

MAD & PWA Lab

Journal

Experiment No.	01
Experiment Title.	To install and configure the Flutter Environment
Roll No.	25
Name	SiddhARTH Jha
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO1: Understand cross platform mobile application development using Flutter framework
Grade:	(R) 8

MPL Lab 1

Aim: To install and configure flutter development environment and set up an Android emulator.

Theory: Flutter is an open-source UI toolkit by Google for building natively compiled applications for mobile, web and desktop from a single codebase. Before developing flutter apps, it's essential to properly install and complete flutter SDK and dependencies.

key components:

- 1) Flutter SDK: Core library and tools required to build and run
- 2) Android SDK: IDE commonly used for flutter development.
- 3) Android Studio / VS Code: Required to build and emulate Android apps.
- 4) Emulator: To test the application.

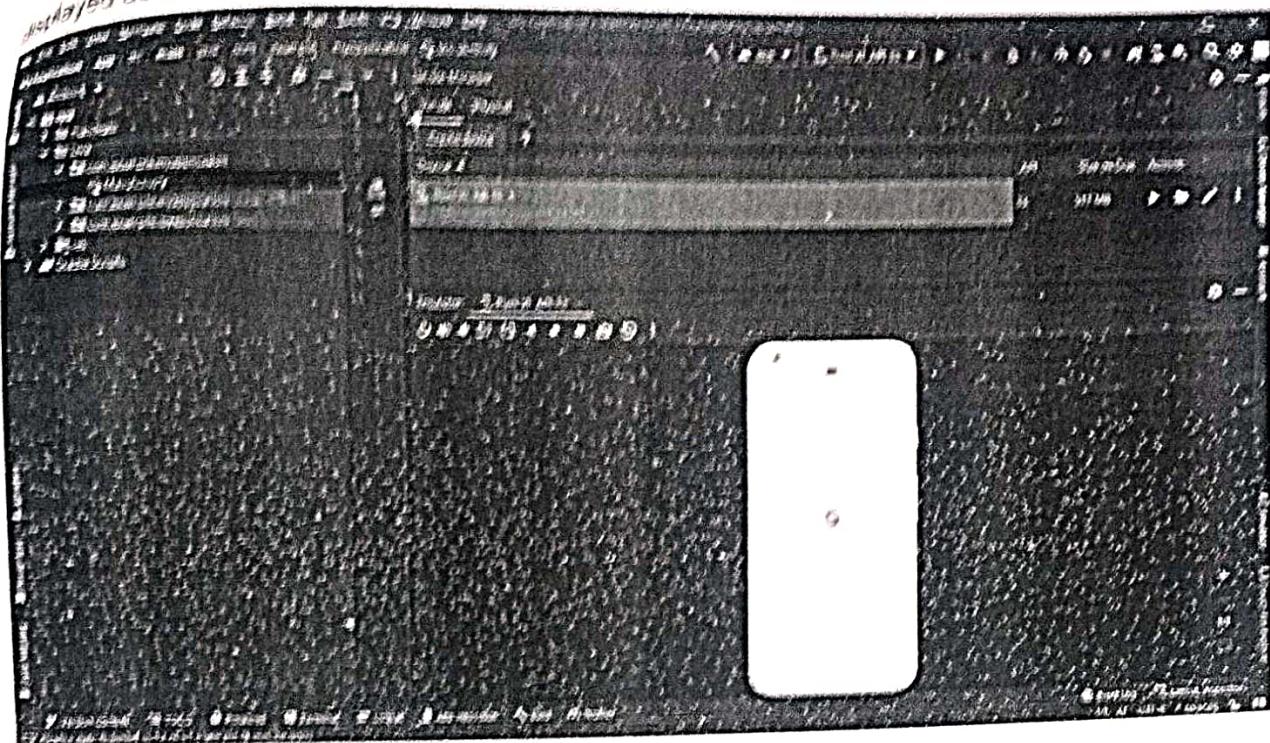
Steps:

- 1) Download and extract flutter SDK.
- 2) Add flutter to system environment variables (PATH)
- 3) Install Android Studio and during installation, select required SDKs and plugins:-
→ flutter, Dart, Android SDK.
- 4) open cmd : flutter doctor

Conclusion:

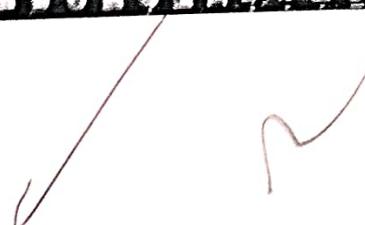
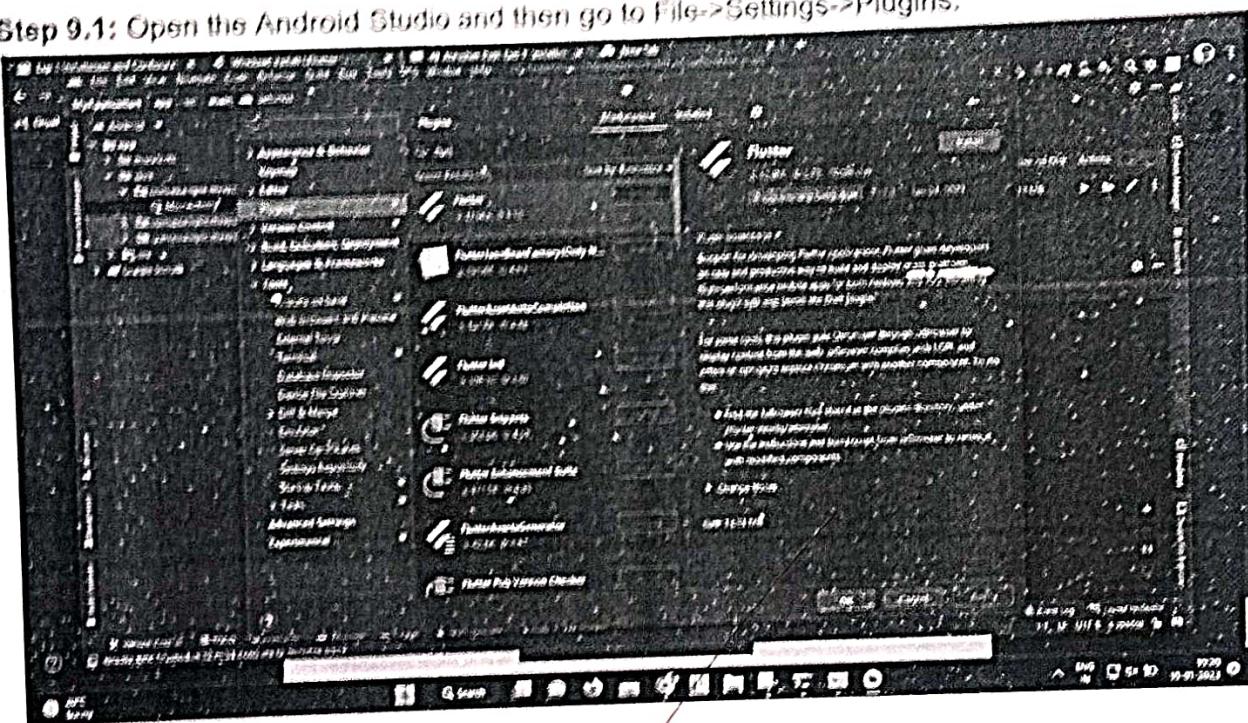
In this experiment, we successfully installed and configured flutter environment for Comfy, including IDE setup, SDK configuration and device setup.

displayed as below screen.



Step 9: 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 9.1: Open the Android Studio and then go to File->Settings->Plugins.



MAD & PWA Lab

Journal

Experiment No.	02
Experiment Title.	To design Flutter UI by including common widgets.
Roll No.	25
Name	Siddharth Tho
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	(A) ✓

MFIL Lab 2

Aim: To create a basic UI layout in flutter app using commonly used widgets like Text, Image, Column, Row, Container and Scaffold.

