

MAD & PWA Lab

Journal

Experiment No.	01
Experiment Title.	To install and configure the Flutter Environment
Roll No.	5
Name	Deepa Behrani
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO1: Understand cross platform mobile application development using Flutter framework
Grade:	A

MAD Experiment No. :- 1

Jo.

→ Aim & Problem Statement :- Learning programming languages requires consistent practice, especially with syntax, concepts and code snippets. Traditional flashcard apps are too generic and don't cater to unique needs of coding enthusiasts such as syntax highlighting, code formatting and tracking coding specific progress. There is a need for a specialized flashcard app that helps programmers retain knowledge efficiently.

→ EXISTING APPS :-

- 1] ANKI :- Used for medical studies, history and exam preparations [MCAT, GRE]
- 2] QUIZLET :- Vocabulary building, language learning, science, history and school subjects
- 3] BRAINSCAPE :- Language learning, business studies, geography and standardized test preparations [SAT, ACT]

→ FEATURES :-

- 1) Category wise flashcards :- Organized cards for multiple languages
- 2) Simple UI :- User-friendly interface ensuring ease of navigation
- 3) Color coded tags :- Tag cards based on difficulty levels
- 4) Daily Reminders :- Push notifications to review flashcards
- 5) Performance graph :- Simple visual representation of progress
- 6) Dark/Light mode :- Provide a comfortable reading experience in different lighting conditions.
- 7) Time tracker :- Track time spent on each flashcards to identify weak areas.

→ FLAWS :-

- 1) Lack of code formatting :- Apps don't support syntax highlighting for code snippets.
- 2) Generic design :- Not tailored to programmers needs.

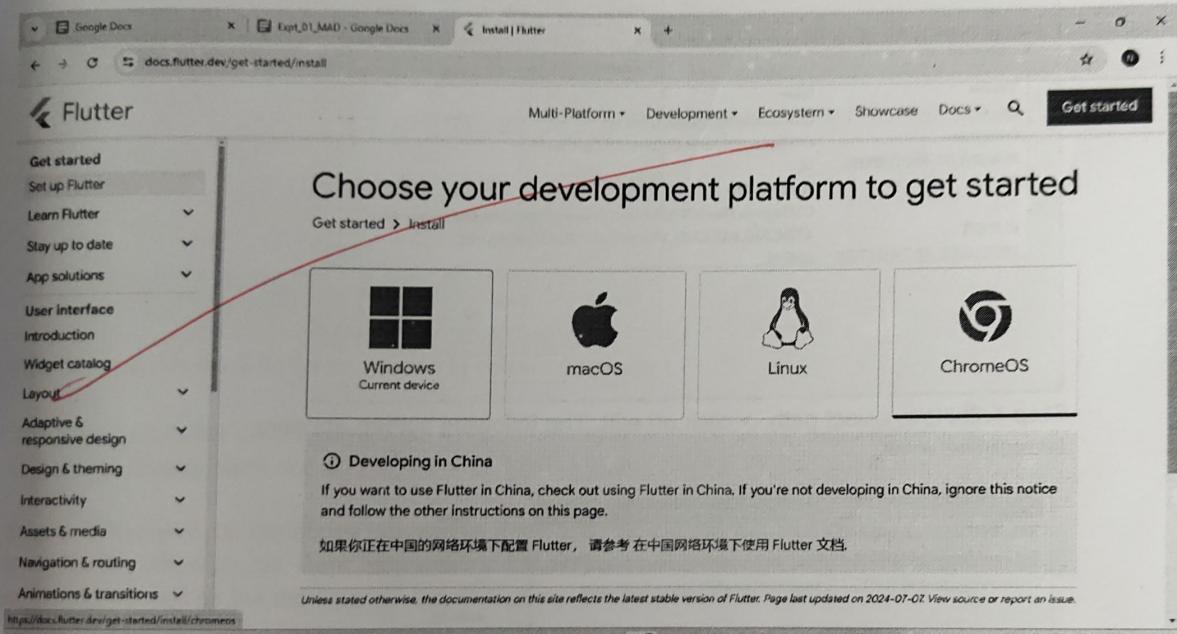
→ **CONCLUSION** :- Proposed ~~flashcard~~ app for coding language bridges the gap left by traditional flashcard apps. Its aim is to support programmers with code-friendly features, UI & performance tracking making it an efficient tool for coding enthusiasts. With its easy to implement yet powerful features this app will standout in Play store, offering personalized and effective learning experience for developers.

(A)

EXP 1: Installation and Configuration of Flutter Environment.

Install the Flutter SDK

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.

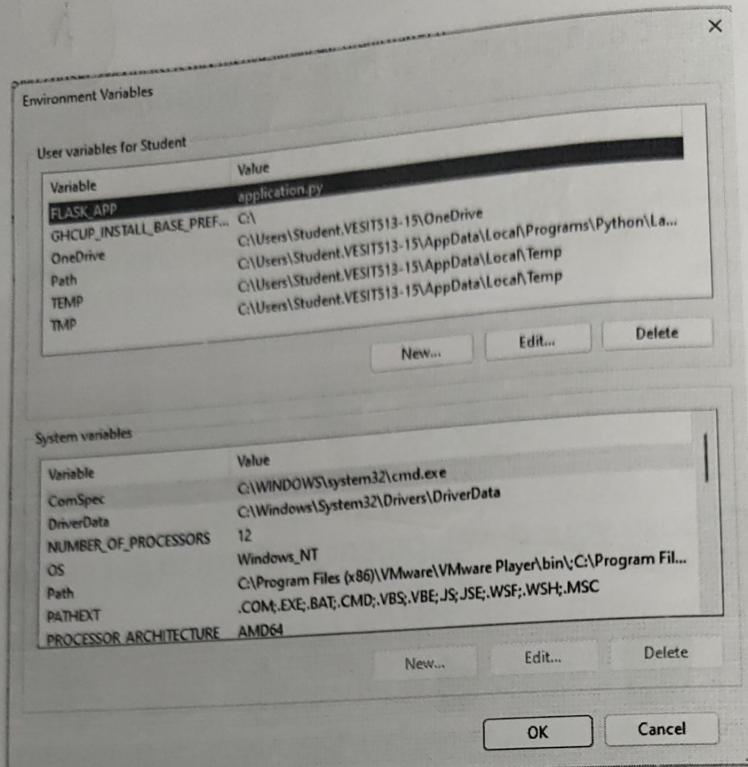


Step 2: Next, to download the latest Flutter SDK, click on the Windows icon. Here, you will find the download link for SDK.

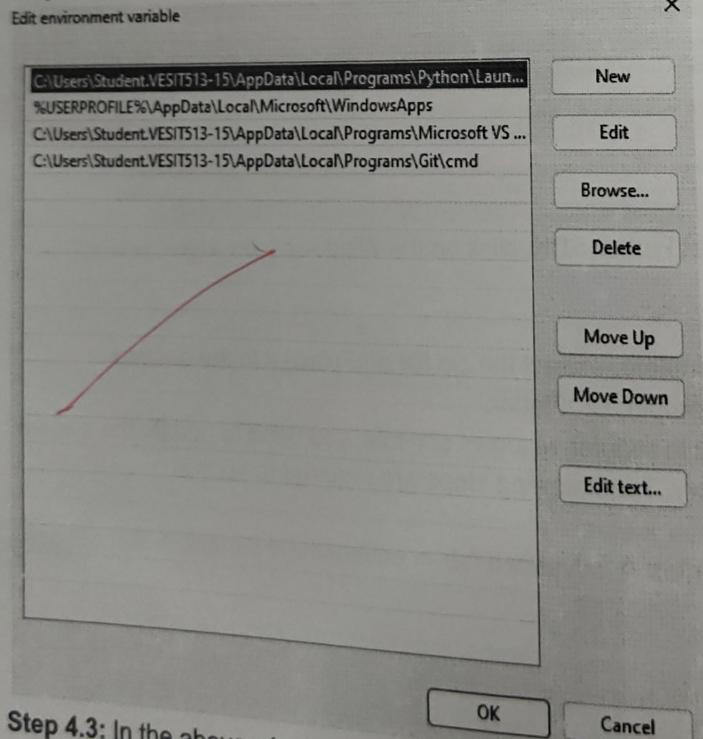
Step 3: When your download is complete, extract the zip file and place it in the desired installation folder or location, for example, C:/Flutter.

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.

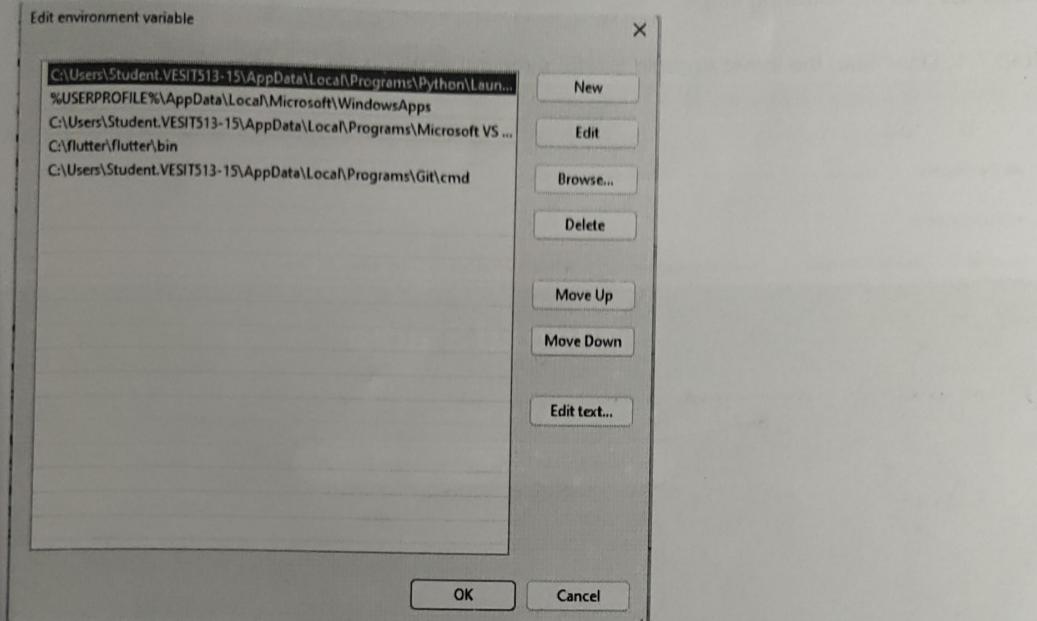


Step 4.2: Now, select path -> click on edit. The following screen appears



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.

