

Name :- Abhishek R. Chauhan

Lab = MPL

Practical No. 1

DOMS	Page No.
Date	/ /

Objective :-

The primary objective of this flutter App is to provide users with a seamless and efficient way to convert media files from one format to another. This app eliminates such barriers by simple yet powerful media convert tool.

By Integrating the open-source FFmpeg library, the app will leverage its robust encoding, decoding, to handle a wide range of media formats.

Features:-

- 1) Support for multiple formats - MP4, MP3, MKV etc.
- 2) Batch conversion
- 3) Customizable output settings
- 4) Fast processing with FFmpeg
- 5) Minimalistic & intuitive UI
- 6) ~~Background~~ Background processing.
- 7) File Management & history.
- 8) preview Before conversion
- 9) Cloud storage for backup.

playstore ref link :- play.google.com/store/tp
details?id=com.Android.MediaConverter



Vivekanand Education Society's Institute of Technology

(An Autonomous Institute Affiliated to University of Mumbai)

Department of Information Technology

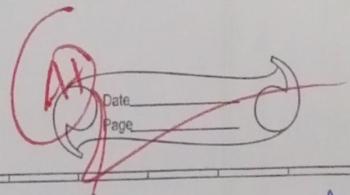
A.Y. 24-25

Mobile App Development and Progressive Web App

Experiment No.	01
Experiment Title.	To install and configure the Flutter Environment.
Roll No.	08
Name	Abhishek Chaurasiya
Class	D15 B
Subject	MAD & PWA
Grade:	(A) ✓

Abhishek R. Chaurasia

DISB - 08



Experiment 1

Installation and configuration of flutter environment

Aim :- To install and configure the flutter environment for cross-platform mobile development

Steps:-

- 1) Download flutter : visit flutter.dev and download the latest SDK
- 2) Setup Environment Variables:
 - Windows : Add C:\src\flutter\bin to the system path
- 3) Verify installation
 - Open terminal
flutter doctor
- 4) Set up an editor:
 - install Android Studio or VS code
- 5) Create and run a flutter project

Conclusion :

The flutter environment was successfully installed and configured, enabling cross-platform mobile app development.

Name: Abhishek Chaurasiya

Roll No:08

Div: D15B

MPL Experiment No.1

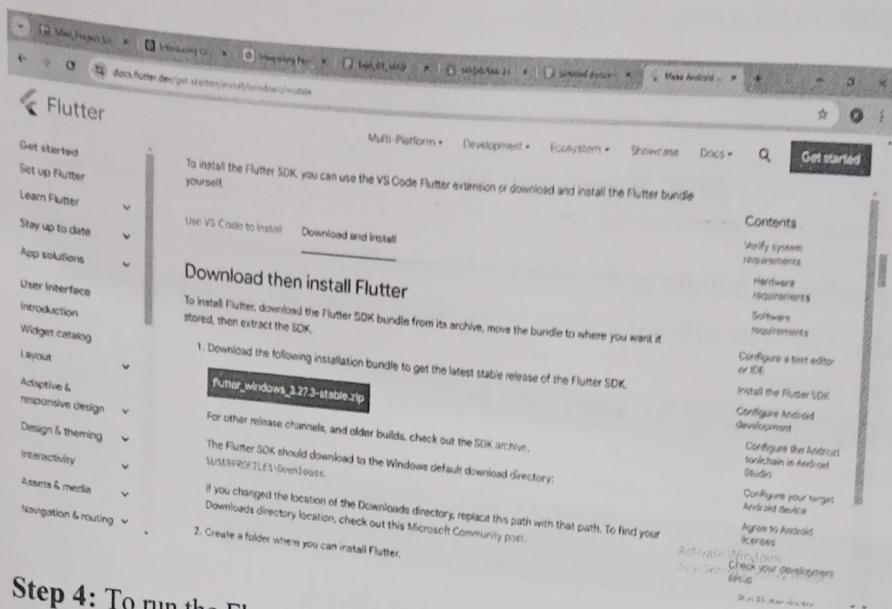
Step 1: Download the installation bundle of the Flutter Software Development Kit for windows.
To download Flutter SDK, Go to its official website <https://docs.flutter.dev/get-started/install>, you will get the following screen.

