

# **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 **ADITYA SAMPATH KUMAR** Of **D15A/D15B** semester **VI**, have successfully completed necessary experiments in the **MAD & PWA Lab** under my supervision in **VES Institute of Technology** during the academic year **2024-2025**.

Lab Assistant

Subject Teacher

**Mrs. Kajal Joseph**

Principal

Head of Department

**Dr. Mrs. Shalu Chopra**

**Name of the Course :** MAD & PWA Lab**Course Code :** ITL604**Year/Sem/Class :** D15A/D15B**A.Y.:** 24-25**Faculty Incharge :** Mrs. Kajal Joseph.**Lab Teachers :** Mrs. Kajal Joseph.**Email :** kajal.jewani@ves.ac.in**Programme Outcomes:** The graduate will be able to:

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

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

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

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

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

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

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

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

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

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

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

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

**Program specific Outcomes**

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

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

**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

# Index

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

# MAD & PWA Lab

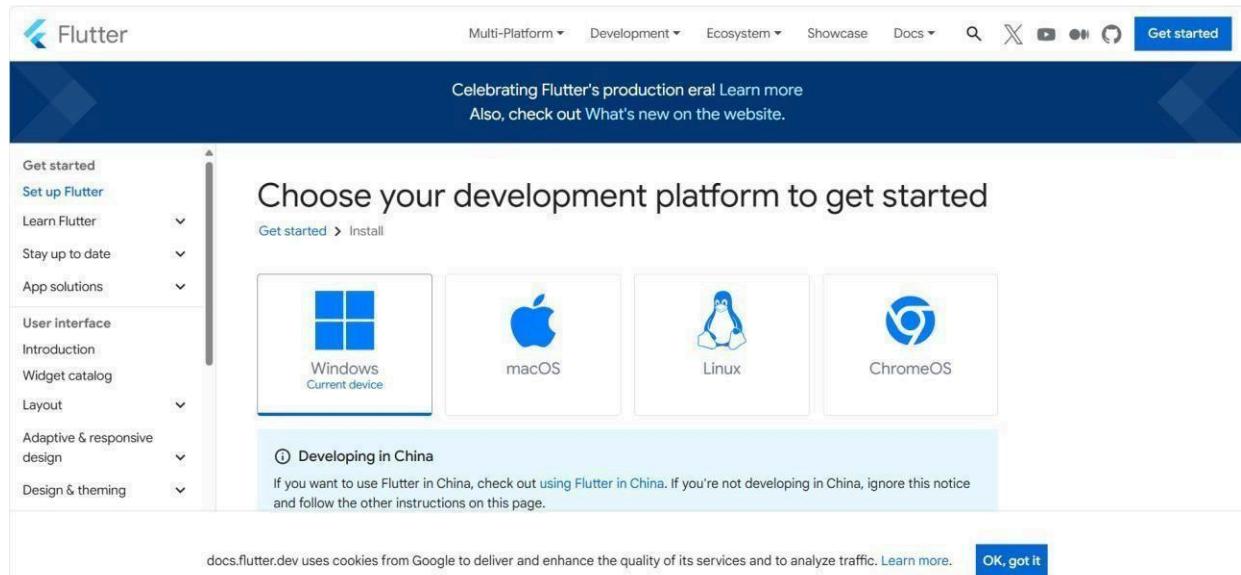
## Journal

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

# EXPERIMENT NO: - 01

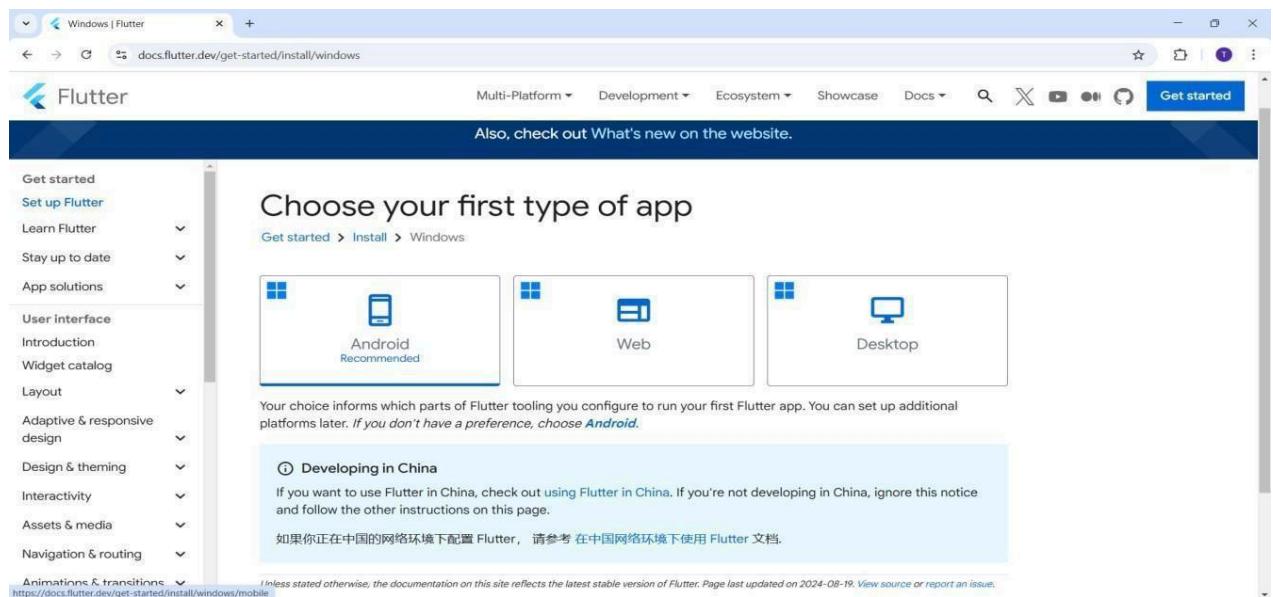
**AIM:** - Installation and Configuration of Flutter Environment.

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



The screenshot shows the Flutter documentation website. On the left, there's a sidebar with navigation links like 'Get started', 'Set up Flutter', 'Learn Flutter', etc. The main content area has a heading 'Choose your development platform to get started' with a sub-section 'Get started > Install'. It features four cards for 'Windows Current device', 'macOS', 'Linux', and 'ChromeOS'. Below these cards is a note about developing in China. At the bottom, there's a cookie consent message and a 'OK, got it' button.

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



This screenshot continues from the previous one, showing the 'Install > Windows' section. It has three cards for 'Android Recommended', 'Web', and 'Desktop'. A note explains that this choice informs tooling configuration. Below is another 'Developing in China' note and a link to the Chinese documentation. At the bottom, there's a footer note about the page's last update.

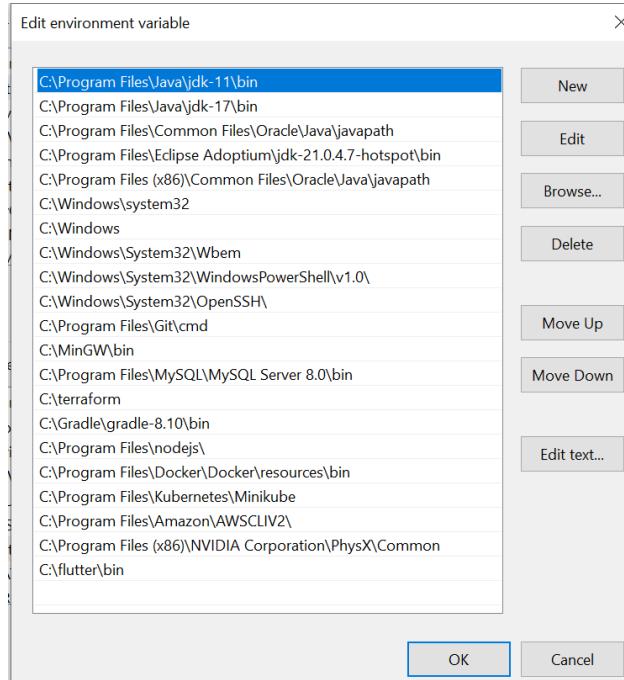
**Step 3:** For Windows, download the stable release (a .zip file).

The screenshot shows a web browser displaying the Flutter documentation at [docs.flutter.dev/get-started/install/windows/mobile](https://docs.flutter.dev/get-started/install/windows/mobile). The page is titled "Download then install Flutter". It provides instructions for downloading the Flutter SDK bundle from its archive and extracting it to a folder. A prominent blue button labeled "flutter\_windows\_3.27.2-stable.zip" is shown, which is the download link for the latest stable release. To the right of the main content, there is a sidebar titled "Contents" with various links related to Flutter setup and development.

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

The screenshot shows the "Extract Compressed (Zipped) Folders" dialog box. The title bar says "Extract Compressed (Zipped) Folders". The main area is titled "Select a Destination and Extract Files". Below it, a message states "Files will be extracted to this folder:" followed by a text input field containing "C:\". To the right of the input field is a "Browse..." button. Below the input field is a checked checkbox labeled "Show extracted files when complete". At the bottom of the dialog are two buttons: "Extract" and "Cancel".

**Step 5 :-** Add Flutter to System PATH. Right-click on the Start Menu > System > Advanced system settings > Environment Variables. Under System Variables, find Path and click Edit. Add the full path to the flutter/bin directory (e.g., C:\flutter\bin).



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

```
Microsoft Windows [Version 10.0.19045.5247]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ACER>$flutter
'$flutter' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\ACER>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.
```

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

```

C:\Users\ACER>flutter doctor
analytics collection will disable both the legacy and new analytics collection systems. You can disable analytics reporting by running `flutter --disable-analytics`

C:\Users\ACER>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.27.4, on Microsoft Windows [Version 10.0.19045.5247], locale en-IN)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✗] Android toolchain - develop for Android devices
    ✗ 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 not installed; this is necessary to develop Windows apps.
      Download at https://visualstudio.microsoft.com/downloads/.
      Please install the "Desktop development with C++" workload, including all of its default components
[!] Android Studio (not installed)
[✓] IntelliJ IDEA Community Edition (version 2023.2)
[✓] VS Code (version 1.96.2)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 3 categories.

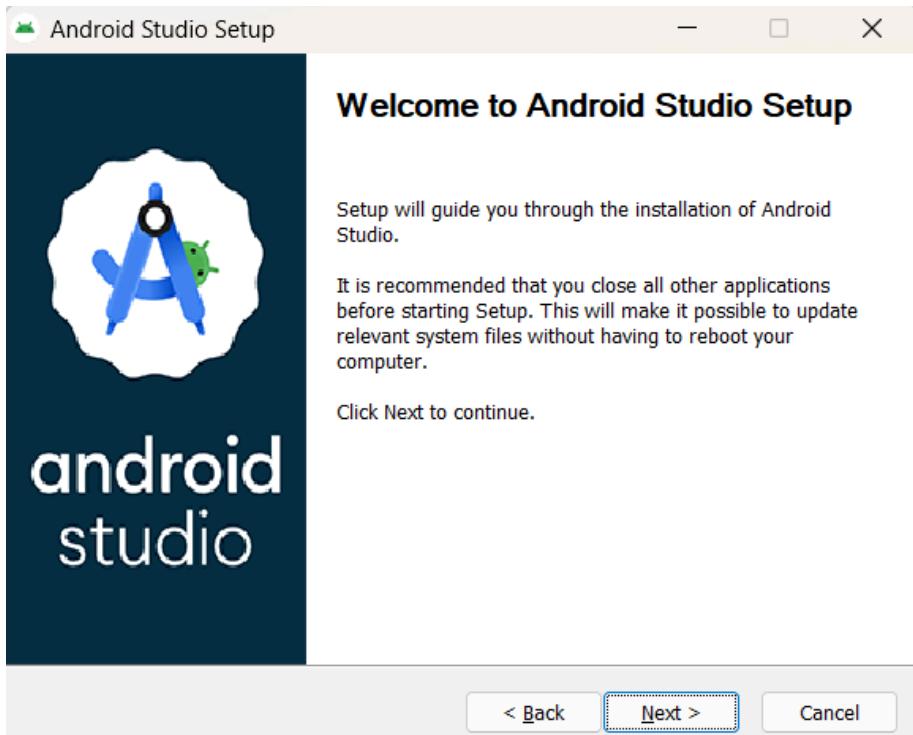
C:\Users\ACER>s

```

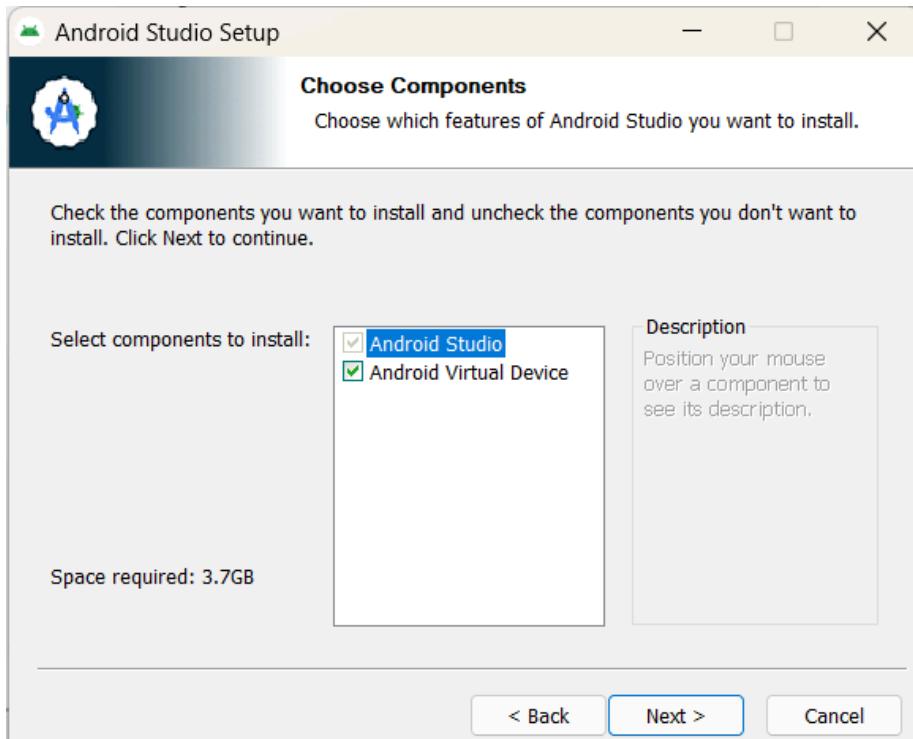
**Step 8 :-** Go to Android Studio and download the installer.

