



Department of Information Technology

CERTIFICATE

This is to certify that Ria Chaudhari 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 2024-2025.

Lab Assistant

Subject Teacher

Mrs. Kajal Joseph

Principal

Head of Department

Dr. Mrs. Shalu Chopra

Name of the Course : MAD & PWA Lab
Year/Sem/Class : D15A/D15B

Course Code : ITL604
A.Y.: 24-25

Faculty Incharge : Mrs. Kajal Joseph.
Lab Teachers : Mrs. Kajal Joseph.
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 Analyse PWA Features and deploy it over app hosting solutions	L3, L4

Index

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

MAD & PWA Lab

Journal

Experiment No.	01
Experiment Title.	To install and configure the Flutter Environment
Roll No.	7
Name	Ria Chaudhari
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO1: Understand cross platform mobile application development using Flutter framework
Grade:	

MPL Experiment 1

AIM: - Installation and Configuration of Flutter Environment.

Step 1: Go to the official Flutter website: <https://docs.flutter.dev/get-started/install>

The screenshot shows the Flutter documentation website's 'Install' page. On the left, a sidebar menu includes 'Get started', 'Set up Flutter', 'Learn Flutter', 'Stay up to date', 'App solutions', 'User interface', 'Introduction', 'Widget catalog', 'Layout', 'Adaptive & responsive design', 'Design & theming', 'Interactivity', 'Assets & media', and 'Navigation & routing'. The main content area has a heading 'Choose your development platform to get started' with sub-sections 'Get started' and 'Install'. It features four cards for 'Windows Current device', 'macOS', 'Linux', and 'ChromeOS'. A note about developing in China is present, stating: 'If you want to use Flutter in China, check out using Flutter in China. If you're not developing in China, ignore this notice and follow the other instructions on this page.' Below the note is a Chinese translation: '如果你正在中国的网络环境下配置 Flutter, 请参考 在中国网络环境下使用 Flutter 文档.' At the bottom, a small note says 'Unless stated otherwise, the documentation on this site reflects the latest stable version of Flutter. Page last updated on 2024-07-07. View source or report an issue.'

Step 2: To download the latest Flutter SDK, click on the Windows icon > Android

The screenshot shows the 'Install' page again, but with a different focus. The main content area has a heading 'Choose your first type of app' with sub-sections 'Get started', 'Install', and 'Windows'. It features three cards for 'Android Recommended' (with a phone icon), 'Web' (with a laptop icon), and 'Desktop' (with a monitor icon). A note below the cards states: 'Your choice informs which parts of Flutter tooling you configure to run your first Flutter app. You can set up additional platforms later. If you don't have a preference, choose [Android](#).' A note about developing in China is also present, identical to the one in the previous screenshot.

Step 3: For Windows, download the stable release (a .zip file).

Step 4: Extract the ZIP file to a folder (e.g., C:\flutter)

Select a Destination and Extract Files

Files will be extracted to this folder:

C:\Users\Ria Chaudhari\Downloads\flutter_windows_3.27.3-stable

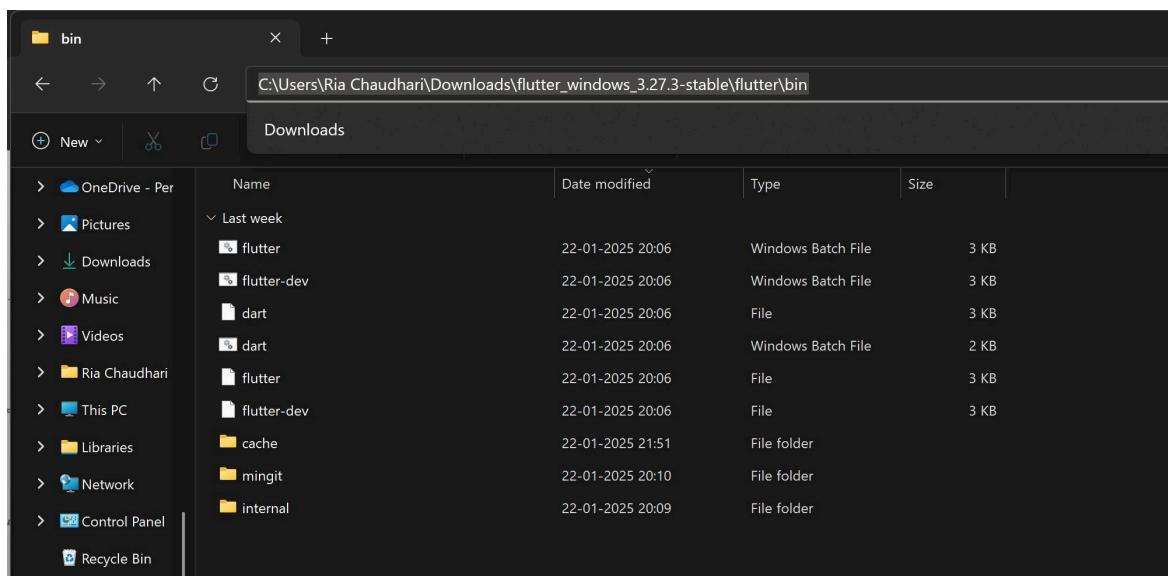
[Browse...](#)

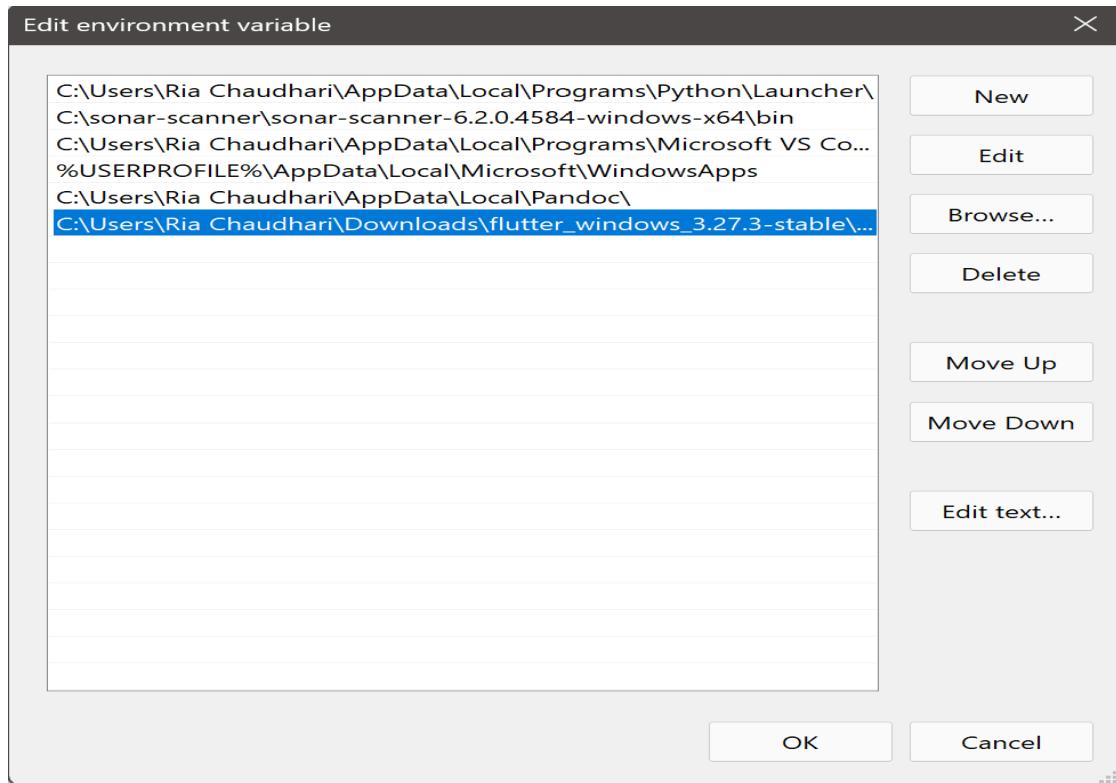
Show extracted files when complete

[Extract](#)

[Cancel](#)

Step 5 :- Add Flutter to System PATH Right-click on the Start Menu > System > Advanced system settings > Environment Variables. Under System Variables, find Path and click Edit. Add the full path to the flutter/bin directory (e.g., C:\flutter\bin).





Step 6 : - Now, run the \$ flutter command in command prompt

```
C:\Users\Ria Chaudhari>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", shows hidden options. If used with "flutter doctor", shows additional
                                diagnostic information. (Use "-vv" to force verbose logging in those cases.)
  -d, --device-id              Target device id or name (prefixes allowed).
  --version                   Reports the version of this tool.
  --enable-analytics          Enable telemetry reporting each time a flutter or dart command runs.
  --disable-analytics         Disable telemetry reporting each time a flutter or dart command runs, until it is
                                re-enabled.
  --suppress-analytics        Suppress analytics reporting for the current CLI invocation.

Available commands:
```

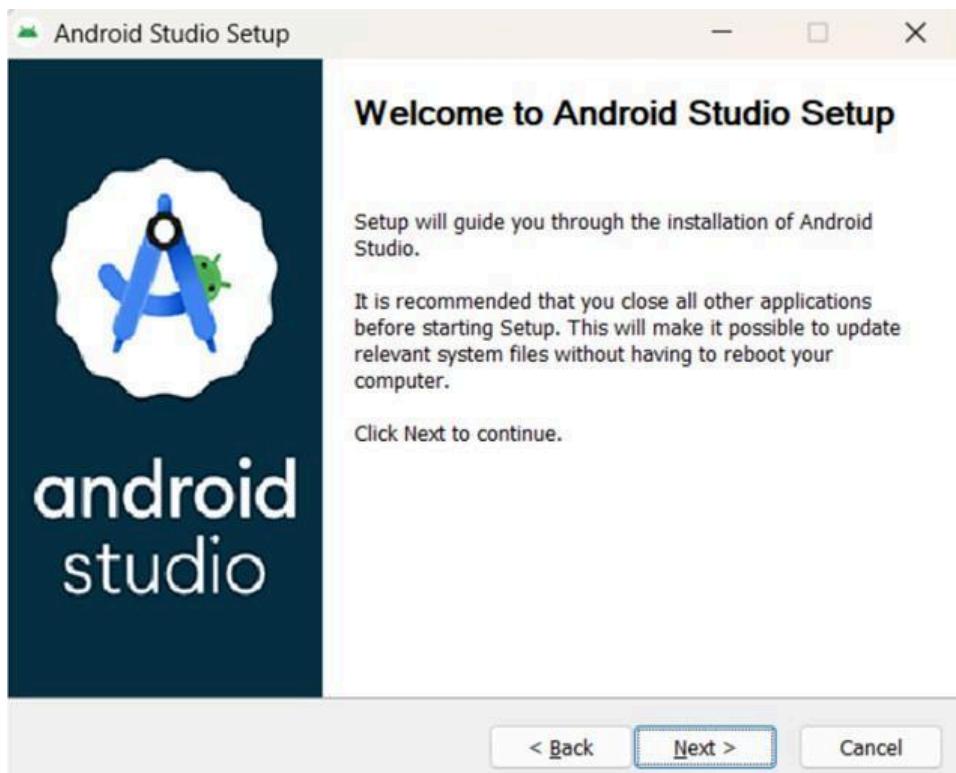
Step 7:- Run the \$ flutter doctor command. This command checks for all the requirements of Flutter app development and displays a report of the status of your Flutter installation

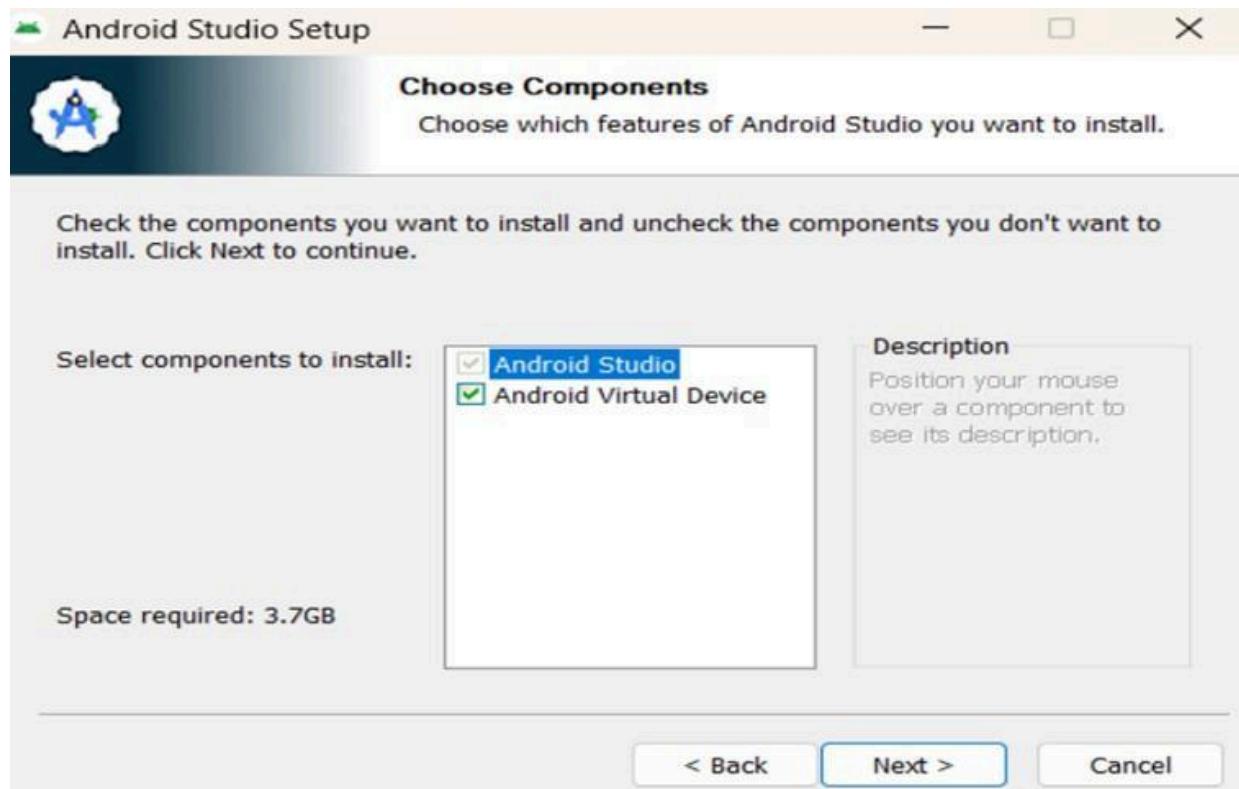
```
C:\Users\Ria Chaudhari>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.27.3, on Microsoft Windows [Version 10.0.26100.2894], locale en-IN)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 35.0.1)
[✓] Chrome - develop for the web
[✗] Visual Studio - develop Windows apps
  ✗ Visual Studio not installed; this is necessary to develop Windows apps.
    Download at https://visualstudio.microsoft.com/downloads/.
    Please install the "Desktop development with C++" workload, including all of its default components
[✓] Android Studio (version 2024.2)
[✓] IntelliJ IDEA Ultimate Edition (version 2024.2)
[✓] VS Code (version unknown)
  ✗ Unable to determine VS Code version.
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 1 category.
```

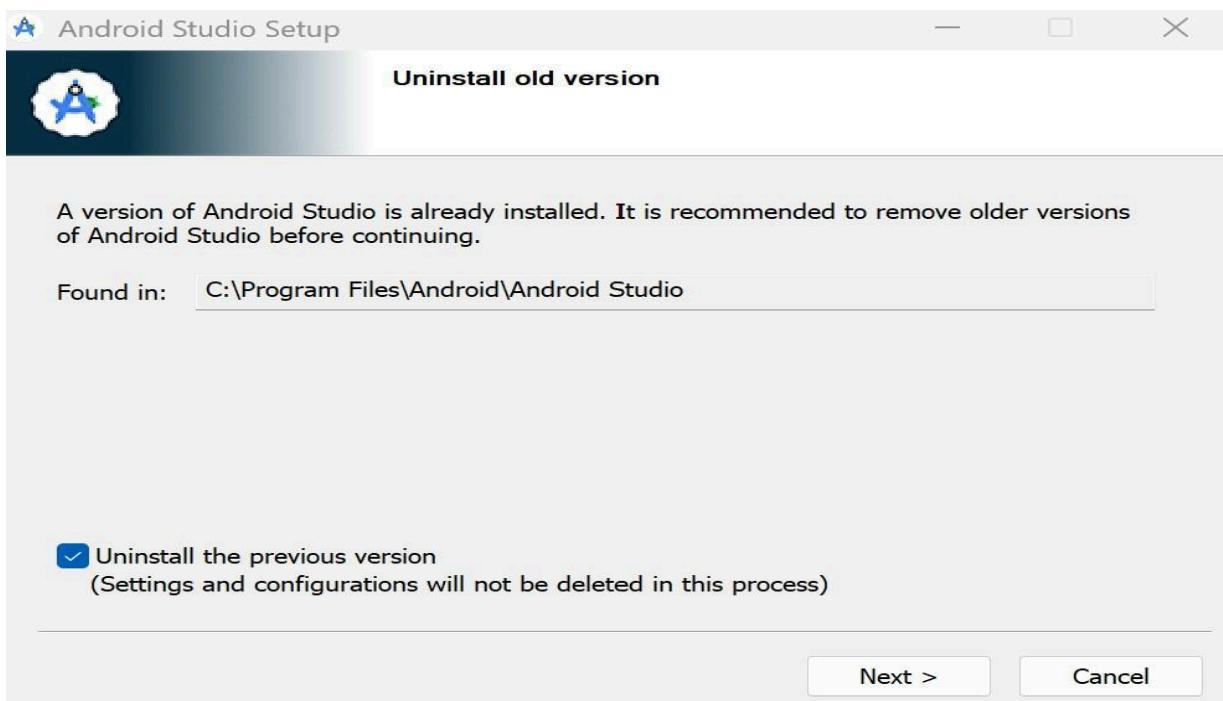
Step 8 : - Go to Android Studio and download the installer