The screenshot shows the 'Get started' section of the Flutter website. On the left is a sidebar with links like 'Get started', 'Set up Flutter', 'Learn Flutter', etc. The main content area has a heading 'Choose your development platform to get started'. It shows four options: 'Windows Current device' (selected), 'macOS', 'Linux', and 'ChromeOS'. Below this, there's a note about developing in China and a link to activate Windows.

The screenshot shows the 'Choose your first type of app' section. It has three options: 'Android Recommended' (selected), 'Web', and 'Desktop'. A red arrow points from the 'Android Recommended' box towards the 'Windows' icon in the previous screenshot. Below the options, there's a note about tooling configuration and a link to activate Windows.

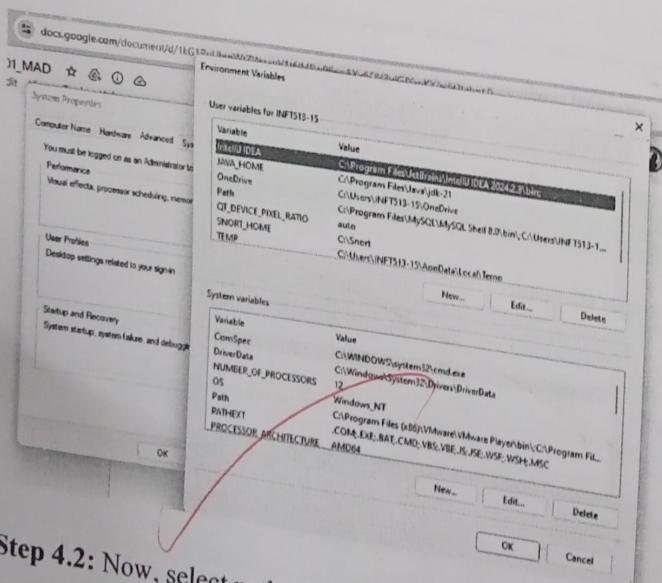
Step 2: Next, to download the latest Flutter SDK, click on the Windows icon. Here, you will find the download link for SDK.

Step 3: When your download is complete, extract the zip file and place it in the desired installation folder or location, for example, C:/Flutter.



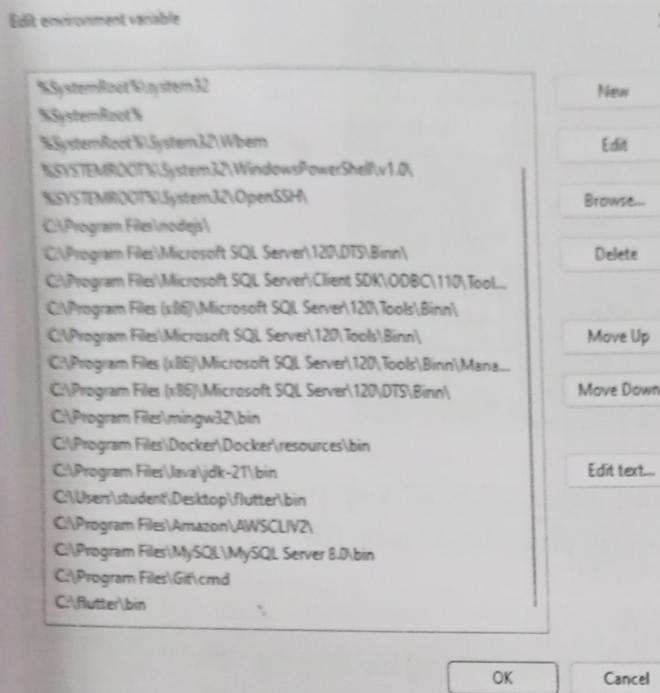
Step 4: To run the Flutter command in regular windows console, you need to update the system path to include the flutter bin directory. The following steps are required to do this:

Step 4.1: Go to MyComputer properties -> advanced tab -> environment variables. You will get the following screen.



Step 4.2: Now, select path -> click on edit. The following screen appears

Step 4.3: In the above window, click on New->write path of Flutter bin folder in variable value -> ok -> ok -> ok



Step 5: Now, run the \$ flutter command in command prompt.

