric(horizontal: 20.0), // Adds padding to align left


child: Column(


crossAxisAlignment: CrossAxisAlignment.start, // Aligns content to left


children: [


Image.asset("images/signin.png"),


const SizedBox(height: 20.0), // Adds spacing


const Text(


"Welcome!",


style: TextStyle(


color: Color.fromARGB(157, 255, 255, 255),
```

**Project Title:**

**Roll No.**

```
fontSize: 34.0,  
fontWeight: FontWeight.w500,  
,  
,  
const Text(  
"Login",  
style: TextStyle(  
color: Colors.white,  
fontSize: 45.0,  
fontWeight: FontWeight.bold,  
,  
,  
const SizedBox(height: 50.0), // Adjust spacing  
const Text(  
"Email",  
style: TextStyle(  
color: Colors.white,  
fontSize: 20.0,  
fontWeight: FontWeight.w500,  
,  
,  
TextField(  
decoration: InputDecoration(  
hintText: "Enter Email",
```

**Project Title:**

**Roll No.**

```
hintStyle: TextStyle(  
  color: Colors.white,  
,  
suffixIcon: Icon(Icons.email, color: Colors.white,),  
,  
),  
const SizedBox(height: 50.0), // Adjust spacing  
const Text(  
  "Password",  
  style: TextStyle(  
    color: Colors.white,  
    fontSize: 20.0,  
    fontWeight: FontWeight.w500,  
,  
,  
  ),  
  decoration: InputDecoration(  
    hintText: "Enter Password",  
    hintStyle: TextStyle(  
      color: Colors.white,  
,  
    suffixIcon: Icon(Icons.password,  
    color: Colors.white,),  
,
```

**Project Title:**

**Roll No.**

```
),  
const SizedBox(height: 20.0),  
Row(  
  mainAxisAlignment: MainAxisAlignment.end,  
  children: [  
    Text(  
      "Forgot Password?",  
      style: TextStyle(  
        color: Colors.white,  
        fontSize: 18.0,  
        fontWeight: FontWeight.w500,  
      ),  
    ),  
  ],),  
SizedBox(height: 50.0,),  
Row(  
  mainAxisAlignment: MainAxisAlignment.spaceBetween,  
  children: [  
    Row(  
      mainAxisAlignment: MainAxisAlignment.center,  
      children: [  
        GestureDetector(  
          onTap:() {Navigator.push(context, MaterialPageRoute(builder: (context)=> Signup()));},  
          child: Container(  
           
```

**Project Title:**

**Roll No.**

```
width: 170,  
padding: EdgeInsets.all(10),  
decoration: BoxDecoration(color: Colors.white,  
borderRadius: BorderRadius.circular(30)),  
child: Text("SignUp",style: TextStyle(color: Color(0xff6b63ff),  
fontSize: 25.0,  
fontWeight: FontWeight.bold),  
textAlign: TextAlign.center,  
),  
,  
,  
],  
,  
Row(  
mainAxisAlignment: MainAxisAlignment.center,  
children: [  
GestureDetector(  
onTap: () {Navigator.push(context, MaterialPageRoute(builder: (context)=> Bottomnav()));},  
child: Container(  
width: 170,  
padding: EdgeInsets.all(10),  
decoration: BoxDecoration(  
color: Color(0xff6b63ff),  
borderRadius: BorderRadius.circular(30)),
```

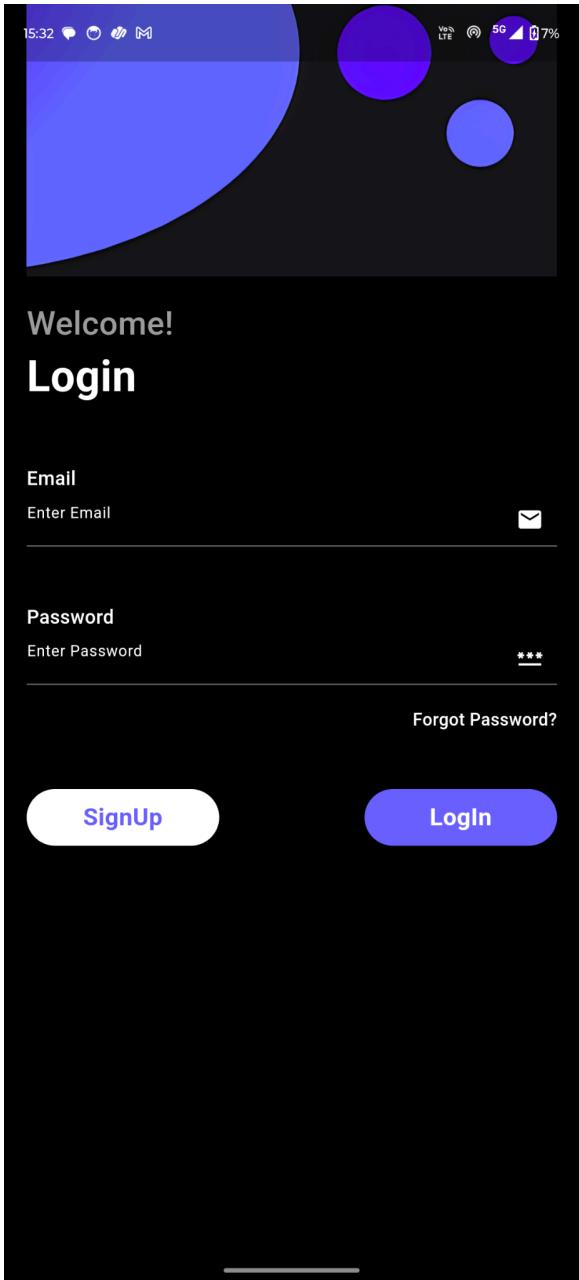
**Project Title:**

**Roll No.**

```
child: Text("Login",style: TextStyle(color: Colors.white,  
fontSize: 25.0,  
fontWeight: FontWeight.bold),  
textAlign: TextAlign.center,  
),  
,  
,  
],  
)  
],  
)  
],  
,  
,  
);  
}  
}
```

**Project Title:**

**Roll No.**



**Project Title:**

**Roll No.**

## MAD & PWA Lab Journal

Experiment No.	05
Experiment Title.	To apply navigation, routing and gestures in Flutter App
Roll No.	19
Name	Manorath Ital
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

**Project Title:**

**Roll No.**

## EXPERIMENT NO: - 05

MANORATH ITAL

D15A/19

**AIM:** - To apply navigation, routing and gestures in Flutter App.

**Theory:** -

In Flutter, the screens and pages are known as routes, and these routes are just a widget. In Android,

a route is similar to an Activity.

In any mobile app, navigating to different pages defines the workflow of the application, and the

way to handle the navigation is known as routing. Flutter provides a basic routing class

MaterialPageRoute and two methods Navigator.push() and Navigator.pop() that shows how to

navigate between two routes. The following steps are required to start navigation in your

application.

Gestures enable the app to respond to user interactions, making the application more dynamic and

responsive.

➤ Navigation and Routing in Flutter

Navigation is the process of moving between different screens or pages in an app.  
Flutter

provides a simple and effective way to handle this through the use of the Navigator widget and routes.

## 1. Using Navigator Widget

The Navigator widget manages a stack of routes, allowing for pushing and popping routes on the stack.

- Pushing a Route: To navigate to a new screen, use Navigator.push().
- Popping a Route: To go back to the previous screen, use Navigator.pop().

```
ElevatedButton(
```

```
    onPressed: () {
```

```
        Navigator.push(
```

```
            context,
```

```
            MaterialPageRoute(builder: (context) => SecondScreen()),
```

```
        );},
```

```
);
```

## 2. Named Routes

Flutter also allows the use of named routes to navigate, which can make the routing process

cleaner, especially in larger applications.

```
MaterialApp(
```

```
    initialRoute: '/',
```

```
    routes: {
```

```
        '/': (context) => HomeScreen(),
```

```
        '/second': (context) => SecondScreen(),
```

```
    },
```

```
);
```

Navigate to the route using Navigator.pushNamed()

```
Navigator.pushNamed(context, '/second');
```

## Handling Gestures in Flutter

Gestures refer to user interactions with the app, such as taps, swipes, pinches, and drags. Flutter

provides several widgets and gesture detectors to handle these interactions.

### Tap Gestures

The most common gesture is the tap, which can be handled using the GestureDetector widget or

specific buttons like InkWell or ElevatedButton.

### Long Press Gesture

For long press gestures, Flutter provides the onLongPress callback in GestureDetector or InkWell.

### Swipe and Drag Gestures

Flutter also provides swipe and drag gesture handling. The onHorizontalDragUpdate and onVerticalDragUpdate callbacks are used for dragging gestures.

**Code:-****main.dart**

```
import 'package:filmy_fun/pages/detail_page.dart';
import 'package:filmy_fun/service/constant.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:filmy_fun/pages/login.dart';
import 'package:filmy_fun/pages/signup.dart';
import 'package:filmy_fun/pages/home.dart';
import 'package:filmy_fun/pages/bottomnav.dart';
import 'package:filmy_fun/pages/booking.dart';
import 'package:flutter_stripe/flutter_stripe.dart';

void main()async {
    WidgetsFlutterBinding.ensureInitialized();
    Stripe.publishableKey= publishedKey;
    await Firebase.initializeApp();
    runApp(const MyApp());
}

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

    // This widget is the root of your application.
    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            title: 'FilmyFun',
            debugShowCheckedModeBanner: false,
            theme: ThemeData(
                // This is the theme of your application.
                //
                // TRY THIS: Try running your application with "flutter run". You'll see
                // the application has a purple toolbar. Then, without quitting the app,

```

```
// try changing the seedColor in the colorScheme below to Colors.green
// and then invoke "hot reload" (save your changes or press the "hot
// reload" button in a Flutter-supported IDE, or press "r" if you used
// the command line to start the app).
//
// Notice that the counter didn't reset back to zero; the application
// state is not lost during the reload. To reset the state, use hot
// restart instead.
//
// This works for code too, not just values: Most code changes can be
// tested with just a hot reload.
colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
useMaterial3: true,
),
home: Signup(),
);
}
}

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

// This widget is the home page of your application. It is stateful, meaning
// that it has a State object (defined below) that contains fields that affect
// how it looks.