Platform	Android Studio package	Size	SHA-256 checksum
Windows (64-bit)	<a href="#">android-studio-2024.2.2.13-windows.exe</a> Recommended	12 GB	7d93af9e7353ff948f09679e8507bf502b7945d2044fd38ef7ff26cb5d3e
Windows (64-bit)	<a href="#">android-studio-2024.2.2.13-windows.zip</a> No .exe installer	12 GB	855945942f9b04ea9fc039de0bf4fb9dbf45tse37abfab7999da013b046b7f7
Mac (64-bit)	<a href="#">android-studio-2024.2.2.13-mac.dmg</a>	13 GB	acfbbe54d6ce8cf2ff9b43590c7addcb9ade2824282f205fd333be77d2e613
Mac (64-bit, ARM)	<a href="#">android-studio-2024.2.2.13-mac_arm.dmg</a>	13 GB	688fb007e612f3f0c18f316179079dc4560f93d8fe4a7dad80c4cfce35edf7
Linux (64-bit)	<a href="#">android-studio-2024.2.2.13-linux.tar.gz</a>	13 GB	b7f9ed4a79597daca7a8fd5746tdbb79a205ef230c21bed82bed8be6b996cb5

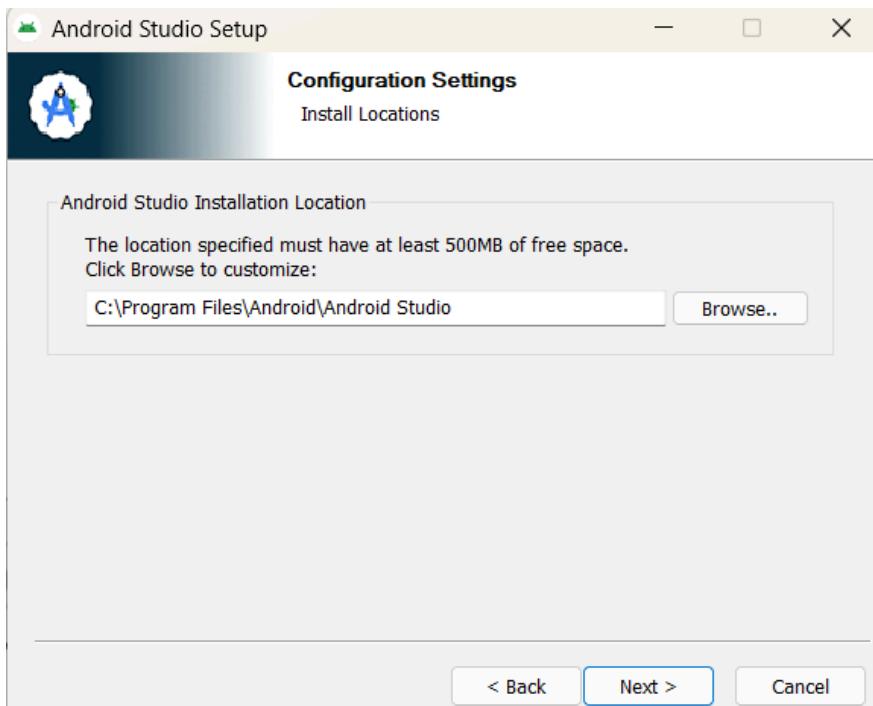
**Step 8.1:** - When the download is complete, open the .exe file and run it. You will get the following dialog box



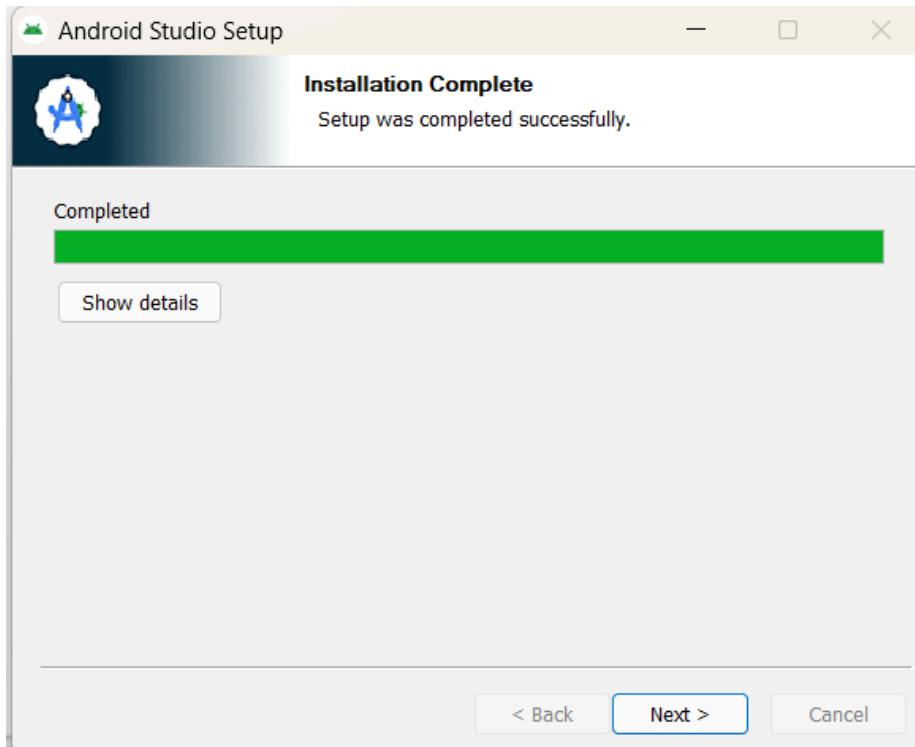
**Step 8.2:** - Select all the Checkboxes and Click on 'Next' Button.

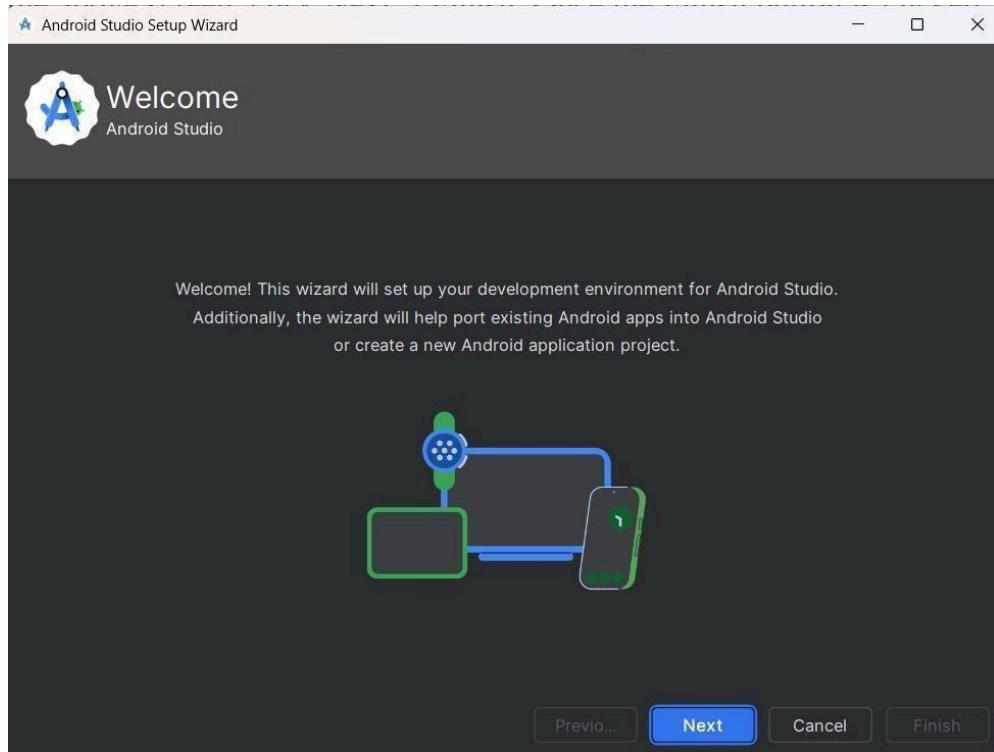


**Step 8.3:** - Change the destination as per your convenience and click on 'Next' Button.

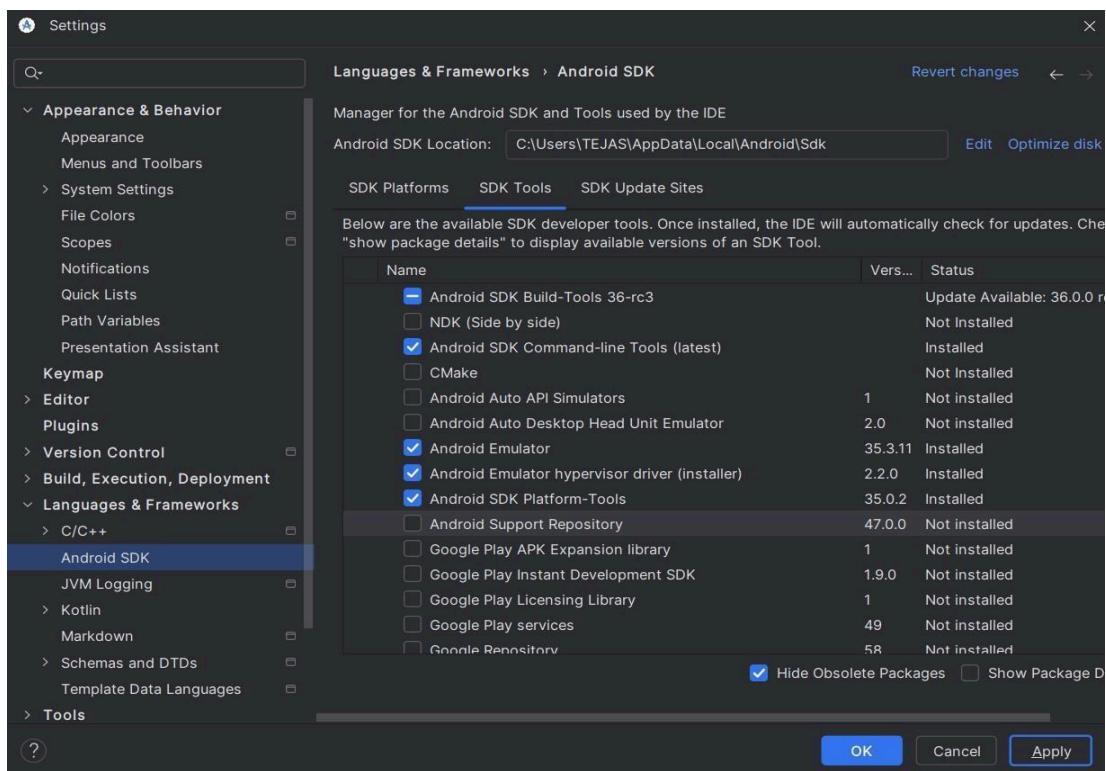


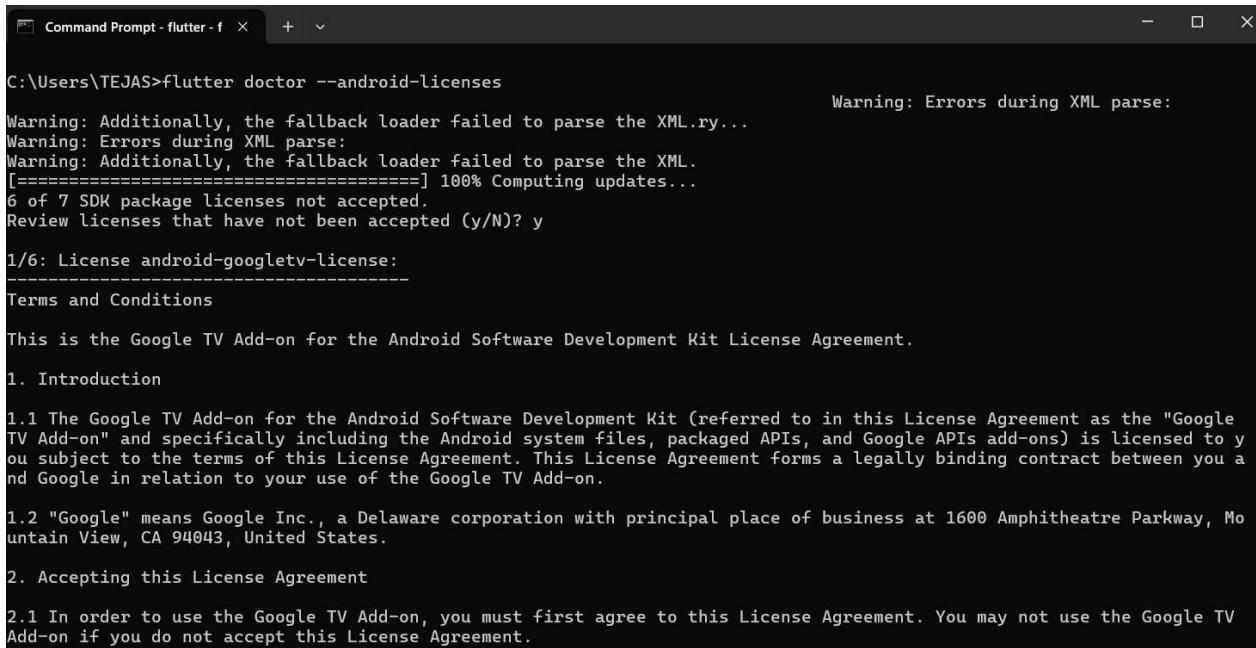
**Step 8.4:** - Follow the steps of the installation wizard. Once the installation wizard completes, you will get the following screen.





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



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

```
C:\Users\TEJAS>flutter doctor --android-licenses
Warning: Additionally, the fallback loader failed to parse the XML.ry...
Warning: Errors during XML parse:
Warning: Additionally, the fallback loader failed to parse the XML.
[=====] 100% Computing updates...
6 of 7 SDK package licenses not 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.

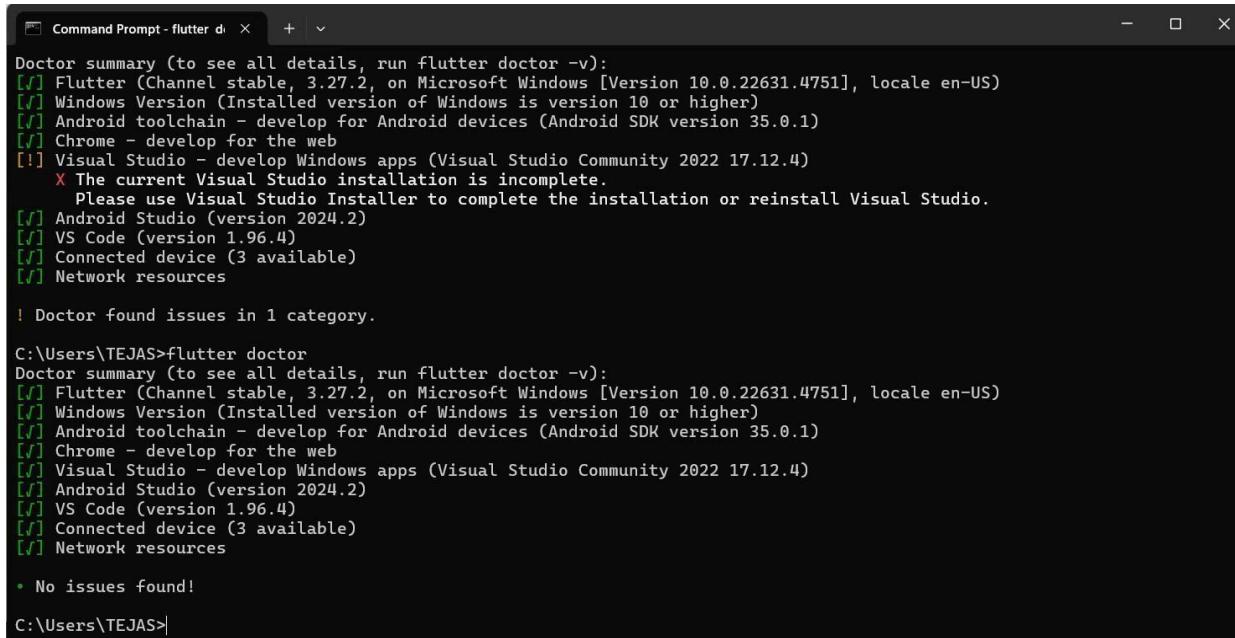
1. Introduction

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.
```



