

# 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 HITESH TANWANI 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 Teacher

**Dr. Ravita Mishra**

Signature:

Head of Department

**Dr. Mrs. Shalu Chopra**

Signature:

Principal

# Index

Sr. No	Experiment Title	LO	DOP	DOS	Grade
1.	To install and configure the Flutter Environment	LO1	21/01		
2.	To design Flutter UI by including common widgets.	LO2	28/1		
3.	To include icons, images, fonts in Flutter app	LO2	4/2		
4.	To create an interactive Form using form widget	LO2	11/2		
5.	To apply navigation, routing and gestures in Flutter App	LO2	18/2		
6.	To Connect Flutter UI with fireBase database	LO3	4/3		
7.	To write meta data of your Ecommerce PWA in a Web app manifest file to enable "add to homescreen feature".	LO4	18/3		
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	25/3		
9.	To implement Service worker events like fetch, sync and push for E-commerce PWA	LO5	1/4		
10.	To study and implement deployment of Ecommerce PWA to GitHub Pages.	LO5	1/4		
11.	To use google Lighthouse PWA Analysis Tool to test the PWA functioning.	LO6	8/4		
12.	Assignment-1	LO1,LO2 ,LO3	6/3		
13.	Assignment-2	LO4,LO5 ,LO6	6/3		



**Vivekanand Education Society's  
Institute of Technology**

(An Autonomous Institute Affiliated to University of Mumbai)

**Department of Information Technology**

A.Y. 24-25

**Mobile App Development and Progressive Web App**

Experiment No.	01
Experiment Title.	To install and configure the Flutter Environment.
Roll No.	57
Name	Hitesh Tanwani
Class	D15 B
Subject	MAD & PWA
Grade:	

HITESH  
D15 B

TANWANI  
57.

## MPL Experiment-1

DATE:

Aim - To study and perform the installation of Android Studio

### Theory

Android Studio is the official integrated development env (IDE) for google's Android OS, build on JetBrain's IntelliJ IDEA software & designed specifically for Android development.

Steps The installation of AS involves IPR along with components

- Android SDK
- SDK Tools
- Gradle Build

Conclusion - These components allow & help to create & test android app across different android version.

We also installed android studio and did setup for same

Name: Hitesh Tanwani  
Class: D15B  
Roll No: 57  
Sub: MPL

# Experiment 1

Aim: Installation and Configuration of Flutter Environment.

## Theory:

Flutter is an open-source framework for building natively compiled applications for mobile, web, and desktop from a single codebase. It is developed and maintained by Google, and it is used to create high-performance, beautiful applications for platforms such as Android, iOS, macOS, Windows, and the web.



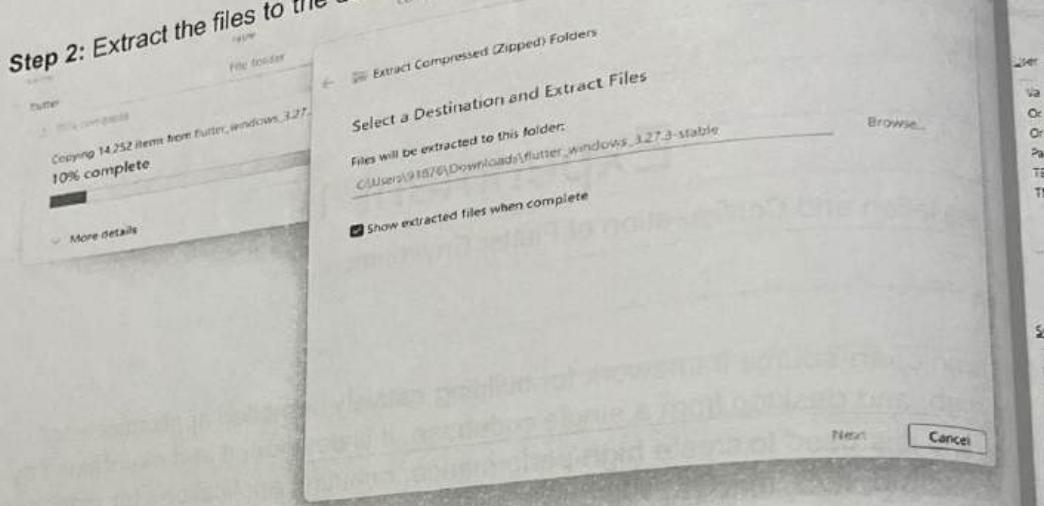
Flutter is particularly known for its fast development cycle, expressive UI, and native performance. It allows developers to write once and run on multiple platforms, drastically reducing development time and effort.

## Steps to download:

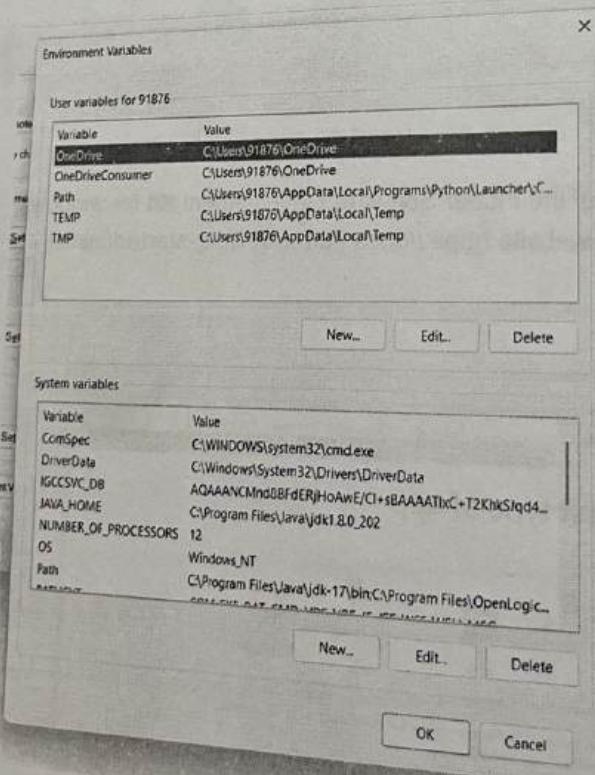
**Step 1:** Download the installation bundle of the Flutter Software Development Kit for windows.  
To download Flutter SDK, Go to its official website <https://docs.flutter.dev/get-started/install>,

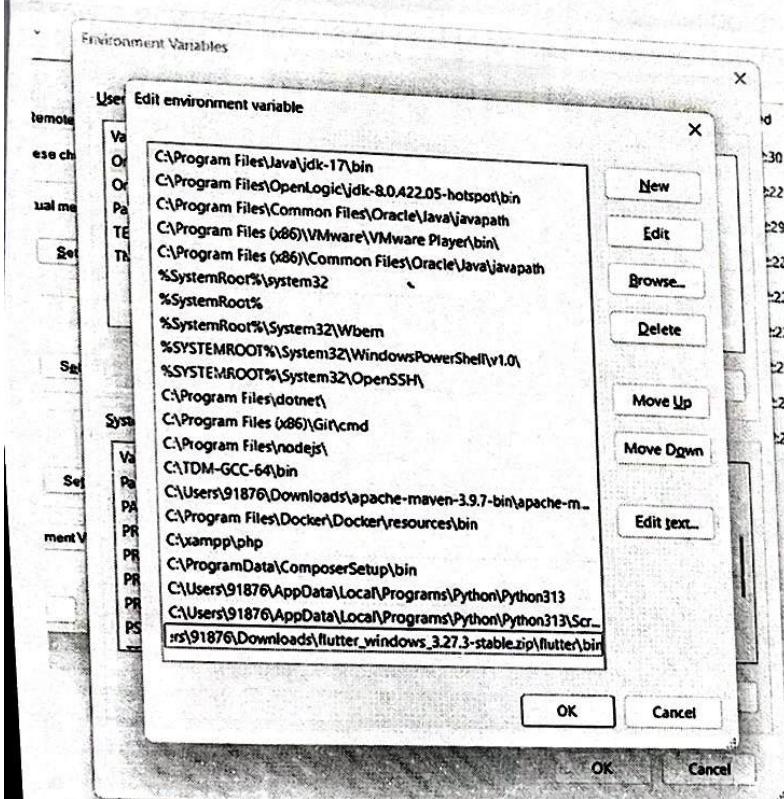
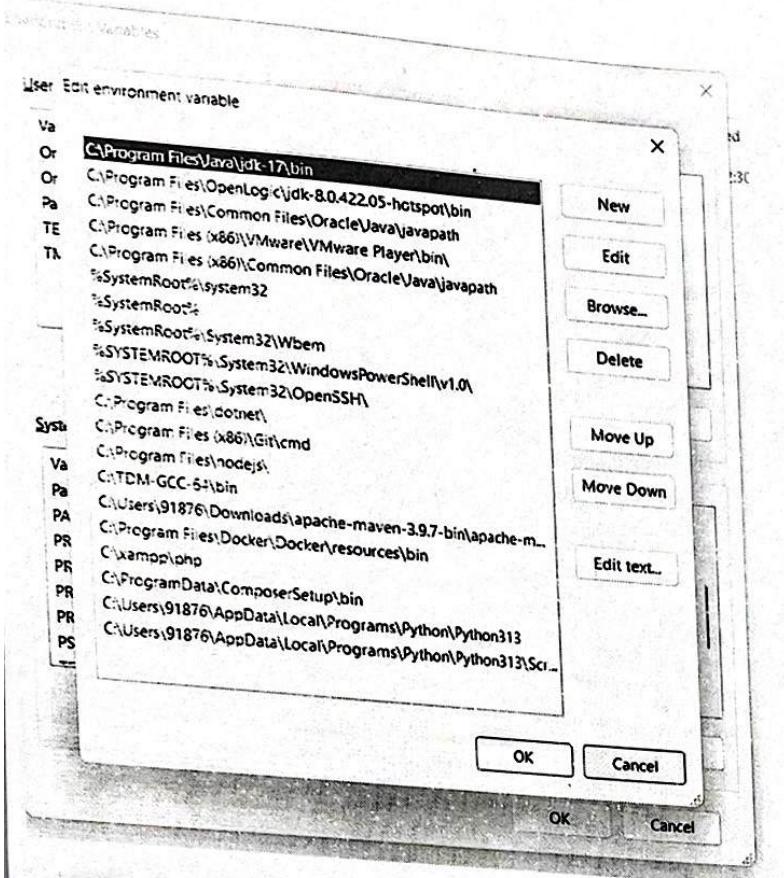
The screenshot shows a web browser displaying the Flutter Documentation website at [docs.flutter.dev/get-started/install/windows](https://docs.flutter.dev/get-started/install/windows). The page title is "Flutter Docs". On the left, there is a sidebar with a navigation menu. The main content area has a heading "Choose your first type of app" with sub-sections "Get started > Install > Windows". It features three large icons: "Android recommended" (with a smartphone icon), "Web" (with a laptop icon), and "Desktop" (with a monitor icon). Below these icons, there is a note: "Your choice informs which parts of Flutter tooling you configure to run your first Flutter app. You can set up additional platforms later. If you don't have a preference, choose Android." There is also a section titled "Developing in China" with a note: "If you want to use Flutter in China, check out Using Flutter in China. If you're not developing in China, ignore this notice and follow the other instructions on this page." At the bottom, there is some Chinese text: "如果您正在中国的网络环境下使用 Flutter，请参考 在中国网络环境下使用 Flutter 文档。"

**Step 2: Extract the files to the desired destination folder**



**Step 3:** To run the Flutter command in regular windows console, you need to update the system path to include the flutter bin directory. The following steps are required to do this:





**Step 4:** Now, run the \$ flutter command in command prompt.

```
Command Prompt - flutter x + v
Microsoft Windows [Version 10.0.26100.2894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91876>flutter
Command exited with code 128: git fetch --tags
Standard error: error: RPC failed; curl 92 HTTP/2 stream 7 was not closed cleanly: CANCEL (err 3)
error: 4336 bytes of body are still expected
fetch-pack: unexpected disconnect while reading sideband packet
fatal: early EOF
fatal: fetch-pack: invalid index-pack output

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).
```

