

Vivekanand Education Society's Institute of Technology

An Autonomous Institute Affiliated to University of Mumbai
Hashu Advani Memorial Complex, Collector Colony, Chembur East, Mumbai - 400074.



Department of Information Technology

CERTIFICATE

This is to certify that Sneha Utekar of D15A semester VI, have successfully completed necessary experiments in the MAD & PWA Lab under my supervision in VES Institute of Technology during the academic year 2023-2024.

Lab Assistant

Subject Teacher

Mrs. Kajal Joseph

Principal

Head of Department

Dr. Mrs. Shalu Chopra

Name of the Course : MAD & PWA Lab**Course Code :** ITL604**Year/Sem/Class :** D15A**A.Y.:** 23-24**Faculty Incharge :** Mrs. Kajal Joseph.**Lab Teachers :** Mrs. Kajal Jewani.**Email :** kajal.jewani@ves.ac.in**Programme Outcomes:** The graduate will be able to:

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

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

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

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

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

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

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

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

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

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

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

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

Program specific Outcomes

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

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

Lab Objectives:

Sr. No.	Lab Objectives
The Lab experiments aims:	
1	Learn the basics of the Flutter framework.
2	Develop the App UI by incorporating widgets, layouts, gestures and animation
3	Create a production ready Flutter App by including files and firebase backend service.
4	Learn the Essential technologies, and Concepts of PWAs to get started as quickly and efficiently as possible
5	Develop responsive web applications by combining AJAX development techniques with the jQuery JavaScript library.
6	Understand how service workers operate and also learn to Test and Deploy PWA.

Lab Outcomes:

Sr. No.	Lab Outcomes	Cognitive levels of attainment as per Bloom's Taxonomy
On Completion of the course the learner/student should be able to:		
1	Understand cross platform mobile application development using Flutter framework	L1, L2
2	Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation	L3
3	Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS	L3, L4
4	Understand various PWA frameworks and their requirements	L1, L2
5	Design and Develop a responsive User Interface by applying PWA Design techniques	L3
6	Develop and Analyze PWA Features and deploy it over app hosting solutions	L3, L4

Index

Sr. No	Experiment Title	LO	DOP	DOS	Grade
1.	To install and configure the Flutter Environment	LO1	19/01	26/01	15
2.	To design Flutter UI by including common widgets.	LO2	26/01	02/02	15
3.	To include icons, images, fonts in Flutter app	LO2	02/02	09/02	15
4.	To create an interactive Form using form widget	LO2	09/02	16/02	15
5.	To apply navigation, routing and gestures in Flutter App	LO2	16/02	23/02	15
6.	To Connect Flutter UI with fireBase database	LO3	23/02	08/03	15
7.	To write meta data of your Ecommerce PWA in a Web app manifest file to enable “add to homescreen feature”.	LO4	08/03	15/03	15
8.	To code and register a service worker, and complete the install and activation process for a new service worker for the E-commerce PWA	LO5	15/03	22/03	15
9.	To implement Service worker events like fetch, sync and push for E-commerce PWA	LO5	22/03	31/03	15
10.	To study and implement deployment of Ecommerce PWA to GitHub Pages.	LO5	29/03	31/03	15
11.	To use google Lighthouse PWA Analysis Tool to test the PWA functioning.	LO6	29/03	31/03	15
12.	Assignment-1	LO1,LO2 ,LO3	28/01	05/02	5
13.	Assignment-2	LO4,LO5 ,LO6	14/03	21/03	4

MAD & PWA Lab

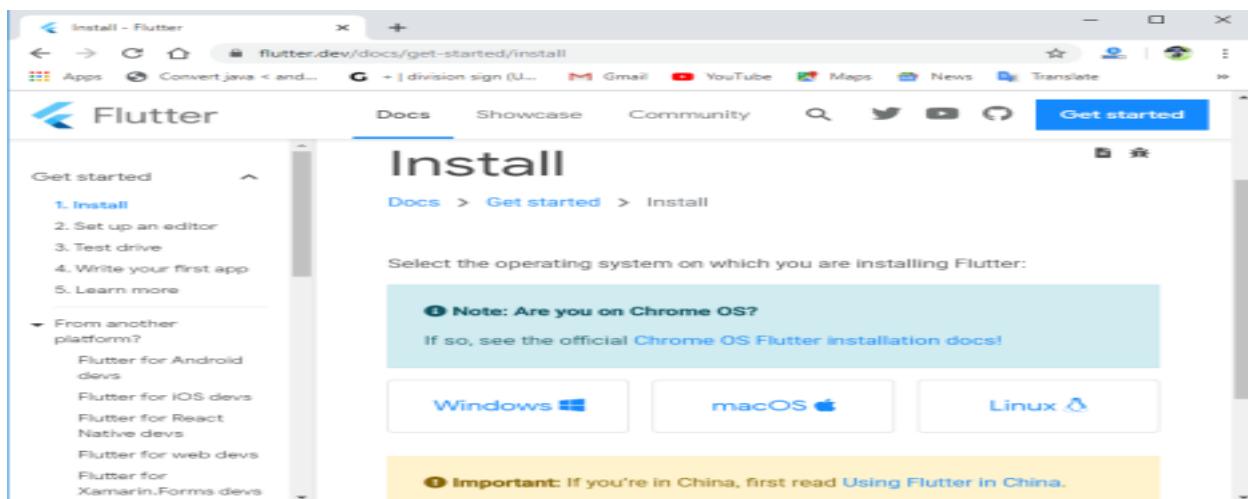
Journal

Experiment No.	01
Experiment Title.	To install and configure the Flutter Environment
Roll No.	66
Name	Sneha Utekar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO1: Understand cross platform mobile application development using Flutter framework
Grade:	15

Experiment 1

AIM: To install Flutter and Android Studio and Create a “Hello World App” using Flutter .

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.

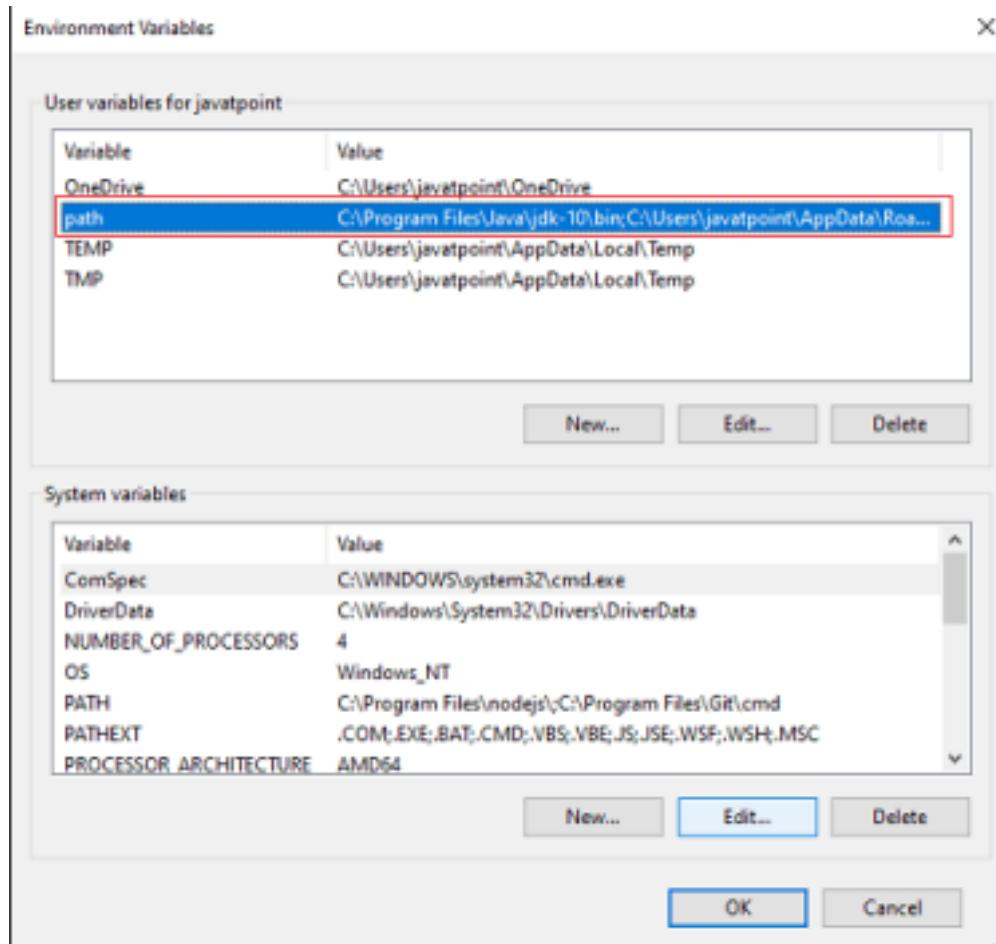


Step 2: Next, to download the latest Flutter SDK, click on the Windows icon. Here, you will find the download link for SDK.

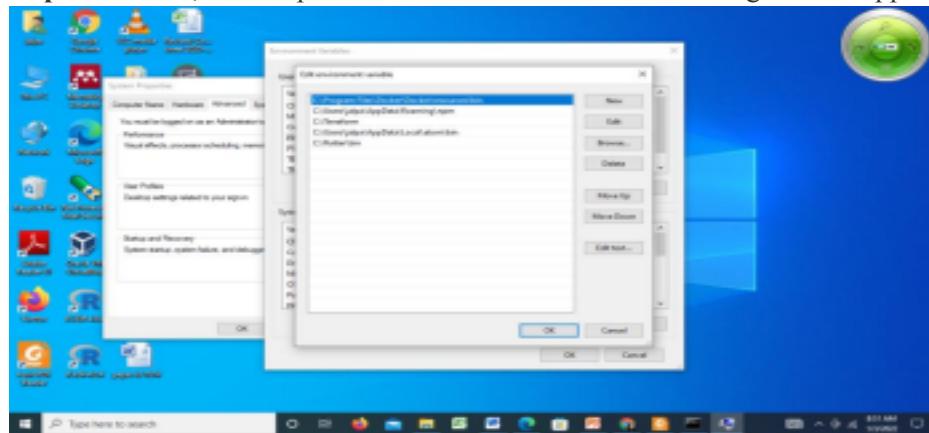
Step 3: When your download is complete, extract the **zip** file and place it in the desired installation folder or location, for example, C: /Flutter.

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.

```

Command Prompt
Microsoft Windows [Version 10.0.19042.1435]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jalpa\flutter
Manage your Flutter app development.

Common commands:

  flutter create <output directory>
    Create a new Flutter project in the specified directory.

  flutter run [<options>]
    Run your Flutter application on an attached device or in an emulator.

Usage: flutter <command> [<arguments>]

Global options:
  -h, --help           Print this usage information.
  -v, --verbose        Noisy logging, including all shell commands executed.
                      If used with "--help", show hidden options. If used with "flutter doctor", show additional
  (use "-vv" to force verbose logging in those cases.)                                diagnostic information
  -d, --device-id      Target device id or name (prefixes allowed).
  --version           Reports the version of this tool.
  --suppress-analytics Suppress analytics reporting when this command runs.

Available commands:

Flutter SDK
  bash-completion   Output command line shell completion setup scripts.
  channel           List or switch Flutter channels.
  config            Configure Flutter settings.
  doctor             Show information about the installed tooling.
  downgrade         Degrade Flutter to the last active version for the current channel.
  precache          Populate the Flutter tool's cache of binary artifacts.
  upgrade           Upgrade your copy of Flutter.

Project
  analyze           Analyze the project's Dart code.
  assemble          Assemble and build Flutter resources.
  build              Build an executable app or install bundle.
  clean              Delete the build/ and .dart_tool/ directories.
  create             Create a new Flutter project.
  drive              Run integration tests for the project on an attached device or emulator.
  format             Format one or more Dart files.

```

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.

```

Select-Command Prompt
See Google's privacy policy:
https://policies.google.com/privacy

C:\Users\jalpa>
C:\Users\jalpa>
C:\Users\jalpa>flutter doctor
Running "Flutter pub get" in flutter-tools...                                         17.86
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 2.8.1, on Microsoft Windows [Version 10.0.19042.1435], locale en-US)
[!] Android toolchain - develop for Android devices
  X Unable to locate Android SDK.
    Install Android Studio from: https://developer.android.com/studio/index.html
    On first launch it will assist you in installing the Android SDK components.
    (or visit https://flutter.dev/docs/get-started/install/windows/android-setup for detailed instructions).
    If the Android SDK has been installed to a custom location, please use
    'flutter config --android-sdk' to update to that location.

[!] Chrome - Develop for the web
[!] Android Studio (not installed)
[!] VS Code (version 1.55.2)
[!] Connected device (2 available)

Doctor found issues in 2 categories.

C:\Users\jalpa>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 2.8.1, on Microsoft Windows [Version 10.0.19042.1435], locale en-US)
[!] android toolchain - develop for Android devices (Android NDK version 23.0.8)
  X cmdline-tools component is missing.
    Run 'flutter doctor --android-licenses' to accept the NDK licenses.
    See https://developer.android.com/studio/command-line for more details.
  X android license status unknown.
    Run 'flutter doctor --android-licenses' to accept the NDK licenses.
    See https://flutter.dev/docs/get-started/install/windows/android-setup for more details.
[!] Chrome - Develop for the web
[!] Android Studio (version 2020.3)
[!] VS Code (version 1.55.2)
[!] Connected device (2 available)

Doctor found issues in 1 category.

```

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 are required to run Flutter as well as the development tools that are available but not connected with the device.

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,

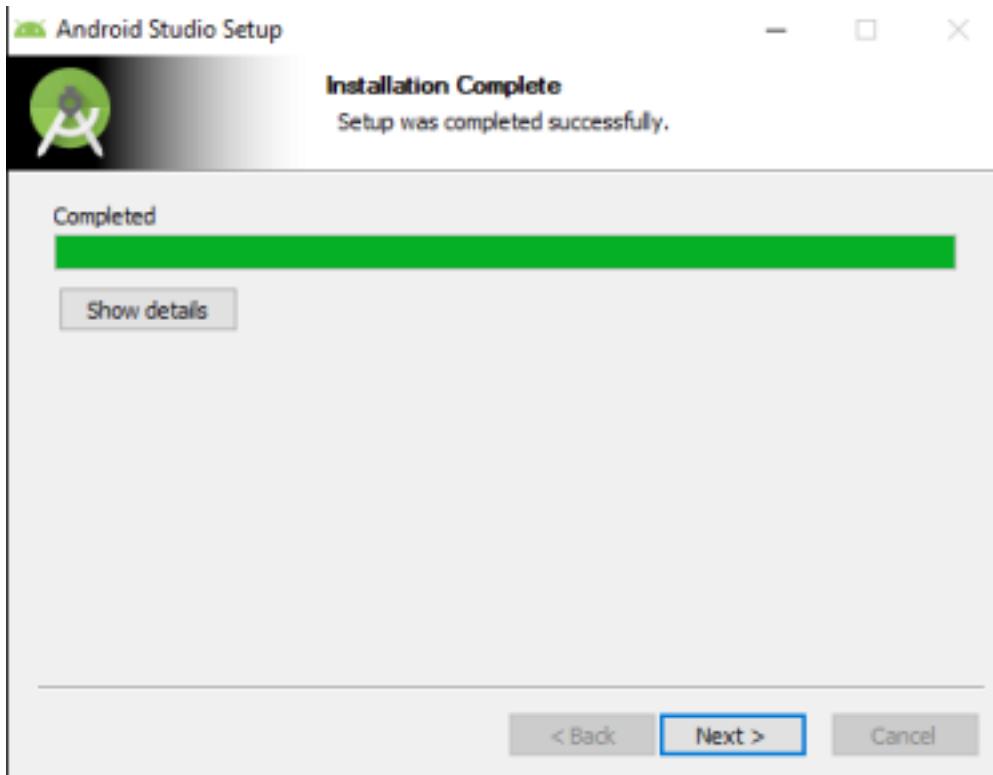
do the following steps.

Step 7.1: Download the latest Android Studio executable or zip file from the [official site](#).

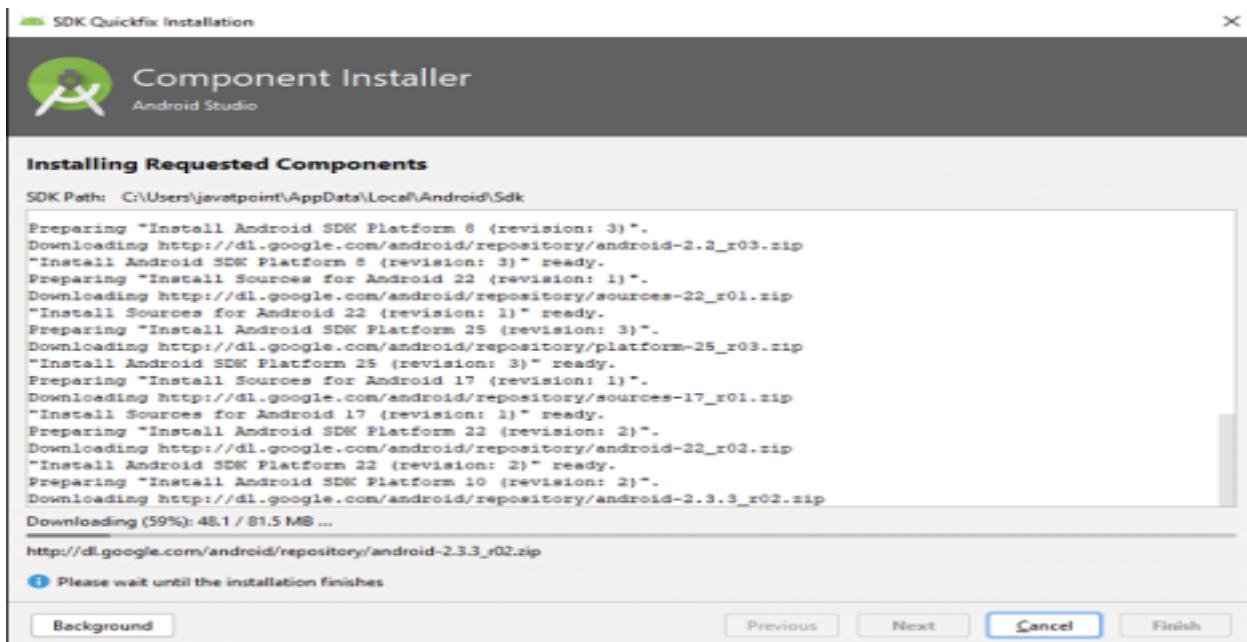
Step 7.2: When the download is complete, open the .exe file and run it. You will get the following dialog box.



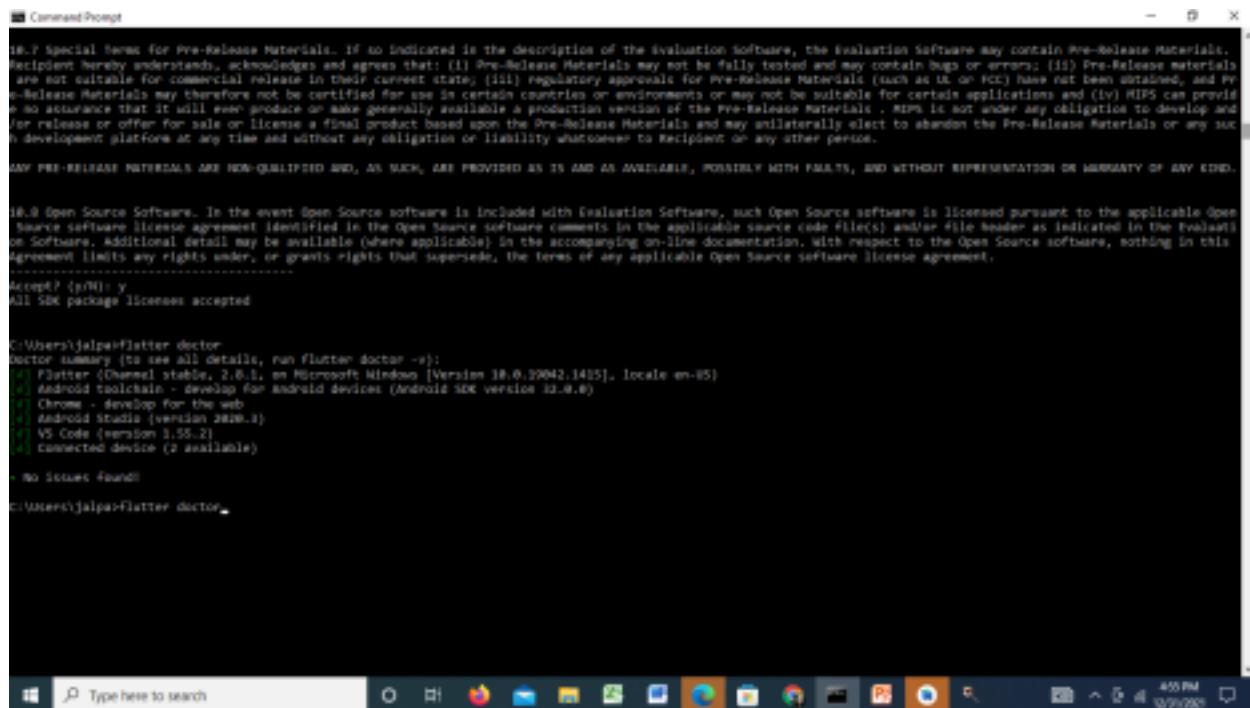
Step 7.3: Follow the steps of the installation wizard. Once the installation wizard completes, you will get the following screen.



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.



```
16.7 Special Terms for Pre-Release Materials. If so indicated in the description of the Evaluation Software, the Evaluation Software may contain Pre-Release Materials. Recipient hereby understands, acknowledges and agrees that: (i) Pre-Release Materials may not be fully tested and may contain bugs or errors; (ii) Pre-Release materials are not suitable for commercial release in their current state; (iii) regulatory approvals for Pre-Release Materials (such as UL or FCC) have not been obtained, and Pre-Release Materials may therefore not be certified for use in certain countries or environments or may not be suitable for certain applications and (iv) MIPs can provide no assurance that it will even produce or make generally available a production version of the Pre-Release Materials. MIPs is not under any obligation to develop and/or release or offer for sale or license a final product based upon the Pre-Release Materials and may unilaterally elect to abandon the Pre-Release Materials or any such development platform at any time and without any obligation or liability whatsoever to Recipient or any other person.
```

ANY PRE-RELEASE MATERIALS ARE NON-QUALIFIED AND, AS SUCH, ARE PROVIDED AS IS AND AS AVAILABLE, POSSIBLY WITH FAULTS, AND WITHOUT REPRESENTATION OR WARRANTY OF ANY KIND.

16.8 Open Source Software. In the event Open Source software is included with Evaluation Software, such Open Source software is licensed pursuant to the applicable Open Source software license agreement identified in the Open Source software comments in the applicable source code file(s) and/or file header as indicated in the Evaluation Software. Additional detail may be available (where applicable) in the accompanying on-line documentation. With respect to the Open Source software, nothing in this Agreement limits any rights under, or grants rights that supersede, the terms of any applicable Open Source software license agreement.

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

C:\Users\jalpa\Flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[+ Flutter (Channel stable, 2.8.1, on Microsoft Windows [Version 10.0.19042.1415], locale en-US)
[+ Android toolchain - develop for Android devices (Android SDK version 32.0.0)
[+ Chrome - Develop for the web
[+ Android Studio (version 2020.3)
[+ VS Code (version 1.55.2)
[+] connected device (2 available)
• No issues found!
C:\Users\jalpa\Flutter doctor

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

CODE:

```
import 'package:flutter/material.dart';
void main() {
  runApp(const MyApp());
}
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Welcome to Flutter',
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Welcome to Flutter'),
        ),
        body: const Center(
          child: Text('Hello Sneha Utekar'),
        ),
      );
  }
}
```

OUTPUT:



CONCLUSION:

We have installed Flutter and Android Studio and also implemented the first “Hello World App” using flutter successfully.

MAD & PWA Lab

Journal

Experiment No.	02
Experiment Title.	To design Flutter UI by including common widgets.
Roll No.	66
Name	Sneha Utekar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	15

Experiment 2

AIM: To design Flutter UI by including common widgets.

THEORY:

1. SplashScreen (splashscreen.dart):

Description: This widget represents the splash screen of the application, displaying an animated logo (Olx) and a loading animation.

Widgets Used:

- **AnimatedTextKit:** Provides a set of pre-built animated text styles.
- **ColorizeAnimatedText:** A specific type of animated text that changes color dynamically.
- **Scaffold:** The top-level container for the widget hierarchy.

2. HomeScreen (home_screen.dart):

Description: The main screen of the application where users can search for items and view banners and categories.

Widgets Used:

Scaffold: The top-level container for the widget hierarchy.

AppBar: The app bar containing the location information and search bar.

TextField: Allows users to input search queries.

BannerWidget: A custom widget for displaying banners and categories.

bottomNavigationBar: Contains a sign-out button.

BannerWidget (banner_widget.dart):

Description: A custom widget representing a banner section with information about cars, animated text, and buttons for buying and selling.

Widgets Used:

- **Neumorphic:** A container with neumorphic (soft shadow) styling.
- **Container:** A box to hold other widgets with specified dimensions.
- **Padding:** Adds padding around its child widgets.
- **Column:** Arranges its children in a vertical column.
- **Row:** Arranges its children in a horizontal row.
- **Text:** Displays text with specified styling.
- **SizedBox:** Provides a box with a fixed size.

- **DefaultTextStyle:** Applies a default text style to its child.
- **AnimatedTextKit:** Displays a set of animated texts.
- **FadeAnimatedText:** A specific type of animated text with a fading effect.
- **NeomorphicButton :**A neuromorphic-styled button.

CODE:

splashscreen.dart:

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

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

  @override
  Widget build(BuildContext context) {
    const colorizeColors = [
      Colors.white,
      Colors.cyan,
    ];

    const colorizeTextStyle = TextStyle(
      fontSize: 50.0,
      fontFamily: 'Horizon',//need to add our own font
      fontWeight: FontWeight.bold,
    );
    return Scaffold(
      backgroundColor:Colors.cyan.shade900 ,
      body:Center(
        child: Column(
          mainAxisSize: MainAxisSize.min,
          children: [
            Image.asset('assets/images/cart.png',
              // color: Colors.white
            ),
            SizedBox(
              height:10,
            ),
            //need an animated text
            AnimatedTextKit(
```

```

        animatedTexts: [
          ColorizeAnimatedText(
            'Olx',
            textStyle: colorizeTextStyle,
            colors: colorizeColors,
          ),
        ],
        isRepeatingAnimation: true,
        onTap: () {
          print("Tap Event");
        },
      ),
    ],
  ),
),
);

}
}
}

```

home_screen.dart:

```

import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:location/location.dart';
import 'package:flutter_geocoder/geocoder.dart';
import 'package:olx_clone/screens/login_screen.dart';
import 'package:olx_clone/widgets/banner_widget.dart';

```

```
class HomeScreen extends StatefulWidget {
```

```
  static const String id = 'home-screen';
  final LocationData? locationData;
```

```
  HomeScreen({required this.locationData});
```

```

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}
```

```

class _HomeScreenState extends State<HomeScreen> {
  String address = 'India';

```

```
Future<String?> getAddress() async {
    // From coordinates
    final coordinates = Coordinates(widget.locationData?.latitude, widget.locationData?.longitude);
    var addresses = await Geocoder.local.findAddressesFromCoordinates(coordinates);
    var first = addresses.first;
    setState(() {
        address = first.addressLine ?? 'Address not found';
    });
    return first.addressLine;
}

@Override
void initState() {
    super.initState();

    getAddress();
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            backgroundColor: Colors.white,
            elevation: 0.0,
            automaticallyImplyLeading: false,
            title: InkWell(
                onTap: () {},
                child: Container(
                    width: MediaQuery.of(context).size.width,
                    child: Padding(
                        padding: const EdgeInsets.only(top: 8, bottom: 8),
                        child: Row(
                            children: [
                                Icon(CupertinoIcons.location_solid, color: Colors.black, size: 18,),
                                Flexible(
                                    child: Text(
                                        address,
                                        style: TextStyle(color: Colors.black, fontSize: 12, fontWeight: FontWeight.bold),
                                    ),
                                ),
                                Icon(Icons.arrow_downward_outlined, color: Colors.black, size: 18,),
                            ],
                        ),
                    ),
                ),
            ),
        ),
    );
}
```



```
        ),  
        ),  
        ],  
        ),  
    );  
}  
}
```

banner_widget.dart:

```
import 'package:animated_text_kit/animated_text_kit.dart';  
import 'package:flutter/material.dart';  
import 'package:flutter_neumorphic_plus/flutter_neumorphic.dart';  
  
class BannerWidget extends StatelessWidget {  
    @override  
    Widget build(BuildContext context) {  
        return Neumorphic(  
            child: Container(  
                width: MediaQuery.of(context).size.width,  
                height: MediaQuery.of(context).size.height * 0.25,  
                color: Colors.cyan.shade800,  
                child: Padding(  
                    padding: const EdgeInsets.all(10.0),  
                    child: Column(  
  
                        mainAxisAlignment: MainAxisAlignment.spaceBetween,  
                        children: [  
                            Padding(  
                                padding: const EdgeInsets.all(10.0),  
                                child: Row(  
                                    mainAxisAlignment: MainAxisAlignment.spaceBetween,  
                                    children: [  
                                        Column(  
                                            crossAxisAlignment: CrossAxisAlignment.start,  
                                            children: [  
                                                SizedBox(height: 10),  
                                                Text(  
                                                    'CARS',  
                                                    style: TextStyle(  
                                                        fontWeight: FontWeight.bold,  
                                                        color: Colors.white,  
                                                        letterSpacing: 1,  
                                                        fontSize: 18,  
                                                    ),  
                                            ],  
                                        ),  
                                    ],  
                                ),  
                            ),  
                        ],  
                    ),  
                ),  
            ),  
        );  
    }  
}
```



```

),
),

Row(
  mainAxisSize: MainAxisSize.min,
  children: [
    Expanded(child: NeumorphicButton(
      onPressed: (){},
      style: NeumorphicStyle(color: Colors.white),
      child: Text('Buy Car',textAlign: TextAlign.center,),
    ),),
    SizedBox(width:20,),
    Expanded(child: NeumorphicButton(
      onPressed: (){},
      style: NeumorphicStyle(color: Colors.white),
      child: Text('Sell Car',textAlign: TextAlign.center,),
    ),),
  ],
),
),
),
),
),
),
),
);
}
}
}

```

main.dart:

```

import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:olx_clone/screens/authentication/email_auth_screen.dart';
import 'package:olx_clone/screens/authentication/phoneauth_screen.dart';
import 'package:olx_clone/screens/home_screen.dart';
import 'package:olx_clone/screens/login_screen.dart';
import 'package:olx_clone/screens/splash_screen.dart';
import 'package:olx_clone/screens/location_screen.dart';
import 'dart:io';

```