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.

```
Command Prompt - flutter - flutter doctor
Run `path/to/sdkmanager --install "cmdline-tools;latest"`
See https://developer.android.com/studio/command-line for more details.
Android license status unknown.
Run `flutter doctor --android-licenses` to accept the SDK licenses.
See https://flutter.dev/docs/get-started/install/windows#android-setup for more details.
Chrome - develop for the web
Visual Studio - develop Windows apps (Visual Studio Community 2022 17.3.1)
Visual Studio is missing necessary components. Please re-run the Visual Studio installer for the "Desktop development with C++" workload, and include these components:
    MSVC v142 - VS 2019 C++ x64/x86 build tools
        - If there are multiple 'build tool' versions available, install the latest
        C++ CMake tools for Windows
Windows 10 SDK
Android Studio (version 2023.1)
VS Code, 64-bit edition (version 1.85.1)
Connected device (3 available)
Network resources

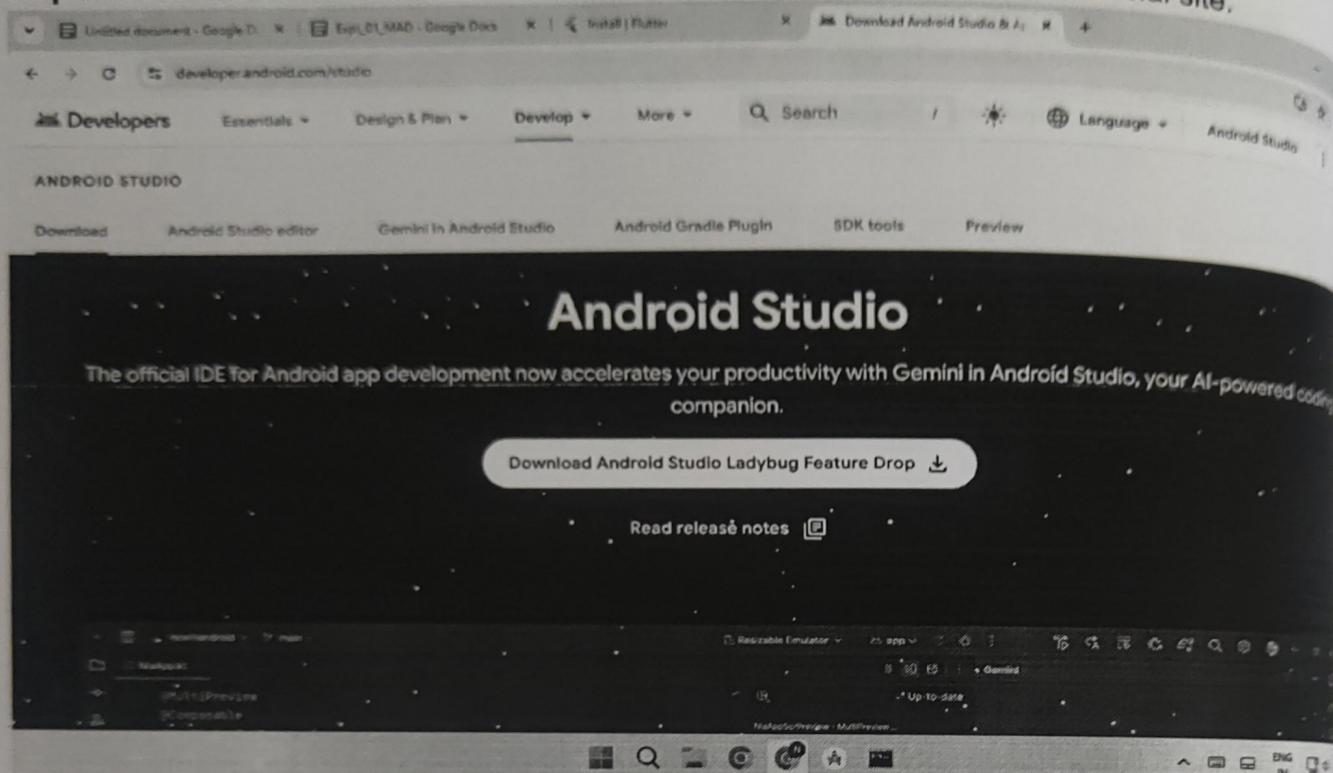
Doctor found issues in 2 categories.

C:\Users\Student.VESITS13-15>
```

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.

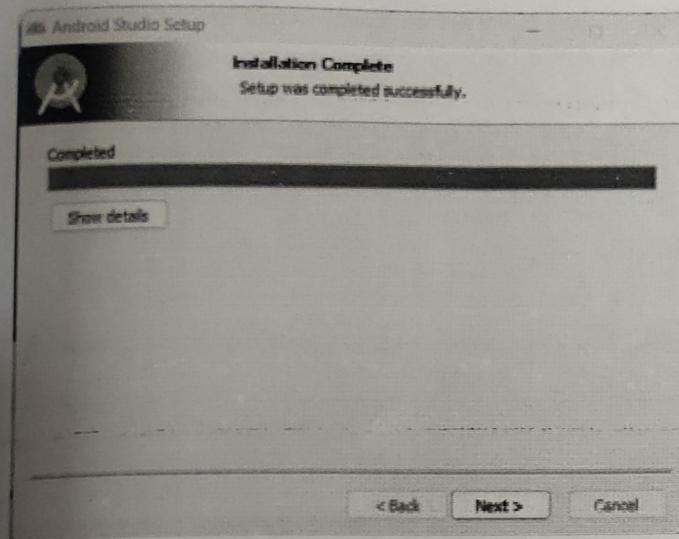
Step 7.1: Download the latest Android Studio executable or zip file from the official site.



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.



Step 7.4: 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 7.5: run the \$ flutter doctor command and Run flutter doctor --android-licenses command.

```
C:\Users\Admin>flutter doctor --android-licenses
Command exited with code 128: git -c log.showSignature=false log @{{upstream}} -n 1 --pretty=format:%ad --date=iso
Standard out:
Standard error: Fatal: no upstream configured for branch 'stable'

Warning: Errors during XML parse:
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.ry...
[=====] 100% Computing updates...
All SDK package licenses accepted.
```

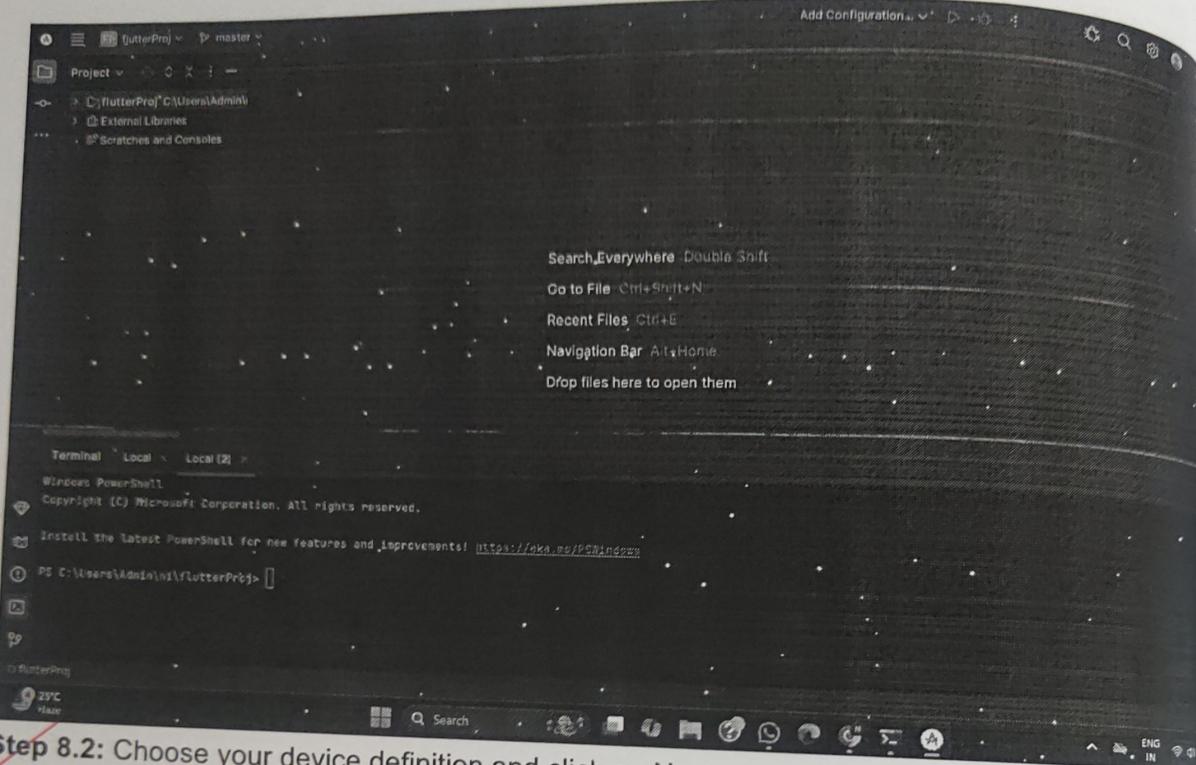
```
C:\Users\Admin>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 3.27.2, on Microsoft Windows [Version 10.0.22631.4751], locale en-IN)
[!] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices (Android SDK version 35.0.1)
[!] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Build Tools 2019 16.11.43)
[!] Android Studio (version 2024.2)
[!] VS Code (version 1.96.4)
[!] Connected device (3 available)
[!] Network resources

* No issues found!
```

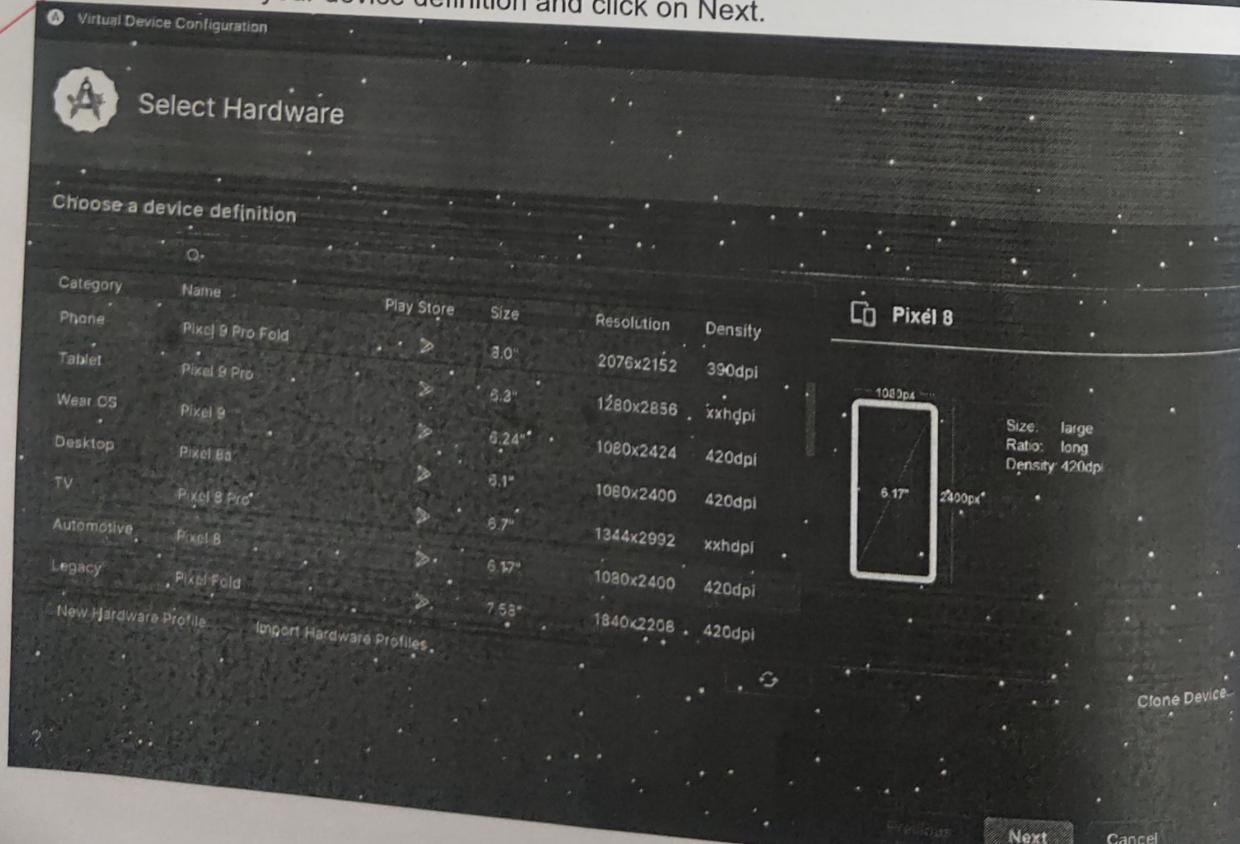
Step 8: Next, you need to set up an Android emulator. It is responsible for running and testing the Flutter application.