**Step 5:** When you run the above command, it will analyze the system and show its report, as shown in the below image. Here, you will find the details of all missing tools, which required to run Flutter as well as the development tools that are available but not connected with the device.

```
Command Prompt - flutter - I x + v
Run "flutter help <command>" for more information about a command.
Run "flutter help -v" for verbose help output, including less commonly used options.

C:\Users\91876>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 3.27.3, on Microsoft Windows [Version 10.0.26100.2894], locale en-US)
[!] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices
  X Unable to locate Android SDK.
    Install Android Studio from: https://developer.android.com/studio/index.html
    On first launch it will assist you in installing the Android SDK components.
    (or visit https://flutter.dev/to/windows-android-setup for detailed instructions).
    If the Android SDK has been installed to a custom location, please use
      'flutter config --android-sdk' to update to that location.

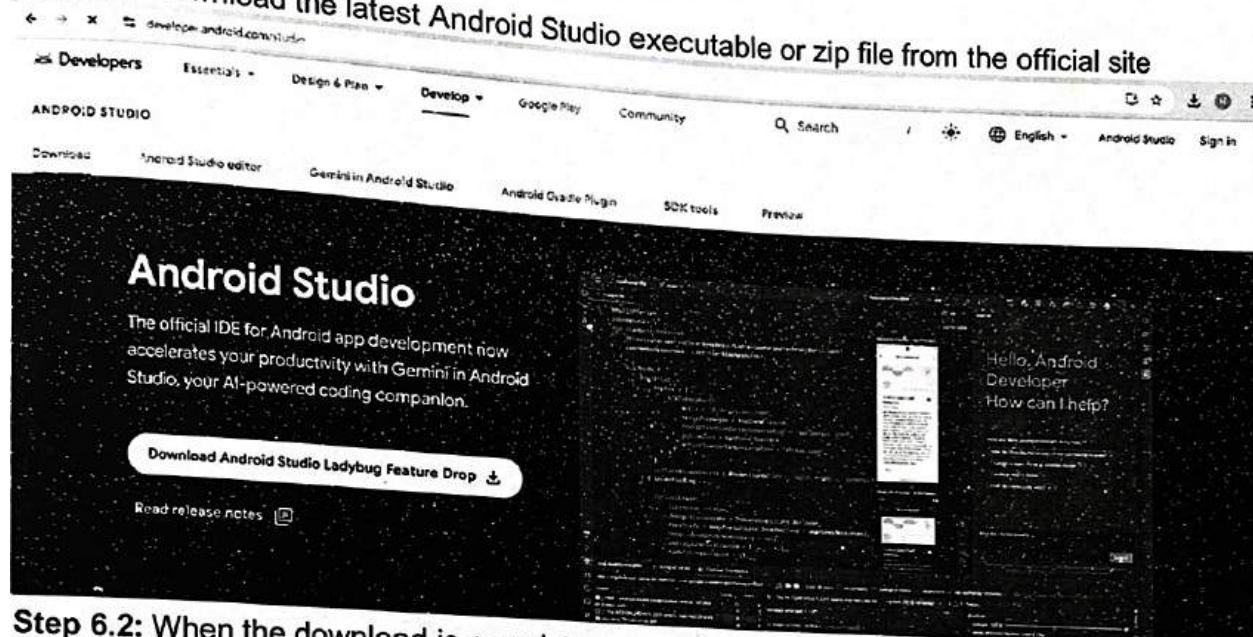
[!] Chrome - develop for the web
[!] Visual Studio - develop Windows apps
  X Visual Studio not installed; this is necessary to develop Windows apps.
    Download at https://visualstudio.microsoft.com/downloads/.
    Please install the "Desktop development with C++" workload, including all of its default components
[!] Android Studio (not installed)
[!] IntelliJ IDEA Ultimate Edition (version 2024.1)
[!] VS Code (version 1.96.4)
[!] Connected device (3 available)
[!] Network resources

! Doctor found issues in 3 categories.

C:\Users\91876>
```

**Step 6:** Install the Android SDK. If the flutter doctor command does not find the Android SDK tool in your system, then you need first to install the Android Studio IDE. To install Android Studio IDE, do the following steps

**Step 6.1:** Download the latest Android Studio executable or zip file from the official site

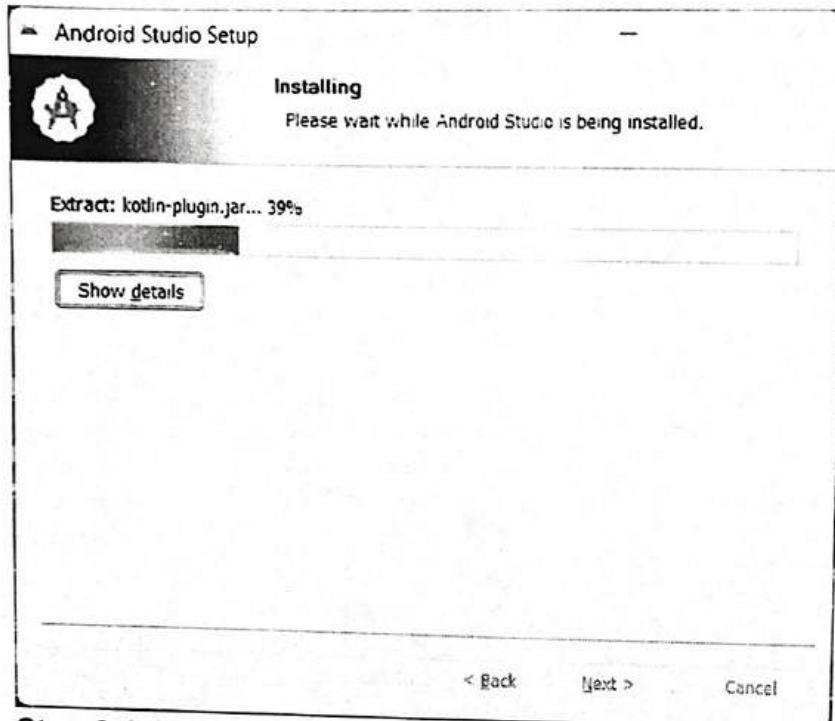


**Step 6.2:** When the download is complete, open the .exe file and run it. You will get the following dialog box.

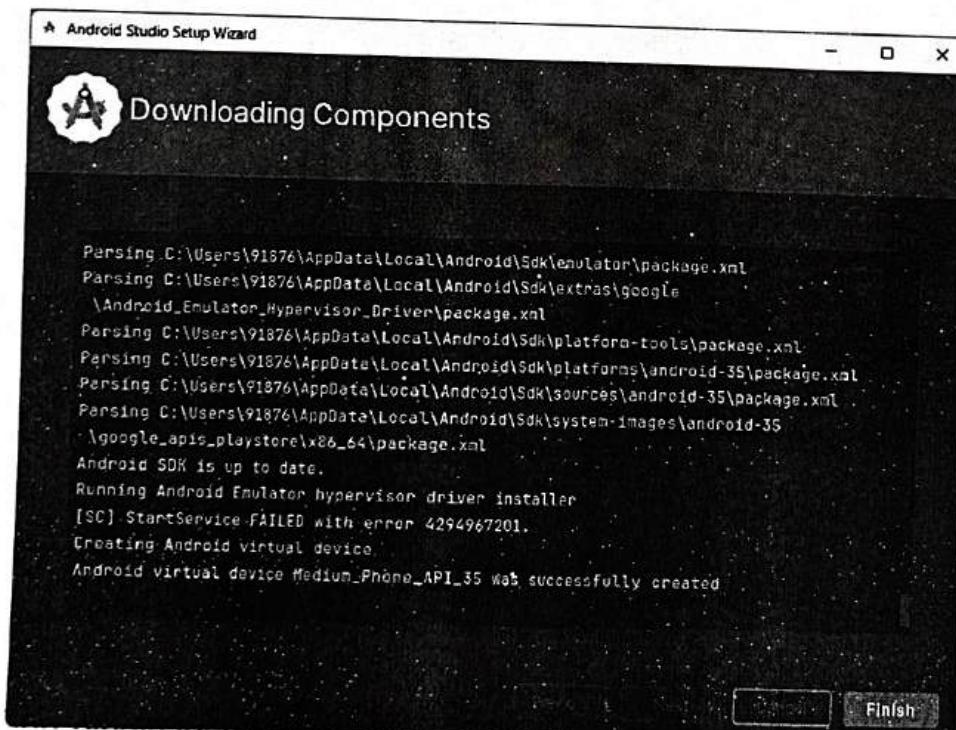


**Step 6.3:** Follow the steps of the installation wizard. Once the installation wizard completes,

you will get the following screen.



**Step 6.4:** In the above screen, click Next-> Finish. Once the Finish button is clicked, you need to choose the 'Don't import Settings option' and click OK. It will start the Android Studio.

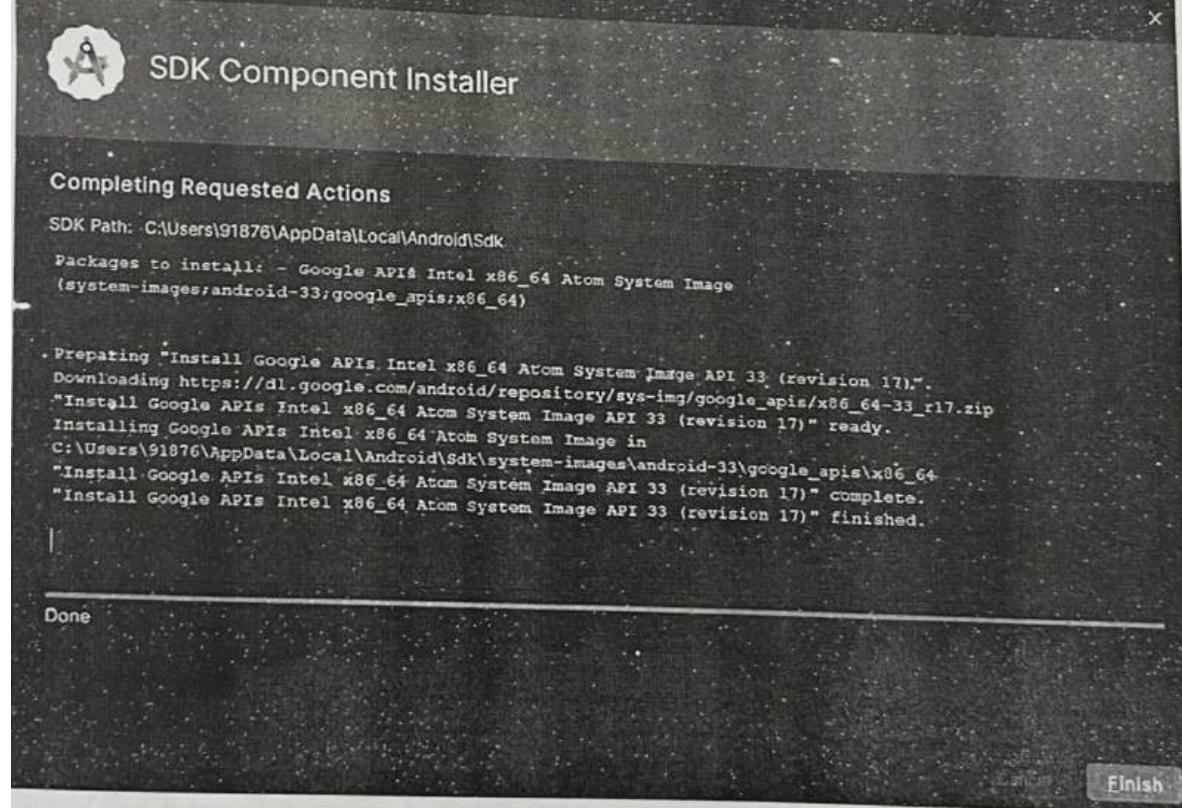


Step 6.5: run the \$ flutter doctor command and Run flutter doctor --android-licenses

```
Command Prompt - Flutter < + ~
C:\Users\91876>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 3.27.3, on Microsoft Windows [Version 10.0.26100.2894], 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)
[X] cmdline-tools component is missing
  Run 'path/to/sdkmanager --install "cmdline-tools;latest"'
  See https://developer.android.com/studio/command-line for more details.
[X] Android license status unknown.
  Run 'flutter doctor --android-licenses' to accept the SDK licenses.
  See https://flutter.dev/to/windows-android-setup for more details.
[!] 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 (version 2024.2)
[!] IntelliJ IDEA Ultimate Edition (version 2024.1)
[!] VS Code (version 1.96.4)
[!] Connected device (3 available)
[!] Network resources
  X A network error occurred while checking "https://github.com/": Failed host lookup: 'github.com'
Doctor found issues in 3 categories

C:\Users\91876>flutter doctor --android-licenses
AndroidManifest.xml not found. Update to the latest Android SDK and ensure that the cmdline-tools are installed.

C:\Users\91876>
● SDK Quickfix Installation



SDK Component Installer



Completing Requested Actions



SDK Path: C:\Users\91876\AppData\Local\Android\Sdk



Packages to install: - Google APIs Intel x86_64 Atom System Image  
(system-images;android-33;google_apis;x86_64)



Preparing "Install Google APIs Intel x86_64 Atom System Image API 33 (revision 17)".  
Downloading https://dl.google.com/android/repository/sys-img/google_apis/x86_64-33_r17.zip  
"Install Google APIs Intel x86_64 Atom System Image API 33 (revision 17)" ready.  
Installing Google APIs Intel x86_64 Atom System Image in  
C:\Users\91876\AppData\Local\Android\Sdk\system-images\android-33\google_apis\x86_64  
"Install Google APIs Intel x86_64 Atom System Image API 33 (revision 17)" complete.  
"Install Google APIs Intel x86_64 Atom System Image API 33 (revision 17)" finished.



Done



Finish