Now, 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.

A screenshot of a Command Prompt window titled 'Command Prompt - Butter'. The window shows the following text:
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.
C:\Users\student>flutter --version

A new version of Flutter is available!
To update to the latest version, run "flutter upgrade".

Flutter 3.27.2 • channel stable • https://github.com/flutter/flutter.git
Framework • revision 68415ad1d9 (2 weeks ago) • 2025-01-13 10:22:03 -0800
Engine • revision e572b006cb
Tools • Dart 3.6.1 • DevTools 2.49.2

C:\Users\student>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.

Step 6: 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.

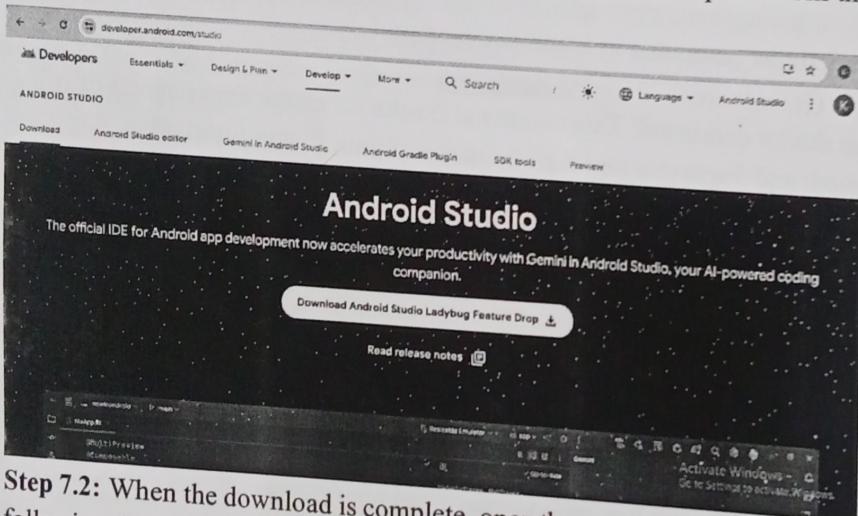
```
C:\Users\student>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 3.27.2, on Microsoft Windows [Version 10.0.22631.4751], locale en-IN)
[!] Windows Version (installed version of Windows is version 19 or higher)
[!] Android toolchain - develop for Android devices (Android SDK version 34.0.8)
  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
  X Visual Studio not installed; this is necessary to develop Windows apps.
    Download at https://visualstudio.microsoft.com/downloads/
    Please install the "Desktop development with C++" workload, including all of its default components
[!] Android Studio (version 2023.1)
[!] VS Code (version 1.96.4)
[!] Connected device (3 available)
[!] Network resources

! Doctor found issues in 2 categories.

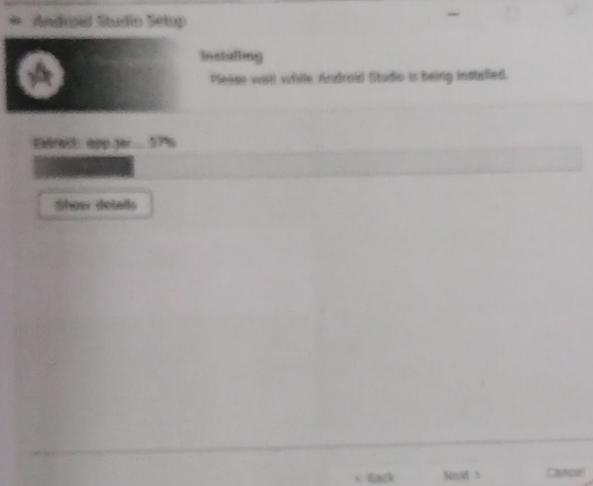
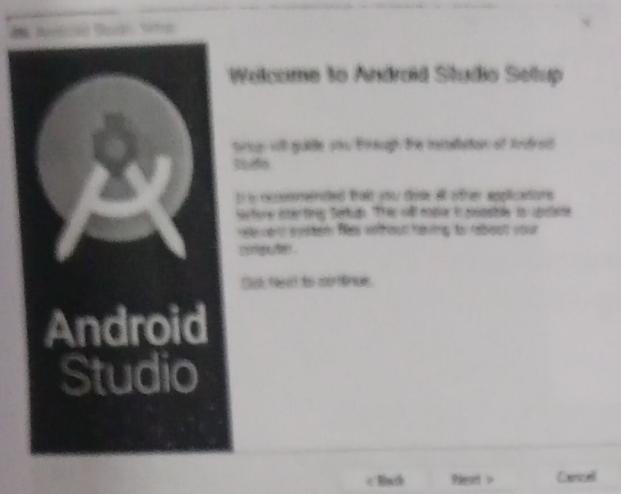
C:\Users\student>
```