Step 8.1: - When the download is complete, open the .exe file and run it. You will get the following dialog box

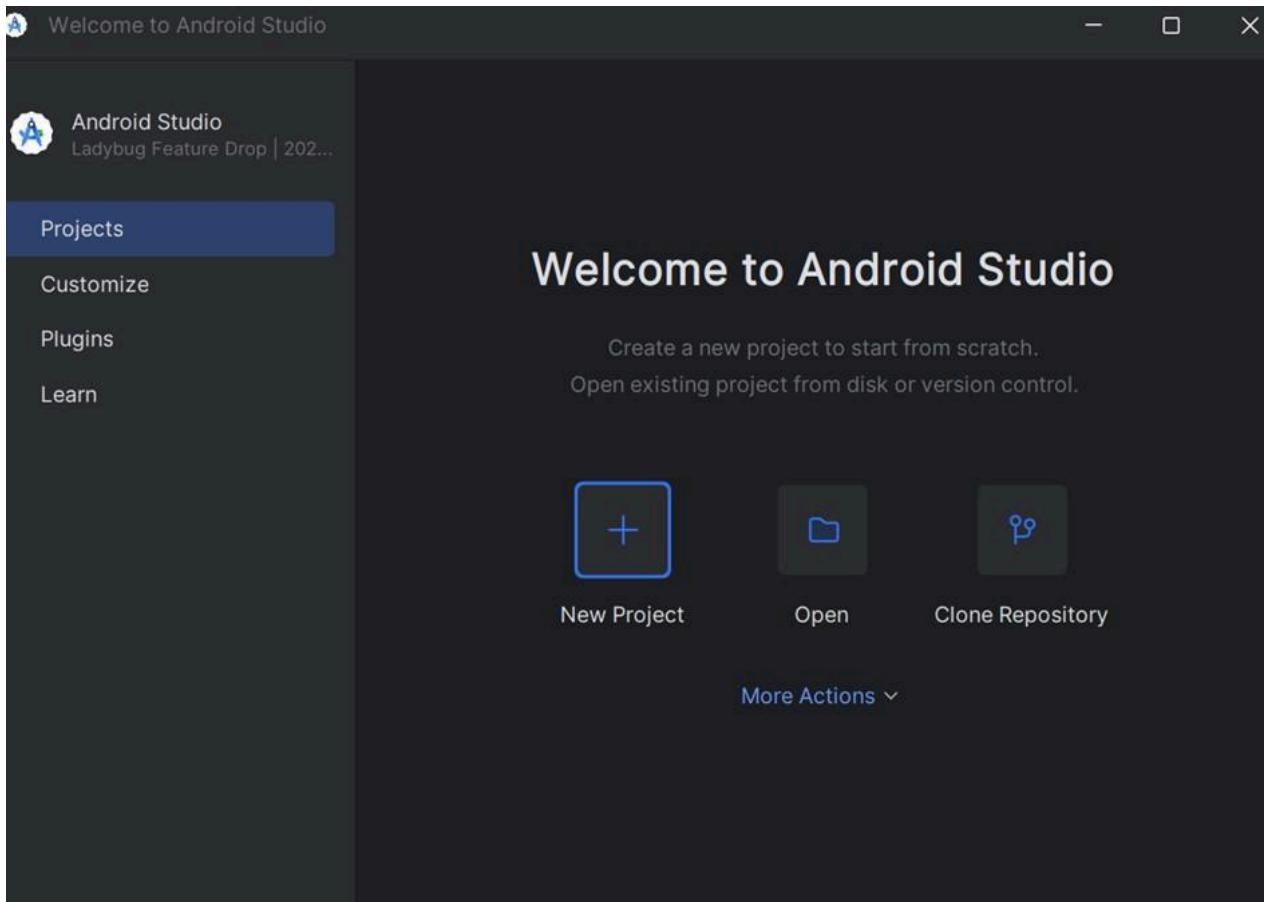




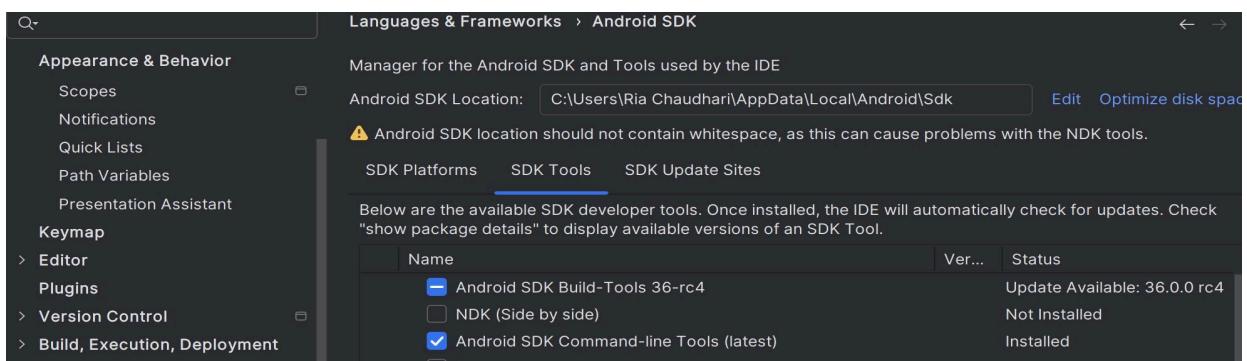
Step 8.3: - Change the destination as per your convenience and click on ‘Next’ Button.



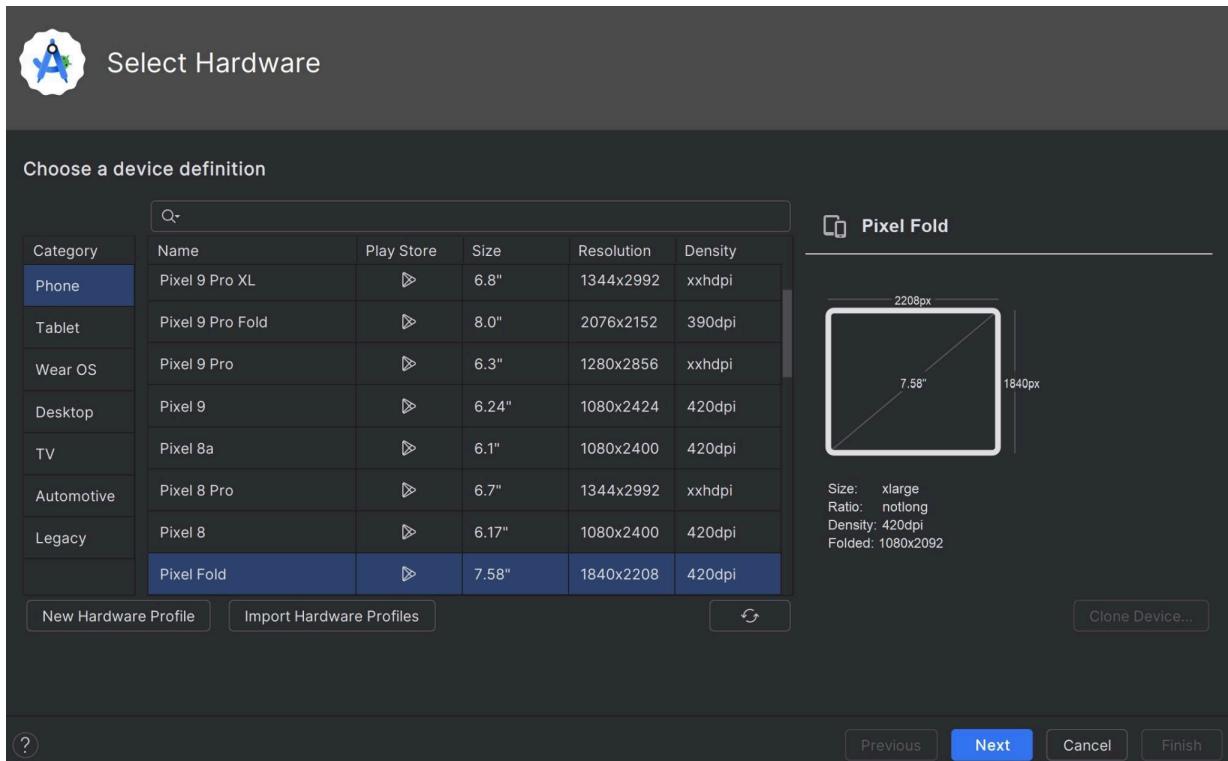
Step 8.4: - Follow the steps of the installation wizard. Once the installation wizard completes, you will get the following screen.



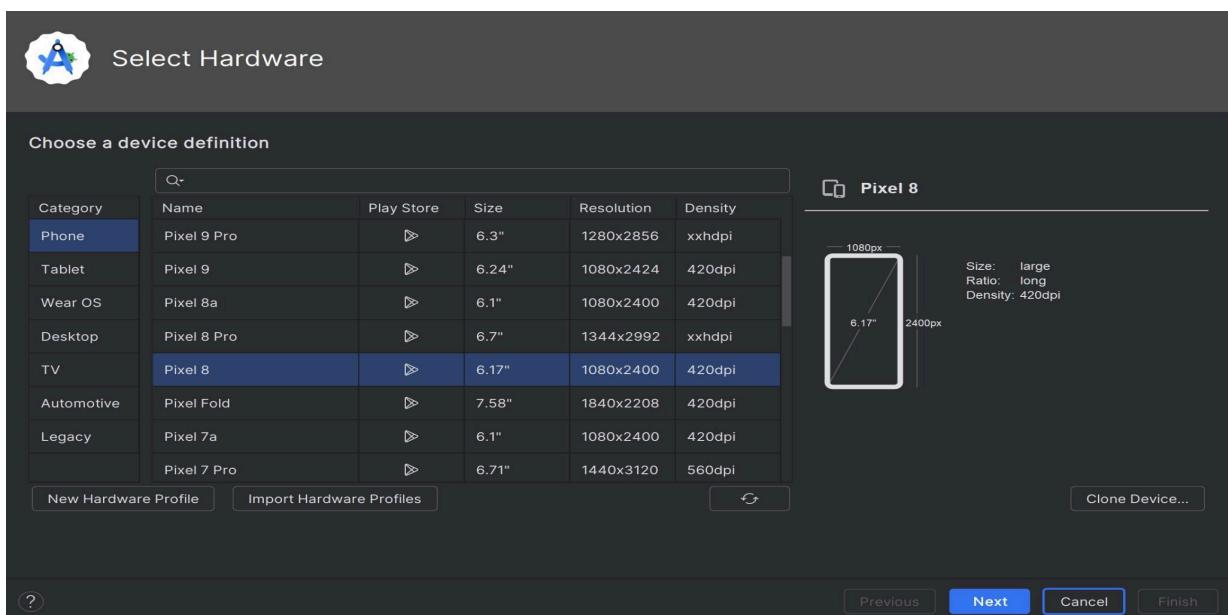
Step 8.5: - Go to Preferences > Appearance & Behavior > System Settings > Android SDK. Select the SDK Tools tab and check Android SDK Command-line Tools and Install it.

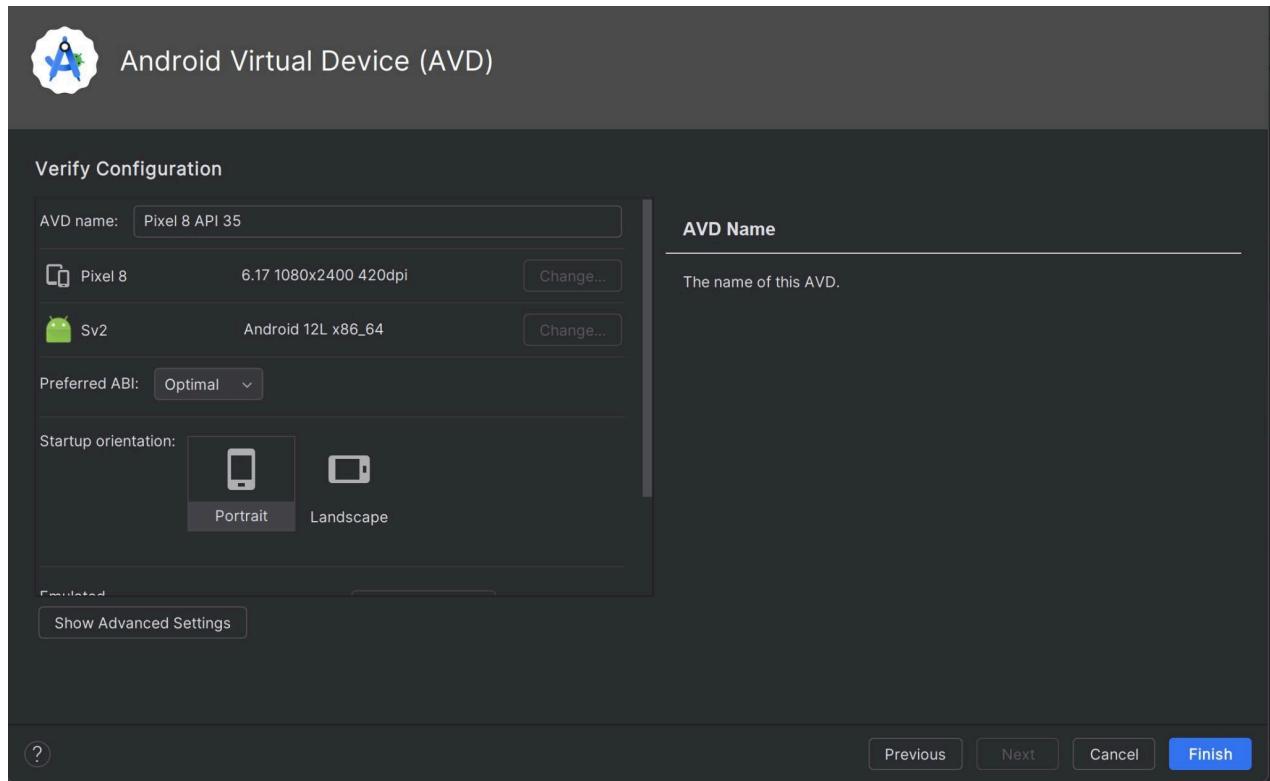


Step 10: - Next, you need to set up an Android emulator. It is responsible for running and testing the Flutter application

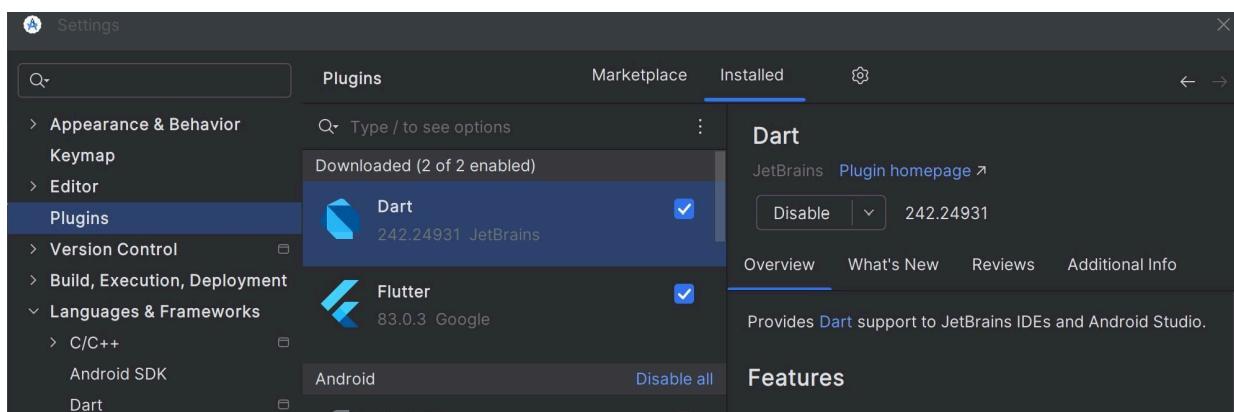


Step 10.1: - Open Android Studio and go to Tools > AVD Manager. Create a new virtual device.





Step 11: - Now, install Flutter and Dart plugin for building Flutter application in Android Studio. These plugins provide a template to create a Flutter application, give an option to run and debug Flutter application in the Android Studio itself



Step 12: - Go to File > New Project > Create Flutter Project, then select the project name and location, and click Next to proceed



```
main.dart × M+ README.md
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10   @override
11   Widget build(BuildContext context) {
12     return MaterialApp(
13       home: Scaffold(
14         appBar: AppBar(
15           title: Text('My App'),
16
17         ), // AppBar
18         body: const Center(
19           child: Text(
20             'Hello, Ria!',
21             style: TextStyle(fontSize: 24),
22           ), // Text
23           ), // Center
24         ), // Scaffold
25       ); // MaterialApp
26   }
27 }
28 }
```

MAD & PWA Lab
Journal

Experiment No.	02
Experiment Title.	To design Flutter UI by including common widgets.
Roll No.	7
Name	Ria Chaudhari
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

MPL EXP 2

Aim: To design Flutter UI by including common widgets

Theory:

Flutter UI is built using widgets, which are the basic building blocks of the application. Widgets in Flutter are classified into two types:

- Stateless Widgets – Do not change state once built. Example: Text, Icon, Container.
- Stateful Widgets – Can change state dynamically. Example: TextField, Checkbox, Switch

Common Widgets in Flutter

1. Container

Used to hold other widgets and apply styling like padding, margin, and decoration.