```

After solving the error:

```
C:\Users\91876>flutter doctor --android-licenses
Warning: Additionally, the fallback loader failed to parse the XML.
Warning: Errors during XML parse: ] 49% Fetch remote repository.
Warning: Additionally, the fallback loader failed to parse the XML.xml.
[ 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 94041, 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.

C:\Users\91876> flutter doctor --android-licenses
Warning: Errors during XML parse: ] 49% Fetch remote repository.
Warning: Errors during XML parse: ] 100% Computing updates...
10. Environmental approval, without first obtaining such license or approval. Recipient also agrees to implement measures to ensure that foreign national employees are authorized to receive any information controlled by U.S. export control laws. An export is "deemed" to take place when information is released to a foreign national wherever located.
10.7 Special Terms for Pre-Release Materials. If so indicated in the description of the Evaluation Software, the Evaluation Software may contain Pre-Release Materials. Recipient hereby understands, acknowledges and agrees that: (i) Pre-Release Materials may not be fully tested and may contain bugs or errors; (ii) Pre-Release materials are not suitable for commercial release in their current state; (iii) regulatory approvals for Pre-Release Materials (such as UL or FCC) have not been obtained, and Pre-Release Materials may therefore not be certified for use in certain countries or environments or may not be suitable for certain applications and (iv) MIPS can provide no assurance that it will ever produce or make generally available a production version of the Pre-Release Materials. MIPS is not under any obligation to develop and/or release or offer for sale or license a final product based upon the Pre-Release Materials and may unilaterally elect to abandon the Pre-Release Materials or any such development platform at any time and without any obligation or liability whatsoever to Recipient or any other person.
ANY PRE-RELEASE MATERIALS ARE NON-QUALIFIED AND, AS SUCH, ARE PROVIDED "AS IS" AND "AS AVAILABLE", POSSIBLY WITH FAULTS, AND WITHOUT REPRESENTATION OR WARRANTY OF ANY KIND.
10.8 Open Source Software. In the event Open Source software is included with Evaluation Software, such Open Source software is licensed pursuant to the applicable Open Source software license agreement identified in the Open Source software comments in the applicable source code file(s) and/or file header as indicated in the Evaluation Software. Additional detail may be available (where applicable) in the accompanying on-line documentation. With respect to the Open Source software, nothing in this Agreement limits any rights under, or grants rights that supersede, the terms of any applicable Open Source software license agreement.

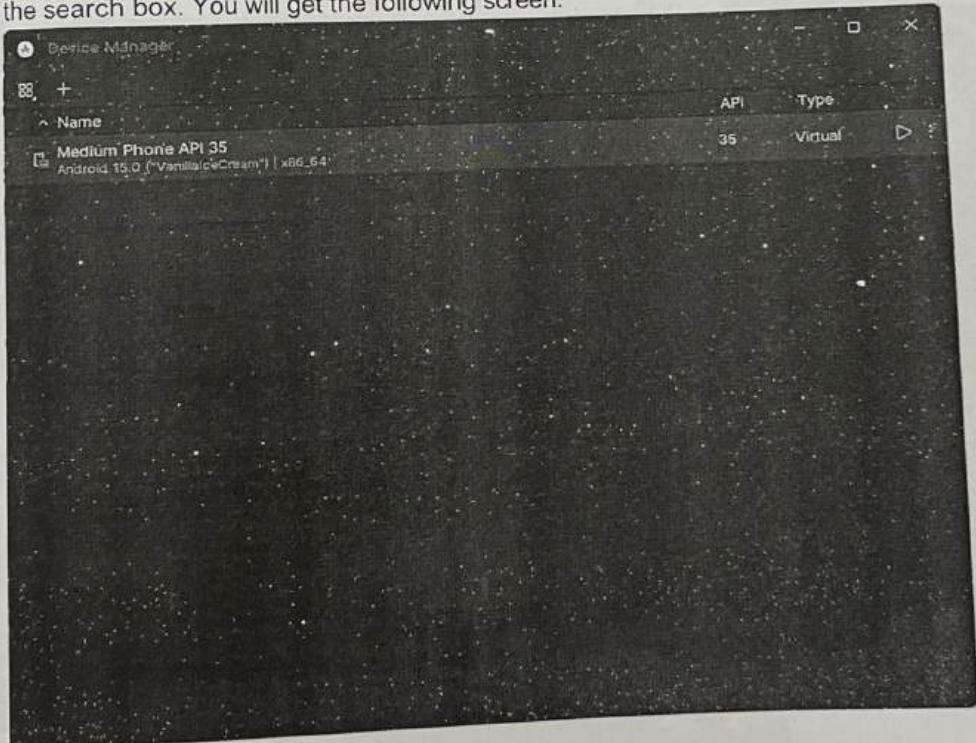
Accept? (y/N): y
All SDK package licenses accepted

C:\Users\91876>
```

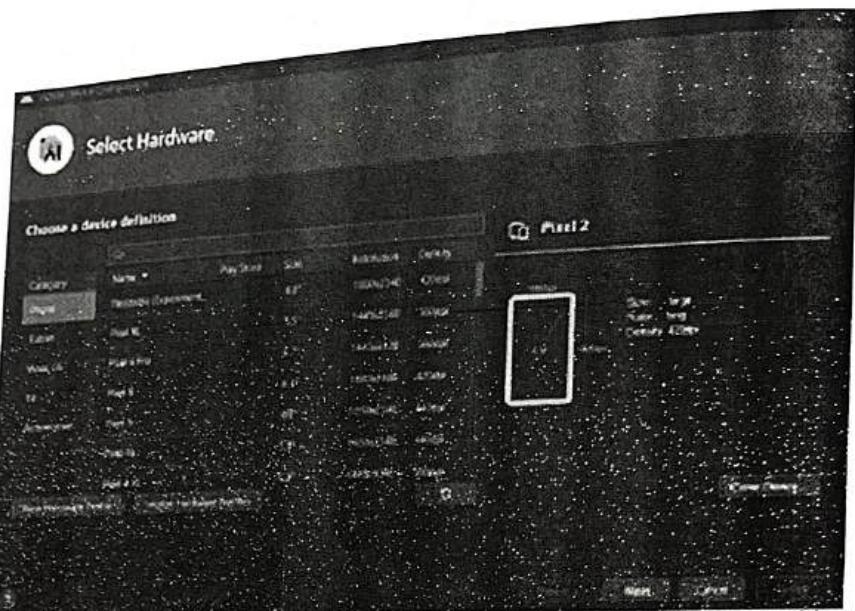
**Step 7:** Next, you need to set up an Android emulator. It is responsible for running and testing the Flutter application.



**Step 7.1:** To set an Android emulator, go to **Android Studio > Tools > Android > AVD Manager** and select **Create Virtual Device**. Or, go to **Help->Find Action->Type Emulator** in the search box. You will get the following screen.

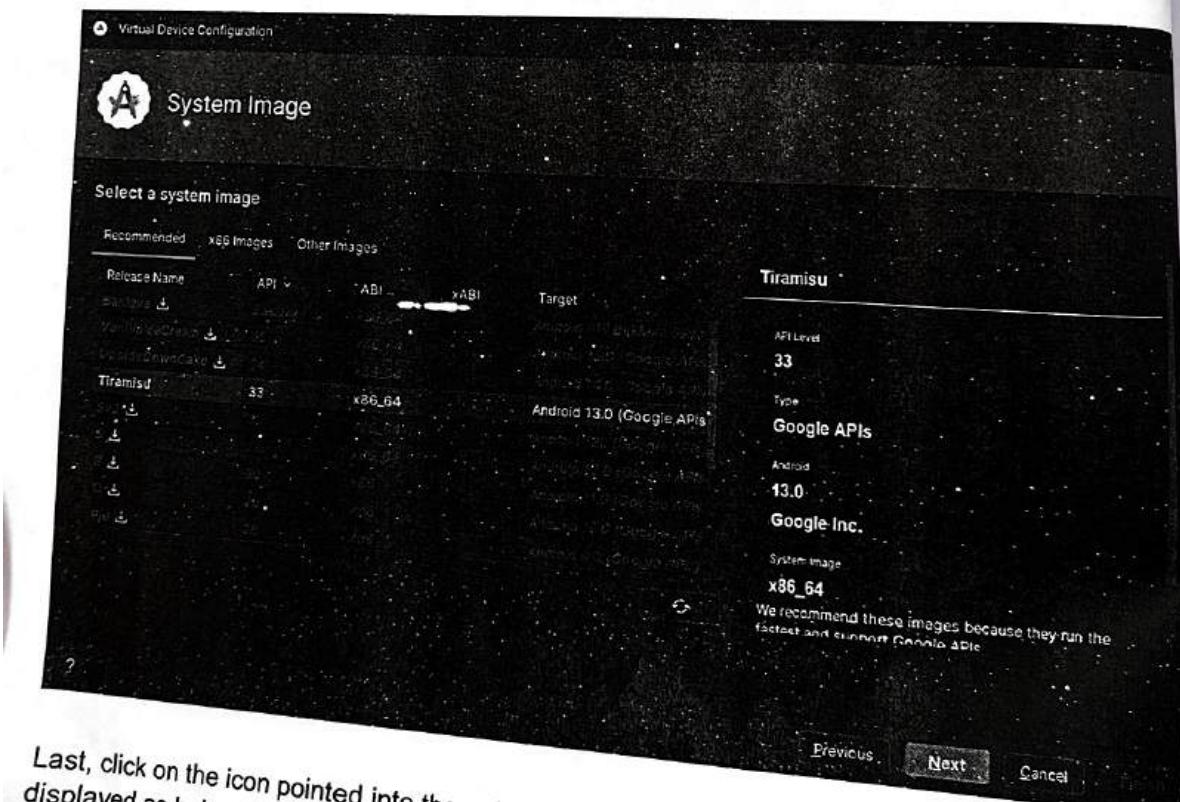


**Step 7.2:** Choose your device definition and click on **Next**.

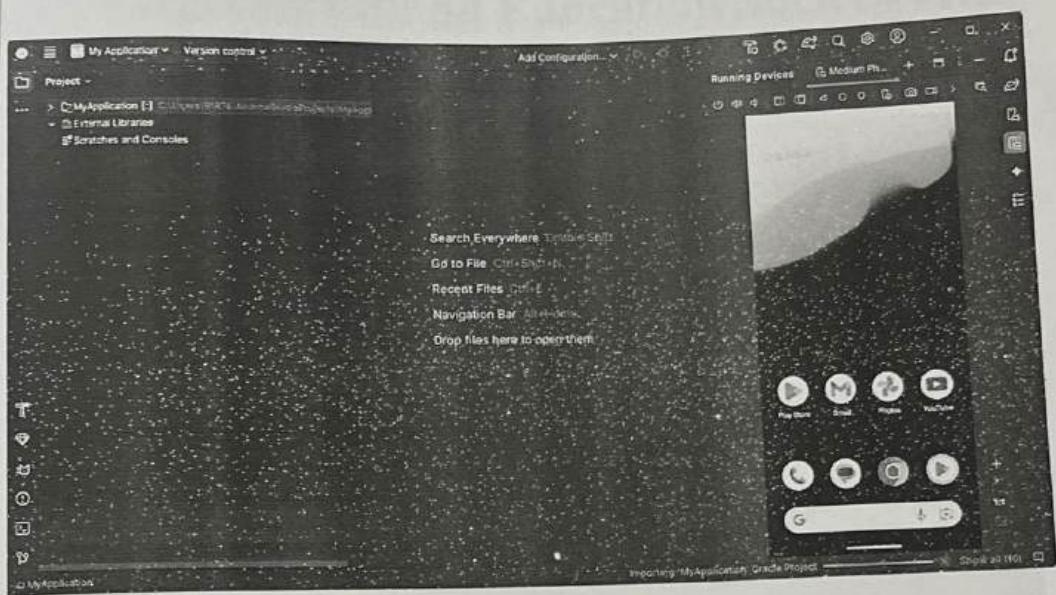


7.3: Select the system image for the latest Android version and click on Next.