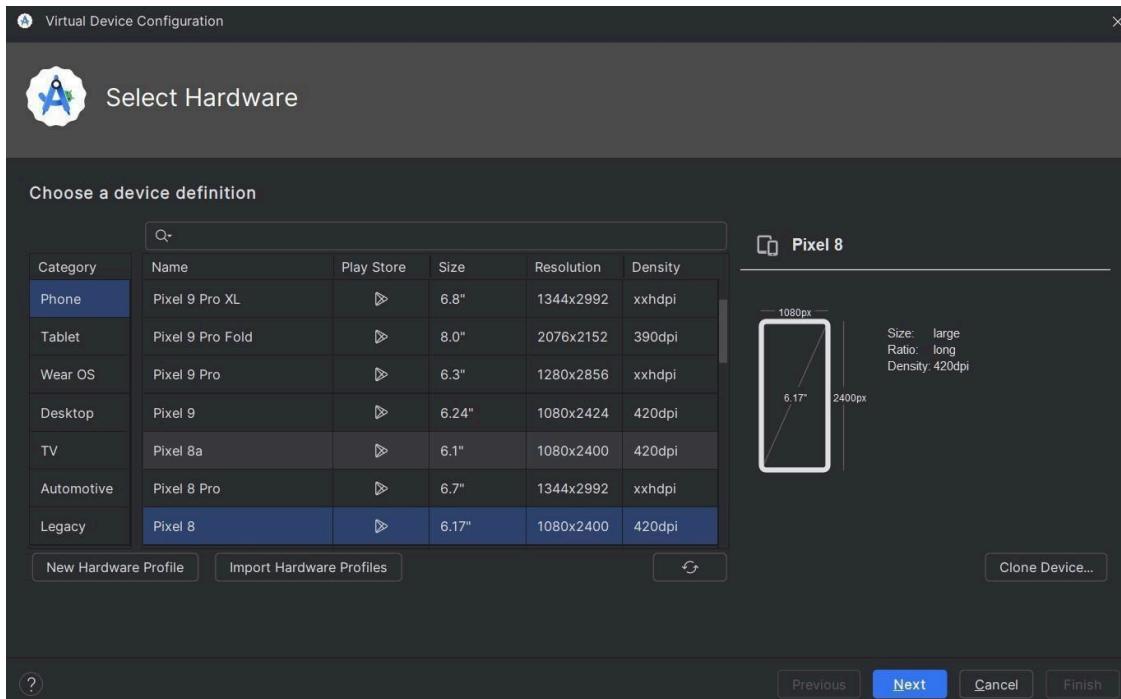
```
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 3.27.2, on Microsoft Windows [Version 10.0.22631.4751], locale en-US)
[!] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices (Android SDK version 35.0.1)
[!] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.12.4)
  X The current Visual Studio installation is incomplete.
    Please use Visual Studio Installer to complete the installation or reinstall Visual Studio.
[!] Android Studio (version 2024.2)
[!] VS Code (version 1.96.4)
[!] Connected device (3 available)
[!] Network resources

! Doctor found issues in 1 category.

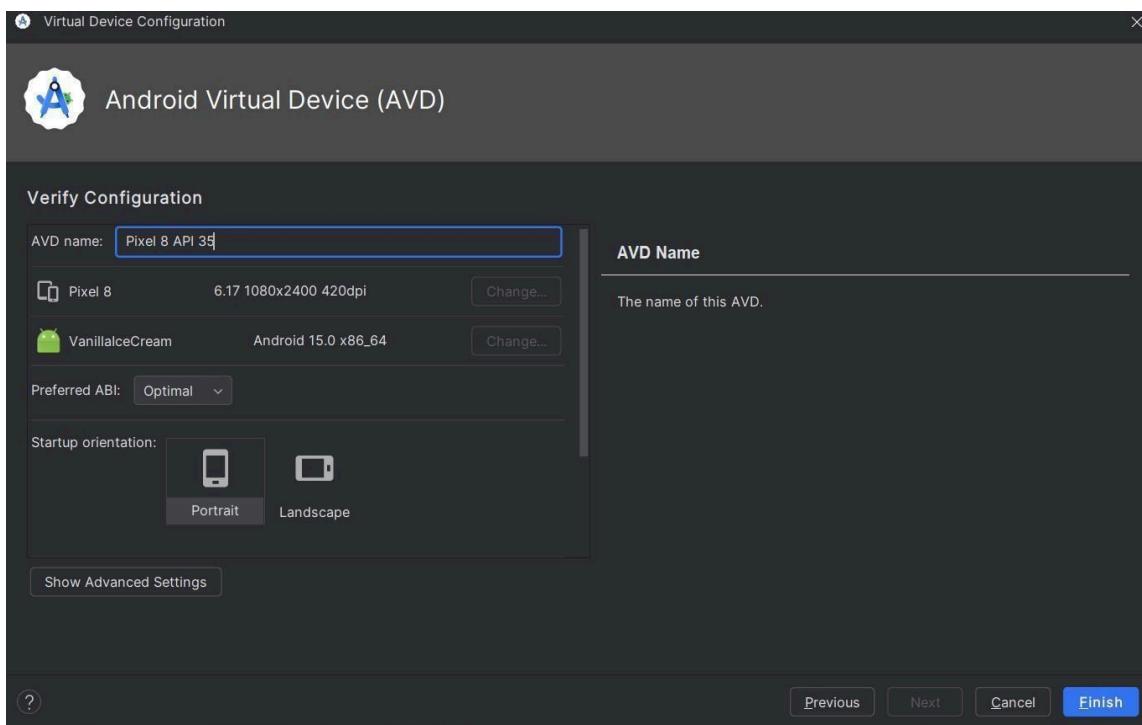
C:\Users\TEJAS>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 3.27.2, on Microsoft Windows [Version 10.0.22631.4751], locale en-US)
[!] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices (Android SDK version 35.0.1)
[!] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.12.4)
[!] Android Studio (version 2024.2)
[!] VS Code (version 1.96.4)
[!] Connected device (3 available)
[!] Network resources

• No issues found!
```

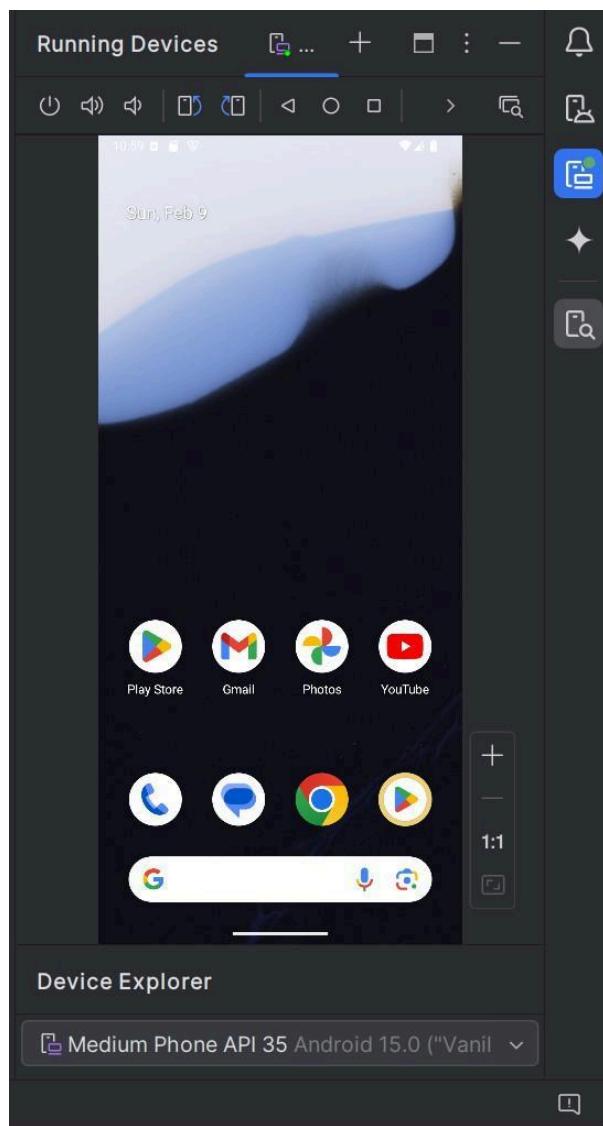
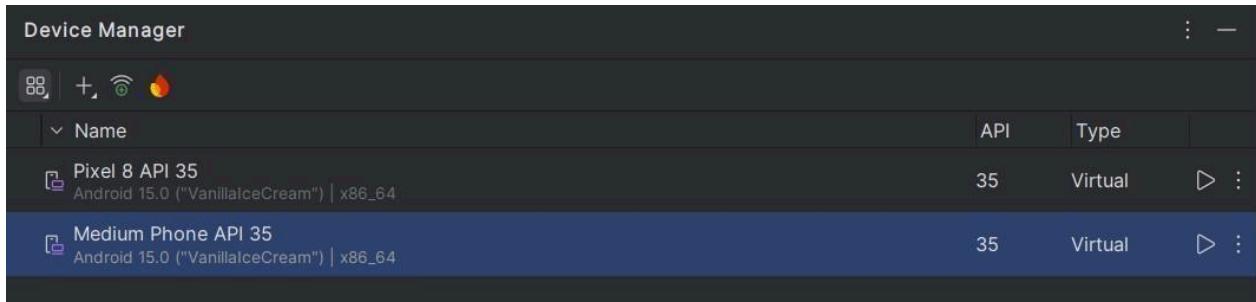
**Step 10:** - Next, you need to set up an Android emulator. It is responsible for running and testing the Flutter application



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



**Step 10.2: - Step 12: -** Go to File > New Project > Create Flutter Project, then select the project name and location, and click Next to proceed. Click on the icon pointed into the red color rectangle. The Android emulator displayed as below screen



# MAD & PWA Lab

## Journal

Experiment No.	02
Experiment Title.	To design Flutter UI by including common widgets.
Roll No.	01
Name	ADITYA SAMPATH KUMAR
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 2

AIM: To design Flutter UI using common widgets

## THEORY:

Flutter is Google's open-source UI toolkit for building natively compiled applications for mobile, web, and desktop from a single codebase. One of Flutter's core strengths lies in its widget-based architecture, where everything—layout, styling, and even app structure—is built using widgets.

### Widgets in Flutter

Widgets are the building blocks of a Flutter app's UI. Flutter provides a rich set of pre-designed widgets that can be composed, customized, and nested to build complex user interfaces. Widgets in Flutter fall into two main categories:

- Stateless Widgets: These are immutable and do not store any state. They are redrawn only when their parent changes.
- Stateful Widgets: These maintain state across rebuilds and can dynamically change during the app's lifecycle.

### Common Flutter Widgets

1. Container  
Used to create rectangular visual elements. It supports styling, padding, margins, borders, and more.
2. Text  
Displays a string of text with optional styling like font size, color, weight, etc.
3. Row and Column  
Used for horizontal (Row) and vertical (Column) layouts. They are essential for UI structuring.
4. Image  
Displays images from assets, network, or memory.

5. ElevatedButton / TextButton / IconButton  
Various buttons used to handle user interactions.
6. Scaffold  
Provides a basic structure for material design apps. It supports elements like AppBar, Drawer, BottomNavigationBar, etc.
7. ListView  
Used to display scrollable lists of widgets dynamically or statically.
8. TextField  
Used for accepting user input.
9. Stack  
Allows overlapping widgets (e.g., for adding a floating action button over an image).
10. Card  
Creates a material design card with rounded corners and shadows.

CODE:

```

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

class CustomAppBar extends StatelessWidget implements PreferredSizeWidget {
  final String title;
  final GlobalKey<ScaffoldState> scaffoldKey;

  const CustomAppBar({super.key,
    required this.title, required
    this.scaffoldKey});

  @override
  Widget build(BuildContext context)
{



  return AppBar(
    backgroundColor: Colors.black,
    title: Text(title, style:
    const TextStyle(color:
    Colors.white)),
    actions: [
      IconButton(icon: const
      Icon(Icons.search, color:
      Colors.white), onPressed: () {}),
      Padding(
        padding: const
        EdgeInsets.symmetric(horizontal:
        8.0),
        child: GestureDetector(
          onTap: () {
            scaffoldKey.currentState?.openEndDrawer(); // Opens right-side drawer
          }
        )
      )
    ]
  );
}

```

```
        },
        child: const
CircleAvatar(
    backgroundColor:
Colors.blue,
    child: Text("R",
style: TextStyle(color:
Colors.white)),
),
),
),
],
);
}

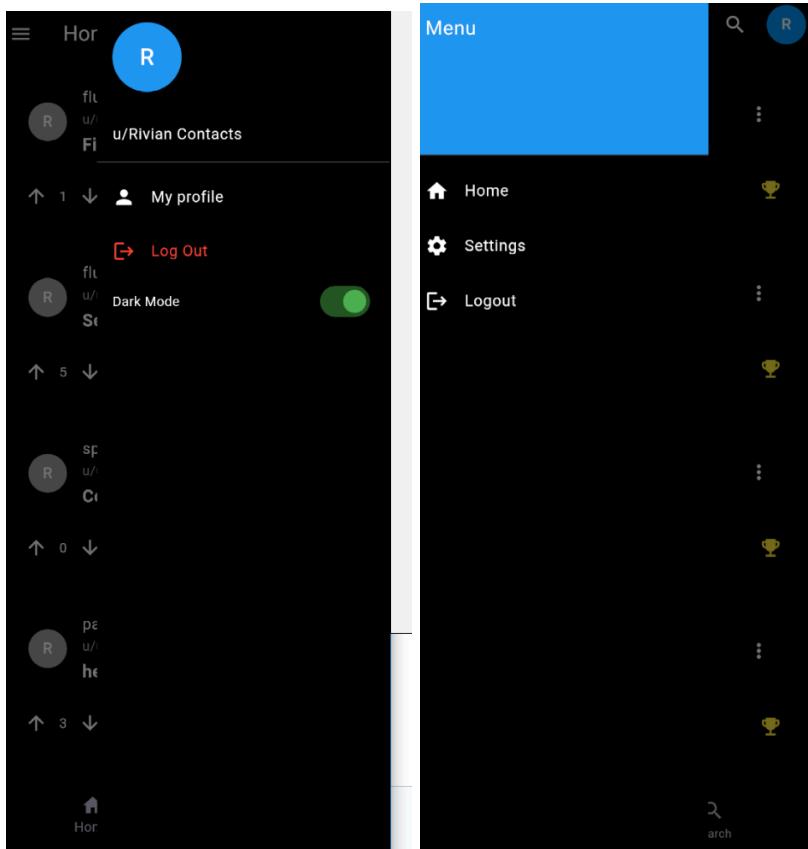
@Override
Size get preferredSize => const
Size.fromHeight(kToolbarHeight);
}

custom_drawer.dart:
import
'package:flutter/material.dart';
import
'package:reddit_clone/screens
/login_screen.dart'; //
Import your login screen

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

  @override
  Widget build(BuildContext
context) {
    return Drawer(
        backgroundColor:
Colors.black,
        child: ListView(
            padding:
EdgeInsets.zero,
            children: [
                const DrawerHeader(
                    decoration:
BoxDecoration(color:
Colors.blue),
                    child:
Text("Menu", style:
TextStyle(color:
Colors.white, fontSize: 20)),
),
                ListTile(
                    leading: const
Icon(Icons.home, color:
Colors.white),
                    title: const
Text("Home", style:
TextStyle(color:
Colors.white)),
                    onTap: () {}),
),
                ListTile(
                    leading: const
Icon(Icons.settings, color:
Colors.white),
                    title: const
Text("Settings", style:
TextStyle(color:
Colors.white)),
                    onTap: () {}),
),
            ],
        ),
    );
}
```

## OUTPUT:



# MAD & PWA Lab

## Journal

Experiment No.	03
Experiment Title.	To include icons, images, fonts in Flutter app
Roll No.	01
Name	ADITYA SAMPATH KUMAR
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 3

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

## THEORY:

Flutter allows developers to create rich and visually engaging user interfaces by easily incorporating **icons, images, and custom fonts** into their applications. These elements help in enhancing the look and feel of an app, improving user experience and branding consistency.

---

### 1. Icons in Flutter

Icons are graphical representations used to indicate actions, features, or status. Flutter provides multiple ways to use icons:

- **Built-in Material Icons**

Flutter includes a large collection of Material Design icons via the `Icons` class.

Example: `Icons.home`, `Icons.search`

- **Custom Icons**

Developers can also include custom icons using `.ttf` files or icon packs like

**Font Awesome** via packages (e.g., `flutter_vector_icons`,  
`font_awesome_flutter`).

### Usage Example:

```
Icon(Icons.favorite, color: Colors.red)
```

---

### 2. Images in Flutter

Images are a key visual component in any app, used for branding, illustrations, or visual communication. Flutter supports:

- **Asset Images**

Stored locally in the project under the assets directory.

- **Network Images**

Loaded from a web URL.

- **Memory or File Images**

Loaded from the device's memory or file system.

---

### Benefits of Including Icons, Images, and Fonts

- **Improved UI/UX:** Enhances app design and user interaction.
- **Brand Identity:** Helps create a consistent brand look and feel.
- **Engagement:** Visually appealing content keeps users interested.