2. Text

Displays text in the application with different styles.

3. Image

Displays images from assets, network, or memory.

4. Button Widgets

Includes ElevatedButton, TextButton, and OutlinedButton.

5. ListView

Used for displaying a scrollable list of widgets.

6. GridView

Used for displaying a collection of items in a grid format.

Syntax for Common Widgets

1. Container

```
Widget Container(  
    width: 200,  
    height: 100,  
    color: Colors.blue,  
    child: Center(  
        child: Text('Hello Flutter!', style: TextStyle(color: Colors.white)),  
    ),  
)
```

2. Text

```
Widget Text(  
    'Welcome to Flutter',  
    style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),  
)
```

3. Image Widget
Image.asset('assets/flutter_logo.png', width: 100, height: 100)

4. ElevatedButton
Widget ElevatedButton(
onPressed: () {
 print('Button Pressed');
},
child: Text('Click Me'),
)

5. ListView
Widget ListView(
children: [
 ListTile(title: Text('Item 1')),
 ListTile(title: Text('Item 2')),
 ListTile(title: Text('Item 3')),
],
)

6. GridView
Widget
GridView.count(
crossAxisCount: 2,
children: [
 Container(color: Colors.red, height: 100), Container(color:
 Colors.green, height: 100), Container(color: Colors.blue,
 height: 100), Container(color: Colors.yellow, height: 100),
],
)

Code:

main.dart

```
import 'package:flutter/material.dart';
import 'package:blinkit/screens/signin_screen.dart';
import 'package:blinkit/screens/homepage.dart';
import 'package:blinkit/screens/categories.dart' show CategoriesScreen;
import 'package:blinkit/screens/bakery_biscuits.dart';
import 'package:blinkit/screens/otpscreen.dart';

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

class MyApp extends StatelessWidget {
```

```

@Override
Widget build(BuildContext context) {
    return MaterialApp(
        debugShowCheckedModeBanner: false,
        title: 'Grocery App',
        theme: ThemeData(
            primarySwatch: Colors.green,
        ),
        initialRoute: '/',
        routes: {
            '/': (context) => SignInScreen(),
            '/otp': (context) => OTPScreen(),
            '/home': (context) => HomePage(),
            '/categories': (context) => CategoriesScreen(),
            '/bakery_biscuits': (context) => BakeryBiscuitsPage(),
        },
    );
}
}
}

```

signin_screen.dart

```

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

class SignInScreen extends StatelessWidget { const
    SignInScreen({Key? key}) : super(key: key);

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            backgroundColor: Colors.white,
            body: Stack(
                children: [
                    // 💕 Background Image at the Top Positioned(
                    top: 0,
                    left: 0,
                    right: 0,
                    child: Container( width:
                        double.infinity,
                        height: MediaQuery.of(context).size.height *
                            0.45, // Takes 45% of screen
                        decoration: BoxDecoration(
                            image: DecorationImage(
                                image: AssetImage("assets/blinkit_onboarding_screen.png"), fit:
                                BoxFit.fitHeight,
                        ),
                    ),
                ],
            ),
        );
    }
}

```

```
// 🔒 Bottom White Login Section
Positioned(
    bottom: 0,
    left: 0,
    right: 0,
    child: Container( width:
        double.infinity,
        height: MediaQuery.of(context).size.height * 0.55,
        decoration: BoxDecoration(
            color: Colors.white,
            borderRadius: BorderRadius.vertical(top: Radius.circular(20)),
        ),
        child: Column(
            mainAxisAlignment: MainAxisAlignment.min,
            children: [
                // Blinkit Logo Image.asset("assets/blinkit_logo.png",
                height: 50),
                SizedBox(height: 16),
                // Title Text(
                "India's last minute app",
                style: GoogleFonts.lato(
                    fontSize: 20, fontWeight: FontWeight.bold),
            ),
            SizedBox(height: 8),
            Text(
                "Log in or sign up",
                style: TextStyle(fontSize: 16, color: Colors.grey[600]),
            ),
            SizedBox(height: 20),
            // Mobile Number Input
            TextField(
                keyboardType: TextInputType.phone,
                decoration: InputDecoration( prefixText:
                "+91 ",
                hintText: "Enter mobile number", border:
                OutlineInputBorder(
                    borderRadius: BorderRadius.circular(10)),
                filled: true,
                fillColor: Colors.grey[200],
            ),
        ),
    ),
),
```

```

        SizedBox(height: 20),

        // Continue Button
        SizedBox(
            width: double.infinity, height:
            50,
            child: ElevatedButton(
                onPressed: () {
                    // Navigate to Home Screen
                    Navigator.pushNamed(context, '/otp');
                },
                style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.grey[700],
                    foregroundColor: Colors.white, shape:
                    RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(8)),
                ),
                child: Text("Continue", style: TextStyle(fontSize: 18)),
            ),
        ),
    ),

    SizedBox(height: 16),

    // Terms & Conditions
    Text(
        "By continuing, you agree to our Terms of service & Privacy policy", textAlign:
        TextAlign.center,
        style: TextStyle(fontSize: 12, color: Colors.grey[600]),
    ),
),
],
),
),
),
],
),
);
},
}
}

```

otpscreen.dart

```

import 'package:flutter/material.dart';

class OTPScreen extends StatefulWidget { @override
    _OTPScreenState createState() => _OTPScreenState();
}

class _OTPScreenState extends State<OTPScreen> {

```

```
final TextEditingController otpController = TextEditingController();

@Override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: Colors.white,
        body: Padding(
            padding: const EdgeInsets.all(24.0),
            child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                crossAxisAlignment: CrossAxisAlignment.center,
                children: [
                    // Title
                    Text(
                        "Enter OTP",
                        style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold),
                    ),
                    SizedBox(height: 10),

                    // Instructions
                    Text(
                        "We have sent a 6-digit code to your phone number.",
                        textAlign: TextAlign.center,
                        style: TextStyle(fontSize: 16, color: Colors.grey[600]),
                    ),
                    SizedBox(height: 30),

                    // OTP Input Field
                    TextField(
                        controller: otpController,
                        keyboardType: TextInputType.number,
                        textAlign: TextAlign.center,
                        maxLength: 6,
                        style: TextStyle(fontSize: 24, letterSpacing: 8),
                        decoration: InputDecoration(
                            counterText: "",
                            hintText: "       ",
                            hintStyle: TextStyle(fontSize: 24, letterSpacing: 8),
                            border: OutlineInputBorder(
                                borderRadius: BorderRadius.circular(12),
                            ),
                        ),
                    ),
                    SizedBox(height: 20),

                    // Verify Button
                    SizedBox(
                        width: double.infinity,
```

```
height: 50,
child: ElevatedButton(
 onPressed: () {
 // Add OTP verification logic here
 Navigator.pushNamed(context, '/home');
},
style: ElevatedButton.styleFrom(
 backgroundColor: Colors.green[700], shape:
 RoundedRectangleBorder(borderRadius:
 BorderRadius.circular(10),
),
),
),
child: Text("Verify
OTP",
style: TextStyle(fontSize: 18, color: Colors.white),
),
),
),
),
SizedBox(height: 15),

// Resend OTP
TextButton( onPressed:
() {
 // Add Resend OTP logic here
},
),
child: Text(
"Resend OTP",
style: TextStyle(fontSize: 16, color: Colors.green[700]),
),
),
],
),
),
);
}
}
```

homepage.dart

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:blinkit/screens/categories.dart';
```

```
void main() {  
    runApp(Blinkit());  
}
```

```
class BlinkitClone extends StatelessWidget {  
    @override
```

```
Widget build(BuildContext context) {
  return MaterialApp(
    debugShowCheckedModeBanner: false,
    initialRoute: '/home',
    routes: {
      '/home': (context) => HomePage(),
      '/categories': (context) => CategoriesScreen(),
    },
  );
}

class HomePage extends StatefulWidget { @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> { int
  _selectedIndex = 0;

  void _onItemTapped(int index) { if
    (index == 2) {
      Navigator.pushNamed(context, '/categories');
    } else {
      setState(() {
        _selectedIndex = index;
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      bottomNavigationBar: BottomNavigationBar(
        currentIndex: _selectedIndex,
        selectedItemColor: Colors.green,
        unselectedItemColor: Colors.black,
        showUnselectedLabels: true,
        onTap: _onItemTapped,
        items: [
          BottomNavigationBarItem(icon: Icon(Icons.home), label: "Home"),
          BottomNavigationBarItem(
            icon: Icon(Icons.shopping_bag), label: "Order Again"),
          BottomNavigationBarItem(
            icon: Icon(Icons.category), label: "Categories"),
          BottomNavigationBarItem(icon:
            Icon(Icons.print), label: "Print"),
        ],
    ),
  }
}
```

```
),
body: SingleChildScrollView(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.start,
    children: [
      Container(
        color: Colors.yellow[700], padding:
        const EdgeInsets.only(
          top: 50, left: 16, right: 16, bottom: 10),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: [
          Row( children:
          [
            Icon(Icons.timer, color: Colors.black),
            const SizedBox(width: 5),
            Text("Blinkit
in",
              style: GoogleFonts.poppins(
                fontSize: 14, fontWeight: FontWeight.w500),
            ),
            const SizedBox(width: 5),
            Text(
              "10 minutes",
              style: GoogleFonts.poppins(
                fontSize: 18, fontWeight: FontWeight.bold),
            ),
            Spacer(),
            Icon(Icons.shopping_cart, color: Colors.black),
            const SizedBox(width: 10),
            CircleAvatar(
              backgroundColor: Colors.white,
              child: Icon(Icons.person, color: Colors.black),
            ),
          ],
        ),
        const SizedBox(height: 5),
        Text(
          "HOME - Ria Chaudhari, 202, Coral Nirmal ▼", style:
          GoogleFonts.poppins(
            fontSize: 14, fontWeight: FontWeight.w500),
        ),
        const SizedBox(height: 10),
        Container(
          padding: EdgeInsets.symmetric(horizontal: 10),
          decoration: BoxDecoration(
            color: Colors.white,
            borderRadius: BorderRadius.circular(10),

```

```
        ),
      child: Row(
        children: [
          Icon(Icons.search, color: Colors.black54),
          Expanded(
            child: TextField(
              decoration: InputDecoration(
                hintText: "Search 'cookies'", border:
                  InputBorder.none, contentPadding:
                    EdgeInsets.symmetric(vertical: 10),
              ),
            ),
          ),
          ],
        ),
        Icon(Icons.mic, color: Colors.black54),
      ],
    ),
  ),
  Container(
    padding: EdgeInsets.symmetric(vertical: 10),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: [
        _CategoryItem(icon: Icons.shopping_bag, label: "All"),
        _CategoryItem(icon: Icons.favorite, label: "Weddings"),
        _CategoryItem(icon: Icons.tv, label: "Electronics"),
        _CategoryItem(icon: Icons.brush, label: "Beauty"),
        _CategoryItem(icon: Icons.child_care, label: "Kids"),
        _CategoryItem(icon: Icons.card_giftcard, label: "Gifting"),
      ],
    ),
  ),
  Container(
    padding: EdgeInsets.all(16),
    decoration: BoxDecoration( color:
      Colors.yellow[100],
      borderRadius: BorderRadius.circular(10),
    ),
    child: Column( children:
      [ Text("Welcome",
        style: GoogleFonts.poppins(
          fontSize: 22, fontWeight: FontWeight.bold)),
        Text("Order now to avail exciting offers!",
          style: GoogleFonts.poppins(fontSize: 14)),
    ],
  ),
);
```

```

Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly, children:
  [
    Text("Flat ₹50 OFF",
      style: GoogleFonts.poppins(
        fontSize: 16,
        fontWeight: FontWeight.bold, color:
        Colors.green)),
    Text("+", style: GoogleFonts.poppins(fontSize: 20)),
    Text("Free Delivery",
      style: GoogleFonts.poppins(
        fontSize: 16,
        fontWeight: FontWeight.bold, color:
        Colors.green)),
  ],
),
],
),
),
),
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16),
  child: Text(
    "Bestsellers",
    style: GoogleFonts.poppins(
      fontSize: 18, fontWeight: FontWeight.bold),
  ),
),
),
GridView.count(
  padding: const EdgeInsets.all(16),
  crossAxisCount: 3,
  shrinkWrap: true,
  physics: NeverScrollableScrollPhysics(),
  children: [
    _BestsellerItem(image:
    "https://via.placeholder.com/400x300.png?text=Placeholder+Image", count: "+135 more"),
    _BestsellerItem(
      image: "https://via.placeholder.com/400x300.png?text=Placeholder+Image", count:
      "+320 more"),
    _BestsellerItem(image:
    "https://via.placeholder.com/400x300.png?text=Placeholder+Image", count: "+23 more"),
    _BestsellerItem(
      image: "https://via.placeholder.com/400x300.png?text=Placeholder+Image", count:
      "+137 more"),
    _BestsellerItem(image:
    "https://via.placeholder.com/400x300.png?text=Placeholder+Image", count: "+114 more"),
    _BestsellerItem(
      image: "https://via.placeholder.com/400x300.png?text=Placeholder+Image", count:
      "+135 more"),
  ],
)

```

```

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

class _CategoryItem extends StatelessWidget { final
IconData icon;
final String label;
_CategoryItem({required this.icon, required this.label});

@Override
Widget build(BuildContext context) {
return Column(
children: [
Icon(icon, size: 28, color: Colors.black),
Text(label, style: GoogleFonts.poppins(fontSize: 12)),
],
);
}
}

class _BestsellerItem extends StatelessWidget { final
String image;
final String count;
_BestsellerItem({required this.image, required this.count});

@Override
Widget build(BuildContext context) {
return Column(
children: [
Image.asset(image, height: 50, fit: BoxFit.cover), Text(count,
style: GoogleFonts.poppins(fontSize: 12)),
],
);
}
}

```

bakery_biscuit.dart

```

import 'package:flutter/material.dart';

class BakeryBiscuitsPage extends StatelessWidget {
@Override
Widget build(BuildContext context) {
return Scaffold(

```

```
appBar: AppBar(  
    backgroundColor: Colors.amber.shade100,  
    elevation: 0,  
    title: Text(  
        "Bakery & Biscuits", style:  
        TextStyle(  
            fontSize: 20, fontWeight: FontWeight.bold, color: Colors.black),  
    ),  
    leading: IconButton(  
        icon: Icon(Icons.arrow_back, color: Colors.black),  
        onPressed: () {  
            Navigator.pop(context);  
        },  
    ),  
,  
),  
body: Row(  
    children: [  
        // Sidebar Categories with Images  
        Container(  
            width: 100,  
            color: Colors.white,  
            child: ListView(  
                children: [  
                    CategoryTile( label:  
                        'Cookies',  
                        image: 'https://via.placeholder.com/400x300.png?text=Placeholder+Image',  
                        isSelected: true),  
                    CategoryTile(  
                        label: 'Cream Biscuits',  
                        image: 'https://via.placeholder.com/400x300.png?text=Placeholder+Image'),  
                    CategoryTile(  
                        label: 'Healthy & Digestive',  
                        image: 'https://via.placeholder.com/400x300.png?text=Placeholder+Image'),  
                    CategoryTile(  
                        label: 'Sweet & Salty', image:  
                            'https://via.placeholder.com/400x300.png?text=Placeholder+Image'),  
                    CategoryTile(  
                        label: 'Glucose & Marie',  
                        image: 'https://via.placeholder.com/400x300.png?text=Placeholder+Image'),  
                    CategoryTile(  
                        label: 'Rusks & Wafers', image:  
                            'https://via.placeholder.com/400x300.png?text=Placeholder+Image'),  
                    CategoryTile(  
                        label: 'Cakes & Rolls', image:  
                            'https://via.placeholder.com/400x300.png?text=Placeholder+Image'),  
                    CategoryTile(  
                        label: 'Baking Ingredients',  
                        image: 'https://via.placeholder.com/400x300.png?text=Placeholder+Image'),  
                ]  
            )  
        )  
    ]  
)
```

```
        ],
        ),
        ),
        Expanded( child:
            Column( children:
                [
                    Expanded(
                        child: GridView.builder(
                            padding: EdgeInsets.all(8),
                            gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
                                crossAxisCount: 2,
                                crossAxisSpacing: 10,
                                mainAxisSpacing: 10,
                                childAspectRatio: 0.7,
                            ),
                            itemCount: products.length, itemBuilder:
                            (context, index) {
                                return ProductTile(product: products[index]);
                            },
                        ),
                    ),
                ],
            ),
        ),
    ],
),
);
}
}
```

```
class CategoryTile extends StatelessWidget { final
String label;
final String image; final
bool isSelected;

CategoryTile(
    {required this.label, required this.image, this.isSelected = false});

@Override
Widget build(BuildContext context) {
    return Padding(
        padding: const EdgeInsets.symmetric(vertical: 8.0), child:
        Column(
            children: [
                Container( height:
50,
                    width: 50,
                    decoration: BoxDecoration(
```