Step 7: 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.

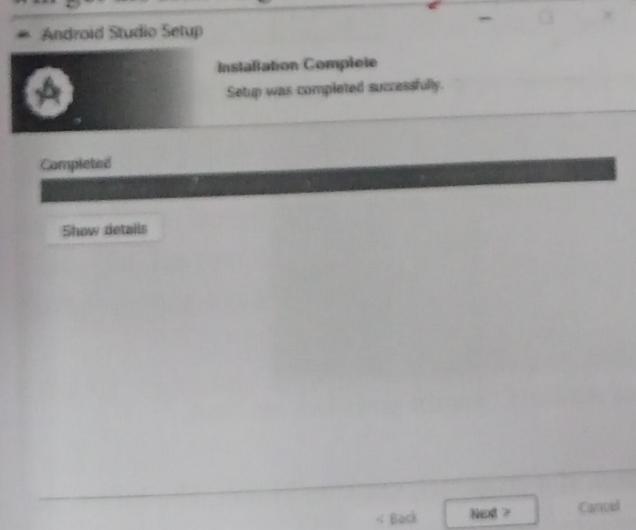
Step 7.1: Download the latest Android Studio executable or zip file from the official site.



Step 7.2: When the download is complete, open the .exe file and run it. You will get the following dialog box



Step 7.3: Follow the steps of the installation wizard. Once the installation wizard completes, you will get the following screen.





Vivekanand Education Society's

Institute of Technology

(An Autonomous Institute Affiliated to University of Mumbai)

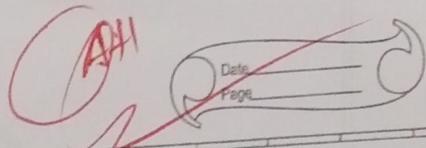
Department of Information Technology

A.Y. 24-25

Mobile App Development and Progressive Web App Lab

Experiment No.	02
Experiment Title.	To design Flutter UI by including common widgets.
Roll No.	08
Name	Abhishek Chaurasiya
Class	D15 B
Subject	MAD & PWA
Grade:	(A+) ✓

Advance DevOps Lab



Experiment 2

Designing Flutter UI with common widgets

Aim: To design a Flutter UI using common widgets like Container, Row, Column, Text, button and Image.

Steps:-

1) Create a flutter Project

→ flutter create my-app
cd my-app

2) Open the project in an Editor
use Android Studio or VS code

3. Modify main.dart to include common widgets

→ ~~? import "package:flutter/material.dart";~~

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

class MyApp extends StatelessWidget {

@override - Widget build(BuildContext context) {
 return MaterialApp(
 home: Scaffold(
 appBar: AppBar(title: Text("Flutter UI"))

1:22

24.0 KB/S

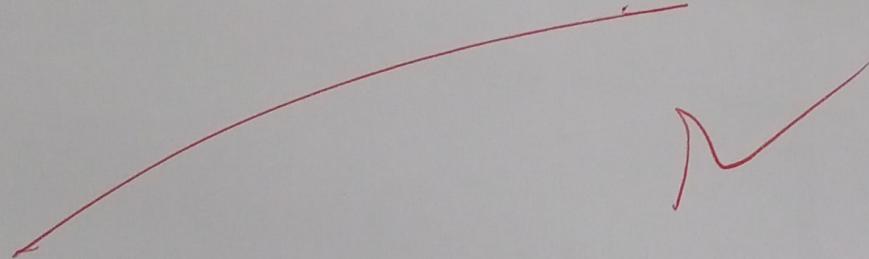
DEBUG

My App



Welcome!