### CODE:

```
login_screen.dart:  
import 'package:flutter/material.dart';  
import 'package:firebase_auth/firebase_auth.dart';  
import 'package:cloud_firestore/cloud_firestore.dart';  
import '../features/home/home_page.dart';  
  
class RedditLoginPage extends StatefulWidget {  
  const RedditLoginPage({super.key});  
  
  @override  
  _RedditLoginPageState createState() =>  
  _RedditLoginPageState();  
  
  class _RedditLoginPageState extends State<RedditLoginPage> {  
    final TextEditingController _usernameController =  
    TextEditingController();  
    final TextEditingController _passwordController =  
    TextEditingController();  
  }  
}
```

```

Future<void> _signIn() async {
    final username =
    _usernameController.text.trim();
    final password =
    _passwordController.text.trim();

    if (username.isNotEmpty &&
    password.isNotEmpty) {

Image.asset('assets/reddit_logo.png'
, height: 40),
    ],
    ),
    const SizedBox(height:
50),
    Image.asset(
'assets/reddit_mascot.png',
    height: 200,
    ),
    onPressed:
    _signUp,
    style:
ElevatedButton.styleFrom(
    backgroundColor: Colors.blueAccent,
    padding:
const EdgeInsets.symmetric(vertical:
12, horizontal: 100),
    shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(8)),
    ),
    child: const
Text("Sign Up", style:
TextStyle(fontSize: 18, color:
Colors.white)),
    ),
    const
SizedBox(height: 10),
}
}

ElevatedButton(
    onPressed:
    _signIn,
    style:
ElevatedButton.styleFrom(
    backgroundColor: Colors.green,
    padding:
const EdgeInsets.symmetric(vertical:
12, horizontal: 108),
    shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(8)),
    ),
    child: const
Text("Sign In", style:
TextStyle(fontSize: 18, color:
Colors.white)),
    ),
    ],
    ),
    );
}
}

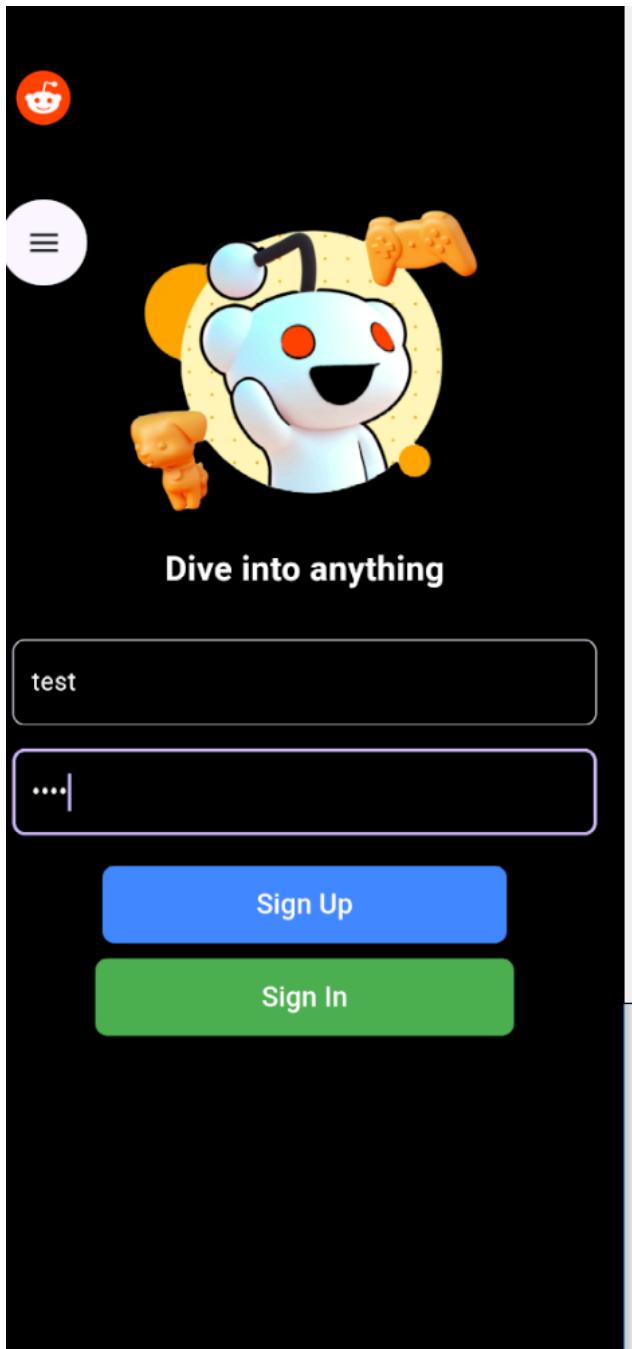
search_page.dart:
import
'package:flutter/material.dar
t';
import
'subreddit_screen.dart';

class SearchPage extends
 StatelessWidget {

```

```
const SearchPage({super.key});  
  
@override  
Widget build(BuildContext context) {  
    final List<Map<String, dynamic>> communities = [  
        {"rank": 415, "name": "r/patientgamers", "members": "512"},  
        {"rank": 416, "name": "r/spikes", "members": "904"},  
        {"rank": 417, "name": "r/Costco", "members": "28.4k"},  
        {"rank": 418, "name": "r/Cartalk", "members": "28.1k"},  
        {"rank": 419, "name": "r/3amjokes", "members": "28.7k"},  
    ];  
  
    return MaterialPageRoute(  
        builder: (context) => SubredditScreen(  
            subredditName: community['name'],  
            ),  
            ),  
            );  
            },  
            );  
            );  
            );  
            );  
    }  
}
```

## OUTPUT:



# MAD & PWA Lab

## Journal

Experiment No.	04
Experiment Title.	To create an interactive Form using form widget
Roll No.	01
Name	ADITYA SAMPATH KUMAR
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: - 04

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

### **THEORY:**

Forms are an essential part of many mobile applications. They are used to collect user input such as login credentials, contact information, feedback, and more. Flutter provides powerful and flexible form-related widgets that help developers create interactive and user-friendly forms.

The Form widget in Flutter acts as a container for grouping and managing multiple input fields. It simplifies form validation, state management, and submission handling.

---

#### 1. Importance of Forms in Apps

Forms enable users to interact with an application by providing input. Examples include:

- Logging in or signing up
- Filling out feedback or survey forms
- Entering personal or payment details
- Submitting contact or registration information

An interactive form enhances usability by providing feedback, validation, and user guidance.

---

#### 2. Key Form-Related Widgets in Flutter

- Form: A wrapper widget that groups form fields and manages validation.
- TextFormField: A common input field used for taking user text input.
- FormState: Used to validate and save the form's state.
- GlobalKey: Often used to uniquely identify the form and access its state.

### 3. Features of an Interactive Form

- Validation: Ensures user inputs meet defined criteria (e.g., valid email or non-empty password).
  - Auto-Saving: Saves the input values on form submission.
  - Feedback: Provides real-time error messages or success indicators.
  - Focus Control: Allows navigation between input fields for a smooth experience.
- 

### 4. Benefits of Using Form Widget

- Structured Input Handling: Keeps the input organized and manageable.
  - Built-in Validation Support: Reduces the effort needed to implement checks manually.
  - Improved User Experience: Ensures accurate input and guides users through the form process.
  - Reusability: Makes it easier to maintain and reuse form components across the app.
- 

## Conclusion

Creating an interactive form using the Form widget in Flutter provides a clean and efficient way to gather user input, perform validation, and manage form state. It is an essential component for building user-driven mobile applications that require structured input and smooth interactivity.

## **CODE:**

**main.dart :-**

```
import 'package:flutter/material.dart';
import 'package:reddit/features/auth/screens/login_screen.dart';
```

```

import 'theme/pallete.dart';

void main() {
  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(
      debugShowCheckedModeBanner: false,
      title: 'Reddit Clone',
      theme: Pallete.darkModeAppTheme, home:
      const LoginScreen(),
    );
}
}
}

```

**login\_screen.dart :-**

```

import 'package:flutter/material.dart';
// import
'package:flutter_riverpod/flutter_riverpod.dart';
// import
'package:reddit/core/common/loader'
'package:reddit/core/common/sign_in_button.dart';
import
'package:reddit/core/constants/consts.dart';
// import
'package:reddit/features/auth/controller/auth_controller.dart';
// import
'package:reddit/responsive/responsive.dart';

// class LoginScreen
extends ConsumerWidget {

```

```

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

  // void
  signInAsGuest(WidgetRef ref, BuildContext context) {
    //
    ref.read(authControllerProvider.notifier).signInAsGuest(context);
  }

  @override
  // Widget
  build(BuildContext context, WidgetRef ref) {
    Widget build(BuildContext context) {
      // final isLoading =
      ref.watch(authControllerProvider);
      return
        Scaffold(

```

**Project Title: Reddit****Roll No. 01**

```
appBar:  
AppBar(  
title:  
Image.asset  
(  
  Constants.log  
oPath,  
height: 40,  
,  
act  
io  
ns  
:  
[  
Te  
xt  
Bu  
tt  
on  
(  
  //  
onPressed: () =>  
signInAsGuest(ref,  
context),  
  onPressed:  
() => {},  
  child: const  
Text( n: [  
  const  
SizedBox(height:  
30), const Text(  
'Dive into  
anything',  
style:  
TextStyle(  
fontSize: 24,  
fontWeight:  
FontWeight.bold,  
  letterSpacing:  
0.5,  
,  
  const  
SizedBox(height:  
30), Padding(  
padding  
: const  
EdgeInsets.all(8.  
0),  
  child:  
Image.asset(  
  
Constants.loginEmotePath,  
} height: 400,  
}  
}
```

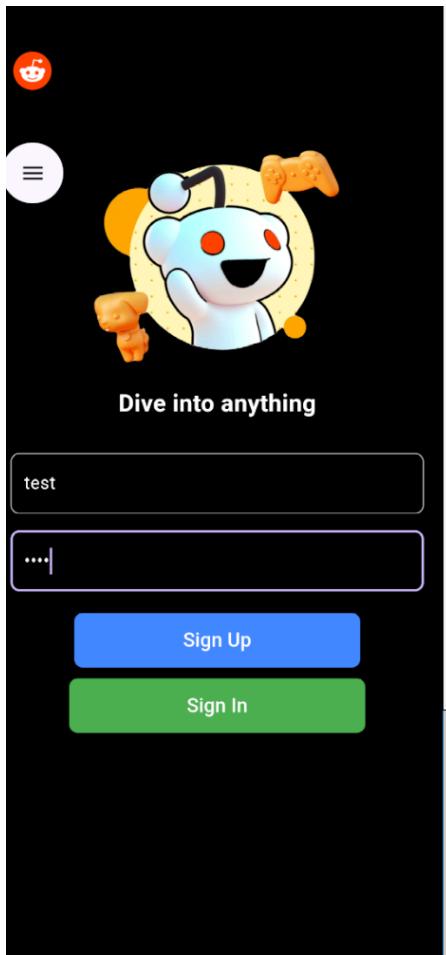
```
 ),  
,
```

**sign\_in\_button.dart :-**

```
import  
'package:flutter/material.dart';  
// import  
'package:flutter_riverpod/flutter_r  
ive_riverpod.dart';  
import  
'package:reddit/core/constants/cons  
stants.dart';  
// import  
'package:reddit/features/auth/contr  
oller/auth_controller.dart';  
import  
'package:reddit/theme/pallete.dart'  
;  
  
// class SignInButton extends  
ConsumerWidget {  
class SignInButton  
extends StatelessWidget {  
final bool isFromLogin;  
const SignInButton({Key? key,  
this.isFromLogin = true}) :  
super(key: key);  
  
// void  
signInWithGoogle(BuildContext  
context, WidgetRef ref) {  
//  
ref.read(authControllerProvider.not  
ifi er).signInWithGoogle(context,  
isFromLogin);  
// }  
  
@override  
// Widget build(BuildContext  
context, WidgetRef ref) {  
Widget build(BuildContext context)  
{ return Padding(  
padding: const  
EdgeInsets.all(18.0),  
child: ElevatedButton.icon(  
// onPressed: () =>  
signInWithGoogle(context, ref),  
onPressed: () =>  
{}, icon:  
Image.asset(Con  
stants.googlePath,  
width: 35,  
),  
label: const Text(  
'Continue with Google',  
style: TextStyle(fontSize:  
18),  
,  
style:  
ElevatedButton.styleFrom(  
backgroundColor:  
Pallete.greyColor,  
minimumSize: const  
Size(double.infinity, 50),
```

```
shape: RoundedRectangleBorder(      ),  
       borderRadius:      );  
BorderRadius.circular(20),      );  
)  
,
```

**OUTPUT :-**



# MAD & PWA Lab

## Journal

Experiment No.	05
Experiment Title.	To apply navigation, routing and gestures in Flutter App
Roll No.	01
Name	ADITYA SAMPATH KUMAR
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 5

AIM: To navigation, routing, and gestures in flutter ap

## THEORY:

In Flutter, building interactive and multi-screen applications involves the use of navigation, routing, and gesture handling. These features are essential for creating user-friendly, responsive, and intuitive mobile applications.

---

### 1. Navigation in Flutter

Navigation refers to the ability to move between different screens or pages in an app. Flutter uses a stack-based navigation system, where each new screen is placed on top of a stack. When the user wants to return to the previous screen, the current screen is removed (or "popped") from the stack.

Navigation helps manage the flow of an app by enabling users to move from a login screen to a home screen, open a settings page, or return to the previous screen.

---

### 2. Routing in Flutter

Routing is the process of defining how users navigate between screens. Flutter supports two main routing techniques:

- Basic Routing: Involves directly specifying the screen to navigate to.
- Named Routing: Uses predefined route names to manage navigation in a structured way, especially in apps with many screens.

Routing ensures that screen transitions are organized and manageable, and it helps maintain code readability and scalability in large applications.

---

### 3. Gestures in Flutter

Gestures are physical interactions made by the user, such as tapping, swiping, pinching, or long-pressing on the screen. Flutter provides robust support for gesture detection, allowing developers to define responses to specific user actions.

Gesture handling improves the user experience by making the app more dynamic and responsive. For example, a swipe can be used to delete an item, or a long press might show additional options.

---

### Importance of Navigation, Routing, and Gestures

- User Flow: Navigation and routing create a logical and smooth flow through the app.
- Usability: Gestures provide intuitive ways for users to interact with the interface.
- Scalability: Routing helps manage transitions and maintain code in larger applications.
- Interactivity: Gestures contribute to a richer, more engaging user experience.

### CODE:

```
search_page.dart:  
import 'package:flutter/material.dart';  
import 'subreddit_screen.dart';  
  
class SearchPage extends StatelessWidget {  
  const SearchPage({super.key});  
  
  @override  
  Widget build(BuildContext context)  
  {  
    final List<Map<String, dynamic>>  
    communities = [  
      {"rank": 415, "name": "r/patientgamers", "members": "512"},  
      {"rank": 416, "name": "r/spikes", "members": "904"},  
      {"rank": 417, "name": "r/Costco", "members": "28.4k"},  
      {"rank": 418, "name": "r/Cartalk", "members": "28.1k"},  
      {"rank": 419, "name": "r/3amjokes", "members": "28.7k"},  
      {"rank": 421, "name": "r/oculus", "members": "27.7k"},  
    ];  
    return Scaffold(...);  
  }  
}
```

```
        {"rank": 422, "name": "r/liberalgunowners", "members": "24.6k"},  
        {"rank": 423, "name": "r/TrollMUA", "members": "22.5k"},  
    ];  
  
    return Scaffold(  
        backgroundColor: Colors.black,  
        appBar: AppBar(  
            title: const Text("Search"),  
            backgroundColor:  
Colors.black,  
        ),  
        body: ListView.builder(  
            itemCount:  
communities.length,  
            itemBuilder: (context,  
index) {  
                final community =  
communities[index];  
                return ListTile(  
                    leading: CircleAvatar(  
                        backgroundColor:  
Colors.grey[800],  
                        child: Text(  
  
community['rank'].toString(),  
                        style: const  
TextStyle(color: Colors.white),  
                    ),  
                    title: Text(  
                        community['name'],  
                        style: const  
TextStyle(color: Colors.white,  
fontSize: 18),  
                    ),  
                    trailing: Text(  
                        community['members'],  
                );  
            },  
        ),  
    );  
  
    return Scaffold(  
        backgroundColor: Colors.black,  
        appBar: AppBar(  
            title: const Text("Search"),  
            backgroundColor:  
Colors.black,  
        ),  
        body: ListView.builder(  
            itemCount:  
communities.length,  
            itemBuilder: (context,  
index) {  
                final community =  
communities[index];  
                return ListTile(  
                    leading: CircleAvatar(  
                        backgroundColor:  
Colors.grey[800],  
                        child: Text(  
  
community['rank'].toString(),  
                        style: const  
TextStyle(color: Colors.white),  
                    ),  
                    title: Text(  
                        community['name'],  
                        style: const  
TextStyle(color: Colors.white,  
fontSize: 18),  
                    ),  
                    trailing: Text(  
                        community['members'],  
                );  
            },  
        ),  
    );  
}
```

```

return Scaffold(
  backgroundColor: Colors.black,
  appBar: AppBar(
    backgroundColor:
Colors.black,
    elevation: 0,
    leading: IconButton(
      icon: const
Icon(Icons.arrow_back, color:
Colors.white),
      onPressed: () =>
Navigator.pop(context),
    ),
  ),
  body: Column(
    crossAxisAlignment:
CrossAxisAlignment.start,
    children: [
      Container(
        height: 120,
        decoration: const
BoxDecoration(
          gradient:
LinearGradient(
            colors: [Colors.red,
Colors.orange, Colors.yellow],
            begin:
Alignment.topCenter,
            end:
Alignment.bottomCenter,
          ),
        ),
      ),
      Padding(
        padding: const
EdgeInsets.all(16.0),
        child: Row(
          children: [
            const CircleAvatar(
              backgroundColor:
Colors.grey,
            ),
            child:
Icon(Icons.reddit, color:
Colors.white),
          ],
        ),
      ),
      const
SizedBox(width: 12),
      Text(
        subredditName,
        style: const
TextStyle(
          color:
Colors.white,
          fontSize: 20,
          fontWeight:
FontWeight.bold),
      ),
      const Spacer(),
      ElevatedButton(
        onPressed: () {
          async {
            final user =
FirebaseAuth.instance.currentUser;
            if (user != null) {
              final userDoc =
FirebaseFirestore.instance.collection('users').doc(user.uid);
              await userDoc.set({
                'followedCommunities':
FieldValue.arrayUnion([subredditName]),
              },
              SetOptions(merge: true));
            }
          },
        },
        style:
ElevatedButton.styleFrom(backgroundC
olor: Colors.grey[800]),
      ),
    ],
  ),
);