// This class is the configuration for the state. It holds the values (in this
// case the title) provided by the parent (in this case the App widget) and
// used by the build method of the State. Fields in a Widget subclass are
// always marked "final".

final String title;

@Override
State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
int _counter = 0;
```

```
void _incrementCounter() {
    setState(() {
        // This call to setState tells the Flutter framework that something has
        // changed in this State, which causes it to rerun the build method below
        // so that the display can reflect the updated values. If we changed
        // _counter without calling setState(), then the build method would not be
        // called again, and so nothing would appear to happen.
        _counter++;
    });
}

@Override
Widget build(BuildContext context) {
    // This method is rerun every time setState is called, for instance as done
    // by the _incrementCounter method above.
    //
    // The Flutter framework has been optimized to make rerunning build
    methods
    // fast, so that you can just rebuild anything that needs updating rather
    // than having to individually change instances of widgets.
    return Scaffold(
        appBar: AppBar(
            // TRY THIS: Try changing the color here to a specific color (to
            // Colors.amber, perhaps?) and trigger a hot reload to see the AppBar
            // change color while the other colors stay the same.
            backgroundColor: Theme.of(context).colorScheme.inversePrimary,
            // Here we take the value from the MyHomePage object that was created by
            // the App.build method, and use it to set our appbar title.
            title: Text(widget.title),
        ),
        body: Center(
            // Center is a layout widget. It takes a single child and positions it
            // in the middle of the parent.
            child: Column(
                // Column is also a layout widget. It takes a list of children and
                // arranges them vertically. By default, it sizes itself to fit its
                // children horizontally, and tries to be as tall as its parent.
                //
                // Column has various properties to control how it sizes itself and
                // how it positions its children. Here we use mainAxisAlignment to
```

```

// center the children vertically; the main axis here is the vertical
// axis because Columns are vertical (the cross axis would be
// horizontal).
//
// TRY THIS: Invoke "debug painting" (choose the "Toggle Debug Paint"
// action in the IDE, or press "p" in the console), to see the
// wireframe for each widget.
mainAxisAlignment: MainAxisAlignment.center,
children: <Widget>[
  const Text(
    'You have pushed the button this many times:',
  ),
  Text(
    '$_counter',
    style: Theme.of(context).textTheme.headlineMedium,
  ),
],
),
),
),
),
),
floatingActionButton: FloatingActionButton(
onPressed: _incrementCounter,
tooltip: 'Increment',
child: const Icon(Icons.add),
), // This trailing comma makes auto-formatting nicer for build methods.
);
}
}

```

## Signup.dart

```

import 'package:filmy_fun/service/database.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:filmy_fun/pages/login.dart';
import 'package:filmy_fun/service/shared_pref.dart';
import 'package:random_string/random_string.dart';
import 'bottomnav.dart';

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

```

**Project Title:**

**Roll No.**

```
@override
State<Signup> createState() => _SignupState();
}

class _SignupState extends State<Signup> {
String email = "", password = "", name = "";
TextEditingController namecontroller = TextEditingController();
TextEditingController passwordcontroller = TextEditingController();
TextEditingController mailcontroller = TextEditingController();

registration() async {
    if (passwordcontroller.text.isNotEmpty &&
        namecontroller.text.isNotEmpty &&
        mailcontroller.text.isNotEmpty) {
        try {
            UserCredential userCredential = await FirebaseAuth.instance
                .createUserWithEmailAndPassword(
                    email: mailcontroller.text, password: passwordcontroller.text);
            String id = randomAlphaNumeric(10);
            Map<String, dynamic> userInfoMap = {
                "Name": namecontroller.text,
                "Email": mailcontroller.text,
                "Id": id,
                "Image": ""
            };
            await SharedpreferenceHelper().saveUserDislayName(namecontroller.text);
            await SharedpreferenceHelper().saveUserEmail(mailcontroller.text);
            await SharedpreferenceHelper().saveUserID(id);
            await SharedpreferenceHelper().saveUserImage("");
            await DatabaseMethods().addUserDetails(userInfoMap, id);
            ScaffoldMessenger.of(context).showSnackBar(SnackBar(
                backgroundColor: Colors.green,
                content: Text(
                    "Registered Successfully!",
                    style: TextStyle(
                        color: Colors.white,
                        fontWeight: FontWeight.bold,
                        fontSize: 20.0),
                )));
        }
    }
}
```

**Project Title:**

**Roll No.**

```
Navigator.pushReplacement(  
    context, MaterialPageRoute(builder: (context) => Bottomnav()));  
} on FirebaseAuthException catch (e) {  
    if (e.code == 'weak-password') {  
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(  
            backgroundColor: Colors.orangeAccent,  
            content: Text(  
                "Password Provided is too Weak",  
                style: TextStyle(fontSize: 18.0),  
            )));
    } else if (e.code == "email-already-in-use") {  
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(  
            backgroundColor: Colors.orangeAccent,  
            content: Text(  
                "Account Already exists",  
                style: TextStyle(fontSize: 18.0),  
            )));
    }
}
}  
}  
  
@override  
Widget build(BuildContext context) {  
    return Scaffold(  
        backgroundColor: Colors.black,  
        body: SingleChildScrollView(  
            child: Padding(  
                padding: const EdgeInsets.symmetric(horizontal: 20.0),  
                child: Column(  
                    mainAxisAlignment: MainAxisAlignment.start,  
                    children: [  
                        const SizedBox(height: 50.0),  
                        Image.asset("images/signin.png"),  
                        const SizedBox(height: 20.0),  
                        const Text(  
                            "Welcome!",  
                            style: TextStyle(  
                                color: Color.fromARGB(157, 255, 255, 255),  
                                fontSize: 34.0,
```

**Project Title:**

**Roll No.**

```
fontWeight: FontWeight.w500,
),
),
),
const Text(
"SignUp",
style: TextStyle(
color: Colors.white,
fontSize: 45.0,
fontWeight: FontWeight.bold,
),
),
),
const SizedBox(height: 30.0),
buildInputField("Name", "Enter Name", Icons.person, namecontroller),
buildInputField("Email", "Enter Email", Icons.email, mailcontroller),
buildInputField("Password", "Enter Password", Icons.lock,
passwordcontroller, obscureText: true),
const SizedBox(height: 30.0),
Center(
child: Column(
children: [
GestureDetector(
onTap: registration,
child: GestureDetector(
onTap: () {
if (namecontroller.text.isNotEmpty &&
mailcontroller.text.isNotEmpty &&
passwordcontroller.text.isNotEmpty) {
setState(() {
name = namecontroller.text;
email = mailcontroller.text;
password = passwordcontroller.text;
});
registration();
}
}),
),
child: Container(
width: 170,
padding: const EdgeInsets.all(12),
decoration: BoxDecoration(
color: Colors.white,
```

**Project Title:**

**Roll No.**

```
borderRadius: BorderRadius.circular(30),  
),  
child: const Text(  
"SignUp",  
style: TextStyle(  
color: Color(0xff6b63ff),  
fontSize: 25.0,  
fontWeight: FontWeight.bold,  
),  
textAlign: TextAlign.center,  
),  
),  
),  
const SizedBox(height: 20.0),  
Row(  
mainAxisAlignment: MainAxisAlignment.center,  
children: [  
    const Text(  
"Already have an account?",  
style: TextStyle(  
color: Color.fromARGB(175, 255, 255, 255),  
fontSize: 18.0,  
fontWeight: FontWeight.w500,  
),  
),  
GestureDetector(  
onTap: () {  
Navigator.push(  
context,  
MaterialPageRoute(  
builder: (context) => const Login()));  
},  
child: const Text(  
" LogIn",  
style: TextStyle(  
color: Colors.white,  
fontSize: 18.0,  
fontWeight: FontWeight.bold,  
),
```

**Project Title:**

**Roll No.**

```
        ),  
        ),  
    ],  
    ),  
    const SizedBox(height: 30.0),  
    ],  
    ),  
),  
);  
}  
}
```