Theory: Flutter's UI is built entirely using widgets - everything from structure to styling is a widget - flutter provides a rich set of pre-built widgets that allows developers to create expressive UIs.

Key Widgets -

Scaffold: Provides basic layout structure with app bar, body, floating action button, etc.

AppBar: Displays app's title and toolbar actions.

Text: For displaying text.

Image: To include images from assets or network.

(Column) Row: Layout widgets used to arrange children.

Steps:

1. Create a new Flutter project.
2. Open lib/main.dart and structure the UI:
 - Scaffold as the root layout.
 - AppBar for top bar with app name
3. Add Text, Image and Container widgets.
4. Run the app on an emulator or device to test the layout.

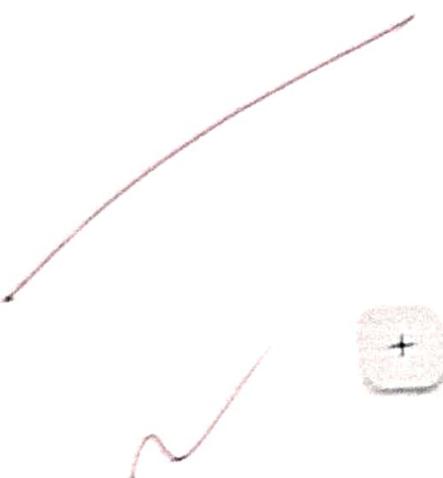
Conclusion:

In this experiment, we built a simple yet structured UI for Comfit using fundamental flutter widgets. This forms the basis for building more advanced and interactive screens in Comfit Application.

Campsite



No approved camps available.



MAD & PWA Lab

Journal

Experiment No.	03
Experiment Title.	To include icons, images, fonts in Flutter app
Roll No.	25
Name	SiddhARTH Jha
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	(B) ✓

MPL Lab 3

Fun: To enhance the visual appearance of the flutter app by adding custom icons, images and fonts.

Theory: Adding ~~to~~ media and typography to flutter enhances the app's design and user experience. Flutter allows integration of custom fonts, local images, and material.

Key concept:

Icons: flutter supports Material Icons out of the box via Icons class

Images: Images can be loaded from local assets.

Fonts: Custom font can be added to give the app a unique visual identity.

Steps:

- 1) Add Image and font files to appropriate folders.
- 2) Register assets in pubspec.yaml file.
- 3) Use Image.asset() for local images.

Steps :

- 1) Create folders like /assets/images and assets/fonts in flutter project.
- 2) Download and place required image files and font files in respective folders.
- 3) Update pubspec.yaml to declare:
 - assets
 - fonts

Conclusion :

This experiment demonstration how to integrate and display icons, images, and fonts within a flutter app. By ~~also~~ customizing media and text styles in Compit app.

← Add New Camp

Camp Name

test 11

Location

kurla

Description

mpl lab



Pick an Image (Optional)

Submit for Verification

✓

MAD & PWA Lab

Journal

Experiment No.	04
Experiment Title.	To create an interactive Form using form widget
Roll No.	25
Name	SiddhARTH Jha
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	(B) ✓

Steps:

- 1) Define a Global key `<FormState>` to track and validate the form.
- 2) Use the `Form` widget to wrap the form fields inside the UI.
- 3) Add `Textform field` widgets for input like name, email or phone.

Conclusion:

In this experiment, an ~~interactive~~ form was implemented in Compose flutter app; allowing user input with proper validation. The use of `Form` and `Textform field` widgets offers clean structure and responsiveness which are vital for data collection modules in any mobile app.

✓

Chrome

localhost:52757

Create an Account

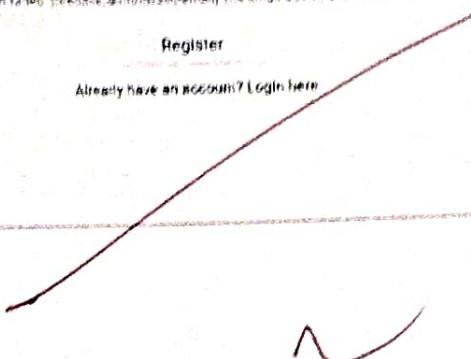
Email: bob

Password: ***

Registration failed. Please activate your account by clicking the link in the email address you have entered.