```

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  Platform.isAndroid ?await Firebase.initializeApp(

```

```

options: const FirebaseOptions(
    apiKey: "AIzaSyDuFtAm59795zlrpp16kTJTbd1Lac95KkI",
    appId: "1:232516563209:android:d2dea587c60de17bac63d8",
    messagingSenderId:"232516563209" ,
    projectId: "olxclone-12618"
)
):await Firebase.initializeApp();
runApp(MyApp());
}

class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
return MaterialApp(
debugShowCheckedModeBanner: false,
theme: ThemeData(
primaryColor: Colors.cyan.shade900,
fontFamily: 'Horizon',
),
initialRoute: SplashScreen.id,
routes: {
SplashScreen.id: (context) => SplashScreen(),
LoginScreen.id: (context) => LoginScreen(),
PhoneAuthScreen.id: (context) => PhoneAuthScreen(),
LocationScreen.id: (context) => LocationScreen(),
HomeScreen.id: (context) => HomeScreen(locationData: null),
EmailAuthScreen.id: (context) => EmailAuthScreen(),
},
);
}

```

pubsec.yaml:

```

name: olx_clone
description: "A new Flutter project."

```

```
publish_to: 'none' # Remove this line if you wish to publish to pub.dev
```

```
version: 1.0.0+1
```

```
environment:
sdk: '>=3.2.5 <4.0.0'
```

```
dependencies:
flutter:
```

sdk: flutter

```
cupertino_icons: ^1.0.2
cloud_firestore: ^4.15.3
animated_text_kit: ^4.2.2
flutter_signin_button: ^2.0.0
firebase_core: ^2.24.2
firebase_auth: ^4.16.0
legacy_progress_dialog: ^0.1.2
location: ^5.0.3
flutter_geocoder: ^0.2.2-nullsafety
google_sign_in: ^6.2.1
email_validator: ^2.1.17
flutter_neumorphic_plus: ^3.3.0
```

dev_dependencies:

```
flutter_test:
  sdk: flutter
```

```
flutter_lints: ^2.0.0
```

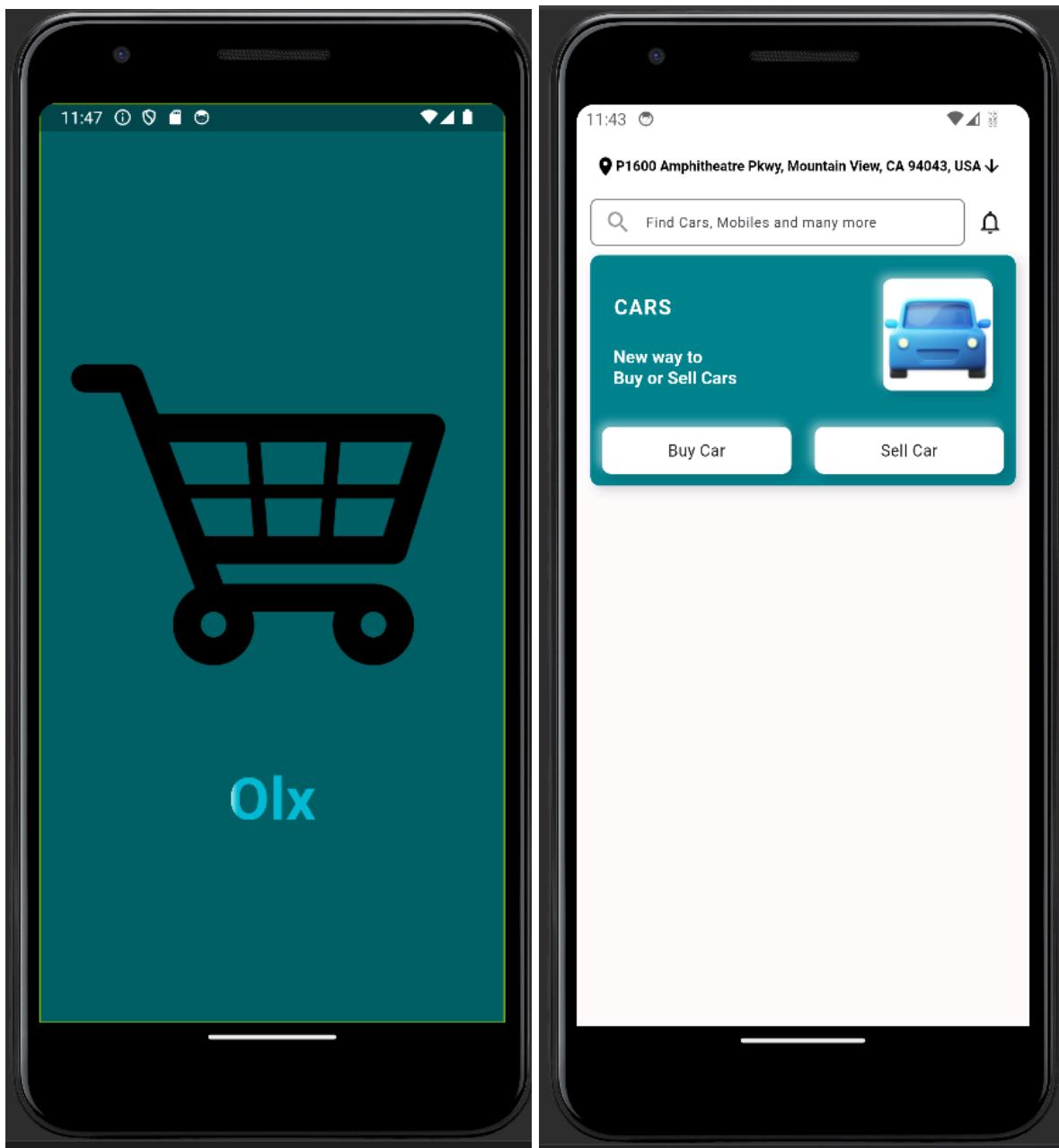
flutter:

```
uses-material-design: true
```

assets:

```
- assets/images/cart.png
- assets/images/map.png
```

OUTPUT:



CONCLUSION:

We have designed the FlutterUi using the common widgets.

MAD & PWA Lab

Journal

Experiment No.	03
Experiment Title.	To include icons, images, fonts in Flutter app
Roll No.	66
Name	Sneha Utekar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	15

Experiment 3

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

THEORY:

A) Icons:

Icons play a crucial role in conveying information and improving the user interface. Flutter provides the `Icon` widget to display various icons from different icon libraries. Commonly used icon libraries include `MaterialIcons`, `FontAwesome`, and `CupertinoIcons`. To include icons:

1. Import the necessary icon library in your Dart file:

```
import 'package:flutter/material.dart'; // Example for Material Icons
```

2. Use the `Icon` widget to display an icon:

```
Icon(Icons.favorite);
```

B) Images:

Images are essential for creating visually appealing interfaces. Flutter supports various image formats, such as JPEG, PNG, and GIF. To include images:

1. Place your image assets in the `assets` directory of your Flutter project.

2. Update your `pubspec.yaml` file to include the assets:

pubsec.yaml:

```
flutter:
  assets:
    - assets/my_image.jpg
```

3. Use the `Image` widget to display an image:

```
Image.asset('assets/my_image.jpg');
```

C) Fonts:

Custom fonts can give your app a unique and distinctive look. To include custom fonts:

1. Place your font files (e.g., TTF or OTF) in the `fonts` directory of your Flutter project.

2. Update your `pubspec.yaml` file to include the fonts:

```
flutter:
  fonts:
    - family: MyCustomFont
      fonts:
```

- asset: fonts/my_custom_font.ttf

3. Use the custom font in your text widget:

```
Text(
  'Hello, Flutter!',
  style: TextStyle(
    fontFamily: 'MyCustomFont',
  ),
);
```

CODE:

location_screen.dart:

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:location/location.dart';
import 'package:olx_clone/screens/home_screen.dart';
import 'package:olx_clone/screens/login_screen.dart';

class LocationScreen extends StatefulWidget {
  static const String id = "location-screen";

  @override
  State<LocationScreen> createState() => _LocationScreenState();
}

class _LocationScreenState extends State<LocationScreen> {
  bool _loading=false;
  Location location = new Location();

  late bool _serviceEnabled;
  late PermissionStatus _permissionGranted;
  late LocationData _locationData;

  Future<LocationData?> getLocation() async {
    _serviceEnabled = await location.serviceEnabled();
    if (!_serviceEnabled) {
      _serviceEnabled = await location.requestService();
      if (!_serviceEnabled) {
        return null;
      }
    }
}
```

```
_permissionGranted = await location.hasPermission();
if (_permissionGranted == PermissionStatus.denied) {
    _permissionGranted = await location.requestPermission();
    if (_permissionGranted != PermissionStatus.granted) {
        return null;
    }
}

_locationData = await location.getLocation();
print(_locationData);

return _locationData;
}

@Override

Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: Colors.white,
        body: Column(
            children: [
                Image.asset('assets/images/map.png', height: 350),
                SizedBox(height: 20),
                Text(
                    'Where do you want \nto buy/sell products?',
                    textAlign: TextAlign.center,
                    style: TextStyle(fontWeight: FontWeight.bold, fontSize: 25),
                ),
                SizedBox(
                    height: 10,
                ),
                Text(
                    'To fully enjoy all that we have to offer,\nwe need to know where to look for it',
                    textAlign: TextAlign.center,
                    style: TextStyle(fontSize: 12),
                ),
                SizedBox(
                    height: 30,
                ),
                Padding(
                    padding: const EdgeInsets.only(left: 20, right: 20, bottom: 10),
                    child: Row(
                        children: [

```

```
Expanded(  
    child: _loading ? Center(child: CircularProgressIndicator(  
        valueColor: AlwaysStoppedAnimation<Color>(Theme.of(context).primaryColor),  
    )),  
    ElevatedButton.icon(  
        style: ButtonStyle(  
            backgroundColor: MaterialStateProperty.all<Color>(  
                Theme.of(context).primaryColor,  
            ),  
            ),  
        icon: Icon(  
            CupertinoIcons.location_fill,  
            color: Colors.white,  
        ),  
        label: Padding(  
            padding: const EdgeInsets.only(top: 15, bottom: 15),  
            child: Text('Around me', style: TextStyle(color: Colors.white, fontWeight:  
                FontWeight.bold),),  
        ),  
        onPressed: () {  
            setState(() {  
                _loading=true;  
            });  
            getLocation().then((value) {  
                print(_locationData.latitude);  
  
                if (value != null) {  
                    Navigator.pushReplacement(  
                        context,  
                        MaterialPageRoute(  
                            builder: (BuildContext context)=>  
                                HomeScreen(locationData: _locationData),  
                        ),  
                    );  
                }  
            });  
        },  
        ),  
    ],  
,  
),  
TextButton(  
    onPressed: () {},  
)
```

```

        child: Text(
            'Set location Manually',
            style: TextStyle(
                fontWeight: FontWeight.bold,
                color: Colors.black,
                fontSize: 18,
                decoration: TextDecoration.underline,
            ),
        ),
    ),
),
],
),
);
}
}
}

```

main.dart:

```

import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:olx_clone/screens/authentication/email_auth_screen.dart';
import 'package:olx_clone/screens/authentication/phoneauth_screen.dart';
import 'package:olx_clone/screens/home_screen.dart';
import 'package:olx_clone/screens/login_screen.dart';
import 'package:olx_clone/screens/splash_screen.dart';
import 'package:olx_clone/screens/location_screen.dart';
import 'dart:io';

void main() async {
    WidgetsFlutterBinding.ensureInitialized();
    Platform.isAndroid ?await Firebase.initializeApp(
        options: const FirebaseOptions(
            apiKey: "AIzaSyDuFtAm59795zlrpp16kTJTbd1Lac95KkI",
            appId: "1:232516563209:android:d2dea587c60de17bac63d8",
            messagingSenderId:"232516563209" ,
            projectId: "olxclone-12618"
        )
    ):await Firebase.initializeApp();
    runApp(MyApp());
}

class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return MaterialApp(

```

```
debugShowCheckedModeBanner: false,  
theme: ThemeData(  
    primaryColor: Colors.cyan.shade900,  
    fontFamily: 'Horizon',  
,  
initialRoute: SplashScreen.id,  
routes: {  
    SplashScreen.id: (context) => SplashScreen(),  
    LoginScreen.id: (context) => LoginScreen(),  
    PhoneAuthScreen.id: (context) => PhoneAuthScreen(),  
    LocationScreen.id: (context) => LocationScreen(),  
    HomeScreen.id: (context) => HomeScreen(locationData: null),  
    EmailAuthScreen.id: (context) => EmailAuthScreen(),  
,  
,  
);
```

pubsec.yaml:

```
name: olx_clone  
description: "A new Flutter project."
```

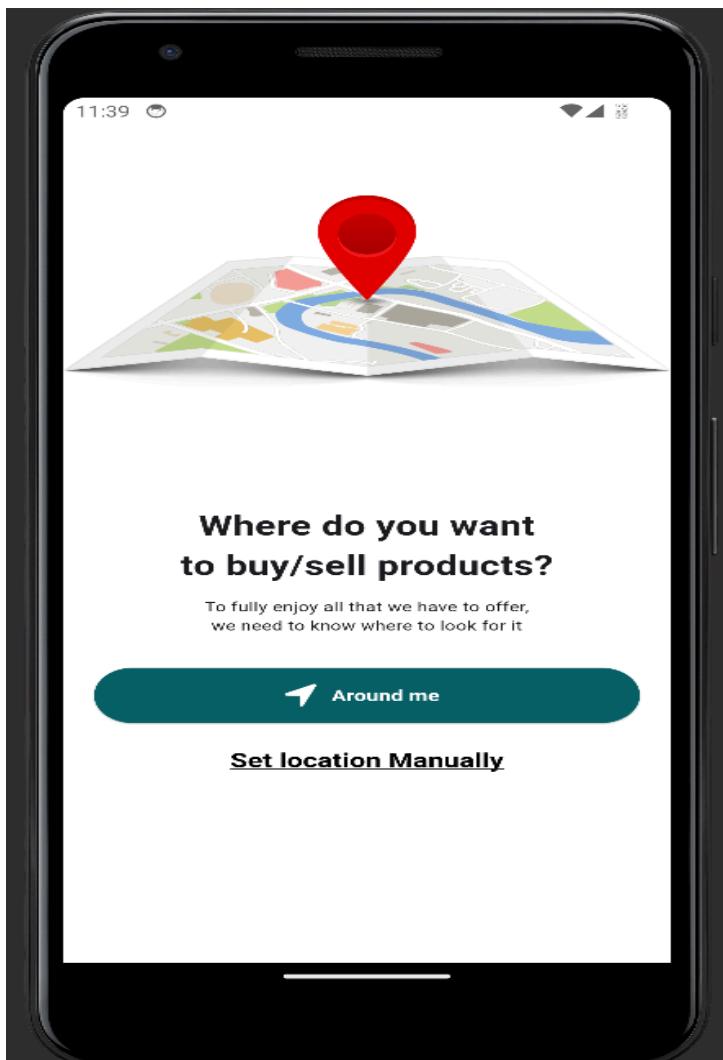
```
publish_to: 'none' # Remove this line if you wish to publish to pub.dev
```

```
version: 1.0.0+1  
environment:  
  sdk: '>=3.2.5 <4.0.0'  
dependencies:  
  flutter:  
    sdk: flutter  
    cupertino_icons: ^1.0.2  
    cloud_firestore: ^4.15.3  
    animated_text_kit: ^4.2.2  
    flutter_signin_button: ^2.0.0  
    firebase_core: ^2.24.2  
    firebase_auth: ^4.16.0  
    legacy_progress_dialog: ^0.1.2  
  location: ^5.0.3  
  flutter_geocoder: ^0.2.2-nullsafety  
  google_sign_in: ^6.2.1  
  email_validator: ^2.1.17  
  flutter_neumorphic_plus: ^3.3.0
```

```
dev_dependencies:  
  flutter_test:
```

```
sdk: flutter  
flutter_lints: ^2.0.0  
flutter:  
uses-material-design: true  
assets:  
- assets/images/cart.png  
- assets/images/map.png
```

OUTPUT:



CONCLUSION:

We have understood how to include the icons, images and fonts in flutter UI. We have implemented these things in our UI.

MAD & PWA Lab

Journal

Experiment No.	04
Experiment Title.	To create an interactive Form using form widget
Roll No.	66
Name	Sneha Utekar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	15

Experiment 4

AIM: To create an interactive Form using form widget

THEORY:

LoginScreen Widget:

Description: Represents the login screen with OLX branding and options to log in using various methods.

Used in: main.dart

Key Widgets:

1. Scaffold
2. Column
3. Container
4. Image.asset
5. Text
6. AuthUi

EmailAuthScreen Widget:

Description: Allows the user to log in or register using email and password.

Used in: login_screen.dart

Key Widgets:

1. Scaffold
2. AppBar
3. Form
4. TextFormField
5. DropdownButtonFormField
6. Checkbox
7. Radio
8. ElevatedButton

GoogleAuthentication Class:

Description: Handles Google authentication and sign-in.

Used in: login_screen.dart

Key Widgets:

SnackBar

PhoneAuthScreen Widget:

Description: Allows the user to log in using their phone number and initiates phone number verification.

Used in: login_screen.dart

Key Widgets:

- 1.Scaffold
- 2.AppBar
- 3.Padding
- 4.AbsorbPointer
- 5.ElevatedButton

PhoneAuthentication Class:

Description: Handles phone number authentication and verification.

Used in: phoneAuth.dart

Key Widgets:

ProgressDialog

OTPScreen Widget:

Description: Allows the user to enter the OTP received via SMS for phone number verification.

Used in: phoneAuth.dart

Key Widgets:

- 1.Scaffold
- 2.AppBar
- 3.Column
- 4.TextFormField
- 5.LinearProgressIndicator

AuthUi Widget:

Description: Displays various authentication options like phone, Google, Facebook, and email.

Used in: login_screen.dart

Key Widgets:

1. Center
2. Column
3. ElevatedButton
4. SignInButton
5. InkWell

CODE:**login_screen.dart:**

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:olx_clone/screens/location_screen.dart';
import 'package:olx_clone/widgets/auth_ui.dart';

class LoginScreen extends StatelessWidget {
  static const String id='login-screen';
  @override

  Widget build(BuildContext context) {

    return Scaffold(
      backgroundColor: Colors.cyan.shade900,
      body:Column(
        children: [
          Expanded(
            child:Container(
              width:MediaQuery.of(context).size.width,//to get device width
              color: Colors.white,
              child: Column(
                children: [
                  SizedBox(height: 50),
                  Image.asset('assets/images/cart.png',
                  color: Colors.cyan.shade900,
                  height: 100,),
                  SizedBox(height: 5,),
                  Text('Olx',style: TextStyle(
                    fontFamily: 'Horizon',
                    fontSize: 50,
                    fontWeight:FontWeight.bold,
                    color: Colors.cyan.shade900,
                  ),),
                ],
              ),
            ),
          ),
        ],
      ),
    );
  }
}
```

```

        ],
        ),
        )),
      Expanded(child:Container(
        child: AuthUi(),
      ),),
      Padding(padding: const EdgeInsets.all(8.0),
      child:Text('If you continue,you are accepting\nTerms and Conditions and Privacy Policy',
      textAlign: TextAlign.center,style:TextStyle(color: Colors.white,fontSize: 10 ),),
      ),
      ],
    ),
  );
}
}
}

```

emailAuthScreen.dart:

```

import 'package:email_validator/email_validator.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:olx_clone/services/emailAuth_service.dart';

class EmailAuthScreen extends StatefulWidget {
  static const String id = 'emailAuth-Screen';

  @override
  State<EmailAuthScreen> createState() => _EmailAuthScreenState();
}

class _EmailAuthScreenState extends State<EmailAuthScreen> {
  final _formKey = GlobalKey<FormState>();
  bool _validate = false;
  bool _login = true;
  bool _loading = false;
  var _emailController = TextEditingController();
  var _passwordController = TextEditingController();
  EmailAuthentication _service = EmailAuthentication();

  _validateEmail() {
    if (_formKey.currentState!.validate()) {
      setState(() {
        _validate = false;
      });
    }
  }
}

```

```
_loading = true;
});
_service.getAdminCredential(
context: context,
isLog: _login,
password: _passwordController.text,
email: _emailController.text,
).then((value) {
setState(() {
_loading = false;
});
});
}
}

@Override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
elevation: 1,
iconTheme: IconThemeData(color: Colors.black),
backgroundColor: Colors.white,
title: Text(
'Login',
style: TextStyle(color: Colors.black),
),
),
),
body: Form(
key: _formKey,
child: Padding(
padding: const EdgeInsets.all(20.0),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
SizedBox(height: 40),
CircleAvatar(
radius: 30,
backgroundColor: Colors.red.shade200,
child: Icon(
CupertinoIcons.person_alt_circle,
color: Colors.red,
size: 60,
),
),
],
),
),
),
);}
```

```
SizedBox(height: 12),
Text(
  'Enter your ${_login ? 'Login' : 'Register'}',
  style: TextStyle(fontSize: 30, fontWeight: FontWeight.bold),
),
SizedBox(height: 10),
Text(
  'Enter Email and Password to ${_login ? 'Login' : 'Register'}',
  style: TextStyle(color: Colors.grey),
),
SizedBox(
  height: 10,
),
TextField(
  controller: _emailController,
  validator: (value) {
    final bool isValid =
      EmailValidator.validate(_emailController.text);
    if (value == null || value.isEmpty) {
      return 'Enter Email';
    }
    if (value.isNotEmpty && isValid == false) {
      return 'Enter Valid Email';
    }
    return null;
  },
  decoration: InputDecoration(
    contentPadding: EdgeInsets.only(left: 10),
    labelText: 'Email',
    filled: true,
    fillColor: Colors.grey.shade300,
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(4),
    ),
  ),
),
SizedBox(
  height: 10,
),
TextField(
  controller: _passwordController,
  decoration: InputDecoration(
    suffixIcon: _validate
      ? IconButton(

```

```
icon: Icon(Icons.clear),
onPressed: () {
    _passwordController.clear();
    setState(() {
        _validate = false;
    });
},
),
: null,
contentPadding: EdgeInsets.only(left: 10),
labelText: 'Password',
filled: true,
fillColor: Colors.grey.shade300,
border: OutlineInputBorder(
    borderRadius: BorderRadius.circular(4),
),
),
),
onChanged: (value) {
if (_emailController.text.isNotEmpty) {
    if (value.length > 3) {
        setState(() {
            _validate = true;
        });
    }
} else {
    setState(() {
        _validate = false;
    });
}
},
),
),
Row(
children: [
Text(_login ? 'New Account?' : 'Already has an Account?'),
TextButton(
child: Text(
    _login ? 'Register' : 'Login',
    style: TextStyle(
        color: Theme.of(context).primaryColor,
        fontWeight: FontWeight.bold,
    ),
),
),
),
onPressed: () {
    setState(() {
```



```
    ),  
    ),  
    ),  
);  
}  
}
```

googleAuth.dart:

```
import 'package:firebase_auth/firebase_auth.dart';  
import 'package:flutter/material.dart';  
import 'package:google_sign_in/google_sign_in.dart';  
  
class GoogleAuthentication {  
  static SnackBar customSnackBar({required String content}) {  
    return SnackBar(  
      backgroundColor: Colors.black,  
      content: Text(  
        content,  
        style: TextStyle(color: Colors.redAccent, letterSpacing: 0.5),  
      ),  
    );  
  }  
  
  static Future<User?> signInWithGoogle({required BuildContext context}) async {  
    FirebaseAuth auth = FirebaseAuth.instance;  
    User? user;  
  
    final GoogleSignIn googleSignIn = GoogleSignIn();  
  
    final GoogleSignInAccount? googleSignInAccount =  
    await googleSignIn.signIn();  
  
    if (googleSignInAccount != null) {  
      final GoogleSignInAuthentication googleSignInAuthentication =  
      await googleSignInAccount.authentication;  
  
      final AuthCredential credential = GoogleAuthProvider.credential(  
        accessToken: googleSignInAuthentication.accessToken,  
        idToken: googleSignInAuthentication.idToken,  
      );  
  
      try {  
        final UserCredential userCredential =  
        await auth.signInWithCredential(credential);  
      } catch (e) {  
        print(e);  
      }  
    }  
  }  
}
```

```

        user = userCredential.user;
    } on FirebaseAuthException catch (e) {
        if (e.code == 'account-exists-with-different-credential') {
            ScaffoldMessenger.of(context).showSnackBar(
                customSnackBar(
                    content: 'account-exists-with-different-credential',
                ),
            );
        } else if (e.code == 'invalid-credential') {
            ScaffoldMessenger.of(context).showSnackBar(
                customSnackBar(
                    content: 'invalid-credential',
                ),
            );
        }
    } catch (e) {
        ScaffoldMessenger.of(context).showSnackBar(
            customSnackBar(
                content: 'Login Failed',
            ),
        );
    }
}

return user;
}
}

```

phoneAuth.dart:

```

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:legacy_progress_dialog/legacy_progress_dialog.dart';
import 'package:olx_clone/services/phoneauth_service.dart';

class PhoneAuthScreen extends StatefulWidget {
    static const String id = 'phone-auth-screen';

    @override
    _PhoneAuthScreenState createState() => _PhoneAuthScreenState();
}

class _PhoneAuthScreenState extends State<PhoneAuthScreen> {

```

```
bool validate = false;  
TextEditingController countryCodeController = TextEditingController(text: '+91');  
TextEditingController phoneNumberController = TextEditingController();  
String counterText = '0';
```

```
PhoneAuthService _service= PhoneAuthService();
```

```
@override  
Widget build(BuildContext context) {  
  
  //Create an instance of ProgressDialog  
  ProgressDialog progressDialog = ProgressDialog(  
    context: context,  
    backgroundColor: Colors.white,  
    textColor: Colors.black,  
    loadingText:'Please wait' ,  
    progressIndicatorColor:Theme.of(context).primaryColor ,  
  
  );  
  
  return Scaffold(  
    appBar: AppBar(  
      elevation: 1,  
      backgroundColor: Colors.white,  
      iconTheme: IconThemeData(  
        color: Colors.black,  
      ),  
      title: Text('Login', style: TextStyle(color: Colors.black)),  
    ),  
    body: Padding(  
      padding: const EdgeInsets.all(20.0),  
      child: Column(  
        mainAxisAlignment: MainAxisAlignment.start,  
        children: [  
          SizedBox(height: 40),  
          CircleAvatar(  
            radius: 30,  
            backgroundColor: Colors.red.shade200,  
            child: Icon(  
              Icons.phone,
```

```
CupertinoIcons.person_alt_circle,  
color: Colors.red,  
size: 60,  
,  
,  
SizedBox(height: 12),  
Text(  
'Enter your Phone',  
style: TextStyle(fontSize: 30, fontWeight: FontWeight.bold),  
,  
SizedBox(height: 10),  
Text(  
'We will send the confirmation code to your phone',  
style: TextStyle(color: Colors.grey),  
,  
Row(  
children: [  
Expanded(  
flex: 1,  
child: TextFormField(  
controller: countryCodeController,  
enabled: false,  
style: TextStyle(color: Colors.black), // Set the text color here  
decoration: InputDecoration(  
labelText: 'Country',  
,  
,  
,  
),  
SizedBox(width: 10),  
Expanded(  
flex: 3,  
child: TextFormField(  
onChanged: (value) {  
setState(() {  
counterText = value.length.toString();  
if (value.length == 10) {  
validate = true;  
}  
if (value.length < 10) {  
setState(() {  
validate = false;  
});  
}  
});  
});  
});
```

```
},
 autofocus: true,
 maxLength: 10,
 keyboardType: TextInputType.phone,
 controller: phoneNumberController,
 decoration: InputDecoration(
  contentPadding: EdgeInsets.only(bottom: 22, top: 22),
  counterText: '$counterText / 10',
  counterStyle: TextStyle(fontSize: 10),
  labelText: 'Number',
  hintText: 'Enter your phone number',
  hintStyle: TextStyle(fontSize: 10, color: Colors.grey),
 ),
),
),
],
),
],
),
),
),
bottomNavigationBar: SafeArea(
 child: Padding(
 padding: const EdgeInsets.all(12.0),
 child: AbsorbPointer(
 absorbing: validate ? false : true,
 child: ElevatedButton(
 style: ButtonStyle(
 backgroundColor: validate
 ? MaterialStateProperty.all(Theme.of(context).primaryColor)
 : MaterialStateProperty.all(Colors.grey),
 ),
 onPressed: () {
 String number = '${countryCodeController.text}${phoneNumberController.text}';
 progressDialog.show();
 _service.verifyPhoneNumber(context, number);
 },
),
child: Padding(
 padding: const EdgeInsets.all(12.0),
 child: Text(
 'Next',
 style: TextStyle(color: Colors.white, fontWeight: FontWeight.bold),
),
```

```

),
),
),
),
),
),
),
);
}
}
}
```

otpscreen.dart:

```

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/services.dart';
import 'package:olx_clone/screens/authentication/phoneauth_screen.dart';
import 'package:olx_clone/screens/location_screen.dart';
import 'package:olx_clone/services/phoneauth_service.dart';

class OTPScreen extends StatefulWidget {
final String number,verId;

OTPScreen({this.number="",this.verId=""});

@Override
State<OTPScreen> createState() => _OTPScreenState();
}

class _OTPScreenState extends State<OTPScreen> {
bool _loading=false;
String error="";

PhoneAuthService _services=PhoneAuthService();
var _text1 = TextEditingController();
var _text2 = TextEditingController();
var _text3 = TextEditingController();
var _text4 = TextEditingController();
var _text5 = TextEditingController();
var _text6 = TextEditingController();

Future<void>phoneCredential(BuildContext context,String otp,)async{
FirebaseAuth _auth=FirebaseAuth.instance;
try{
PhoneAuthCredential credential = PhoneAuthProvider.credential(
```

```
verificationId:widget.verId, smsCode:otp );
final User? user=(await _auth.signInWithCredential(credential)).user;
if(user!=null){
    _services.addUser(context,user.uid);

}else {
    print("Login Failed");
    if (mounted) {
        setState(() {
            error = "Login Failed";
        });
    }
}
}catch(e){
    print(e.toString());
    if(mounted){
        setState(() {
            error="Invalid OTP";
        });
    }
}
}
@Override
Widget build(BuildContext context) {
    final node=FocusScope.of(context);
    return Scaffold(
        appBar: AppBar(
            elevation: 1,
            backgroundColor: Colors.white,
            title:Text('Login',style: TextStyle(color:Colors.black),),
            automaticallyImplyLeading:false ,
        ),
        body:Padding(
            padding: const EdgeInsets.only(left:20,right:20),
            child:Column(
                mainAxisAlignment:CrossAxisAlignment.start ,
                children: [
                    SizedBox(height: 40,),
                    CircleAvatar(
                        radius: 30,
                        backgroundColor: Colors.red.shade200,
                        child:Icon(CupertinoIcons.person_alt_circle,
```