```
borderRadius: BorderRadius.circular(10),
image: DecorationImage(
    image: AssetImage(image), fit:
    BoxFit.cover,
),
),
),
),
SizedBox(height: 5),
Text(
    label,
    textAlign: TextAlign.center,
    style: TextStyle(
        color: isSelected ? Colors.green : Colors.black,
        fontWeight: isSelected ? FontWeight.bold : FontWeight.normal,
    ),
),
],
),
);
}
}
```

```
class ProductTile extends StatelessWidget {
final Product product;
ProductTile({required this.product});
```

```
@override
```

```
Widget build(BuildContext context) {
return Card(
shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(10)), child:
Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Expanded(
child: Image.network(
product.imageUrl, fit:
BoxFit.cover, width:
double.infinity,
),
),
Padding(
padding: const EdgeInsets.all(8.0),
child: Text(
product.name,
style: TextStyle(fontWeight: FontWeight.bold),
),
),
)
```

```

Padding(
padding: const EdgeInsets.symmetric(horizontal: 8.0),
child: Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    Text( '₹${product.price}', style: TextStyle(
      color: Colors.green, fontWeight: FontWeight.bold),
    ),
    ElevatedButton(
      onPressed: () {},
      child: Text('ADD'),
    ),
  ],
),
),
],
),
),
),
);
}
}

class Product { final
String name;
final String imageUrl; final
double price;

Product({required this.name, required this.imageUrl, required this.price});
}

List<Product> products = [
Product(
  name: 'Sunfeast Baked Creations Walnut & Choco Chip Cookies',
  imageUrl: 'https://via.placeholder.com/400x300.png?text=Placeholder+Image',
  price: 134.0),
Product(
  name: 'Hide & Seek Chocolate Chip Cookies (100 g)',
  imageUrl: 'https://via.placeholder.com/400x300.png?text=Placeholder+Image',
  price: 29.0),
Product(
  name: 'Hide & Seek Chocolate Chip Cookies (200 g)',
  imageUrl: 'https://via.placeholder.com/400x300.png?text=Placeholder+Image',
  price: 45.0),
Product(
  name: 'Britannia Good Day Cashew Biscuit',
  imageUrl: 'https://via.placeholder.com/400x300.png?text=Placeholder+Image',
  price: 49.0),
];

```

];

Output:

The image displays two side-by-side screenshots of a web browser window titled "Grocery App".

Left Screenshot (localhost:57084): This is the initial sign-up screen. It features a logo consisting of three yellow squares arranged in a triangle. Below the logo, the text "India's last minute app" is displayed. Underneath that, there are links for "Log in or sign up". A large input field is labeled "Enter mobile number". Below this input field is a dark grey button labeled "Continue". At the bottom of the page, a small note states: "By continuing, you agree to our Terms of service & Privacy policy".

Right Screenshot (localhost:61687/#/otp): This is the OTP verification screen. The title "Enter OTP" is centered at the top. Below it, a message says "We have sent a 6-digit code to your phone number." To the right of this message is a light blue input field containing a placeholder "-----". To the right of the input field is a green button labeled "Verify OTP". Below the "Verify OTP" button is a link "Resend OTP".

Grocery App

localhost:57084/#/home

Blinkit in 10 minutes

HOME - Ria Chaudhari, 202, Coral Nirmal ▾

Search 'cookies'

All Weddings Electronics Beauty Kids Gifting

Welcome

Order now to avail exciting offers!

Flat ₹50 OFF + Free Delivery

Bestsellers

+135 more +320 more +23 more

+137 more +114 more +135 more

Home Order Again Categories Print

Grocery App

localhost:57084/#/bakery_b...

← Bakery & Biscuits

Cookies

Cream Biscuits

Healthy & Digestive

Sweet & Salty

Glucose & Marie

Rusks & Wafers

Cakes & Rolls

[object ProgressEvent]

[object ProgressEvent]

Sunfeast Baked Creations Walnut & Choco Chip Cookies ₹134 ADD

Hide & Seek Chocolate Chip Cookies (100 g) ₹29 ADD

[object ProgressEvent]

[object ProgressEvent]

Hide & Seek Chocolate Chip Cookies (200 g) ₹45 ADD

Britannia Good Day Cashew Biscuit ₹49 ADD

MAD & PWA LabJournal

Experiment No.	03
Experiment Title.	To include icons, images, fonts in Flutter app
Roll No.	7
Name	Ria Chaudhari
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

MPL EXP 3

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

Theory:

Flutter allows adding **icons, images, and custom fonts** to improve visual representation. These assets can be stored in the **assets** folder and defined in the **pubspec.yaml** file.

1. Icons in Flutter

Flutter provides built-in icons through the **Icons** class, which contains Material Icons. We can also use external icons like Font Awesome or custom SVG icons.

2. Images in Flutter

Images can be included in Flutter from:

Asset images (stored locally)

Network images (fetched from a URL)

Memory images (loaded dynamically)

3. Fonts in Flutter

By default, Flutter uses the Roboto font, but custom fonts can be added by including .ttf or .otf font files in the assets. The font must be declared in **pubspec.yaml**.

Syntax for Including Icons, Images, and Fonts

1. Using Icons in Flutter

```
Icon( Icons.home,  
      size: 50,  
      color: Colors.blue,  
)
```

To use external icons (e.g., **Font Awesome**):

```
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
```

```
Icon(  
      FontAwesomeIcons.twitter,  
      size: 50,  
      color: Colors.blue,  
)
```

```
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
Icon(
  FontAwesomeIcons.twitter, size:
  50,
  color: Colors.blue,
)
```

Ensure you add the dependency in pubspec.yaml:

```
dependencies:
  font_awesome_flutter: ^10.7.0
```

2. Using Images in Flutter

A. Asset Image (Local)

Add the image in the assets/images/ folder.

Declare it in pubspec.yaml:

```
flutter:
  assets:
    - assets/images/sample.png
```

Use it in the code:

```
Image.asset('assets/images/sample.png', width: 100, height: 100)
```

B. Network

```
Image
Image.network(
  'https://example.com/sample.jpg',
  width: 100,
  height: 100,
)
```

3. Using Custom Fonts in

Flutter Place the font file in

assets/fonts/. Declare it in

pubspec.yaml:

```
flutter:
  fonts:
    - family: CustomFont
      fonts:
        - asset: assets/fonts/CustomFont-Regular.ttf
        - asset: assets/fonts/CustomFont-Bold.ttf
          weight: 700
```

Use it in Text widget:

```
Text(
  'Hello Flutter!', style:
  TextStyle(
```

```
fontFamily: 'CustomFont', fontSize:  
24,
```

```

        fontWeight: FontWeight.bold,
),
)

```

Widgets Used

Widgets	Description
Icon	Displays built-in Material icons.
Image.asset	Loads an image from local assets.
Image.network	Loads an image from a URL.
ImageIcon	Displays an image as an icon.
Text	Displays text with custom fonts.

Important Properties

Property	Description
size	Defines the size of an icon or image.
color	Changes the color of an icon.
fit	Controls how an image fits inside a widget (BoxFit).
fontFamily	Specifies the custom font for text.
fontSize	Adjusts the size of the text.

Code:

homepage.dart

```

import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:blinkit/screens/categories.dart';

```

```

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

```

```

class BlinkitClone extends StatelessWidget {

```

```

  @override

```

```

  Widget build(BuildContext context) {

```

```

    return MaterialApp(

```

```

      debugShowCheckedModeBanner: false,

```

```

      initialRoute: '/home',

```

```
routes: {
  '/home': (context) => HomePage(),
  '/categories': (context) => CategoriesScreen(),
},
);
}
}

class HomePage extends StatefulWidget {
@override
_HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
int _selectedIndex = 0;

void _onItemTapped(int index) {
  if (index == 2) {
    Navigator.pushNamed(context, '/categories');
  } else {
    setState(() {
      _selectedIndex = index;
    });
  }
}

@Override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.white,
    bottomNavigationBar: BottomNavigationBar(
      currentIndex: _selectedIndex,
      selectedItemColor: Colors.green,
      unselectedItemColor: Colors.black,
      showUnselectedLabels: true,
      onTap: _onItemTapped,
      items: [
        BottomNavigationBarItem(icon: Icon(Icons.home), label: "Home"),
        BottomNavigationBarItem(
          icon: Icon(Icons.shopping_bag), label: "Order Again"),
        BottomNavigationBarItem(
          icon: Icon(Icons.category), label: "Categories"),
        BottomNavigationBarItem(icon: Icon(Icons.print), label: "Print"),
      ],
    ),
  );
}
```

```
),
body: SingleChildScrollView(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.start,
    children: [
      Container(
        color: Colors.yellow[700], padding:
        const EdgeInsets.only(
          top: 50, left: 16, right: 16, bottom: 10), child:
      Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: [
          Row( children:
            [
              Icon(Icons.timer, color: Colors.black),
              const SizedBox(width: 5),
              Text( "Blinkit
in",
                style: GoogleFonts.poppins(
                  fontSize: 14, fontWeight: FontWeight.w500),
              ),
              const SizedBox(width: 5),
              Text(
                "10 minutes",
                style: GoogleFonts.poppins(
                  fontSize: 18, fontWeight: FontWeight.bold),
              ),
              Spacer(),
              Icon(Icons.shopping_cart, color: Colors.black),
              const SizedBox(width: 10),
              CircleAvatar(
                backgroundColor: Colors.white,
                child: Icon(Icons.person, color: Colors.black),
              ),
            ],
          ),
          const SizedBox(height: 5), Text(
            "HOME - Ria Chaudhari, 202, Coral Nirmal ▼",
            style: GoogleFonts.poppins(
              fontSize: 14, fontWeight: FontWeight.w500),
          ),
          const SizedBox(height: 10),
          Container(
```



```
),
child: Column(
  children: [
    Text("Welcome",
      style: GoogleFonts.poppins(
        fontSize: 22, fontWeight: FontWeight.bold)),
    Text("Order now to avail exciting offers!",
      style: GoogleFonts.poppins(fontSize: 14)),
    Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: [
        Text("Flat ₹50 OFF",
          style: GoogleFonts.poppins(
            fontSize: 16,
            fontWeight: FontWeight.bold,
            color: Colors.green)),
        Text("+", style: GoogleFonts.poppins(fontSize: 20)),
        Text("Free Delivery",
          style: GoogleFonts.poppins(
            fontSize: 16,
            fontWeight: FontWeight.bold,
            color: Colors.green)),
      ],
    ),
  ],
),
),
),
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16), child:
Text(
  "Bestsellers",
  style: GoogleFonts.poppins(
    fontSize: 18, fontWeight: FontWeight.bold),
),
),
),
GridView.count(
  padding: const EdgeInsets.all(16), crossAxisCount:
  3,
  shrinkWrap: true,
  physics: NeverScrollableScrollPhysics(),
  children: [
    _BestsellerItem(image: "assets/onion.jpg", count: "+135 more"),
    _BestsellerItem(
      image: "assets/oranges.jpg", count: "+320 more"),
  ],
)
```

```
        _BestsellerItem(image: "assets/lays.jpg", count: "+23 more"),
        _BestsellerItem(
            image: "assets/fortune.jpg", count: "+137 more"),
        _BestsellerItem(image: "assets/sprite.jpg", count: "+114 more"),
        _BestsellerItem(
            image: "assets/parle_g.jpg", count: "+135 more"),
    ],
),
],
),
),
),
);
}
}
```

```
class _CategoryItem extends StatelessWidget {
    final IconData icon;
    final String label;
    _CategoryItem({required this.icon, required this.label});

    @override
    Widget build(BuildContext context) {
        return Column(
            children: [
                Icon(icon, size: 28, color: Colors.black),
                Text(label, style: GoogleFonts.poppins(fontSize: 12)),
            ],
        );
    }
}
```

```
class _BestsellerItem extends StatelessWidget {
    final String image;
    final String count;
    _BestsellerItem({required this.image, required this.count});
```

```
    @override
    Widget build(BuildContext context) {
        return Column(
            children: [
                Image.asset(image, height: 50, fit: BoxFit.cover),
                Text(count, style: GoogleFonts.poppins(fontSize: 12)),
            ],
        );
    }
}
```

```
 }  
 }
```

categories.dart

```
import 'package:flutter/material.dart';  
  
class CategoriesScreen extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      backgroundColor: Colors.white,  
      appBar: AppBar(  
        backgroundColor: Colors.amber.shade100,  
        elevation: 0,  
        title: Column(  
          crossAxisAlignment: CrossAxisAlignment.start,  
          children: [  
            Text("Blinkit  
in",  
              style: TextStyle(fontSize: 14, color: Colors.black54),  
            ),  
            Text(  
              "11 minutes",  
              style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold, color:  
                Colors.black),  
            ),  
            Text(  
              "HOME - Ria Chaudhari, 202, Coral Nirmal ▼", style:  
              TextStyle(fontSize: 14, color: Colors.black54),  
            ),  
          ],  
        ),  
        actions: [  
          IconButton(  
            icon: Icon(Icons.card_giftcard, color: Colors.black),  
            onPressed: () {},  
          ),  
          IconButton(  
            icon: Icon(Icons.person, color: Colors.black),  
            onPressed: () {},  
          ),  
        ],  
        body: Column(  
          children: [  
            Text("Blinkit  
in",  
              style: TextStyle(fontSize: 14, color: Colors.black54),  
            ),  
            Text(  
              "11 minutes",  
              style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold, color:  
                Colors.black),  
            ),  
            Text(  
              "HOME - Ria Chaudhari, 202, Coral Nirmal ▼", style:  
              TextStyle(fontSize: 14, color: Colors.black54),  
            ),  
          ],  
        ),  
      ),  
    );  
  }  
}
```

```
children: [
  Padding(
    padding: const EdgeInsets.all(10.0),
    child: TextField(
      decoration: InputDecoration(
        hintText: 'Search "gifts"',
        prefixIcon: Icon(Icons.search, color: Colors.black54),
        suffixIcon: Icon(Icons.mic, color: Colors.black54),
        filled: true,
        fillColor: Colors.white,
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(10),
          borderSide: BorderSide.none,
        ),
      ),
    ),
  ),
  Expanded(
    child: SingleChildScrollView(
      child: Padding(
        padding: const EdgeInsets.all(10.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            buildCategorySection(context, "Grocery & Kitchen", [
              {"name": "Vegetables & Fruits", "image": "assets/vegetables_fruits.png"},
              {"name": "Atta, Rice & Dal", "image": "assets/atta_rice_dal.png"},
              {"name": "Oil, Ghee & Masala", "image": "assets/oil_ghee_masala.jpg"},
              {"name": "Dairy, Bread & Eggs", "image": "assets/dairy_bread_eggs.png"},
              {"name": "Bakery & Biscuits", "image": "assets/bakery_biscuits.png"},
              {"name": "Dry Fruits & Cereals", "image": "assets/dry_fruits_cereals.png"},
              {"name": "Chicken, Meat & Fish", "image": "assets/chicken_meat_fish.png"},
              {"name": "Kitchenware & Appliances", "image": "assets/kitchenware_appliances.png"},
            ]),
            buildCategorySection(context, "Snacks & Drinks", [
              {"name": "Chips & Namkeen", "image": "assets/chips_namkeen.png"},
              {"name": "Sweets & Chocolates", "image": "assets/sweets_chocolates.png"},
              {"name": "Drinks & Juices", "image": "assets/drinks_juices.png"},
            ]),
          ],
        ),
      ),
    ),
  ),
]
```

```

        {"name": "Tea, Coffee & Milk Drinks", "image":  

"assets/tea_coffee_milk.png"},  

        {"name": "Instant Food", "image": "assets/instant_food.png"},  

        {"name": "Sauces & Spreads", "image": "assets/sauces_spreads.png"},  

        {"name": "Paan Corner", "image": "assets/paan_corner.png"},  

        {"name": "Ice Creams & Desserts", "image":  

"assets/icrecream_desserts.png"},  

    ]),  

    ],  

    ),  

    ),  

    ),  

    ),  

    ),  

    Container(  

padding: EdgeInsets.all(10),  

color: Colors.blue.shade100,  

child: Row(  

mainAxisAlignment: MainAxisAlignment.center,  

children: [  

    Icon(Icons.delivery_dining, color: Colors.blue.shade800),  

    SizedBox(width: 5),  

    Text("Get FREE delivery", style: TextStyle(fontWeight: FontWeight.bold, color:  

Colors.blue.shade800)),  

    SizedBox(width: 5),  

    Text("on all orders above ₹149", style: TextStyle(color: Colors.black54)),  

],  

),  

),  

],  

),  

);  

}

```

```

Widget buildCategorySection(BuildContext context, String title, List<Map<String,  

String>> categories) {  

return Column(  

crossAxisAlignment: CrossAxisAlignment.start,  

children: [  

    SizedBox(height: 10),  

    Text(  

    title,  

    style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),  

),  

    GridView.builder(

```

```
        shrinkWrap: true,
        physics: NeverScrollableScrollPhysics(),
        gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 4,
            crossAxisSpacing: 10,
            mainAxisSpacing: 10,
            childAspectRatio: 0.8,
        ),
        itemCount: categories.length,
        itemBuilder: (context, index) {
            return GestureDetector(
                onTap: () {
                    if (categories[index]["name"] == "Bakery & Biscuits") {
                        Navigator.pushNamed(context, '/bakery_biscuits');
                    }
                },
                child: Column(
                    children: [
                        Container(
                            height: 60,
                            width: 60,
                            decoration: BoxDecoration( color:
                                Colors.grey.shade200,
                                borderRadius: BorderRadius.circular(10),
                                image: DecorationImage(
                                    image: AssetImage(categories[index]["image"]!), fit:
                                    BoxFit.cover,
                            ),
                            ),
                        ),
                        SizedBox(height: 5),
                        Text(
                            categories[index]["name"]!,
                            textAlign: TextAlign.center,
                            style: TextStyle(fontSize: 12),
                        ),
                    ],
                );
            },
        ],
    );
}
```