```

```

        child: const
Text("Follow", style:
TextStyle(color: Colors.white)),
),
const
SizedBox(width: 8),
ElevatedButton(
 onPressed: () {},
style:
ElevatedButton.styleFrom(
backgroundColor: Colors.grey[800]),
child: const
Text("Mod Tools",
style:
TextStyle(color: Colors.white)),
),
],
),
const Divider(color:
Colors.grey),
Expanded(
child: Padding(
padding: const
EdgeInsets.all(16.0),
child: Column(
crossAxisAlignment:
CrossAxisAlignment.start,
children: [
Text(

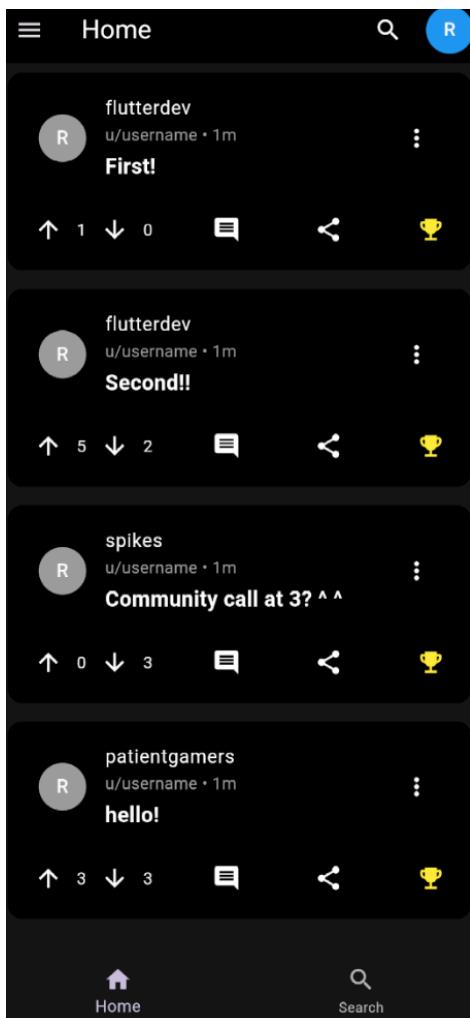
```

```

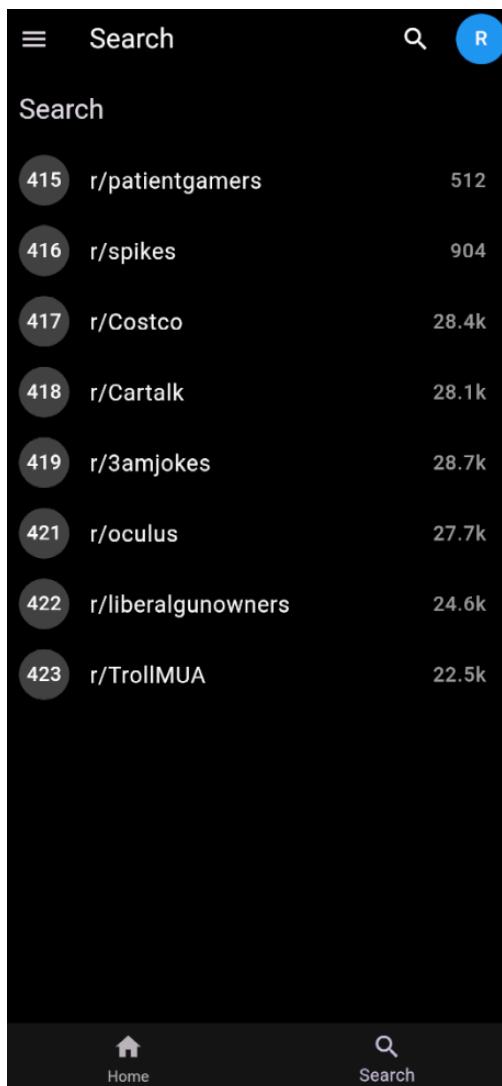
"Lorem Ipsum
History",
style: const
TextStyle(
color:
Colors.white,
fontSize:
18,
fontWeight:
FontWeight.bold),
),
const
SizedBox(height: 8),
Text(
"Contrary to
popular belief, Lorem Ipsum is not
simply random text...", style: const
TextStyle(color: Colors.grey),
),
],
),
),
],
),
);
}
}

```

OUTPUT:



**after clicking on search icon or using the swipe gesture:**



# MAD & PWA Lab

## Journal

Experiment No.	06
Experiment Title.	To Connect Flutter UI with fireBase database
Roll No.	01
Name	ADITYA SAMPATH KUMAR
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO3: Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS
Grade:	

# EXPERIMENT 6

AIM: To connect Flutter UI with firebase

## THEORY:

Firebase is a comprehensive backend-as-a-service (BaaS) platform developed by Google. It provides a wide range of tools and services such as authentication, real-time databases, cloud storage, and analytics to help developers build and scale modern mobile and web applications.

Integrating Firebase with Flutter allows developers to enhance their apps with powerful backend capabilities while focusing on building rich user interfaces using Flutter.

---

### 1. Why Use Firebase with Flutter?

- Real-time Functionality: Firebase offers real-time database and cloud-based data synchronization, making it ideal for chat apps, live feeds, and collaborative apps.
- User Authentication: Secure user sign-in and sign-up using email/password, phone number, or social providers (Google, Facebook, etc.).
- Data Storage: Store structured data in Firestore or files (like images and videos) in Firebase Cloud Storage.
- Notifications: Send push notifications using Firebase Cloud Messaging (FCM).
- Analytics & Crash Reporting: Monitor user behavior and app performance in real-time.

---

### 2. Common Firebase Services Used in Flutter Apps

- Firebase Authentication: For managing user login and identity verification.
- Cloud Firestore: A scalable NoSQL database used to store and sync data in real time.

- Firebase Realtime Database: A JSON-based database with real-time syncing.
  - Firebase Storage: For uploading and retrieving media files like images and videos.
  - Firebase Cloud Messaging: Enables sending push notifications to user devices.
- 

### 3. Benefits of Connecting Flutter UI with Firebase

- Cross-Platform Support: Firebase works seamlessly with Android, iOS, and web, which fits perfectly with Flutter's cross-platform nature.
  - Scalability: Easily scales from a small prototype to a large production app.
  - Security: Firebase provides built-in security rules and user authentication.
  - Time-Saving: Reduces the need to build a backend from scratch, speeding up development.
- 

### Conclusion

Connecting Flutter UI with Firebase combines the power of a beautiful front-end framework with a robust and feature-rich backend platform. This integration allows developers to build modern, responsive, and scalable apps quickly and efficiently, while still offering advanced features like real-time updates, user management, and secure data handling.

### CODE:

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

```

class PostList extends StatelessWidget {
  final List<String> followedCommunities;

  const PostList({super.key,
    required this.followedCommunities});

  @override
  Widget build(BuildContext context)
  {
    return
      followedCommunities.isEmpty
        ? const Center(
          child: Text(
            "Follow communities to
            see posts!",
            style:
            TextStyle(color: Colors.white),
          ),
        )
        : StreamBuilder(
          stream:
          FirebaseFirestore.instance
            .collection('communities')
              .where('name',
               whereIn: followedCommunities)
              .snapshots(),
          builder: (context,
            snapshot) {
            if (!snapshot.hasData
            || snapshot.data!.docs.isEmpty) {
              return const
                Center(child:
                  CircularProgressIndicator());
            }
            final communities =
            snapshot.data!.docs;
            return FutureBuilder(
              future: Future.wait(
                communities.map((communityDoc) async {
                  final
                  postsSnapshot =
                  await
                  communityDoc.reference.collection('p
                  osts').get();
                  return
                  postsSnapshot.docs.map((postDoc) {
                    return {
                      'post':
                      postDoc,
                      'communityId': communityDoc.id,
                    };
                  }).toList();
                })),
                builder: (context,
                  futureSnapshot) {
                  if
                  (!futureSnapshot.hasData) {
                    return const
                    Center(child:
                      CircularProgressIndicator());
                  }
                  final allPosts =
                  futureSnapshot.data!.expand((posts)
                    => posts).toList();
                  if
                  (allPosts.isEmpty) {
                    return const
                    Center(
                      child: Text(
                        "No posts
                        found in your communities!");
                  }
                });
              });
            }
          );
        );
      );
    );
  }
}

```

```

        style:           );
        ),
        );
    }
}

return
ListView.builder(
    itemCount:
allPosts.length,
    itemBuilder:
(context, index) {
    final postData
= allPosts[index];
    final post =
postData['post'] as
DocumentSnapshot?;
    final
communityId =
postData['communityId'] as String?;

    if (post ==
null || communityId == null) {
        return const
SizedBox.shrink(); // Skip invalid
posts
    }

    return
PostCard(
    postId:
post.id ?? '',
    subredditId:
communityId,
    title:
post['title'] ?? 'Untitled Post',
    upvotes:
(post['upvotes'] ?? 0) as int,
    downvotes:
(post['downvotes'] ?? 0) as int,
    );
},
)
);
),
),
);
}
}

build.gradle:
plugins {
    id "com.android.application"
    id "kotlin-android"
    id
'com.google.gms.google-services' // //
Firebase Plugin
    id
"dev.flutter.flutter-gradle-plugin"
// Keep this last
}

android {
    namespace =
"com.android.reddit_clone"
    compileSdkVersion 34
    ndkVersion = flutter.ndkVersion

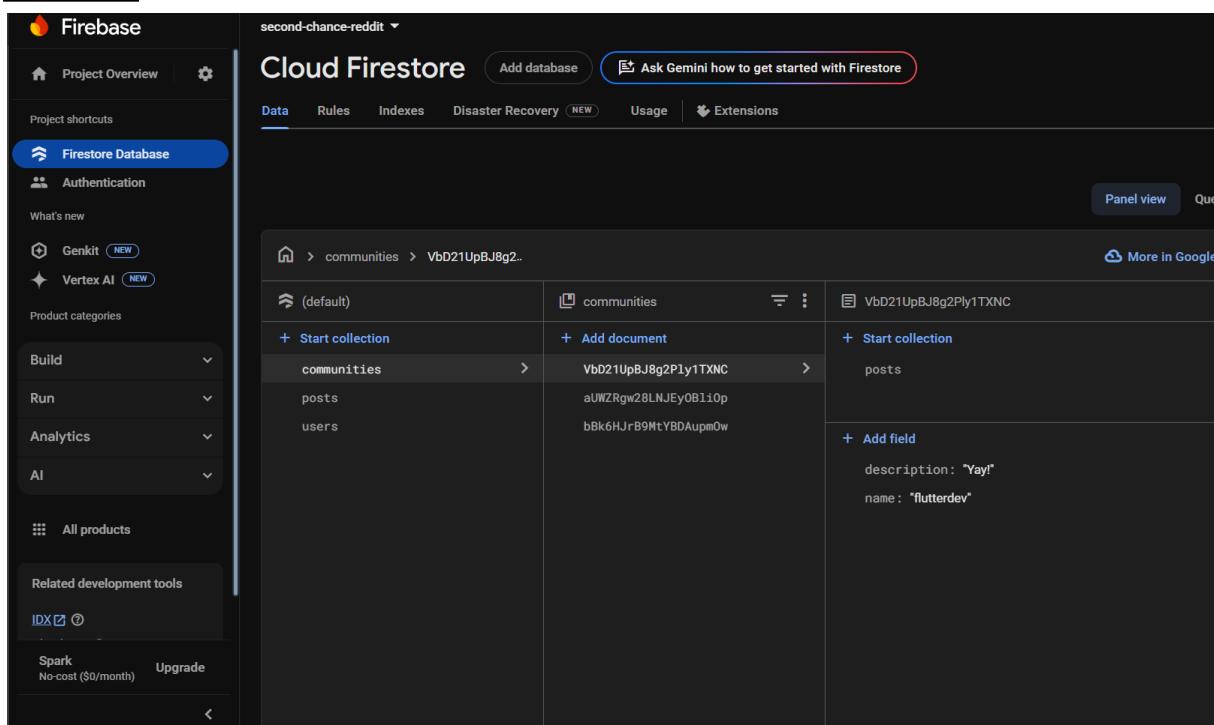
    compileOptions {
        sourceCompatibility =
JavaVersion.VERSION_1_8
        targetCompatibility =
JavaVersion.VERSION_1_8
    }

    kotlinOptions {
        jvmTarget =
JavaVersion.VERSION_1_8
    }

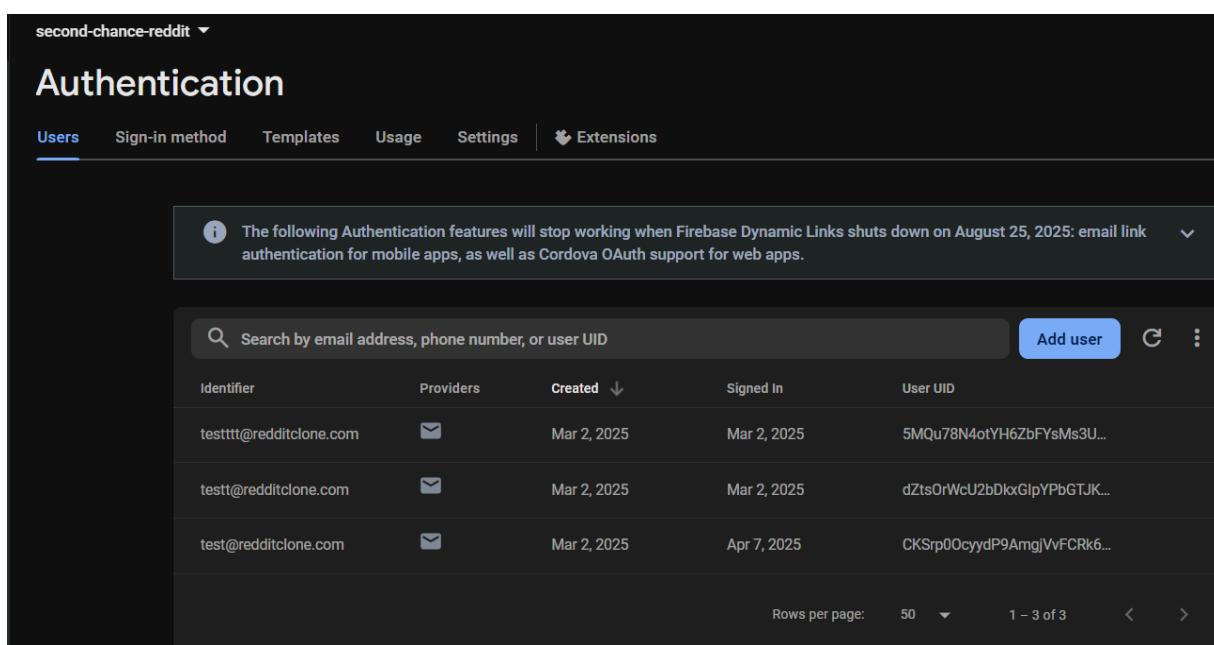
    defaultConfig {
        // TODO: Specify your own
unique Application ID (htt

```

```
ps://developer.android.com/studio/bu  
ild/application-id.html).  
    applicationId =  
"com.android.reddit_clone"  
        // You can update the  
following values to match your  
application needs.  
        // For more information,  
see:  
https://flutter.dev/to/review-gradle  
-config.  
        minSdkVersion 23  
        targetSdkVersion 33  
        multiDexEnabled true  
  
        versionCode =  
flutter.versionCode  
        versionName =  
flutter.versionName  
    }  
  
    buildTypes {  
        release {  
            // TODO: Add your own  
signing config for the release  
build.  
            // Signing with the  
debug keys for now, so `flutter run  
--release` works.  
            signingConfig =  
signingConfigs.debug  
        }  
    }  
}  
  
flutter {  
    source = "../.."  
}  
  
dependencies {  
    // Import Firebase BoM (Bill of  
Materials)
```

OUTPUT:


The screenshot shows the Firebase Cloud Firestore interface for a project named "second-chance-reddit". The left sidebar is collapsed, showing options like Project Overview, Authentication, Genkit, Vertex AI, Build, Run, Analytics, AI, All products, and Related development tools. The main area displays a hierarchical document structure under the "communities" collection. A specific document, "VbD21UpBJ8g2Ply1TXNC", is expanded, showing its fields: posts (with document ID "aUWZRgw28LNJEy0B1i0p") and users (with document ID "bBk6HJrB9MtYBDAupm0w"). The "posts" field contains a single field "description" with the value "Yay!" and another field "name" with the value "flutterdev".

The screenshot shows the Firebase Authentication interface for the same project. The top navigation bar includes tabs for Users, Sign-in method, Templates, Usage, Settings, and Extensions. A warning message in a box states: "The following Authentication features will stop working when Firebase Dynamic Links shuts down on August 25, 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps." Below this, a search bar and an "Add user" button are visible. A table lists three users: "testttt@redditclone.com", "testt@redditclone.com", and "test@redditclone.com", along with their provider (email), creation date (Mar 2, 2025), sign-in date, and user UID.