```
size:60,  
color: Colors.red,  
,  
,  
SizedBox(height: 10,),  
Text(  
"Welcome Back",  
style:TextStyle(fontWeight: FontWeight.bold,  
fontSize: 25,  
,  
,  
),  
SizedBox(height: 10,),  
Row(  
children: [  
Expanded(  
child: RichText(  
text:TextSpan(text: 'We sent a 6-digit code to ',  
style: TextStyle(color: Colors.grey,  
fontSize: 12),  
children: [  
TextSpan(  
text:widget.number,  
style: TextStyle(fontWeight: FontWeight.bold,  
color: Colors.black,  
fontSize: 12)  
,  
]),  
],),  
),  
InkWell(  
onTap:(){  
Navigator.push(  
context, MaterialPageRoute(builder: (context) => PhoneAuthScreen(),));  
},  
child: Icon(Icons.edit)),  
],  
),  
SizedBox(height:12,),  
Row(  
children: [  
Expanded(  
)
```

```
        child: TextFormField(
            textAlign: TextAlign.center,
            controller: _text1 ,
            keyboardType: TextInputType.number,
           textInputAction: TextInputAction.next,
            decoration: InputDecoration(
                border:OutlineInputBorder()
            ),
            onChanged:(value){
                if(value.length==1){
                    node.nextFocus();
                }
            },
        ),
        SizedBox(width: 10,),
        Expanded(
            child: TextFormField(
                textAlign: TextAlign.center,
                controller: _text2 ,
                keyboardType: TextInputType.number,
               textInputAction: TextInputAction.next,
                decoration: InputDecoration(
                    border:OutlineInputBorder()
                ),
                onChanged:(value){
                    if(value.length==1){
                        node.nextFocus();
                    }
                },
            ),
        ),
        SizedBox(width: 10,),
        Expanded(
            child: TextFormField(
                textAlign: TextAlign.center,
                controller: _text3,
                keyboardType: TextInputType.number,
               textInputAction: TextInputAction.next,
                decoration: InputDecoration(
                    border:OutlineInputBorder()
                ),
                onChanged:(value){
                    if(value.length==1){
```

```
        node.nextFocus());
    },
},
),
),
SizedBox(width: 10,),
Expanded(
child: TextFormField(
textAlign: TextAlign.center,
controller: _text4 ,
keyboardType: TextInputType.number,
textInputAction: TextInputAction.next,
decoration: InputDecoration(
border: OutlineInputBorder()
),
onChanged:(value){
if(value.length==1){
node.nextFocus();
}
},
),
),
SizedBox(width: 10,),
Expanded(
child: TextFormField(
textAlign: TextAlign.center,
controller: _text5 ,
keyboardType: TextInputType.number,
textInputAction: TextInputAction.next,
decoration: InputDecoration(
border: OutlineInputBorder()
),
onChanged:(value){
if(value.length==1){
node.nextFocus();
}
},
),
),
),
SizedBox(width: 10,),
Expanded(
child: TextFormField(
textAlign: TextAlign.center,
controller: _text6 ,
```

```
        keyboardType: TextInputType.number,
       textInputAction: TextInputAction.next,
        decoration: InputDecoration(
            border:OutlineInputBorder(
                )
            ),
        onChanged:(value){
            if(value.length==1){
                if (_text1.text.length==1){
                    if (_text2.text.length==1){
                        if (_text3.text.length==1){
                            if (_text4.text.length==1){
                                if (_text5.text.length==1){
                                    String _otp="${_text1.text}${_text2.text}"
                                    '${_text3.text}${_text4.text}${_text5.text}${_text6.text}';
                                    setState(() {
                                        _loading=true;
                                    });
                                    //login
                                    phoneCredential(context, _otp);
                                }
                            }
                        }
                    }
                }
            }
        }
    }

}else{
    setState(() {
        _loading=false;
    });
},
),
),
],
),
SizedBox(height: 18,),
if(_loading)
Align(
    alignment: Alignment.center,
```

```

        child: SizedBox(
            width: 50,
            child: LinearProgressIndicator(
                backgroundColor: Colors.grey.shade200 ,
                valueColor: AlwaysStoppedAnimation<Color>(Theme.of(context).primaryColor) ,
            ),
        ),
    ),
),
SizedBox(height: 18,),
Text(error,style: TextStyle(color: Colors.red,fontSize: 12),)

],
),
),
);
}
}

```

AuthUI.dart:

```

import 'package:flutter/material.dart';
import 'package:flutter_signin_button/button_list.dart';
import 'package:flutter_signin_button/button_view.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:olx_clone/screens/authentication/google_auth.dart';
import 'package:olx_clone/screens/authentication/phoneauth_screen.dart';
import 'package:olx_clone/screens/authentication/email_auth_screen.dart';
import 'package:olx_clone/services/phoneauth_service.dart';

```

```

class AuthUi extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return Center(
            child: Column(
                mainAxisAlignment: MainAxisAlignment.min,
                children: [
                    SizedBox(
                        width: 220,
                        child: ElevatedButton(
                            style: ElevatedButton.styleFrom(
                                primary: Colors.white,
                                shape: new RoundedRectangleBorder(
                                    borderRadius: new BorderRadius.circular(3.0),
                            ),

```

```
),
onPressed: () {
  Navigator.pushNamed(context, PhoneAuthScreen.id);
},
child: Row(
  children: [
    Icon(Icons.phone_android_outlined, color: Colors.black),
    SizedBox(width: 8),
    Text('Continue with Phone', style: TextStyle(color: Colors.black)),
  ],
),
),
),
),
),
SignInButton(
  Buttons.Google,
  text: ('Continue with Google'),
  onPressed: () async {
    User? user = await GoogleAuthentication.signInWithGoogle(context: context);
    if (user != null) {
      PhoneAuthService _authentication = PhoneAuthService();
      _authentication.addUser(context, user.uid);
    }
  },
),
SignInButton(
  Buttons.FacebookNew,
  text: ('Continue with Facebook'),
  onPressed: () {},
),
InkWell(
  onTap: () {
    Navigator.pushNamed(context, EmailAuthScreen.id);
  },
),
child: Container(
  child: Text(
    'Login with Email',
    style: TextStyle(
      color: Colors.white,
      fontWeight: FontWeight.bold,
      fontSize: 18,
    ),
  ),
),
decoration: BoxDecoration(
  border: Border(bottom: BorderSide(color: Colors.white)),
),
```

```

    ),
    ),
    ),
    ],
),
);
}
}
```

main.dart:

```

import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:olx_clone/screens/authentication/email_auth_screen.dart';
import 'package:olx_clone/screens/authentication/phoneauth_screen.dart';
import 'package:olx_clone/screens/home_screen.dart';
import 'package:olx_clone/screens/login_screen.dart';
import 'package:olx_clone/screens/splash_screen.dart';
import 'package:olx_clone/screens/location_screen.dart';
import 'dart:io';
```

```

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  Platform.isAndroid ?await Firebase.initializeApp(
    options: const FirebaseOptions(
      apiKey: "AIzaSyDuFtAm59795zlrpp16kTJTbd1Lac95KkI",
      appId: "1:232516563209:android:d2dea587c60de17bac63d8",
      messagingSenderId:"232516563209" ,
      projectId: "olxclone-12618"
    )
  ):await Firebase.initializeApp();
  runApp(MyApp());
}
```

```

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primaryColor: Colors.cyan.shade900,
        fontFamily: 'Horizon',
```

```
        ),  
        initialRoute: SplashScreen.id,  
        routes: {  
            SplashScreen.id: (context) => SplashScreen(),  
            LoginScreen.id: (context) => LoginScreen(),  
            PhoneAuthScreen.id: (context) => PhoneAuthScreen(),  
            LocationScreen.id: (context) => LocationScreen(),  
            HomeScreen.id: (context) => HomeScreen(locationData: null),  
            EmailAuthScreen.id: (context) => EmailAuthScreen(),  
        },  
    );
```

```
pubsec.yaml:  
name: olx_clone  
description: "A new Flutter project."  
  
publish_to: 'none' # Remove this line if you wish to publish to pub.dev  
  
version: 1.0.0+1  
  
environment:  
  sdk: '>=3.2.5 <4.0.0'  
  
dependencies:  
  flutter:  
    sdk: flutter  
  
  cupertino_icons: ^1.0.2  
  cloud_firestore: ^4.15.3  
  animated_text_kit: ^4.2.2  
  flutter_signin_button: ^2.0.0  
  firebase_core: ^2.24.2  
  firebase_auth: ^4.16.0  
  legacy_progress_dialog: ^0.1.2  
  location: ^5.0.3  
  flutter_geocoder: ^0.2.2-nullsafety  
  google_sign_in: ^6.2.1  
  email_validator: ^2.1.17  
  flutter_neumorphic_plus: ^3.3.0  
  
dev_dependencies:  
  flutter_test:
```

sdk: flutter

flutter_lints: ^2.0.0

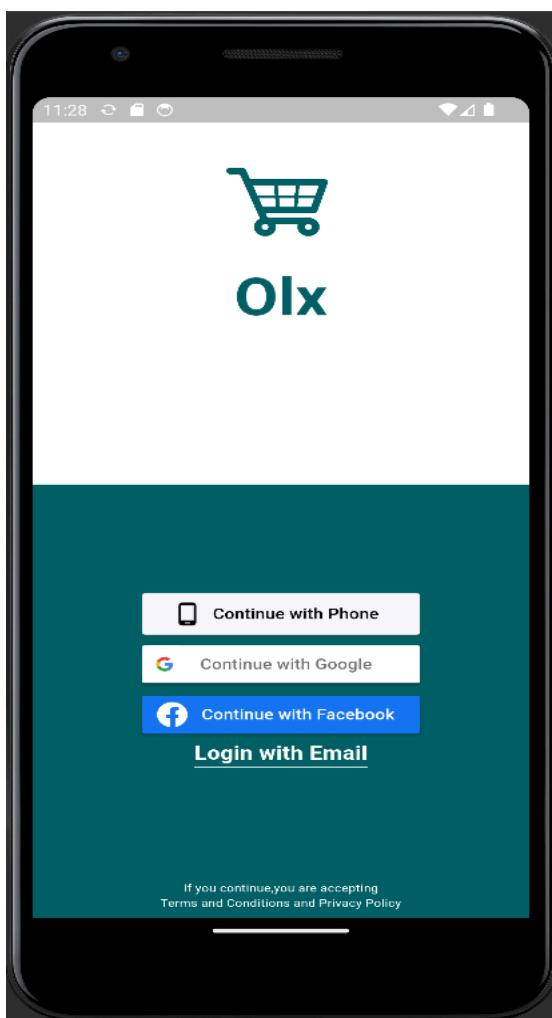
flutter:

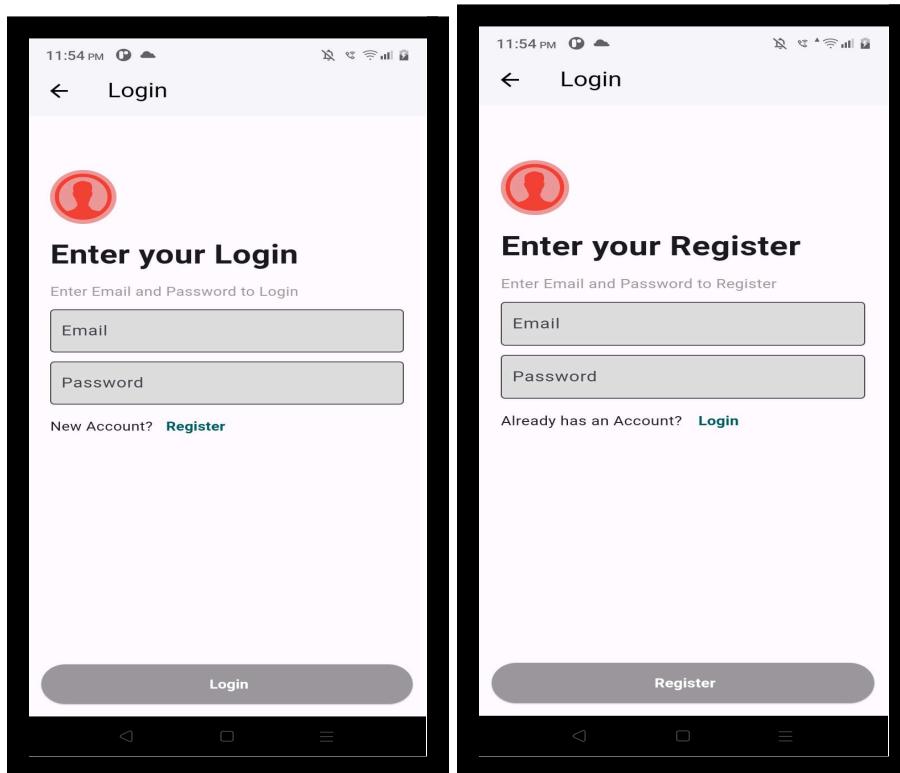
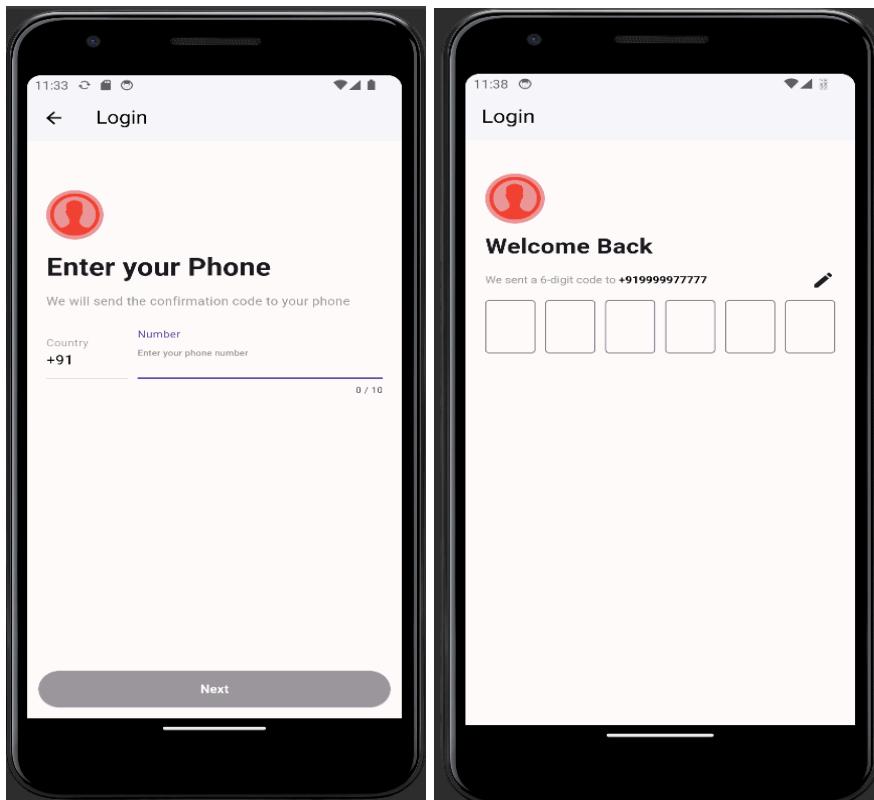
uses-material-design: true

assets:

- assets/images/cart.png
- assets/images/map.png

OUTPUT:





CONCLUSION:

We have successfully implemented the form in Flutter UI using widgets.

MAD & PWA Lab

Journal

Experiment No.	05
Experiment Title.	To apply navigation, routing and gestures in Flutter App
Roll No.	66
Name	Sneha Utekar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	15

Experiment 5

AIM: To apply navigation, routing and gestures in Flutter App

THEORY:

In a Flutter app, implementing navigation, routing, and gestures effectively is crucial for creating a seamless user experience. Let's delve into a theory on how to apply these concepts in your Flutter app:

A) Navigation and Routing:

a. Navigation:

Navigation in Flutter refers to moving between different screens or "routes" within your app. It's essential to organize your app's navigation in a clear and intuitive manner.

b. Routing:

Routing involves defining routes and managing transitions between them. Flutter provides a built-in routing system that allows you to define routes using a Route object. Common types of routes include MaterialPageRoute, CupertinoPageRoute, and Custom routes.

Application:

1. Route Management:

Define routes for each screen in your app using MaterialApp or CupertinoApp's "routes" parameter. This maps route names to builder functions.

```
routes: {  
  '/home': (context) => HomeScreen(),  
  '/profile': (context) => ProfileScreen(),  
  // Add more routes as needed  
}
```

2. Navigation:

Use Navigator.push() to navigate to a new route when a user interacts with UI elements such as buttons or gestures.

```
onPressed: () {  
  Navigator.pushNamed(context, '/profile');  
}
```

3. Navigation with Data: Pass data between routes using arguments.

```
Navigator.pushNamed(context, '/profile', arguments: userData);
```

4. Back Navigation:

Allow users to navigate back using Navigator.pop().

```
onPressed: () {  
  Navigator.pop(context);  
}
```

5. Named Routes vs. Route Builders:

Decide whether to use named routes or route builders based on your app's complexity and navigation requirements. Named routes are simpler, while route builders offer more customization.

C) Gestures:

a. Gestures:

Gestures enable users to interact with your app through touch-based actions like tapping, swiping, dragging, etc. Flutter provides GestureDetector and various gesture recognizers to handle these interactions.

b. Theory/Application:

1. GestureDetector:

Wrap your UI elements with GestureDetector to detect gestures.

```
GestureDetector(  
  onTap: () {  
    // Handle tap gesture  
  },  
  onDoubleTap: () {  
    // Handle double tap gesture  
  },  
  // Add more gesture callbacks as needed  
  child: YourWidget(),  
)
```

CODE:**location_screen.dart:**

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:location/location.dart';
import 'package:olx_clone/screens/home_screen.dart';
import 'package:olx_clone/screens/login_screen.dart';

class LocationScreen extends StatefulWidget {
    static const String id = "location-screen";

    @override
    State<LocationScreen> createState() => _LocationScreenState();
}

class _LocationScreenState extends State<LocationScreen> {
    bool _loading=false;
    Location location = new Location();

    late bool _serviceEnabled;
    late PermissionStatus _permissionGranted;
    late LocationData _locationData;

    Future<LocationData?> getLocation() async {
        _serviceEnabled = await location.serviceEnabled();
        if(!_serviceEnabled) {
            _serviceEnabled = await location.requestService();
            if(!_serviceEnabled) {
                return null;
            }
        }

        _permissionGranted = await location.hasPermission();
        if(_permissionGranted == PermissionStatus.denied) {
            _permissionGranted = await location.requestPermission();
            if(_permissionGranted != PermissionStatus.granted) {
                return null;
            }
        }
    }
}
```

```
_locationData = await location.getLocation();
print(_locationData);

return _locationData;
}

@Override

Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: Colors.white,
        body: Column(
            children: [
                Image.asset('assets/images/map.png', height: 350),
                SizedBox(height: 20),
                Text(
                    'Where do you want \nto buy/sell products?',
                    textAlign: TextAlign.center,
                    style: TextStyle(fontWeight: FontWeight.bold, fontSize: 25),
                ),
                SizedBox(
                    height: 10,
                ),
                Text(
                    'To fully enjoy all that we have to offer,\nwe need to know where to look for it',
                    textAlign: TextAlign.center,
                    style: TextStyle(fontSize: 12),
                ),
                SizedBox(
                    height: 30,
                ),
                Padding(
                    padding: const EdgeInsets.only(left: 20, right: 20, bottom: 10),
                    child: Row(
                        children: [
                            Expanded(
                                child: _loading ? Center(child: CircularProgressIndicator(
                                    valueColor: AlwaysStoppedAnimation<Color>(Theme.of(context).primaryColor),
                                )) : ElevatedButton.icon(
                                    style: ButtonStyle(
                                        backgroundColor: MaterialStateProperty.all<Color>(
                                            Theme.of(context).primaryColor,
                                        ),
                                    ),
                                ),
                            ),
                        ],
                    ),
                ),
            ],
        ),
    );
}
```



```

    ), // Removed the semicolon and added a comma
    ],
),
);
}
}

```

HomeScreen.dart:

```

import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:location/location.dart';
import 'package:flutter_geocoder/geocoder.dart';
import 'package:olx_clone/screens/login_screen.dart';
import 'package:olx_clone/widgets/banner_widget.dart';

class HomeScreen extends StatefulWidget {
  static const String id = 'home-screen';
  final LocationData? locationData;

  HomeScreen({required this.locationData});

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  String address = 'India';

  Future<String?> getAddress() async {
    // From coordinates
    final coordinates = Coordinates(widget.locationData?.latitude, widget.locationData?.longitude);
    var addresses = await Geocoder.local.findAddressesFromCoordinates(coordinates);
    var first = addresses.first;
    setState(() {
      address = first.addressLine ?? 'Address not found';
    });
    return first.addressLine;
  }

  @override
  void initState() {
    super.initState();
  }
}

```

```
getAddress();  
}  
  
Future<void> _signOut() async {  
  try {  
    await FirebaseAuth.instance.signOut();  
    Navigator.pushReplacementNamed(context, LoginScreen.id);  
  } catch (e) {  
    print('Error signing out: $e');  
  }  
}  
  
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      backgroundColor: Colors.white,  
      elevation: 0.0,  
      automaticallyImplyLeading: false,  
      title: InkWell(  
        onTap: () {},  
        child: Container(  
          width: MediaQuery.of(context).size.width,  
          child: Padding(  
            padding: const EdgeInsets.only(top: 8, bottom: 8),  
            child: Row(  
              children: [  
                Icon(CupertinoIcons.location_solid, color: Colors.black, size: 18,),  
                Flexible(  
                  child: Text(  
                    address,  
                    style: TextStyle(color: Colors.black, fontSize: 12, fontWeight: FontWeight.bold),  
                  ),  
                ),  
                Icon(Icons.arrow_downward_outlined, color: Colors.black, size: 18,),  
              ],  
            ),  
          ),  
        ),  
      ),  
      body: Column(  
        children: [  
          Container(  
           
```



```

        child: Text(
          'Sign Out',
          style: TextStyle(color: Colors.red),
        ),
      ),
    ),
  );
}
}

```

main.dart:

```

import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:olx_clone/screens/authentication/email_auth_screen.dart';
import 'package:olx_clone/screens/authentication/phoneauth_screen.dart';
import 'package:olx_clone/screens/home_screen.dart';
import 'package:olx_clone/screens/login_screen.dart';
import 'package:olx_clone/screens/splash_screen.dart';
import 'package:olx_clone/screens/location_screen.dart';
import 'dart:io';

```

```

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  Platform.isAndroid ?await Firebase.initializeApp(
    options: const FirebaseOptions(
      apiKey: "AIzaSyDuFtAm59795zlrpp16kTJTbd1Lac95KkI",
      appId: "1:232516563209:android:d2dea587c60de17bac63d8",
      messagingSenderId:"232516563209" ,
      projectId: "olxclone-12618"
    )
  ):await Firebase.initializeApp();
  runApp(MyApp());
}

```

```

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primaryColor: Colors.cyan.shade900,
        fontFamily: 'Horizon',
      ),
      initialRoute: SplashScreen.id,
    );
  }
}

```

```
routes: {
  SplashScreen.id: (context) => SplashScreen(),
  LoginScreen.id: (context) => LoginScreen(),
  PhoneAuthScreen.id: (context) => PhoneAuthScreen(),
  LocationScreen.id: (context) => LocationScreen(),
  HomeScreen.id: (context) => HomeScreen(locationData: null),
  EmailAuthScreen.id: (context) => EmailAuthScreen(),
},
);
```

pubsec.yaml:

```
name: olx_clone
description: "A new Flutter project."
publish_to: 'none' # Remove this line if you wish to publish to pub.dev
```

```
version: 1.0.0+1
```

environment:

```
sdk: '>=3.2.5 <4.0.0'
```

dependencies:

```
flutter:
  sdk: flutter
cupertino_icons: ^1.0.2
cloud_firestore: ^4.15.3
animated_text_kit: ^4.2.2
flutter_signin_button: ^2.0.0
firebase_core: ^2.24.2
firebase_auth: ^4.16.0
legacy_progress_dialog: ^0.1.2
location: ^5.0.3
flutter_geocoder: ^0.2.2-nullsafety
google_sign_in: ^6.2.1
email_validator: ^2.1.17
flutter_neumorphic_plus: ^3.3.0
```

dev_dependencies:

```
flutter_test:
  sdk: flutter
```

```
flutter_lints: ^2.0.0
```

flutter:

uses-material-design: true

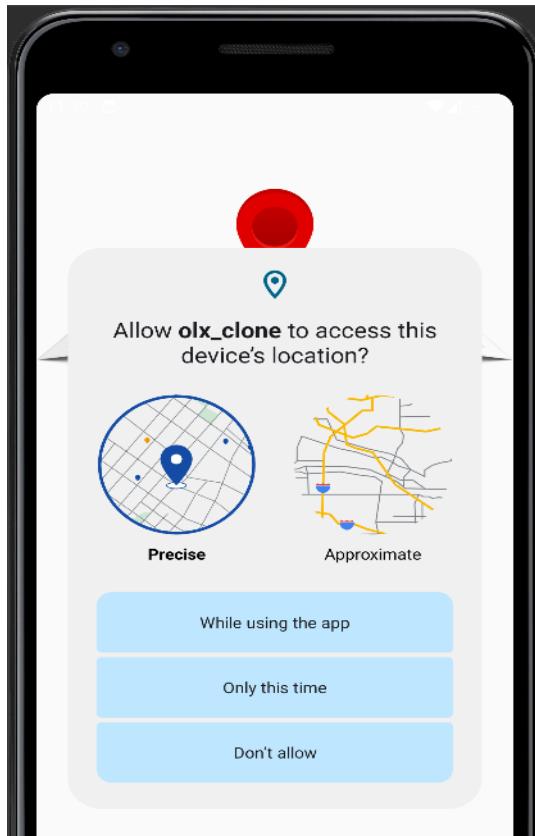
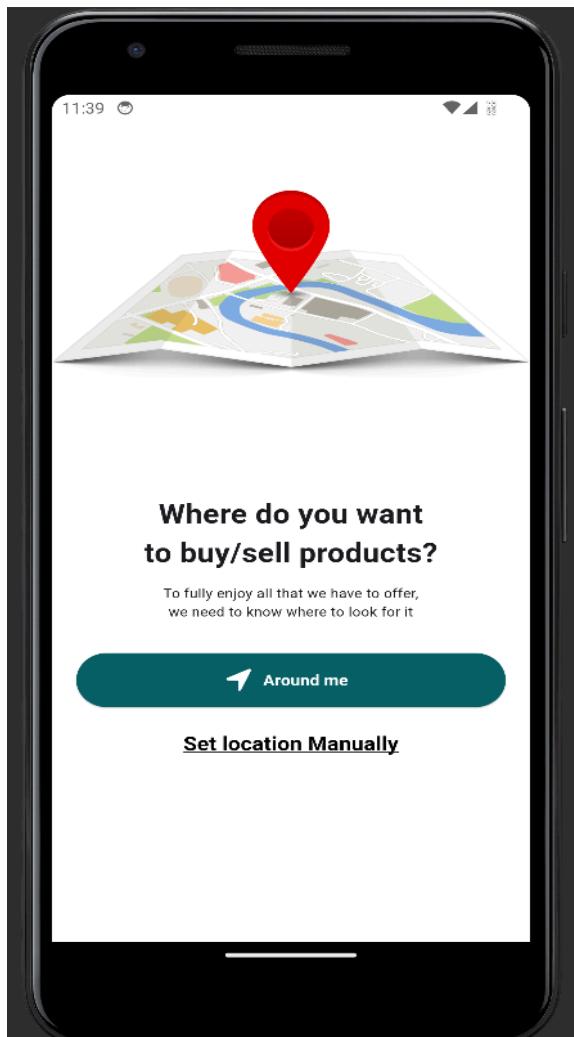
assets:

- assets/images/cart.png
- assets/images/map.png

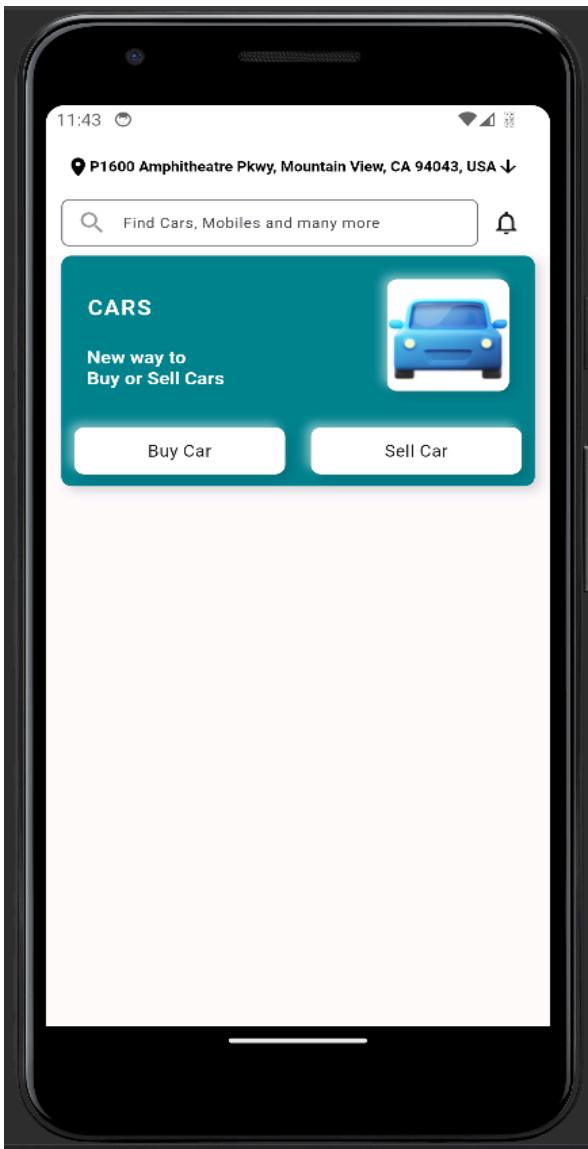
OUTPUT:

Location Screen:

**After clicking on ‘Around me’ button
It will navigate to this page**



Navigation to Home Screen:



CONCLUSION:

We have implemented navigation ,routing and gestures in the pages.