Step 8.1: 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.

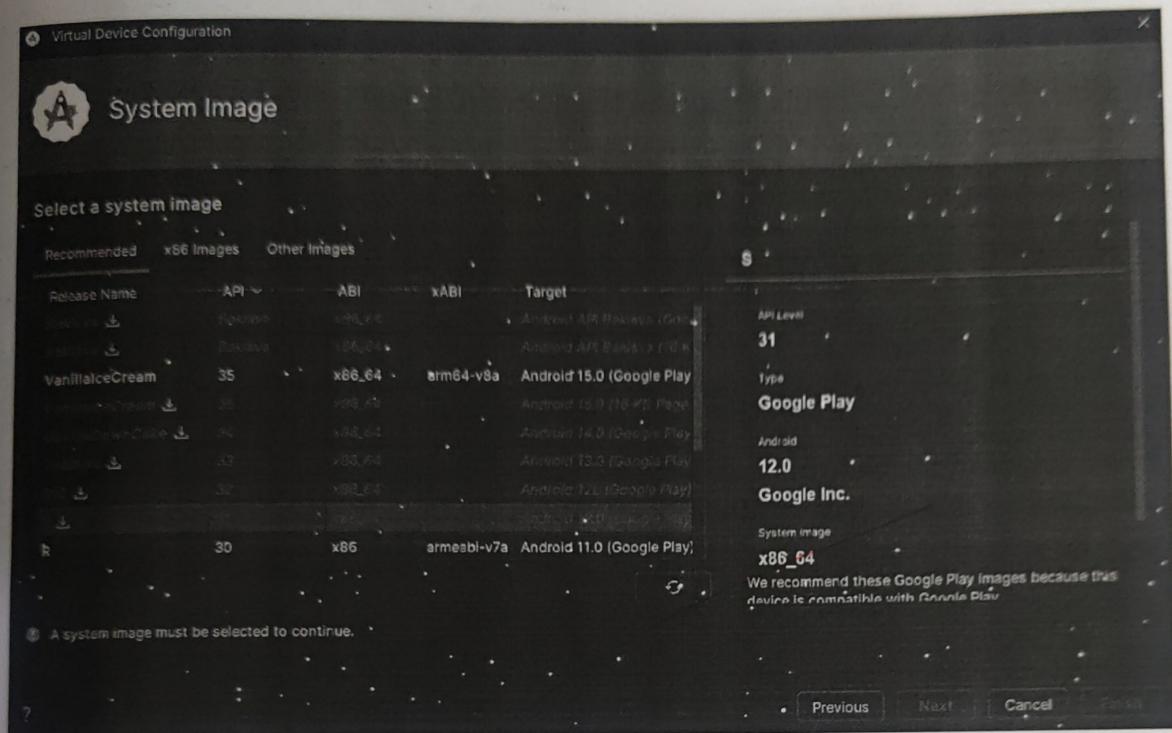


Step 8.2: Choose your device definition and click on Next.



Step 8.3: Select the system image for the latest Android version and click on Next.

Step 8.4: Now, verify the all AVD configuration. If it is correct, click on Finish. The following screen appears.

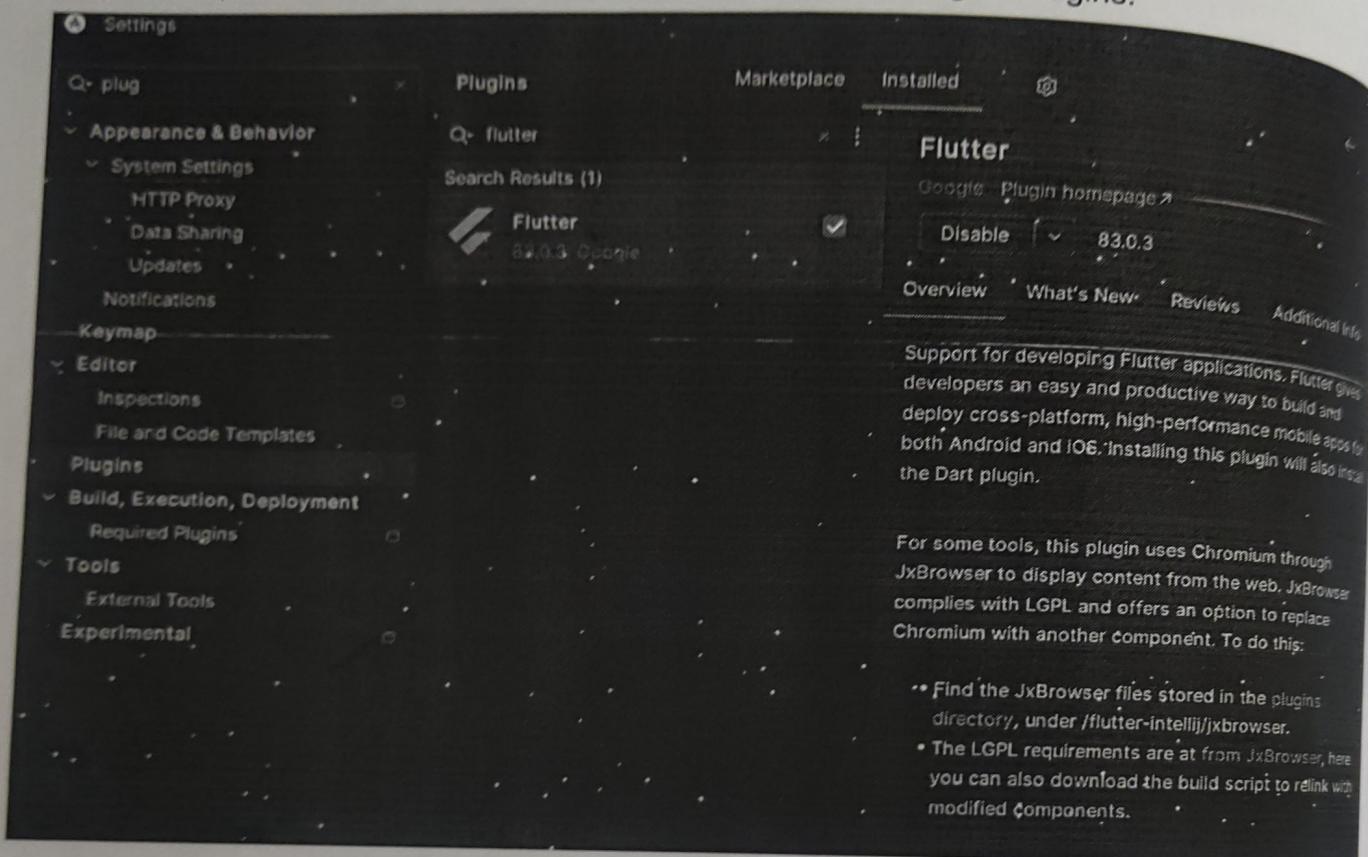


Step 8.5: Last, click on the icon pointed into the red color rectangle. The Android emulator displayed as below screen.



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.



Step 9.2: 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 9.3: Restart the Android Studio.

Conclusion: In this experiment, we successfully installed and configured the Flutter environment on a Windows system. We downloaded the Flutter SDK, set up the system path, and verified the installation using the flutter doctor command. Additionally, we installed Android Studio, set up the Android SDK, configured an emulator for testing, and integrated Flutter and Dart plugins into Android Studio. The experiment provided a comprehensive understanding of setting up a Flutter development environment, ensuring that all necessary tools and dependencies are correctly configured for seamless application development.

MAD & PWA Lab Journal

Experiment No.	02
Experiment Title.	To design Flutter UI by including common widgets.
Roll No.	5
Name	Deepa Behrani
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	A++

Experiment No : 02.

80 (A)

- + Aim : To design flutter UI by including common widgets.
- + Theory : Flutter provides a rich set of pre-built widgets that helps developers design beautiful and responsive UI.

Widgets are categorized into 2 types:

- i) Stateful
- ii) Stateless

Stateful Widget : Do not change dynamically
(ex: Text, Icon)

Stateless Widget : Maintain state and update UI dynamically
(ex: Textfield, checkbox)

Flutter follows a ~~widgets~~ based composition approach, where UIs are built by combining smaller widgets into a widget tree.

Common Widget Layout includes :

- i) Structural Widget : Scaffold, container, column, Row
- ii) Interactive Widget : Elevated Button, Textfield, switch
- iii) Styling widget : Padding, ^{Align}, sized Box

With the help of material design & cupertino widgets, flutter ensures a native like experience on both android and iOS.

Conclusion :- flutter simplifies UI's design with its reusable & customizable widgets, allowing developers to build beautiful interfaces efficiently. The Hot Reload feature accelerates the development process by instantly reflecting UI changes. By using flutter's widget-based approach, developers can build complex, scalable & responsive applications with minimal effort.