```
Widget buildInputField(String label, String hint, IconData icon,  
    TextEditingController controller,  
    {bool obscureText = false}) {  
    return Padding(  
        padding: const EdgeInsets.only(bottom: 20.0),  
        child: Column(  
            crossAxisAlignment: CrossAxisAlignment.start,  
            children: [  
                Text(  
                    label,  
                    style: const TextStyle(  
                        color: Colors.white,  
                        fontSize: 20.0,  
                        fontWeight: FontWeight.w500,  
                    ),  
                ),  
                TextField(  
                    controller: controller,  
                    obscureText: obscureText,  
                    cursorColor: Colors.white, // Cursor color set to white  
                    style: const TextStyle(color: Colors.white), // Text color set to white  
                    decoration: InputDecoration(  
                        hintText: hint,  
                        hintStyle: const TextStyle(color: Colors.white), // Hint text color set to  
white
```

```

suffixIcon: Icon(icon, color: Colors.white),
enabledBorder: const UnderlineInputBorder(
borderSide: BorderSide(color: Colors.white), // Underline color
),
focusedBorder: const UnderlineInputBorder(
borderSide: BorderSide(color: Colors.white), // Underline when
focused
),
),
),
],
),
);
);
}
}

```

## Login.dart

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:filmy_fun/pages/bottomnav.dart';
import 'package:filmy_fun/pages/home.dart';
import 'package:filmy_fun/service/database.dart';
import 'package:filmy_fun/service/shared_pref.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:filmy_fun/pages/signup.dart';

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

@Override
State<Login> createState() => _LoginState();
}

class _LoginState extends State<Login> {
String email = "", password = "", myname = "", myid = "", myimage = "";
 TextEditingController passwordcontroller = TextEditingController();
 TextEditingController mailcontroller = TextEditingController();

userLogin() async {

```

**Project Title:**

**Roll No.**

```
try {
    await FirebaseAuth.instance.signInWithEmailAndPassword(
        email: email,
        password: password,
    );
    QuerySnapshot querySnapshot =
        await DatabaseMethods().getUserbyemail(email);
    myname = "${querySnapshot.docs[0]["Name"]}";
    myid = "${querySnapshot.docs[0]["Id"]}";
    myimage = "${querySnapshot.docs[0]["Image"]}";

    await SharedpreferenceHelper().saveUserImage(myimage);
    await SharedpreferenceHelper().saveUserEmail(email);
    await SharedpreferenceHelper().saveUserDislayName(myname);
    await SharedpreferenceHelper().saveUserID(myid);
    Navigator.push(context, MaterialPageRoute(builder: (context) =>
Bottonnav()));
} on FirebaseAuthException catch (e) {
    String errorMessage = "";
    if (e.code == 'user-not-found') {
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
            content: Text("No user found for that email."),
            style: TextStyle(fontSize: 18.0, color: Colors.black),
        ));
    } else if (e.code == 'wrong-password') {
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
            backgroundColor: Colors.white,
            content: Text("Wrong password provided."),
        ));
    }
}
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: Colors.black,
        body: Padding(
            padding: const EdgeInsets.symmetric(horizontal: 20.0),
            child: Column(
```

**Project Title:**

**Roll No.**

```
crossAxisAlignment: CrossAxisAlignment.start,  
children: [  
    Image.asset("images/signin.png"),  
    const SizedBox(height: 20.0),  
    const Text(  
        "Welcome!",  
        style: TextStyle(  
            color: Color.fromARGB(157, 255, 255, 255),  
            fontSize: 34.0,  
            fontWeight: FontWeight.w500,  
        ),  
    ),  
    GestureDetector(  
        onTap: () {  
            if (mailcontroller.text.isNotEmpty &&  
                passwordcontroller.text.isNotEmpty) {  
                setState(() {  
                    email = mailcontroller.text;  
                    password = passwordcontroller.text;  
                    userLogin();  
                });  
            }  
        },  
        child: const Text(  
            "Login",  
            style: TextStyle(  
                color: Colors.white,  
                fontSize: 45.0,  
                fontWeight: FontWeight.bold,  
            ),  
        ),  
    ),  
    const SizedBox(height: 50.0),  
    const Text(  
        "Email",  
        style: TextStyle(  
            color: Colors.white,  
            fontSize: 20.0,  
            fontWeight: FontWeight.w500,  
        ),  
    ),
```

## **Project Title:**

**Roll No.**

```
        ),  
        TextField(  
            controller: mailcontroller,  
            cursorColor: Colors.black,  
            style: const TextStyle(color: Colors.white),  
            decoration: const InputDecoration(  
                hintText: "Enter Email",  
                hintStyle: TextStyle(color: Colors.grey),  
            ),  
        ),  
        const SizedBox(height: 50.0),  
        const Text(  
            "Password",  
            style: TextStyle(  
                color: Colors.white,  
                fontSize: 20.0,  
                fontWeight: FontWeight.w500,  
            ),  
        ),  
        TextField(  
            controller: passwordcontroller,  
            cursorColor: Colors.black,  
            style: const TextStyle(color: Colors.white),  
            obscureText: true,  
            decoration: const InputDecoration(  
                hintText: "Enter Password",  
                hintStyle: TextStyle(color: Colors.grey),  
            ),  
        ),  
        const SizedBox(height: 20.0),  
        Row(  
            mainAxisAlignment: MainAxisAlignment.end,  
            children: [  
                Text(  
                    "Forgot Password?",  
                    style: TextStyle(  
                        color: Colors.white,  
                        fontSize: 18.0,  
                        fontWeight: FontWeight.w500,  
                    ),  
                ),  
            ],  
        ),  
    ),  
);
```

**Project Title:**

**Roll No.**

```
)  
],  
,  
const SizedBox(height: 50.0),  
Row(  
    mainAxisAlignment: MainAxisAlignment.spaceBetween,  
    children: [  
        GestureDetector(  
            onTap: () {  
                Navigator.push(context,  
                    MaterialPageRoute(builder: (context) => Signup()));  
            },  
            child: Container(  
                width: 170,  
                padding: EdgeInsets.all(10),  
                decoration: BoxDecoration(  
                    color: Colors.white,  
                    borderRadius: BorderRadius.circular(30)),  
                child: Text(  
                    "SignUp",  
                    style: TextStyle(  
                        color: Color(0xff6b63ff),  
                        fontSize: 25.0,  
                        fontWeight: FontWeight.bold),  
                    textAlign: TextAlign.center,  
                ),  
                ),  
                ),  
                ),  
                GestureDetector(  
                    onTap: () {  
                        Navigator.push(  
                            context,  
                            MaterialPageRoute(  
                                builder: (context) => const Bottomnav()));  
                    },  
                    child: Container(  
                        width: 170,  
                        padding: EdgeInsets.all(10),  
                        decoration: BoxDecoration(  
                            color: Color(0xff6b63ff),
```