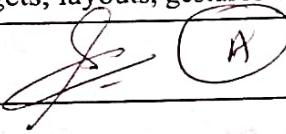
[Register](#)

Already have an account? [Login here](#)



MAD & PWA Lab

Journal

Experiment No.	05
Experiment Title.	To apply navigation, routing and gestures in Flutter App
Roll No.	25
Name	SiddhARTH Jha
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	 A

Steps:

- 1) Create multiple screens for navigation.
- 2) Configure routing in MaterialApp widget with named routes or use direct navigation.
- 3) Use Navigator.push() to move to another screen.
- 4) Add interactive buttons or icons to trigger navigation.

Conclusion:

This experiment successfully demonstrated Screen navigation and user gesture handling within the ~~Capstone~~ Flutter app.

Through proper routing and gesture recognition.

✓

Admin Panel



Pending Requests

Rejected Camps

No camps in this section.

MAD & PWA Lab

Journal

Experiment No.	06
Experiment Title.	To Connect Flutter UI with fireBase database
Roll No.	25
Name	Sidharth Jha
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO3: Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS
Gender	 

- 3) Download google-service.json file and place it in android/app/directory.
- 4) Update build.gradle
- 5) Initialize using main.dart

Conclusion:

By integrating firebase with Complete flutter app, we unlocked essential backend capabilities. This setup enables the app to scale further with secure authentication, real-time handling and user analytics.

The screenshot shows the Firebase Authentication console for a project named "campst". The left sidebar lists various Firebase services: Project Overview, Authentication (selected), Firestore Database, Cloud Functions, App Distribution, Google, Vertex AI, Product Catalogs, Build, Run, Analytics, AI, All products, Spark, and No-code (Scenarios). The main area is titled "Authentication" and contains tabs for Users, Sign-in method, Templates, Usage, Security, and Extensions. A message at the top states: "The following Authentication features will stop working when Precise Dynamic Links ends down on August 25, 2022: email link authentication for mobile apps, as well as Cookies OAuth support for web apps." Below this is a table of users:

User	Provider	Created	Last Sign-in	Provider ID
bob@gmail.com	Email	Apr 13, 2022	Apr 13, 2022	bob@gmail.com
alice@gmail.com	Email	Apr 22, 2022	Apr 22, 2022	alice@gmail.com
edgar@gmail.com	Email	Apr 22, 2022	Apr 22, 2022	edgar@gmail.com
sophia@gmail.com	Email	Apr 22, 2022	Apr 22, 2022	sophia@gmail.com
matthew@gmail.com	Email	Mar 24, 2022	Mar 24, 2022	matthew@gmail.com

The screenshot shows the Cloud Firestore console for the same project "campst". The left sidebar includes Project Overview, Authentication, Firestore Database (selected), Cloud Functions, App Distribution, Google, Vertex AI, Product Catalogs, Build, Run, Analytics, AI, All products, Spark, and No-code (Scenarios). The main area is titled "Cloud Firestore" and contains tabs for Data, Rules, Indexes, Disaster Recovery, Usage, and Extensions. It features a banner about Firestore Enterprise Edition with MongoDB compatibility. The Data tab displays a document structure under the "campst" collection:

```
campst > camp > 603ec20d9c1f...
```

Document ID: 603ec20d9c1f...