MAD LAB EXPERIMENT NO. : 02

NAME : DEEPA BEHRANI

CLASS : D15B

ROLL_NO: 05

Exploring Flutter Widgets

CODE :

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

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

class FlashcardApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flashcards',
      home: FlashcardScreen(),
    );
  }
}

class FlashcardScreen extends StatefulWidget {
  @override
  _FlashcardScreenState createState() => _FlashcardScreenState();
}

class _FlashcardScreenState extends State<FlashcardScreen> {
  int currentIndex = 0;
  bool showAnswer = false;
  final List<Map<String, String>> flashcards = [
    {'question': 'What is a variable in programming?', 'answer': 'A storage location with a name that holds data.' },
    {'question': 'What does the "const" keyword do in Dart?', 'answer': 'Declares a compile-time constant.' },
    {'question': 'What is a function in programming?', 'answer': 'A reusable block of code that performs a task.' }
  ];
}
```

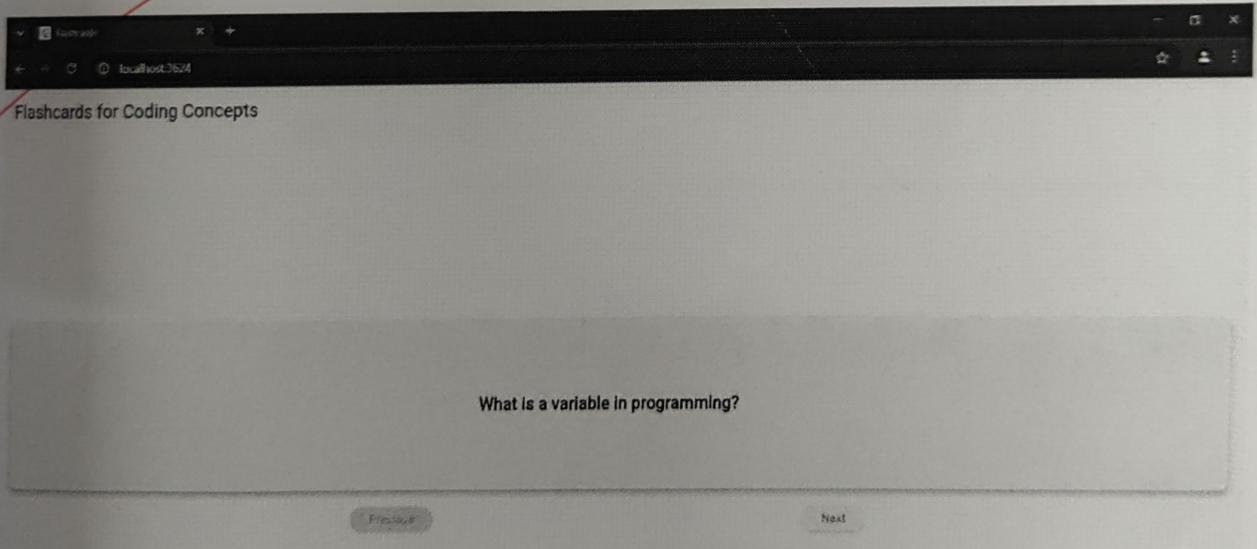
```
void nextFlashcard() {
    if (currentIndex < flashcards.length - 1) {
        setState(() {
            currentIndex++;
            showAnswer = false; });
    }
}

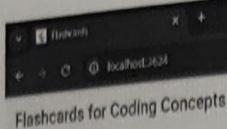
void previousFlashcard() {
    if (currentIndex > 0) {
        setState(() {
            currentIndex--;
            showAnswer = false; });
    }
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(title: Text('Flashcards for Coding Concepts')),
        body: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
                GestureDetector(
                    onTap: () {
                        setState(() {
                            showAnswer = !showAnswer; });
                    },
                    child: Card(
                        elevation: 5,
                        shape: RoundedRectangleBorder(
                            borderRadius: BorderRadius.circular(10)),
                        margin: EdgeInsets.all(20),
                        child: Container(
                            padding: EdgeInsets.all(20),
                            height: 200,
                            alignment: Alignment.center,
                            child: Text(

```

```
showAnswer  
    ? flashcards[currentIndex]['answer']!  
    : flashcards[currentIndex]['question']!,  
    textAlign: TextAlign.center,  
    style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold), ), ), ),  
  
Row(  
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
    children: [  
        ElevatedButton(  
            onPressed: currentIndex > 0 ? previousFlashcard : null,  
            child: Text('Previous'), ),  
        ElevatedButton(  
            onPressed:  
                currentIndex < flashcards.length - 1 ? nextFlashcard : null,  
            child: Text('Next'),  
        ), ], ); } }  
  