Select File



Convert

Saved

Profile





Vivekanand Education Society's

Institute of Technology

(An Autonomous Institute Affiliated to University of Mumbai)

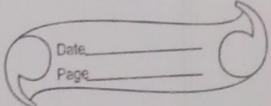
Department of Information Technology

A.Y. 24-25

Mobile App Development and Progressive Web App Lab

Experiment No.	03
Experiment Title.	To include icons, images, fonts in Flutter app
Roll No.	08
Name	Abhishek Chaurasiya
Class	D15 B
Subject	MAD & PWA
Grade:	(A) ✓

Abhishek R. Chourasia
DISB- 08



Experiment - 3

Including icons, images and fonts in flutter app.

Aim :- To include icons, images and custom fonts in a flutter app.

Steps

1) Create a flutter project

2) Add images Assets :-

- Place images in assets/images/
- Update pubspec.yaml

flutter:

assets:

- assets/images/sample.png

3) Include custom font

- place font file in assets/fonts/.

- Update pubspec.yaml

flutter:

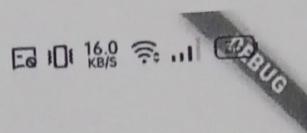
fonts

- family: CustomFont

fonts

- asset: assets/fonts/CustomFont.ttf

1:22 16.0 KB/S



My App



Abhishek Chaurasiya



Sign-in methods



Enable more sign-in methods

G

→ Sign out

Delete account

Convert

Saved

Profile





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.	08
Name	Abhishek Chaurasiya
Class	D15 B
Subject	MAD & PWA
Grade:	(A) ✓

Experiment - 4

Creating an interactive

Creating an interactive form using the
form widget in flutter

Aim: To create an interactive form using the form
widget in flutter including text fields, validation
and a submit button

~~Steps:~~

- Add form widget
 - use form and assign a ~~to~~ global key < GlobalKey</>()
 - Add Text formfield for user input with validation.
- Add Submit button
 - validation and display a success msg using scaffold messenger

Conclusion:-

A flutter form was successfully created using
form widget, allowing user input validation and
submit button

1:26

1:26

15.0 Kbps



Sign in

Welcome to FlutterFire, please sign in!

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

Email

Password

[Forgotten password?](#)

[Sign in](#)



[Sign in with Google](#)

By signing in, you agree to our terms and conditions.

Register

Welcome to Flutterfire, please sign up!

Already have an account? [Sign in](#)

Email

Password

[Confirm password](#)

[Register](#)



[Sign in with Google](#)

By signing in, you agree to our terms and conditions.



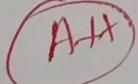
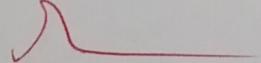
Vivekanand Education Society's Institute of Technology

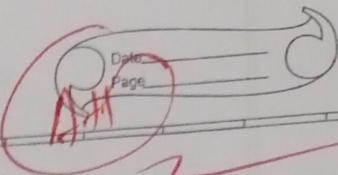
(An Autonomous Institute Affiliated to University of Mumbai)

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.	08
Name	Abhishek Chaurasiya
Class	D15 B
Subject	MAD & PWA
Grade:	 



Experiment - 5

Applying Navigation, Routing and Gestures in a flutter App.

Aim:-

To implement navigation, routing and gestures in a flutter application to enable smooth screen transitions.

Explanation:-

Navigation and routing in flutter help users move between different screens.

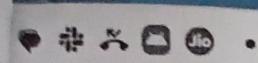
Gestures allow users to interact with the app through taps, swipes and other touch inputs.

i) Use the `GestureDetector` ~~widget~~ to detect user interaction ~~like tap, double tap, swipe, or long-press~~

Conclusion:-

Navigation and gestures enhance user experience in flutter apps by enabling smooth transitions and interactive elements.

Implementing them correctly ensures intuitive app navigation and responsiveness.



16.0 KB/S DEBUG

App



Welcome!

↑ Select File

Selected video: VID20250303130714.mp4



Open File

Convert

Convert

Saved

Profile



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.	08
Name	Abhishek Chaurasiya
Class	D15 B
Subject	MAD & PWA
Grade:	(A+) <i>[Handwritten grade circled]</i>

Experiment - 6

setting up firebase for android and ios
in a flutter App.