**Project Title:**

**Roll No.**

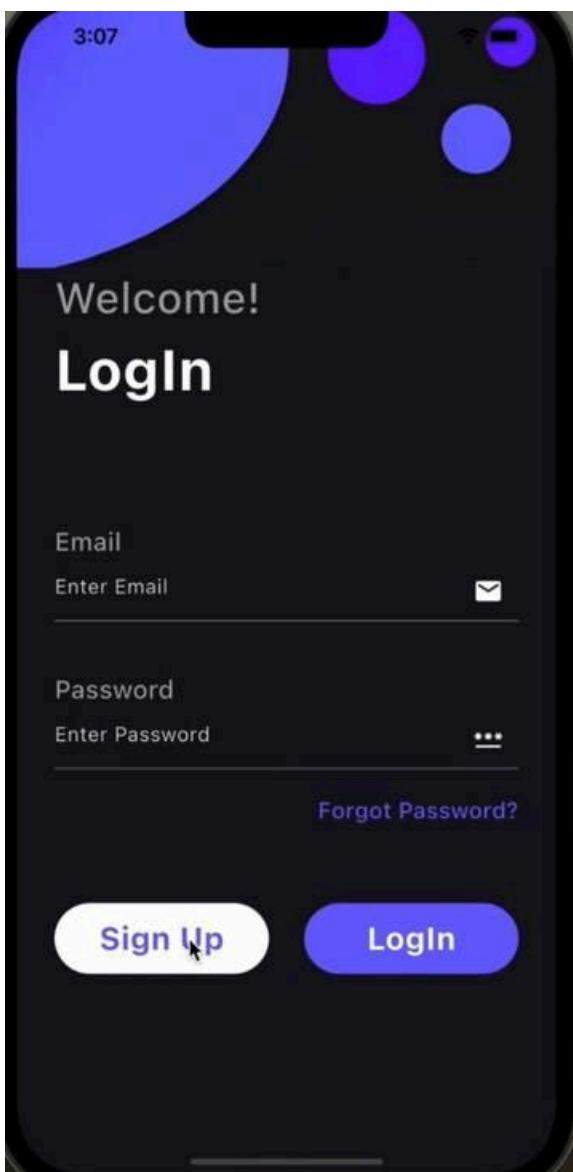
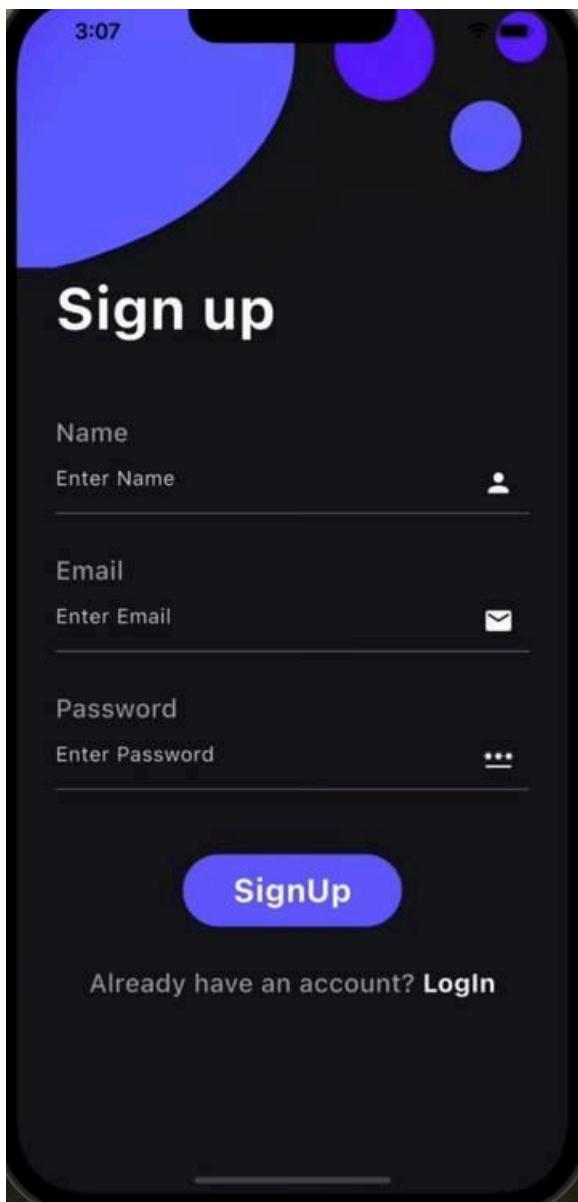
```
        borderRadius: BorderRadius.circular(30)),  
        child: Text(  
            "LogIn",  
            style: TextStyle(  
                color: Colors.white,  
                fontSize: 25.0,  
                fontWeight: FontWeight.bold),  
            textAlign: TextAlign.center,  
        ),  
    ),  
    ),  
    ),  
    ],  
),  
),  
),  
);  
}  
}
```

**OUTPUT:**

After clicking on Already have an account? It navigates to login

Project Title:

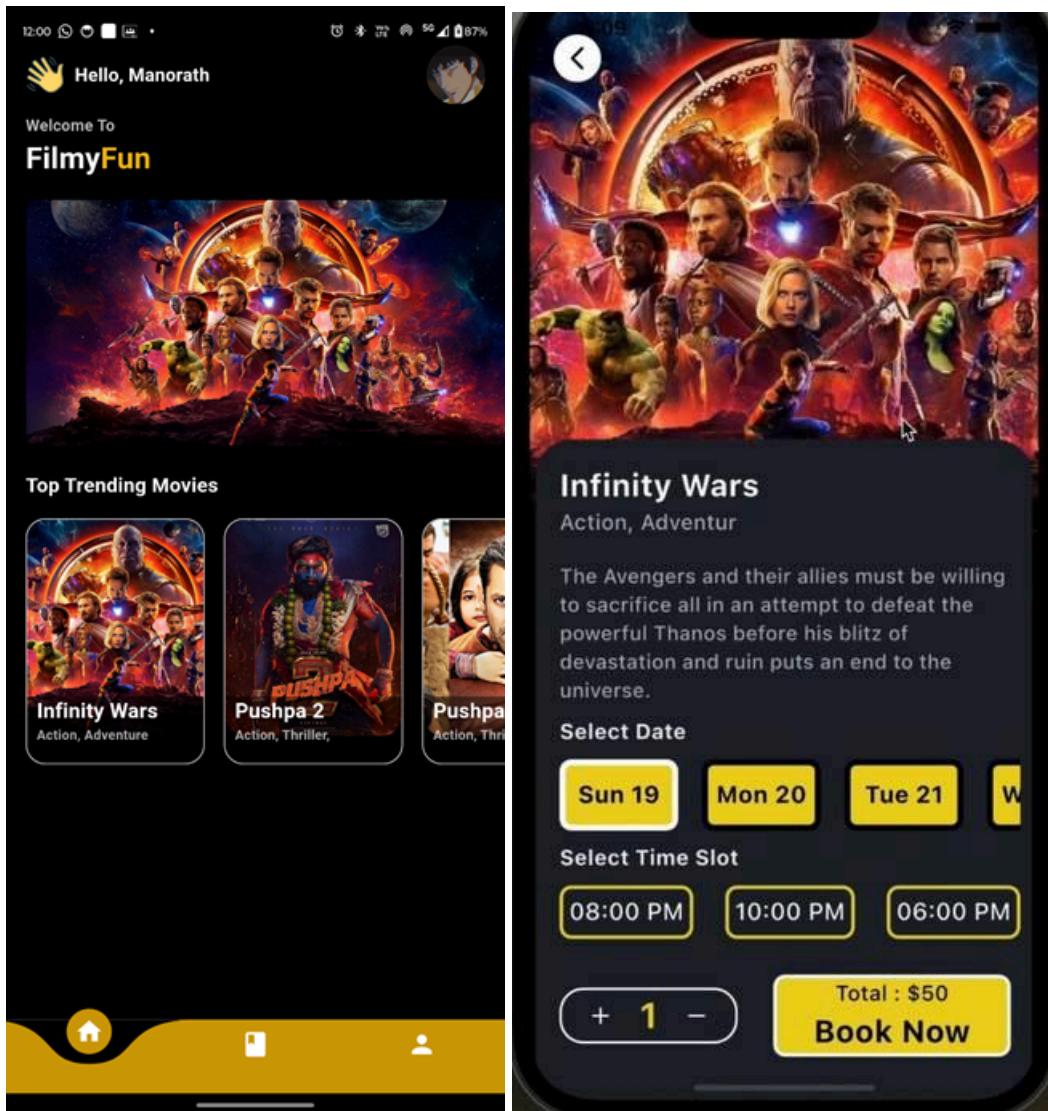
Roll No.



**Project Title:**

**Roll No.**

On Homepage after clicking on Movie image it navigates to Details page



**Project Title:**

**Roll No.**

## MAD & PWA Lab Journal

Experiment No.	06
Experiment Title.	To Connect Flutter UI with fireBase database
Roll No.	19
Name	Manorath Ital
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:	

**Project Title:**

**Roll No.**

## MAD & PWA Lab Journal

Experiment No.	07
Experiment Title.	To write meta data of your Ecommerce PWA in a Web app manifest file to enable “add to homescreen feature”.
Roll No.	19
Name	Manorath Ital
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO4: Understand various PWA frameworks and their requirements
Grade:	

## Experiment No. 7

Name:- Manorath Ital

Roll No:- 19

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

### **Theory:-**

#### **Regular Web App**

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

#### **Progressive Web App**

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

#### **Difference between PWAs vs. Regular Web Apps:**

A Progressive Web is different and better than a Regular Web app with features like:

##### **1. Native Experience**

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

##### **2. Ease of Access**

Unlike other mobile apps, PWAs do not demand longer download time and make memory space available for installing the applications. The PWAs can be shared and installed by a link, which cuts down the number of steps to install and use. These applications can easily keep an app icon on the user's home screen, making the app easily accessible to the users and helps the brands remain in the users' minds, and improving the chances of interaction.

##### **3. Faster Services**

<b>Project Title:</b>	<b>Roll No.</b>
-----------------------	-----------------

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

#### 4. Engaging Approach

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

#### 5. Updated Real-Time Data Access

Another plus point of PWAs is that these apps get updated on their own. They do not demand the end-users to go to the App Store or other such platforms to download the update and wait until installed.

In this app type, the web app developers can push the live update from the server, which reaches the apps residing on the user's devices automatically. Therefore, it is easier for the mobile app developer to provide the best of the updated functionalities and services to the end-users without forcing them to update their app.

#### 6. Discoverable

PWAs reside in web browsers. This implies higher chances of optimizing them as per the Search Engine Optimization (SEO) criteria and improving the Google rankings like that in websites and other web apps.

#### 7. Lower Development Cost

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

#### Pros and cons of the Progressive Web App

The main features are:

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

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

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

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

<b>Project Title:</b>	<b>Roll No.</b>
-----------------------	-----------------

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

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

Reactivable — Make it easy to reactivate the application thanks to capabilities such as web notifications.

Installable — They allow the user to “save” the apps that he considers most useful with the corresponding icon on the screen of his mobile terminal (home screen) without having to face all the steps and problems related to the use of the app store.

Linkable — Easily shared via URL without complex installations.