OUTPUT:
```





What is a function in programming?

Previous

Next

Flashcards for Coding Concepts

What does the "const" keyword do in Dart?

Previous

Next

MAD & PWA Lab

Journal

Experiment No.	03
Experiment Title.	To include icons, images, fonts in Flutter app
Roll No.	5
Name	Deepa Behrani
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	A++

Aim : To include icons, images, fonts in flutter app

Theory : In flutter, icons, images and ^{custom} fonts enhance the visual appeal of an application. flutter provides built-in support for adding these elements using different widgets & configuration.

Icons : flutter includes material icons by default which can be used with the icon widget.

ex : Icon (Icons home, size: 50, color: colors.blue)

For custom icons, developers can use third-party icons packs like font awesome or import custom icon fonts.

Images : flutter supports assets images, network images & file images are stored in the assets folder and in pubspec.yaml.

Fonts : custom fonts can be added by downloading font files and defining them in pubspec.yaml

The custom font is used in text widget

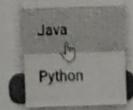
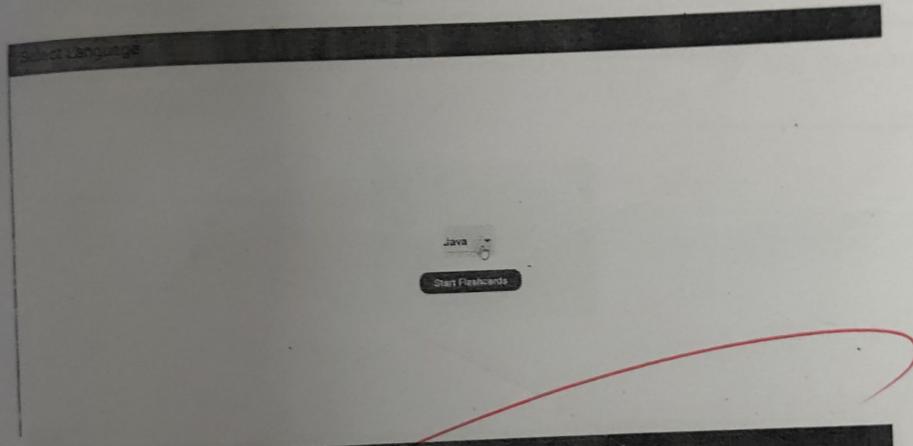
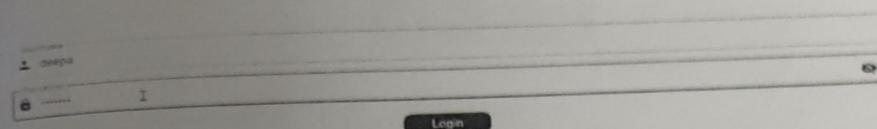
ex: Text ('Hello, flutter!', style: TextStyle
(fontFamily: 'custom font', fontSize: 24)),

Conclusion : By including icons, images & custom fonts, flutter apps can achieve a visually appealing & customized user interface. The easy integration of assets ensure a smooth development process making flutter a powerful framework for designing modern & engaging

EXPERIMENT NO: 03

DEEPA BEHRANI

D15B - 05



Java Features

What is Java?

Features

Java is a high-level, object-oriented
programming language.
It is platform-independent due to
JVM.
Java follows the typical (Hello World)
Sun Microsystems principle.
It is widely used for enterprise.

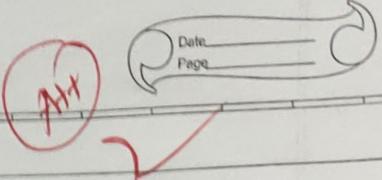
J.

MAD & PWA Lab

Journal

Experiment No.	04
Experiment Title.	To create an interactive Form using form widget
Roll No.	5
Name	Deepa Behrani
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	A++

Experiment - 04



Date _____
Page _____

Aim :- Create an interactive form using form widget

Theory :- Introduction to forms in flutter

Forms in flutter allows user to enter & submit data. The form widget is used to group multiple form fields such as text form fields and manage this validation & submission.

Components of flutter form

→ Global key <form state> : Helps in managing the forms state.

→ Text form field : Allows users to input text.

→ Validation logic : It should include validation and submit form.

→ Submit button used to validate & 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 button to validate & submit the form.

Conclusion: This experiment demonstrates how to create an interactive form in flutter using the form widget.

EXPERIMENT NO: 04

DEEPA BEHRANI

D15B - 05

CardCore Login

Username _____

Password _____

Login

Select Language ⓘ

Python ⓘ

Easy ▾

Flashcard

Cheatsheet

Quiz

Java ⓘ

Easy ▾

Flashcard

Cheatsheet

Quiz

JavaScript ⓘ

Easy ▾

Flashcard

Cheatsheet

Quiz

← Python Flashcards (Easy)

What is a list?

Show Answer

Next

MAD & PWA Lab

Journal

Experiment No.	05
Experiment Title.	To apply navigation, routing and gestures in Flutter App
Roll No.	5
Name	Deepa Behrani
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	A++

Experiment 5

Date _____
Page _____

8.
Max

Aim : To apply navigation, routing & gestures in flutter app

Theory : Navigation in flutter

Navigation in flutter refers to switching between different screens (pages) in an application. It is managed using navigation 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 screen.
- 3] Named screen : (`navigator.pushNamed`) navigator using pre-defined routes name

~~Routing In flutter~~

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

Types of routing

- 1] Imperative
- 2] Declarative

Gestures in flutter

Flutter provides the gesture detector widget to handle user touch interactions like tapping, swiping & dragging.

- 1] Tap gesture
- 2] Double tap gesture.

EXPERIMENT NO: 05

DEEPA BEHRANI

D15B - 05

← Python Cheatsheet

↔ def: Define function

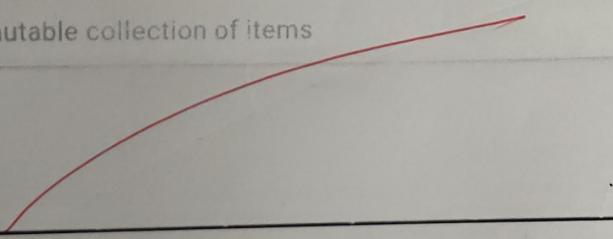
↔ for: Loop construct

↔ import: Include module

← Python Quiz - Easy

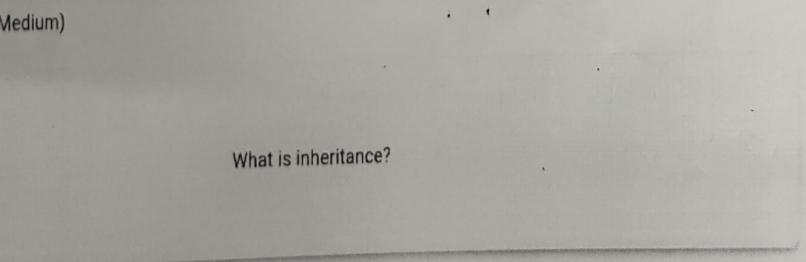
What is a list?

Answer: A mutable collection of items



← Java Flashcards (Medium)

What is inheritance?



Show Answer

Next

← Java Flashcards (Medium)

One class acquires properties of another.

Show Answer

Next

← Java Quiz - Medium

What is inheritance?

Answer: Acquiring properties of another class

← TypeScript Flashcards (Difficult)

Explain Generics.

Show Answer

Next

← TypeScript Flashcards (Difficult)

Write reusable and type-safe code.

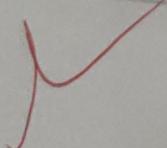
Show Answer

Next

← TypeScript Quiz - Difficult

Explain generics in TS

Answer: Create reusable components



MAD & PWA Lab

Journal

Experiment No.	06
Experiment Title.	To Connect Flutter UI with fireBase database
Roll No.	5
Name	Deepa Behrani
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
Grade:	

Experiment 6.

Aim : Setting up Firebase with flutter for android & iOS apps

Theory : Firebase is a Backend as a service (BaaS) provided by Google that helps developers build mobile & web applications with features like authentication, real-time database, cloud storage & analysis.

Steps to set up Firebase in a flutter app

- 1] Create a Firebase project
- 2] Add Firebase project
- 3] Install Firebase to flutter app
- 4] Initialize Firebase to flutter app.
- 5] Verify Firebase connection.

Conclusion : Firebase is a powerful BaaS that simplifies backend development for mobile apps. The integration process includes creating a Firebase project, adding platform-specific configuration files, & initializing Firebase in flutter. Once setup, Firebase can be used for authentication, real-time database & cloud storage.

MPL FIREBASE EXPERIMENT 06

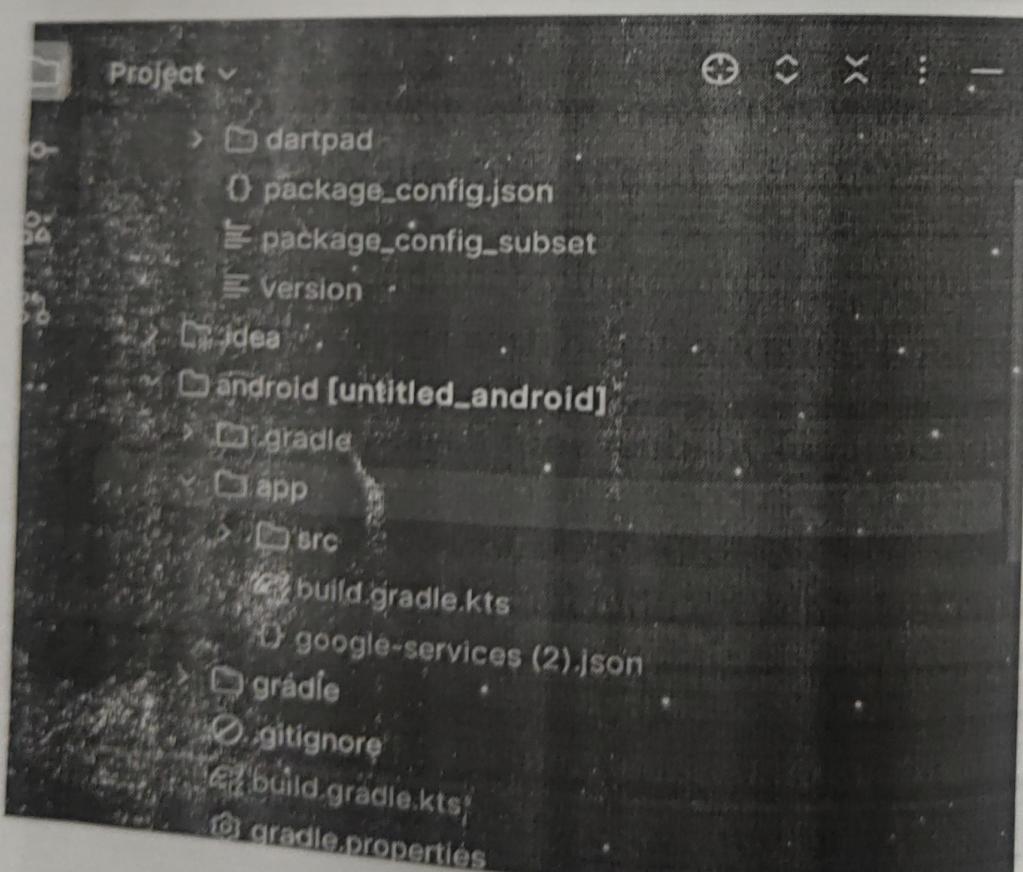
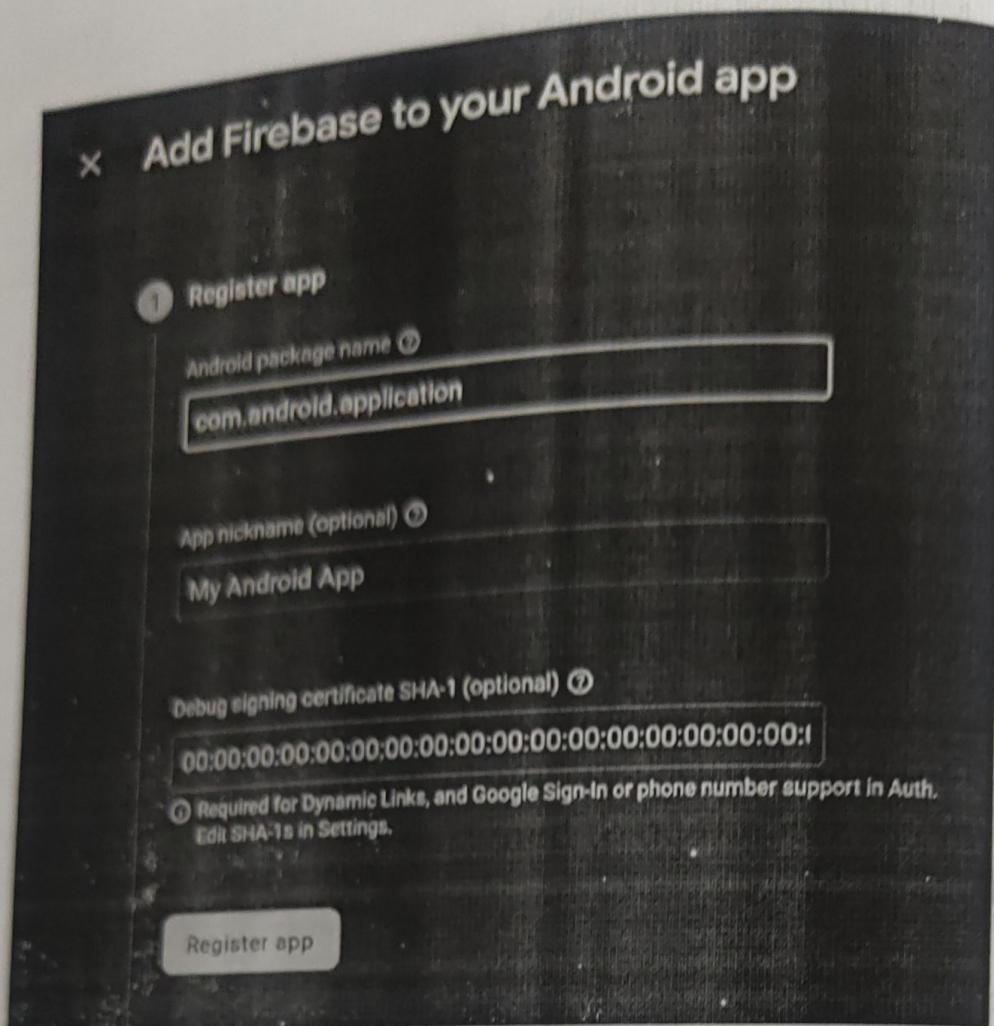
NAME : DEEPA BEHRANI

CLASS : D15B

ROLL_NO : 05

The screenshot shows the Firebase console interface. At the top, there's a navigation bar with icons for Home, Projects, Functions, Hosting, Storage, Database, Firestore, Cloud ML, and Authentication. Below the navigation bar, the title "MPL FIREBASE EXPERIMENT 06" is displayed. The main area is titled "Your projects". It lists three projects: "card-core" (last updated 20 days ago), "CardCore-flashcard" (last updated 1 day ago), and "cardcore" (last updated 9 days ago). Each project entry includes a thumbnail image of a person holding a smartphone, the project name, and a link to "ves.ac.in". A large button labeled "Create a Firebase project" is visible on the left side.

The screenshot shows a modal dialog box titled "Create a project". Inside the dialog, the text "Let's start with a name for your project" is displayed above a text input field. The input field contains the text "CardCore-flashCard". Below the input field, there are two buttons: "Create project" and "Cancel". At the bottom of the dialog, there are links for "Already have a Google Cloud project?" and "Add Firebase to Google Cloud project".



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

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

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

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.	5
Name	Deepa Behrani
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO4: Understand various PWA frameworks and their requirements
Grade:	A ++

MAD LAB Experiment - 07

Date _____
Page _____

Ques

Aim :- To develop a progressive web app (PWA) that can be installed on devices like a native application and provides offline functionality using a service worker and a web manifest file.

Theory : A progressive web app (PWA) is a modern web application 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 & serve cached responses when offline.

2] Web app manifest :

A JSON file that provides metadata about the PWA such as the app name, theme color, start URI & icons that makes the web app installed on device.

3] Caching Strategy :

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

Conclusion : The implementation successfully demonstrated the PWA architecture by enabling installation offline access & caching. Users can install the app & even if there is no internet connection previously loaded pages will still be accessible. This project highlights the benefits of PWAs such as improved performance, reliability & user engagement.

Teacher's Sign.: _____

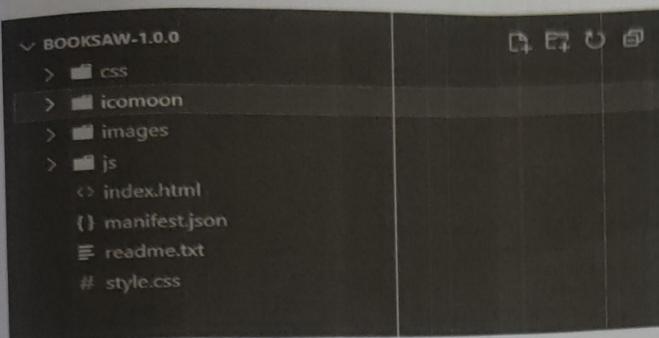
PWA EXPERIMENT - 07

Name- DEEPA BEHRANI

Roll no-05

D15B

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

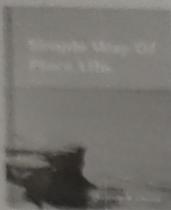


```
index.html manifest.json

manifest.json
{
  "name": "BookSaw - Free Book Store",
  "short_name": "BookSaw",
  "start_url": "index.html",
  "display": "standalone",
  "background_color": "#ffffff",
  "theme_color": "#4a90e2",
  "orientation": "portrait",
  "scope": "./",
  "description": "An elegant free HTML template for an online book store.",
  "icons": [
    {
      "src": "images/pattern1.png",
      "sizes": "144x144",
      "type": "image/png",
      "purpose": "any"
    },
    {
      "src": "images/item1.png",
      "sizes": "192x192",
      "type": "image/png",
      "purpose": "any"
    },
    {
      "src": "images/client-image1.png",
      "sizes": "512x512",
      "type": "image/png",
      "purpose": "any"
    }
  ]
}
```

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.

Featured Books



Simple Way Of Piece Life



Great Travel At Desert



The Lady Beauty Scarlet



Once Upon A Time

ADD TO CART

Application

- ▶ Manifest
- ▶ Service workers
- ▶ Storage

Storage

- ▶ Local storage

Service workers

- Offline Update on reload Bypass for network

Service workers from other origins

[See all registrations](#)

ServiceWorker

Open DevTools window and pause JavaScript execution on Service Worker startup for debugging.

Registrations in: C:\Users\Deepa\AppData\Local\Google\Chrome\User Data\Profile 6 (12)