Aim:- To integrate firebase for android and ios to enable backend services like authentication database, and cloud storage.

Explanation:

firebase provided backend services that enhance app functionality. Setting up firebase in flutter involved adding firebase sdk.

Steps:-

- 1) Create Firebase project
- 2) Configure Firebase ~~account~~ for android
- 3) Add Firebase Dependencies in flutter.
- 4) Initialize firebase in main file.

Conclusion:-

Setting up firebase in flutter enables backend services, making the app more dynamic and interactive. Proper configuration ensured seamless integration for android.

console.firebaseio.google.com

Firebase Authentication

Project Overview Authentication

Users Sign-in method Templates Usage Settings Extensions

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/React Native support for web apps.

Search by email address, phone number, or user ID

Username	Passport	Create	Last updated	User ID
Unconfirmed@gmail.com		Mar 8, 2025	Mar 8, 2025	10:6995c9c-00ff-4e00-aed0...
Unconfirmed123@q1q1q...		Mar 8, 2025	Mar 8, 2025	1a500f1d-0000-4000-9400...

Page range: 1 - 1 of 2

Sign-in providers

Provider Status

- Email/Password Enabled
- Google Pending

Add new provider

Advanced

SMS Multi-factor Authentication

Allow your users to add an extra layer of security to their account. Once enabled, integrated and configured, users can sign in to their account in two steps: using SMS. LEARN MORE

★ MFA and other advanced features are available with Identity Platform, Google Cloud's complete customer identity solution built in partnership with Firebase. This upgrade is available on both the Spark and Blaze plans.

Upgrade to Blaze

Spark Blaze (Serverless) Upgrade

console.firebaseio.google.com

Firebase Authentication

Project Overview Authentication

Users Sign-in method Templates Usage Settings Extensions

Sign-in providers

Provider Status

- Email/Password Enabled
- Google Pending

Add new provider

Advanced

SMS Multi-factor Authentication

Allow your users to add an extra layer of security to their account. Once enabled, integrated and configured, users can sign in to their account in two steps: using SMS. LEARN MORE

★ MFA and other advanced features are available with Identity Platform, Google Cloud's complete customer identity solution built in partnership with Firebase. This upgrade is available on both the Spark and Blaze plans.

Upgrade to Blaze

Spark Blaze (Serverless) Upgrade



Vivekanand Education Society's Institute of Technology

(An Autonomous Institute Affiliated to University of Mumbai)

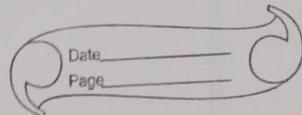
Department of Information Technology

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.	08
Name	Abhishek Chaurasiya
Class	D15 B
Subject	MAD & PWA
Grade:	(A) ✓

Name :- Abhishek R. Chaurasiya
Class:- DISB 108



MPL - 7

Aim!:- To write metadata of your e-commerce PWA in a web app manifest file to enable 'Add to home screen feature'.

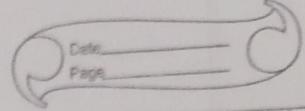
Theory:-

PWA :- Progressive Web App, It is defined as app which is a web app but capabilities similar to native apps.

This is great in scenario where the app data is not changed very frequently.

Conclusion:- By the end of this project session we implemented add to homescreen ability into a web app.

Name:- Abhishek R Chauhan
Class:- DISB 108



MPL - 8

Aim :- Register Service worker and activate

Theory:-

Service worker:-

A service worker is a script that runs in background of web browser, separate from main page and does not need the webpage to open to work.

Operations:-

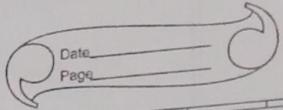
Network request

Offline experiences

Background tasks

Conclusion:- ~~By the end of this practical session we implemented service worker script and activated it.~~

Name: Abhishek R. Chauhan
Class: DISB108



MPL - 9

Aim: Implement service worker events.