**Step 7.4:** Now, verify the all AVD configuration. If it is correct, click on Finish. The following screen appears.

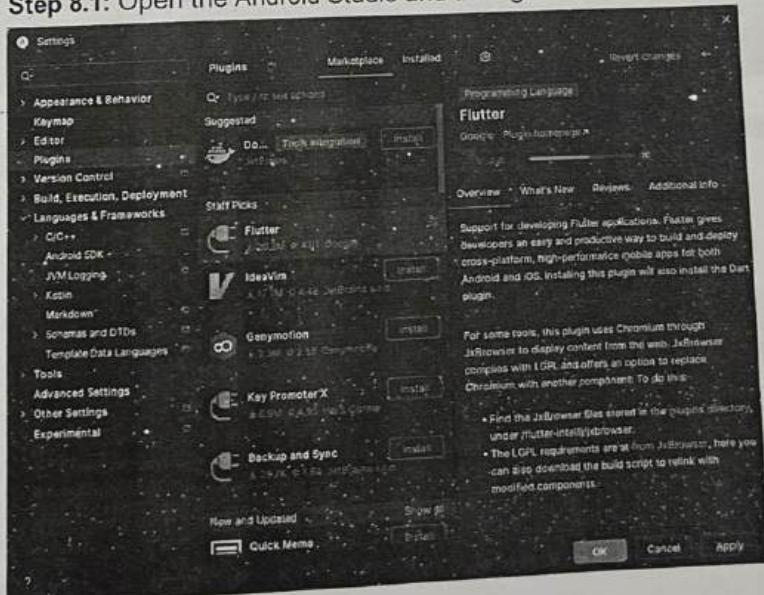


Last, click on the icon pointed into the red color rectangle. The Android emulator displayed as below screen.

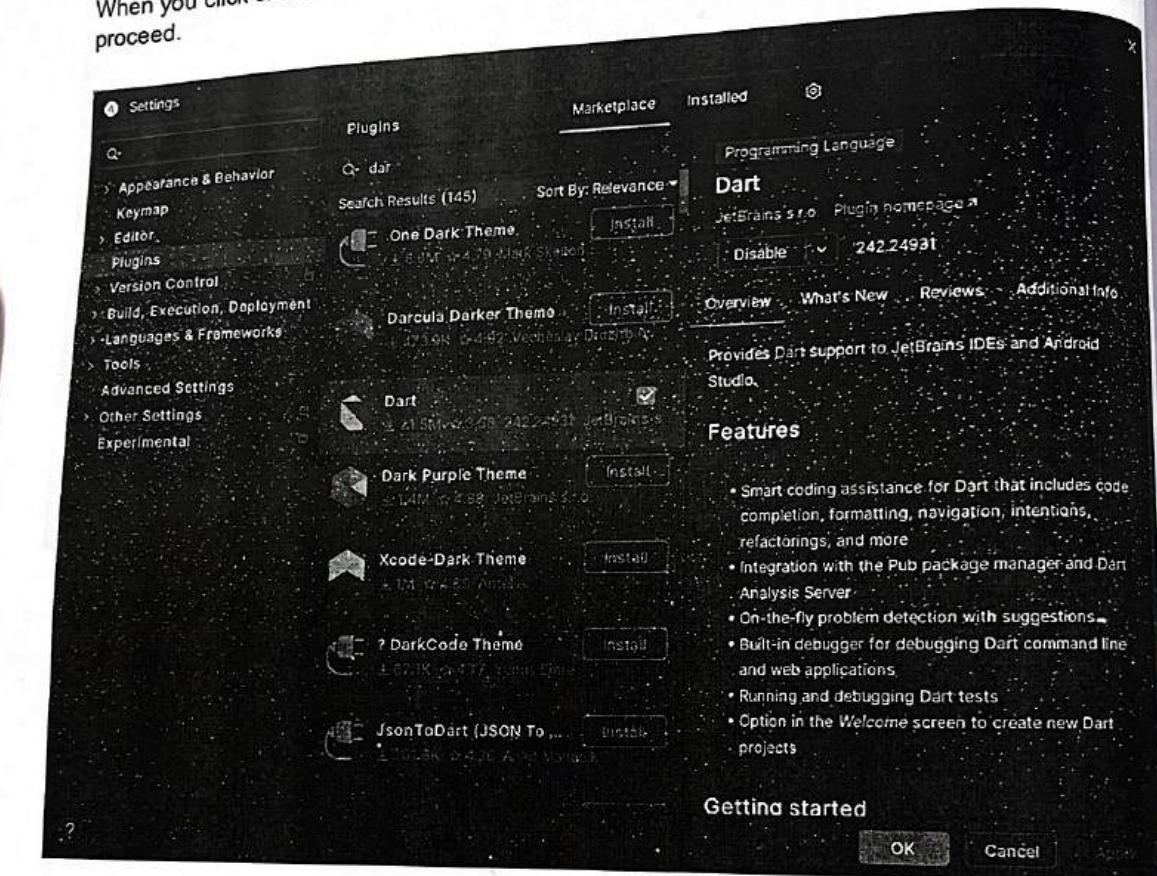


**Step 8:** Now, install Flutter and Dart plugin for building Flutter application in Android Studio. These plugins provide a template to create a Flutter application, give an option to run and debug Flutter application in the Android Studio itself. Do the following steps to install these plugins.

**Step 8.1:** Open the Android Studio and then go to File->Settings->Plugins.



**Step 8.2:** Now, search the Flutter plugin. If found, select Flutter plugin and click install. When you click on install, it will ask you to install Dart plugin as below screen. Click yes to proceed.



**Step 8.3:** Restart the Android Studio.

### Conclusion:

Installing Flutter and setting up the development environment is the first step towards building cross-platform applications. By following the proper installation steps, developers can leverage Flutter's fast development cycle, expressive UI, and performance to create high-quality applications for multiple platforms, all from a single codebase.



**Vivekanand Education Society's  
Institute of Technology**

(An Autonomous Institute Affiliated to University of Mumbai)

**Department of Information Technology**

A.Y. 24-25

**Mobile App Development and Progressive Web App Lab**

Experiment No.	02
Experiment Title.	To design Flutter UI by including common widgets.
Roll No.	57
Name	Hitesh Tanweiri
Class	D15 B
Subject	MAD & PWA
Grade:	

HITESH TANWANI  
DEP DIS B 57  
6 MPL

Exp - 2

DATE:

Aim - To design flutter UI by excluding common widgets.

Theory -

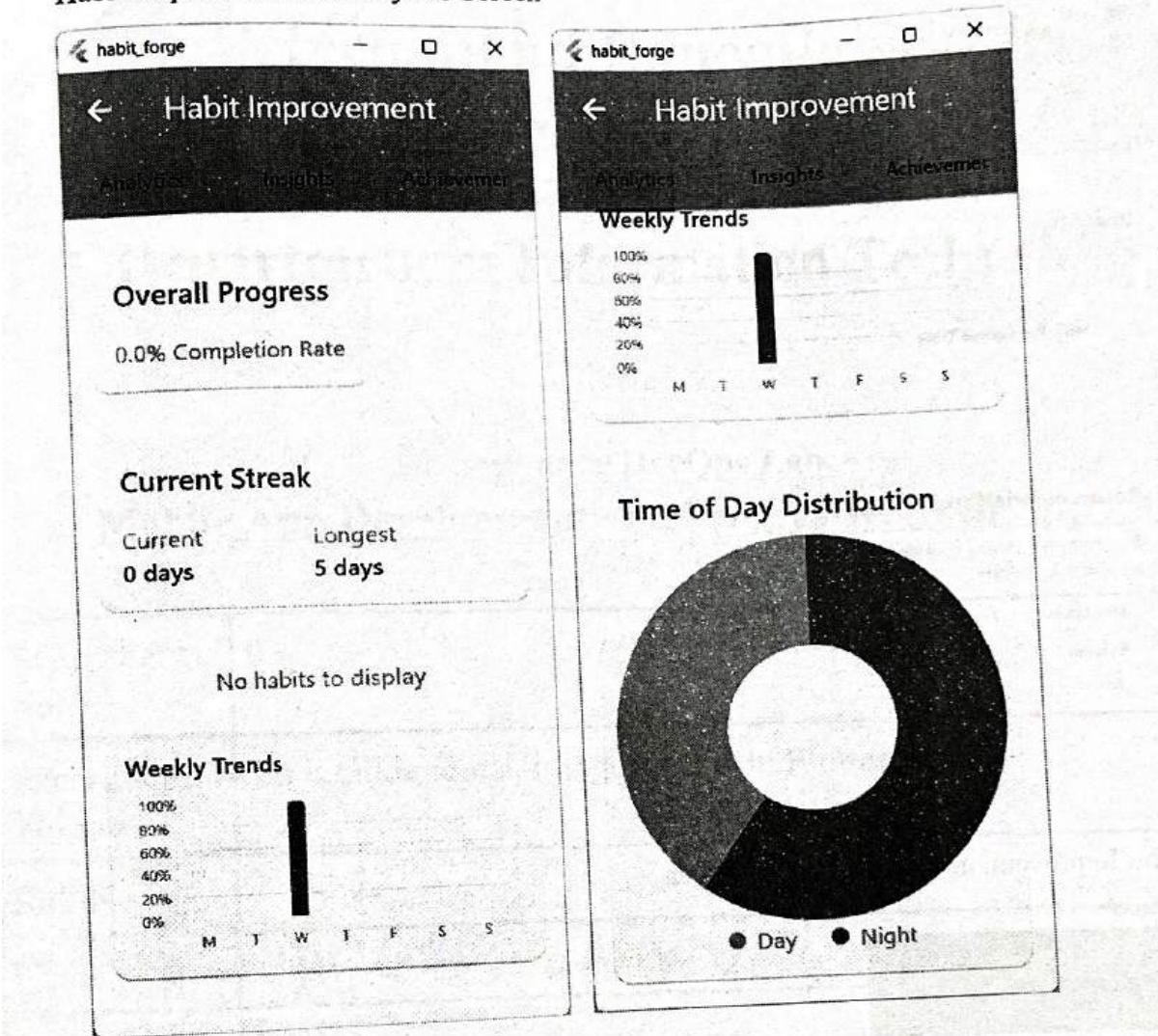
commonly used flutter widgets are scaffold, Bottom Navigation Bar & buttons.

This helps in understanding the layout structure, state management and interacting flutter applications. The experiment focuses on creating accessibility & visibility in mind.

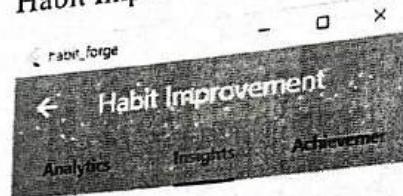
Conclusion

This exp helped us in understanding the importance of widgets, hierarchy & UI operation in flutter development.

## Habit Improvement / Analytics Screen



## Habit Improvement / Insights screen



### Insights

Best Performing Day

Most Productive Time

### Recommendations

Try setting a morning routine based  
on your success rate

Meditation   Exercise  
Reading

## Habit Improvement / Achievements screen





**Vivekanand Education Society's**  
**Institute of Technology**  
(An Autonomous Institute Affiliated to University of Mumbai)  
**Department of Information Technology**

A.Y. 24-25

Advance DevOps Lab  
Mobile App Development and Progressive Web App Lab

Experiment No.	03
Experiment Title..	To include icons, images, fonts in Flutter app
Roll No.	57
Name	Hitesh Tanwar
Class	D15 B
Subject	MAD & PWA
Grade:	

HITESH TANWANI  
57 DISB

MPL

ENP-3

DATE:

Aim -

To integrate icons & images and custom fonts into flutter app.