## 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.	01
Name	ADITYA SAMPATH KUMAR
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO4: Understand various PWA frameworks and their requirements
Grade:	

## EXPERIMENT 7

**Aim-**To write meta data of your Ecommerce PWA in a Web app manifest file to enable add to home screen feature.

### THEORY -

#### **Regular Web App**

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

#### **Progressive Web App**

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

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

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

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

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

## **2. Ease of Access**

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

## **3. Faster Services**

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

## **4. Engaging Approach**

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

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

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

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

## **6. Discoverable**

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

## **7. Lower Development Cost**

Progressive web apps can be installed on the user device like a native device, but it does not demand submission on an App Store. This makes it far more

cost-effective than native mobile applications while offering the same set of functionalities.

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

The main features are:

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

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

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

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

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

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

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

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

**Linkable** — Easily shared via URL without complex installations.

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

**Weaknesses refer to:**

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

**Code: -**

```
//Manifest.json:  
{  
  "short_name": "Movie Rec",  
  "name": "Movie Recommendation  
App", "icons": [  
    {  
      "src": "favicon.ico",  
      "sizes": "64x64 32x32 24x24 16x16",  
      "type": "image/x-icon"  
    },  
    {  
      "src": "logo192.png",  
      "type": "image/png",  
      "sizes": "192x192",  
      "purpose": "any  
maskable"  
    },  
    {  
      "src": "logo512.png",  
      "type":  
      "image/png",  
      "sizes": "512x512"  
    }  
  "start_url":  
  ".", "display":
```

```
"standalone",  
"theme_color": "#000000",  
"background_color": "#ffffff",
```

```
"description": "Find your next
favorite movie", "orientation":
"portrait-primary",
"categories": ["entertainment", "movies"]
}
```

### **//Serviceworker.js**

```
// Import workbox from CDN (you can also use the
workbox-webpack-plugin)

importScripts('https://storage.googleapis.com/workbox-
cdn/releases/6.4.1/workbox-sw.js');

// Precache and route setup
workbox.routing.registerRoute(
  ({request}) =>
  request.destination === 'image',
  new
  workbox.strategies.CacheFirst({
    cacheName: 'images',
    plugins: [
      new
      workbox.expiration.ExpirationPlugin(
        { maxEntries: 60,
          maxAgeSeconds: 30 * 24 * 60 * 60, // 30 Days
        }),
      ],
    })
  );

// Cache CSS and JavaScript
// Files
workbox.routing.registerRoute(
```

```
({request}) => request.destination === 'script' ||
  request.destination === 'style', new
  workbox.strategies.StaleWhileRevalidate({
    cacheName: 'static-resources',
  })
);

//index.html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width,
initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="descripti
      on"
      content="Movie recommendations app - find your next
      favorite film"
    />
    <link rel="apple-touch-icon"
      href="%PUBLIC_URL%/logo192.png" />
    <link rel="manifest"
      href="%PUBLIC_URL%/manifest.json" />
    <title>Movie Recommendations</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this
      app.</noscript>
  </body>
</html>
```

```
<div id="root"></div>  
</body>  
</html>
```

## Output

//Open folder in VS Code and run the website

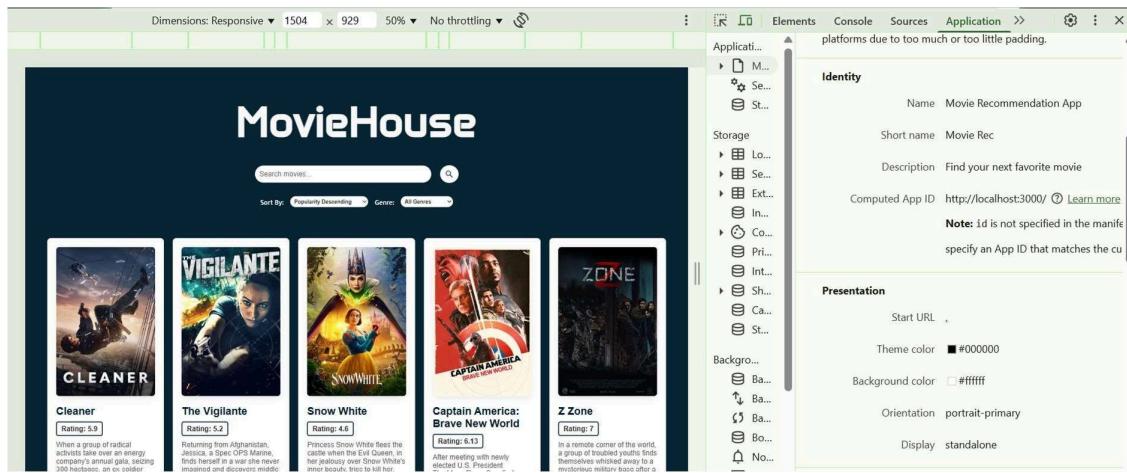
```
HOME@LAPTOP-9JIMM8I3 MINGW64 /e/New folder/Movie-Recommendation-App (main)
$ npm start

Compiled successfully!

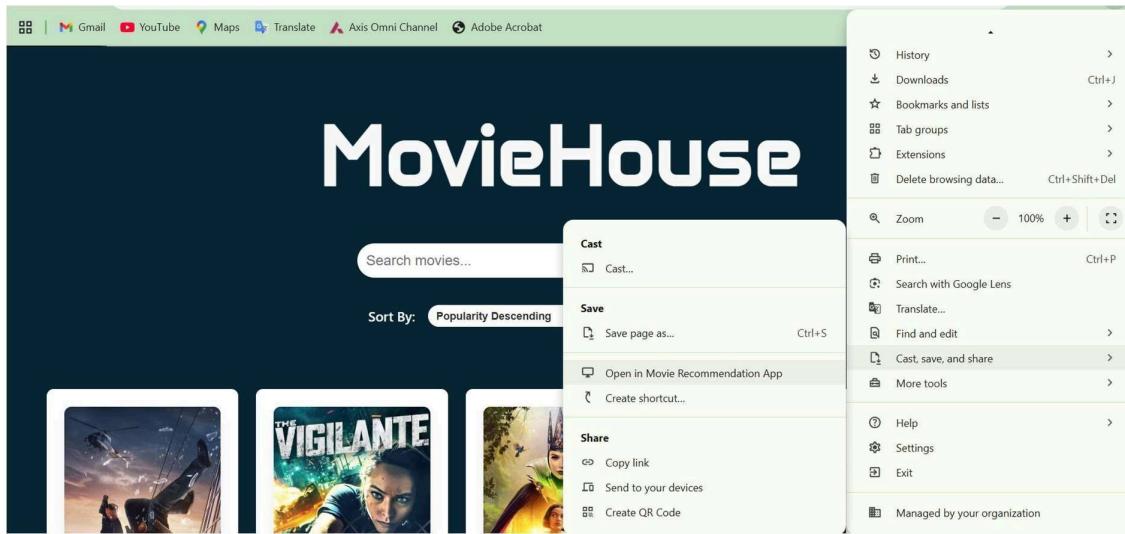
You can now view movie-recommendation-app in the browser.

Local: http://localhost:3000
On Your Network: http://192.168.162.1:3000
```

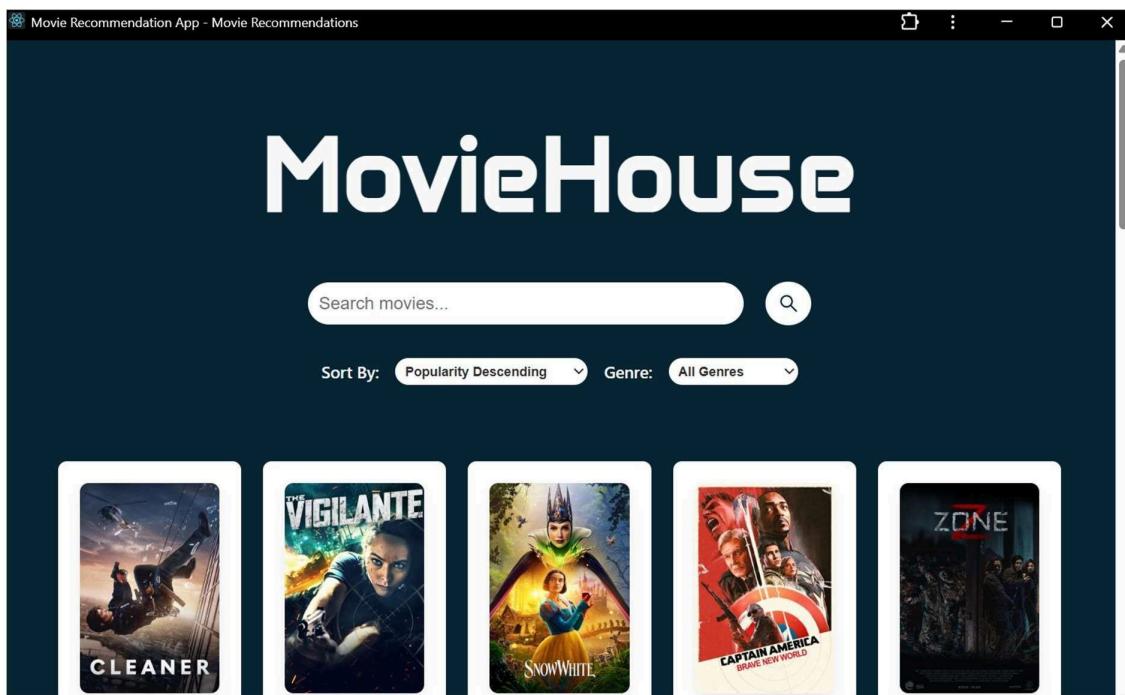
//open developers tools -> Applications



// Install the app by following this route Click on three dots on top right corner of the browser from app option ->Cast,share,save->Install the app



// This is the app



## 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.	01
Name	ADITYA SAMPATH KUMAR
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. 8

**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?

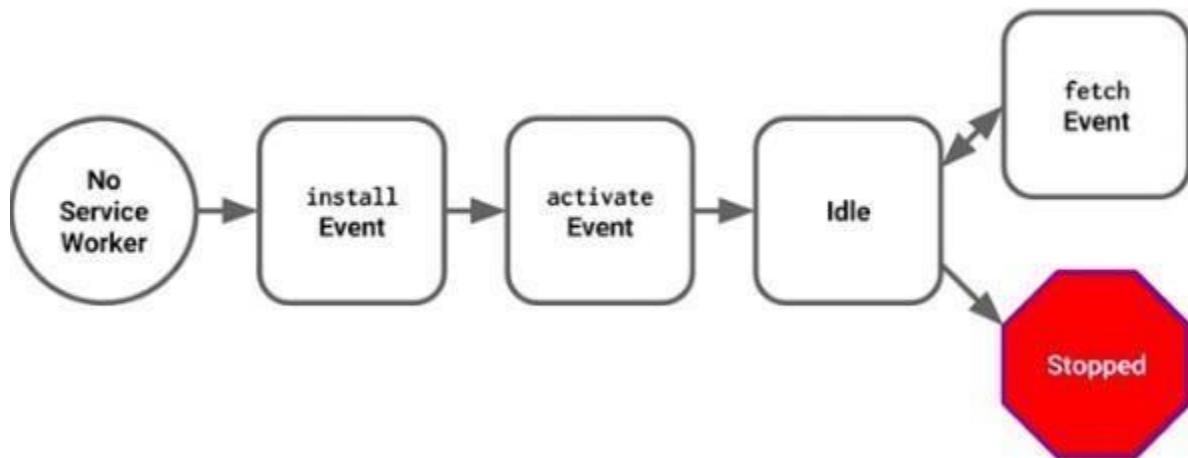
You can't access the **Window**

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



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: main.js

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

In this case we are setting the scope of the service worker to `/app/`, which means the service worker will control requests from pages like `/app/`, `/app/lower/` and `/app/lower/lower`, but not from pages like `/app` or `/`, which are higher.

If you want the service worker to control higher pages e.g. `/app` (without the trailing slash) you can indeed change the scope option, but you'll also need to set

the Service-Worker-Allowed HTTP Header in your server config for the request serving the service worker script.

```
main.js
navigator.serviceWorker.register('/app/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) {// 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-

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <meta name="theme-color" content="#000000" />
  <meta
    name="description"
    content="Movie recommendations app - find your next favorite film"
  />
  <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
  <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
  <title>Movie Recommendations</title>
</head>
```

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

```
service-worker.js
importScripts('https://storage.googleapis.com/workbox-cdn/releases/6.4.1/workbox-sw.js');

if (workbox) {
  console.log('✅ Workbox loaded');

  const CACHE_NAME = 'pwa-cache-v1';
  const PRECACHE_ASSETS = [
    '/',
    '/index.html',
    '/favicon.ico',
    '/logo192.png',
    '/logo512.png',
    '/manifest.json',
    '/static/css/main.css',
    '/static/js/main.js'
  ];

  // Precache assets manually
  workbox.precaching.precacheAndRoute(PRECACHE_ASSETS);

  // Cache images
  workbox.routing.registerRoute(
    ({ request }) => request.destination === 'image',
    new workbox.strategies.CacheFirst({
      cacheName: 'images-cache',
      plugins: [
        new workbox.expiration.ExpirationPlugin({
          maxEntries: 60,
          maxAgeSeconds: 30 * 24 * 60 * 60, // 30 Days
        }),
      ],
    })
  );
}
```

```
        ],
    })
);

// Cache CSS & JS
workbox.routing.registerRoute(
  ({ request }) => request.destination === 'script' || request.destination === 'style',
  new workbox.strategies.StaleWhileRevalidate({
    cacheName: 'static-resources',
  })
);

// Debug Fetch Requests
self.addEventListener('fetch', (event) => {
  console.log('[Service Worker] Fetching:', event.request.url);
  event.respondWith(
    caches.match(event.request).then((cachedResponse) => {
      return cachedResponse || fetch(event.request);
    })
  );
});

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

// Activate Event
self.addEventListener('activate', (event) => {
  console.log('[Service Worker] Activating...!');
  event.waitUntil(
    caches.keys().then((cacheNames) => {
      return Promise.all(
        cacheNames.map((cache) => {
```

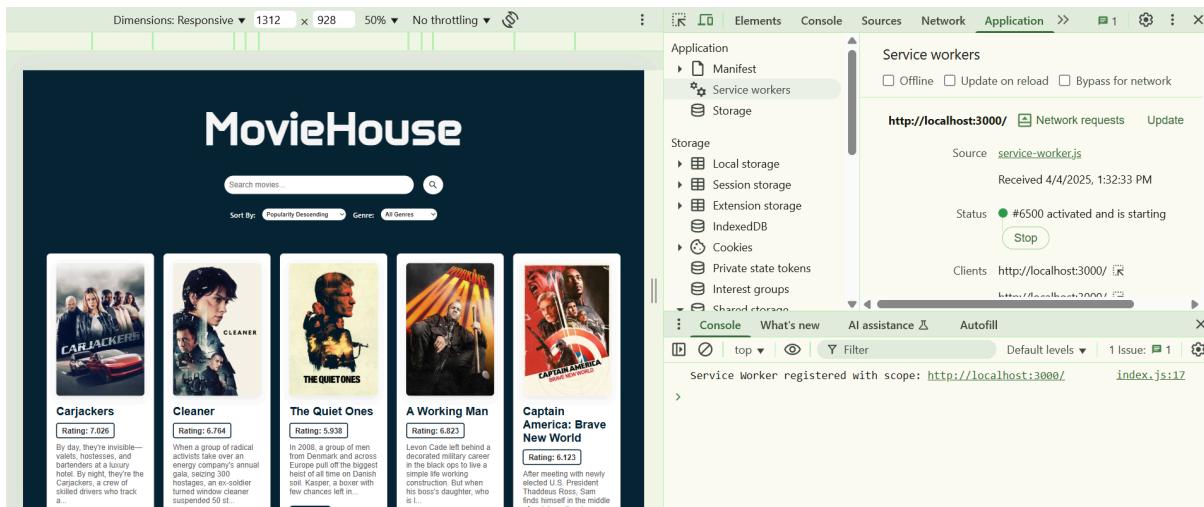
```

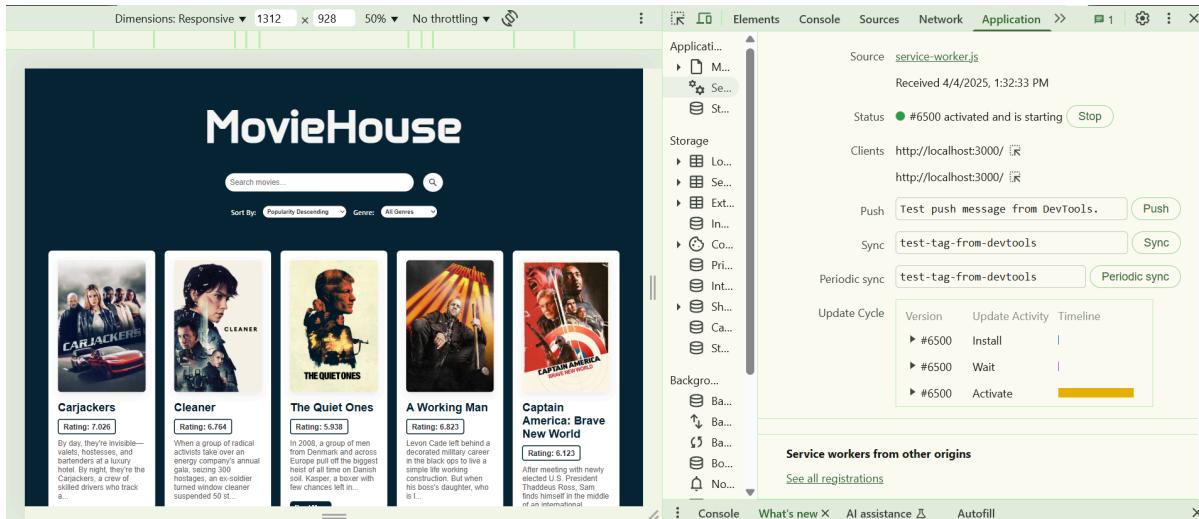
        if (cache !== CACHE_NAME) {
          console.log('[Service Worker] Deleting old cache:', cache);
          return caches.delete(cache);
        }
      })
    );
  );
  self.clients.claim();
});

} else {
  console.log('✖ Workbox failed to load');
}