MAD & PWA Lab

Journal

Experiment No.	06
Experiment Title.	To Connect Flutter UI with fireBase database
Roll No.	66
Name	Sneha Utekar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO3: Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS
Grade:	15

Experiment 6

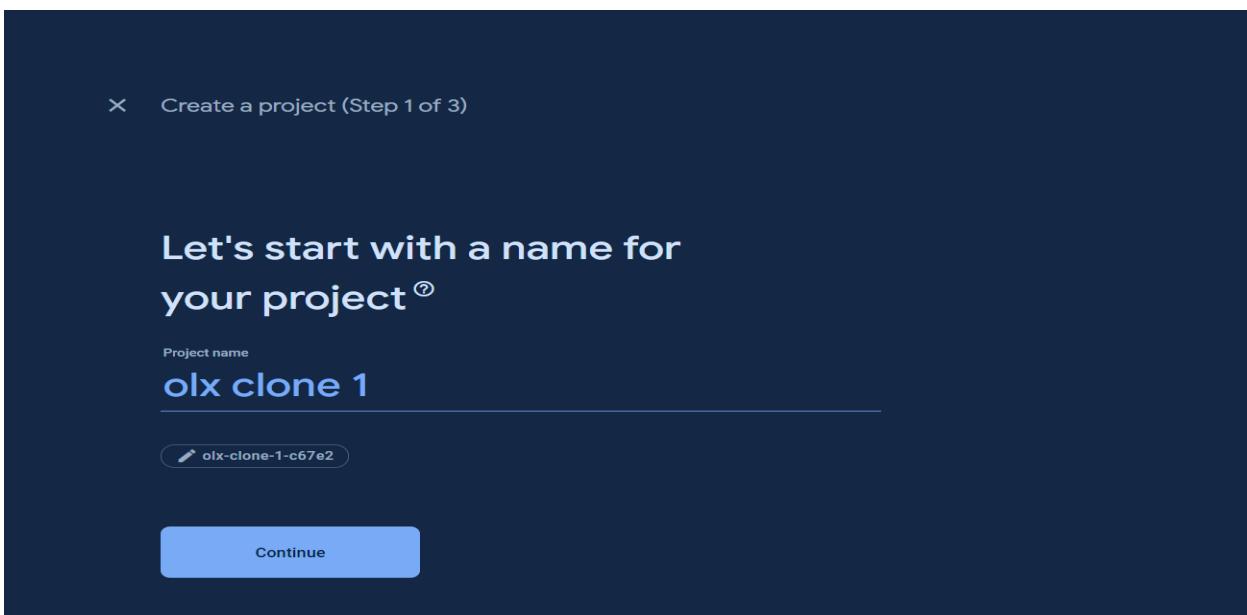
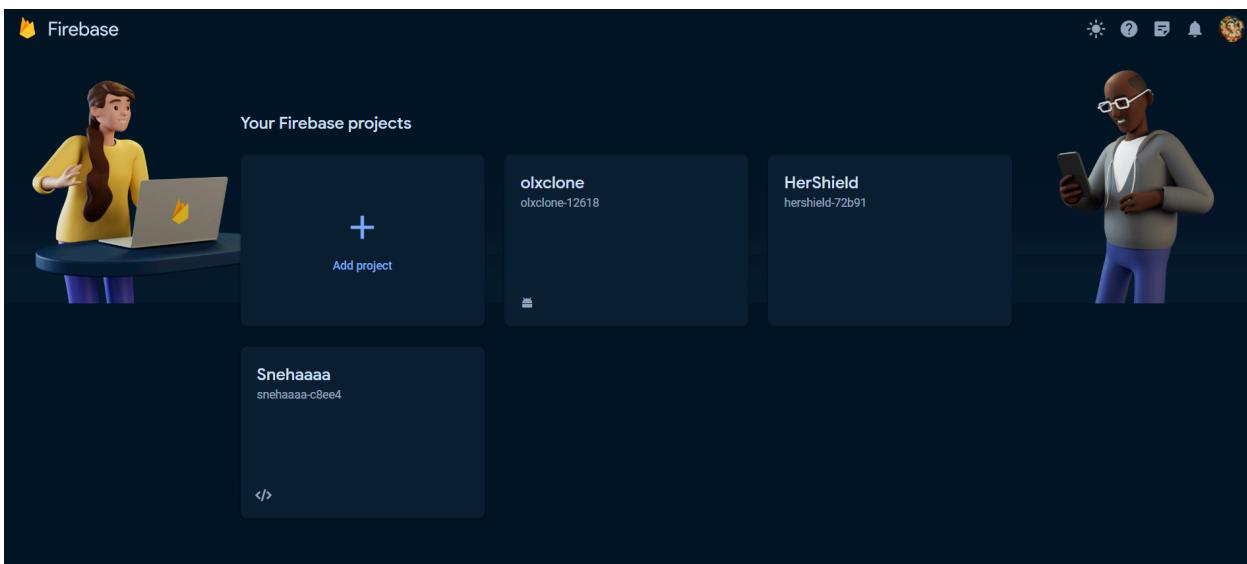
AIM: To connect flutter UI with firebase database.

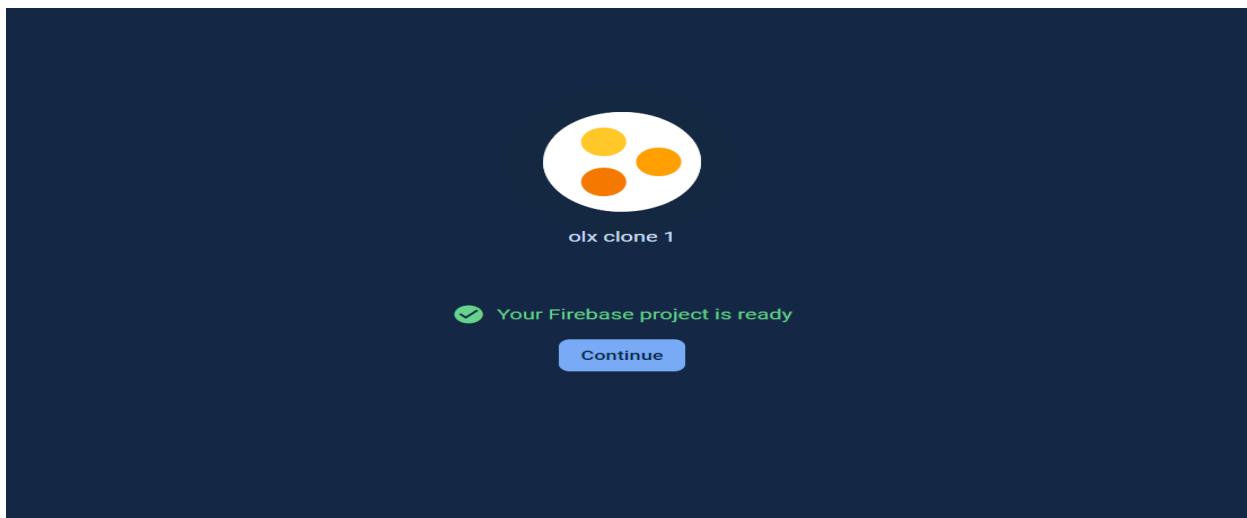
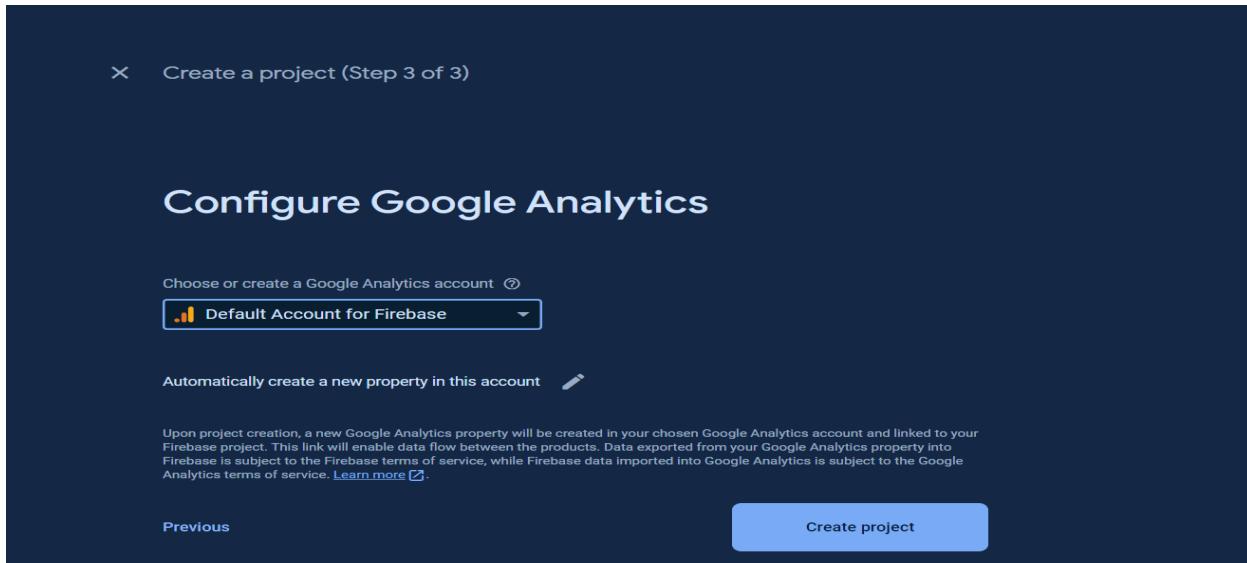
THEORY:

Firebase is a great backend solution for anyone that wants to use authentication, databases, cloud functions, ads, and countless other features within an app.

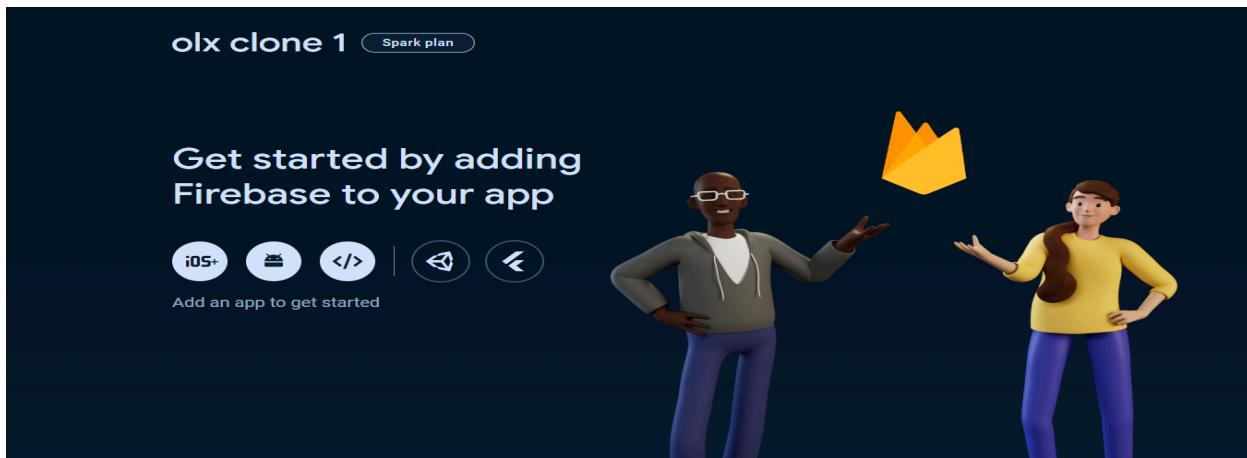
OUTPUT:

1.Create a new firebase project:

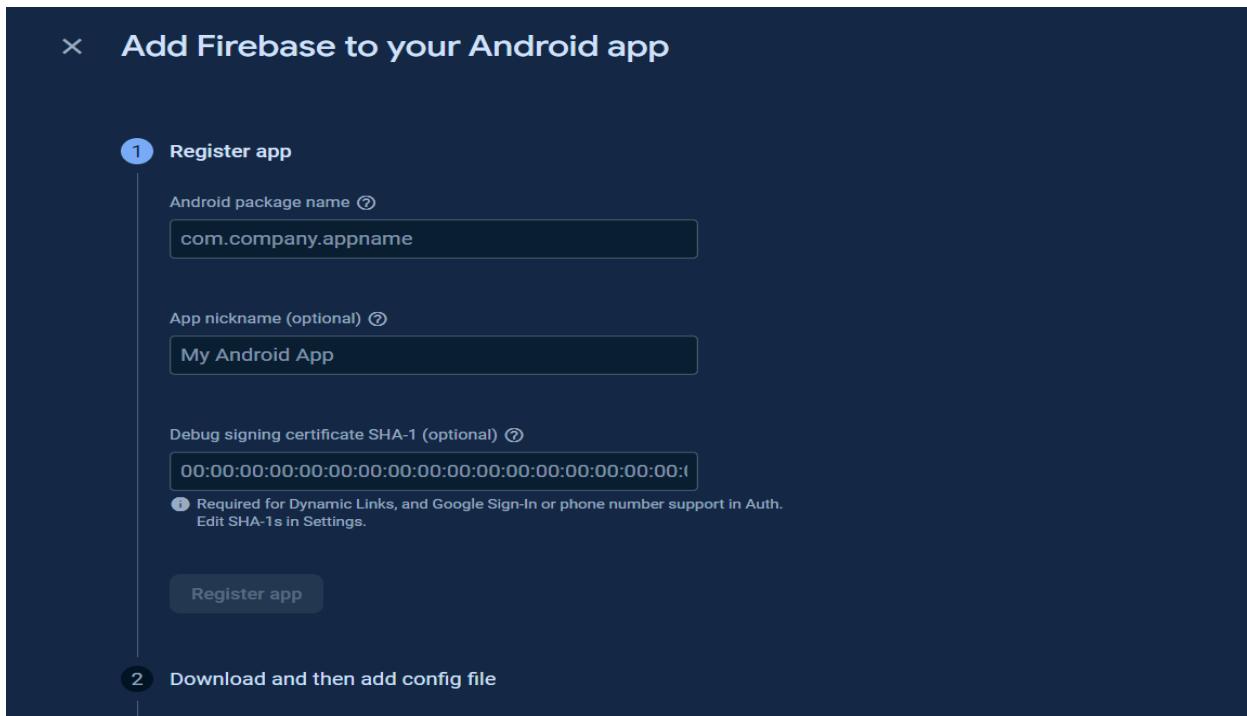




2. Select the Android option:

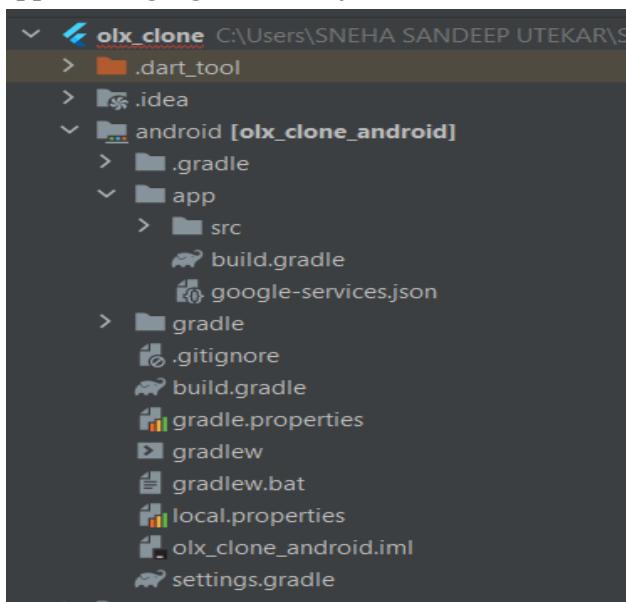


3.Add firebase to your app:



4.After completing all the steps ,you will get a google-services.json file.Download that file and add it to your Android project:

app→src→google-services.json:



google-services.json:

```
{
  "project_info": {
    "project_number": "232516563209",
    "project_id": "olxclone-12618",
    "storage_bucket": "olxclone-12618.appspot.com"
  },
  "client": [
    {
      "client_info": {
        "mobilesdk_app_id": "1:232516563209:android:d2dea587c60de17bac63d8",
        "android_client_info": {
          "package_name": "com.example.olx_clone"
        }
      },
      "oauth_client": [],
      "api_key": [
        {
          "current_key": "AIzaSyDuFtAm59795zlrpp16kTJTbd1Lac95KkI"
        }
      ],
      "services": {
        "appinvite_service": {
          "other_platform_oauth_client": []
        }
      }
    }
  ],
  "configuration_version": "1"
}
```

Add dependencies related to firebase in pubsec.yaml:

```
name: olx_clone
description: "A new Flutter project."

publish_to: 'none' # Remove this line if you wish to publish to pub.dev
version: 1.0.0+1

environment:
  sdk: '>=3.2.5 <4.0.0'
dependencies:
  flutter:
    sdk: flutter
```

```

cupertino_icons: ^1.0.2
cloud_firestore: ^4.15.3
animated_text_kit: ^4.2.2
flutter_signin_button: ^2.0.0
firebase_core: ^2.24.2
firebase_auth: ^4.16.0
legacy_progress_dialog: ^0.1.2
location: ^5.0.3
flutter_geocoder: ^0.2.2-nullsafety
google_sign_in: ^6.2.1
email_validator: ^2.1.17
flutter_neumorphic_plus: ^3.3.0
open_mail_app: ^0.4.5
csc_picker: ^0.2.7
provider: ^6.1.1
image_picker: ^0.7.5+3
firebase_storage: ^11.6.5
firebase_analytics: ^10.3.0

```

dev_dependencies:

```

flutter_test:
  sdk: flutter

```

flutter_lints: ^2.0.0

flutter:

uses-material-design: true

assets:

```

- assets/images/cart.png
- assets/images/map.png

```

CODE:

emailAuth_service:

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:olx_clone/screens/authentication/email_verification_screen.dart';

import '../screens/location_screen.dart';

class EmailAuthentication {
  CollectionReference users = FirebaseFirestore.instance.collection('users');

  Future<DocumentSnapshot> getAdminCredential({

```

```
required String email,
required String password,
required bool isLog,
required BuildContext context,
}) async {
DocumentSnapshot _result = await users.doc(email).get();
if (isLog) {
  emailLogin(email, password, context);
} else {
  if (_result.exists) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(
        content: Text('An Account Already Exists with this Email'),
      ),
    );
  } else {
    emailRegister(email, password, context);
  }
}

return _result;
}

emailLogin(String email, String password, BuildContext context) async {
try {
  UserCredential userCredential = await FirebaseAuth.instance
    .signInWithEmailAndPassword(
      email: email,
      password: password);
  if(userCredential.user?.uid!=null){
    Navigator.pushReplacementNamed(context,LocationScreen.id);
  }
} on FirebaseAuthException catch (e) {
  if (e.code == 'user-not-found') {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('No user found for that email')),
    );
  } else if (e.code == 'wrong-password') {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(
        content: Text('Wrong Password provided for that user')),
    );
  }
}
```

```
}
```

```
emailRegister(String email, String password, BuildContext context) async {
  try {
    UserCredential userCredential = await FirebaseAuth.instance
        .createUserWithEmailAndPassword(email: email, password: password);
    if (userCredential.user?.uid != null) {
      //login success .Add user data to firestore
      return users.doc(userCredential.user!.uid).set({
        'uid': userCredential.user!.uid,
        'mobile': null,
        'email': userCredential.user!.email,
        'name':null,
      }).then((value)async {
        //before going to location screen will send email verification
        await userCredential.user?.sendEmailVerification().then((value){
          //after sending verification email,screen will move to emailverification screen
          Navigator.pushReplacementNamed(context, EmailVerificationScreen.id);
        });
      });
    }
  }).catchError((onError) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('Failed to add user')),
    );
  });
}

} on FirebaseAuthException catch (e) {
  if (e.code == 'weak-password') {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('The password provided is too weak.')),
    );
  } else if (e.code == 'email-already-in-use') {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(
        content: Text('The account already exists for that email.')),
    );
  }
} catch (e) {
  print(e);
  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(content: Text('Error occurred')),
  );
}
```

```

}
}
```

phoneAuth_service:

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:olx_clone/screens/location_screen.dart';
import 'package:olx_clone/screens/authentication/otp_screen.dart';
class PhoneAuthService {
  FirebaseAuth auth = FirebaseAuth.instance;
  User? user=FirebaseAuth.instance.currentUser;
  CollectionReference users = FirebaseFirestore.instance.collection('users');

  Future<void> addUser(context,uid)async {
    // Call the user's CollectionReference to add a new user
    final QuerySnapshot result=await users.where('uid',isEqualTo: uid).get();
    List <DocumentSnapshot> document=result.docs;
    if(document.length>0){
      Navigator.pushReplacementNamed(context,LocationScreen.id);
    }else{
      return users.doc(user?.uid)
        .set({
          'uid': user?.uid, // John Doe
          'mobile': user?.phoneNumber, // Stokes and Sons
          'email': user?.email, // 42
          'name':null
        })
        .then((value) {
      Navigator.pushReplacementNamed(context,LocationScreen.id);
    })
        .catchError((error) => print("Failed to add user: $error"));
    }
  }

  Future<void> verifyPhoneNumber(BuildContext context, number) async {
    final PhoneVerificationCompleted verificationCompleted =
      (PhoneAuthCredential credential) async {
    await auth.signInWithCredential(credential);
  };

    final PhoneVerificationFailed verificationFailed =
      (FirebaseAuthException e) {
```

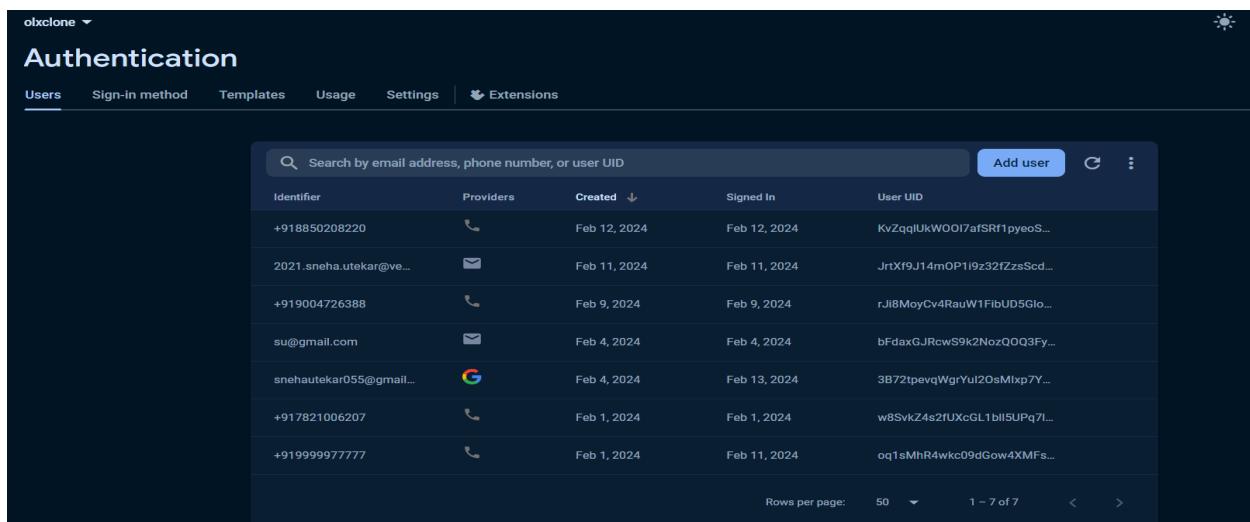
```

if (e.code == 'invalid-phone-number') {
  print('The provided phone number is not valid.');
}
print("The error is ${e.code}");
};

final PhoneCodeSent codeSent = (String verId, int ?resendToken) async {
  Navigator.push(
    context, MaterialPageRoute(builder: (context) => OTPScreen(number: number,verId: verId,)),);
};

try {
  auth.verifyPhoneNumber(phoneNumber: number,
    verificationCompleted: verificationCompleted,
    verificationFailed: verificationFailed,
    codeSent: codeSent,
    timeout: const Duration(seconds: 60),
    codeAutoRetrievalTimeout: (String verificationId) {
      print(verificationId);
    });
} catch (e) {
  print('Error ${e.toString()}');
}
}
}
}

```

Firebase Results:**Authentication:**


The screenshot shows the Firebase Authentication console under the 'olxclone' project. The 'Authentication' tab is selected, and the 'Users' tab is active. The interface includes a search bar at the top, followed by a table displaying user information. The columns in the table are: Identifier, Providers, Created, Signed In, and User UID. The table lists seven users, each with their corresponding details like creation date, sign-in history, and unique User UID.

Identifier	Providers	Created	Signed In	User UID
+91850208220	📞	Feb 12, 2024	Feb 12, 2024	KvZqlUkWOOI7afSRf1pyeoS...
2021.sneha.utekar@ve...	✉️	Feb 11, 2024	Feb 11, 2024	JrtXf9J14mOP19z32fZzsSd...
+919004726388	📞	Feb 9, 2024	Feb 9, 2024	rJl8MoyCv4RauW1FibUD5Glo...
su@gmail.com	✉️	Feb 4, 2024	Feb 4, 2024	bFdaxGJRcwS9k2NozOOQ3Fy...
snehautekar055@gmail...	GOOGLE	Feb 4, 2024	Feb 13, 2024	3B72tpevqWgrYu12OsMlxp7Y...
+917821006207	📞	Feb 1, 2024	Feb 1, 2024	w8SvkZ4s2fUxGL1bl5UPq7I...
+919999977777	📞	Feb 1, 2024	Feb 11, 2024	oq1sMhR4wkc09dGow4XMFs...

Sign-in methods:

The screenshot shows the Firebase Authentication interface. At the top, there are tabs for Users, Sign-in method (which is selected), Templates, Usage, Settings, and Extensions. A yellow warning message in a box states: "To prevent abuse, new projects currently have a sent SMS daily quota of 10/day. To increase this quota, please add a billing account to the project." Below this, under "Sign-in providers", there is a table with three rows: Email/Password (Enabled), Phone (Enabled), and Google (Enabled). There is also a blue button labeled "Add new provider".

Firebase Database:

Before creating the database add the following rules:

The screenshot shows the Cloud Firestore Rules tab. On the left, there is a sidebar with a star icon and a timestamp "Feb 3, 2024 • 2:05 PM". The main area displays the security rules code:

```

1  rules_version = '2';
2
3  service cloud.firestore {
4      match /databases/{database}/documents {
5          match /{document=**} {
6              allow read, write: if true;
7          }
8      }
9  }
10

```

These are the users:

(default)	users	3B72tpevqWgrYui2OsMlxp7YFqh1
+ Start collection	+ Add document	+ Start collection
categories	3B72tpevqWgrYui2OsMlxp7YFqh1 >	+ Add field
users >	JrtXF9J14mOP19z32fZzsScdw33 dnyanu2003@gmail.com su@gmail.com	address: "101, Om Sainath Apartment, near Ranu gas godown, behind Nikshita apartment, Shivalik Nagar, Mahalaxmi Nagar, Ambernath, Maharashtra 421501, India" email: "snehautekar055@gmail.com" location: [19.2003207° N, 73.1925926° E] mobile: null name: null uid: "3B72tpevqWgrYui2OsMlxp7YFqh1"

Firebase Storage:

Before using the firebase Storage ,add the following rules:

```

service firebase.storage {
  match '/{bucket}/{...}' {
    match '/allPaths=**' {
      // Allow access to authenticated users
      allow read, write: if request.auth != null;
    }
  }
}

```

I have used firebase storage to store the images ,icons and categories

Name	Type	Last modified
banner/	Folder	-
categories/	Folder	-
productImage/	Folder	-

Icons for categories:

gs://olxclone-12618.appspot.com > categories

Name	Size	Type	Last modified
Bike.png	13.84 KB	image/png	Feb 12, 2024
Books.png	16.69 KB	image/png	Feb 12, 2024
Cars.png	1.84 KB	image/png	Feb 12, 2024
Electronics(Monitor).png	12.1 KB	image/png	Feb 12, 2024
Electronics(mobile).png	12.55 KB	image/png	Feb 12, 2024
Fashion.png	13.62 KB	image/png	Feb 12, 2024
Furniture.png	19.36 KB	image/png	Feb 12, 2024
Pets.png	17.37 KB	image/png	Feb 12, 2024
Sports.png	20.29 KB	image/png	Feb 12, 2024

Icons for banner :

gs://olxclone-12618.appspot.com > banner

Name	Size	Type	Last modified
Icons8-car-100.png	20.51 KB	image/png	Feb 4, 2024

Images:

gs://olxclone-12618.appspot.com > productimage

Name	Size	Type	Last modified
1708022786637814	17.73 KB		Feb 16, 2024
1708022804036798	18.25 KB		Feb 16, 2024
1708022815471922	18.25 KB		Feb 16, 2024
1708022831233708	56.92 KB		Feb 16, 2024
1708022984091655	18.25 KB		Feb 16, 2024
1708023089662516	17.73 KB		Feb 16, 2024
1708023117372742	18.25 KB		Feb 16, 2024
1708023232207909	17.73 KB		Feb 16, 2024

1708022786637814

Name: 1708022786637814
Size: 18,156 bytes
Type:
Created: Feb 16, 2024, 12:16:29 AM
Updated: Feb 16, 2024, 12:16:29 AM
File location
Other metadata

firebase_service.dart:

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter_geocoder/geocoder.dart';
import 'package:olx_clone/screens/home_screen.dart';

class FirebaseService {

    CollectionReference users=FirebaseFirestore.instance.collection('users');
    CollectionReference categories=FirebaseFirestore.instance.collection('categories');
    User? user=FirebaseAuth.instance.currentUser;

    Future<void> updateUser(Map<String,dynamic>data,context) {
        return users
            .doc(user?.uid)
            .update(data)
            .then((value) {
                Navigator.pushNamed(context,HomeScreen.id);
            })
            .catchError((error) {
                ScaffoldMessenger.of(context).showSnackBar(
                    const SnackBar(
                        content:Text('Failed to update location'),
                    ),
                );
            });
    }