```
}

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

class BakeryBiscuitsPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.amber.shade100,
        elevation: 0,
        title: Text(
          "Bakery & Biscuits",
          style: TextStyle(
            fontSize: 20, fontWeight: FontWeight.bold, color: Colors.black),
        ),
        leading: IconButton(
          icon: Icon(Icons.arrow_back, color: Colors.black),
          onPressed: () {
            Navigator.pop(context);
          },
        ),
        ),
      body: Row(
        children: [
          // Sidebar Categories with Images
          Container(
            width: 100,
            color: Colors.white,
            child: ListView(
              children: [
                CategoryTile( label:
                  'Cookies',
                  image: 'assets/cookies.png',
                  isSelected: true),
                CategoryTile(
                  label: 'Cream Biscuits',
                  image: 'assets/cream_biscuits.png'),
                CategoryTile(
                  label: 'Healthy & Digestive',
                  image: 'assets/healthy_digestive.png'),
                CategoryTile(
                  label: 'Sweet & Salty', image: 'assets/sweet_salty.png'),
              ],
            ),
          ),
        ],
      ),
    );
  }
}
```

```

CategoryTile(
    label: 'Glucose & Marie',
    image: 'assets/glucose_marie.png'),
CategoryTile(
    label: 'Rusks & Wafers', image: 'assets/rusks_wafers.png'),
CategoryTile(
    label: 'Cakes & Rolls', image: 'assets/cakes_rolls.png'), CategoryTile(
    label: 'Baking Ingredients',
    image: 'assets/baking_ingredients.png'),
],
),
),
),
Expanded( child:
Column(
children: [
Expanded(
    child: GridView.builder(
        padding: EdgeInsets.all(8),
        gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 2,
            crossAxisSpacing: 10,
            mainAxisSpacing: 10,
            childAspectRatio: 0.7,
        ),
        itemCount: products.length,
        itemBuilder: (context, index) {
            return ProductTile(product: products[index]);
        },
    ),
),
),
),
],
),
),
),
);
}
}

```

```

class CategoryTile extends StatelessWidget {
    final String label;
    final String image; final
    bool isSelected;

```

```
CategoryTile(
    {required this.label, required this.image, this.isSelected = false});

@Override
Widget build(BuildContext context) {
    return Padding(
        padding: const EdgeInsets.symmetric(vertical: 8.0),
        child: Column(
            children: [
                Container(
                    height: 50,
                    width: 50,
                    decoration: BoxDecoration(
                        borderRadius: BorderRadius.circular(10),
                        image: DecorationImage(
                            image: AssetImage(image), fit:
                            BoxFit.cover,
                        ),
                    ),
                ),
                SizedBox(height: 5),
                Text(
                    label,
                    textAlign: TextAlign.center,
                    style: TextStyle(
                        color: isSelected ? Colors.green : Colors.black,
                        fontWeight: isSelected ? FontWeight.bold : FontWeight.normal,
                    ),
                ),
            ],
        );
    }
}

class ProductTile extends StatelessWidget { final
Product product;
ProductTile({required this.product});

@Override
Widget build(BuildContext context) {
```

return Card(

```
shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(10)), child:  
Column(  
  mainAxisAlignment: MainAxisAlignment.start,  
  children: [  
    Expanded(  
      child: Image.network(  
        product.imageUrl, fit:  
        BoxFit.cover, width:  
        double.infinity,  
      ),  
    ),  
    Padding(  
      padding: const EdgeInsets.all(8.0), child:  
      Text(  
        product.name,  
        style: TextStyle(fontWeight: FontWeight.bold),  
      ),  
    ),  
    Padding(  
      padding: const EdgeInsets.symmetric(horizontal: 8.0),  
      child: Row(  
        mainAxisAlignment: MainAxisAlignment.spaceBetween,  
        children: [  
          Text(  
            '₹${product.price}',  
            style: TextStyle(  
              color: Colors.green, fontWeight: FontWeight.bold),  
          ),  
          ElevatedButton( onPressed:  
            () {},  
            child: Text('ADD'),  
          ),  
        ],  
      ),  
    ),  
  ],  
);  
}  
  
}  
  
class Product { final  
  String name;  
  final String imageUrl;
```

```
    final double price;  
  
    Product({required this.name, required this.imageUrl, required this.price});  
}  
  
List<Product> products = [
```

```
    Product(  
        name: 'Sunfeast Baked Creations Walnut & Choco Chip Cookies',  
        imageUrl: 'assets/sunfeast_cookies.png',  
        price: 134.0),
```

```
    Product(  
        name: 'Hide & Seek Chocolate Chip Cookies (100 g)',  
        imageUrl: 'assets/hide_seek_100.png',  
        price: 29.0),
```

```
    Product(  
        name: 'Hide & Seek Chocolate Chip Cookies (200 g)',  
        imageUrl: 'assets/hide_seek_200.png',  
        price: 45.0),
```

```
    Product(  
        name: 'Britannia Good Day Cashew Biscuit',  
        imageUrl: 'assets/brittania_goodday.png', price:  
        49.0),
```

```
];
```

pubsec.yaml:

```
flutter:
```

```
  assets:  
    - assets\blinit_onboarding_screen.png  
    - assets\blinit_logo.png  
    - assets\fortune.jpg  
    - assets\lays.jpg  
    - assets\onion.jpg  
    - assets\oranges.jpg  
    - assets\parle_g.jpg  
    - assets\sprite.jpg  
    - assets\atta_rice_dal.png  
    - assets\bakery_biscuits.png  
    - assets\dairy_bread_eggs.png  
    - assets\oil_ghee_masala.jpg  
    - assets\vegetables_fruits.png  
    - assets\dry_fruits_cereals.png  
    - assets\kitchenware_appliances.png  
    - assets\chips_namkeen.png  
    - assets\chicken_meat_fish.png  
    - assets\icream_deserts.png
```

- assets\instant_food.png
- assets\paan_corner.png
- assets\sauces_spreads.png
- assets\sweets_chocolates.png
- assets\tea_coffee_milk.png
- assets\drinks_juices.png
- assets\hide_seek_100.png
- assets/sunfeast_cookies.png
- assets\hide_seek_200.png
- assets\brittania_goodday.png
- assets\lotus_biscoff.png
- assets\nankhatai.png
- assets\cookies.png
- assets\cream_biscuits.png
- assets\sweet_salty.png
- assets\glucose_marie.png
- assets\rusks_wafers.png
- assets\cakes_rolls.png
- assets\baking_ingredients.png
- assets\healthy_digestive.png

dependencies:

flutter:

sdk: flutter

The following adds the Cupertino Icons font to your application.

Use with the CupertinoIcons class for iOS style icons.

cupertino_icons: ^1.0.8

google_fonts: ^6.2.1

lucide_icons: ^0.257.0

Output:

Grocery App

localhost:55481

India's last minute app

Log in or sign up

Enter mobile number

Continue

By continuing, you agree to our Terms of service & Privacy policy

Grocery App

localhost:55481/#/home

Blinkit in 10 minutes

HOME - Ria Chaudhari, 202, Coral Nirmal ▾

Search 'cookies'

All Weddings Electronics Beauty Kids Gifting

Welcome

Order now to avail exciting offers!

Flat ₹50 OFF + Free Delivery

Bestsellers

Home Order Again Categories Print

Blinkit in
← 11 minutes
HOME - Ria Chaudhari, 202, Coral Nirmal

Search "gifts"

Grocery & Kitchen

- Vegetables & Fruits
- Atta, Rice & Dal
- Oil, Ghee & Masala
- Dairy, Bread & Eggs
- Bakery & Biscuits
- Dry Fruits & Cereals
- Chicken, Meat & Fish
- Kitchenware & Appliances

Snacks & Drinks

- Chips & Namkeen
- Sweets & Chocolates
- Drinks & Juices
- Tea, Coffee & Milk Drinks
- Instant Food
- Sauces & Spreads
- Paan Corner
- Ice Creams & Desserts

Get FREE delivery on all orders above ₹149

← Bakery & Biscuits

Cookies

Sunfeast Baked Creations Walnut & Choco Chip Cookies
₹134 ADD

Cream Biscuits

Hide & Seek Chocolate Chip Cookies (100 g)
₹29 ADD

Healthy & Digestive

Sweet & Salty

Glucose & Marie

Rusks & Wafers

Cakes & Rolls

MAD & PWA Lab
Journal

Experiment No.	04
Experiment Title.	To create an interactive Form using form widget
Roll No.	7
Name	Ria Chaudhari
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

MPL EXP 4

Aim: To create an interactive Form using form widget

Theory:

Forms in Flutter allow users to input data and validate it before submission. The Form widget acts as a container for form fields like TextFormField, Checkbox, Dropdown, RadioButton, etc.

Key Components of a Form in Flutter:

1. Form Widget → Acts as a container for form elements.
2. GlobalKey<FormState> → Used to manage form validation and submission.
3. TextFormField → Used to take user input.
4. Validation (validator method) → Ensures correct data entry.
5. Submit Button (onPressed) → Saves the form data when clicked.

Syntax for Creating a Form in Flutter

1. Basic Form Structure

```
Form(  
  key: _formKey, // GlobalKey for form validation  
  child: Column(  
    children: [  
      TextFormField(  
        decoration: InputDecoration(labelText: 'Enter your name'),  
        validator: (value) {  
          if (value == null || value.isEmpty) {  
            return 'Please enter your name';  
          }  
          return null;  
        },  
      ),  
      ElevatedButton(  
        onPressed: () {  
          if (_formKey.currentState!.validate()) {  
            print('Form Submitted Successfully');  
          }  
        },  
        child: Text('Submit'),  
      ),  
    ],  
  ),  
)
```

],
),
)

2. Full Example: Interactive Form with Multiple Fields

```
import 'package:flutter/material.dart';
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: FormScreen(),
    );
  }
}

class FormScreen extends StatefulWidget {
  @override
  _FormScreenState createState() => _FormScreenState();
}

class _FormScreenState extends State<FormScreen> { final
  _formKey = GlobalKey<FormState>();
  final TextEditingController nameController = TextEditingController();
  final TextEditingController emailController = TextEditingController();

  void submitForm() {
    if (_formKey.currentState!.validate()) {
      print('Name: ${nameController.text}');
      print('Email: ${emailController.text}');
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Interactive Form')),
      body: Padding(
        padding: EdgeInsets.all(16.0),
        child: Form(
          key: _formKey,
          child: Column(
            children: [
              TextFormField(
                decoration: InputDecoration(
                  labelText: 'Name',
                ),
                controller: nameController,
              ),
              TextFormField(
                decoration: InputDecoration(
                  labelText: 'Email',
                ),
                controller: emailController,
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

```

controller: nameController,
decoration: InputDecoration(labelText: 'Name'),
validator: (value) {
  if (value == null || value.isEmpty) {
    return 'Please enter your name';
  }
  return null;
},
),
TextField(
  controller: emailController,
  decoration: InputDecoration(labelText: 'Email'),
  keyboardType: TextInputType.emailAddress,
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Please enter your email';
    } else if (!value.contains('@')) { return
      'Enter a valid email';
    }
    return null;
},
),
SizedBox(height: 20),
ElevatedButton( onPressed:
submitForm, child:
Text('Submit'),
),
],
),
),
),
);
},
);
}
}

```

Common Widgets & Properties in Forms

Widget	Description	Important Properties
Form	Container for form elements	key (for validation)

TextField	Input field for text entry	controller, keyboardType, validator
-----------	----------------------------	-------------------------------------

TextField	Basic text input without validation	decoration, obscureText (for passwords)
DropDownButton	Dropdown menu for selecting options	items, onChanged, value
CheckBox	Checkbox for selecting options	value, onChanged
Radio	Radio button for selecting one option	value, groupValue, onChanged
Switch	Toggle switch for enabling/disabling options	value, onChanged
ElevatedButton	Button for submitting form	onPressed, child
OutlinedButton	Button with an outline	onPressed, child
IconButton	Button with an icon	icon, onPressed
Padding	Provides space around widgets	padding
SizedBox	Provides spacing between widgets	height, width

Code:

main.dart

```
import 'package:flutter/material.dart';
import 'package:blinkit/screens/signin_screen.dart';
import 'package:blinkit/screens/homepage.dart';
import 'package:blinkit/screens/categories.dart' show CategoriesScreen;
import 'package:blinkit/screens/bakery_biscuits.dart';
```

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

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

```

return MaterialApp(
  debugShowCheckedModeBanner: false,
  title: 'Grocery App',
  theme: ThemeData(
    primarySwatch: Colors.green,
  ),
  initialRoute: '/',
  routes: {
    '/': (context) => SignInScreen(),
    '/home': (context) => HomePage(),
    '/categories': (context) => CategoriesScreen(),
    '/bakery_biscuits': (context) => BakeryBiscuitsPage(),
  },
);
}
}

```

signin_screen.dart

```

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

class SignInScreen extends StatelessWidget { const
  SignInScreen({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: Stack(
        children: [
          // 🖤 Background Image at the Top Positioned(
          top: 0,
          left: 0,
          right: 0,
          child: Container( width:
            double.infinity,
            height: MediaQuery.of(context).size.height *
              0.45, // Takes 45% of screen
            decoration: BoxDecoration(
              image: DecorationImage(
                image:

```

AssetImage("https://via.placeholder.com/400x300.png?text=Placeholder+Image"), fit:
 BoxFit.fitHeight,
),
),
),
),
),
}

```
// ❤️ Bottom White Login Section
Positioned(
    bottom: 0,
    left: 0,
    right: 0,
    child: Container( width:
        double.infinity,
        height: MediaQuery.of(context).size.height * 0.55,
        decoration: BoxDecoration(
            color: Colors.white,
            borderRadius: BorderRadius.vertical(top: Radius.circular(20)),
        ),
        child: Column(
            mainAxisSize: MainAxisSize.min,
            children: [
                // Blinkit Logo Image.asset("assets/blinkit_logo.png",
                height: 50),
                SizedBox(height: 16),

                // Title Text(
                "India's last minute app",
                style: GoogleFonts.lato(
                    fontSize: 20, fontWeight: FontWeight.bold),
                ),
                SizedBox(height: 8),
                Text(
                    "Log in or sign up",
                    style: TextStyle(fontSize: 16, color: Colors.grey[600]),
                ),
                SizedBox(height: 20),

                // Mobile Number Input
                TextField(
                    keyboardType: TextInputType.phone,
                    decoration: InputDecoration( prefixText:
                    "+91 ",
                    hintText: "Enter mobile number", border:
                    OutlineInputBorder(
                        borderRadius: BorderRadius.circular(10)),
                    filled: true,
                    fillColor: Colors.grey[200],
                ),
                ),
                SizedBox(height: 20),

                // Continue Button
            ],
        ),
    ),
);
```

```

SizedBox(
  width: double.infinity, height:
  50,
  child: ElevatedButton(
    onPressed: () {
      // Navigate to Home Screen
      Navigator.pushNamed(context, '/home');
    },
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.grey[700],
      foregroundColor: Colors.white, shape:
      RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(8)),
    ),
    child: Text("Continue", style: TextStyle(fontSize: 18)),
  ),
),
),

SizedBox(height: 16),

// Terms & Conditions
Text(
  "By continuing, you agree to our Terms of service & Privacy policy", textAlign:
  TextAlign.center,
  style: TextStyle(fontSize: 12, color: Colors.grey[600]),
),
],
),
),
],
),
);
}
}
}

```

otpscreen.dart

```

import 'package:flutter/material.dart';

class OTPScreen extends StatefulWidget { @override
  _OTPScreenState createState() => _OTPScreenState();
}

class _OTPScreenState extends State<OTPScreen> {
  final TextEditingController otpController = TextEditingController();

  @override

```

```
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.white,
    body: Padding(
      padding: const EdgeInsets.all(24.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
          // Title
          Text(
            "Enter OTP",
            style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold),
          ),
          SizedBox(height: 10),

          // Instructions
          Text(
            "We have sent a 6-digit code to your phone number.",
            textAlign: TextAlign.center,
            style: TextStyle(fontSize: 16, color: Colors.grey[600]),
          ),
          SizedBox(height: 30),

          // OTP Input Field
          TextField(
            controller: otpController,
            keyboardType: TextInputType.number,
            textAlign: TextAlign.center,
            maxLength: 6,
            style: TextStyle(fontSize: 24, letterSpacing: 8),
            decoration: InputDecoration(
              counterText: "",
              hintText: "     ",
              hintStyle: TextStyle(fontSize: 24, letterSpacing: 8),
              border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(12),
              ),
            ),
            ),
          ),
          SizedBox(height: 20),

          // Verify Button
          SizedBox(
            width: double.infinity,
            height: 50,
            child: ElevatedButton(
              onPressed: () {

```

```
// Add OTP verification logic here
Navigator.pushNamed(context, '/home');
},
style: ElevatedButton.styleFrom(
  backgroundColor: Colors.green[700], shape:
  RoundedRectangleBorder( borderRadius:
  BorderRadius.circular(10),
),
),
),
child: Text( "Verify
OTP",
  style: TextStyle(fontSize: 18, color: Colors.white),
),
),
),
),
SizedBox(height: 15),

// Resend OTP
TextButton( onPressed:
() {
  // Add Resend OTP logic here
},
child: Text(
  "Resend OTP",
  style: TextStyle(fontSize: 16, color: Colors.green[700]),
),
),
],
),
),
);
}
}
```

Output:

The image displays two side-by-side screenshots of a web browser window, likely from a mobile device, showing the sign-up process for a grocery app.