Fields:

- (default)
- + Start collection
- + Add document
- camp > + Start collection
- + Add field
- 603ec20d9c1f... > createdAt: April 2, 2022 at 9:32:23 AM UTC+200
- description: "Hello"
- locations: "Hello"
- name: "Hello"
- submittedBy: "sophia0512@gmail.com"
- verified: true

N

MAD & PWA Lab

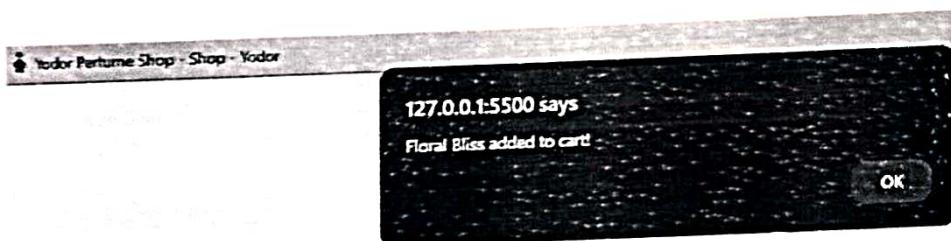
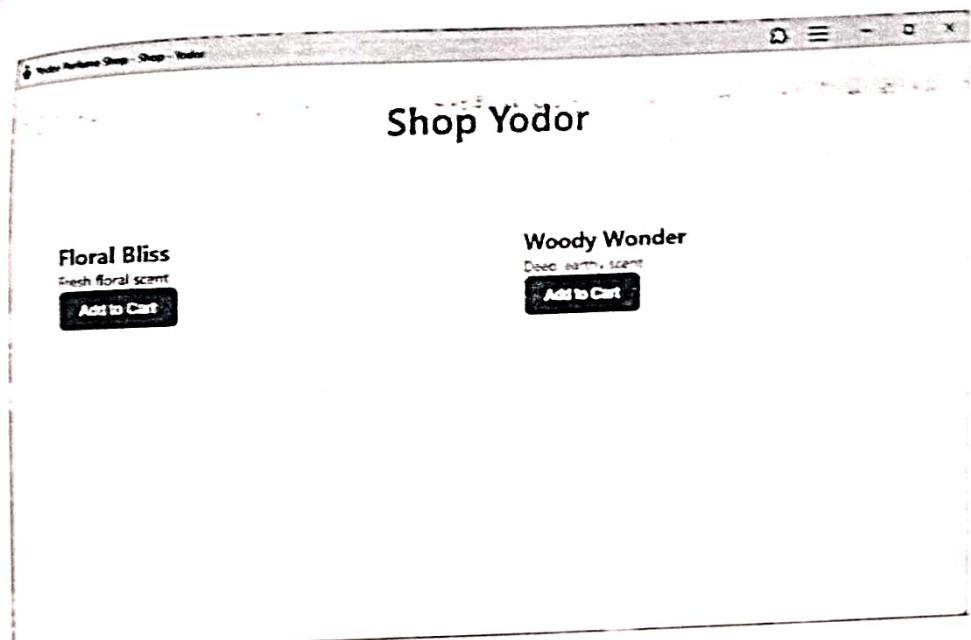
Journal

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.	25
Name	Siddhesh Tha
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO4: Understand various PWA frameworks and their requirements
Grade:	<i>80% (A)</i>

Conclusion:

The Add to Home screen feature recently
transforms your app to installable form,
enhancing user engagement.

This step minimizes the experience of using
native apps without needing app stores.



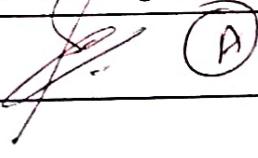
Floral Bliss
Fresh floral scent
Add to Cart