    Future<String>getAddress(lat,long)async{
        final coordinates = new Coordinates(lat,long);
        var addresses = await Geocoder.local.findAddressesFromCoordinates(coordinates);
        var first = addresses.first;

        return first.addressLine ?? "Address not found";
    }

    Future<DocumentSnapshot>getUserData()async{
        DocumentSnapshot doc= await users.doc(user?.uid).get();
        return doc;
    }
}
```

cat_provider.dart:

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:olx_clone/services/firebase_service.dart';

class CategoryProvider with ChangeNotifier{
  FirebaseService _service=FirebaseService();

  late DocumentSnapshot doc;
  late String selectedCategory;
  late String selectedSubCategory= "";
  DocumentSnapshot? userDetails;
  List<String> urlList = [];
  Map<String,dynamic> dataToFirestore={}; //this is the data we are going to upload to firestore

  getCategory(selectedCat){
    this.selectedCategory=selectedCat;
    notifyListeners();
  }

  getSubCategory(selectedsubCat){
    this.selectedSubCategory=selectedsubCat;
    notifyListeners();
  }
  getCatSnapshot(snapshot){
    this.doc=snapshot;
    notifyListeners();
  }

  getImages(List<String> urls) {
    urlList.addAll(urls); // Append the new URLs to the existing list
    notifyListeners();
  }

  getData(data) {
    this.dataToFirestore=data;
    notifyListeners();
  }

  getUserDetails(){
    _service.getUserData().then((value){
      this.userDetails=value;
    });
  }
}
```

```

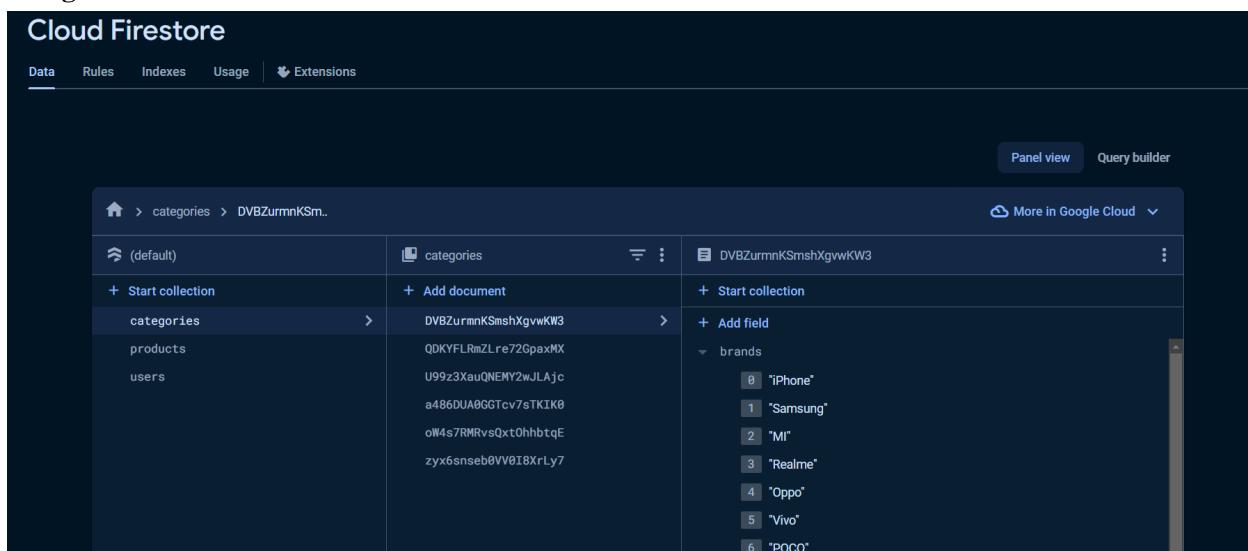
    notifyListeners();

  });
}
}
}

```

Cloud Firestore:

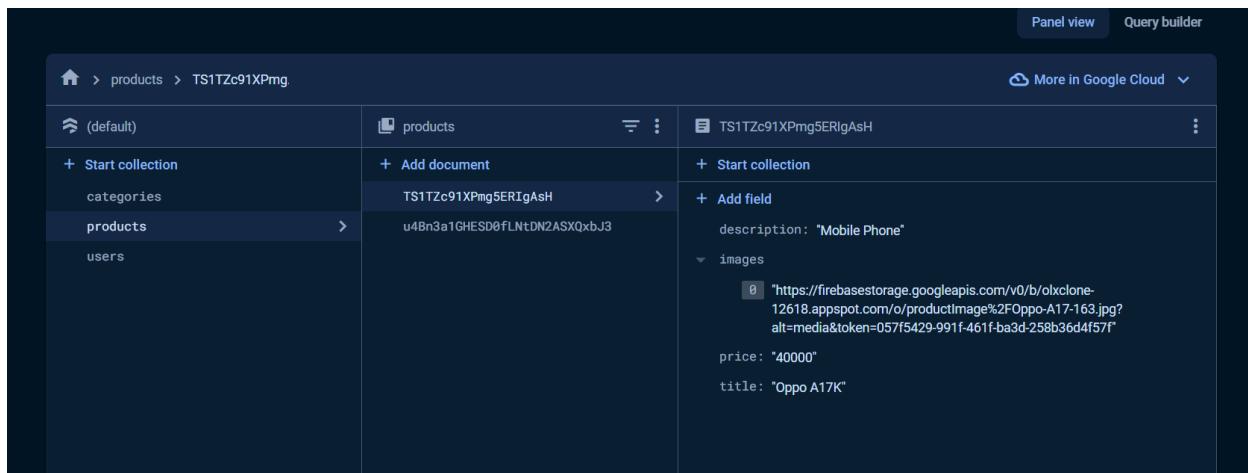
Categories:



The screenshot shows the Cloud Firestore interface for the 'categories' collection. The left sidebar lists collections: 'categories' (selected), 'products', and 'users'. The main area shows a document named 'DVBZurmnnKSmshXgvwKW3' under the 'categories' collection. This document contains three sub-fields: 'products' (document ID 'QDKYFLRmZLre72GpaxMX'), 'users' (document ID 'U99z3Xau0NEMY2wJLAjc'), and 'brands' (array of strings). The 'brands' array contains six items: 'iPhone', 'Samsung', 'MI', 'Realme', 'Oppo', and 'Vivo'.

Document ID	Field	Value
DVBZurmnnKSmshXgvwKW3	products	QDKYFLRmZLre72GpaxMX
DVBZurmnnKSmshXgvwKW3	users	U99z3Xau0NEMY2wJLAjc
DVBZurmnnKSmshXgvwKW3	brands	<ul style="list-style-type: none"> 0 "iPhone" 1 "Samsung" 2 "MI" 3 "Realme" 4 "Oppo" 5 "Vivo" 6 "POCO"

products:



The screenshot shows the Cloud Firestore interface for the 'products' collection. The left sidebar lists collections: 'products' (selected), 'categories', and 'users'. The main area shows a document named 'TS1TZc91XPmg5ERigAsH' under the 'products' collection. This document contains two fields: 'description' (value: "Mobile Phone") and 'images' (array of objects). Each object in the 'images' array has a key 'url' with a value pointing to a Firebase storage URL for an Oppo A17K phone image.

Document ID	Field	Value
TS1TZc91XPmg5ERigAsH	description	"Mobile Phone"
TS1TZc91XPmg5ERigAsH	images	<ul style="list-style-type: none"> 0 "https://firebasestorage.googleapis.com/v0/b/olxclone-12618.appspot.com/o/productImage%2FOppo-A17-163.jpg?alt=media&token=057f5429-991f-461f-ba3d-258b36d4f57f"

users:

The screenshot shows the Google Cloud Firestore interface. On the left, there's a sidebar with a home icon, followed by navigation steps: users > qBdjErqNDRdaV. Below this, there are three main sections: 'Start collection' (categories, products, users), 'Add document' (with a plus sign), and a specific document view for 'qBdjErqNDRdaVg6R8Je8vDJUwYq2'. This document view includes fields like 'address', 'contactDetails', 'comtactEmail', 'contactMobile', 'name', 'email', 'location', and 'mobile'. At the top right, there are 'Panel view' and 'Query builder' buttons.

(default)	users	qBdjErqNDRdaVg6R8Je8vDJUwYq2
+ Start collection	+ Add document	+ Start collection
categories	qBdjErqNDRdaVg6R8Je8vDJUwYq2	+ Add field
products	rwMImVN8iXRDmIDWICwPimu5Bgz1	address: "101, Om Sainath Apartment, near Ranu gas godown, behind Nikshita apartment, Shivalik Nagar, Mahalaxmi Nagar, Ambernath, Maharashtra 421501, India"
users	u4Bn3a1GHESD0fLNtDN2ASQxbJ3	contactDetails
		address: "101, Om Sainath Apartment, near Ranu gas godown, behind Nikshita apartment, Shivalik Nagar, Mahalaxmi Nagar, Ambernath, Maharashtra 421501, India"
		comtactEmail: "2021.sneha.utekar@ves.ac.in"
		contactMobile: "7821006207"
		name: "Sneha Utekar"
		email: "2021.sneha.utekar@ves.ac.in"
		location: [19.2003743° N, 73.192612° E]
		mobile: null

CONCLUSION:

We have understood how to connect the database with flutter UI.

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.	66
Name	Sneha Utekar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO4: Understand various PWA frameworks and their requirements
Grade:	15

Experiment 7

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

Theory:

Regular Web App

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

Progressive Web App

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

Difference between PWAs vs. Regular Web Apps:

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

1. Native Experience

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

2. Ease of Access

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

3. Faster Services

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

4. Engaging Approach

As already shared, the PWAs can employ push notifications and other native device features

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

5. Updated Real-Time Data Access

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

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

6. Discoverable

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

7. Lower Development Cost

Progressive web apps can be installed on the user device like a native device, but it does not demand submission on an App Store. This makes it far more cost-effective than native mobile applications while offering the same set of functionalities.

The main features are:

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

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

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

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

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

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

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

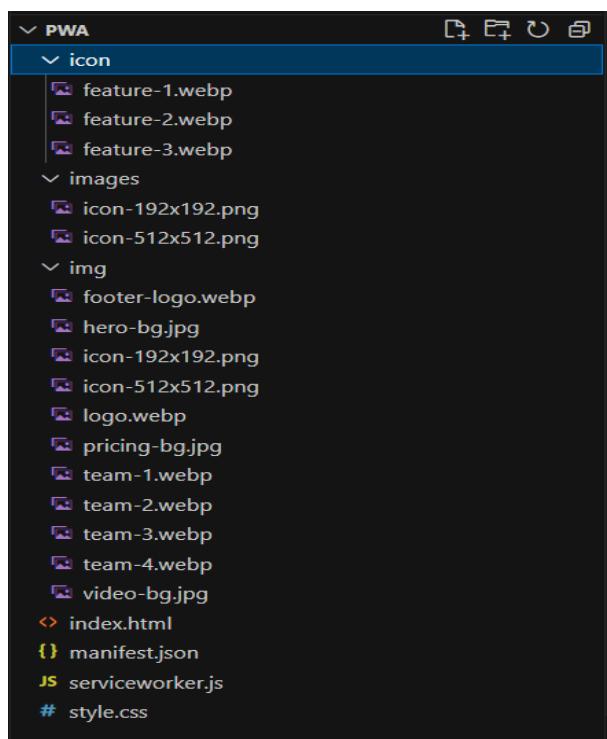
Installable — They allow the user to “save” the apps that he considers most useful with the corresponding icon on the screen of his mobile terminal (home screen) without having to face all the steps and problems related to the use of the app store.

Linkable — Easily shared via URL without complex installations.

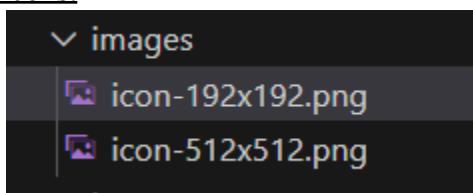
Offline — Once more it is about putting the user before everything, avoiding the usual error message in case of weak or no connection. The PWA are based on two particularities: first of all the ‘skeleton’ of the app, which recalls the page structure, even if its contents do not respond and its elements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

CODE AND OUTPUT:

Folder Structure



Icons:



Index.html:

```
<!DOCTYPE html>
<html>
<head>

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="X-UA-Compatible"
      content="ie=edge">

<title>PWA Tutorial</title>

<meta name="apple-mobile-web-app-status-bar" content="#aa7700">
<meta name="theme-color" content="black">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfI"
      crossorigin="anonymous">

      <!-- Font Owesome icon -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.2/css/all.min.css"
      integrity="sha512-HK5fgLBL+xu6dm/Ii3z4xhlSUyZgTT9tuc/hSrtw6uzJOvgRr2a9jyxxT1ely+BxFAmJKVSTbpMCuL7qxO8w=="
      crossorigin="anonymous" />

<link rel="stylesheet" href="style.css">
<!-- Manifest File link -->
<link rel="manifest" href="manifest.json">

</head>

<body>
    <!-- Header section -->
    <div class="container-fluid header">
        <div class="row">
            <div class="col-3 bg-warning logoSection p-2">
                
            </div>
            <div class="col-9">
                <div class="row">
                    <div class="col-6 menu">
```

```

<ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Courses</a></li>
    <li><a href="#">News</a></li>
    <li><a href="#">Pages</a></li>
    <li><a href="#">Contact</a></li>
</ul>
</div>
<div class="col-6 social-icon">
    <div class="row">
        <div class="col-5 icon">
            <a href=""><i class="fas fa-search"></i></a>
            <a href=""><i class="fab fa-facebook"></i></a>
            <a href=""><i class="fab fa-instagram"></i></a>
            <a href=""><i class="fab fa-twitter"></i></a>
            <a href=""><i class="fab fa-dribbble"></i></a>
        </div>
        <div class="col-4 header-btn">
            <a href="#" class="btn btn-warning">Get Started</a>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<section class="slider-section">
    <div class="container">
        <div class="row">
            <div class="col-12 slider-content">
                <h3>Best Options for you</h3>
                <h1>Drive safe & get lichence</h1>
                <div class="slider-btn">
                    <a href="#" class="btn btn-warning px-4">Contact Us</a>
                    <a href="#" class="btn btn-primary px-4">See Courses</a>
                </div>
            </div>
        </div>
    </div>
</div>
</div>
</div>
<!-- Feature Section started -->
<section class="feature">

```

```

<div class="container">
  <div class="row">
    <div class="col-lg-4 col-md-4 col-sm-12">
      <div class="content content-center">
        <span>Why choose us ?</span>
        <h2>Our feature</h2>
        <p class="paragraph">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod
          tempor incididunt ut labore et dolore magna aliqua viverra maecenas facilisis. </p>
        <a href="#" class="btn btn-own">See Courses</a>
      </div>
    </div>
    <div class="col-lg-4 col-md-4 col-sm-12">
      <div class="single-icon">
        
        <h3>Unlimited car support</h3>
      </div>
      <div class="single-icon">
        
        <h3>Driving School Insures</h3>
      </div>
    </div>
    <div class="col-lg-4 col-md-4 col-sm-12">
      <div class="single-icon content-center">
        
        <h3>Any Time Any Location</h3>
      </div>
    </div>
  </div>
</div>
</section>

<!-- Welcome section started -->
<section class="container welcome">
  <div class="row">
    <div class="col-lg-6 col-md-12 col-sm-12 welcome-img">
      
    </div>
    <div class="col-lg-6 col-md-12 col-sm-12 shadow-wrap">
      <div class="content content-center">
        <span>Welcome to Online traffic school</span>
        <h2>looking for lessons?</h2>
      </div>
    </div>
  </div>
</section>

```

<p class="paragraph">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor

incididunt ut labore et dolore magna aliqua. Quis ipsum vidnas accumsan lacus vel facilisis.

```
</p>
<a href="#" class="btn btn-own">Learn More</a>
</div>
</div>
</div>
</section>
```

<!-- Contact Form Started -->

```
<section class="application-form">
<div class="container">
<div class="content text-center">
<span>Giving Best Options For You</span>
<h2 class="p-3">Application Form</h2>
</div>
<form class="row form">
<div class="col-lg-4 col-md-6 col-sm-12">
<input type="text" placeholder="Your Name">
</div>
<div class="col-lg-4 col-md-6 col-sm-12">
<input type="text" placeholder="Your Name">
</div>
<div class="col-lg-4 col-md-6 col-sm-12">
<input type="text" placeholder="Your Name">
</div>
<div class="col-lg-4 col-md-6 col-sm-12">
<input type="text" placeholder="Your Name">
</div>
<div class="col-lg-4 col-md-6 col-sm-12">
<select name="" id="course">
<option value="">Course Type</option>
<option value="">Safe Driving Course</option>
<option value="">Motorhome Drivers</option>
</select>
</div>
<div class="col-lg-4 col-md-6 col-sm-12">
<select name="" id="car">
<option value="">Car Type</option>
<option value="">Hatchback</option>
<option value="">Sedan</option>
</select>
```

```
        </div>
    </form>
</div>
<div class="row">
    <div class="col-12 text-center p-4">
        <input type="submit" value="Submit" class="btn btn-own ">
    </div>
</div>
</div>
</section>
```

```
<!-- Pricing section started here -->
<section class="pricing">
  <div class="container">
    <div class="content text-center">
      <span>Get Special Offer</span>
      <h2 class="p-3">Our Pricing</h2>
    </div>
    <div class="row">
      <div class="col-lg-4 col-md-6 col-sm-12">
        <div class="box">
          <div class="box-heading">
            <span>20% off</span>
            <h1>$ 199</h1>
            <h5>Personal Driving</h5>
          </div>
          <div class="box-list-item">
            <p>Full course theory</p>
            <p>Full driving course</p>
            <p>Training in first aid</p>
            <p>Practical sessions</p>
            <p>Psychological support</p>
          </div>
          <div class="box-btn btn-own">
            <a href="#">Get Started</a>
          </div>
        </div>
      </div>
      <div class="col-lg-4 col-md-6 col-sm-12">
        <div class="box">
          <div class="box-heading">
            <span>20% off</span>
```

```
<h1>$ 199</h1>
<h5>Personal Driving</h5>
</div>
<div class="box-list-item">
    <p>Full course theory</p>
    <p>Full driving course</p>
    <p>Training in first aid</p>
    <p>Practical sessions</p>
    <p>Psychological support</p>
</div>
<div class="box-btn btn-own">
    <a href="#">Get Started</a>
</div>

</div>
</div>
<div class="col-lg-4 col-md-6 col-sm-12">
    <div class="box">
        <div class="box-heading">
            <span>20% off</span>
            <h1>$ 199</h1>
            <h5>Personal Driving</h5>
        </div>
        <div class="box-list-item">
            <p>Full course theory</p>
            <p>Full driving course</p>
            <p>Training in first aid</p>
            <p>Practical sessions</p>
            <p>Psychological support</p>
        </div>
        <div class="box-btn btn-own">
            <a href="#">Get Started</a>
        </div>

        </div>
        </div>
    </div>
</div>

</div>
</section>

<!-- Instructor section -->
<section class="instructor-section">
    <div class="container">
```

```
<div class="row">
  <div class="col-lg-10 col-md-10 col-sm-12">
    <div class="content text-left">
      <span>Our Great Team</span>
      <h2>Our Instructors</h2>
    </div>
  </div>
  <div class="col-lg-2 col-md-2 col-sm-12 my-4">
    <a href="#" class="btn btn-own">View All</a>
  </div>
</div>
<div class="row">
  <div class="col-lg-6 col-md-6 col-sm-12 instructor-box">
    <div class="row instructor-wrap">
      <div class="col-5 instructor-img">
        
      </div>
      <div class="col-7 instructor-details">
        <h3>DAVID WARNER</h3>
        <span>Instructor</span>
        <p>Morbi accumsan ipsum velit. Nam nec tellus a odio tincidunt auctor.</p>
        <div class="instructor-social">
          <a href=""><i class="fab fa-facebook"></i></a>
          <a href=""><i class="fab fa-instagram"></i></a>
          <a href=""><i class="fab fa-twitter"></i></a>
          <a href=""><i class="fab fa-dribbble"></i></a>
        </div>
      </div>
    </div>
  </div>
  <div class="col-lg-6 col-md-6 col-sm-12 instructor-box">
    <div class="row instructor-wrap">
      <div class="col-5 instructor-img">
        
      </div>
      <div class="col-7 instructor-details">
        <h3>DAVID WARNER</h3>
        <span>Instructor</span>
        <p>Morbi accumsan ipsum velit. Nam nec tellus a odio tincidunt auctor.</p>
        <div class="instructor-social">
          <a href=""><i class="fab fa-facebook"></i></a>
          <a href=""><i class="fab fa-instagram"></i></a>
          <a href=""><i class="fab fa-twitter"></i></a>
          <a href=""><i class="fab fa-dribbble"></i></a>
        </div>
      </div>
    </div>
  </div>
</div>
```



```
<!-- Footer Section -->
<footer class="footer">
  <div class="container">
    <div class="row footer-wrap">
      <div class="col-lg-3 col-md-6 col-sm-12 footer-details">
        <h3>Company</h3>
        <a href="#">About Us</a>
        <a href="#">Company</a>
        <a href="#">Press & Blog</a>
        <a href="#">Privacy Policy</a>
        <a href="#">Faq</a>
      </div>
      <div class="col-lg-3 col-md-6 col-sm-12 footer-details">
        <h3>Courses</h3>
        <a href="#">Winter driving</a>
        <a href="#">Program for seniors</a>
        <a href="#">Adult in car lessons</a>
        <a href="#">Defensive driving</a>
        <a href="#">Stick shit lessons</a>
      </div>
      <div class="col-lg-3 col-md-6 col-sm-12 footer-details">
        <h3>USEFUL LINKS</h3>
        <a href="#">Home</a>
        <a href="#">Drupal Themes</a>
        <a href="#">WordPress Themes</a>
        <a href="#">Support</a>
        <a href="#">Services</a>
      </div>
      <div class="col-lg-3 col-md-6 col-sm-12 footer-details">
        
        <a href="#">Address : 15 Division Street, New York, NY 12032, United States of America</a>
        <a href="#">Phone : +0 (123) 456789</a>
        <a href="#">Email : Kaseo@gmail.com</a>
        <a href="#">Fax : +8 (123) 456 789</a>
      </div>
    </div>
    <div class="footer-copyright">
      <span>Copyright &copy 2021 All rights reserved By Al Siam</span>
    </div>
  </div>
</footer>
```

```
<!-- Bootstrap Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOOn0"
crossorigin="anonymous">
</script>
<script>
  window.addEventListener('load', () => {
    registerSW();
  });

  // Register the Service Worker
  async function registerSW() {
    if ('serviceWorker' in navigator) {
      try {
        await navigator
          .serviceWorker
          .register('serviceworker.js');
      }
      catch (e) {
        console.log('SW registration failed');
      }
    }
  }
</script>

</body>
```

serviceworker.js

```
var staticCacheName = "pwa";  
  
self.addEventListener("install", function (e) {  
  e.waitUntil(  
    caches.open(staticCacheName).then(function (cache) {  
      return cache.addAll(["/"]);  
    })  
  );  
});  
  
self.addEventListener("fetch", function (event) {  
  console.log(event.request.url);  
  event.respondWith(  
    // Cache logic here  
  );  
});
```

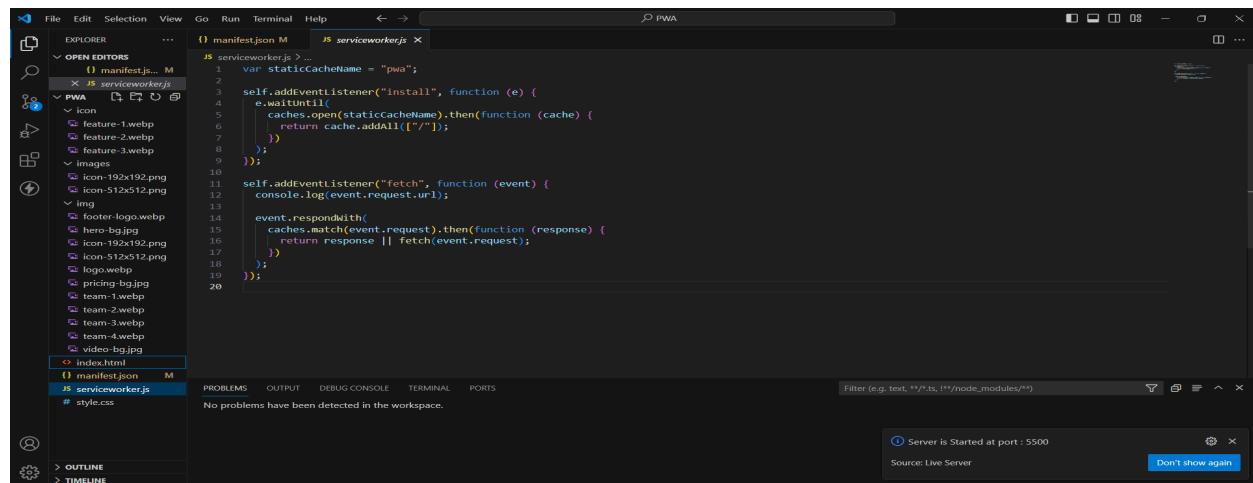
```

caches.match(event.request).then(function (response) {
  return response || fetch(event.request);
})
);
});

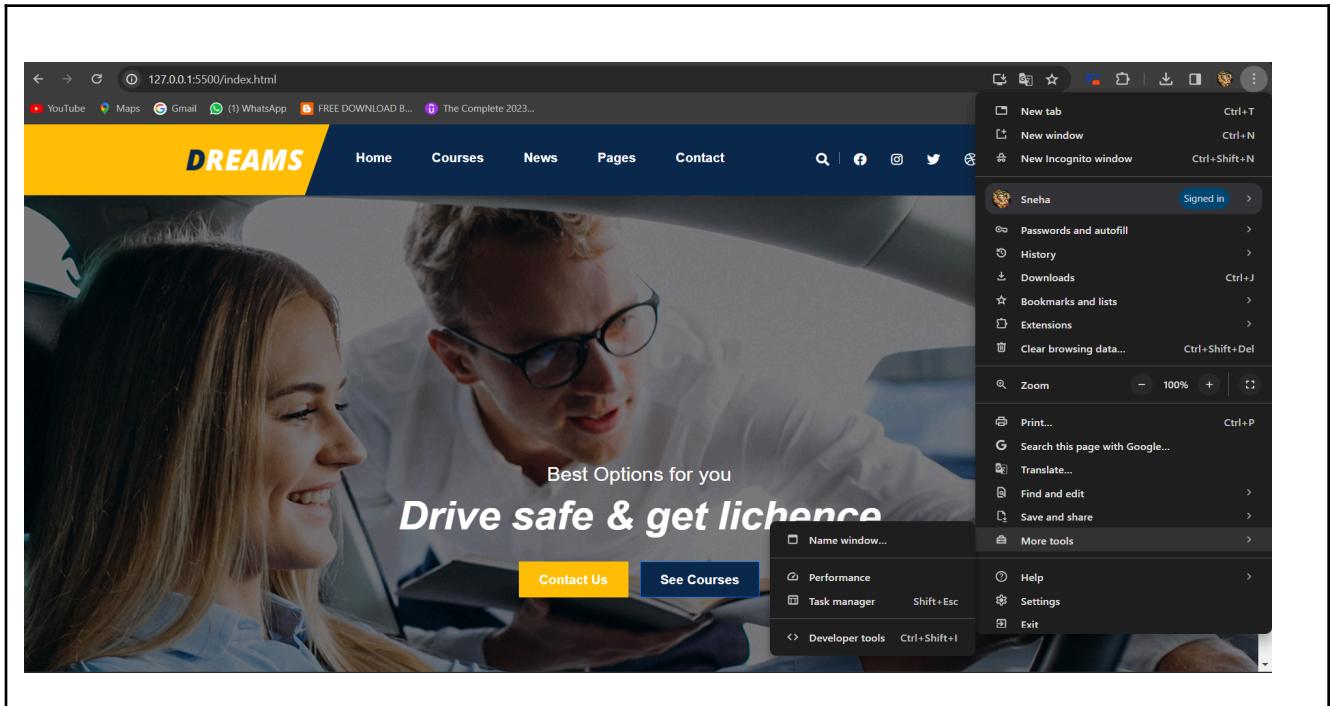
```

manifest.json

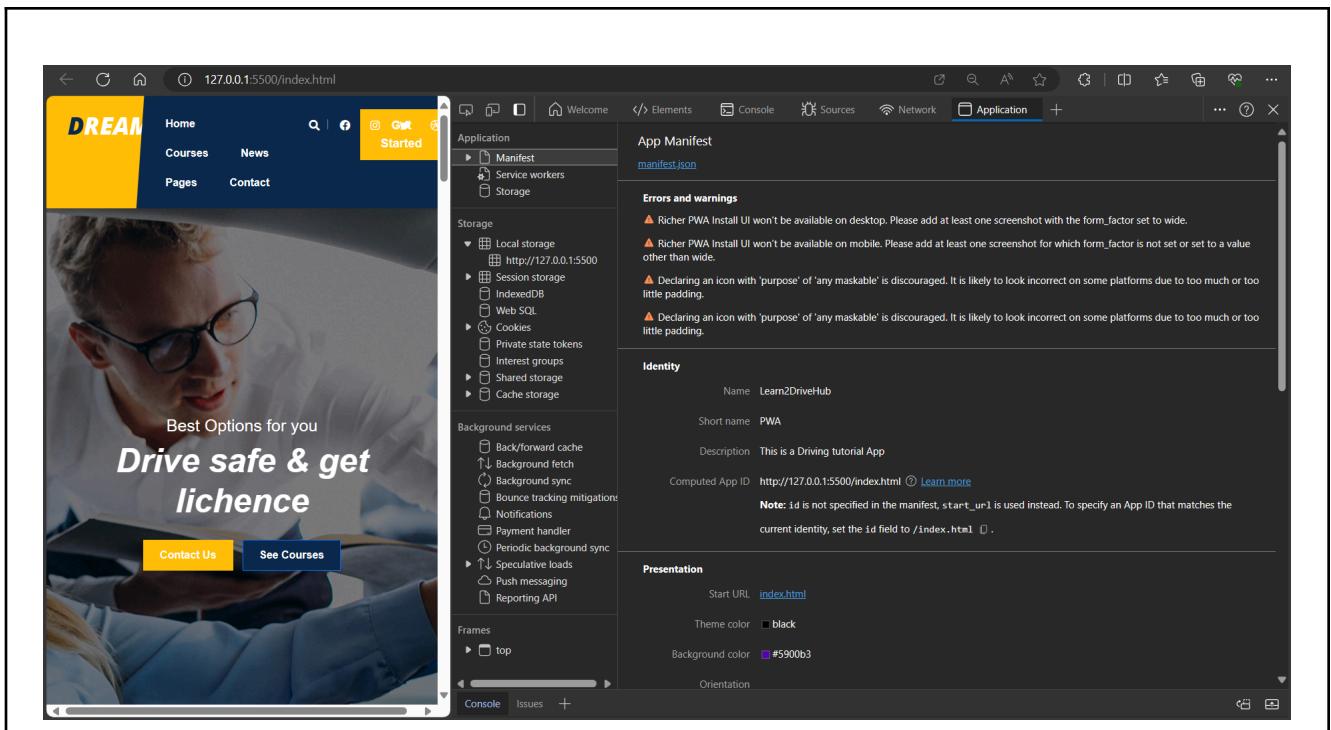
```
{
  "name": "Learn2DriveHub",
  "short_name": "PWA",
  "start_url": "index.html",
  "display": "standalone",
  "background_color": "#5900b3",
  "theme_color": "black",
  "scope": ".",
  "description": "This is a Driving tutorial App",
  "icons": [
    {
      "src": "images/icon-192x192.png",
      "sizes": "192x192",
      "type": "image/png",
      "purpose": "any maskable"
    },
    {
      "src": "images/icon-512x512.png",
      "sizes": "512x512",
      "type": "image/png",
      "purpose": "any maskable"
    }
  ]
}
```