Theory - Flutter provides various ways to include typography in a app.

Icon flutter has built in icons from the material library & supports custom icons & images.

Fonts - Custom fonts can be added by defining them in pubspec.yaml file.

Conclusion - Adding icons, images & fonts in flutter enhances the app visual appeal & branding.

## Login

Email

Password

Login

[Don't have an account? Sign up](#)

## ← Sign Up

Email

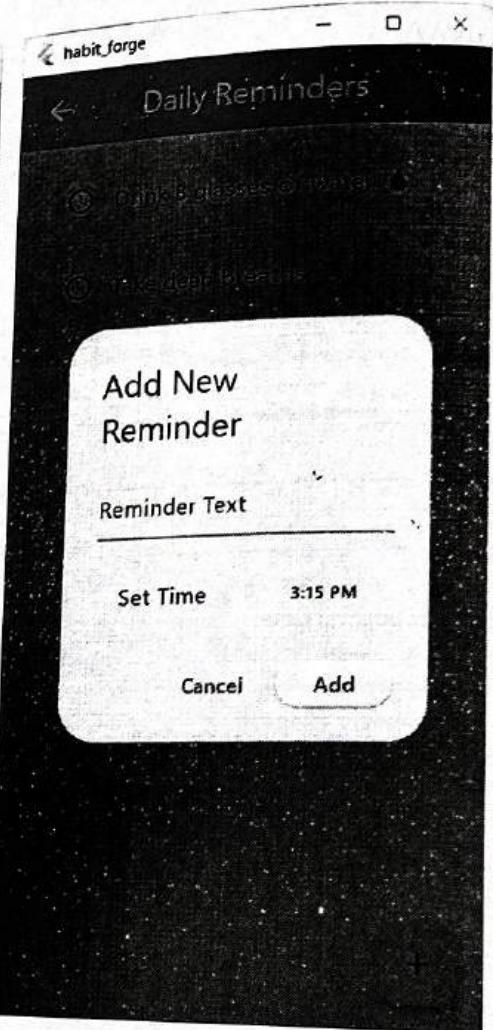
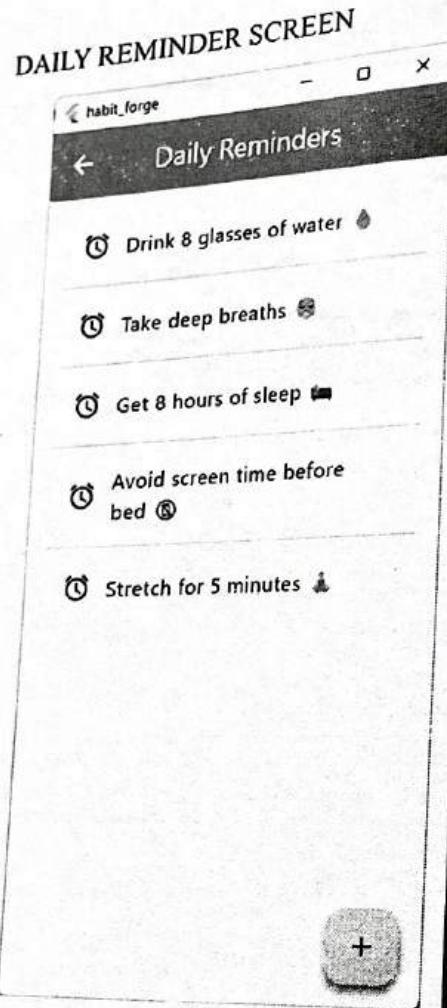
Password

Confirm Password

Sign Up

[Already have an account? Login](#)

## DAILY REMINDER SCREEN





## Vivekanand Education Society's Institute of Technology

(An Autonomous Institute Affiliated to University of Mumbai)

### Department of Information Technology

A.Y. 24-25

#### Mobile App Development and Progressive Web App Lab

Experiment No.	04
Experiment Title.	To create an interactive Form using form widget.
Roll No.	57
Name	Hitesh Tewani
Class	D15 B
Subject	MAD & PWA
Grade:	

HITESH  
DISB  
57  
MPL

TANWANI

## Experiment - 04

DATE:

Aim : Create an interactive form using form widget.

Theory : Intro to forms in flutter  
Form in flutter allows user to enter & submit data. The form widget is used to group multiple form fields such as text form field, manage their validation components of flutter form

- 1) Global key <form state>: helps in managing the form's state
- 2) Text form field : Allows users to input text
- 3) Validation logic: It should include validation & submit the form

### Implementation Steps

1. Create a form with a global key <form state>
2. Add a text field with validation logic
3. Create a btn to validate & submit the form

Conclusion:- The experiment demonstrate how to create an interactive form in flutter using the form widget.

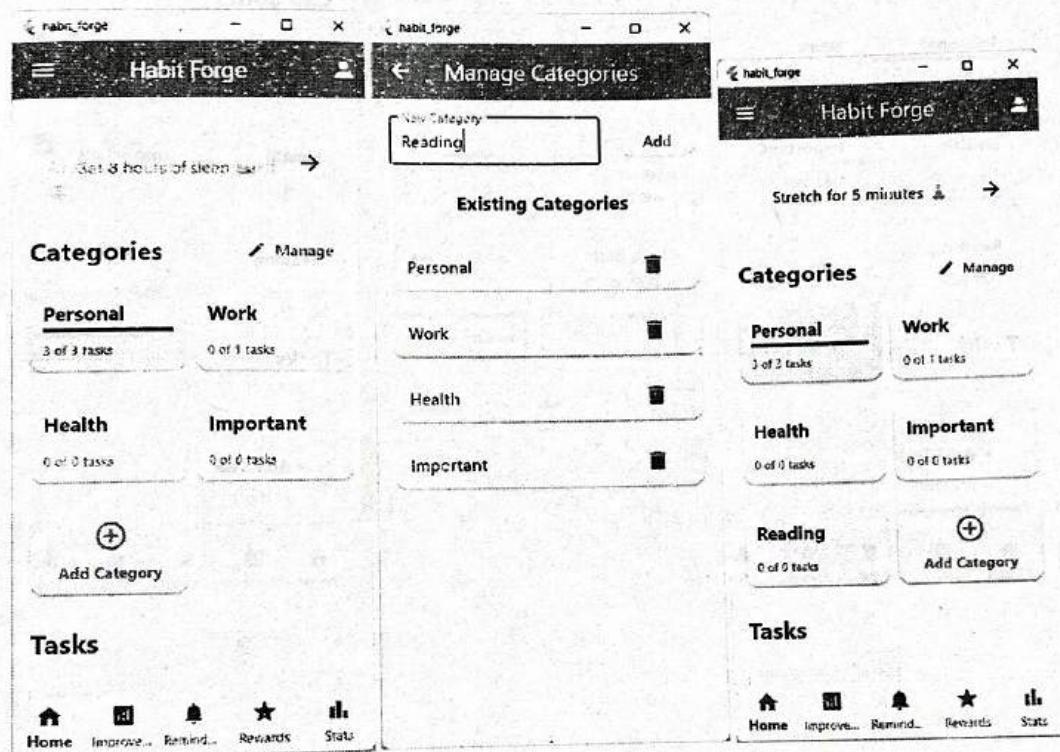
## MPL EXPERIMENT NO: 04

HITESH TANWANI

D15B - 57

Exp 4 To create an interactive Form using form widget

Home / add category



## Home/ add new task

The image displays three screenshots of the Habit Forge mobile application interface.

**Left Screenshot (Home Screen):**

- Header:** habit\_forge, Habit Forge, Categories.
- Categories Section:**
  - Personal:** 3 of 3 tasks
  - Work:** 0 of 1 tasks
  - Health:** 0 of 0 tasks
  - Important:** 0 of 0 tasks
  - Reading:** 0 of 0 tasks
- Tasks Section:**
  - A large plus sign (+) button labeled "+ Add Task".
  - Text: "0/0 tasks".
- Bottom Navigation:** Home, Improve, Remind, Rewards, Stats.

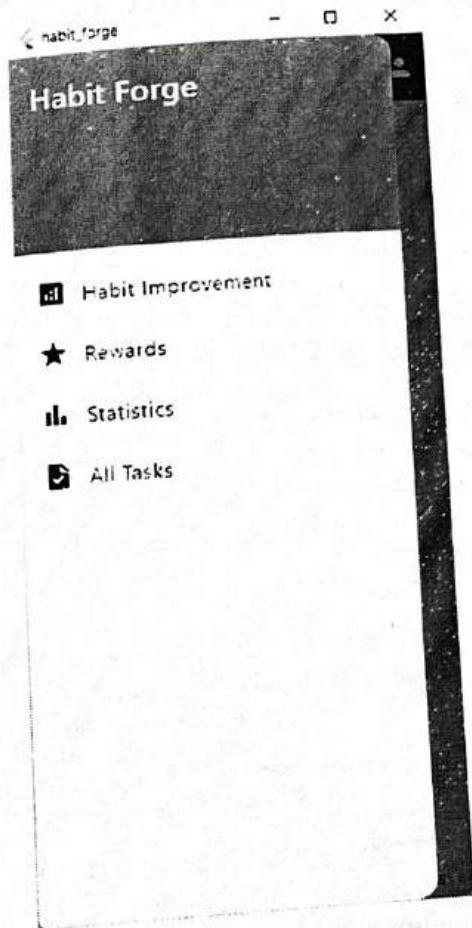
**Middle Screenshot (Add New Task Screen):**

- Header:** habit\_forge, Add New Task.
- Form Fields:**
  - Task Title: run
  - Description: run 3kms in evening today
  - Category: Health (dropdown menu)
  - Due Date: 2025-04-16
  - Frequency: 1
  - Frequency Type: weekly (dropdown menu)
- Buttons:** Add Task.

**Right Screenshot (Home Screen):**

- Header:** habit\_forge, Habit Forge, Categories.
- Categories Section:**
  - Personal:** 3 of 4 tasks
  - Work:** 0 of 1 tasks
  - Health:** 0 of 1 tasks
  - Important:** 0 of 1 tasks
  - Reading:** 0 of 0 tasks
- Tasks Section:**
  - A large plus sign (+) button labeled "+ Add Task".
  - Text: "0/0 tasks".
- Bottom Navigation:** Home, Improve, Remind, Rewards, Stats.

navigation menu





## Vivekanand Education Society's Institute of Technology

(An Autonomous Institute Affiliated to University of Mumbai)

### Department of Information Technology

A.Y. 24-25

#### Mobile App Development and Progressive Web App Lab

Experiment No.	05
Experiment Title.	To apply navigation, routing and gestures in Flutter App.
Roll No.	57
Name	Hitesh Tanwani
Class	D15 B
Subject	MAD & PWA
Grade:	

HITESH TAWANI  
D15B 5F  
MPL

## Experiment - D5.

DATE:

dim: To apply navigation, routing & gestures in flutter app.

Theory :

Navigat<sup>n</sup> in flutter  
Navigation in flutter refers to switching between different screen (pages) in an application. It is managed using navigator widget, which maintains a stack of routes.

Types of Navigation

1. Push Navigation:- (Navigator.push) adds new screen on top
2. Pop Navigation:- (Navigator.pop) removes the current
3. Named routes:- (Navigator.pushNamed) Navigation using pre-defined route's name

## Routing In Flutter

Routing is the mechanism that determines how a user navigates between different pages.

## Types of Routing

1. Imperative
2. Declarative

## Gestures in flutter

Flutter provides the `GestureDetector` widget to handle user touch interactions like: tapping, swiping & dragging.

1. Tap gesture

2. Double tap gesture

## Conclusion:

Navigation allows smooth transition between different pages. Routing makes the app structured & easier to manage. Gesture enhances user interaction by enabling touch based navigation.