```

## OUTPUT





The screenshot shows the Chrome DevTools Application tab open for the URL `http://localhost:3000`. The left sidebar lists various storage components: IndexedDB, Cookies, Private state ..., Interest gro..., Shared storag..., Cache storage, Background services, and Reporting API. The Cache storage section is expanded, showing sub-options like pwa-cach... and static-reso.... The main panel displays information about the default cache bucket: Bucket name is default, Is persistent is No, Durability is relaxed, Quota is 0 B, and Expiration is None. Below this is a table listing cache entries:

#	Name	Res...	Cont...	Con...	Tim...	Vary...
0	/static/js/bundle.js	basic	appl...	0	4/4/...	Acce...
1	/css2?family=Anta&family=...	opa...	text/...	0	4/4/...	Sec...

At the bottom of the panel, it says "Total entries: 2".

## 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.	01
Name	ADITYA SAMPATH KUMAR
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

## EXPERIMENT 9

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

### Theory:

#### Service Worker

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

Things to note about Service Worker:

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

#### Fetch Event

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

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

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

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

### Sync Event

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

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

### Push Event

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

We can check in the following example.

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

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

CODE-

Service-worker.js

```
// Import Workbox
```

```
importScripts('https://storage.googleapis.com/workbox-cdn/releases/6.4.1/workbox-sw.js');
```

```
if (workbox) {
```

```
    console.log('Workbox Loaded Successfully');
```

```
// Precache assets
workbox.precaching.precacheAndRoute(self.__WB_MANIFEST || []);

// Cache Images (Cache-First Strategy)
workbox.routing.registerRoute(
  ({ request }) => request.destination === 'image',
  new workbox.strategies.CacheFirst({
    cacheName: 'images-cache',
    plugins: [
      new workbox.expiration.ExpirationPlugin({
        maxEntries: 60,
        maxAgeSeconds: 30 * 24 * 60 * 60, // 30 Days
      }),
    ],
  })
);

// Cache CSS & JS (Stale-While-Revalidate Strategy)
workbox.routing.registerRoute(
  ({ request }) => request.destination === 'script' || request.destination === 'style',
  new workbox.strategies.StaleWhileRevalidate({
    cacheName: 'static-resources',
  })
);
```

```
);
```

```
// Cache API Responses (Network-First Strategy)
workbox.routing.registerRoute(
  ({ url }) => url.origin.includes('api.themoviedb.org'),
  new workbox.strategies.NetworkFirst({
    cacheName: 'api-cache',
    plugins: [
      new workbox.expiration.ExpirationPlugin({
        maxEntries: 50,
        maxAgeSeconds: 5 * 60, // 5 minutes
      }),
    ],
  })
);
```

```
}
```

```
// Fetch Event Logging (Works outside Workbox)
self.addEventListener('fetch', (event) => {
  console.log(`Fetch event detected: ${event.request.url}`);
  if (event.request.url.includes('api.themoviedb.org')) {
    console.log(`Intercepting API Request: ${event.request.url}`);
  }
});
```

```
event.respondWith(  
  caches.match(event.request).then((cachedResponse) => {  
    return cachedResponse || fetch(event.request).then((response) => {  
      console.log(` API Response Fetched: ${event.request.url}`);  
      return response;  
    }).catch((err) => {  
      console.error(` API Fetch Failed: ${event.request.url}`, err);  
      return new Response('API fetch failed', { status: 500 });  
    });  
  })  
);  
}  
});  
  
// Background Sync Event (Syncing Watchlist)  
self.addEventListener('sync', (event) => {  
  if (event.tag === 'sync-watchlist') {  
    console.log(' Sync event triggered: sync-watchlist');  
    event.waitUntil(  
      syncWatchlist().then(() => {  
        console.log(' Sync successful');  
      }).catch((err) => {  
        console.error(' Sync failed:', err);  
      });  
    );  
  }  
});
```

```
    })
  );
}

});

// Example Function to Sync Watchlist
async function syncWatchlist() {
  console.log(' Syncing watchlist data...');
  return fetch('/sync-watchlist', { method: 'POST' })
    .then(() => console.log(' Sync request sent successfully!'))
    .catch(() => console.log(' Sync request failed, retrying later.'));
}

// Push Notification Event
self.addEventListener('push', (event) => {
  console.log('Push notification received');

  const notificationData = event.data ? event.data.text() ;
  console.log(`✉ Push payload: ${notificationData}`);

  const options = {
    body: notificationData,
    icon: '/logo192.png',
    badge: '/logo192.png',
```

```
};

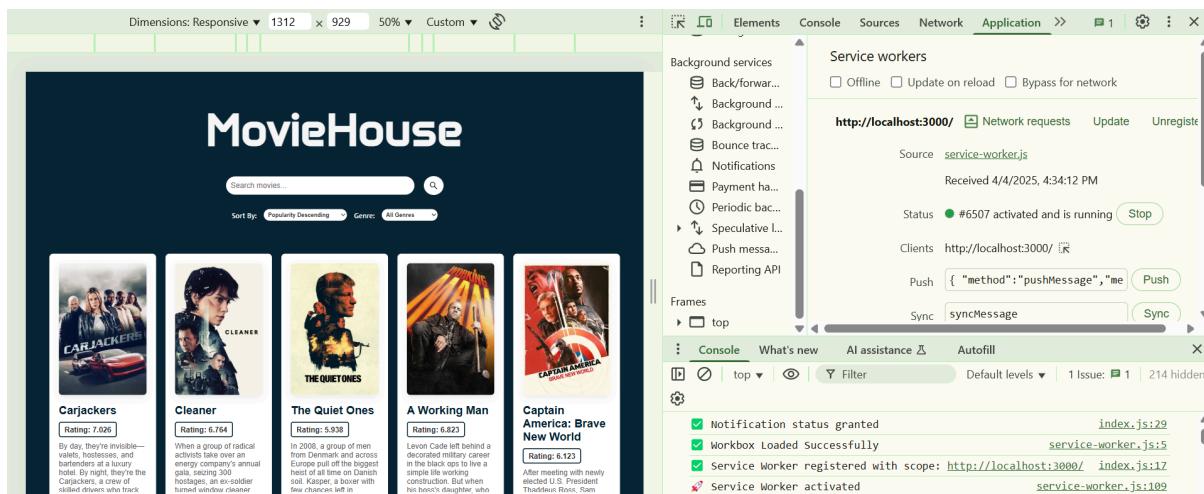
event.waitUntil(
  self.registration.showNotification(' Movie House', options)
  .then(() => console.log(' Push notification sent successfully'))
  .catch((err) => console.error(' Push notification failed:', err))
);

// Activate event - Cleanup old caches
self.addEventListener('activate', (event) => {
  console.log(' Service Worker activated');
  event.waitUntil(
    caches.keys().then((cacheNames) =>
      Promise.all(
        cacheNames.map((cache) => {
          if (!['images-cache', 'static-resources', 'api-cache'].includes(cache)) {
            console.log(`🗑 Deleting old cache: ${cache}`);
            return caches.delete(cache);
          }
        })
      )
    );
});
```

```
self.clients.claim();
});
```

## OUTPUT

Service worker activated



## Fetch Event

Dimensions: Responsive ▾ 1312 x 929 50% Custom ▾

MovieHouse

Search: karate Kid

Sort By: Popularity Descending | Genres: All Genres

The Karate Kid Part II (Rating: 6.2/10)  
Summoned by his dying father, Miyagi returns to his homeland of Okinawa, where he begins a year exile. There he must confront Yukie, the love

The Karate Kid Part III (Rating: 5.8/10)  
Despondent over the closing of his karate school, Cobra Kai teacher John Kreese joins a gangster's crew and martial artist to get

Karate Kid: Legends (Rating: 0)  
After a brief respite, rising prodigy Li Fong is uprooted from his home in Beijing and moves to New York City with his mother. When a

The Karate Kid (Rating: 7.1/10)  
Daniel LaRusso moves to Los Angeles to live with his mother, Lucille, and soon strikes up a relationship with her son, who becomes the target of bull...

The Next Karate Kid (Rating: 5.5/10)  
Mr. Miyagi decides to take his life as a troubled teenager under his wing after he overhears his mother scheme for her parents' demise and struggles to...

Service workers

Offline  Update on reload  Bypass for network

http://localhost:3000/ Network requests Update Unregister

Source service-worker.js

Received 4/4/2025, 4:36:39 PM

Status ● #6508 activated and is running Stop

Push { "method": "pushMessage", "me" } Push

Sync syncMessage Sync

Periodic sync test-tag-from-devtool: Periodic sync

Console What's new AI assistance Autofill

Fetch event detected: http://localhost:3000/manifest.json service-worker.js:50

Fetch successful: http://localhost:3000/manifest.json service-worker.js:55

Fetch event detected: https://fonts.gstatic.com/s/anta/v1/gybzhcQ3KsTyVs7PQ.woff2 service-worker.js:50

## Sync Event

Dimensions: Responsive ▾ 1312 x 929 50% Offline ▾

Tablet - 768px

MovieHouse

Search: my name is khan

Sort By: Popularity Descending | Genres: All Genres

My Name Is Khan (Rating: 8.0/10)  
Proven Khan, a Muslim from the Dharavi section of Mumbai, has Asperger's syndrome. He marries a Hindu single mother, Mandira, and San

My Name Is Khan (Rating: 9)  
Written about the daughter of a wealthy man, falls in love with Sagar, a man who is poor and who hates how the rich treat the poor, she poses

Application

Manifest Service worker... Storage

Storage Local storage Session stor... Extension stor... IndexedDB Cookies Private state ... Interest gro... Shared stor...

http://localhost:3000/ Network requests Update Unregister

Source service-worker.js

Received 4/4/2025, 4:47:38 PM

Status ● #6510 activated and is running Stop

Push { "method": "pushMessage", "me" } Push

Sync syncMessage Sync

Periodic sync test-tag-from-devtool: Periodic sync

Console What's new AI assistance Autofill

https://image.tmdb.org/t/p/w500/1d27rwKwCHpQdfFP0gW7MGinKyk.jpg'

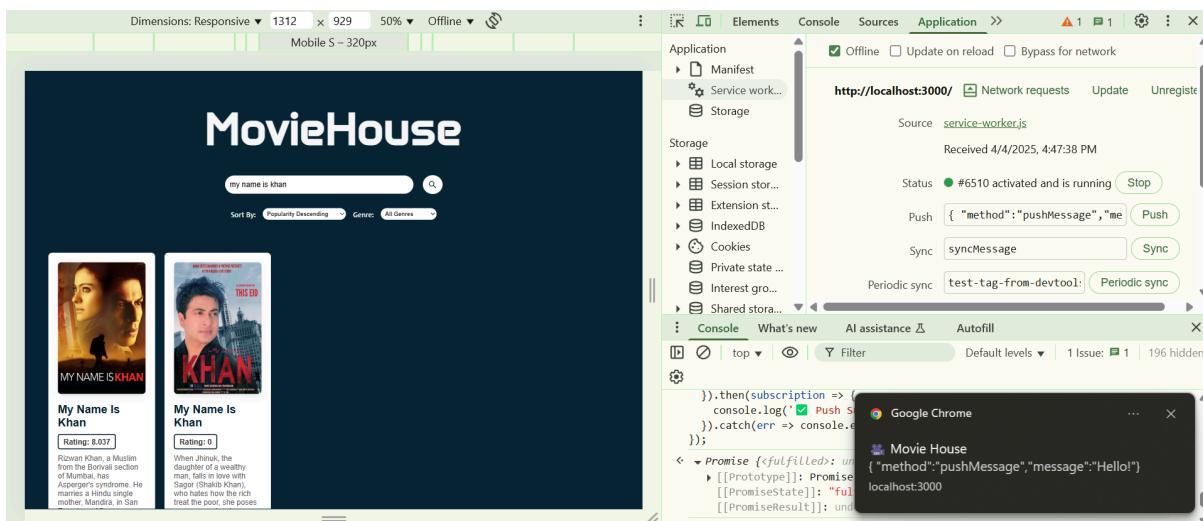
Sync event triggered: sync-watchlist service-worker.js:72

Syncing watchlist data... service-worker.js:85

Sync request sent successfully! service-worker.js:87

Sync successful service-worker.js:75

## Push Event



# MAD & PWA Lab

## Journal

Experiment No.	10
Experiment Title.	To study and implement deployment of Ecommerce PWA to GitHub Pages.
Roll No.	01
Name	ADITYA SAMPATH KUMAR
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**

**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**

##### **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.

#### 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

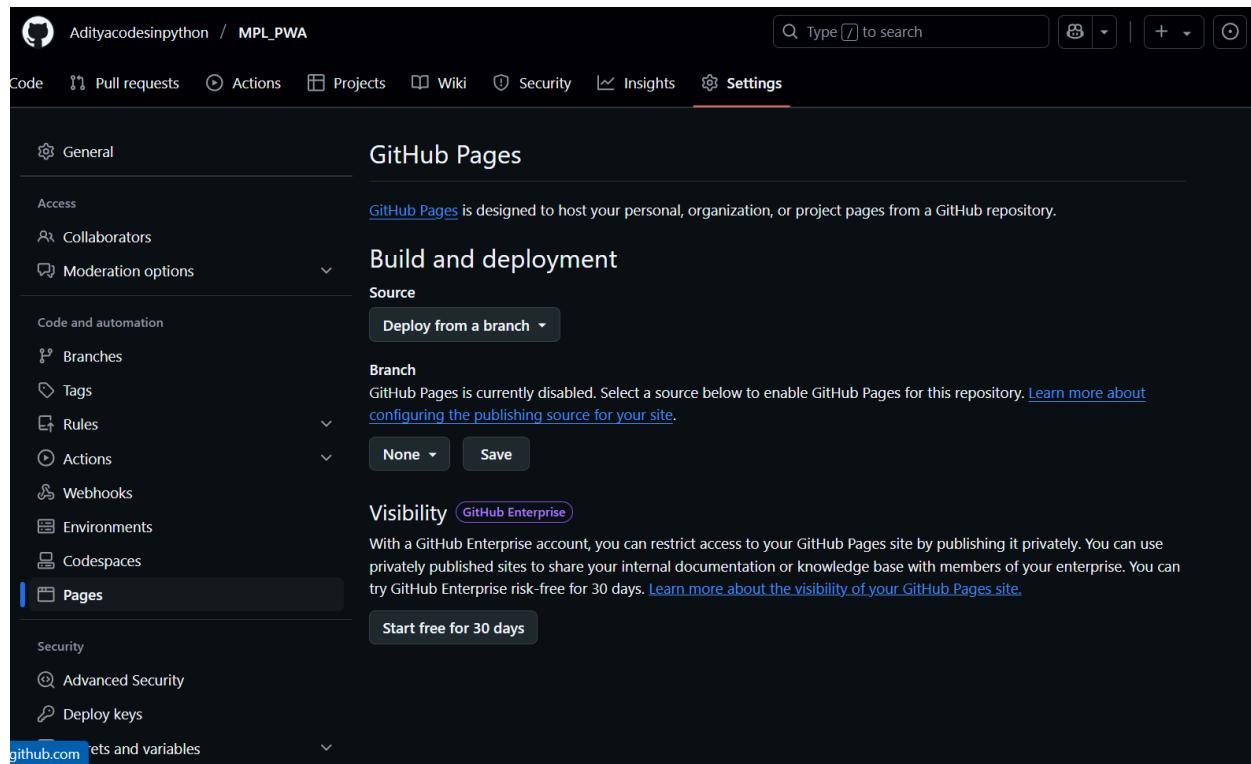
- Hosted by Google, ensuring reliability and security.
- Comes with authentication, messaging, and real-time database services.
- Free HTTPS support for all custom domains.