Starting the server

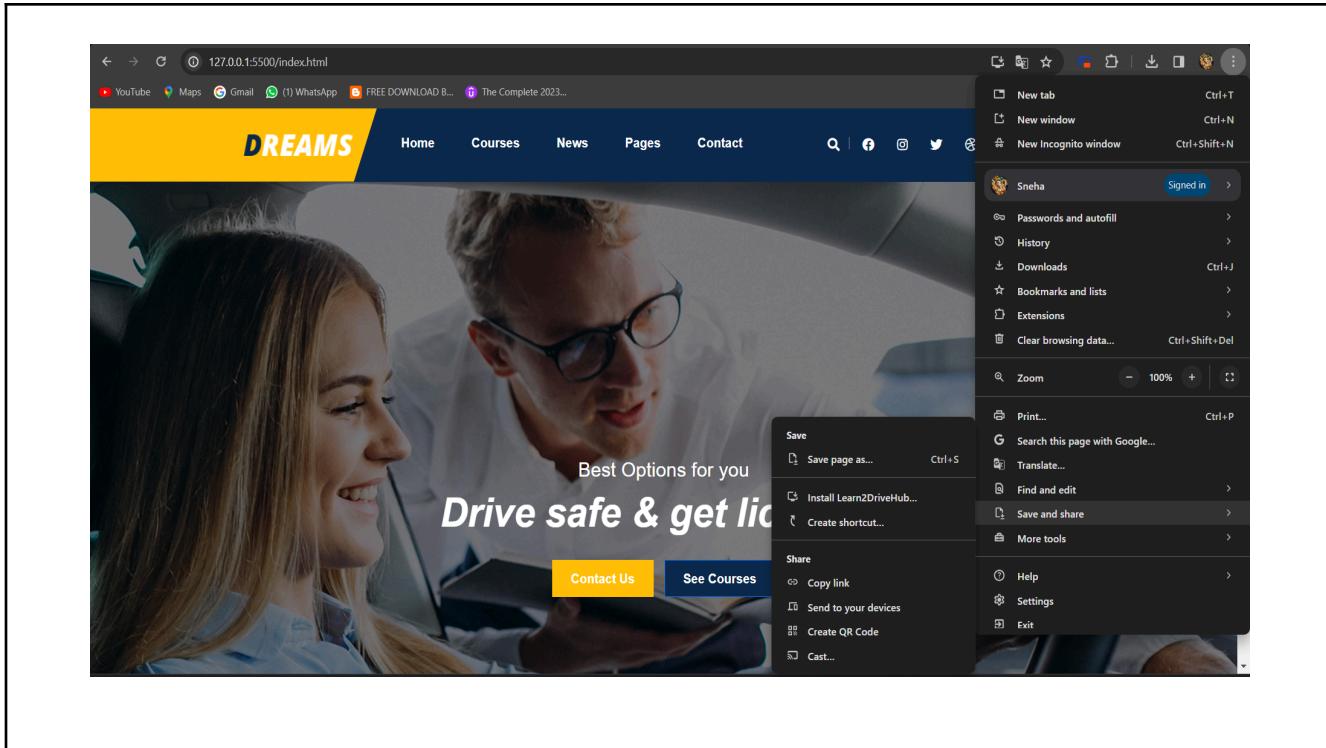
Going to the developers option



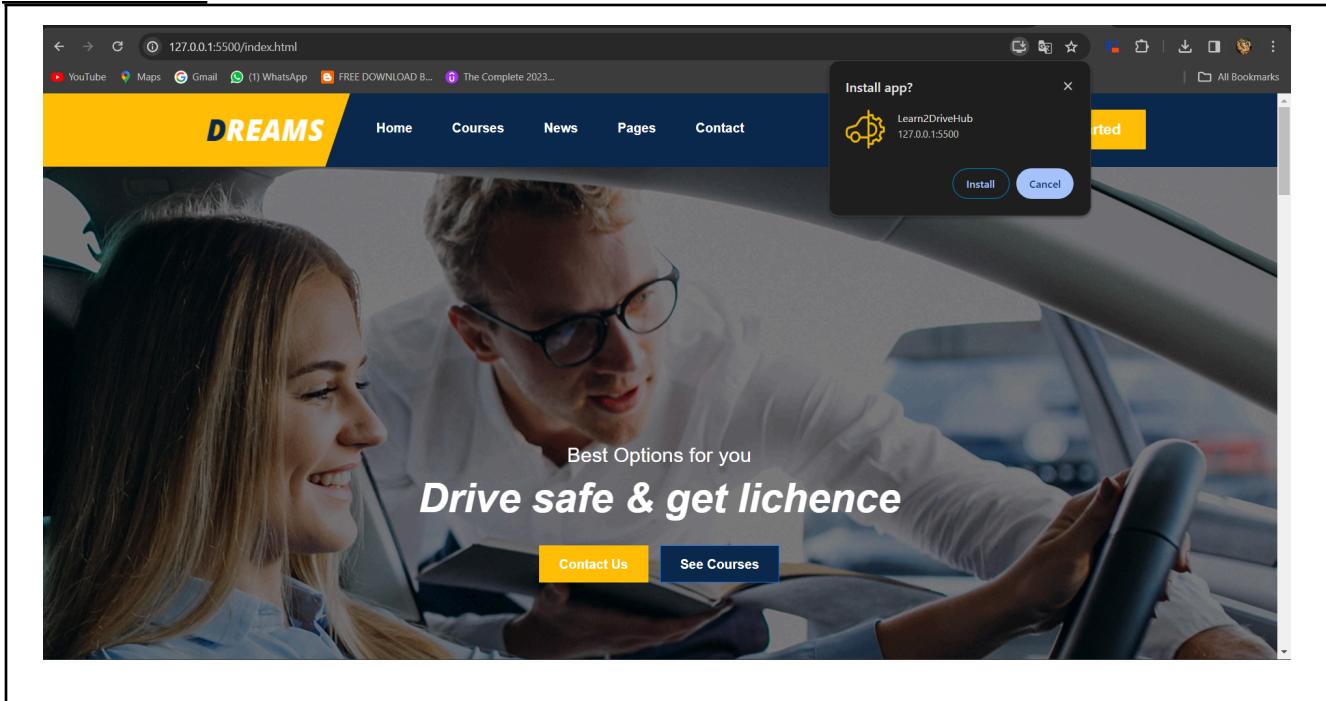
Application



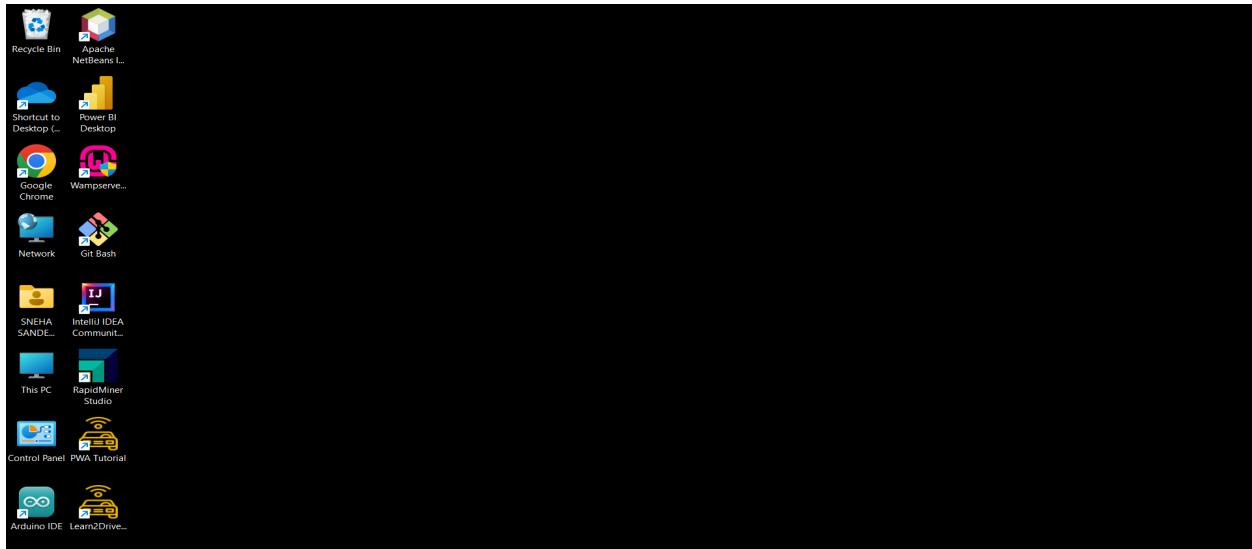
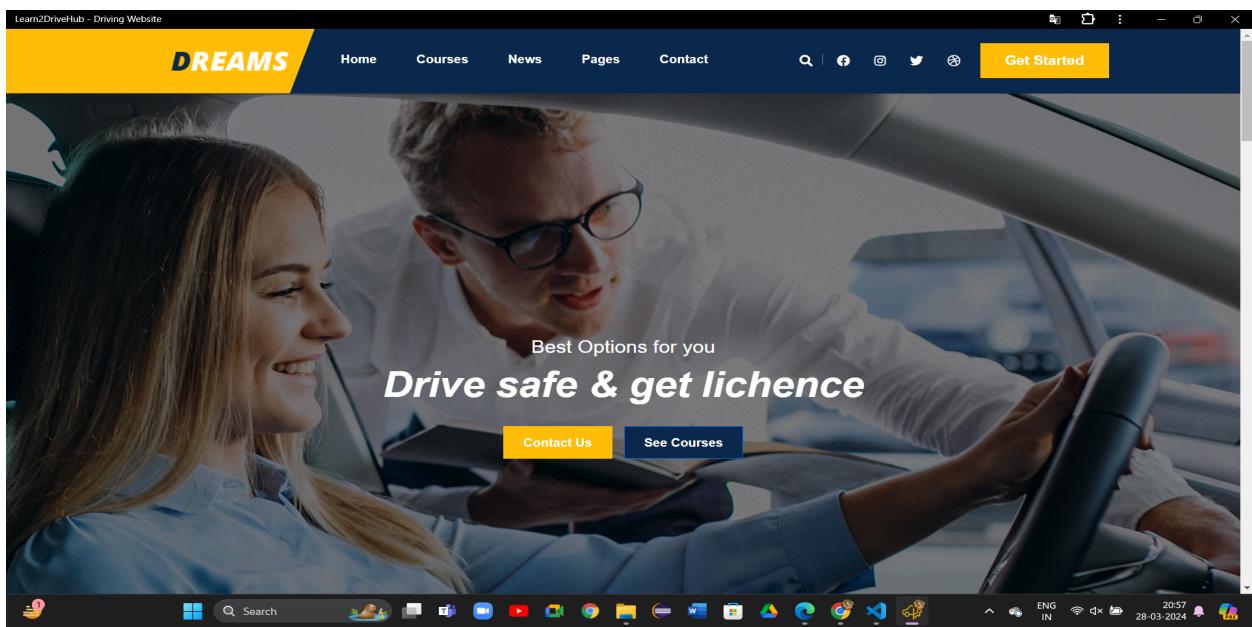
Click on 3 dots —>Save and Share—> install App



Click on install



This is the Driving app that is being created at taskbar and Desktop



Conclusion:

- Thus “add to homescreen feature was successfully implemented”
- We created a shortcut icon to open our websites app locally
- We also added an image to the app icon

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.	66
Name	Sneha Utekar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	15

Experiment 8

Aim: To code and register a service worker, and complete the install and activation process for a new service worker for the E-commerce PWA.

Theory:

Service Worker

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

Things to note about Service Worker:

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

What can we do with Service Workers?

- You can dominate Network Traffic
You can manage all network traffic of the page and do any manipulations. For example, when the page requests a CSS file, you can send plain text as a response or when the page requests an HTML file, you can send a png file as a response. You can also send a true response too.
- You can Cache
You can cache any request/response pair with Service Worker and Cache API and you can access these offline content anytime.
- You can manage Push Notifications
You can manage push notifications with Service Worker and show any information message to the user.
- You can Continue
Although Internet connection is broken, you can start any process with Background Sync of Service Worker.

What can't we do with Service Workers?

- You can't access the Window

You can't access the window, therefore, You can't manipulate DOM elements. But, you can communicate to the window through post Message and manage processes that you want.

- You can't work it on 80 Port

Service Worker just can work on HTTPS protocol. But you can work on localhost during development.

Service Worker Cycle

A service worker goes through three steps in its life cycle:

- Registration
- Installation
- Activation

Registration

To install a service worker, you need to register it in your main JavaScript code. Registration tells the browser where your service worker is located, and to start installing it in the background. Let's look at an example:

main.js

```
if ('serviceWorker' in navigator) {  
  navigator.serviceWorker.register('/service-worker.js') .then(function(registration) {  
    console.log('Registration successful, scope is:', registration.scope);  
  })  
  .catch(function(error) {  
    console.log('Service worker registration failed, error:', error);  
  });  
}
```

This code starts by checking for browser support by examining `navigator.serviceWorker`. The service worker is then registered with `navigator.serviceWorker.register`, which returns a promise that resolves when the service worker has been successfully registered. The scope of the service worker is then logged with `registration.scope`. If the service worker is already installed, `navigator.serviceWorker.register` returns the registration object of the currently active service worker.

The scope of the service worker determines which files the service worker controls, in other words, from which path the service worker will intercept requests. The default scope is the location of the service worker file, and extends to all directories below. So if `service-worker.js` is located in the root directory, the service worker will control requests from all files at this domain.

You can also set an arbitrary scope by passing in an additional parameter when registering.

For example:

main.js

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

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

If you want the service worker to control higher pages e.g. /app (without the trailing slash) you can indeed change the scope option, but you'll also need to set the Service-Worker-Allowed HTTP Header in your server config for the request serving the service worker script.

main.js

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

Installation

Once the browser registers a service worker, installation can be attempted. This occurs if the service worker is considered to be new by the browser, either because the site currently doesn't have a registered service worker, or because there is a byte difference between the new service worker and the previously installed one.

A service worker installation triggers an install event in the installing service worker. We can include an install event listener in the service worker to perform some task when the service worker installs. For instance, during the install, service workers can precache parts of a web app so that it loads instantly the next time a user opens it (see caching the application shell). So, after that first load, you're going to benefit from instant repeat loads and your time to interactivity is going to be even better in those cases. An example of an installation event listener looks like this:

service-worker.js

```
// Listen for install event, set callback
self.addEventListener('install', function(event) {
// Perform some task
});
```

Activation

Once a service worker has successfully installed, it transitions into the activation stage. If

there are any open pages controlled by the previous service worker, the new service worker enters a waiting state. The new service worker only activates when there are no longer any pages loaded that are still using the old service worker. This ensures that only one version of the service worker is running at any given time.

When the new service worker activates, an activate event is triggered in the activating service worker. This event listener is a good place to clean up outdated caches (see the Offline Cookbook for an example).

service-worker.js

```
self.addEventListener('activate', function(event) {
  // Perform some task
});
```

Once activated, the service worker controls all pages that load within its scope, and starts listening for events from those pages. However, pages in your app that were loaded before the service worker activation will not be under service worker control. The new service worker will only take over when you close and reopen your app, or if the service worker calls `clients.claim()`. Until then, requests from this page will not be intercepted by the new service worker. This is intentional as a way to ensure consistency in your site.

CODE AND OUTPUT:

1) Index.html :

```
<!DOCTYPE html>
<html>
<head>

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="X-UA-Compatible"
      content="ie=edge">

<title>Driving Website</title>

<meta name="apple-mobile-web-app-status-bar" content="#aa7700">
<meta name="theme-color" content="black">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css" rel="stylesheet"
      integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfI"
      crossorigin="anonymous">

      <!-- Font Awesome icon -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.2/css/all.min.css"
```

```
integrity="sha512-HK5fgLBL+xu6dm/Ii3z4xhlSUyZgTT9tuc/hSrtw6uzJOvgRr2a9jyxT1ely+B+xFAmJKVSTbpM/CuL7qxO8w=="
crossorigin="anonymous" />

<link rel="stylesheet" href="style.css">
<!-- Manifest File link -->
<link rel="manifest" href="manifest.json">

</head>

<body>
<!-- Header section -->
<div class="container-fluid header">
<div class="row">
<div class="col-3 bg-warning logoSection p-2">

</div>
<div class="col-9">
<div class="row">
<div class="col-6 menu">
<ul>
<li><a href="#">Home</a></li>
<li><a href="#">Courses</a></li>
<li><a href="#">News</a></li>
<li><a href="#">Pages</a></li>
<li><a href="#">Contact</a></li>
</ul>
</div>
<div class="col-6 social-icon">
<div class="row">
<div class="col-5 icon">
<a href=""><i class="fas fa-search"></i></a>
<a href=""><i class="fab fa-facebook"></i></a>
<a href=""><i class="fab fa-instagram"></i></a>
<a href=""><i class="fab fa-twitter"></i></a>
<a href=""><i class="fab fa-dribbble"></i></a>
</div>
<div class="col-4 header-btn">
<a href="#" class="btn btn-warning">Get Started</a>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

<section class="slider-section">
```

```

<div class="container">
  <div class="row">
    <div class="col-12 slider-content">
      <h3>Best Options for you</h3>
      <h1>Drive safe & get lichence</h1>
      <div class="slider-btn">
        <a href="#" class="btn btn-warning px-4">Contact Us</a>
        <a href="#" class="btn btn-primary px-4">See Courses</a>
      </div>
    </div>
  </div>
</div>
</section>

<!-- Feature Section started -->
<section class="feature">
  <div class="container">
    <div class="row">
      <div class="col-lg-4 col-md-4 col-sm-12">
        <div class="content content-center">
          <span>Why choose us ?</span>
          <h2>Our feature</h2>
          <p class="paragraph">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua viverra maecenas facilisis. </p>
          <a href="#" class="btn btn-own">See Courses</a>
        </div>
      </div>
      <div class="col-lg-4 col-md-4 col-sm-12">
        <div class="single-icon">
          
          <h3>Unlimited car support</h3>
        </div>
        <div class="single-icon">
          
          <h3>Driving School Insures</h3>
        </div>
      </div>
      <div class="col-lg-4 col-md-4 col-sm-12">
        <div class="single-icon content-center">
          
          <h3>Any Time Any Location</h3>
        </div>
      </div>
    </div>
  </div>
</section>

```

```

<!-- Welcome section started -->
<section class="container welcome">
  <div class="row">
    <div class="col-lg-6 col-md-12 col-sm-12 welcome-img">
      
    </div>
    <div class="col-lg-6 col-md-12 col-sm-12 shadow-wrap">
      <div class="content content-center">
        <span>Welcome to Online traffic school</span>
        <h2>looking for lessons?</h2>
        <p class="paragraph">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor  

          incididunt ut labore et dolore magna aliqua. Quis ipsum vidnas accumsan lacus vel facilisis.</p>
        <a href="#" class="btn btn-own">Learn More</a>
      </div>
    </div>
  </div>
</section>

<!-- Contact Form Started -->
<section class="application-form">
  <div class="container">
    <div class="content text-center">
      <span>Giving Best Options For You</span>
      <h2 class="p-3">Application Form</h2>
    </div>
    <form class="row form">
      <div class="col-lg-4 col-md-6 col-sm-12">
        <input type="text" placeholder="Your Name">
      </div>
      <div class="col-lg-4 col-md-6 col-sm-12">
        <input type="text" placeholder="Your Name">
      </div>
      <div class="col-lg-4 col-md-6 col-sm-12">
        <input type="text" placeholder="Your Name">
      </div>
      <div class="col-lg-4 col-md-6 col-sm-12">
        <input type="text" placeholder="Your Name">
      </div>
      <div class="col-lg-4 col-md-6 col-sm-12">
        <select name="" id="course">
          <option value="">Course Type</option>
          <option value="">Safe Driving Course</option>
          <option value="">Motorhome Drivers</option>
        </select>
      </div>
      <div class="col-lg-4 col-md-6 col-sm-12">
        <select name="" id="car">
          <option value="">Car Type</option>
        </select>
      </div>
    </form>
  </div>
</section>

```

```
<option value="">Hatchback</option>
<option value="">Sedan</option>
</select>
</div>
</form>
</div>
<div class="row">
<div class="col-12 text-center p-4">
<input type="submit" value="Submit" class="btn btn-own ">
</div>
</div>
</div>
</section>
<!-- Pricing section started here --&gt;
&lt;section class="pricing"&gt;
&lt;div class="container"&gt;
&lt;div class="content text-center"&gt;
&lt;span&gt;Get Special Offer&lt;/span&gt;
&lt;h2 class="p-3"&gt;Our Pricing&lt;/h2&gt;
&lt;/div&gt;
&lt;div class="row"&gt;
&lt;div class="col-lg-4 col-md-6 col-sm-12"&gt;
&lt;div class="box"&gt;
&lt;div class="box-heading"&gt;
&lt;span&gt;20% off&lt;/span&gt;
&lt;h1&gt;$ 199&lt;/h1&gt;
&lt;h5&gt;Personal Driving&lt;/h5&gt;
&lt;/div&gt;
&lt;div class="box-list-item"&gt;
&lt;p&gt;Full course theory&lt;/p&gt;
&lt;p&gt;Full driving course&lt;/p&gt;
&lt;p&gt;Training in first aid&lt;/p&gt;
&lt;p&gt;Practical sessions&lt;/p&gt;
&lt;p&gt;Psychological support&lt;/p&gt;
&lt;/div&gt;
&lt;div class="box-btn btn-own"&gt;
&lt;a href="#"&gt;Get Started&lt;/a&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;div class="col-lg-4 col-md-6 col-sm-12"&gt;
&lt;div class="box"&gt;
&lt;div class="box-heading"&gt;
&lt;span&gt;20% off&lt;/span&gt;
&lt;h1&gt;$ 199&lt;/h1&gt;
&lt;h5&gt;Personal Driving&lt;/h5&gt;
&lt;/div&gt;
&lt;div class="box-list-item"&gt;</pre>
```

```

<p>Full course theory</p>
<p>Full driving course</p>
<p>Training in first aid</p>
<p>Practical sessions</p>
<p>Psychological support</p>
</div>
<div class="box-btn btn-own">
    <a href="#">Get Started</a>
</div>

</div>
</div>
<div class="col-lg-4 col-md-6 col-sm-12">
    <div class="box">
        <div class="box-heading">
            <span>20% off</span>
            <h1>$ 199</h1>
            <h5>Personal Driving</h5>
        </div>
        <div class="box-list-item">
            <p>Full course theory</p>
            <p>Full driving course</p>
            <p>Training in first aid</p>
            <p>Practical sessions</p>
            <p>Psychological support</p>
        </div>
        <div class="box-btn btn-own">
            <a href="#">Get Started</a>
        </div>
    </div>
    </div>
</div>

</div>
</div>
</div>

</div>
</div>
</div>

</div>
</div>
</div>

<!-- Instructor section -->
<section class="instructor-section">
    <div class="container">
        <div class="row">
            <div class="col-lg-10 col-md-10 col-sm-12">
                <div class="content text-left">
                    <span>Our Great Team</span>
                    <h2>Our Instructors</h2>
                </div>
            </div>
            <div class="col-lg-2 col-md-2 col-sm-12 my-4">
                <a href="#" class="btn btn-own">View All</a>
            </div>
        </div>
    </div>
</section>

```

```
</div>
</div>
<div class="row">
  <div class="col-lg-6 col-md-6 col-sm-12 instructor-box">
    <div class="row instructor-wrap">
      <div class="col-5 instructor-img">
        
      </div>
      <div class="col-7 instructor-details">
        <h3>DAVID WARNER</h3>
        <span>Instructor</span>
        <p>Morbi accumsan ipsum velit. Nam nec tellus a odio tincidunt auctor.</p>
        <div class="instructor-social">
          <a href=""><i class="fab fa-facebook"></i></a>
          <a href=""><i class="fab fa-instagram"></i></a>
          <a href=""><i class="fab fa-twitter"></i></a>
          <a href=""><i class="fab fa-dribbble"></i></a>
        </div>
      </div>
    </div>
  </div>
  <div class="col-lg-6 col-md-6 col-sm-12 instructor-box">
    <div class="row instructor-wrap">
      <div class="col-5 instructor-img">
        
      </div>
      <div class="col-7 instructor-details">
        <h3>DAVID WARNER</h3>
        <span>Instructor</span>
        <p>Morbi accumsan ipsum velit. Nam nec tellus a odio tincidunt auctor.</p>
        <div class="instructor-social">
          <a href=""><i class="fab fa-facebook"></i></a>
          <a href=""><i class="fab fa-instagram"></i></a>
          <a href=""><i class="fab fa-twitter"></i></a>
          <a href=""><i class="fab fa-dribbble"></i></a>
        </div>
      </div>
    </div>
  </div>
  <div class="col-lg-6 col-md-6 col-sm-12 instructor-box">
    <div class="row instructor-wrap">
      <div class="col-5 instructor-img">
        
      </div>
      <div class="col-7 instructor-details">
        <h3>DAVID WARNER</h3>
        <span>Instructor</span>
        <p>Morbi accumsan ipsum velit. Nam nec tellus a odio tincidunt auctor.</p>
        <div class="instructor-social">
```



```

</div>
<div class="col-lg-3 col-md-6 col-sm-12 footer-details">
    <h3>USEFUL LINKS</h3>
    <a href="#">Home</a>
    <a href="#">Drupal Themes</a>
    <a href="#">WordPress Themes</a>
    <a href="#">Support</a>
    <a href="#">Services</a>
</div>
<div class="col-lg-3 col-md-6 col-sm-12 footer-details">
    
    <a href="#">Address : 15 Division Street, New York, NY 12032, United States of America</a>
    <a href="#">Phone : +0 (123) 456789</a>
    <a href="#">Email : Kaseo@gmail.com</a>
    <a href="#">Fax : +8 (123) 456 789</a>
</div>

</div>
<div class="footer-copyright">
    <span>Copyright &copy 2021 All rights reserved By Al Siam</span>
</div>
</div>
</footer>

<!-- Bootstrap Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js"
       integrity="sha384-b5kHyXgcpbZJO/tY9Ul7kGkfI1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOJ0"
       crossorigin="anonymous">
</script>
<script src="./app.js"></script>

</body>

</html>

```

2) app.js

```

if ('serviceWorker' in navigator) {
    window.addEventListener('load', () => {
        navigator.serviceWorker.register('/service-worker.js')
            .then(registration => {
                console.log('Service Worker registered with scope:', registration.scope);
            })
            .catch(error => {
                console.error('Service Worker registration failed:', error);
            });
    });
}

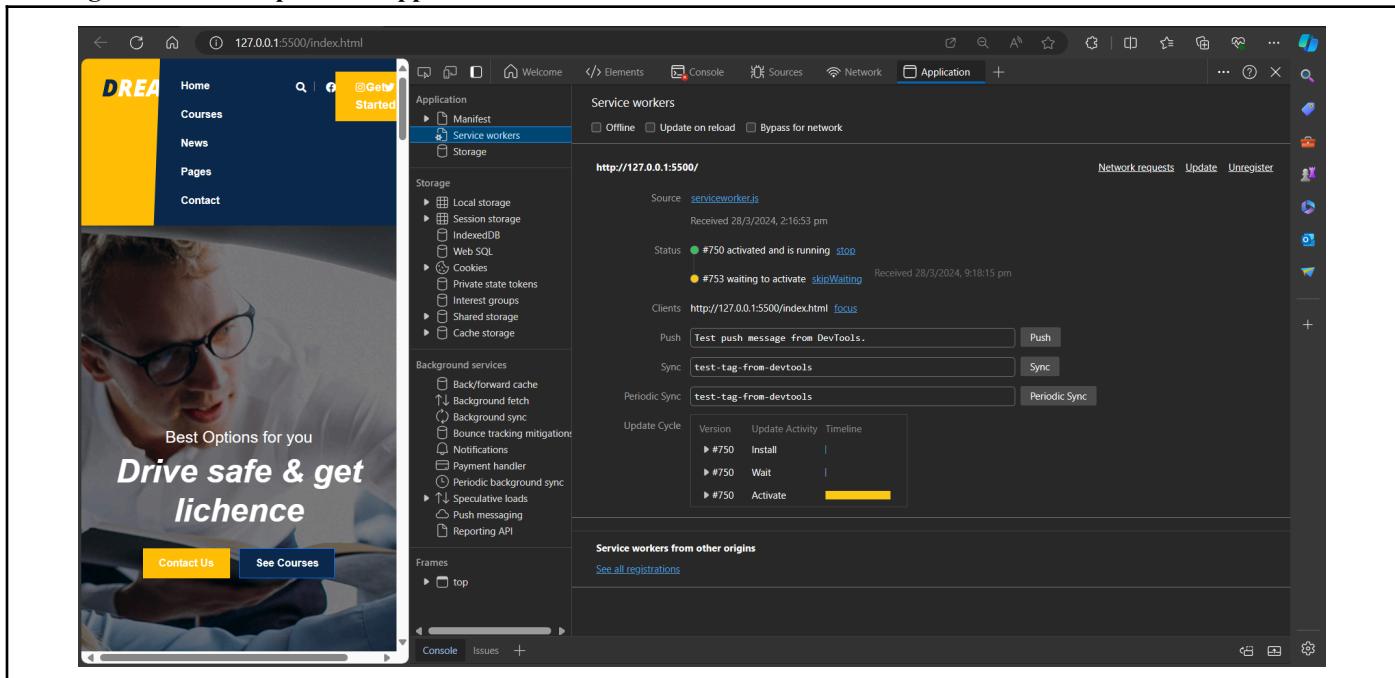
```

3) service-worker.js

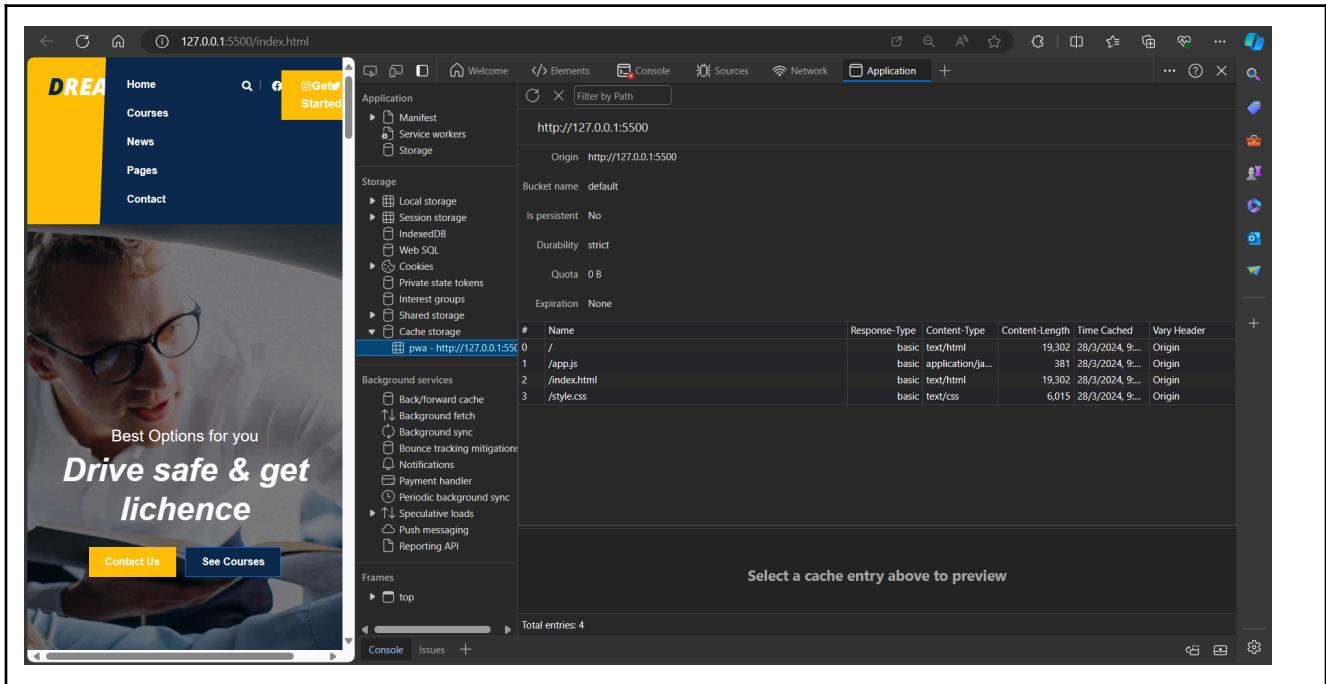
```

var cacheName = "pwa";
const assetsToCache = [
  '/',
  '/index.html',
  '/style.css',
  '/app.js',
];
self.addEventListener('install', event => {
  event.waitUntil(
    caches.open(cacheName)
      .then(cache => {
        return cache.addAll(assetsToCache);
      })
  );
});
self.addEventListener('activate', event => {
  event.waitUntil(
    caches.keys().then(cacheNames => {
      return Promise.all(
        cacheNames.filter(name => {
          return name !== cacheName;
        }).map(name => {
          return caches.delete(name);
        })
      );
    });
});
```

Now Right click —> Inspect —> Application —> Service workers



Now scroll down to Cache Storage



Conclusion:

- Registered a service worker, and completed the installation and activation process for a new service worker.
- Also checked the Cache Storage

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.	66
Name	Sneha Utekar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	15

Experiment 9

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

Theory:

Service Worker

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

Things to note about Service Worker:

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

Fetch Event

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

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

- CacheFirst - In this function, if the received request has cached before, the cached response is returned to the page. But if not, a new response requested from the network.
- NetworkFirst - In this function, firstly we can try getting an updated response from the network, if this process completed successfully, the new response will be cached and returned. But if this process fails, we check whether the request has been cached before or not. If a cache exists, it is returned to the page, but if not, this is up to you.