MPL EXPERIMENT NO: 05

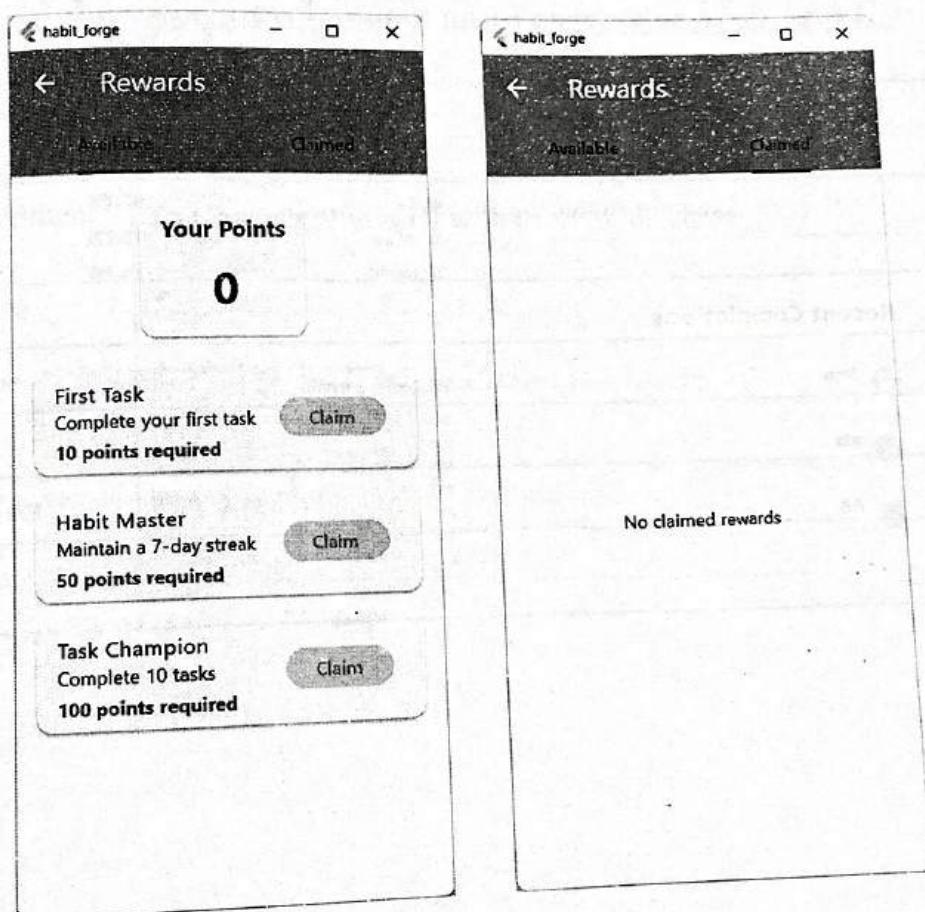
HITESH TANWANI

D15B - 57

Exp 5 To apply navigation, routing and gestures in Flutter App

## Rewards

Rewards/Available and Rewards/Claimed



## Statistics





**Vivekanand Education Society's  
Institute of Technology**

(An Autonomous Institute Affiliated to University of Mumbai)

**Department of Information Technology**

A.Y. 24-25

**Mobile App Development and Progressive Web App Lab**

Experiment No.	06
Experiment Title.	To Connect Flutter UI with fireBase database
Roll No.	57
Name	Hitesh Tamwani
Class	D15 B
Subject	MAD & PWA
Grade:	

HITESH TANWANI  
DISB 7  
MPL

## Experiment 6

DATE:

Aim:- Setting up firebase for flutter for android & ios apps

Theory:-

What is firebase?  
Firebase is a Back-end - as - a - service (BaaS) provided by google that helps developers build mobile & web application with features like authentication, real time database, cloud storage & analysis.

Steps to set up firebase in a flutter app

1. Create a firebase
2. add firebase to your flutter project
3. Install firebase packages in flutter
4. Initialize
5. verify firebase connection .

Conclusion :-

Firebase is a powerful Baas that simplifies files backend development for mobile apps. The integration process include creating a firebase project adding platform - specific configuration files & initializing firebase

## MPL FIREBASE EXPERIMENT 06

NAME : HITESH TANWANI

CLASS : D15B

ROLL\_NO : 57

The screenshot shows the 'Create a project' screen in the Firebase console. The project name is 'habitForge'. There are three checkboxes at the top:

- I accept the [Firebase Terms of Service](#)
- I confirm that I will use Firebase exclusively for purposes relating to my hobby, business, craft, or profession
- Join the [Google Developer Group](#) to enrich your developer journey with access to AI assistance, learning resources, solve badges, and more!

Below these are two buttons: 'Already have a Google Cloud project? Add Firebase to Google Cloud project' and a large 'Continue' button. On the right, there's a decorative illustration of a person standing next to a laptop with gears.

**x Create a project**

habitForge

I accept the [Firebase Terms of Service](#)

I confirm that I will use Firebase exclusively for purposes relating to my hobby, business, craft, or profession

Join the [Google Developer Group](#) to enrich your developer journey with access to AI assistance, learning resources, solve badges, and more!

Already have a Google Cloud project? [Add Firebase to Google Cloud project](#)

**Continue**

**x Create a project**

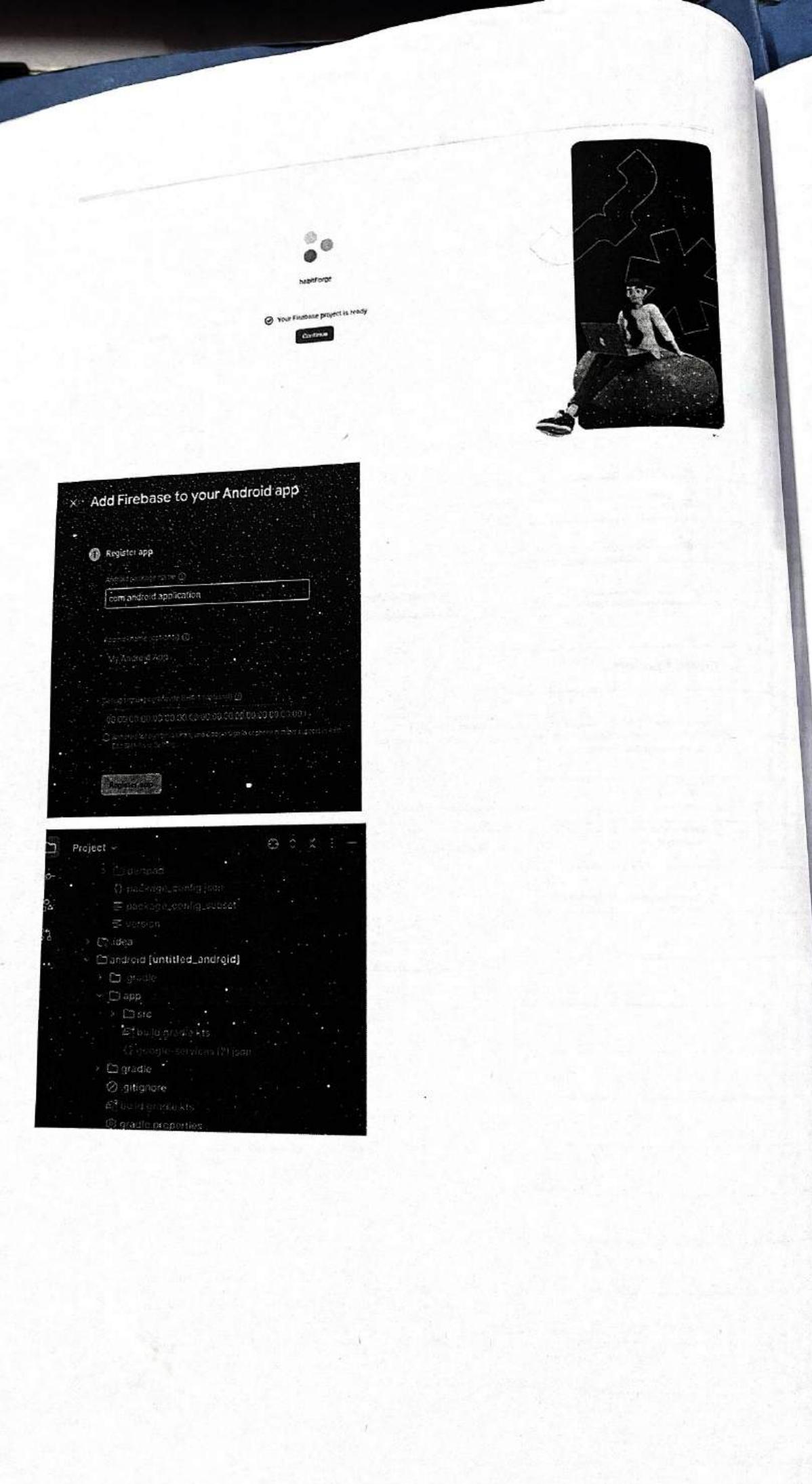
Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In App Messaging, Remote Config, A/B Testing, and Cloud Functions.

Google Analytics enables:

- A/B testing
- User segmentation & targeting across Firebase products
- Breakdown logs in Crashlytics
- Event-based Cloud Functions Triggers
- Free unlimited reporting

Enable Google Analytics for this project Recommended

**Previous** **Continue**



```
plugins {
    id("com.android.application")
    id("kotlin-android")
    // The Flutter Gradle Plugin must be applied after the Android and Kotlin Gradle plugins
    id("dev.flutter.flutter-gradle-plugin")
    id('com.google.gms.google-services')
}

android {
    namespace = "com.example.unitedad"
    compileSdk = flutter.compileSdkVersion
    ndkVersion = flutter.ndkVersion

    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }
}
```



**Vivekanand Education Society's  
Institute of Technology**

(An Autonomous Institute Affiliated to University of Mumbai)

**Department of Information Technology**

A.Y. 24-25

**Mobile App Development and Progressive Web App Lab**

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.	57
Name	Hitesh Tamang
Class	D15 B
Subject	MAD & PWA
Grade:	

## Experiment-7

DATE:

dim: To develop a progressive web App (PWA) that can be installed on a device like a native app and provides offline functionality using a service worker and a web manifest file.

### Theory :

A progressive web App (PWA) is a modern web app that provides an app-like experience on the web. It allows users to install the app on their device, work offline and load content faster. PWAs leverage the following key components

#### 1. Service Worker

A background script that enables offline functionality by caching resources. It intercepts network requests and serves cached responses when offline.

#### 2. Web App Manifest

A JSON file that provides metadata about the PWA. Such as the app's name, theme, start URL and icons. It makes the web app installable on device.

### 3. Caching Strategy

The cache first approach is used in PWA meaning that if resources are available in cache they are served otherwise they are fetched from network.

### Conclusion:-

The implementation successfully demands the PWA architecture by enabling installation, offline access and caching. Users can install the app & even if there is ~~not~~ no internet connection, previously loaded pages will be still accessible.

## PWA EXPERIMENT - 07

Name- Hitesh Tanwani

Roll no-57

D15B

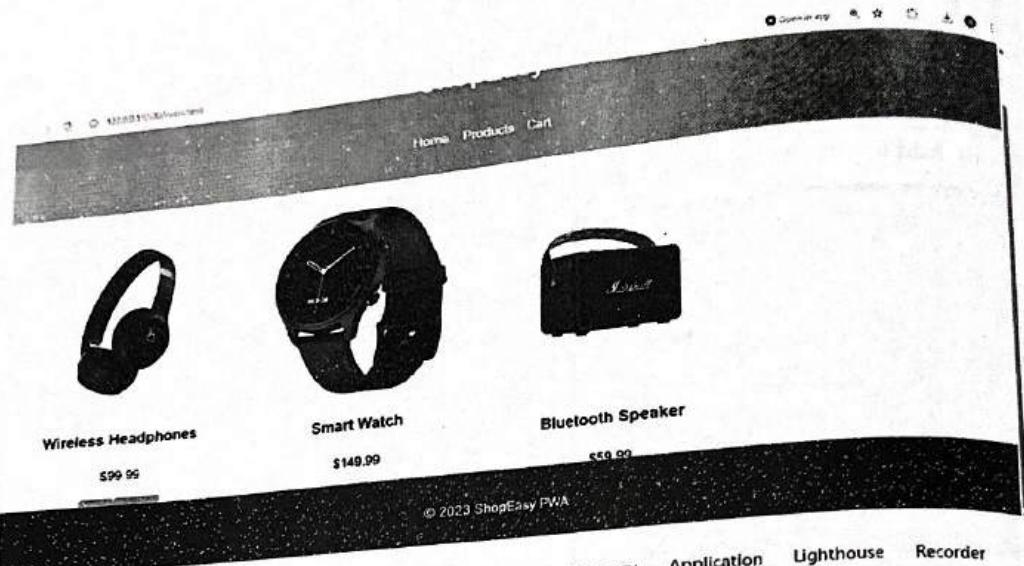
Aim: Add to your home screen feature on a web application.