Screenshot 1 (Left):

- Title Bar:** Grocery App
- Address Bar:** localhost:57084
- Content:**
 - A small yellow icon resembling a phone or a delivery van.
 - Text:** India's last minute app
 - Links:** Log in or sign up
 - Form:** A light gray input field labeled "Enter mobile number".
 - Button:** A dark gray button labeled "Continue".
 - Text:** By continuing, you agree to our [Terms of service](#) & [Privacy policy](#)

Screenshot 2 (Right):

- Title Bar:** Grocery App
- Address Bar:** localhost:61687/#/otp
- Content:**
 - Section Header:** Enter OTP
 - Text:** We have sent a 6-digit code to your phone number.
 - Input Field:** An empty input field with a dashed placeholder for a 6-digit code.
 - Button:** A green button labeled "Verify OTP".
 - Text:** Resend OTP

MAD & PWA Lab
Journal

Experiment No.	05
Experiment Title.	To apply navigation, routing and gestures in Flutter App
Roll No.	7
Name	Ria Chaudhari
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

MPL EXP 5

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

Theory:

1. Navigation & Routing

Navigation in Flutter allows users to move between different screens (pages). Flutter uses the Navigator widget to manage screen transitions.

- Named Routes: Uses predefined route names for navigation.
- Push & Pop Navigation: Uses Navigator.push() to move forward and Navigator.pop() to go back.

2. Gestures in Flutter

Flutter provides the GestureDetector widget to detect user interactions like taps, swipes, pinches, and long presses.

Widget	Description	Properties
Navigator	Manages app navigation	push(), pop()
MaterialPageRoute	Defines a new screen transition	builder, settings
GestureDetector	Detects user gestures	onTap, onLongPress, onPanUpdate
InkWell	Detects tap gestures with visual feedback	onTap, borderRadius
WillPopScope	Handles back navigation behavior	onWillPop

Syntax for Navigation & Routing

1. Basic Navigation Using Navigator.push() & pop()

```
Navigator.push(  
  context,  
  MaterialPageRoute(builder: (context) => SecondScreen()),  
)
```

To go back:

```
Navigator.pop(context);
```

2. Named Routes Navigation (Define in main.dart)

Copy

Edit

```
void main() {
  runApp(MaterialApp(
    initialRoute: '/',
    routes: {
      '/': (context) => HomeScreen(),
      '/second': (context) => SecondScreen(),
    },
  )));
}
```

Navigate using:

```
Navigator.pushNamed(context, '/second');
```

Syntax for Gestures in Flutter

1. GestureDetector for Tap and Double Tap

```
GestureDetector(
  onTap: () {
    print('Single tap detected');
  },
  onDoubleTap: () {
    print('Double tap detected');
  },
  child: Container(
    width: 100,
    height: 100,
    color: Colors.blue,
  ),
)
```

2. InkWell for Clickable Widgets with Ripple Effect

```
InkWell(
  onTap: () {
    print('InkWell tapped');
  },
  child: Padding(
    padding: EdgeInsets.all(10),
    child: Text('Tap Me'),
  ),
)
```

Full Example: Navigation & Gestures in Flutter

dart

Copy

Edit

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      initialRoute: '/',
      routes: {
        '/': (context) => HomeScreen(),
        '/second': (context) => SecondScreen(),
      },
    );
  }
}

class HomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Home Screen')),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            // Navigate to Second Screen
            ElevatedButton(
              onPressed: () { Navigator.pushNamed(context, '/second'); },
              child: Text('Go to Second Screen'),
            ),
            SizedBox(height: 20),
            // GestureDetector Example
            GestureDetector(
              onTap: () { print('Box tapped'); },
              onDoubleTap: () {

```

```

        print('Box double tapped');
    },
    child: Container(
        width: 100,
        height: 100,
        color: Colors.blue,
        child: Center(child: Text('Tap Me')),
    ),
),
],
),
),
);
}
}

class SecondScreen extends StatelessWidget {
@Override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(title: Text('Second Screen')), body:
Center(
child: ElevatedButton( onPressed:
() { Navigator.pop(context);
},
child: Text('Go Back'),
),
),
);
}
}

```

Code:

homepage.dart

```

import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:blinkit/screens/categories.dart';

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

class BlinkitClone extends StatelessWidget {

```

```
@override
Widget build(BuildContext context) {
    return MaterialApp(
        debugShowCheckedModeBanner: false,
        initialRoute: '/home',
        routes: {
            '/home': (context) => HomePage(),
            '/categories': (context) => CategoriesScreen(),
        },
    );
}

class HomePage extends StatefulWidget {
    @override
    _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
    int _selectedIndex = 0;

    void _onItemTapped(int index) {
        if (index == 2) {
            Navigator.pushNamed(context, '/categories');
        } else {
            setState(() {
                _selectedIndex = index;
            });
        }
    }

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            backgroundColor: Colors.white,
            bottomNavigationBar: BottomNavigationBar(
                currentIndex: _selectedIndex,
                selectedItemColor: Colors.green,
                unselectedItemColor: Colors.black,
                showUnselectedLabels: true,
                onTap: _onItemTapped,
                items: [
                    BottomNavigationBarItem(icon: Icon(Icons.home), label: "Home"),
                    BottomNavigationBarItem(

```

```
        icon: Icon(Icons.shopping_bag), label: "Order Again"),
    BottomNavigationBarItem(
        icon: Icon(Icons.category), label: "Categories"),
    BottomNavigationBarItem(icon: Icon(Icons.print), label: "Print"),
],
),
body: SingleChildScrollView(
    child: Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: [
            Container(
                color: Colors.yellow[700], padding:
                const EdgeInsets.only(
                    top: 50, left: 16, right: 16, bottom: 10), child:
            Column(
                mainAxisAlignment: MainAxisAlignment.start,
                children: [
                    Row( children:
                    [
                        Icon(Icons.timer, color: Colors.black),
                        const SizedBox(width: 5),
                        Text( "Blinkit
                            in",
                            style: GoogleFonts.poppins(
                                fontSize: 14, fontWeight: FontWeight.w500),
                        ),
                        const SizedBox(width: 5),
                        Text(
                            "10 minutes",
                            style: GoogleFonts.poppins(
                                fontSize: 18, fontWeight: FontWeight.bold),
                        ),
                        Spacer(),
                        Icon(Icons.shopping_cart, color: Colors.black),
                        const SizedBox(width: 10),
                        CircleAvatar(
                            backgroundColor: Colors.white,
                            child: Icon(Icons.person, color: Colors.black),
                        ),
                    ],
                ),
            ),
            const SizedBox(height: 5), Text(
                "HOME - Ria Chaudhari, 202, Coral Nirmal ▼",
            
```

```
        style: GoogleFonts.poppins(
            fontSize: 14, fontWeight: FontWeight.w500),
        ),
        const SizedBox(height: 10),
        Container(
            padding: EdgeInsets.symmetric(horizontal: 10),
            decoration: BoxDecoration(
                color: Colors.white,
                borderRadius: BorderRadius.circular(10),
            ),
            child: Row(
                children: [
                    Icon(Icons.search, color: Colors.black54),
                    Expanded(
                        child: TextField(
                            decoration: InputDecoration(
                                hintText: "Search 'cookies'", border: InputBorder.none, contentPadding: EdgeInsets.symmetric(vertical: 10),
                            ),
                    ),
                    ),
                    ),
                    ],
                ),
            ),
        ),
    ),
    Container(
        padding: EdgeInsets.symmetric(vertical: 10), child:
    Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly, children: [
        _CategoryItem(icon: Icons.shopping_bag, label: "All"),
        _CategoryItem(icon: Icons.favorite, label: "Weddings"),
        _CategoryItem(icon: Icons.tv, label: "Electronics"),
        _CategoryItem(icon: Icons.brush, label: "Beauty"),
        _CategoryItem(icon: Icons.child_care, label: "Kids"),
        _CategoryItem(icon: Icons.card_giftcard, label: "Gifting"),
    ],
),
),
),
```

```
Container(
  padding: EdgeInsets.all(16),
  decoration: BoxDecoration( color:
    Colors.yellow[100],
    borderRadius: BorderRadius.circular(10),
  ),
  child: Column(
    children: [
      Text("Welcome",
        style: GoogleFonts.poppins(
          fontSize: 22, fontWeight: FontWeight.bold)),
      Text("Order now to avail exciting offers!",
        style: GoogleFonts.poppins(fontSize: 14)),
      Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: [
          Text("Flat ₹50 OFF",
            style: GoogleFonts.poppins(
              fontSize: 16,
              fontWeight: FontWeight.bold,
              color: Colors.green)),
          Text("+", style: GoogleFonts.poppins(fontSize: 20)),
          Text("Free Delivery",
            style: GoogleFonts.poppins(
              fontSize: 16,
              fontWeight: FontWeight.bold,
              color: Colors.green)),
        ],
      ),
    ],
  ),
  Padding(
    padding: const EdgeInsets.symmetric(horizontal: 16), child:
    Text(
      "Bestsellers",
      style: GoogleFonts.poppins(
        fontSize: 18, fontWeight: FontWeight.bold),
    ),
  ),
  GridView.count(
    padding: const EdgeInsets.all(16), crossAxisCount:
    3,
    shrinkWrap: true,
```

```
physics: NeverScrollableScrollPhysics(),
children: [
    _BestsellerItem(image: "assets/onion.jpg", count: "+135 more"),
    _BestsellerItem(
        image: "assets/oranges.jpg", count: "+320 more"),
    _BestsellerItem(image: "assets/lays.jpg", count: "+23 more"),
    _BestsellerItem(
        image: "assets/fortune.jpg", count: "+137 more"),
    _BestsellerItem(image: "assets/sprite.jpg", count: "+114 more"),
    _BestsellerItem(
        image: "assets/parle_g.jpg", count: "+135 more"),
],
),
],
),
),
);
}
}
```

```
class _CategoryItem extends StatelessWidget {
final IconData icon;
final String label;
_CategoryItem({required this.icon, required this.label});

@Override
Widget build(BuildContext context) {
return Column(
children: [
Icon(icon, size: 28, color: Colors.black),
Text(label, style: GoogleFonts.poppins(fontSize: 12)),
],
);
}
}
```

```
class _BestsellerItem extends StatelessWidget {
final String image;
final String count;
_BestsellerItem({required this.image, required this.count});

@Override
Widget build(BuildContext context) {
return Column(
```

```
        children: [
          Image.asset(image, height: 50, fit: BoxFit.cover),
          Text(count, style: GoogleFonts.poppins(fontSize: 12)),
        ],
      );
    }
}
```

categories.dart

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

class CategoriesScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Colors.amber.shade100,
        elevation: 0,
        title: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text("Blinkit
in",
              style: TextStyle(fontSize: 14, color: Colors.black54),
            ),
            Text(
              "11 minutes",
              style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold, color: Colors.black),
            ),
            Text(
              "HOME - Ria Chaudhari, 202, Coral Nirmal ▼", style:
              TextStyle(fontSize: 14, color: Colors.black54),
            ),
          ],
        ),
        actions: [
          IconButton(
            icon: Icon(Icons.card_giftcard, color: Colors.black),
            onPressed: () {},
          ),
          IconButton(
            icon: Icon(Icons.person, color: Colors.black),

```

```
        onPressed: () {},
    ),
],
),
body: Column(
    children: [
Padding(
    padding: const EdgeInsets.all(10.0), child:
TextField(
    decoration: InputDecoration(
        hintText: 'Search "gifts"',
        prefixIcon: Icon(Icons.search, color: Colors.black54),
        suffixIcon: Icon(Icons.mic, color: Colors.black54), filled:
        true,
        fillColor: Colors.white, border:
OutlineInputBorder(
            borderRadius: BorderRadius.circular(10),
            borderSide: BorderSide.none,
        ),
    ),
),
),
),
),
Expanded(
    child: SingleChildScrollView(
        child: Padding(
            padding: const EdgeInsets.all(10.0),
            child: Column(
                mainAxisAlignment: MainAxisAlignment.start, children: [
                    buildCategorySection(context, "Grocery & Kitchen", [
                        {"name": "Vegetables & Fruits", "image":
"assets/vegetables_fruits.png"},
                        {"name": "Atta, Rice & Dal", "image": "assets/atta_rice_dal.png"},
                        {"name": "Oil, Ghee & Masala", "image": 'assets/oil_ghee_masala.jpg'},
                        {"name": "Dairy, Bread & Eggs", "image":
"assets/dairy_bread_eggs.png"},
                        {"name": "Bakery & Biscuits", "image": "assets/bakery_biscuits.png"},
                        {"name": "Dry Fruits & Cereals", "image":
"assets/dry_fruits_cereals.png"},
                        {"name": "Chicken, Meat & Fish", "image":
"assets/chicken_meat_fish.png"},
                        {"name": "Kitchenware & Appliances", "image":
"assets/kitchenware_appliances.png"},
                    ]),
                ],
            ),
        ),
    )));

```

```

        buildCategorySection(context, "Snacks & Drinks", [
            {"name": "Chips & Namkeen", "image": "assets/chips_namkeen.png"}, 
            {"name": "Sweets & Chocolates", "image": 
                "assets/sweets_chocolates.png"}, 
            {"name": "Drinks & Juices", "image": "assets/drinks_juices.png"}, 
            {"name": "Tea, Coffee & Milk Drinks", "image": 
                "assets/tea_coffee_milk.png"}, 
            {"name": "Instant Food", "image": "assets/instant_food.png"}, 
            {"name": "Sauces & Spreads", "image": "assets/sauces_spreads.png"}, 
            {"name": "Paan Corner", "image": "assets/paan_corner.png"}, 
            {"name": "Ice Creams & Desserts", "image": 
                "assets/icrecream_desserts.png"}, 
        ]),
    ],
),
),
),
),
),
),
Container(
    padding: EdgeInsets.all(10),
    color: Colors.blue.shade100,
    child: Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
            Icon(Icons.delivery_dining, color: Colors.blue.shade800),
            SizedBox(width: 5),
            Text("Get FREE delivery", style: TextStyle(fontWeight: FontWeight.bold, color: Colors.blue.shade800)),
            SizedBox(width: 5),
            Text("on all orders above ₹149", style: TextStyle(color: Colors.black54)),
        ],
),
),
),
],
),
);
}
}

```

```

Widget buildCategorySection(BuildContext context, String title, List<Map<String, String>> categories) {
    return Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
            SizedBox(height: 10),

```

```
Text(
  title,
  style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
),
GridView.builder( shrinkWrap:
  true,
  physics: NeverScrollableScrollPhysics(),
  gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
    crossAxisCount: 4,
    crossAxisSpacing: 10,
    mainAxisSpacing: 10,
    childAspectRatio: 0.8,
),
itemCount: categories.length,
itemBuilder: (context, index) {
  return GestureDetector(
    onTap: () {
      if (categories[index]["name"] == "Bakery & Biscuits") {
        Navigator.pushNamed(context, '/bakery_biscuits');
      }
    },
    child: Column(
      children: [
        Container(
          height: 60,
          width: 60,
          decoration: BoxDecoration( color:
            Colors.grey.shade200,
            borderRadius: BorderRadius.circular(10),
            image: DecorationImage(
              image: AssetImage(categories[index]["image"]!), fit:
              BoxFit.cover,
            ),
        ),
      ],
      ),
      SizedBox(height: 5),
      Text(
        categories[index]["name"]!,
        textAlign: TextAlign.center,
        style: TextStyle(fontSize: 12),
      ),
    ],
  );
};
```

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

bakery_biscuits.dart
import 'package:flutter/material.dart'; import
'package:provider/provider.dart';
import 'package:blinkit/providers/cart_provider.dart';

class BakeryBiscuitsPage extends StatelessWidget {
    final List<Product> products = [
        Product(name: 'Sunfeast Baked Creations Walnut & Choco Chip Cookies',
imageUrl: 'assets/sunfeast_cookies.png', price: 134.0),
        Product(name: 'Hide & Seek Chocolate Chip Cookies (100 g)', imageUrl:
'assets/hide_seek_100.png', price: 29.0),
        Product(name: 'Hide & Seek Chocolate Chip Cookies (200 g)', imageUrl:
'assets/hide_seek_200.png', price: 45.0),
        Product(name: 'Britannia Good Day Cashew Biscuit', imageUrl: 'assets/brittania_goodday.png',
price: 49.0),
    ];
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            backgroundColor: Colors.amber.shade100,
            elevation: 0,
            title: Text(
                "Bakery & Biscuits",
                style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold, color: Colors.black),
            ),
            leading: IconButton(
                icon: Icon(Icons.arrow_back, color: Colors.black),
                onPressed: () => Navigator.pop(context),
            ),
        ),
        body: Row(
            children: [
                //  Sidebar with categories and images
                Container(
                    width: 100,
```

true),

C O I O R : C O I O R S . W H I T E , C O H I I D :