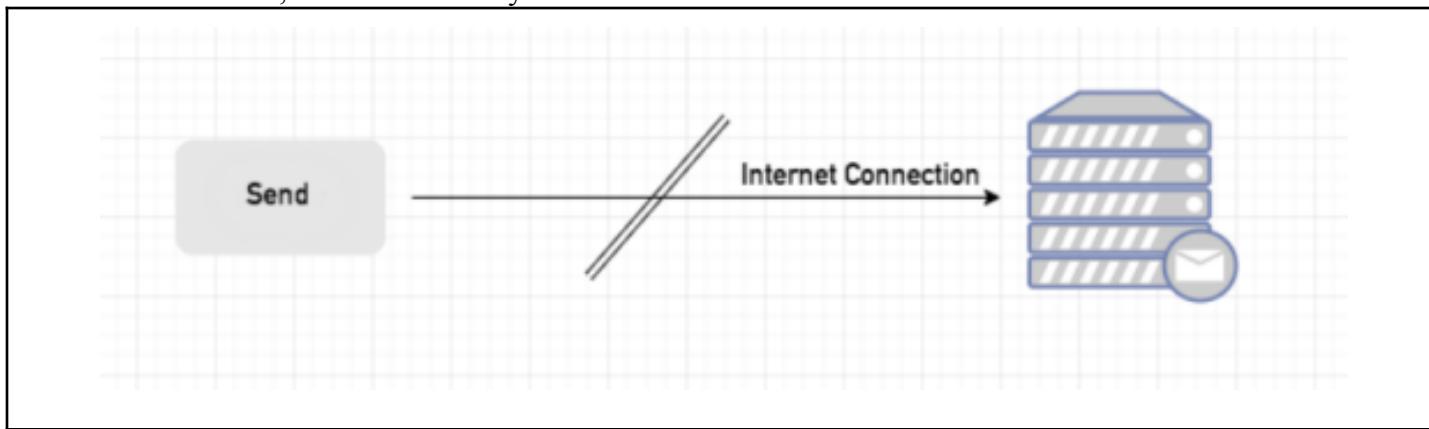
You can return dummy content or information messages to the page.

Sync Event

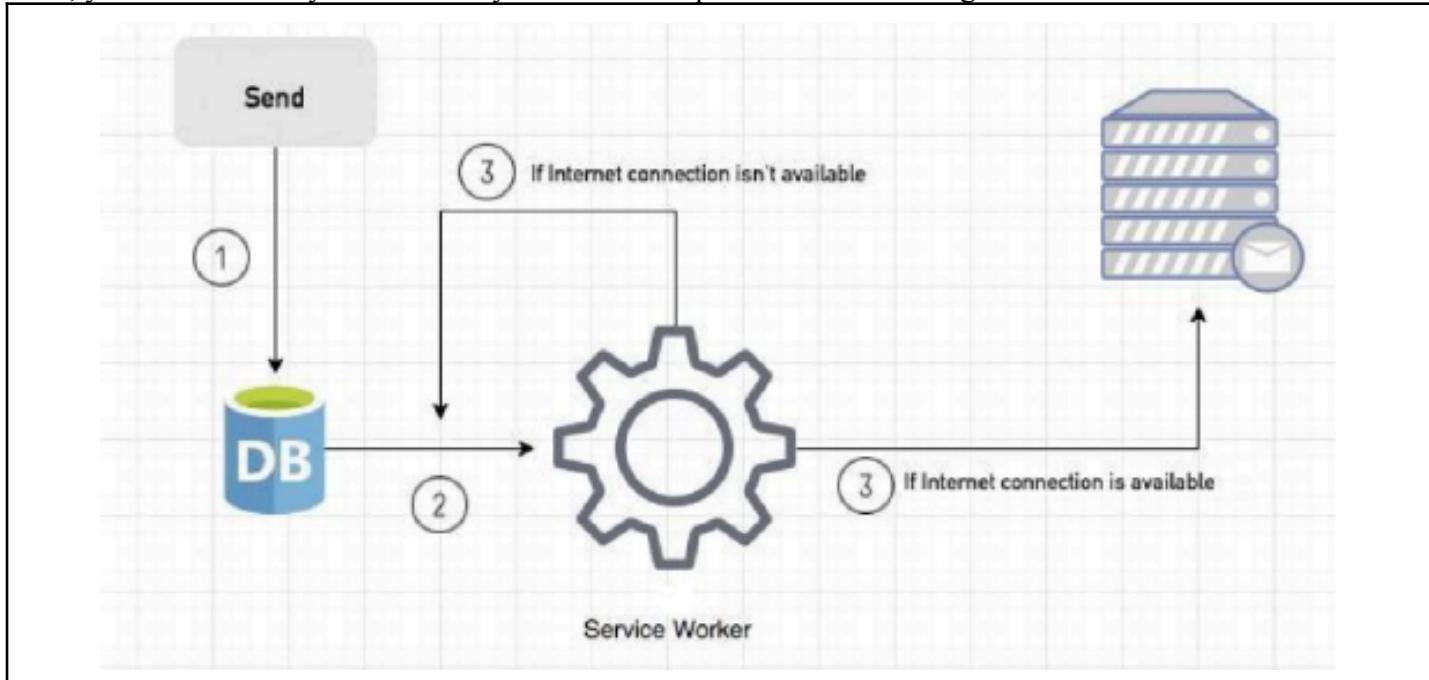
Background Sync is a Web API that is used to delay a process until the Internet connection is stable. We can adapt this definition to the real world; there is an e-mail client application that works on the browser and we want to send an email with this tool. Internet connection is broken while we are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

Here is a job for the Background Sync.

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



Here, you can create any scenario for yourself. A sample is in the following for this case.



1. When we click the “send” button, email content will be saved to IndexedDB.

2. Background Sync registration.

3. If the Internet connection is available, all email content will be read and sent to Mail Server.

If the Internet connection is unavailable, the service worker waits until the connection is available even though the window is closed. When it is available, email content will be sent to Mail Server.

You can see the working process within the following code block.

Event Listener for Background Sync Registration

Event Listener for sw.js

Push Event

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

We can check in the following example.

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

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

CODE AND OUTPUT:

Index.html:

```
<!DOCTYPE html>
<html>
<head>

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="X-UA-Compatible"
      content="ie=edge">

<title>Driving Website</title>
```

```
<meta name="apple-mobile-web-app-status-bar" content="#aa7700">
<meta name="theme-color" content="black">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css" rel="stylesheet"
      integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfI"
      crossorigin="anonymous">

      <!-- Font Awesome icon -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.2/css/all.min.css"
      integrity="sha512-HK5fgLBL+xu6dm/Ii3z4xhlSUyZgTT9tuc/hSrtw6uzJOvgRr2a9jyxxT1ely+B+xFAmJKVSTbp
      M/CuL7qxO8w=="
      crossorigin="anonymous" />

<link rel="stylesheet" href="style.css">
<!-- Manifest File link -->
<link rel="manifest" href="manifest.json">

</head>

<body>
<!-- Header section -->
<div class="container-fluid header">
    <div class="row">
        <div class="col-3 bg-warning logoSection p-2">
            
        </div>
        <div class="col-9">
            <div class="row">
                <div class="col-6 menu">
                    <ul>
                        <li><a href="#">Home</a></li>
                        <li><a href="#">Courses</a></li>
                        <li><a href="#">News</a></li>
                        <li><a href="#">Pages</a></li>
                        <li><a href="#">Contact</a></li>
                    </ul>
                </div>
                <div class="col-6 social-icon">
                    <div class="row">
                        <div class="col-5 icon">
                            <a href=""><i class="fas fa-search"></i></a>
                            <a href=""><i class="fab fa-facebook"></i></a>
                            <a href=""><i class="fab fa-instagram"></i></a>
                            <a href=""><i class="fab fa-twitter"></i></a>
                            <a href=""><i class="fab fa-dribbble"></i></a>
                        </div>
                        <div class="col-4 header-btn">
                            <a href="#" class="btn btn-warning">Get Started</a>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```

        </div>
    </div>
</div>
</div>
</div>
</div>
</div>

<section class="slider-section">
<div class="container">
<div class="row">
<div class="col-12 slider-content">
<h3>Best Options for you</h3>
<h1>Drive safe & get lichence</h1>
<div class="slider-btn">
<a href="#" class="btn btn-warning px-4">Contact Us</a>
<a href="#" class="btn btn-primary px-4">See Courses</a>
</div>
</div>
</div>
</div>
</div>
</section>

<!-- Feature Section started -->
<section class="feature">
<div class="container">
<div class="row">
<div class="col-lg-4 col-md-4 col-sm-12">
<div class="content content-center">
<span>Why choose us ?</span>
<h2>Our feature</h2>
<p class="paragraph">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua viverra maecenas facilisis. </p>
<a href="#" class="btn btn-own">See Courses</a>
</div>
</div>
<div class="col-lg-4 col-md-4 col-sm-12">
<div class="single-icon">

<h3>Unlimited car support</h3>
</div>
<div class="single-icon">

<h3>Driving School Insures</h3>
</div>
</div>
<div class="col-lg-4 col-md-4 col-sm-12">
<div class="single-icon content-center">

<h3>Any Time Any Location</h3>
</div>
</div>

```

```
</div>
</div>
</section>

<!-- Welcome section started -->
<section class="container welcome">
  <div class="row">
    <div class="col-lg-6 col-md-12 col-sm-12 welcome-img">
      
    </div>
    <div class="col-lg-6 col-md-12 col-sm-12 shadow-wrap">
      <div class="content content-center">
        <span>Welcome to Online traffic school</span>
        <h2>looking for lessons?</h2>
        <p class="paragraph">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor  
incididunt ut labore et dolore magna aliqua. Quis ipsum vidnas accumsan lacus vel facilisis.</p>
        <a href="#" class="btn btn-own">Learn More</a>
      </div>
    </div>
  </div>
</section>

<!-- Contact Form Started -->
<section class="application-form">
  <div class="container">
    <div class="content text-center">
      <span>Giving Best Options For You</span>
      <h2 class="p-3">Application Form</h2>
    </div>
    <form class="row form">
      <div class="col-lg-4 col-md-6 col-sm-12">
        <input type="text" placeholder="Your Name">
      </div>
      <div class="col-lg-4 col-md-6 col-sm-12">
        <input type="text" placeholder="Your Name">
      </div>
      <div class="col-lg-4 col-md-6 col-sm-12">
        <input type="text" placeholder="Your Name">
      </div>
      <div class="col-lg-4 col-md-6 col-sm-12">
        <input type="text" placeholder="Your Name">
      </div>
      <div class="col-lg-4 col-md-6 col-sm-12">
        <select name="" id="course">
          <option value="1">Frontend Development</option>
          <option value="2">Backend Development</option>
          <option value="3">Machine Learning</option>
          <option value="4">Data Science</option>
          <option value="5">Blockchain</option>
        </select>
      </div>
    </form>
  </div>
</section>
```

```

<option value="">Course Type</option>
<option value="">Safe Driving Course</option>
<option value="">Motorhome Drivers</option>
</select>
</div>
<div class="col-lg-4 col-md-6 col-sm-12">
<select name="" id="car">
<option value="">Car Type</option>
<option value="">Hatchback</option>
<option value="">Sedan</option>
</select>
</div>
</form>
</div>
<div class="row">
<div class="col-12 text-center p-4">
<input type="submit" value="Submit" class="btn btn-own ">
</div>
</div>
</div>
</section>

```

```

<!-- Pricing section started here -->
<section class="pricing">
<div class="container">
<div class="content text-center">
<span>Get Special Offer</span>
<h2 class="p-3">Our Pricing</h2>
</div>
<div class="row">
<div class="col-lg-4 col-md-6 col-sm-12">
<div class="box">
<div class="box-heading">
<span>20% off</span>
<h1>$ 199</h1>
<h5>Personal Driving</h5>
</div>
<div class="box-list-item">
<p>Full course theory</p>
<p>Full driving course</p>
<p>Training in first aid</p>
<p>Practical sessions</p>
<p>Psychological support</p>
</div>
<div class="box-btn btn-own">
<a href="#">Get Started</a>
</div>

```

```
</div>
</div>
<div class="col-lg-4 col-md-6 col-sm-12">
<div class="box">
<div class="box-heading">
<span>20% off</span>
<h1>$ 199</h1>
<h5>Personal Driving</h5>
</div>
<div class="box-list-item">
<p>Full course theory</p>
<p>Full driving course</p>
<p>Training in first aid</p>
<p>Practical sessions</p>
<p>Psychological support</p>
</div>
<div class="box-btn btn-own">
<a href="#">Get Started</a>
</div>

</div>
</div>
<div class="col-lg-4 col-md-6 col-sm-12">
<div class="box">
<div class="box-heading">
<span>20% off</span>
<h1>$ 199</h1>
<h5>Personal Driving</h5>
</div>
<div class="box-list-item">
<p>Full course theory</p>
<p>Full driving course</p>
<p>Training in first aid</p>
<p>Practical sessions</p>
<p>Psychological support</p>
</div>
<div class="box-btn btn-own">
<a href="#">Get Started</a>
</div>

</div>
</div>
</div>
```

```
<div class="container">
  <div class="row">
    <div class="col-lg-10 col-md-10 col-sm-12">
      <div class="content text-left">
        <span>Our Great Team</span>
        <h2>Our Instructors</h2>
      </div>
    </div>
    <div class="col-lg-2 col-md-2 col-sm-12 my-4">
      <a href="#" class="btn btn-own">View All</a>
    </div>
  </div>
  <div class="row">
    <div class="col-lg-6 col-md-6 col-sm-12 instructor-box">
      <div class="row instructor-wrap">
        <div class="col-5 instructor-img">
          
        </div>
        <div class="col-7 instructor-details">
          <h3>DAVID WARNER</h3>
          <span>Instructor</span>
          <p>Morbi accumsan ipsum velit. Nam nec tellus a odio tincidunt auctor.</p>
          <div class="instructor-social">
            <a href=""><i class="fab fa-facebook"></i></a>
            <a href=""><i class="fab fa-instagram"></i></a>
            <a href=""><i class="fab fa-twitter"></i></a>
            <a href=""><i class="fab fa-dribbble"></i></a>
          </div>
        </div>
      </div>
    </div>
    <div class="col-lg-6 col-md-6 col-sm-12 instructor-box">
      <div class="row instructor-wrap">
        <div class="col-5 instructor-img">
          
        </div>
        <div class="col-7 instructor-details">
          <h3>DAVID WARNER</h3>
          <span>Instructor</span>
          <p>Morbi accumsan ipsum velit. Nam nec tellus a odio tincidunt auctor.</p>
          <div class="instructor-social">
            <a href=""><i class="fab fa-facebook"></i></a>
            <a href=""><i class="fab fa-instagram"></i></a>
            <a href=""><i class="fab fa-twitter"></i></a>
            <a href=""><i class="fab fa-dribbble"></i></a>
          </div>
        </div>
      </div>
    </div>
  </div>
```

```

<div class="col-lg-6 col-md-6 col-sm-12 instructor-box">
  <div class="row instructor-wrap">
    <div class="col-5 instructor-img">
      
    </div>
    <div class="col-7 instructor-details">
      <h3>DAVID WARNER</h3>
      <span>Instructor</span>
      <p>Morbi accumsan ipsum velit. Nam nec tellus a odio tincidunt auctor.</p>
      <div class="instructor-social">
        <a href=""><i class="fab fa-facebook"></i></a>
        <a href=""><i class="fab fa-instagram"></i></a>
        <a href=""><i class="fab fa-twitter"></i></a>
        <a href=""><i class="fab fa-dribbble"></i></a>
      </div>
    </div>
  </div>
</div>
<div class="col-lg-6 col-md-6 col-sm-12 instructor-box">
  <div class="row instructor-wrap">
    <div class="col-5 instructor-img">
      
    </div>
    <div class="col-7 instructor-details">
      <h3>DAVID WARNER</h3>
      <span>Instructor</span>
      <p>Morbi accumsan ipsum velit. Nam nec tellus a odio tincidunt auctor.</p>
      <div class="instructor-social">
        <a href=""><i class="fab fa-facebook"></i></a>
        <a href=""><i class="fab fa-instagram"></i></a>
        <a href=""><i class="fab fa-twitter"></i></a>
        <a href=""><i class="fab fa-dribbble"></i></a>
      </div>
    </div>
  </div>
</div>
</div>
<!-- Footer Section -->
<footer class="footer">
  <div class="container">
    <div class="row footer-wrap">
      <div class="col-lg-3 col-md-6 col-sm-12 footer-details">
        <h3>Company</h3>
        <a href="#">About Us</a>
        <a href="#">Company</a>
        <a href="#">Press & Blog</a>
      </div>
    </div>
  </div>
</footer>

```

```

<a href="#">Privacy Policy</a>
<a href="#">Faq</a>
</div>
<div class="col-lg-3 col-md-6 col-sm-12 footer-details">
<h3>Courses</h3>
<a href="#">Winter driving</a>
<a href="#">Program for seniors</a>
<a href="#">Adult in car lesions</a>
<a href="#">Defensive driving</a>
<a href="#">Stick shit lessons</a>
</div>
<div class="col-lg-3 col-md-6 col-sm-12 footer-details">
<h3>USEFUL LINKS</h3>
<a href="#">Home</a>
<a href="#">Drupal Themes</a>
<a href="#">WordPress Themes</a>
<a href="#">Support</a>
<a href="#">Services</a>
</div>
<div class="col-lg-3 col-md-6 col-sm-12 footer-details">

<a href="#">Address : 15 Division Street, New York, NY 12032, United States of America</a>
<a href="#">Phone : +0 (123) 456789</a>
<a href="#">Email : Kaseo@gmail.com</a>
<a href="#">Fax : +8 (123) 456 789</a>
</div>
</div>
<div class="footer-copyright">
<span>Copyright &copy 2021 All rights reserved By Al Siam</span>
</div>
</div>
</footer>

<!-- Bootstrap Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-b5kHyXgcpbZJO/tY9Ul7kGkfIS0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOJ0"
crossorigin="anonymous">
</script>
<script src="/app.js"></script>
<script>
if ('Notification' in window) {
  Notification.requestPermission().then(function (result) {
    if (result === 'granted') {
      console.log('Notification permission granted');
    } else {
      console.warn('Notification permission denied');
    }
  });
}

```

```
        }  
    </script>
```

```
</body>
```

```
</html>
```

serviceworker.js:

```
self.addEventListener("install", function (event) {  
    event.waitUntil(preLoad());  
});  
self.addEventListener("fetch", function (event) {  
    event.respondWith(  
        checkResponse(event.request).catch(function () {  
            console.log("Fetch from cache successful!");  
            return returnFromCache(event.request);  
        })  
    );  
    console.log("Fetch successful!");  
    event.waitUntil(addToCache(event.request));  
});  
self.addEventListener("sync", (event) => {  
    if (event.tag === "syncMessage") {  
        console.log("Sync successful!");  
    }  
});  
self.addEventListener("push", function (event) { if (event && event.data) {  
    try {  
        var data = event.data.json();  
        if (data && data.method === "pushMessage") { console.log("Push notification sent");  
            self.registration.showNotification("Learn2Drive", { body: data.message,  
        });  
        }  
    }  
    } catch (error) {  
        console.error("Error parsing push data:", error);  
    }  
});  
var preLoad = function () {  
    return caches.open("offline").then(function (cache) { // caching index and important routes  
        return cache.addAll([  
            '/',  
            '/index.html',  
            '/style.css',  
            '/app.js',  
        ]);  
    });  
};
```

```
]);
});
};

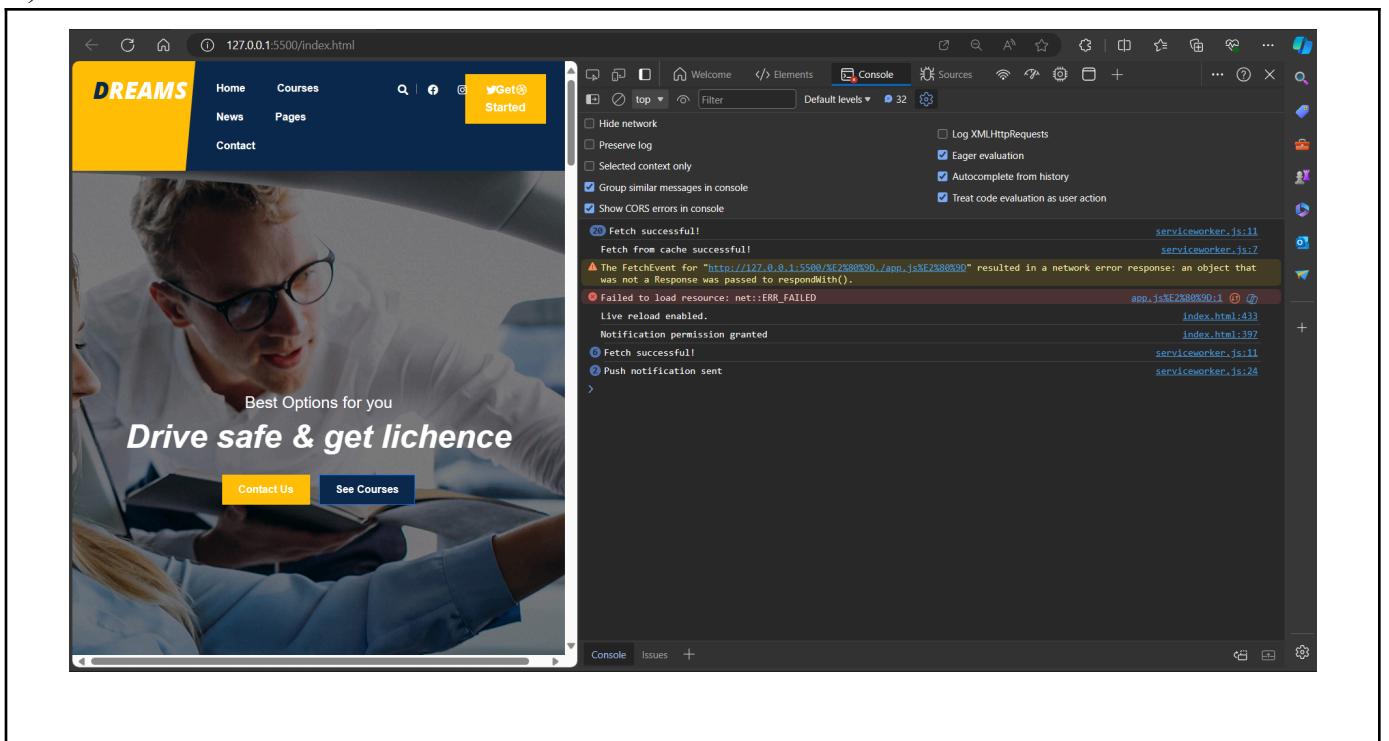
var checkResponse = function (request) { return new Promise(function (fulfill, reject) { fetch(request)
.then(function (response) {
if (response.status !== 404) {
fulfill(response);
} else {
reject(new Error("Response not found"));
}
})
.catch(function (error) {
reject(error);
});
});
};

var returnFromCache = function (request) {
return caches.open("offline").then(function (cache) {
return cache.match(request).then(function (matching) {
if (!matching || matching.status == 404) {
return cache.match("offline.html");
} else {
return matching;
}
});
});
};

var addToCache = function (request) {
return caches.open("offline").then(function (cache) {
return fetch(request).then(function (response) {
return cache.put(request, response.clone()).then(function () {
return response;
});
});
});
};

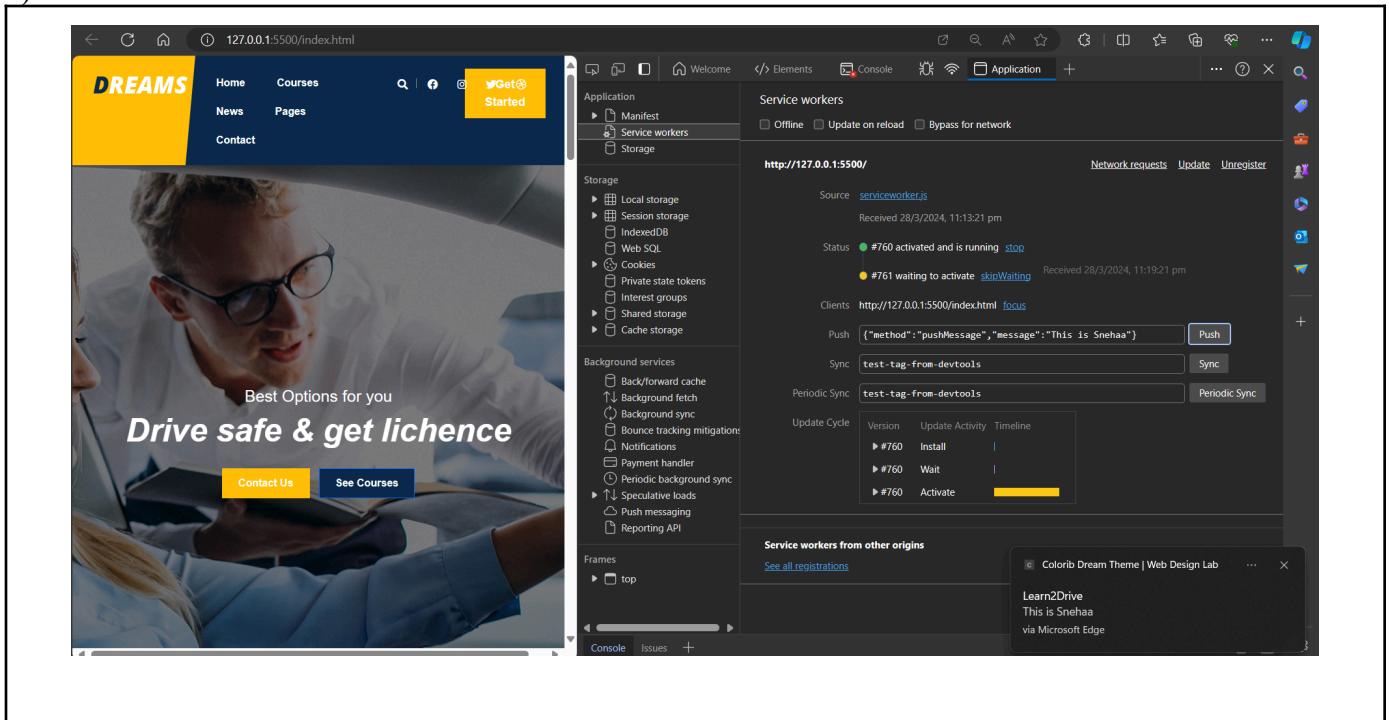
});
```

1) Fetch Event



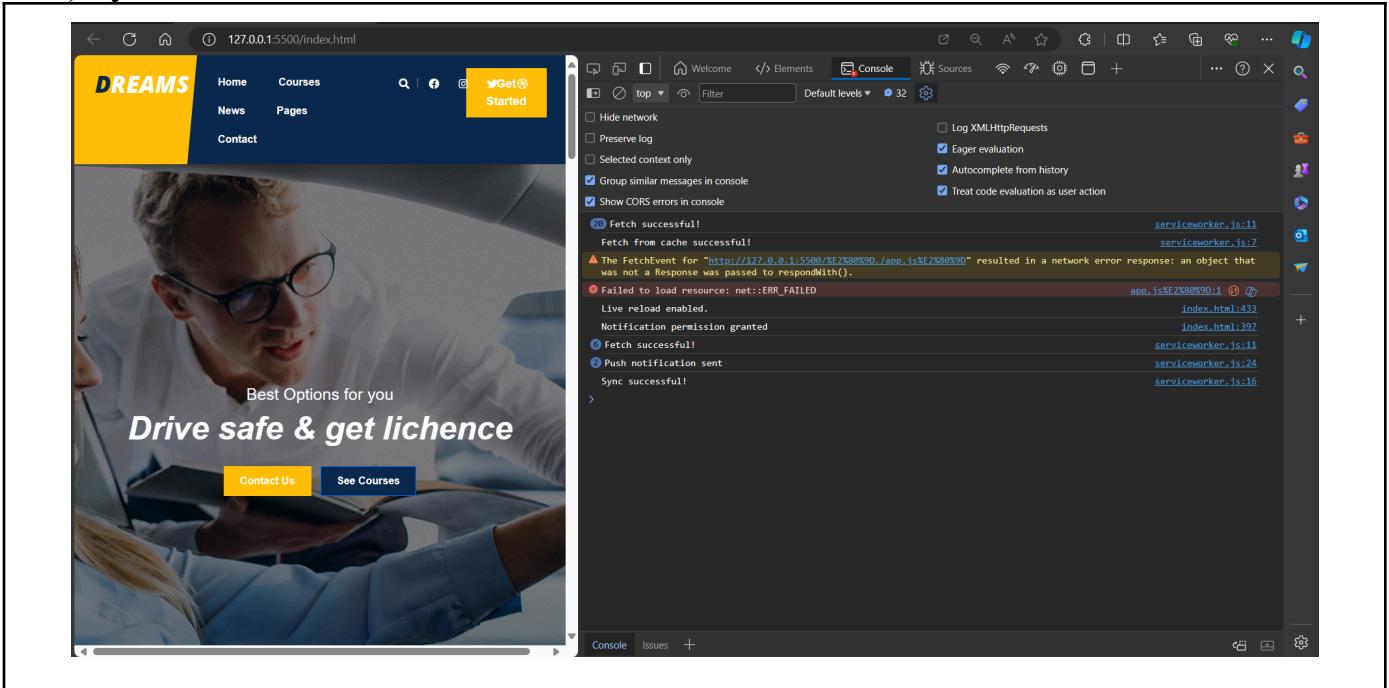
The screenshot shows a web browser window with a yellow header bar containing the word "DREAMS". The main content area features a man in glasses looking at a car's dashboard, with the text "Best Options for you" and "Drive safe & get licence". Below the image are two buttons: "Contact Us" and "See Courses". To the right of the browser window is the Microsoft Edge DevTools interface. The "Console" tab is selected, showing a list of log entries. One entry is highlighted in red, indicating a network error: "Failed to load resource: net::ERR_FAILED". Other entries include "Fetch successful!", "Fetch from cache successful!", and "Push notification sent". The DevTools sidebar on the left shows various developer tools like Application, Service workers, and Storage.