The image shows a screenshot of a code editor displaying the content of the manifest.json file. The manifest.json file defines a service worker for a PWA named "ShopEasy". It includes icons for different sizes (192x192 and 512x512 pixels) and specifies the start URL as "/". The manifest also defines a theme color (#4287f5) and a background color (#fffff1).

```
name: "ShopEasy",
short_name: "ShopEasy",
start_url: "/",
scope: "/",
icons: [
  {
    "src": "icons/icon-192x192.png",
    "sizes": "192x192",
    "type": "image/png",
    "purpose": "any maskable"
  },
  {
    "src": "icons/icon-512x512.png",
    "sizes": "512x512",
    "type": "image/png",
    "purpose": "any maskable"
  }
],
theme_color: "#4287f5",
background_color: "#fffff1",
display: "standalone",
orientation: "portrait"
```

On adding Icon images and manifest.json file to the file structure, we could see the option to install the website as if it were an application.



© 2023 ShopEasy PWA

Application Elements Sources Net... Performance Memory Application Lighthouse Recorder

Service workers

Offline  Update on reload  Bypass for network

Service workers from other origins

[See all registrations](#)

Storage

Local storage  Session storage  Extension storage  IndexedDB

ServiceWorker

Open DevTools sidebar and press F12 to enable Service Worker for debugging.

Registration ID: C:\Users\H1479\appData\Local\Temp\CurrentUser\Script\07

Script identifier: file:///C:/Users/H1479/appData/Local/Temp/CurrentUser/Script/07

Script origin: chrome-extension://file:///C:/Users/H1479/appData/Local/Temp/CurrentUser/Script/07

Script type: module

Script status: ACTIVATED

Running status: STOPPED

Fetch handler status: NOT HANDLED

Fetch handler type: NO HANDLER

Script creation timestamp: 2023-07-10T10:45:23.000Z

Version process ID: 0

Version status ID: -1

DevTools agent status ID: -2

Log

Dispatcher: Start

Registration ID: 104

Script identifier: file:///C:/Users/H1479/appData/Local/Temp/CurrentUser/Script/07

Script origin: chrome-extension://file:///C:/Users/H1479/appData/Local/Temp/CurrentUser/Script/07

Script type: module

Script status: ACTIVATED

Running status: STOPPED

Fetch handler status: NOT HANDLED

Fetch handler type: NO HANDLER

Script creation timestamp: 2023-07-10T10:45:23.000Z

Version process ID: 0

Version status ID: -1

DevTools agent status ID: -2

Log



**Vivekanand Education Society's  
Institute of Technology**

(An Autonomous Institute Affiliated to University of Mumbai)

**Department of Information Technology**

A.Y. 24-25

**Mobile App Development and Progressive Web App Lab**

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.	57
Name	Hitesh Tanwani
Class	D15 B
Subject	MAD & PWA
Grade:	

HITESH TAWARI  
DISB 57  
MPL

## EXPERIMENT - 8

DATE:

Aim:- To develop progressive web App (PWA) that enables installation offline functionality using a service worker and a web manifest file.

### Theory:-

A progressive web App (PWA) is a modern web app that provides an app like experience on the web & on the web.

It allows user to install the app on their device, work offline and load content faster. PWAs leverage the following key components:

#### 1. Service Worker:

A background script that enables offline functionality by caching resources. It intercepts network requests and serves cached response when offline.

#### 2. Web App Manifest

A JSON file that provides metadata about the PWA, such as the app's name, theme, start url and icons. It makes the web app installed on device.

### 3. Caching strategy

The cache first approach is used in PWA meaning that if resources are available in cache they are served otherwise they are fetched from the network.

### Conclusion:

The implementation successfully demonstrated the architecture by enabling installation, offline access and caching where users can install the app even if there is no internet connection previously loaded pages will still be accessible.

## EXPERIMENT 08

NAME : Hitesh Tanwani

CLASS : D15B

ROLL\_NO : 57

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.

Create service-worker.js



```
1 // This is a manifest file at the root of your web app.
2 // See https://developers.google.com/web/fundamentals/web-components/codelab
3 // Learn how to build a service worker by reading https://bit.ly/CRA-PWA
4
5 const CACHE_NAME = 'v1';
6 const urlsToCache = [
7   '/',
8   '/index.html',
9   '/index.css',
10  '/index.js',
11  '/images/icon-192x192.png',
12  '/images/icon-512x512.png'
13 ];
14
15 // This event listener is triggered when a page is being loaded
16 self.addEventListener('install', event => {
17   console.log('Service worker installing');
18   event.waitUntil(
19     caches.open(cacheName)
20       .then(cache => {
21         return cache.addAll(urlsToCache);
22       })
23     );
24 });
25
26 // This event listener is triggered when a page is being activated
27 self.addEventListener('activate', event => {
28   console.log('Service worker activating');
29   event.waitUntil(
30     caches.keys().then(keys => {
31       return Promise.all(
32         keys.map(key => {
33           if (key !== CACHE_NAME) {
34             return caches.delete(key);
35           }
36         })
37       );
38     })
39   );
40 });
41
42 // This event listener is triggered when a request is made for a resource
43 self.addEventListener('fetch', event => {
44   event.respondWith(
45     caches.match(event.request)
46       .then(response => {
47         if (response) {
48           return response;
49         }
50         return fetch(event.request);
51       })
52     );
53 });
54
55 // If you want to test your service worker without enabling it,
56 // add the following to your app's manifest:
57 // "serviceWorker": "script:myservice-worker.js"
```

Create a cacheable file called offline.html to be displayed in the absence of an internet connection.



A screenshot of a code editor showing the content of an offline.html file. The file contains HTML and CSS code for a service worker. The CSS defines a body with a font-family of Arial, sans-serif, text-align: center, padding: 2rem, and background-color: #f5f5f5. It also includes a h1 element with a color of #4285f4 and a div with a width of 100px and margin: 1rem. The HTML part includes a title, a meta tag for viewport, and a button with an onclick event that navigates back to the home page.

```
<html>
  <head>
    <title>offline | ShopEasy</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
      body {
        font-family: Arial, sans-serif;
        text-align: center;
        padding: 2rem;
        background-color: #f5f5f5;
      }
      h1 {
        color: #4285f4;
      }
      div {
        width: 100px;
        margin: 1rem;
      }
    </style>
  </head>
  <body>
    
    <h1>You're offline!</h1>
    <p>No internet connection detected.</p>
    <p>You can still browse cached pages. We'll sync your data when you're back online.</p>
    <button onclick="window.location.href = '/'>Return to Home</button>
  </body>
</html>
```





**Vivekanand Education Society's  
Institute of Technology**

(An Autonomous Institute Affiliated to University of Mumbai)

**Department of Information Technology**

A.Y. 24-25

**Mobile App Development and Progressive Web App Lab**

Experiment No.	09
Experiment Title.	To implement Service worker events like fetch, sync and push for E-commerce PWA.
Roll No.	57
Name	Hitesh Tamwani
Class	D15 B
Subject	MAD & PWA
Grade:	

HITESH TANWAR  
DISB 57

MPL

## EXPERIMENT - 9

DATE:

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

Theory:

Service worker is a background script that runs independently in a browser and enables functionalities like offline caching, background sync and push notification. It acts as a proxy between the web app and the network.

Key Features of Service workers:

- Runs in the background without user interaction
- Intercepts network requests and manages caching using the Cache API
- Works only over HTTPS for security reasons

\* Service Worker Events:

Fetch Event: - Manages network requests and caching

Sync Event: Enables background Sync when

DATE:

internet is restored

Push Event: Handles push notifications  
for updates, offers or order  
statuses

Conclusion: An application which  
uses service workers to provide  
offline functionality.

We successfully implemented services  
worker events like fetch, sync & push in  
an E-commerce PWA enabling offline  
functionalities, background sync & real-time  
push notifications to improve user experience.

PWA EXPERIMENT No : 09

NAME : Hitesh Tanwani

CLASS : D15B

ROLL\_NO : 57

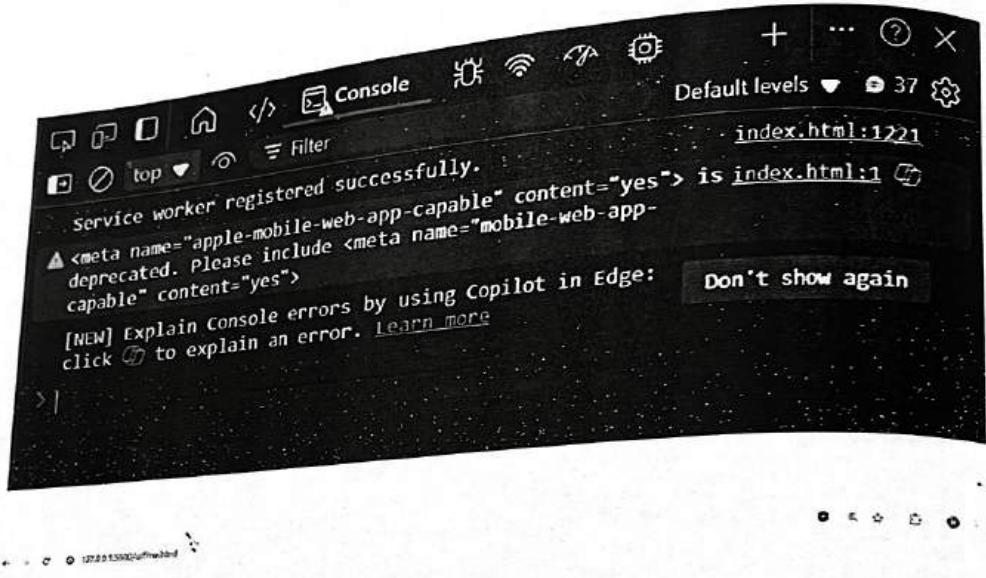
Aim: To implement Service worker events like fetch, sync and push for E-commerce PWA.

```
// Activate event
self.addEventListener("activate", event => {
  event.waitUntil(
    caches.keys().then(keys =>
      promise.all(
        keys.map(key => {
          if (key !== staticCacheName) {
            return caches.delete(key);
          }
        })
      )
    )
  );
}

// Fetch event
self.addEventListener("fetch", event => {
  event.respondWith(
    caches.match(event.request).then(response =>
      return response || fetch(event.request).catch(() => caches.match('./offline.html'));
    )
  );
});
```

Make the following changes to the service-worker.js

```
// Install Event: Cache assets
// Activate Event: Cleanup old caches
// Fetch Event: Supports both Cache-First & Network-First
// Sync Event: Retry sending data when online
// Func on to send pending screenshots to the server
// Push Event: Display push notifications
```



Conclusion: We implemented the functionality of offline web cache capture so that in the absence of a stable internet connection on, the app would display a generic waiting page.



**Vivekanand Education Society's  
Institute of Technology**

(An Autonomous Institute Affiliated to University of Mumbai)

**Department of Information Technology**

A.Y. 24-25

**Mobile App Development and Progressive Web App Lab**

Experiment No.	10
Experiment Title.	To study and implement deployment of Ecommerce PWA to GitHub Pages.
Roll No.	57
Name	Hitesh Tamwani
Class	D15 B
Subject	MAD & PWA
Grade:	

AIM: To study and implement the deployment for an E-commerce PWA to GitHub Pages.

THEORY:- GitHub Pages is a free hosting service provided by GitHub for deploying static websites directly from a repository.

\* Features of GitHub Pages :-

- supports Jekyll blogging
- allows custom URLs
- Provides an automatic page generator.

\* Pros of GitHub Pages

- free and easy to use
- direct integration with GitHub repositories.
- supports custom domain.

\* Cons of GitHub Pages

- code is public unless using a paid private repository
- No HTTPS support for custom domains
- limited plugin support for Jekyll.

- \* Firebase as an Alternative
- \* Firebase is a real time Backend platform by google of offering cloud based storage, and authentication services.

### Conclusion:-

This successfully developed an Ecommerce PWA using GitHub Pages, leveraging its free and easy setup for static website hosting.

## PWA EXPERIMENT NO : 10

NAME : Hitesh Tanwani

CLASS : D15B

ROLL\_NO: 57

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



GitHub Pages

Your site is live at <https://christianmewes134.github.io/commerce.html>

Build and deployment

Source: Deploy from a branch

Branch: Your GitHub Pages site is currently being built from the main branch. Learn more about configuring the publishing source for your site.