```

Scope: chrome-extension://ghbmnnjooekpmoeecnllnnbdlohhki/
Storage key:
Origin: chrome-extension://ghbmnnjooekpmoeecnllnnbdlohhki
Top-level site: chrome-extension://ghbmnnjooekpmoeecnllnnbdlohhki
Ancestor chain bit: SameSite
Registration ID: 3
Navigation preload enabled: false
Navigation preload header length: 4
Active worker:
  Installation Status: ACTIVATED
  Running Status: STOPPED
  Fetch handler existence: DOES_NOT_EXIST
  Fetch handler type: NO_HANDLER
  Script: chrome-extension://ghbmnnjooekpmoeecnllnnbdlohhki/service_worker_bin_prod.js
  Version ID: 3
  Renderer process ID: 0
  Renderer thread ID: -1
  DevTools agent route ID: 2
  Client:
    ID: 11cd29dc-f693-428f-a3a2-5740006e30bc
    URL: chrome-extension://ghbmnnjooekpmoeecnllnnbdlohhki/offscreendocument.html?randomPercentageForSampling=78.80747974157394&sessionId=39bd848b1b0ec3d9
  Log:

```

[Unregister](#) [Start](#)

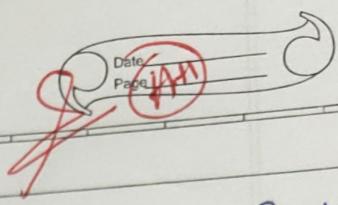
Scope: chrome-extension://neajdpkdcplabooebfdakcijh/

Conclusion: Through this experiment we learnt to add the 'add to my webpage' feature to our web application. This is the most fundamental step to be performed while building progressive web applications.

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.	5
Name	Deepa Behrani
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	A++



PWA Experiment No.: 08

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} ~~random~~ web application that provides an app like experience 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 service 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~~ Caching strategy: The cache first approach is used in this PWA meaning that if resources are available in cache they are served first. Otherwise they are fetched from the network.

Conclusion: The implementation successfully demonstrated the architecture by enabling installation offline access and caching. Users can install the app and even if there is no internet connection previously loaded pages will still be accessible. This project highlights the benefits of PWAs such as improved performance, reliability and user engagement making them a powerful alternative to traditional native apps.

EXPERIMENT 08

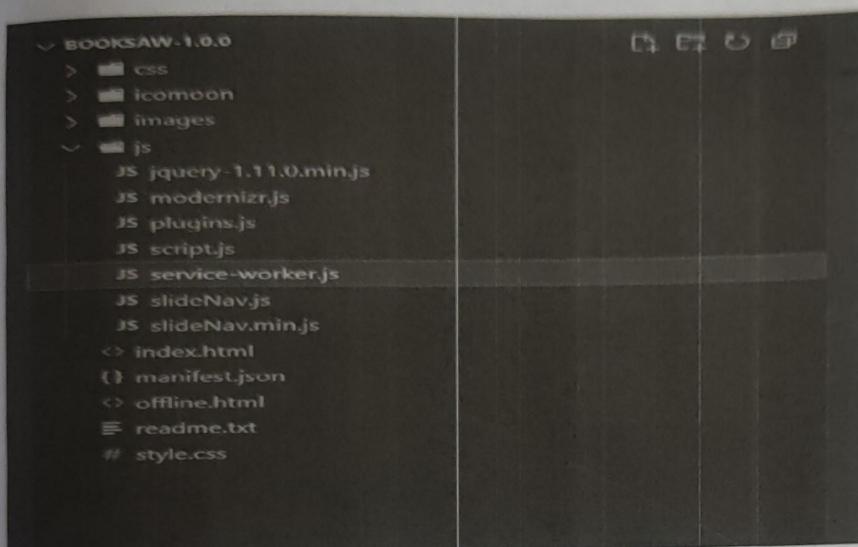
NAME : DEEPA BEHRANI

CLASS : D15B

ROLL_NO : 05

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



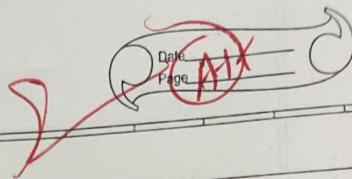
Create a cacheable file called offline.html to be displayed in the absence of an internet connection.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>offline</title>
<style>
body {
    font-family: sans-serif;
    background: #f2f2f2;
    text-align: center;
    padding: 50px;
}
h1 {
    color: #333;
}
</style>
</head>
<body>
<h1>You're offline</h1>
<p>Looks like you've lost connection. Please check your internet and try again.</p>
</body>
</html>
```

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.	5
Name	Deepa Behrani
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	A++



PWA Experiment - 9.

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 notifications. It acts as a proxy between web application and the network, enhancing performance, reliability and user experience.

Key features of service worker:

- = Runs in the background without user interaction.
- = Intercepts network requests and manages caching using the cache API.
- = Works only over HTTPS for security reasons.
- = User promises and indexed DB to handle data persistence.

Service worker Events :-

- = Fetch Event : Manages network requests and cache first. Uses cached data first fetching from the network only if needed.
- = Network First : Tries fetching from the network first. Falls back to cache if offline.
- = Sync Event : Enables background sync when the internet is restored.

Conclusion: Thus, we successfully implemented service worker events like fetch, sync and push in an e-commerce PWA, enabling offline functionality, background sync & real time notifications to improve user experience.

PWA EXPERIMENT No : 09

NAME : DEEPA BEHRANI
CLASS : D15B
ROLL_NO : 05

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
// Function to send pending screenshots to the server
// Push Event: Display push notifications

MAD & PWA Lab Journal

Experiment No.	10
Experiment Title.	To study and implement deployment of Ecommerce PWA to GitHub Pages.
Roll No.	5
Name	Deepa Behrani
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	A++

Experiment 11

SCB

Date _____
Page _____

Aim: To study and Implement the google lighthouse PWA Analysis tool to test the progressive web App (PWA) functioning.

Theory: Google Lighthouse is an open source tool by Google used to audit web applications based on various parameters including performance, PWA compatibility & best practices. It helps developers analyze and optimize web apps to function like native applications. Key Audit metrics:

- 1] Performance: Measures loading speed, rendering time and responsiveness.
- 2] PWA compliances: Ensures the app meets PWA standards such as service workers, offline functionality and manifests.
- 3] Accessibility: Evaluates the app's usability for differently abled users, including screen reader compatibility.
- 4] Best practices: Checks HTTPS usage, security policies and deprecated code elements.

Lighthouse provides actionable insights to improve website efficiency, a better user experience and higher engagement.

Conclusion: Thus we successfully used the google Lighthouse PWA Analysis tool to evaluate and optimize the PWA improving its performance, accessibility and compliance with best practices.

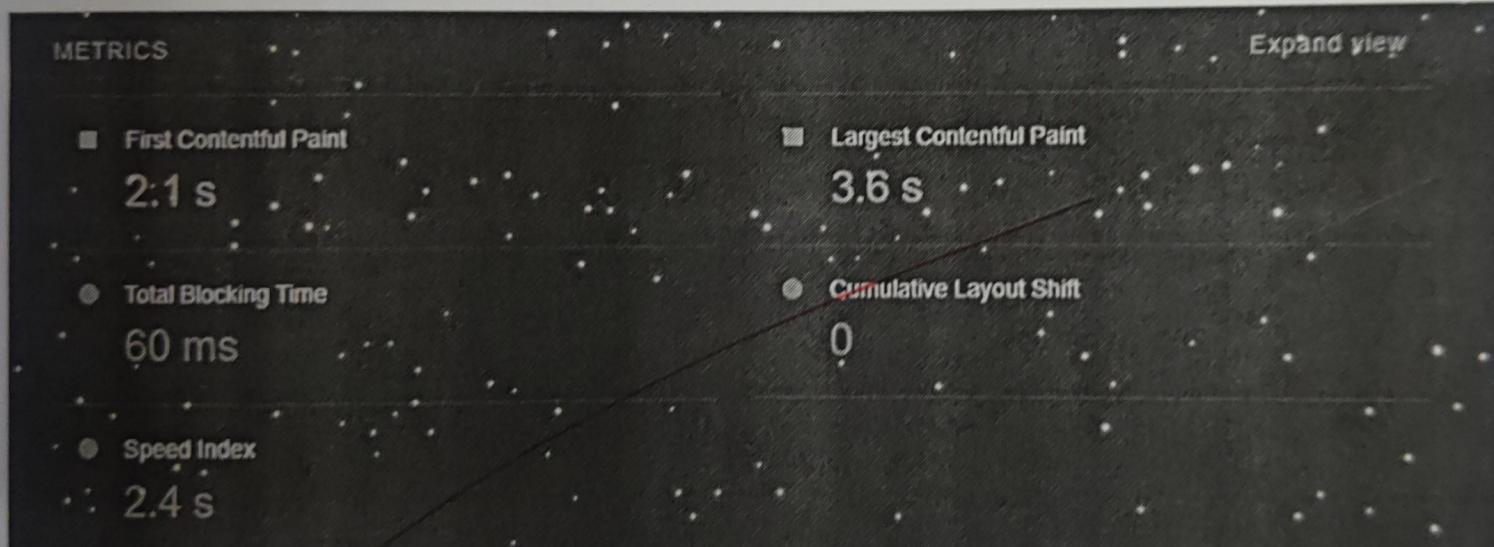
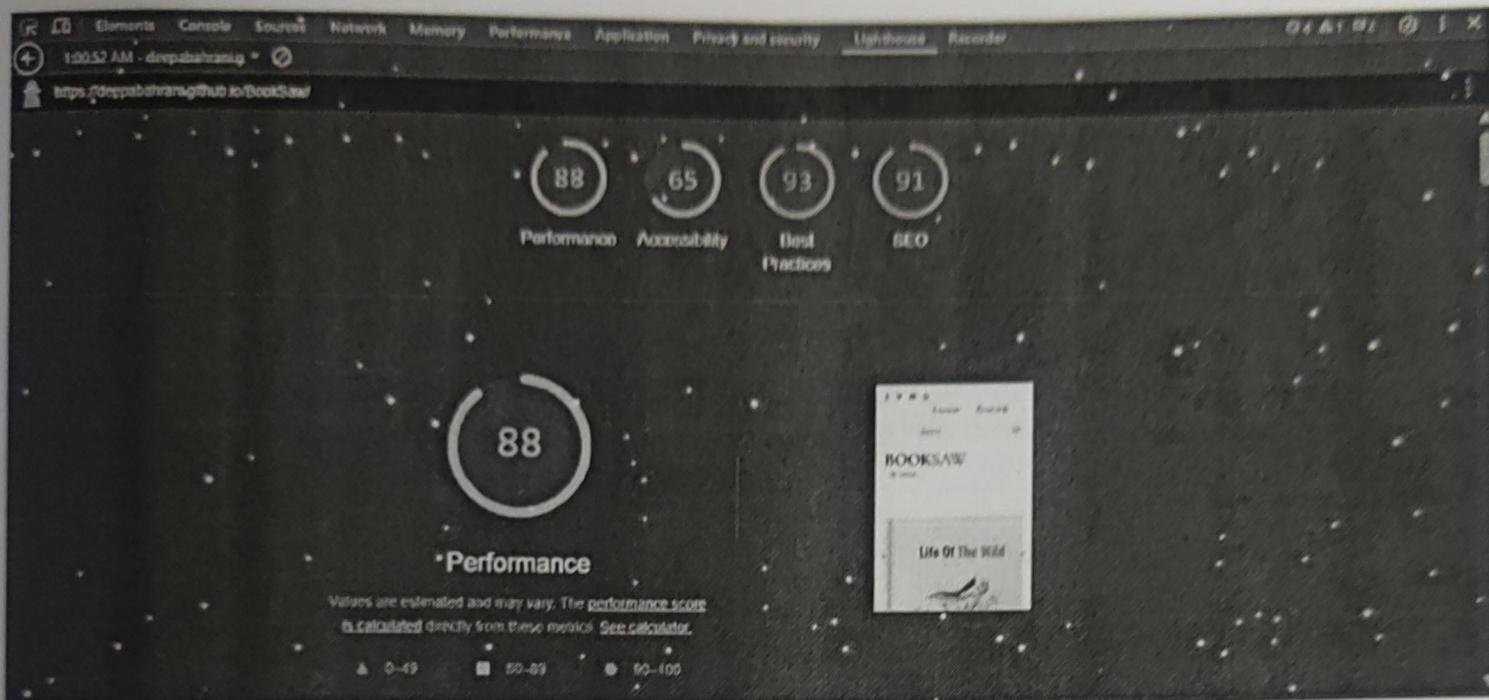
PWA EXPERIMENT NO : 11