Woody Wonder
Deep, earthy scent
Add to Cart

✓ ✓

MAD & PWA Lab

Journal

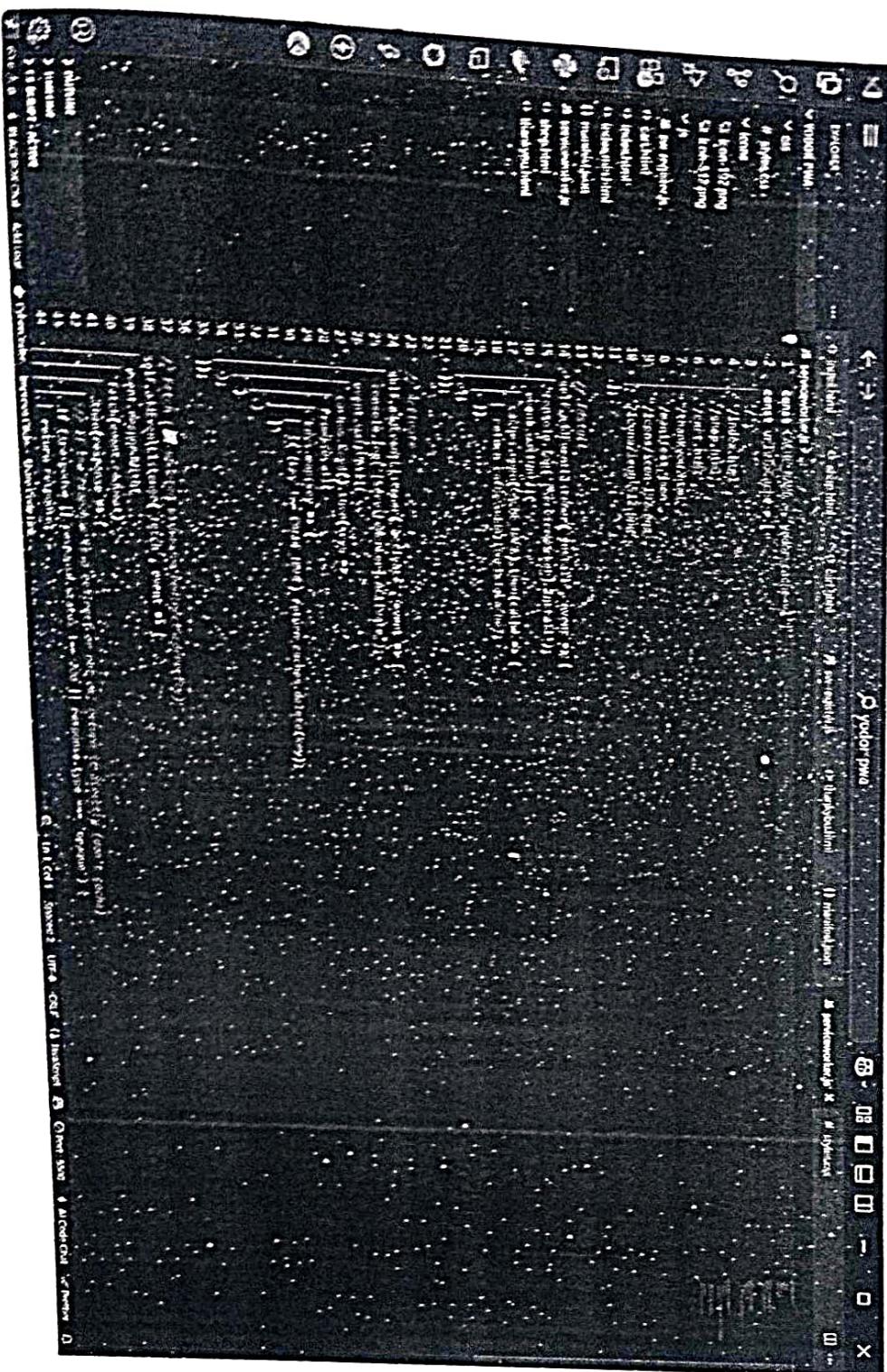
Experiment No.	08
Experiment Title.	To code and register a service worker, and complete the install and activation process for a new service worker for the E-commerce PWA
Roll No.	25
Name	Siddharth Jha
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	 (A)

Steps:

- 1) Place service-worker.js file in root directory
- 2) Inside index.html, register Service worker using JS.
- 3) In service worker file, handle install event to cache necessary app shell files.
- 4) Implement activate event to clean up old caches and ensure latest version.

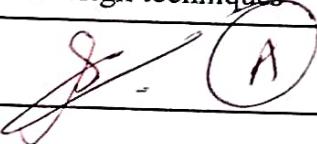
Conclusion:

The service worker was successfully registered, installed and activated in Yodot PDF. This lays the foundation for enhanced offline capabilities, faster load times, etc.