```

CategoryTile(label: 'Cookies', image: 'assets/cookies.png', isSelected: false),
CategoryTile(label: 'Cream', image: 'assets/healthy_digestive.png'),
CategoryTile(label: 'Sweet & Salty', image: 'assets/sweet_salty.png'), CategoryTile(label: 'Glucose & Marie', image: 'assets/glucose_marie.png'), CategoryTile(label: 'Rusks & Wafers', image: 'assets/rusks_wafers.png'), CategoryTile(label: 'Cakes & Rolls', image: 'assets/cakes_rolls.png'), CategoryTile(label: 'Baking Ingredients', image: 'assets/baking_ingredients.png'),
],
),
),
),
// ✅ Products grid
Expanded(
child: Column(
children: [
Expanded(
child: GridView.builder(
padding: EdgeInsets.all(8),
gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
crossAxisCount: 2,
crossAxisSpacing: 10,
mainAxisSpacing: 10,
childAspectRatio: 0.7,
),
itemCount: products.length,
itemBuilder: (context, index) {
return ProductTile(product: products[index]);
},
),
),
),
),
buildCartButton(context), // ✅ Floating cart button at bottom
],
),
),
),
],
);
}

```

```
// ✅ Floating Cart Button

Widget buildCartButton(BuildContext context) {
  return Consumer<CartProvider>(
    builder: (context, cart, child) { return
      cart.cartItems.isNotEmpty
        ? Container(
            padding: EdgeInsets.all(10),
            color: Colors.green,
            width: double.infinity,
            child: Row(
              mainAxisAlignment: MainAxisAlignment.spaceBetween,
              children: [
                Text("${cart.cartItems.length} items in cart",
                  style: TextStyle(color: Colors.white, fontSize: 16)), ElevatedButton(
                  style: ElevatedButton.styleFrom(backgroundColor: Colors.white),
                  onPressed: () {
                    Navigator.pushNamed(context, '/checkout'); // ✅ Navigate to

```

Checkout Page

```
                },
                child: Text("View Cart", style: TextStyle(color: Colors.green)),
              ),
            ],
          ),
        )
      : SizedBox();
    },
  );
}
}
```

```
class CategoryTile extends StatelessWidget {
  final String label;
  final String image; final
  bool isSelected;
  CategoryTile({required this.label, required this.image, this.isSelected = false});

  Widget build(BuildContext context) {
    return Padding(

```

@override

```
    padding: const EdgeInsets.symmetric(vertical: 8.0),

```

child: Column(

```
children: [
    Container(
        height: 50,
        width: 50,
        decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(10),
            image: DecorationImage(
                image: AssetImage(image), fit:
                    BoxFit.cover,
            ),
        ),
    ),
    SizedBox(height: 5),
    Text(
        label,
        textAlign: TextAlign.center,
        style: TextStyle(
            color: isSelected ? Colors.green : Colors.black,
            fontWeight: isSelected ? FontWeight.bold : FontWeight.normal,
        ),
    ),
],
),
];
);
);
}
}
```

```
class ProductTile extends StatelessWidget { final
Product product;
ProductTile({required this.product});
```

```
@override
```

```
Widget build(BuildContext context) {
    return Card(
        shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(10)), child:
        Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
                Expanded(
                    child: Image.asset(
                        product.imageUrl, fit:
                        BoxFit.cover, width:
```

double.infinity,

```

),
),
Padding(
  padding: const EdgeInsets.all(8.0), child:
  Text(
    product.name,
    style: TextStyle(fontWeight: FontWeight.bold),
  ),
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 8.0),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      Text(
        '₹${product.price}',
        style: TextStyle(color: Colors.green, fontWeight: FontWeight.bold),
      ),
      ElevatedButton(
        onPressed: () {
          Provider.of<CartProvider>(context, listen: false)
            .addToCart(product.name, product.price);
        },
        child: Text('ADD'),
      ),
    ],
),
),
],
),
);
}
}

```

```

class Product { final
  String name;
  final String imageUrl;
  final double price;

```

```

  Product({required this.name, required this.imageUrl, required this.price});
}

```

checkout.dart

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

```
import 'package:provider/provider.dart';
import 'package:blinkit/providers/cart_provider.dart';

class CheckoutPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final cart = Provider.of<CartProvider>(context);

    return Scaffold(
      appBar: AppBar(
        title: Text("Checkout"),
        backgroundColor: Colors.white,
        actions: [
          IconButton(icon: Icon(Icons.share), color: Colors.green), onPressed: () {}),
        ],
      ),
      body: Column(
        children: [
          Container(
            padding: EdgeInsets.all(10),
            child: Row(
              children: [
                Icon(Icons.timer, color: Colors.green),
                SizedBox(width: 5),
                Text("Free delivery in 10 minutes", style: TextStyle(fontWeight: FontWeight.bold)),
              ],
            ),
          ),
          Expanded(
            child: ListView.builder(
              itemCount: cart.cartItems.length,
              itemBuilder: (context, index) { final item = cart.cartItems[index]; return ListTile(
                title: Text(item["name"]),
                trailing: Text("₹${item["price"]}", style: TextStyle(color: Colors.green)),
              );
            },
          ),
        ],
      ),
    );
  }
}
```

```
child: Column(  
    children: [  
        ListTile(  
            title: Text("Total Amount"),  
            trailing: Text("₹${cart.totalAmount}"),  
            style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold)),  
        ),  
        ElevatedButton(  
            style: ElevatedButton.styleFrom(backgroundColor: Colors.green),  
            onPressed: () {  
                cart.clearCart(); //  Clear cart after placing order  
                Navigator.pop(context);  
            },  
            child: Padding(  
                padding: const EdgeInsets.all(12.0),  
                child: Text("Place Order",  
                    style: TextStyle(color: Colors.white, fontSize: 18)),  
            ),  
            ),  
        ],  
        ),  
    ],  
    ),  
);  
}  
}  
}
```

Output:

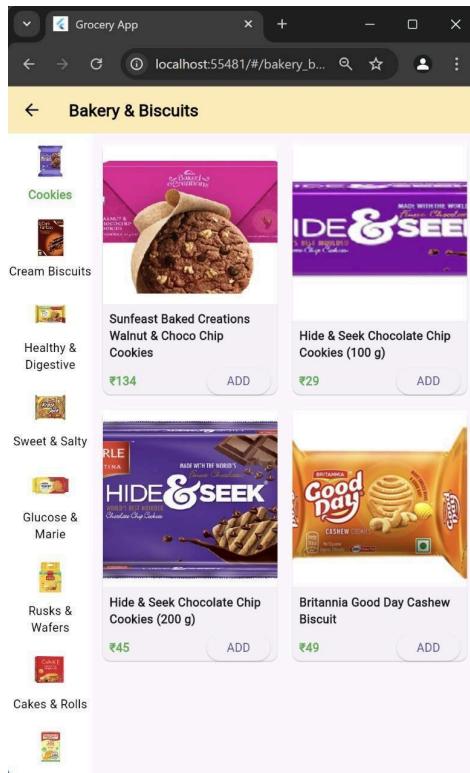
This is the homepage.

The screenshot shows the homepage of a grocery application. At the top, there's a banner with the text "Blinkit in 10 minutes" and a shopping cart icon. Below it, the address "HOME - Ria Chaudhari, 202, Coral Nirmal" is displayed. A search bar with the placeholder "Search 'cookies'" and a microphone icon is present. Below the search bar are six category icons: All, Weddings, Electronics, Beauty, Kids, and Gifting. A yellow header bar below the search bar contains the word "Welcome", a message "Order now to avail exciting offers!", and promotional text "Flat ₹50 OFF" and "Free Delivery". The main content area features a "Bestsellers" section with six product thumbnails: dried fruits, snacks, chips, vegetables, oils, and cereals. At the bottom of the screen, there are four navigation icons: Home (a house), Order Again (a shopping bag), Categories (a grid of icons), and Print (a printer).

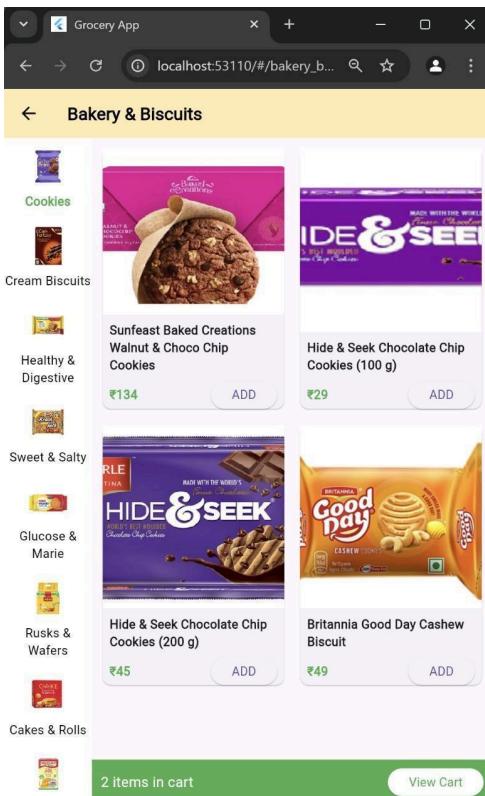
When the user clicks on categories icon at the bottom of the screen it will navigate the user to the categories page.

The screenshot shows the categories page of the grocery application. At the top, there's a banner with the text "Blinkit in 11 minutes" and a shopping cart icon. Below it, the address "HOME - Ria Chaudhari, 202, Coral Nirmal" is displayed. A search bar with the placeholder "Search 'gifts'" and a microphone icon is present. The main content area is divided into several sections: "Grocery & Kitchen" (Vegetables & Fruits, Atta, Rice & Dal, Oil, Ghee & Masala, Dairy, Bread & Eggs); "Snacks & Drinks" (Bakery & Biscuits, Dry Fruits & Cereals, Chicken, Meat & Fish, Kitchenware & Appliances); and "Instant Food" (Chips & Namkeen, Sweets & Chocolates, Drinks & Juices, Tea, Coffee & Milk Drinks). Below these sections, there are four more categories: Instant Food, Sauces & Spreads, Paan Corner, and Ice Creams & Desserts. A blue banner at the bottom of the page offers "Get FREE delivery on all orders above ₹149".

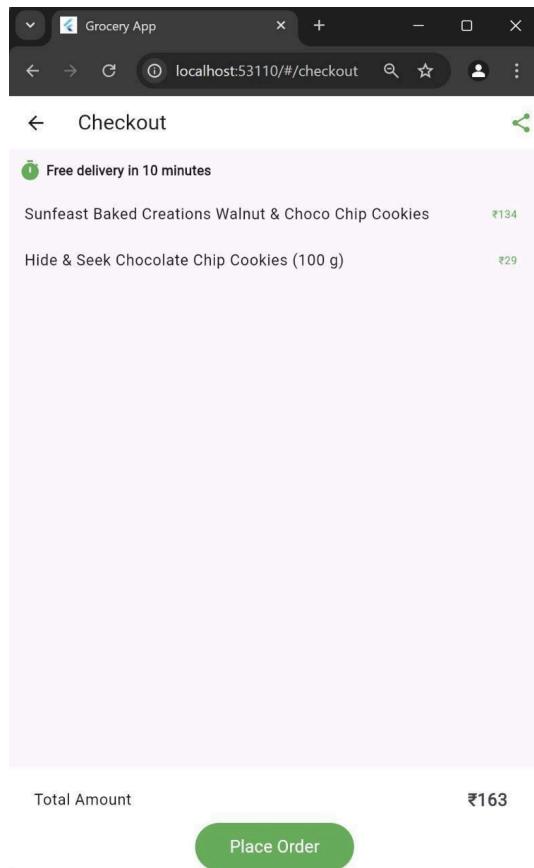
When the user clicks on Bakery and Biscuits under Grocery and Kitchen section it will navigate the user to bakery and biscuits section.



When the user clicks on 'Add' button on an item it will show a floating cart at the bottom of the screen.



When the user clicks on View Cart at the bottom of the screen it will show the user all the products the user has added into the cart along with the total amount the user has to pay and thus navigate to the checkout page.



MAD & PWA Lab
Journal

Experiment No.	06
Experiment Title.	To Connect Flutter UI with fireBase database
Roll No.	7
Name	Ria Chaudhari
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:	

MPL Experiment 6

AIM: - To connect Flutter UI with Firebase Database

Theory: - Firebase is a cloud-based platform by Google that provides backend services like authentication, real-time databases, and cloud storage. In Flutter, Firebase can be integrated using the Firebase SDK to store and retrieve data dynamically. The Cloud Firestore database enables real-time data synchronization, making it ideal for Flutter applications.

Steps to Connect Flutter UI with Firebase Database

1. Create a Firebase Project

- Go to Firebase Console.
- Click on "Add Project" → Configure settings → Create the project.

2. Add Firebase to Flutter App

- Open your Flutter project.

Run: flutter pub add firebase_core firebase_firestore

- Configure Firebase in android/app/google-services.json (for Android) and ios/Runner/GoogleService-Info.plist (for iOS).

3. Initialize Firebase in

Flutter Modify main.dart:

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

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

4. Connect Firestore Database

- In Firebase Console → Firestore → Create a database. Create a Firestore instance in Flutter:

```
import 'package:cloud_firestore/cloud_firestore.dart';
FirebaseFirestore firestore = FirebaseFirestore.instance;
```

5. Perform CRUD Operations

```
Add Data: firestore.collection('users').add({'name': 'John', 'age': 25}); Retrieve
Data: firestore.collection('users').get().then((snapshot) {
for (var doc in snapshot.docs) { print(doc.data()); }
});
```

Update Data: `firebase.collection('users').doc('docId').update({'age': 26})`; Delete Data: `firebase.collection('users').doc('docId').delete()`;

6. Run the App & Test Execute `flutter run` and verify Firebase data operations in Firestore.

This setup enables a Flutter UI to interact with Firebase in real-time, ensuring seamless data storage and retrieval.

Code:

`build.gradle` file:

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

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

    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }

    kotlinOptions {
        jvmTarget = JavaVersion.VERSION_1_8
    }

    defaultConfig {
        // TODO: Specify your own unique Application ID
        // (https://developer.android.com/studio/build/application-id.html).
        applicationId = "com.example.blinkit"
        // You can update the following values to match your application needs.
        // For more information, see: https://flutter.dev/to/review-gradle-config. minSdk
        = flutter.minSdkVersion
        targetSdk = flutter.targetSdkVersion
        versionCode = flutter.versionCode
        versionName = flutter.versionName
        multiDexEnabled true
    }
}
```

```

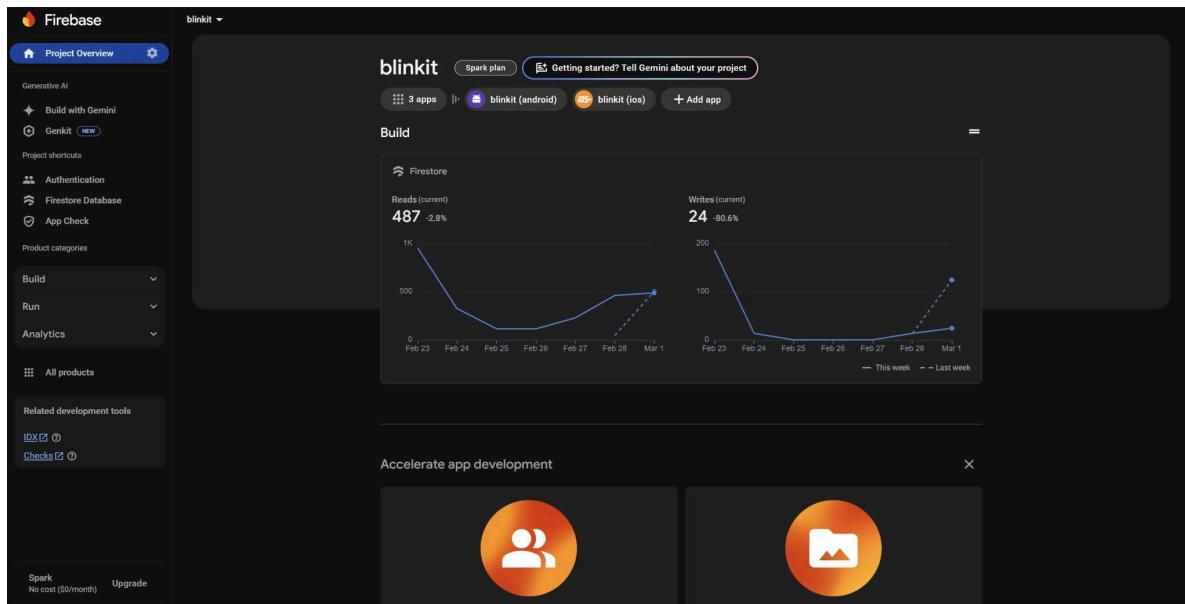
buildTypes {
    release {
        // TODO: Add your own signing config for the release build.
        // Signing with the debug keys for now, so `flutter run --release` works.
        signingConfig = signingConfigs.debug
    }
}
}

flutter {
    source = "../../"
}

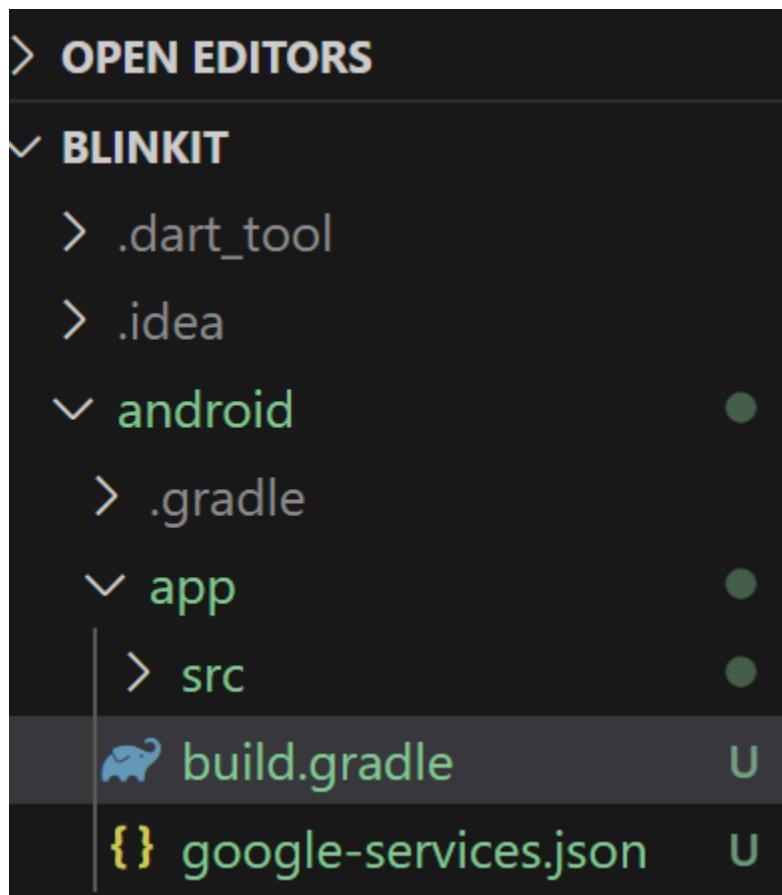
dependencies{
    implementation 'com.google.android.gms:play-services-recaptcha:18.2.0'
    implementation platform('com.google.firebase:firebase-bom:32.7.0') implementation
    'com.google.firebaseio:firebase-appcheck-playintegrity' implementation
    'com.google.firebaseio:firebase-appcheck-recaptcha-enterprise' implementation
    'com.google.firebaseio:firebase-appcheck:17.1.1'
}

apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services' // Add this line