NAME : DEEPA BEHRANI

CLASS : D15B

ROLL_NO: 05

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



Journal

Experiment No.	11
Experiment Title.	To use google Lighthouse PWA Analysis Tool to test the PWA functioning.
Roll No.	5
Name	Deepa Behrani
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO6: Develop and Analyze PWA Features and deploy it over app hosting solution
Grade:	A++

Experiment 10

Q1

Q2 Q3

Q4

Q5

Q6

Q7

Q8

Q9

Q10

Q11

Q12

Q13

Q14

Q15

Q16

Q17

Q18

Q19

Q20

Q21

Q22

Q23

Q24

Q25

Q26

Q27

Q28

Q29

Q30

Q31

Q32

Q33

Q34

Q35

Q36

Q37

Q38

Q39

Q40

Q41

Q42

Q43

Q44

Q45

Q46

Q47

Q48

Q49

Q50

Q51

Q52

Q53

Q54

Q55

Q56

Q57

Q58

Q59

Q60

Q61

Q62

Q63

Q64

Q65

Q66

Q67

Q68

Q69

Q70

Q71

Q72

Q73

Q74

Q75

Q76

Q77

Q78

Q79

Q80

Q81

Q82

Q83

Q84

Q85

Q86

Q87

Q88

Q89

Q90

Q91

Q92

Q93

Q94

Q95

Q96

Q97

Q98

Q99

Q100

Q101

Q102

Q103

Q104

Q105

Q106

Q107

Q108

Q109

Q110

Q111

Q112

Q113

Q114

Q115

Q116

Q117

Q118

Q119

Q120

Q121

Q122

Q123

Q124

Q125

Q126

Q127

Q128

Q129

Q130

Q131

Q132

Q133

Q134

Q135

Q136

Q137

Q138

Q139

Q140

Q141

Q142

Q143

Q144

Q145

Q146

Q147

Q148

Q149

Q150

Q151

Q152

Q153

Q154

Q155

Q156

Q157

Q158

Q159

Q160

Q161

Q162

Q163

Q164

Q165

Q166

Q167

Q168

Q169

Q170

Q171

Q172

Q173

Q174

Q175

Q176

Q177

Q178

Q179

Q180

Q181

Q182

Q183

Q184

Q185

Q186

Q187

Q188

Q189

Q190

Q191

Q192

Q193

Q194

Q195

Q196

Q197

Q198

Q199

Q200

Q201

Q202

Q203

Q204

Q205

Q206

Q207

Q208

Q209

Q210

Q211

Q212

Q213

Q214

Q215

Q216

Q217

Q218

Q219

Q220

Q221

Q222

Q223

Q224

Q225

Q226

Q227

Q228

Q229

Q230

Q231

Q232

Q233

Q234

Q235

Q236

Q237

Q238

Q239

Q240

Q241

Q242

Q243

Q244

Q245

Q246

Q247

Q248

Q249

Q250

Q251

Q252

Q253

Q254

Q255

Q256

Q257

Q258

Q259

Q260

Q261

Q262

Q263

Q264

Q265

Q266

Q267

Q268

Q269

Q270

Q271

Q272

Q273

Q274

Q275

Q276

Q277

Q278

Q279

Q280

Q281

Q282

Q283

Q284

Q285

Q286

Q287

Q288

Q289

Q290

Q291

Q292

Q293

Q294

Q295

Q296

Q297

Q298

Q299

Q300

Q301

Q302

Q303

Q304

Q305

Q306

Q307

Q308

Q309

Q310

Q311

Q312

Q313

Q314

Q315

Q316

Q317

Q318

Q319

Q320

Q321

Q322

Q323

Q324

Q325

Q326

Q327

Q328

Q329

Q330

Q331

Q332

Q333

Q334

Q335

Q336

Q337

Q338

Q339

Q340

Q341

Q342

Q343

Q344

Q345

Q346

Q347

Q348

Q349

Q350

Q351

Q352

Q353

Q354

Q355

Q356

Q357

Q358

Q359

Q360

Q361

Q362

Q363

Q364

Q365

Q366

Q367

Q368

Q369

Q370

Q371

Q372

Q373

Q374

Q375

Q376

Q377

Q378

Q379

Q380

Q381

Q382

Q383

Q384

Q385

Q386

Q387

Q388

Q389

Q390

Q391

Q392

Q393

Q394

Q395

Q396

Q397

Q398

Q399

Q400

Q401

Q402

Q403

Q404

Q405

Q406

Q407

Q408

Q409

Q410

Q411

Q412

Q413

Q414

Q415

Q416

Q417

Q418

Q419

Q420

Q421

Q422

Q423

Q424

Q425

Q426

Q427

Q428

Q429

Q430

Q431

Q432

Q433

Q434

Q435

Q436

Q437

Q438

Q439

Q440

Q441

Q442

Q443

Q444

Q445

Q446

Q447

Q448

Q449

Q450

Q451

Q452

Q453

Q454

Q455

Q456

Q457

Q458

Q459

Q460

Q461

Q462

Q463

Q464

Q465

Q466

Q467

Q468

Q469

Q470

Q471

Q472

Q473

Q474

Q475

Q476

Q477

Q478

Q479

Q480

Q481

Q482

Q483

Q484

Q485

Q486

Q487

Q488

Q489

Q490

Q491

Q492

Q493

Q494

Q495

Q496

Q497

Q498

Q499

Q500

Q501

Q502

Q503

Q504

Q505

Q506

Q507

Q508

Q509

Q510

Q511

Q512

Q513

Q514

Q515

Q516

Q517

Q518

Q519

Q520

Q521

Q522

Q523

Q524

Q525

Q526

Q527

Q528

Q529

Q530

Q531

Q532

Q533

Q534

Q535

Q536

Q537

Q538

Q539

Q540

Q541

Q542

Q543

Q544

Q545

Q546

Q547

Q548

Q549

Q550

Q551

Q552

Q553

Q554

Q555

Q556

Q557

Q558

Q559

Q560

Q561

Q562

Q563

Q564

Q565

Q566

Q567

Q568

Q569

Q570

Q571

Q572

Q573

Q574

Q575

Q576

Q577

Q578

Q579

Q580

Q581

Q582

Q583

Q584

Q585

Q586

Q587

Q588

Q589

Q590

Q591

Q592

Q593

Q594

Q595

Q596

Q597

Q598

Q599

Q600

Q601

Q602

Q603

Q604

Q605

Q606

Q607

Q608

Q609

Q610

Q611

Q612

Q613

Q614

Q615

Q616

Q617

Q618

Q619

Q620

Q621

Q622

Q623

Q624

Q625

Q626

Q627

Q628

Q629

Q630

Q631

Q632

Q633

Q634

Q635

Q636

Q637

Q638

Q639

Q640

Q641

Q642

Q643

Q644

Q645

Q646

Q647

Q648

Q649

Q650

Q651

Q652

Q653

Q654

Q655

Q656

Q657

Q658

Q659

Q660

Q661

Q662

Q663

Q664

Q665

Q666

Q667

Q668

Q669

Q670

Q671

Q672

Q673

Q674

Q675

Q676

Q677

Q678

Q679

Q680

Q681

Q682

Q683

Q684

Q685

Q686

Q687

Q688

Q689

Q690

Q691

Q692

Q693

Q694

Q695

Q696

Q697

Q698

Q699

Q700

Q701

Q702

Q703

Q704

Q705

Q706

Q707

Q708

Q709

Q710

Q711

Q712

Q713

Q714

Q715

Q716

Q717

Q718

Q719

Q720

Q721

Q722

Q723

Q724

Q725

Q726

Q727

Q728

Q729

Q730

Q731

Q732

Q733

Q734

Q735

Q736

Q737

Q738

Q739

Q740

Q741

Q742

Q743

Q744

Q745

Q746

Q747

Q748

Q749

Q750

Q751

Q752

Q753

Q754

Q755

Q756

Q757

Q758

Q759

Q760

Q761

Q762

Q763

Q764

Q765

Q766

Q767

Q768

Q769

Q770

Q771

Q772

Q773

Q774

Q775

Q776

Q777

Q778

Q779

Q780

Q781

Q782

Q783

Q784

Q785

Q786

Q787

Q788

Q789

Q790

Q791

Q792

Q793

Q794

Q795

Q796

Q797

Q798

Q799

Q800

Q801

Q802

Q803

Q804

Q805

Q806

Q807

Q808

Q809

Q810

Q811

Q812

Q813

Q814

Q815

Q816

Q817

Q818

Q819

Q820

Q821

Q822

Q823

Q824

Q825

Q826

Q827

Q828

Q829

Q830

Q831

Q832

Q833

Q834

Q835

Q836

Q837

Q838

Q839

Q840

Q841

Q842

Q843

Q844

Q845

Q846

Q847

Q848

Q849

Q850

Q851

Q852

Q853

Q854

Q855

Q856

Q857

Q858

Q859

Q860

Q861

Q862

Q863

Q864

Q865

Q866

Q867

Q868

Q869

Q870

Q871

Q872

Q873

Q874

Q875

Q876

Q877

Q878

Q879

Q880

Q881

Q882

Q883

Q884

Q885

Q886

Q887

Q888

Q889

Q890

Q891

Q892

Q893

Q894

Q895

Q896

Q897

Q898

Q899

Q900

Q901

Q902

Q903

Q904

Q905

Q906

Q907

Q908

Q909

Q910

Q911

Q912

Q913

Q914

Q915

Q916

Q917

Q918

Q919

Q920

Q921

Q922

Q923

Q924

Q925

Q926

Q927

Q928

Q929

Q930

Q931

Q932

Q933

Q934

Q935

Q936

Q937

Q938

Q939

Q940

Q941

Q942

Q943

Q944

Q945

Q946

Q947

Q948

Q949

Q950

Q951

Q952

Q953

Q954

Q955

Q956

Q957

Q958

Q959

Q960

Q961

Q962

Q963

Q964

Q965

Q966

Q967

Q968

Q969

Q970

Q971

Q972

Q973

Q974

Q975

Q976

Q977

Q978

Q979

Q980

Q981

Q982

Q983

Q984

Q985

Q986

Q987

Q988

Q989

Q990

Q991

Q992

Q993

Q994

Q995

Q996

Q997

Q998

Q999

Q1000

Q1001

Q1002

Q1003

Q1004

Q1005

Q1006

Q1007

Q1008

Q1009

Q1010

Q1011

Q1012

Q1013

Q1014

Q1015

Q1016

Q1017

Q1018

Q1019

Q1020

Q1021

Q1022

Q1023

Q1024

Q1025

Q1026

Q1027

Q1028

Q1029

Q1030

Q1031

Q1032

Q1033

Q1034

Q1035

Q1036

Q1037

Q1038

Q1039

Q1040

Q1041

Q1042

Q1043

Q1044

Q1045

Q1046

Q1047

Q1048

Q1049

Q1050

Q1051

Q1052

Q1053

Q1054

Q1055

Q1056

Q1057

Q1058

Q1059

Q1060

Q1061

Q1062

Q1063

Q1064

Q1065

Q1066

Q1067

Q1068

Q1069

Q1070

Q1071

Q1072

Q1073

Q1074

Q1075

Q1076

Q1077

Q1078

Q1079

Q1080

Q1081

Q1082

Q1083

Q1084

Q1085

Q1086

Q1087

Q1088

Q1089

Q1090

Q1091

Q1092

Q1093

Q1094

Q1095

Q1096

Q1097

Q1098

Q1099

Q1100

Q1101

Q1102

Q1103

Q1104

Q1105

Q1106

Q1107

Q1108

Q1109

Q1110

Q1111

Q1112

Q1113

Q1114

Q1115

Q1116

Q1117

Q1118

Q1119

Q1120

Q1121

Q1122

Q1123

Q1124

Q1125

Q1126

Q1127

Q1128

Q1129

Q1130

Q1131

Q1132

Q1133

Q1134

Q1135

Q1136

Q1137

Q1138

Q1139

Q1140

Q1141

Q1142

Q1143

Q1144

Q1145

Q1146

Q1147

Q1148

Q1149

Q1150

Q1151

Q1152

Q1153

Q1154

Q1155

Q1156

Q1157

Q1158

Q1159

Q1160

Q1161

Q1162

Q1163

Q1164

Q1165

Q1166

Q1167

Q1168

Q1169

Q1170

Q1171

Q1172

Q1173

Q1174

Q1175

Q1176

Q1177

Q1178

Q1179

Q1180

Q1181

Q1182

Q1183

Q1184

Q1185

Q1186

Q1187

Q1188

Q1189

Q1190

Q1191

Q1192

Q1193

Q1194

Q1195

Q1196

Q1197

Q1198

Q1199

Q1200

Q1201

Q1202

Q1203

Q1204

Q1205

Q1206

Q1207

Q1208

Q1209

Q1210

Q1211

Q1212

Q1213

Q1214

Q1215

Q1216

Q1217

Q1218

Q1219

Q1220

Q1221

Q1222

Q1223

Q1224

Q1225

Q1226

Q1227

Q1228

Q1229

Q1230

Q1231

Q1232

Q1233

Q1234

Q1235

Q1236

Q1237

Q1238

Q1239

Q1240

Q1241

Q1242

Q1243

Q1244

Q1245

Q1246

Q1247

Q1248

Q1249

Q1250

Q1251

Q1252

Q1253

Q1254

Q1255

Q1256

Q1257

Q1258

Q1259

Q1260

Q1261

Q1262

Q1263

Q1264

Q1265

Q1266

Q1267

Q1268

Q1269

Q1270

Q1271

Q1272

Q1273

Q1274

Q1275

Q1276

Q1277

Q1278

Q1279

Q1280

Q1281

Q1282

Q1283

Q1284

Q1285

Q1286

Q1287