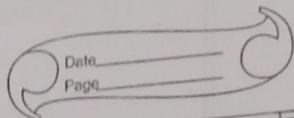
Theory ↗

Service worker events

- 1) fetch :- Intercept all network requests
- 2) push:- To send notification to client
- 3) install :- when SW is installed
- 4) activate :- when SW is activated
- 5) sync :- To sync the local cache
- 6) message :- Two way communication between SW and client app

Conclusion: we implemented all service worker events

Name:- Abhishek R. Chauhan
Class:- ATE DISB/08



MPL - 10

Aim :- Deploy PWA to ~~github~~ github pages

Theory :-

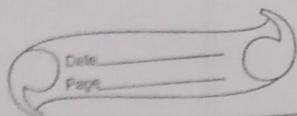
github pages:-

It is free hosting site by ~~github~~ github
It is used to deploy static pages
directly from the repository.

PWA apps ~~can~~ be stored in repository
and we can trigger deployment of same

Conclusion :- By the end of this practical
session we successfully deployed PWA app to
~~github~~ github pages.

Name:- Abhishek R. Chauhan
class:- DISR/08



MPL-11

Aim:- To use ~~the~~ google lighthouse PWA analysis tool to test pwa functionality.

Theory :-

~~the~~ Google Lighthouse ! -

This is a tool by ~~to~~ ~~got~~ google which is used to assess the performance of the web apps.

Conclusion :-

We tested the functionality of the pwa app - using google lighthouse.

And we ~~got~~ good results .

MPL Assignment - 1

Q1) Explain the key features & advantages of using flutter for mobile app development.

→ Key features:-

- 1) Single codebase - write one codebase for android & ios, reducing development effort & maintenance.
- 2) Hot reload - instantly see changes in the app without ~~restarting~~ restarting making development faster & more interactive.
- 3) Fast performance - uses the dart language & a compiled approach for smooth & high performance apps.
- 4) Open source & strong community, ensuring continuous improvement & resources.

Advantages:-

- 1) Faster Development time: Hot reload & single codebase reduce development time significantly.
 - 2) Cost effective: Since the code runs on both android & ios, business save on development & maintenance cost.
 - 3) Reduce performance issues.
- b) Discuss how the flutter framework differs from traditional approaches & why it has gained popularity.
- How flutter differs from traditional approaches.

- 1) UI rendering - Traditional apps use native components while flutter has its own rendering engine (Skia) for faster performance.
- 2) ~~Customization~~ Customization - traditional UI design depends on platform specific components, but flutter provides fully customizable widgets.

Why has flutter gained popularity?

- 1) Faster development with ~~hot~~ hot reloading.
- 2) ~~cross~~ cross platform efficiency
- 3) consistent UI across devices
 - a) improved performance

(Q2)

- a) Describe the concept of ~~the~~ concept of the widget tree in flutter. Explain how widget composition is used to build complex UI.

→ widget tree in flutter.

In flutter, the widget tree is the fundamental structure that represents the UI of an app. It is a hierarchical arrangement of widgets. Each widget defines a part of the user interface. flutter UI is entirely built using widgets which can be stateless or stateful. The widget tree determines ~~for~~ how the UI is rendered and updated when changes occur.

when changes occurs

widget composition in flutter

widget composition refers to building complex UI by combining smaller reusable widgets instead of creating large monolithic UI components, flutter encourages breaking the UI into smaller reusable widgets that can be reused & nested with each other

provide examples of commonly used widgets & their roles in creating a widget tree.

Structural widgets

base widgets act as the foundation building a UI

~~Material App~~: The root widget of a flutter app that provides essential configuration

Container - A versatile widget used for styling, padding, margin & background

E.g. MaterialApp (

home: Scaffold (

appbar: AppBar (title: Text ("Flutter widget tree"))

body: Container (

padding: EdgeInsets.all (10.0)

child: FText ("Hello Flutter!") :

),

),

);

2) Input & interaction widgets

Text field: Accepts text input from user.

Elevated Button - A button with elevation.

Gesture Detector - Detects gestures like tap, long press, swipe & long press.

e.g. Column (

children [

Text(hintDecoration: InputDecoration(

labelText: "Enter name")

Elevated Button (

onPressed: () {

print("Button Pressed"); } ,

child: Text ("submit"),

});

(Q3)

a) Discuss the important state management in flutter application.