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

Weaknesses refer to:

iOS support from version 11.3 onwards;

Greater use of the device battery;

Not all devices support the full range of PWA features (same speech for iOS and Android operating systems);

It is not possible to establish a strong re-engagement for iOS users (URL scheme, standard web notifications);

Support for offline execution is however limited;

Lack of presence on the stores (there is no possibility to acquire traffic from that channel);

There is no “body” of control (like the stores) and an approval process;

Limited access to some hardware components of the devices;

Little flexibility regarding “special” content for users (eg loyalty programs, loyalty, etc.).

Code:-

manifest.json

{

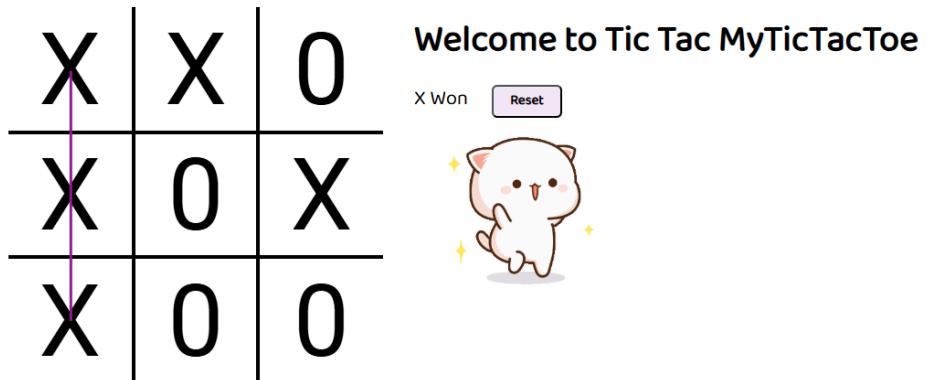
```
"name": "Tic Tac Toe",
"short_name": "TicTacToe",
"description": "A simple Tic-Tac-Toe game.",
"start_url": ".",
"display": "standalone",
"background_color": "#ffffff",
"theme_color": "#4CAF50",
"icons": [
```

**Project Title:**

**Roll No.**

```
{  
  "src": "icons/tic-tac-toe-icon.png",  
  "sizes": "192x192",  
  "type": "image/png"  
},  
{  
  "src": "icons/tic-tac-toe-icon.png",  
  "sizes": "512x512",  
  "type": "image/png"  
}  
]  
}
```

**Output:-**



<https://mano-rath.github.io/tic-tac-toe/>

**Conclusion:-**

**Project Title:**

**Roll No.**

Hence, we learnt how to write a metadata of our website PWA in a Web App Manifest File to enable add to homescreen feature.

**Project Title:**

**Roll No.**

## MAD & PWA Lab Journal

Experiment No.	08
Experiment Title.	To code and register a service worker, and complete the install and activation process for a new service worker for the E-commerce PWA
Roll No.	19
Name	Manorath Ital
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

# **EXPERIMENT NO: - 08**

**Name:** - Manorath Ital**Class:** - D15A**Roll:No:** - 19

**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 dominate **Network Traffic**

You can manage all network traffic of the page and do any manipulations. For example, when the page requests a CSS file, you can send plain text as a response or when the page requests an HTML file, you can send a png file as a response. You can also send a true response too.

- You can **Cache**

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

- You can manage **Push Notifications**

You can manage push notifications with Service Worker and show any information message to the user.

- You can **Continue**

Although Internet connection is broken, you can start any process with Background Sync of Service Worker.

### What can't we do with Service Workers?

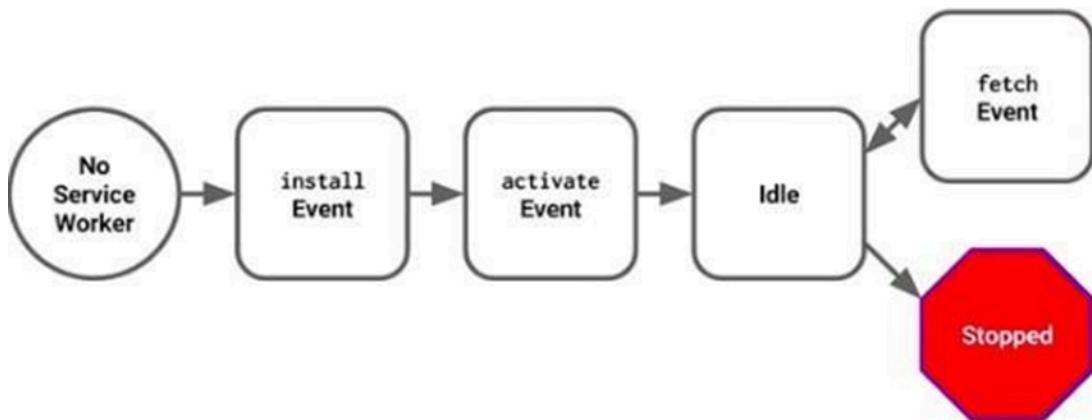
- You can't access the **Window**

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

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

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

### Service Worker Cycle



<b>Project Title:</b>	<b>Roll No.</b>

A service worker goes through three steps in its life cycle:

- Registration
- Installation
- Activation

## **Registration**

To install a service worker, you need to register it in your main JavaScript code. Registration tells the browser where your service worker is located, and to start installing it in the background. Let's look at an example:

main.js

```
if ('serviceWorker' in navigator) { navigator.serviceWorker.register('/service-worker.js')
  .then(function(registration) {
    console.log('Registration successful, scope is:', registration.scope);
  })
  .catch(function(error) {
    console.log('Service worker registration failed, error:', error);
  });
}
```

This code starts by checking for browser support by examining `navigator.serviceWorker`. The service worker is then registered with `navigator.serviceWorker.register`, which returns a promise that resolves when the service worker has been successfully registered. The scope of the service worker is then logged with `registration.scope`. If the service worker is already installed, `navigator.serviceWorker.register` returns the registration object of the currently active service worker.

The scope of the service worker determines which files the service worker controls, in other words, from which path the service worker will intercept requests. The default scope is the location of the service worker file, and extends to all directories below. So if `service-worker.js` is located in the root directory, the service worker will control requests from all files at this domain.

You can also set an arbitrary scope by passing in an additional parameter when registering. For example:

**Project Title:**

**Roll No.**

main.js

```
navigator.serviceWorker.register('/service-worker.js', { scope: '/app/'  
});
```

## Installation

Once the browser registers a service worker, installation can be attempted. This occurs if the service worker is considered to be new by the browser, either because the site currently doesn't have a registered service worker, or because there is a byte difference between the new service worker and the previously installed one.

A service worker installation triggers an install event in the installing service worker. We can include an install event listener in the service worker to perform some task when the service worker installs. For instance, during the install, service workers can precache parts of a web app so that it loads instantly the next time a user opens it (see caching the application shell). So, after that first load, you're going to benefit from instant repeat loads and your time to interactivity is going to be even better in those cases. An example of an installation event listener looks like this:

service-worker.js

```
// Listen for install event, set callback  
self.addEventListener('install',  
function(event) {  
  
  // Perform some task  
  
});
```

## Activation

Once a service worker has successfully installed, it transitions into the activation stage. If there are any open pages controlled by the previous service worker, the new service worker enters a waiting state. The new service worker only activates when there are no longer any pages loaded that are still using the old service worker. This ensures that only one version of the service worker is running at any given time.

When the new service worker activates, an activate event is triggered in the activating service worker. This event listener is a good place to clean up outdated caches (see the Offline Cookbook for an example).

service-worker.js

```
self.addEventListener('activate', function(event) {
```

**Project Title:**

**Roll No.**

```
// Perform some task
});
```

Once activated, the service worker controls all pages that load within its scope, and starts listening for events from those pages. However, pages in your app that were loaded before the service worker activation will not be under service worker control. The new service worker will only take over when you close and reopen your app, or if the service worker calls `clients.claim()`. Until then, requests from this page will not be intercepted by the new service worker. This is intentional as a way to ensure consistency in your site.