MAD & PWA Lab

Journal

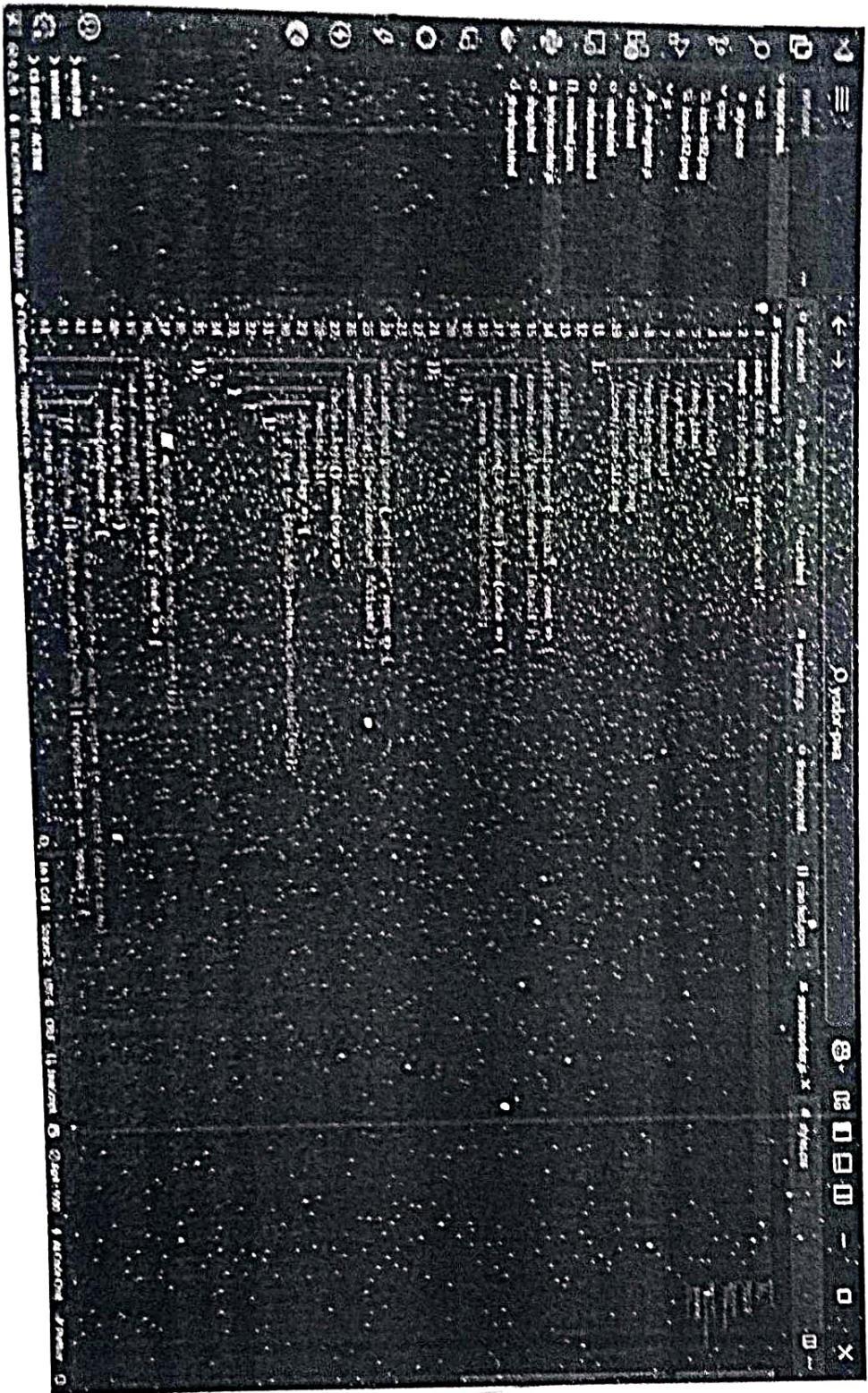
Experiment No.	09
Experiment Title.	To implement Service worker events like fetch, sync and push for E-commerce PWA
Roll No.	25
Name	Siddharth Jha
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

Conclusion:

The fetch, sync and push event were successfully implemented in Yodor's PWA service worker. This enabled reliable sync, passing with modern capabilities.