2) Push Event



This screenshot shows the same "DREAMS" website and DevTools interface as the previous one. However, the DevTools sidebar has been switched to the "Application" tab. In the "Service workers" section, it shows a service worker named "serviceworker.js" that was activated at 28/3/2024, 11:13:21 pm. A push message was received at 28/3/2024, 11:19:21 pm with the message "This is Snehaa". The "Network requests" section is also visible on the right side of the DevTools panel.

3) Sync Event



Conclusion:

In this experiment, we have successfully implemented service worker events like fetch, sync and push for my Learn2Drive PWA and found out output for above implementation.

MAD & PWA Lab

Journal

Experiment No.	10
Experiment Title.	To study and implement deployment of Ecommerce PWA to GitHub Pages.
Roll No.	66
Name	Sneha Utekar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	15

Experiment 10

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

Theory:

GitHub Pages

Public web pages are freely hosted and easily published. Public webpages hosted directly from your GitHub repository. Just edit, push, and your changes are live.

GitHub Pages provides the following key features:

- Blogging with Jekyll
- Custom URL
- Automatic Page Generator

Reasons for favoring this over Firebase:

- Free to use
- Right out of github
- Quick to set up

GitHub Pages is used by Lyft, CircleCI, and HubSpot.

GitHub Pages is listed in 775 company stacks and 4401 developer stacks.

Pros

- Very familiar interface if you are already using GitHub for your projects.
- Easy to set up. Just push your static website to the gh-pages branch and your website is ready.
- Supports Jekyll out of the box.
- Supports custom domains. Just add a file called CNAME to the root of your site, add an A record in the site's DNS configuration, and you are done.

Cons

- The code of your website will be public, unless you pay for a private repository.
- Currently, there is no support for HTTPS for custom domains. It's probably coming soon though.
- Although Jekyll is supported, plug-in support is rather spotty.

Firebase

The Realtime App Platform. Firebase is a cloud service designed to power real-time, collaborative applications. Simply add the Firebase library to your application to gain access to a shared data structure; any changes you make to that data are automatically synchronized with the Firebase cloud and with other clients within milliseconds.

Some of the features offered by Firebase are:

- Add the Firebase library to your app and get access to a shared data structure. Any changes made to that data are automatically synchronized with the Firebase cloud and with other clients within milliseconds.
- Firebase apps can be written entirely with client-side code, update in real-time out-of-the-box, interoperate well with existing services, scale automatically, and provide strong data security.
- Data Accessibility- Data is stored as JSON in Firebase. Every piece of data has its own URL which can be used in Firebase's client libraries and as a REST endpoint. These URLs can also be entered into a browser to view the data and watch it update in real-time.

Reasons for favoring over GitHub Pages:

- Realtime backend made easy
- Fast and responsive

Instacart, 9GAG, and Twitch are some of the popular companies that use Firebase. Firebase has a broader approval, being mentioned in 1215 company stacks & 4651 developer stacks

Pros

- Hosted by Google. Enough said.
- Authentication, Cloud Messaging, and a whole lot of other handy services will be available to you.
- A real-time database will be available to you, which can store 1 GB of data. ● You'll also have access to a blob store, which can store another 1 GB of data. ●
- Support for HTTPS. A free certificate will be provisioned for your custom domain within 24 hours.

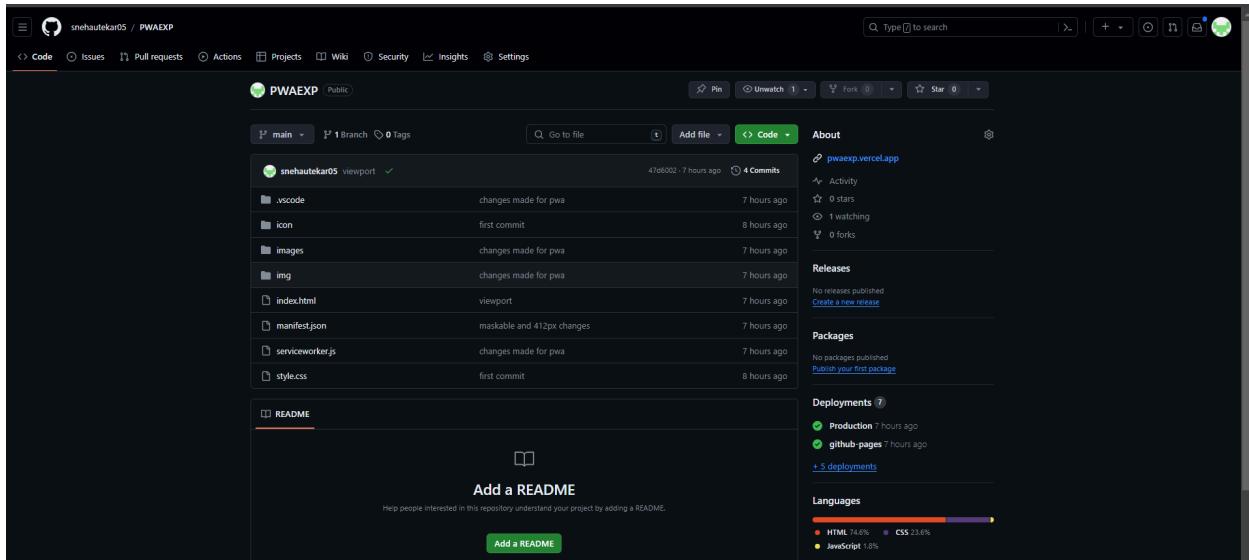
Cons

- Only 10 GB of data transfer is allowed per month. But this is not really a big problem, if you use a CDN or AMP.
- Command-line interface only.
- No in-built support for any static site generator.

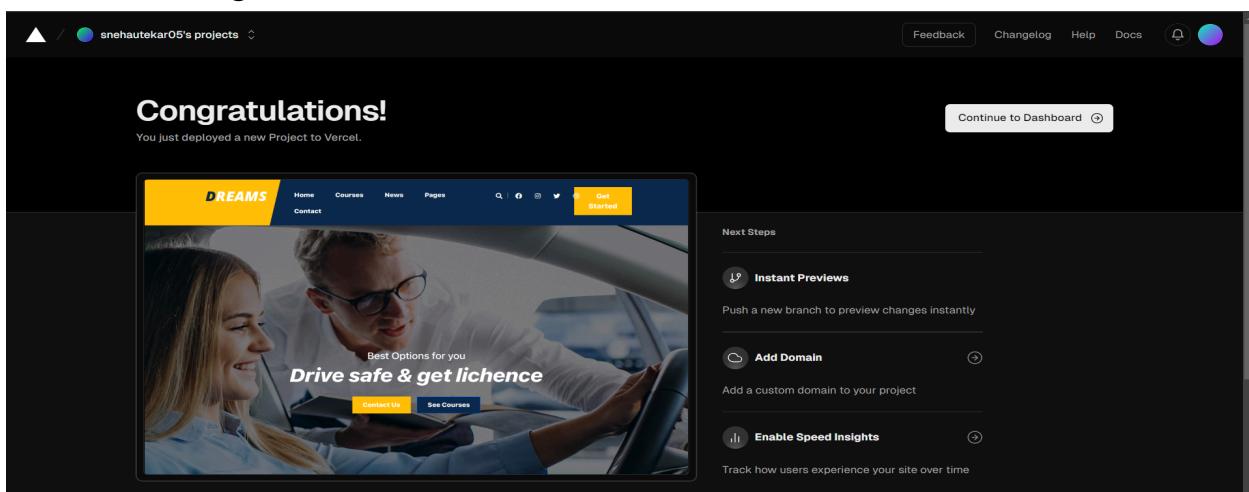
CODE AND OUTPUT:

GITHUB REPOSITORY LINK :<https://github.com/snehautekar05/PWAEXP>

GITHUB SCREENSHOT :



Website Hosting on Vercel:



HOSTED WEBSITE LINK : <https://snehautekar05.github.io/PWAEXP/>
<https://pwaexp-snehautekar05s-projects.vercel.app/>

Conclusion:

Successfully hosted website on GitHub Pages and Vercel

MAD & PWA Lab

Journal

Experiment No.	11
Experiment Title.	To use google Lighthouse PWA Analysis Tool to test the PWA functioning.
Roll No.	66
Name	Sneha Utekar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO6: Develop and Analyze PWA Features and deploy it over app hosting solution
Grade:	15

Experiment 11

Aim: To use google Lighthouse PWA Analysis Tool to test the PWA functioning.

Theory:

Google Lighthouse :

Google Lighthouse is a tool that lets you audit your web application based on a number of parameters including (but not limited to) performance, based on a number of metrics, mobile compatibility, Progressive Web App (PWA) implementations, etc. All you have to do is run it on a page or pass it a URL, sit back for a couple of minutes and get a very elaborate report, not much short of one that a professional auditor would have compiled in about a week.

The best part is that you have to set up almost nothing to get started. Let's begin by looking at some of the top features and audit criteria used by Lighthouse.

Key Features and Audit Metrics

Google Lighthouse has the option of running the Audit for Desktop as well as mobile version of your page(s). The top metrics that will be measured in the Audit are:

Performance: This score is an aggregation of how the page fared in aspects such as (but not limited to) loading speed, time taken for loading for basic frame(s), displaying meaningful content to the user, etc. To a layman, this score is indicative of how decently the site performs, with a score of 100 meaning that you figure in the 98th percentile, 50 meaning that you figure in the 75th percentile and so on.

PWA Score (Mobile): Thanks to the rise of Service Workers, app manifests, etc., a lot of modern web applications are moving towards the PWA paradigm, where the objective is to make the application behave as close as possible to native mobile applications. Scoring points are based on the Baseline PWA checklist laid down by Google which includes Service Worker implementation(s), viewport handling, offline functionality, performance in script-disabled environments, etc.

Accessibility: As you might have guessed, this metric is a measure of how accessible your website is, across a plethora of accessibility features that can be implemented in your page (such as the 'aria-' attributes like aria-required, audio captions, button names, etc.). Unlike the other metrics though, Accessibility metrics score on a pass/fail basis i.e. if all possible elements of the page are not screen-reader friendly (HTML5 introduced features that would make pages easy to interpret for screen readers used by visually challenged people like tag names, tags such as <section>, <article>, etc.), you get a 0 on that score. The aggregate of these scores is your Accessibility metric score.

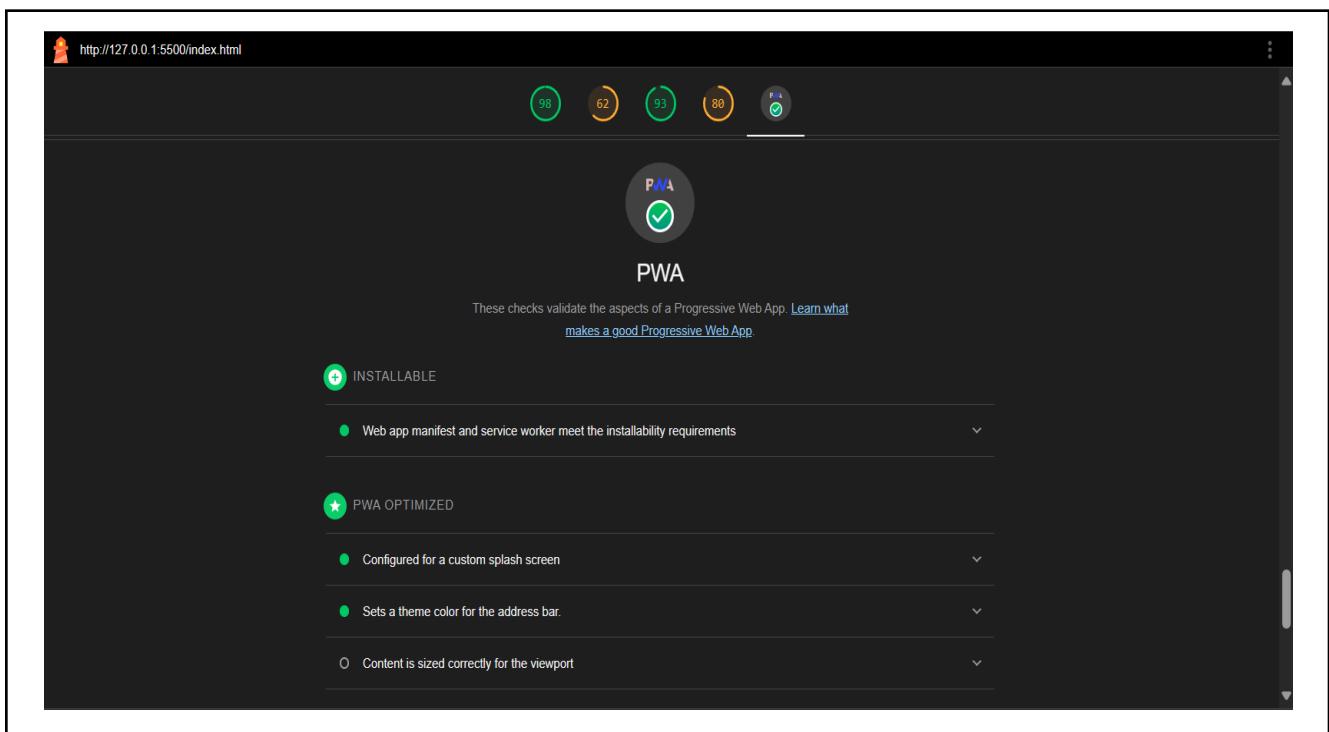
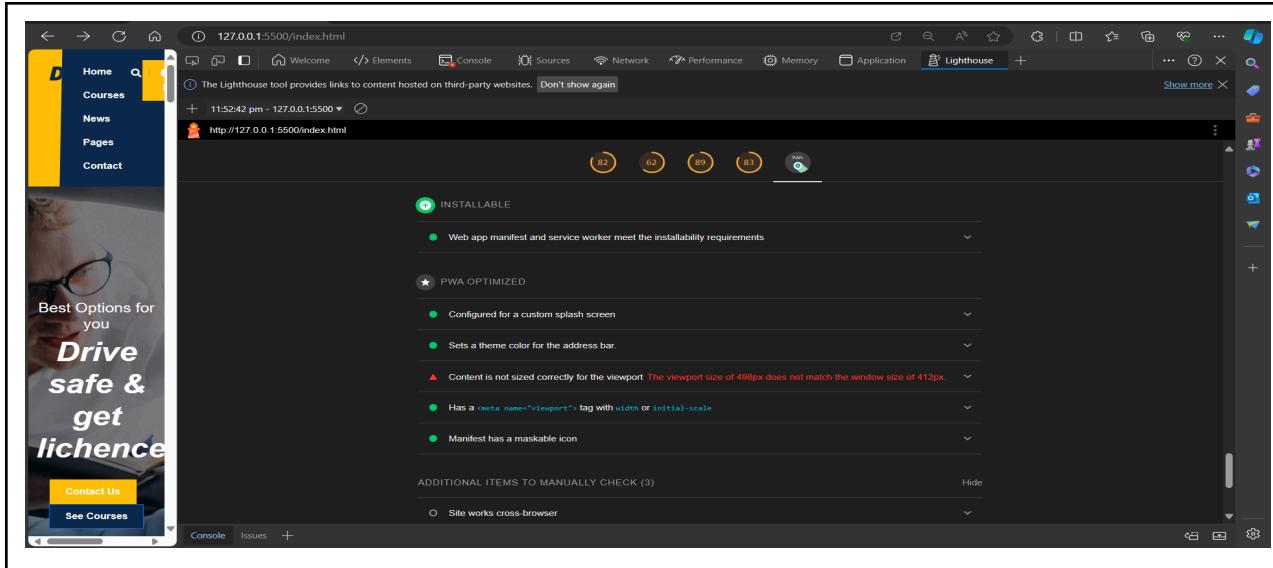
Best Practices: As any developer would know, there are a number of practices that have been deemed 'best' based on empirical data. This metric is an aggregation of many such points, including but not limited to: Use of HTTPS

Avoiding the use of deprecated code elements like tags, directives, libraries, etc. Password input with paste-into disabled Geo-Location and cookie usage alerts on load, etc.

CODE AND OUTPUT:

manifest.json

```
{  
  "name": "Learn2DriveHub",  
  "short_name": "PWA",  
  "start_url": "index.html",  
  "display": "standalone",  
  "background_color": "#5900b3",  
  "theme_color": "black",  
  "scope": ".",  
  "description": "This is a Driving tutorial App",  
  "icons": [  
    {  
      "src": "images/icon-192x192.png",  
      "sizes": "192x192",  
      "type": "image/png",  
      "purpose": "any maskable"  
    },  
    {  
      "src": "images/icon-512x512.png",  
      "sizes": "512x512",  
      "type": "image/png",  
      "purpose": "any maskable"  
    }  
  ]  
}
```



Conclusion:

- We analyzed the website with the help of lighthouse tool and found some issues with the website
- Hence we made changes in the manifest.json file we added "purpose":"any maskable" for maskable error face
- Also added a meta tag in index.html to resolve the theme error faced

MAD & PWA Lab

Journal

Experiment No.	Assignment-1
Assignment 1 Questions	<p>1. Flutter Overview: Explain the key features and advantages of using Flutter for mobile app development. Discuss how the Flutter framework differs from traditional approaches and why it has gained popularity in the developer community.</p> <p>2. Widget Tree and Composition: Describe the concept of the widget tree in Flutter. Explain how widget composition is used to build complex user interfaces. Provide examples of commonly used widgets and their roles in creating a widget tree.</p> <p>3. State Management in Flutter: Discuss the importance of state management in Flutter applications. Compare and contrast the different state management approaches available in Flutter, such as setState, Provider, and Riverpod. Provide scenarios where each approach is suitable.</p> <p>4. Firebase Integration in Flutter: Explain the process of integrating Firebase with a Flutter application. Discuss the benefits of using Firebase as a backend solution. Highlight the Firebase services commonly used in Flutter development and provide a brief overview of how data synchronization is achieved.</p>
Roll No.	66
Name	Sneha Utekar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	<p>LO1: Understand cross platform mobile application development using Flutter framework</p> <p>LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation</p> <p>LO3: Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS</p>
Grade:	5

Assignment-1

SNEHA UTEKAR
D15A
(66)

Assignment-1 (Flutter)

Q.1. Flutter Overview - Explain the key features and advantages of using flutter for mobile app development. Discuss how the flutter framework differs from traditional approaches and why it has gained popularity in the developer community.

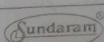
→ ① Key features of flutter:

- ⓐ single codebase for multiple platforms - flutter allows developer to write code and deploy it on both iOS and Android platforms.
- ⓑ Hot Reload -
This enables developers to instantly see the results of code changes they make.
- ⓒ Expressive UI:
Developers have the flexibility to create expressive and flexible UI's
- ⓓ Integration with other tools.
Flutter can easily integrate with other popular development tools and frameworks.

② Advantages of Flutter:

- ⓐ Faster Development -
uses single codebase for multiple platforms
- ⓑ Consistent UI across Platforms:
Widgets provide a consistent look and feel across different platforms.
- ⓒ Cost-Efficiency:
Developing and maintaining single database for both iOS

FOR EDUCATIONAL USE



f for android reduces development cost and resources

3) Differ from Traditional Approach:

- (a) Traditional approach uses a hierarchical structure for UI components whereas flutter uses a widget-based approach.
- (b) Flutter compiles to native ARM code, providing performance comparable to native applications
- (c) Hot Reloads allows to see changes made instantly.

Flutter's popularity is driven by increased productivity, a growing community, flexibility in UI design, cross-platform development capabilities and adoption by major companies.

Q. ② Widget Tree and composition's Describe the concept of widget tree in flutter. Explain how widget composition is used to build complex user interaction. Provide examples of commonly used widgets and their roles in creating a widget tree

→ Widget Tree:

- The Widget Tree is a hierarchical structure of widgets that defines the user interface of an application
- Every visual element, from simple components to complex layouts, is represented by a widget
- Widget can be categorized in 2 types:

(a) Stateless Widget:

It is immutable and cannot change over time.

Eg. Images, text.

⑥ Stateful Widget:

Widget that can change its state over time.

Eg. buttons, forms.

Widget composition:

- Widget composition in flutter involves combining multiple simple widgets to create more complex and compound widgets.
- This composability is a powerful concept that allows developers to build sophisticated user interfaces by nesting widgets within each other.

Commonly Used Widgets:

① Container:

A box model for padding, margin and decoration.

② Column and Row:

Layout widgets for arranging children vertically or horizontally.

③ Stack:

Overlapping widgets, allowing them to be layered on top of each other.

④ List View:

A scrollable list of widgets.

⑤ Grid View:

A scrollable grid of widgets.

⑦ Appbar:

A material of design app bar typically at the top of screen

⑧ Textfield:

An input field for users to enter text.

⑨ Button Widgets:

Interactive buttons for user actions

Q.3 State Management in Flutter: Discuss the importance of State management in Flutter applications, compare and contrast the different state management approaches available in Flutter, such as setState, Provider and Riverpod. Provide scenarios where each approach is suitable

→ Importance of State Management:

- State management is crucial in Flutter applications to handle dynamic nature of UI components
- It is used for handling changes in the application's data and updating the user interface accordingly
- In Flutter app, the UI is a function of the application state and managing state efficiently leads to better performance and maintainability.

Comparison of State Management Approaches:

① setState:

- suitable for small apps
- When the app's state is simple and doesn't require complex management
- Ideal for learning flutter or prototyping

(b) Provider:

- scoped and modular state.
- suitable for medium to large-sized apps.
- efficient for managing state across different parts of the app.
- Reduces boilerplate code compared to 'setState'.

(c) Riverpod:

- Advanced State Management.
- Excellent for large and complex applications.
- offers dependency injection, making it scalable and maintainable.
- Allows more flexibility and control over state management.

Scenarios in Each Approach:

• SetState:
small apps, prototypes, or when simplicity is priority.

• Provider:

Medium-sized apps, where a scoped approach to state management is needed.

• Riverpod:

Large and complex applications that required advanced state management features.

Q4) Firebase Integration in Flutter: Explain the process of integrating Firebase with a Flutter application. Discuss the benefits of using Firebase as a backend solution. Highlight the Firebase services commonly used in Flutter development and provide a brief overview of how data synchronization is achieved.

→ Firebase Integration in Flutter:

Process of Integrating Firebase:

① Setup Firebase Project:

- Create a new project on the Firebase console
- Add your Flutter app to the project.

② Configure Flutter Project:

- Add Firebase configuration files to your Flutter project.

③ Add Firebase Dependencies:

- Include necessary Firebase dependencies in your 'pubspec.yaml' file.

④ Initialize Firebase in Flutter:

- Initialize Firebase in your Flutter App, typically in the 'main.dart' file.

Benefits of Using Firebase:

• Realtime Database:

Provides a NoSQL database with real-time synchronization.

• Authentication:

Offers easy-to-use authentication services.

- Cloud Functions:

Allows serverless backend logic.

- Cloud Storage:

Storage solutions for images, videos and other files.

Firebase services:

- Firestore: NoSQL database for scalable & real-time data.
- Authentication: User Authentication with various providers.
- Cloud Functions: Serverless backend logic.
- Cloud Storage: For storing and serving user-generated content.

Data synchronization:

- Firebase's real-time database and Firestore provide automatic data synchronization.
- changes in database trigger updates to connected clients in real-time.

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 apps 2. Define responsive web design and explain its importance in the context of Progressive Web Apps. Compare and contrast responsive, fluid, and adaptive web design approaches. 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.	66
Name	Sneha Utekar
Class	D15A
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:	4

Assignment- 2

SNEHA UTEKAR

D15A

(66)

ASSIGNMENT - 2

- 1) Define progressive web app (PWA) and explain its significance in modern web development. Discuss the key characteristics that differentiate PWA from traditional mobile app.
- i) A progressive web app (PWA) is a type of web application that uses modern web capabilities to deliver an app-like experience to users. PWAs are built using standard web technologies such as HTML, CSS, & JS but are designed to provide more native-like experience for users, especially on mobile devices.

Key characteristics of PWA:

(B) (4)

① Progressive Enhancement:

They are built using progressive enhancement principle, meaning they provide a basic experience for all users and then enhance meaning based on the capabilities of the user's device and browser.

② Responsive Design:

PWAs are designed to feel and behave like native mobile apps. They include multiple features such as animations, gestures and transitions to provide a more immersive user experience.

③ Reliability:

PWAs are designed to be reliable, even when users are offline or have slow or unreliable internet connection. They cache content and resources to ensure that the app remains functional.

④ Fast Performance

⑤ Discoverability:

PWAs are easily discoverable and shareable, as they are built using web technologies such as and can be accessed via a URL.

⑥ Security -

PWA's are served over HTTPS to ensure security.

⑦ Cross platform compatibility

Q2) Define responsive web design and explain its importance in the context of progressive web app. compare and contrast responsive, fluid and adaptive web design approaches

- • Responsive Web design is a web design approach that ensures a website adapts its layout and content to seamlessly function and display well across various screen sizes, from desktop to smartphones.

• Importance of PWA

① Foundation for App-like Experience:

RWD ensures proper layout and navigation across devices, mimicking the adaptability of native apps.

② Accessibility:

A RWD makes PWA accessible to a wider audience using diverse devices.

Responsive v/s Fluid v/s Adaptive

<u>Feature</u>	<u>Responsive</u>	<u>Fluid</u>	<u>Adaptive</u>
① Layout	flexible, adapt to any screen size	continuously adjusts based on relative units.	use multiple fixed width layouts.
② Development effort	Moderate	less effort for basic layout	More efforts to create multiple layout.
③ Control	Good control over overall layout.	less control over element appearance on extreme sizes	High control over layout for each device category.
④ Suitability for PWAs	Ideal foundation due to its versatility.	can be used but may require adjustment for optimal experience.	Less common for PWA's due to development overhead

Q3) Describe the life cycle of service workers, including registration, installation and activation phases.

- Service workers are crucial components of progressive Web Apps (PWAs) that enable features such as offline support, push notifications, and background synchronization.
- The lifecycle of service worker involves several key phases: registration, installation and activation.

① Registration:

- 1st phase of lifecycle. occurs in the main Javascript file of web application.
- Registration is done using the navigator.serviceWorker.register() method, which takes the path to service worker file as a file parameter.
- This method returns a promise that resolves to a service worker registration object, which can be used to interact with service worker.

② Installation:

- Once service worker registered, browser downloads and starts installation process of service worker script.
- During installation, the service worker script is executed and the 'install' event is fired. Inside 'Install' event handler, developers can define what resources should be cached for offline access using caching strategies or precaching.
- Installation process is critical for setting up the service worker's initial cache and preparing it to take control of web application.

③ Activation:

- After installation phase, Activation phase
- It occurs when service worker is ready to take control of web application.
- During activation, 'activate' event is fired, giving service worker an opportunity to cleanup any old caches or perform other tasks necessary for application functionality.
- Developers can use 'activate' event handler to manage cache cleanup, versioning and other activation related tasks.
- Service remains active until it is explicitly unregistered or updated. It's important to note that service workers run separately from main web page and operate on a different thread allowing them to perform tasks like network interception and caching without blocking UI.

Q4) Explain the use of Indexed DB in the service worker for data storage.

- IndexedDB is a low-level API in web browsers that allow web applications to store and retrieve large amounts of structured data on the client side.
- It is particularly useful for offline web applications and service workers.
- How indexedDB used in the context of service workers, for data storage.

① Offline Data Storage:

Service workers are background scripts that run separately from web pages and enable features like push notifications & offline functionality. IndexedDB provides a way for service workers to store data.

workers to store data locally on the user's device, allowing web application to work offline and improve performance by caching frequently accessed data.

② Asynchronous Database Operations:

- IndexedDB operates asynchronously, meaning that database operations like adding, retrieving, updating or deleting data do not block the main thread of the web application. This asynchronous nature is well-suited for service workers which need to handle multiple tasks simultaneously without affecting user's experience.

③ Structured Data storage: IndexedDB stores data in key-value pairs within object stores similar to how data is organized in a database. Service workers can create, access and manage object stores to store different types of data required by the web application, such as user preferences, cached content or application state.

④ Data caching and Retrieval:

- Service workers can be used indexedDB to cache resources like HTML, CSS, Javascript and images when the user is online.
- This cached data can then be retrieved and served from local IndexedDB storage when the user is offline providing seamless experience even without an internet connection.