## **Code :**

```
// Service Worker Script
```

```
const CACHE_NAME = 'blogbreeze-v1'; // Cache name to identify the version of
cached content const urlsToCache = [
  '/', // Home page
  'index.html', // Main HTML file
  'ani.html', // Any additional pages
  you want to cache
  '/src/assets/react.svg', // App icon
  '/public/icon.png', // Favicon
  '/src/main.jsx', // Main JS file for app functionality
  // 'https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css' //
  FontAwesome for icons
];
```

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

**Project Title:**

**Roll No.**

```
        });
    );
});

// Activate Service Worker

self.addEventListener('activate',
  (event) => { console.log('Service
Worker: Activated');

// Remove old caches if there
are any event.waitUntil(
caches.keys().then((cacheNa
mes) => {
  return Promise.all(
    cacheNames.map((cach
eName) => {
      if (cacheName !==
CACHE_NAME) {
        return
        caches.delete(cacheN
ame);
      }
    })
  );
});
});

// Fetch event: Intercept network requests and serve cached content
self.addEventListener('fetch', (event) => {
  console.log('Service Worker: Fetching',
  event.request.url); event.respondWith(
    caches.match(event.request)
    .then((cachedResponse) => {
```

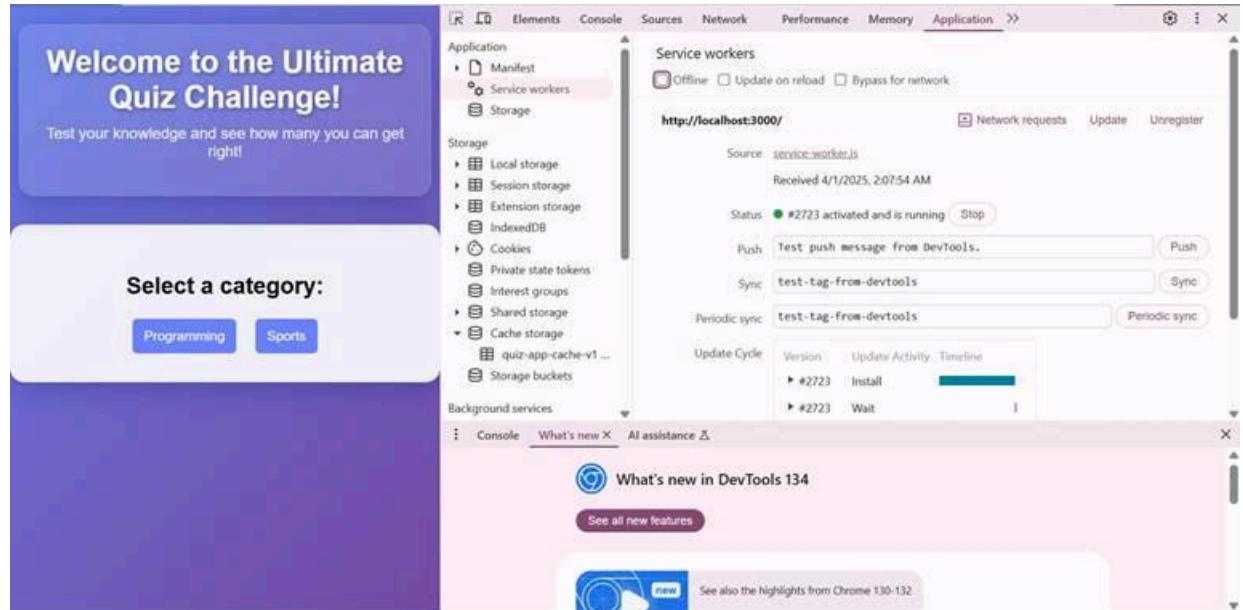
**Project Title:**

**Roll No.**

```
// Return cached content if found, otherwise fetch
from network return cachedResponse ||
fetch(event.request);

})
;

});
```



**Project Title:**

**Roll No.**

The screenshot shows a web browser window with a quiz application. The main content area displays the title "Welcome to the Ultimate Quiz Challenge!" and a message "Test your knowledge and see how many you can get right!". Below this, there is a section titled "Select a category:" with two buttons: "Programming" and "Sports".

The browser's address bar shows the URL `http://localhost:3000/`. To the right of the address bar is the Chrome DevTools interface, specifically the "Application" tab. The "Service workers" section shows a service worker named "service-worker.js" is active and running. The "Storage" section lists various types of storage like Local storage, Session storage, Extension storage, Cookies, and Cache storage. The "Background services" section shows a "Background fetch" entry.

A sidebar on the right side of the DevTools shows a "What's new" section for DevTools 134, featuring a "See all new features" button and a "See also the highlights from Chrome 130-132" link.

This screenshot shows the same quiz application and DevTools interface as the previous one, but with the "Network" tab selected instead of "Application".

The "Network" tab displays a table of network requests for the URL `http://localhost:3000`. The table includes columns for #, Name, Response, Content-Type, Content-Length, and Time. There are 8 entries listed:

#	Name	Response	Content-Type	Content-Length	Time
0	/	basic	text/html	644	4/1/20
1	/favicon.ico	basic	image/x-icon	0	4/1/20
2	/index.html	basic	text/html	644	4/1/20
3	/logo192.png	basic	image/png	5,347	4/1/20
4	/logo512.png	basic	image/png	9,664	4/1/20
5	/manifest.json	basic	application/json	517	4/1/20
6	/static/css/main.a0b9aca2.css	basic	text/css	0	4/1/20
7	/static/js/main.904af396.js	basic	application/javascript	0	4/1/20

At the bottom of the DevTools sidebar, there is a message: "No cache entry selected" and "Select a cache entry above to preview".

**Conclusion :** Thus we have learnt to code and register a service worker, and complete the install and activation process for a new service worker for the PWA.

**Project Title:**

**Roll No.**

## MAD & PWA Lab Journal

Experiment No.	09
Experiment Title.	To implement Service worker events like fetch, sync and push for E-commerce PWA
Roll No.	19
Name	Manorath Ital
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

# **EXPERIMENT NO: - 09**

**Name:** - Manorath Ital**Class:** - D15A**Roll:No:** - 19

## **EXPERIMENT NO. 9**

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

### **Theory:**

#### **Service Worker**

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

Things to note about Service Worker:

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

## Fetch Event

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

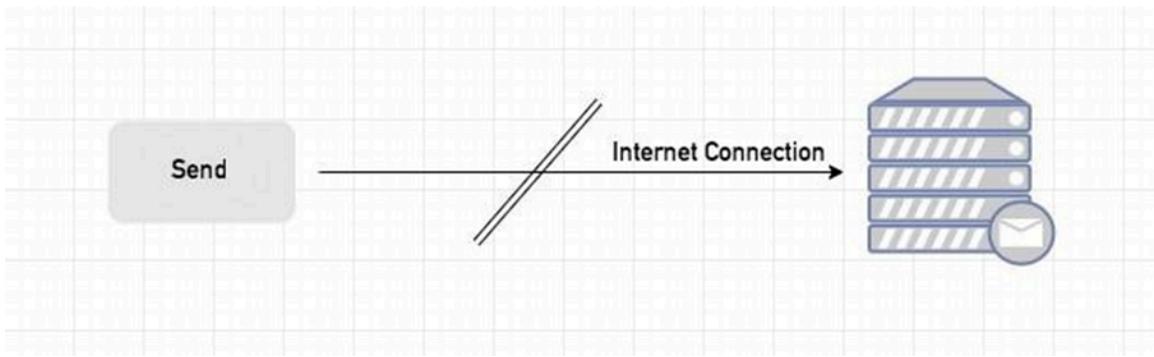
Of course, you can use many different methods but you can find in the following example a “cache first” and “network first” approach. In this example, if the request’s and current location’s origin are the same (Static content is requested.), this is called “cacheFirst” but if you request a targeted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

- **CacheFirst** - In this function, if the received request has cached before, the cached response is returned to the page. But if not, a new response requested from the network.
- **NetworkFirst** - In this function, firstly we can try getting an updated response from the network, if this process completed successfully, the new response will be cached and returned. But if this process fails, we check whether the request has been cached before or not. If a cache exists, it is returned to the page, but if not, this is up to you. You can return dummy content or information messages to the page.

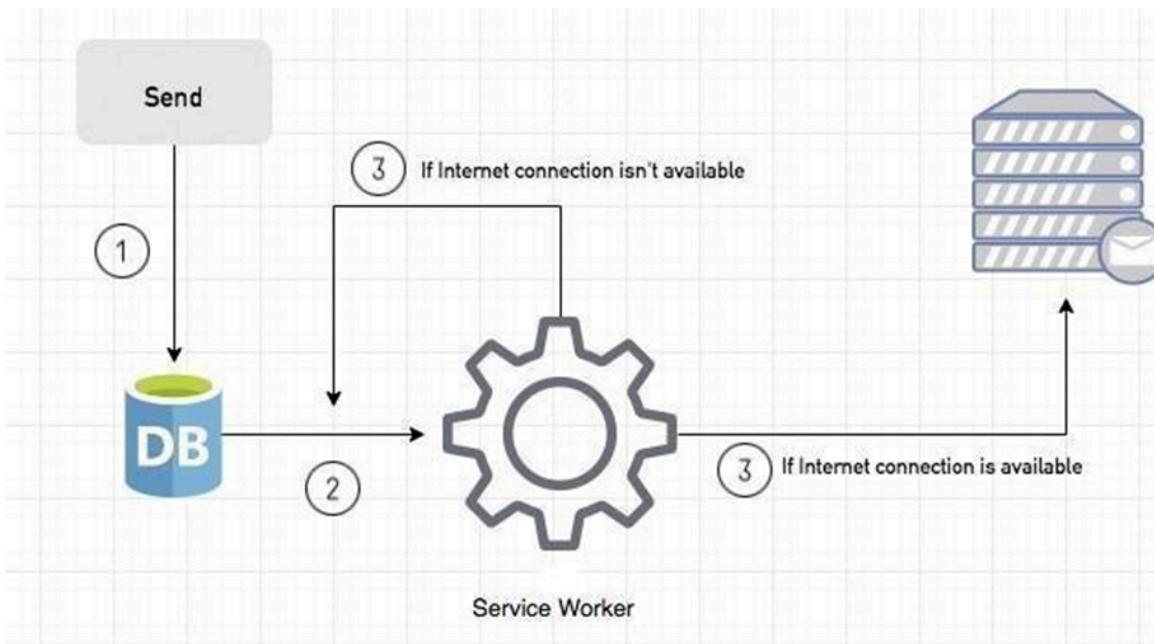
## Sync Event

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

Here is a job for the Background Sync.