Q1288

Q1289

Q1290

Q1291

Q1292

Q1293

Q1294

Q1295

Q1296

Q1297

Q1298

Q1299

Q1300

Q1301

Q1302

Q1303

Q1304

Q1305

Q1306

Q1307

Q1308

Q1309

Q1310

Q1311

Q1312

Q1

PWA EXPERIMENT NO : 10

NAME : DEEPA BEHRANI

CLASS : D15B

ROLL_NO: 05

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

The screenshot shows a GitHub repository named 'BookSaw'. The repository has 1 branch and 0 tags. The 'master' branch contains several commits from the user 'deepabahrani' over the past minute. The commits are all identical and labeled 'bookSaw'. The repository has no description, website, or topics provided. It has 0 stars, 1 watching, and 0 forks. There are no releases or packages published.

The screenshot shows the GitHub Pages settings for the 'BookSaw' repository. The repository is live at <https://deepabahrani.github.io/BookSaw/>. The site was last deployed by 'deepabahrani' 5 minutes ago. The build and deployment section shows that the site is built from the 'master' branch. A red arrow points from the bottom left towards this section. The GitHub Pages environment is currently using the 'gh-pages' environment, which is the default for GitHub Pages. Learn more about deploying to GitHub Pages using custom workflows.

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.	5
Name	Deepa Behrani
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:	A++ (415)

MPL ASSIGNMENT 1

(Q1) (a) Explain the key features and advantages of using flutter for mobile app development.

→ Key Features of flutter:

- (1) Single codebase: flutter allows developers to write a single codebase for both android and iOS reducing development effort.
- (2) Fast development with hot reload: This feature enables developers to see changes instantly without restarting the entire app.
- (3) UI with widgets: flutter provides customizable and pre-designed widgets to get visually appealing user interface.
- (4) High performance: uses the dart programming language which compiles into native ARM code for optimized performance.
- (5) Support for web and desktop: Besides mobile apps, flutter also supports web and desktop development.
- (6) Access to Native features: Developers can use platform-specific APIs & third party plugins to integrate native functionalities.
- (7) Open source: free and open source allowing developers to customize and enhance the framework.
- (8) Backed by google: Ensures long-term support and continuous improvements.

~~ADVANTAGES OF USING FLUTTER~~

- Faster development due to code reusability and reloads.
- Cost efficiency as a single codebase reduces development and maintenance costs.
- Consistent UI across multiple platforms.
- Better performance compared to hybrid frameworks as it compiles directly to native code.
- Strong integration with firebase for backend services like authentication & database management.

build small, reusable widget that are combined to form complete UIs.

ex: A column widget can hold multiple Text & Button widget, creating a structured layout.

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

→ i) Structural widget

⇒ Scaffold: Provide basic structure of a screen.

⇒ Container: Used for layout styling.

⇒ Column & Row: Used for vertical & horizontal layout.

2) Interactive widget

⇒ Textfield: for user input

⇒ Elevated Button: Clickable buttons.

3) Styling widget

⇒ Padding: - Adds spacing around widget

⇒ Align, centre: - Adjust alignment.

4) List & scrollable widget

⇒ ListView: - Scrollable list

⇒ GridView: - Provide / Display items in grid.

ex: Simple widget tree

scaffold (

appBar: AppBar(title: Text("Flutter App")),

body: Column (

children: [

Text("Welcome to flutter!"),

b) Compare and contrast the different state management in flutter approaches available in flutter such as setState, Provider, & Riverpod. Provide scenarios where each approach is suitable.

→ Comparison of state management approaches in flutter

Approaches	Description	Suitable Scenarios	setState	Basic state management by calling setState() to update UI. Small apps, simple UI updates (e.g. toggling a switch)	Provider	User inherited widget to efficiently manage state across the widget tree. Medium sized apps needing global state sharing (e.g. user authentication)	Riverpod	more scalable than provider with improved dependency injection & state handling, large complex apps requiring modular & scalable state management (e.g. e-commerce apps)
------------	-------------	--------------------	----------	---	----------	---	----------	--

4a) 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 Benefits :-

Integration process :

Setup firebase console :

Create a firebase project

Register the App for android & iOS

Download & add google-services.json (Android) or Google Service Info.plist (iOS)

Install firebase Dependencies

Firebase, Analytics App usage analytics

Data Synchronization in firestore:

firestore allows real time data syncing using
Snapshot listener.

Ex: of real time listener in firestore:
dart.

~~firebase~~

```
firebase.firestore.instance.collection('message').snapshots(),
listen ((snapshot) {
  for (var doc in snapshot.docs) {
    print(doc.data());
  }
});
```

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.	5
Name	Deepa Behrani
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:	A++ (5/5)

MPL ASSIGNMENT 2:

Q1 Define progressive web app (PWA) & explain its significance in modern development. Q2 Discuss the key characteristics PWA's from traditional mobile apps.

Soln:

A progressive web app (PWA) is a type of web application that works like a mobile app & runs in a browser.

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.

	PWA	Traditional App
Installation:	Direct from Browser	Download from app store.
Internet required	work offline with caching	Usually requires internet.
Performance	Fast with service workers	Fast but need installation
Updates	Automatic	manual
Development cost	Lower	Higher

Q3 Define responsive web design & explain its importance in the context of progressive web apps. Compare & contrast responsive fluid & adaptive web design approaches.

Soln: Definition of responsive web design:

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 tablets & desktops without needing separate websites.

Importance :

- 1) Better user experience
- 2) faster Load time
- 3) SEO Benefits
- 4) cost effective.

Approach	How it works	Pros
responsive	I use flexible grids I use CSS media queries	I works on all devices
fluid	I Use % based widths, I Instead of fixed pixels	works well on different screen size,

(Q3) Describes the lifecycle of service workers, registration, installation & activation phases

Q3: Lifecycle of service workers:

A service workers is a script that runs background & helps a web app work off faster & send push notifications. It has 3 phases.

- 1] Registration phase : Browser register the worker using Javascript. Eg :

```
navigator.serviceWorker.register('service-worker.js')
```
- 2] Then () => console.log ("Service worker registered")
catch (error) => console.log ("Unsuccessful")

2) Installation phase: The service worker downloads necessary files (HTML, CSS, JS) & stores them in cache. If successful, it moves to the activation phase.
eg: self.addEventListener('install', event => {
 event.waitUntil(
 cache.open('app-cache').then(cache => {
 return cache.addAll(['index.html',
 'styles.css']);
 })
);
});

3] Activation phase.

The old service worker is replaced with new one. Unreachable cache files from the previous version are deleted. Final step: fetch and sync.
Once activated, the service worker intercepts network requests and serves cached files & syncs data.

Q] Explain the use of ~~internal~~ IndexedDB in the service worker for data storage.

→ Use of IndexedDB in service worker for data storage: IndexedDB is a browser database that stores large amount of structured data like JSON objects. It helps PWAs work offline by serving & retrieving data efficiently.

Why use IndexedDB in service workers?

1] Offline support: Store data when offline and sync it later.

2] Efficient storage: Saves structured data like user settings, cart items or form inputs.

3] Faster access: Retrieves data quickly without needing a network request.

4] Persistent Data: Data remains saved after the browser is closed.