→ In flutter, state refers to data that can change during the lifetime of an application. This includes:

- User input, UI changes, Network changes, Application states.

There are two types of state:-

1) Ephemeral state:

2) App wide states -

3) code maintainability & scalability

4) Data consistency & synchronization

Trade-off & constraint The different state management approaches available in flutter, such as ~~setstate~~ providers & Riverpod, provide scenarios where one approach is suitable.

~~SetState~~ - local state

pros - simple built-in easy to use

cons - Not scalable caused unnecessary re-render
best use cases: small UI updates.

~~Provider~~ - App wide state

pros - lightweight, recommended by flutter

~~Scenarios for each approach~~

Use ~~SetState~~ when simple UI element like
~~toggle~~ used in ~~UI~~ UI.

Use providers when sharing state across multiple
widgets such as managing user authentication or
theme changes

Use riverpod when building a complex, scalable app with
cost management

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

1) Go to firebase project console

2) Download the config Google-services.json

Step 2: Add firebase dependencies

pubspec.yaml:

dependencies:

firebase_core: ^latest - version

firebase_auth: ^latest - version

cloud_firestore: ^latest - version

Step 3: Initialize firebase

Step 4: Use firebase service

i) firebaseauth

ii)

Benefits of using firebase

1) Serverless Backend - No need

2) Realtime Data Sync

3) Scalable & Secure

4) Highlight the commonly used firebase services used in flutter development.

→ 1) Authentication

✓ 2) Data store

3) Realtime Storage

4) Cloud messaging

5) Firebase Analytics

6) Firebase Crash Analytics

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

periment	Assignment-2
le.	
ll No.	08
ame	Abhishek Chaurasiya
ass	D15 B
bject	MAD & PWA
rade:	(09) ✓

MPL Assignment -2

Jav./
Date: 03
Page:

Define PWA and explain its significance in modern web

Progressive Web Apps (PWAs) and their Significance.

A progressive web app is a type of web application that leverages modern web technologies to provide an app-like experience on the web. PWAs combine the best features of traditional web apps and native mobile applications, offering improved performance, offline capabilities.

Significance in modern web development:

- Cross-platform compatibility
- Offline functionality
- Improved performance.
- App-like experience
- SEO-friendly

Key characteristics of PWAs vs Traditional mobile apps

Feature	PWA	Traditional mobile
Installation	No app store required add to home screen	Installed via app store
Offline Support	Yes, using service workers	Yes, via local storage and background sync

Q2

Define responsive web design and explain its importance in the context of progressive Web apps. Compare and contrast responsive, fluid and adaptive design approaches.



Responsive web design is a design approach that ensures a website's layout adjusts dynamically to different screen sizes and resolutions.

Importance of responsive web design

- PWAs need to function seamlessly across multiple devices
- Ensures a consistent user experience regardless of device
- Enhances accessibility and usability

Comparison of responsive, fluid, and adaptive web design

Design Type	Description	Pros	Cons
Responsive	uses flexible grids and media queries to adjust layout	flexible, considerate, user experience	more complex
Fluid	uses percentage-based width for elements	smooth scalability	can break
Adaptive	uses predefined layouts optimized for specific screen sizes	for diff devices	more effort required

Describe The life cycle of service workers, including registration, install and activation phase

Service workers are background script that enable offline support, caching and background sync in PWAs. Their life cycle consists of three main phases

1) Registration:-

- The service worker is registered in browser using navigator.serviceWorker.register().
- This process occurs in the main Javascript thread of application.

2) Installation:-

- Runs when a new service worker is detected
- can cache assets for offline use
- If the installation fails, the service worker is discarded

3) Activation:-

- After ~~installation~~-installation, it moves to activation phase
- can clear old caches and manage background sync
- becomes fully functional after activation

Q4 Explain the use of indexedDB in the service workers
for data storage.



IndexedDB is a client-side NoSQL database used for storing structured data in browser. It allows PWAs to store large amount of data persistently.

How IndexedDB works with Service Workers

- Stores data locally, enabling offline access.
- Supports complex queries unlike localStorage.

Benefits of using indexedDB in PWAs

- Enables efficient offline storage.
- Improves app performance.
- Allows background syncing when the connection is re-established.

J.