## 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: [Tourly - Travel agency](#)

## Github Screenshot:



Adityacodesinpython / MPL\_PWA

Type [I] to search

Pull requests Actions Projects Wiki Security Insights Settings

## github-pages deployments

Latest deployments

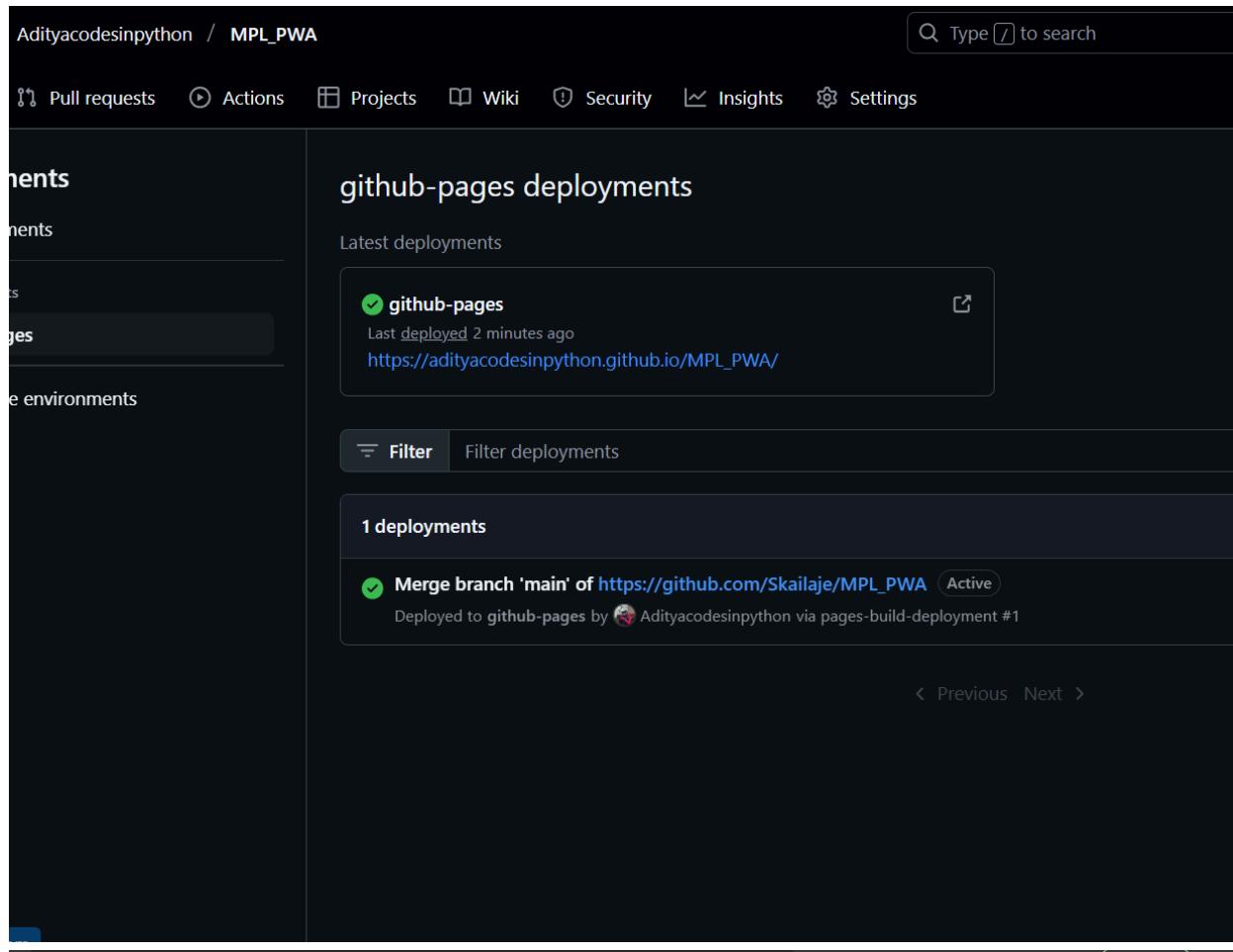
github-pages Last deployed 2 minutes ago [https://adityacodesinpython.github.io/MPL\\_PWA/](https://adityacodesinpython.github.io/MPL_PWA/)

Filter Filter deployments

1 deployments

Merge branch 'main' of https://github.com/Skailaje/MPL\_PWA Active  
Deployed to github-pages by Adityacodesinpython via pages-build-deployment #1

< Previous Next >



https://adityacodesinpython.github.io/MPL\_PWA/

For Further Inquiries : +01 (123) 4567 90

**TOURLY**

HOME ABOUT US DESTINATION PACKAGES GALLERY CONTACT US BOOK NOW

# EMBARK ON A JOURNEY TO DISCOVER THE WORLD

Explore breathtaking destinations, immerse yourself in diverse cultures, and create unforgettable memories. Your adventure begins here!

LEARN MORE BOOK NOW



## MAD & PWA Lab

### Journal

Experiment No.	11
Experiment Title.	To use google Lighthouse PWA Analysis Tool to test the PWA functioning.
Roll No.	01
Name	ADITYA SAMPATH KUMAR
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO6: Develop and Analyze PWA Features and deploy it over app hosting solution
Grade:	

# EXPERIMENT NO: - 11

**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:

- Service Worker implementation – Ensuring offline support and background synchronization.
- App Manifest compliance – Providing metadata for better mobile integration.
- Viewport configuration – Optimizing mobile responsiveness.
- Performance in script-disabled environments – Ensuring the page functions even when JavaScript is disabled.

A high PWA score indicates that the application meets essential PWA criteria and provides an app-like user experience.

### 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:

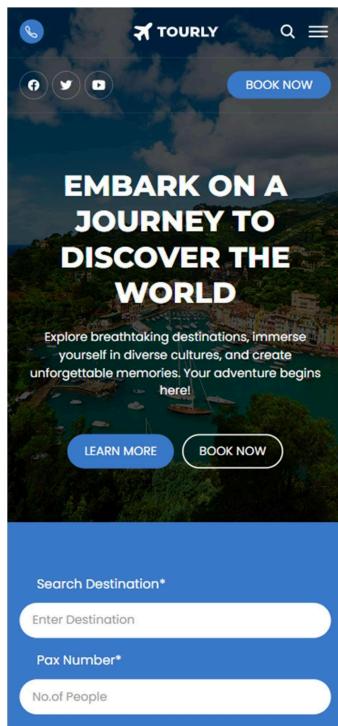
- Use of HTTPS – Ensuring secure data transmission.
- Avoiding deprecated code – Preventing the use of outdated elements, directives, and libraries.
- Secure password inputs – Disabling paste-into fields to mitigate credential theft risks.
- User security alerts – Prompting users about geo-location access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

### Manifest.json

```
{
  "id": "/",
  "name": "TOURLY - Travel Ecommerce",
  "short_name": "TLY",
  "description": "A travel ecommerce platform for booking tours and experiences.", "start_url": "/index.html",
  "scope": "/",
  "display": "standalone",
  "background_color": "#FFFFFF",
  "theme_color": "#000000",
  "orientation": "portrait",
  "lang": "en",
  "categories": ["travel", "shopping"],
  "ecommerce": [],
  "icons": [
    {
      "src": "assets/images/logo_tourly_192.png",
      "sizes": "192x192",
      "type": "image/png", "purpose": "any maskable"
    },
    {
      "src": "assets/images/logo_tourly_512.webp",
      "sizes": "512x512",
      "type": "image/webp",
      "purpose": "any maskable"
    },
    {
      "src": "assets/images/logo_tourly_192.webp",
      "sizes": "192x192",
      "type": "image/webp",
      "purpose": "any maskable"
    }
  ]
}
```

],  
"screenshots": [  
{

```
"src": "assets/screenshots/homepage.png",
  "sizes": "1080x1920",
  "type": "image/png"
},
{
  "src": "assets/screenshots/booking.png",
  "sizes": "1080x1920",
  "type": "image/png"
}
],
"shortcuts": [
  {
    "name": "Book a Tour",
    "short_name": "Book",
    "description": "Go directly to tour booking",
    "url": "/book.html",
    "icons": [
      {
        "src": "images/shortcut-icon.png",
        "sizes": "96x96",
        "type": "image/png"
      }
    ]
  },
  {
    "name": "View Popular Tours",
    "short_name": "Popular",
    "description": "Explore popular tours",
    "url": "/popular.html",
    "icons": [
      {
        "src": "images/popular-shortcut-
icon.png",
        "sizes": "96x96",
        "type": "image/png"
      }
    ]
  }
],
{
  "iarc_rating_id": "e10c6b5e-3b3c-4d3a-9f3b-1e5f3e2b5f3c",
  "dir": "ltr",
  "related_applications": [
    {
      "platform": "play",
      "url": "https://play.google.com/store/apps/details?id=com.tourly.app",
      "id": "com.tourly.app"
    },
    {
      "platform": "itunes",
      "url": "https://apps.apple.com/app/id1234567890"
    }
  ],
  "prefer_related_applications": false
}
```



ⓘ The Lighthouse tool provides links to content hosted on third-party websites. [Don't show again](#) [Show more](#) [X](#)

+ 10:42:45 pm - 127.0.0.1:8081 ▾ [🔗](#)

⚠ http://127.0.0.1:8081/ [⋮](#)

99

Performance

96

Accessibility

93

Best Practices

91

SEO

99

Performance

TOURLY

EMBARK ON A JOURNEY TO DISCOVER THE WORLD

Explore breathtaking destinations, immerse yourself in diverse cultures, and create unforgettable memories. Your adventure begins here!

LEARN MORE [BOOK NOW](#)

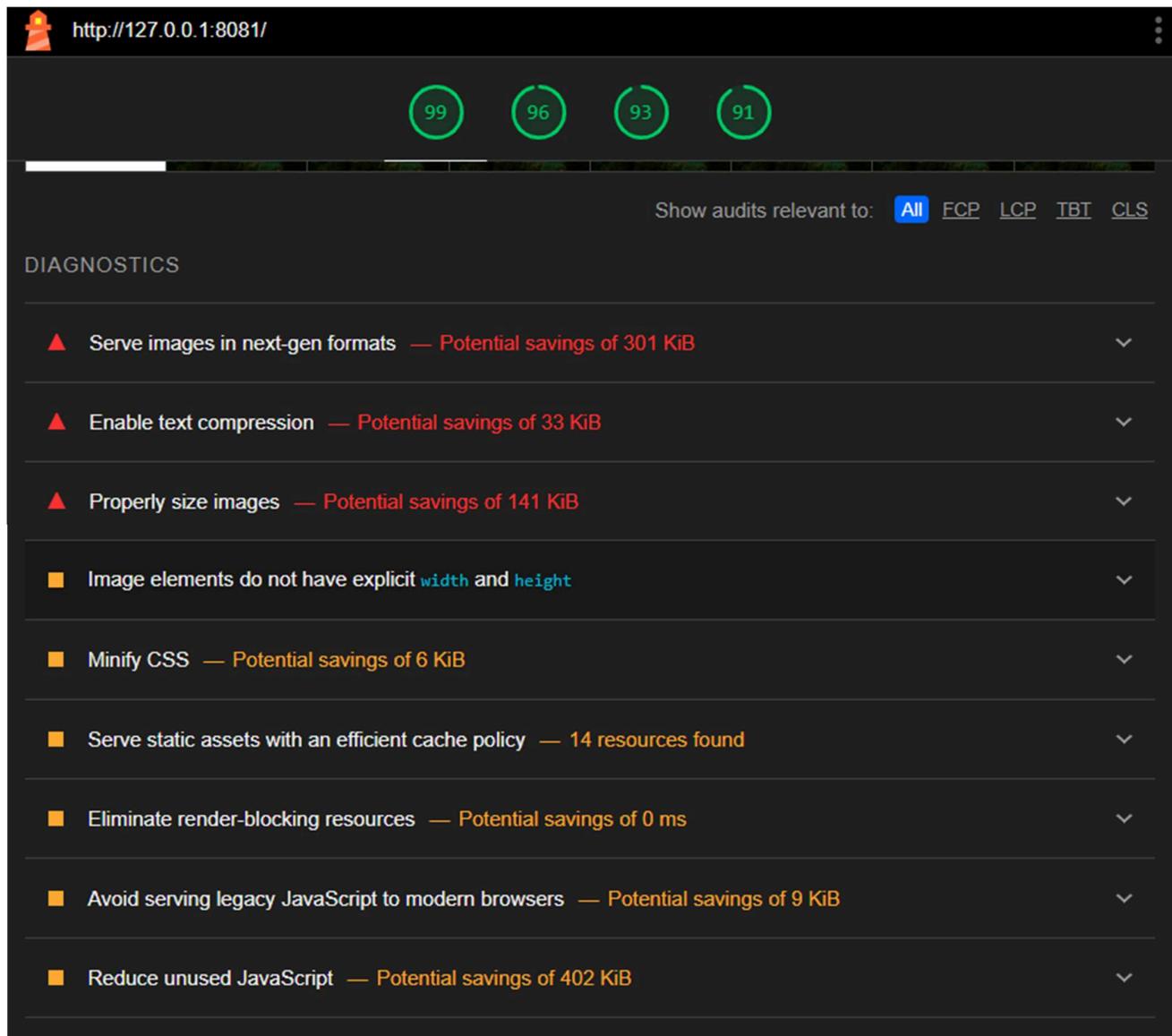
▲ 0–49

■ 50–89

● 90–100

METRICS

[Expand view](#)



The screenshot shows the Lighthouse audit results for the URL <http://127.0.0.1:8081/>. The overall score is 91. The report includes sections for performance, accessibility, and progressive web app features.

**Performance Score: 91**

Show audits relevant to: **All** FCP LCP TBT CLS

**DIAGNOSTICS**

- ▲ Serve images in next-gen formats — Potential savings of 301 KiB
- ▲ Enable text compression — Potential savings of 33 KiB
- ▲ Properly size images — Potential savings of 141 KiB
- Image elements do not have explicit `width` and `height`
- Minify CSS — Potential savings of 6 KiB
- Serve static assets with an efficient cache policy — 14 resources found
- Eliminate render-blocking resources — Potential savings of 0 ms
- Avoid serving legacy JavaScript to modern browsers — Potential savings of 9 KiB
- Reduce unused JavaScript — Potential savings of 402 KiB

The screenshot shows the Lighthouse audit results for the [TOURLY - Travel Ecommerce a...](#) site. The overall score is 91, indicated by a green circle. The audit includes sections for SEO, Performance, and Accessibility.

**Performance Score: 99**

Key metrics contributing to the high performance score:

- SI (First Input Delay): 0 ms
- FCP (First Contentful Paint): 0 ms
- LCP (Largest Contentful Paint): 0 ms
- TBT (Total Blocking Time): 0 ms
- CLS (Cumulative Layout Shift): 0 ms

**SEO Score: 91**

The SEO audit highlights the following areas:

- Mobile friendly: Yes
- Mobile optimization: Good
- Page speed: Very fast
- Content: Good
- Security: Good
- Mobile best practices: Good
- Server response time: Good
- Image optimization: Good
- Page size: Good
- JavaScript delivery: Good
- Font loading: Good
- Character encoding: Good
- SSL certificate: Good
- HTTP/2 support: Good
- Page size: Good
- JavaScript delivery: Good
- Font loading: Good
- Character encoding: Good
- SSL certificate: Good
- HTTP/2 support: Good

**Accessibility Score: 91**

The accessibility audit found no critical or major issues.

**Overall Summary:** The site has a high overall score of 91, indicating excellent performance across all audited categories. The performance score is particularly strong, with all metrics achieving 0 ms.

# MAD & PWA Lab

## Journal

Experiment No.	Assignment-1
Assignment 1 Questions	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
Roll No.	01
Name	ADITYA SAMPATH KUMAR
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	<p>LO1: Understand cross platform mobile application development using Flutter framework</p> <p>LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation</p> <p>LO3: Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS</p>
Grade:	

# 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.	01
Name	ADITYA SAMPATH KUMAR
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:	