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



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

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

1. When we click the “send” button, email content will be saved to IndexedDB.
2. Background Sync registration.
3. **If the Internet connection is available**, all email content will be read and sent to Mail Server.

**If the Internet connection is unavailable**, the service worker waits until the connection is available even though the window is closed. When it is available, email content will be sent to Mail Server.

## 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.

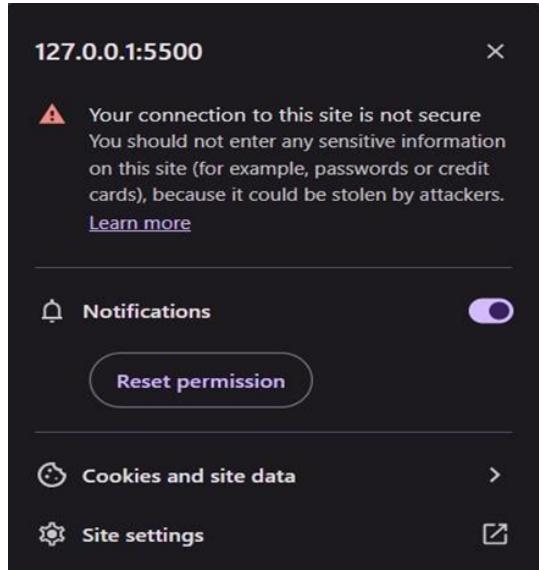
“Notification.requestPermission();” is the necessary line to show notification to the user. If you don’t want to show any notification, you don’t need this line.

**Project Title:**

**Roll No.**

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” proper.



A screenshot of a browser's developer tools application panel. On the left, there is a preview of a web page titled "Welcome to the Ultimate Quiz Challenge!". The page content includes "Test your knowledge and see how many you can get right!", a "Trigger Push Notification" button, and a "Select a category:" section with "Programming" and "Sports" buttons. On the right, the developer tools panel shows the "Application" tab selected. It displays a tree view of storage and service workers. Under "Service workers", it shows a list for "http://localhost:3000/" with a source file "service-worker.js", a status of "#2972 activated and is running", and a client at "http://localhost:3000/". It also shows a "Push" event with a payload {"method": "pushMessage", "message": "Hello from Service Worker!"} and a "Sync" event with a payload "syncMessage". The "Console" tab at the bottom shows log messages: "Fetch successful!", "Serving from cache: http://localhost:3000/manifest.json", "Fetch successful!", "Notification displayed!", and "Sync successful!". The "Sources" tab is also visible in the header.

**Project Title:**

**Roll No.**

## MAD & PWA Lab Journal

Experiment No.	10
Experiment Title.	To study and implement deployment of Ecommerce PWA to GitHub Pages.
Roll No.	19
Name	Manorath Ital
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

## **EXPERIMENT NO: - 10**

**Manorath Ital****D15A-19**

**AIM:** - To study and implement deployment of Ecommerce PWA to GitHub Pages.

**Theory: -**

GitHub Pages: Static Website Hosting Made Simple

GitHub Pages is a free hosting service that allows users to publish public webpages directly from a GitHub repository. It is particularly useful for static websites, project documentation, and blogs.

**Key Features**

- Jekyll Integration: Built-in support for Jekyll enables easy blogging.
- Custom Domains: Allows users to configure their own URLs.
- Automatic Page Deployment: Simply push your changes to the repository, and the updates go live.

**Why Choose GitHub Pages?**

- Completely Free: No hosting charges.
- Seamless GitHub Integration: Works directly with your repositories.
- Quick Setup: Just create a repository, push your files, and your site is live.

**Who Uses GitHub Pages?**

Companies like Lyft, CircleCI, and HubSpot use GitHub Pages for their documentation and static sites. It is widely adopted, appearing in 775 company stacks and 4,401 developer stacks.

**Pros & Cons**

Project Title:	Roll No.
----------------	----------

## Pros

- Familiar interface for GitHub users.
- Simple deployment via the `gh-pages` branch.
- Supports custom domains with easy DNS configuration.

## Cons

- Repositories need to be public unless you have a paid plan.
  - Limited HTTPS support for custom domains (expected to improve).
  - Jekyll plugins have limited support.
- 

## Firebase: A Full-Featured Real-Time Backend

Firebase is a cloud-based real-time application platform developed by Google. It enables developers to build dynamic, collaborative applications with ease.

### Key Features

- Real-Time Database: Automatically syncs data across all connected clients.
- Cloud-Based Storage: JSON-based storage accessible via REST APIs.
- Scalable Infrastructure: Works well with existing services and scales automatically.
- Authentication & Cloud Messaging: Secure login and push notifications.

### Why Choose Firebase?

- Instant Backend Setup: No need to build a separate backend.
- Fast & Responsive: Real-time data synchronization.
- Built-in HTTPS: Free SSL certificates for custom domains.

Project Title:	Roll No.
Who Uses Firebase?	

## Who Uses Firebase?

Companies like Instacart, 9GAG, and Twitch rely on Firebase for their backend needs. Firebase is widely adopted, appearing in 1,215 company stacks and 4,651 developer stacks.

## Pros & Cons

### Pros

- Hosted by Google, ensuring reliability and security.
- Comes with authentication, messaging, and real-time database services.
- Free HTTPS support for all custom domains.

### Cons

- Limited Free Plan: 10 GB of data transfer per month (can be mitigated with a CDN).
- Command-Line Deployment: No GUI for hosting.
- No Built-in Static Site Generator Support: Unlike GitHub Pages, Firebase doesn't natively support static site generators like Jekyll.

**hosted link:** [website link](#)

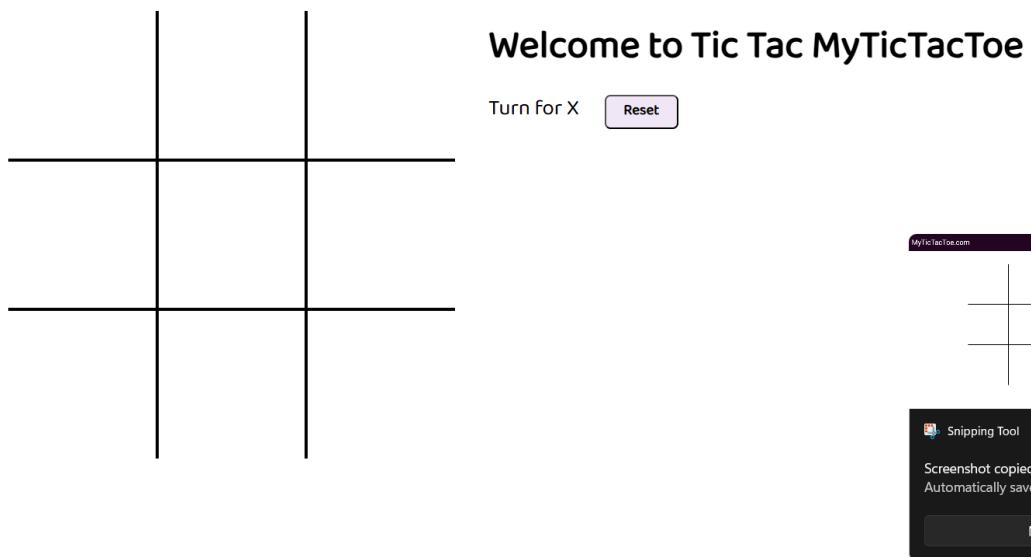
## Github Screenshot:

**Project Title:**

**Roll No.**

The screenshot shows a GitHub Actions build log for a project named "tic-tac-toe". The build was triggered via a dynamic event last month. It consists of three steps: "build", "report-build-status", and "deploy". The "build" step took 19 seconds, "report-build-status" took 4 seconds, and "deploy" took 10 seconds. The final artifact is a link to the deployed site: <https://mano-rath.github.io/tic-tac-toe/>. The status is marked as "Success" with a total duration of 40 seconds and 1 artifact.

MyTicTacToe.com



**Project Title:**

**Roll No.**

## MAD & PWA Lab Journal

Experiment No.	11
Experiment Title.	To use google Lighthouse PWA Analysis Tool to test the PWA functioning.
Roll No.	19
Name	Manorath Ital
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO6: Develop and Analyze PWA Features and deploy it over app hosting solution
Grade:	

<b>Project Title:</b>	<b>Roll No.</b>
-----------------------	-----------------

## EXPERIMENT NO: - 11

Manorath Ital  
D15A/19

**AIM:** - To use google Lighthouse PWA Analysis Tool to test the PWA functioning.

### **THEORY:** -

Google Lighthouse is an open-source tool that audits web applications based on multiple key parameters, including performance, accessibility, Progressive Web App (PWA) implementation, and best practices. It provides a detailed, automated report that helps developers optimize their websites efficiently. Unlike traditional manual audits, which can take days or even weeks, Lighthouse generates insights within minutes.

One of the key advantages of Lighthouse is its ease of use—no complex setup is required. Simply run it on a webpage or provide a URL, and it will generate an extensive performance report.