Main: main / push / force

Learn how to add a static theme to your site.

Your site was last deployed in the default environment by the main branch via the GitHub workflow.

Custom domain

Custom domains allow you to serve your site from a domain other than <https://christianmewes134.github.io/>.

ShopEasy

Home Products Cart

ShopEasy

Home Products Cart

Wireless Headphones Smart Watch

\$89.99 \$149.99

Add to cart Add to cart

Bluetooth Speaker

Bluetooth Speaker

Memory

Note: ShopEasy

Port name: ShopEasy

Compose App ID: https://christianmewes134.github.io/commerce.html

Make ID is not specified in the manifest, start, or LWP used instead. To specify an App ID that matches the current identity, set the LWP App ID.

Presentation

Start URL: https://christianmewes134.github.io/commerce.html

Theme color: #A42B54

Background colors

Background cache

Background font

Background size

Background type

Phone scaling mode



# Vivekanand Education Society's Institute of Technology

(An Autonomous Institute Affiliated to University of Mumbai)

## Department of Information Technology

A.Y. 24-25

### Mobile App Development and Progressive Web App Lab

Experiment No.	11
Experiment Title.	To use google Lighthouse PWA Analysis Tool to test the PWA functioning.
Roll No.	17
Name	Hitesh Tanwani
Class	D15 B
Subject	MAD & PWA
Grade:	

AIM:- To study and implement the google lighthouse PWA Analysis Tool test the progressive web App (PWA) functioning.

Theory:- Google Lighthouse is open source tool by google used to audit web application based on various parameters including performance PWA compliance accessibility and best practices.

#### Key Audit Metrics

1. Performance - Measures loading speed, rendering time and responsiveness.
2. PWA compliance - Ensures the app meets PWA standards such as service workers.
3. Accessibility - Evaluates the app's ability for differently abled users including screen-reader compatibility.

Lighthouse provides actionable insights to improve website efficiency ensuring a better user experience & higher engagement.

14257011

8713

19 DATE:

Conclusion:-

We used the google lighthouse PWA analysis tool to evaluate and optimize the PWA, improving its performances, accessibility and compliance with best practices.

After this we will implement - responsive design - making our site mobile friendly.

Then we will convert - conventional ACID - to mobile ACID.

Finally we will upload it - utilizing Cloudinary - to hold it without any cost.

Finally uploaded our web application to cloudinary - which will be accessible from anywhere in the world.

## PWA EXPERIMENT NO : 11

NAME : Hitesh Tanwani

CLASS : D15B

ROLL\_NO: 57

Aim: To study and implement google lighthouse PWA analysis toll to test the progressive Web App (PWA) functionality





# Vivekanand Education Society's Institute of Technology

(An Autonomous Institute Affiliated to University of Mumbai)

## Department of Information Technology

A.Y. 24-25

### Mobile App Development and Progressive Web App Lab

Experiment No.	Assignment-1
Title.	
Roll No.	57
Name	Hitesh Tanwani
Class	D15 B
Subject	MAD & PWA
Grade:	

(03)

A

## ASSIGNMENT - 1

Explain the key features and advantages of using Flutter for mobile app development

Flutter is a popular open source UI toolkit developed by Google for building natively compiled applications for mobile (iOS & Android) web and desktop from a single code base.

### Key Features of Flutter:

1. Single codebase: Write once, run on multiple platforms (iOS, Android, web)

2. Dart Programming: Uses Dart which is optimized for fast performance and ahead-of-time compilation

3. Hot Reload: Instantly reflects changes in the app without restarting, making development faster and more efficient

4. Rich Widget Library: Provides a vast collection of customizable widgets that support Material design and Cupertino styles for a native look and feel.

## Advantages of Using Flutter:

1. Faster Development Time
2. Cost Effective
3. Consistent UI

Q.1(b) Discuss how the Flutter framework differs from traditional approaches and why it has gained popularity in the developer community?

→ Flutter uses a single database for multiple platforms, unlike native development that requires separate code for iOS (Swift) & Android (Kotlin). It does not rely on platform-specific UI components instead render everything using its own Skia graphic engine, ensuring consistency. Unlike React Native, which uses JavaScript bridge, Flutter compiles directly to native ARM code, offering better performance. Its hot reload feature allows development to see changes instantly making development faster & more efficient.

Flutter has gained popularity due to its faster development, cost efficiency & cross-platform support. Business prefers it as it reduces development time & cost while delivering high performance apps.

- 2) Describe the concept of the widget tree in Flutter. Explain the widget composition is used to build complex UI
- In flutter everything is a widget (button, text, layout, etc). These widgets are arranged in hierarchical structure known as the widget tree. The widget tree determines the UI.

Widget composition to build complex UI:

- Flutter encourages a composition-based approach rather than inheritance
- Instead of creating large, monolithic widget, developers build small, reusable widget that are combined to form complex UIs.

e.g. A column widget can hold multiple Text & Button widget creating a structured layout.

Q. 2b Provide example of commonly used widgets & their role in creating a widget tree

→ 1) Structural widget

- scaffold :- Provide basic structure of screen.
- container:- used for layout styling.
- column & row:- used for vertical & Horizontal layout.

2) Interactive widget

- text field:- for user input
- elevated button:- clickable buttons

3) Styling Widget

- padding :- Add spacing around widget
- align center:- Adjust alignment

4) List & scrollable widget

- list view:- scrollable list
- grid view:- provide display items in grid.

```
on simple widget tree  
scaffold ( width: 1000,  
            height: 800,  
            appBar: AppBar (title: Text ("Flutter App"),  
            body: Column (children:  
                [Text ("Welcome to flutter !"),  
                 Elevated Button (onPressed: () {  
                     setState(() {  
                         child: Text ("Tick Me");  
                     });  
                 }),  
                 Text ("Tick Me");  
             ],  
         ),  
     );
```

Q) Discuss the importance of state management in flutter application.

Importance of State Management in flutter application state management refers to handling dynamic data that changes overtime.

In flutter, the UI rebuilds where the state changes, ensuring the app remains interactive & responsive. Proper state management helps in performance optimization, code maintainability & better UI behaviour.

- ex. simple widget tree

Q.3b compare and contrast the different state management approach in flutter approaches such as Set State, provider Riverpod. Provide scenarios where each approach is suitable.

#### → 1. Set State

##### • ~~SetState~~

setState is the simplest state management approach in Flutter. It's built into the framework and is ideal for managing small, local state within a single widget.

Pros - easy, no dependencies

Cons - limited, not scalable.

Best for: small apps, UI state (eg toggle buttons).

#### 2. Provider

~~Provider~~ is a popular state management package in Flutter, designed to share state across widgets while separating business logic from UI code.

Pros: separates UI from logic, reactive

Cons: verbose for simple cases

Best For: Medium apps (eg Auth, shopping cart).

3.

### 3. Riverpod

Riverpod builds on the concept of Provider but offers more advanced features like better testability, no reliance on the widget tree.

Pros: Decouples from widget tree, better testability

Cons: Steeper learning curve

Best for: Large apps with complex state, sync management, or advanced features (eg. fetching data from API's)

4

a) Explain the process of integrating firebase with a flutter application. discuss the benefits of using firebase as a backend solution.

Integrating firebase with flutter & its Benifits.

→ Integrating process

- Set up firebase console.
- Create a firebase project.
- Register the App for Android & ios
- Download & add google-service.json or Google Services - info.plist (ios)

dependencies:

- firebase\_core : latest version
  - firebase\_auth : latest version.
- yaml
- Initialize Firebase in Flutter

dart

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

Benefits:-

No need to manage servers. Provide authentication, database & cloud function, scalable & cost-effective.

b) highlight the firebase services commonly used in flutter development & provide brief overview of how data synchronization is achieved.

→ Firebase offers several powerful services commonly used in flutter development enabling easy integration for backend functionalities

### 1. ~~Firebase Authentication~~

Simplifies user authentication with methods like email / password, google sign in and anonymous login.

In flutter it is integrated using the firebase-auth plugin, enabling quick sign ups, log-ins

### 2 ~~Cloud Fire store~~

It is a NoSQL document-based database that supports real-time syncing.

The cloud-firestore plugin allows reading and writing documents, setting up - real time listeners.

### 3. Firebase Realtime Database

Unlike firestore, the realtime database uses a JSON tree structure for storing key values pairs.

In flutter, the firebase database plugin helps us set up listeners for real-time updates, making it easy to build apps like Chat applications.



# Vivekanand Education Society's

## Institute of Technology

(An Autonomous Institute Affiliated to University of Mumbai)

### Department of Information Technology

A.Y. 24-25

#### Mobile App Development and Progressive Web App Lab

Experiment No.	Assignment-2
Title.	
Roll No.	57
Name	Hitesh Tanwani
Class	D15 B
Subject	MAD & PWA
Grade:	

MPL Assignment - 2

DATE: 06/03/25

NAME : HITESH TANHANI

CLASS:- D15B

ROLL NO: 57

SUBJECT : MPL

ASSIGNMENT - 2

Define progressive Web App (PWA) and explain its significance in Modern web development discuss the key characteristics that differentiate PWAs from traditional mobile apps

A progressive Web App (PWA) is a type of web application that works like a mobile app but runs in a browser. It can be installed on a device, works offline and provides a fast and smooth user experience.

### Significance of PWA in Modern Web Development

1. Cross-Platform compatibility
2. Offline support
3. Fast performance
4. No App store required
5. Lower Development Cost.

### Key Differences Between PWA and Traditional mobile Apps:

Feature	PWA	Traditional Mobile App
Installation	Direct from browser	Download from App Store
Internet	Works offline	Usually requires internet
Performance	Fast with service workers	Faster but needs installation

updates	Automatic, no app store approval	Manual updates needed
Development cost	Lower (one codebase for all)	Higher (separate apps for each platform)

Q.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.

→ Responsive Web Design (RWD) is a technique that makes web pages adjust automatically to different screen size and devices. It ensures a good user experience on mobiles, tablets and desktops without needing separate versions of website.

### Importance of Responsive Design in PWAs

1. Better User Experience
2. Faster Load Time
3. SEO Benefits
4. Cost Effective

## Key Differences

- Responsive adapts dynamically to all screens
- Fluid resizes smoothly but may not be fully optimized
- Adaptive loads different layouts based on device type

3 Describe the lifecycle of service workers including registration, installation and activation phases.



## Lifecycle

A service worker is a script that runs in the background and helps a web app work offline, load faster, and send push notifications. Its lifecycle has 3 main phases ..

### 1. Registration Phase

- The browser registers the service worker using Javascript.

Code Example:

```
if ('serviceWorker' in navigator) {  
    navigator.serviceWorker.register('/sw.js')  
        .then(() => console.log('Service Worker  
Registered'))  
        .catch(error => console.log('Registration  
Failed:', error))
```

This tells the browser to install and activate the Service Worker.

## 2. Installation Phase

- The Service Worker downloads necessary files (HTML, CSS, JS) and stores them in cache
- If successful, it moves to the activation phase

## Code Example:

```
self.addEventListener('install', event => {
  event.waitUntil(
    caches.open('app-cache').then(cache => {
      return cache.addAll(['/index.htm',
        '/style.css']);
    })
  );
});
```

This ensures the app loads even without the internet

### 3. Activation Phase

- The Old Service Worker is replaced with new One
- Unused cache files the previous version are deleted.

code Example:

```

self.addEventListener('activate', event =>
  event.waitUntil(
    cache.keys().then(keys => {
      return Promise.all(keys.map(key => {
        if (key !== 'app-cache') {
          return caches.delete(key);
        }
      }));
    })
  );
)
);

```

- The Service Worker is now fully active and control network requests.

### Final Step: Fetch & Sync

Once activated, the Service Worker intercepts network requests, serves cached files, and syncs data when the internet is available.

The lifecycle makes PWAs faster, more reliable, and capable of working offline.

DATE: 4

4 Explain the use of Indexed DB in the Service Worker for Data Storage

Indexed DB is a client-side NoSQL database used in Service Workers for storing structured data, enabling offline access, caching and background synchronization in web application.

Why Use IndexedDB in Service Workers?

- Offline storage: Saves API responses and users data for offline use.
- Persistent storage: Retains data even after page reloads.
- Data synchronization: Updates stored data when internet is available.
- Asynchronous & Non Blocking:

IndexedDB runs in the background, prevent performance issues.