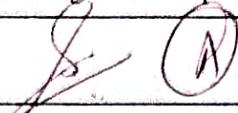
3

0



✓

MAD & PWA Lab Journal

Experiment No.	10
Experiment Title.	To study and implement deployment of Ecommerce PWA to GitHub Pages.
Roll No.	25
Name	Siddharth Tha
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

- 2) Ensure all paths in HTML are relative
- 3) Commit and push your changes.
- 4) Go to app's settings > Pages, choose the branch and /root
- 5) Save changes and GitHub will publish your PWA at `https://username.github.io/app-name/`.

Conclusion:

The Yodor PWA was successfully deployed to GitHub Pages, providing a live, installable, and offline-capable web experience.

This process shows how simple and effective GitHub pages are for hosting PWAs.

N

Pages

Yodor - Perfume PWA

x +

v - o x

Shop Now

< > C

dist1githubidod/pwaw/

≡

Yodor

Discover Signature Scents

Curated Perfume Collections, Just for You

Shop Now

MAD & PWA Lab

Journal

Experiment No.	11
Experiment Title.	To use google Lighthouse PWA Analysis Tool to test the PWA functioning.
Roll No.	25
Name	Siddharth Jha
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO6: Develop and Analyze PWA Features and deploy it over app hosting solution
Grade:	(B) <i>N</i>

Steps:

- 1) Open deployed yodoy PWA in Google Chrome.
- 2) Right click anywhere and select Inspect
→ Lighthouse tab
- 3) Set device mode (mobile) and categories (Performance and PWA)
- 4) Click Analyze Page Load or Generate Report.

Conclusion:

Lighthouse helped validate that the yodoy PWA is installable, works offline and follows modern web practices.

It provided actionable insights to improve performance and user experience.



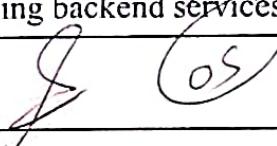
Discover Signature Scents

Curated Perfume Collections, Just for You

Shop Now

MAD & PWA Lab

Journal

Experiment No.	Assignment-1
Assignment 1 Questions	<p>1. a) Explain the key features and advantages of using Flutter for mobile app development. b) Discuss how the Flutter framework differs from traditional approaches and why it has gained popularity in the developer community.</p> <p>2. a) Describe the concept of the widget tree in Flutter. Explain how widget composition is used to build complex user interfaces. b) Provide examples of commonly used widgets and their roles in creating a widget tree.</p> <p>3. a) Discuss the importance of state management in Flutter applications. b) 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. a) Firebase Integration in Flutter: Explain the process of integrating Firebase with a Flutter application. b) 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.	25
Name	Siddhesh Jha
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO1: Understand cross platform mobile application development using Flutter framework LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation LO3: Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS
Grade:	

(A1) (92)

Date _____
Page _____

A.Y.: 2024-25

MOBILE PROGRAMMING LANGUAGE

ASSIGNMENT NO. 1

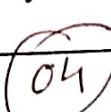
NAME : SIDDHARTH JHA

DIVISION : DISB

ROLL NO : 25

MAD & PWA Lab

Journal

Experiment No.	Assignment-2
Assignment 2 Questions	<ol style="list-style-type: none">1. Define Progressive Web App (PWA) and explain its significance in modern web development. Discuss the key characteristics that differentiate PWAs from traditional mobile apps2. 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.3. Describe the lifecycle of Service Workers, including registration, installation, and activation phases.4. Explain the use of IndexedDB in the Service Worker for data storage.
Roll No.	25
Name	Siddharth Jha
Class	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:	 

(04/05)

A.Y.: 2024-25

MOBILE PROGRAMMING

LANGUAGE

ASSIGNMENT NO. [2]

NAME : SIDDHARTH JHA

DIVISION : D15B

ROLL NO : [25]