#### **Key Features and Audit Metrics**

Lighthouse can audit both desktop and mobile versions of a webpage. The core evaluation criteria include:

##### **1. Performance**

This metric measures how efficiently a webpage loads and displays content. Key factors influencing the performance score include:

- Page load speed – How quickly the page becomes visible to the user.
- First Contentful Paint (FCP) – The time taken for the first piece of content to appear.
- Largest Contentful Paint (LCP) – The time taken for the main content to fully load.
- Cumulative Layout Shift (CLS) – Measures how visually stable a page is (i.e., avoiding unexpected shifts in content).
- Time to Interactive (TTI) – The time it takes for the page to become fully functional. Lighthouse assigns a score from 0 to 100 based on percentile rankings, where:
  - 100 → Top 2% of websites (98th percentile)
  - 50 → Around the 75th percentile
  - Lower scores → Indicate areas that need optimization

##### **2. Progressive Web App (PWA) Score (Mobile)**

With the rise of PWAs, modern web applications aim to provide a native app-like experience. Lighthouse evaluates the PWA implementation based on Google's Baseline PWA Checklist, which includes:

<b>Project Title:</b>	<b>Roll No.</b>
<ul style="list-style-type: none"> <li>• Service Worker implementation – Ensuring offline support and background synchronization.</li> <li>• App Manifest compliance – Providing metadata for better mobile integration.</li> <li>• Viewport configuration – Optimizing mobile responsiveness.</li> <li>• Performance in script-disabled environments – Ensuring the page functions even when JavaScript is disabled.</li> </ul> <p>A high PWA score indicates that the application meets essential PWA criteria and provides an app-like user experience.</p>	

### 3. Accessibility

Accessibility ensures that web applications are usable by individuals with disabilities. Lighthouse audits a webpage based on:

- ARIA attributes – Enhancing accessibility through attributes like aria-required.
- Text alternatives for media – Ensuring audio and visual content is accessible.
- Semantic HTML – Proper use of <section>, <article>, <button>, and other elements that improve screen-reader compatibility.

Unlike other metrics, accessibility checks follow a pass/fail approach—if a necessary feature is missing, it significantly impacts the score. A higher accessibility score ensures inclusivity for users with visual or cognitive impairments.

### 4. Best Practices

This metric evaluates whether the website follows modern web development best practices, including:

- Use of HTTPS – Ensuring secure data transmission.
- Avoiding deprecated code – Preventing the use of outdated elements, directives, and libraries.
- Secure password inputs – Disabling paste-into fields to mitigate credential theft risks.
- User security alerts – Prompting users about geo-location access and cookie usage on load. A high score indicates that the website follows industry standards, improving security, usability, and maintainability.

### Manifest.json

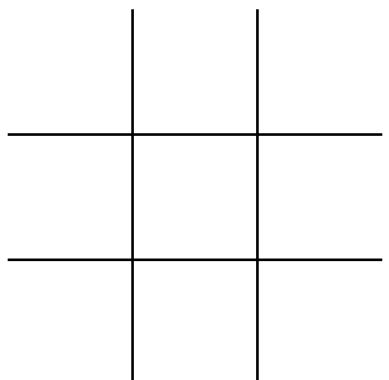
```
{
  "name": "Tic Tac Toe",
  "short_name": "TicTacToe",
  "description": "A simple Tic-Tac-Toe game.",
  "start_url": ".",
  "display": "standalone",
  "background_color": "#ffffff",
```

**Project Title:**

**Roll No.**

```
"theme_color": "#4CAF50",
"icons": [
  {
    "src": "icons/tic-tac-toe-icon.png",
    "sizes": "192x192",
    "type": "image/png"
  },
  {
    "src": "icons/tic-tac-toe-icon.png",
    "sizes": "512x512",
    "type": "image/png"
  }
]
```

MyTicTacToe.com

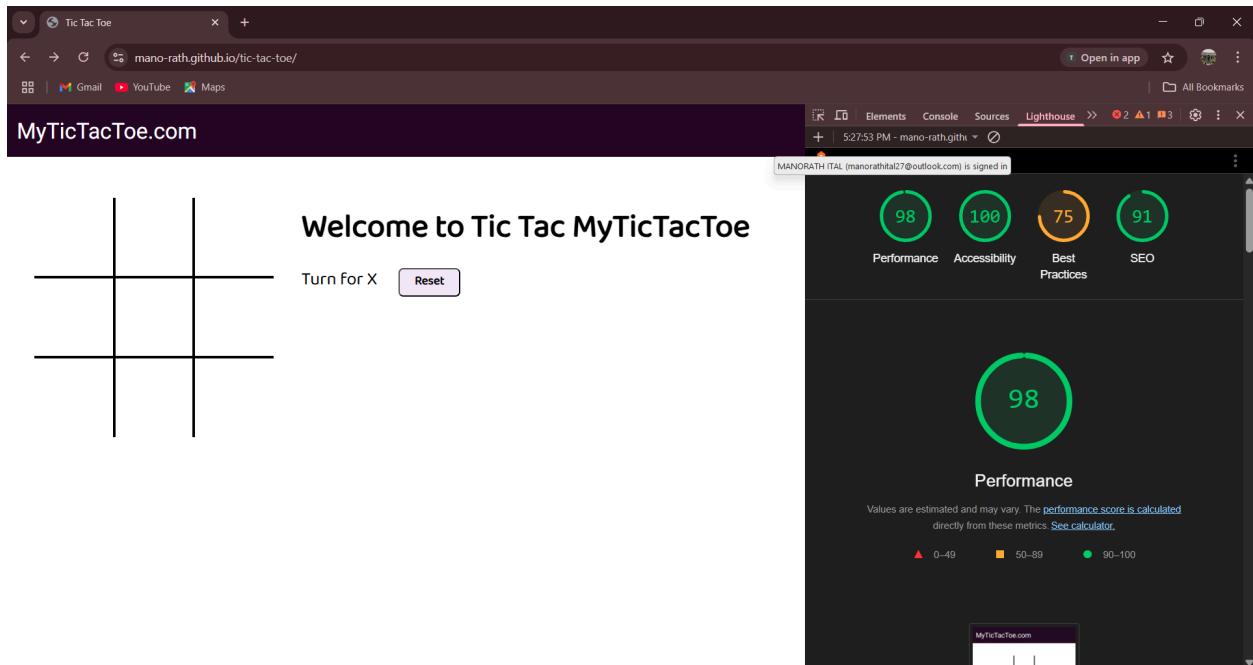


Welcome to Tic Tac MyTicTacToe

Turn for X

**Project Title:**

**Roll No.**



Project Title:

Roll No.



The dashboard displays four circular performance metrics with numerical values: 98, 100, 75, and 91. Below these metrics is a section titled "DIAGNOSTICS" which lists various performance optimization opportunities with their potential savings or findings.

- ▲ Eliminate render-blocking resources — Potential savings of 710 ms
- Serve static assets with an efficient cache policy — 6 resources found
- Defer offscreen images — Potential savings of 45 KiB
- Reduce unused JavaScript — Potential savings of 48 KiB
- Avoid enormous network payloads — Total size was 3,498 KiB
- Minimize third-party usage — Third-party code blocked the main thread for 100 ms
- Avoid long main-thread tasks — 4 long tasks found
- Avoid chaining critical requests — 3 chains found
- Largest Contentful Paint element — 1,950 ms

Project Title:

Roll No.

The screenshot shows the Lighthouse performance audit interface. At the top, there are tabs for Elements, Console, Sources, and Lighthouse, with Lighthouse being the active tab. Below the tabs, it shows the URL <https://mano-rath.github.io/tic-tac-toe/>. The main area displays four circular performance metrics: Performance (100), Accessibility (100), Best Practices (74), and SEO (91). A large green circle with a white border and a dark green center shows a score of 100 under the heading "Performance". Below this, the word "Performance" is written in a large, bold, white font. A explanatory text block follows, stating: "Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)" At the bottom, there is a color legend: a red triangle for 0–49, an orange square for 50–89, and a green circle for 90–100. At the very bottom, a small screenshot of the website shows the title "MyTicTacToe.com" and the welcome message "Welcome to Tic Tac MyTicToe".

Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

▲ 0–49   ■ 50–89   ● 90–100

MyTicTacToe.com

Welcome to Tic Tac MyTicToe

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

# MAD & PWA Lab

## Journal

Experiment No.	Assignment-1
Assignment 1 Questions	<ol style="list-style-type: none"><li>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.</li><li>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.</li><li>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.</li><li>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.</li></ol>
Roll No.	
Name	
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO1: Understand cross platform mobile application development using Flutter framework LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation 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	<ol style="list-style-type: none"><li>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</li><li>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.</li><li>3. Describe the lifecycle of Service Workers, including registration, installation, and activation phases.</li><li>4. Explain the use of IndexedDB in the Service Worker for data storage.</li></ol>
Roll No.	
Name	
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO4:Understand various PWA frameworks and their requirements LO5: Design and Develop a responsive User Interface by applying PWA Design techniques LO6:Develop and Analyze PWA Features and deploy it over app hosting solutions
Grade:	