```



The screenshot shows the Firebase Project Overview page for a project named 'blinkit'. The left sidebar includes sections for Generative AI, Build with Gemini, Genkit (NEW), Project shortcuts, Authentication, Firestore Database, App Check, Product categories, Build, Run, Analytics, and Related development tools (IDX, Checks). The main area displays 'Your apps' with three entries: 'Android apps' (blinkit (android) com.example.blinkit selected), 'Apple apps' (blinkit (ios) com.example.blinkit), and 'Web apps' (blinkit (web) Web app). A 'SDK setup and configuration' panel on the right provides instructions to reconfigure the Firebase SDKs, showing fields for App ID (1:881153130994:android:618d7bc94e72713bfab6b), App nickname (blinkit (android)), Package name (com.example.blinkit), SHA certificate fingerprints, and Type (Type). Buttons include 'See SDK instructions', 'google-services.json', 'Add fingerprint', and 'Remove this app'.



Firebase Project Overview Authentication

Sign-in method

Users Sign-in method Templates Usage Settings Extensions

Sign-in providers

Provider Status

- Email/Password Enabled
- Phone Disabled

Add new provider

Advanced

SMS multi-factor authentication

Allow your users to add an extra layer of security to their account. Once enabled, integrated and configured, users can sign in to their account in two steps, using SMS [Learn more](#).

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

Upgrade to enable

Firebase Project Overview Authentication

Sign-in method

Users Sign-in method Templates Usage Settings Extensions

The following authentication features will stop working when Firebase Dynamic Links shuts down on 25 August 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.

Search by email address, phone number or user UID

Add user

Identifier	Providers	Created	Signed in	User UID
ria.chaudhari68@gmail...	✉	20 Feb 2025		hC1t0yKTTUH6pIDNP4A&ix...
2022.ria.chaudhari@ve...	✉	20 Feb 2025	2 Mar 2025	EKBpqT2KM8ci44tgZr0h5zV6...

Rows per page: 50 | 1 - 2 of 2

Firebase Project Overview Cloud Firestore

Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing Configure App Check

Panel view Query builder

(default) category Alta,Rice&Dal

+ Start collection + Add document + Start collection

category

- home
 - Bakery&Biscuits
 - Cleaning Essentials
 - Fruits&Vegetables
 - Home&Office
 - Personal Care

Alta,Rice&Dal

+ Add field

1

image: "https://m.media-amazon.com/images/I/61IAOQVdzHL_AC_UL640_FMwebp_QL6"

name: "Tata Sampann High In Fibre Thick Poha"

price: 54

2

image: "https://cdn.grofers.com/cdn-cgi/image/f=auto,fit=scale-down,q=70,metadata:none,w=270/app/assets/products/sliding_e3db-4bf-f459-143942092454.jpg?ts=1706909777"

name: "Whole Farm Premium Corn Flour (Starch)"

price: 38

3

image: "https://cdn.grofers.com/cdn-cgi/image/f=auto,fit=scale-down,q=70,metadata:none,w=270/app/images/products/sliding"

name: "Whole Farm Grocery Kabuli Chana (Medium Size)"

price: 38

4

Database location: asia-south1

Index	Field	Value
1	image	"https://m.media-amazon.com/images/I/61IAOQVdzHL_AC_UL640_FMwebp_QL6"
1	name	"Tata Sampann High In Fibre Thick Poha"
1	price	54
2	image	"https://cdn.grofers.com/cdn-cgi/image/f=auto,fit=scale-down,q=70,metadata:none,w=270/app/assets/products/sliding_e3db-4bf-f459-143942092454.jpg?ts=1706909777"
2	name	"Whole Farm Premium Corn Flour (Starch)"
2	price	38
3	image	"https://cdn.grofers.com/cdn-cgi/image/f=auto,fit=scale-down,q=70,metadata:none,w=270/app/images/products/sliding"
3	name	"Whole Farm Grocery Kabuli Chana (Medium Size)"
3	price	38

Ria Chaudhari D15A07
Eesha Chavan D15A08
Shraeyaa Dhaigude D15A15

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.	7
Name	Ria Chaudhari
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO4: Understand various PWA frameworks and their requirements
Grade:	

PWA EXP7

Aim: To write meta data of your PWA in a Web app manifest file to enable “add to homescreen 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:

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.

Features of the Progressive Web App

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.

Code:

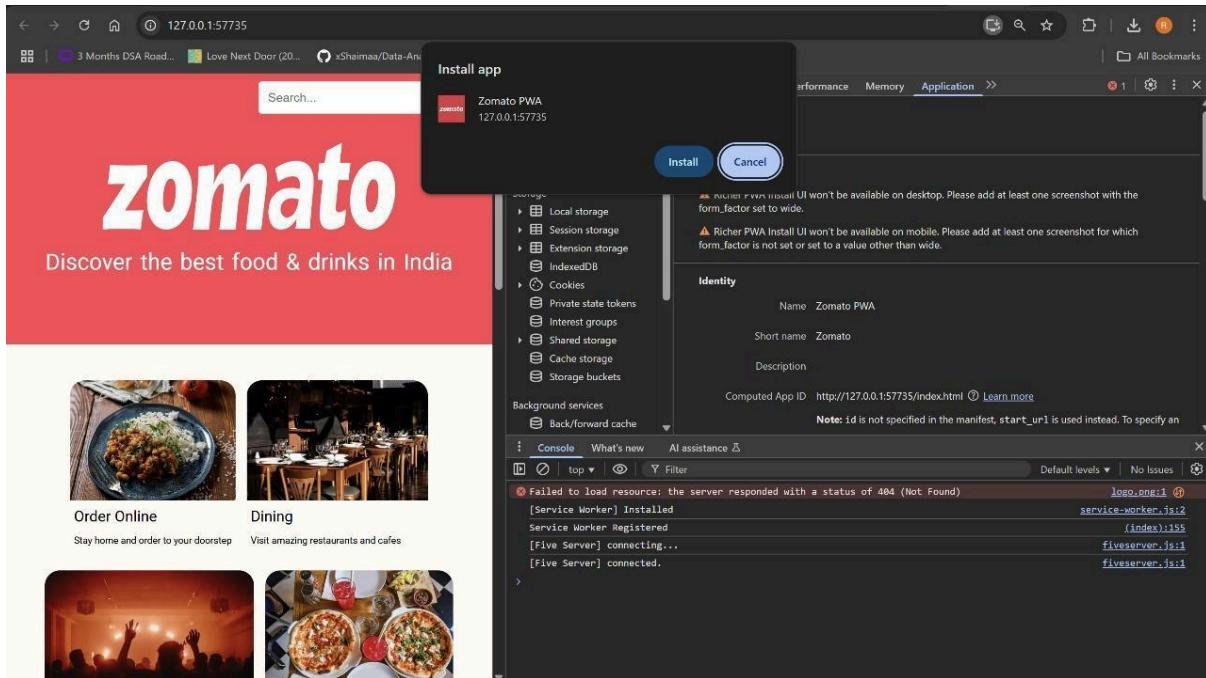
//manifest.json

```
{  
  "name": "Zomato PWA",  
  "short_name": "Zomato",  
  "start_url": "./index.html",  
  "display": "standalone",  
  "background_color": "#ffffff",  
  "theme_color": "#ff5a5f",  
  "orientation": "portrait",  
  "scope": "./",  
  "icons": [  
    {  
      "src": "images/icon-192x192.png", "sizes":  
      "192x192",  
      "type": "image/png"  
    },  
    {  
      "src": "images/icon-512x512.png", "sizes":  
      "512x512",  
      "type": "image/png"  
    }  
  ]  
}
```

//service-worker.js

```
self.addEventListener('install', function(e) {  
  console.log('[Service Worker] Installed');  
});  
  
self.addEventListener('fetch', function(e) {  
  console.log('[Service Worker] Fetch intercepted:', e.request.url);  
});
```

Output:



Shortcut added to home screen



Conclusion:

Hence, we learnt how to write a metadata of our E-commerce website PWA in a Web App Manifest File to enable add to homescreen feature.

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.	7
Name	Ria Chaudhari
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

PWA EXP8

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 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.

Codes:

```
//Serviceworker.js
// sw.js - Complete Service Worker for E-commerce PWA
const CACHE_NAME = "ecommerce-pwa-v2";
const urlsToCache = ["/", "/index.html", "/offline.html", "/css/main.min.css", "/js/app.min.js"];

self.addEventListener("install", (event) => {
  event.waitUntil(
    caches.open(CACHE_NAME).then((cache) => {
      console.log("[Service Worker] Caching files");
      return cache.addAll(urlsToCache);
    })
  );
});

self.addEventListener("activate", (event) => {
  event.waitUntil(
    caches.keys().then((cacheNames) => {
      return Promise.all(
        cacheNames.map((cache) => { if
          (cache !== CACHE_NAME) {
            console.log("[Service Worker] Deleting old cache:", cache);
            return caches.delete(cache);
          }
        })
      );
    })
  );
});
```

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

```
// Fetch Event
// =====
self.addEventListener('fetch', (event) => {
  const { request } = event;
  const url = new URL(request.url);
  // 1. Skip non-GET requests and chrome-extension if
  (request.method !== 'GET' || url.protocol ===
  'chrome-extension:') {
    return;
  }
```

```
// 2. API Requests (Network First with Cache Fallback)
if (url.pathname.startsWith('/api/')) {
  event.respondWith(
    fetch(request)
    .then(networkResponse => {
      // Cache successful API responses if
      (networkResponse.ok) {
        const clone = networkResponse.clone();
        caches.open(API_CACHE)
        .then(cache => cache.put(request, clone));
      }
      return networkResponse;
    })
    .catch(() => {
      // Return cached version if available
      return caches.match(request)
      .then(cachedResponse => cachedResponse || Response.json(
        { error: 'Network error' },
        { status: 503 }
      ));
    })
  );
}
```

```

return;
}

// 3. Static Assets (Cache First with Network Fallback)
event.respondWith(
caches.match(request)
.then(cachedResponse => {
// Return cached version if found if
(cachedResponse) {
return cachedResponse;
}
// Otherwise fetch from network return
fetch(request)
.then(networkResponse => {
// Cache successful responses if
(networkResponse.ok) {
const clone = networkResponse.clone();
caches.open(CACHE_NAME)
.then(cache => cache.put(request, clone));
}
return networkResponse;
})
.catch(() => {
// Special handling for HTML pages
if (request.headers.get('accept').includes('text/html')) { return
caches.match('/offline.html');
}
// Return placeholder for images
if (request.headers.get('accept').includes('image')) { return
caches.match('/images/placeholder-product.jpg');
}
});
});
});
});
});
// =====

```

```

// Background Sync
// =====
self.addEventListener('sync', (event) => { if
(event.tag === 'sync-cart') { event.waitUntil(
// Get cart data from IndexedDB

```

```
getCartData()
.then(cartItems => {
return fetch('/api/cart-sync', {
method: 'POST',
headers: { 'Content-Type': 'application/json' },
body: JSON.stringify(cartItems)
});
})
.then(() => {
return showNotification('Cart Synced', 'Your cart has been updated');
})
.catch(err => { console.error('Sync failed:', err);
})
);
}
);
});
// =====
// Push Notifications
// =====
self.addEventListener('push', (event) => { let
data = {};
try {
data = event.data.json();
} catch (e) {
data = {
title: 'New Update',
body: 'Check out our latest products!', icon:
'/icons/icon-192x192.png',
url: '/'
};
}
const options = {
body: data.body,
icon: data.icon || '/icons/icon-192x192.png',
badge: '/icons/icon-96x96.png',
data: {
url: data.url || '/'
}
};
event.waitUntil(
self.registration.showNotification(data.title, options)
);
```

```
});

self.addEventListener('notificationclick', (event) => {
event.notification.close();
event.waitUntil(
clients.matchAll({ type: 'window' })
.then(clientList => {
for (const client of clientList) {
if (client.url === event.notification.data.url && 'focus' in
client) {
return client.focus();
}
}
if (clients.openWindow) {
return clients.openWindow(event.notification.data.url);
}
})
);
});
// =====
// Helper Functions
// =====
async function getCartData() {
// In a real app, you would use IndexedDB
return new Promise(resolve => { resolve([]); });
}
async function showNotification(title, body) {
return self.registration.showNotification(title, { body });
}
```

Output:

The screenshot shows the Zomato PWA application manifest in the Chrome DevTools Application tab. The manifest.json file is open, displaying the following content:

```
manifest.json
```

App Manifest

Errors and warnings

- Richer PWA Install UI won't be available on desktop. Please add at least one screenshot with the form_factor set to wide.
- Richer PWA Install UI won't be available on mobile. Please add at least one screenshot for which form_factor is not set or set to a value other than wide.
- Actual size (192x192)px of icon http://127.0.0.1:5500/images/icon-192x192.png does not match specified size (512x512px)

Identity

Name: Zomato PWA
Short name: Zomato
Description: Computed App ID: http://127.0.0.1:5500/index.html [Learn more](#)
Note: id is not specified in the manifest, start_url is used instead. To specify an App ID that matches the current identity, set the id field to /index.html.

Presentation

Start URL: /index.html

Console

[Service Worker] Fetch intercepted: http://127.0.0.1:5500/styles.css service-worker.js:6
[Service Worker] Fetch intercepted: http://127.0.0.1:5500/manifest.json service-worker.js:6

The screenshot shows the Zomato PWA service workers in the Chrome DevTools Application tab. The service工人子节点是选中的。右侧显示了当前正在运行的服务工人：

Service workers

http://127.0.0.1:5500 - deleted

Source: service-worker.js
Received: 4/6/2025, 7:18:46 PM
Status: #4704 is redundant

Push: title: 'Hello!', message: 'This is a default push notification' [Push](#)
Sync: test-tag-from-devtools [Sync](#)
Periodic sync: test-tag-from-devtools [Periodic sync](#)

Update Cycle

Version	Update Activity	Timeline
#4704	Install	
#4704	Wait	
#4704	Activate	<div style="width: 100%;"> </div>

Service workers from other origins

[See all registrations](#)

Console

[Service Worker] Fetch intercepted: http://127.0.0.1:5500/styles.css service-worker.js:6
[Service Worker] Fetch intercepted: http://127.0.0.1:5500/manifest.json service-worker.js:6

Ria Chaudhari D15A07
Eesha Chavan D15A08
Shraeyaa Dhaigude D15A15

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.	7
Name	Ria Chaudhari
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

PWA EXP9

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.

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:

```
//Serviceworker.js
// sw.js - Complete Service Worker for E-commerce PWA
const CACHE_NAME = 'ecommerce-pwa-v2';
const API_CACHE = 'ecommerce-api-v1';
const ASSETS_TO_CACHE = [
 '/',
 '/index.html', '/manifest.json',
 '/offline.html',
 '/css/main.min.css',
 '/js/app.min.js',
 '/icons/icon-192x192.png',
 '/icons/icon-512x512.png',
 '/images/placeholder-product.jpg'
];
// =====
// Install Event
// =====
self.addEventListener('install', (event) => {
event.waitUntil( caches.open(CACHE_NAME)
.then((cache) => {
console.log('[Service Worker] Cache opened');
```

```
return cache.addAll(ASSETS_TO_CACHE);
})
.then(() => self.skipWaiting())
);
});
// =====
// Activate Event
// =====
self.addEventListener('activate', (event) => {
event.waitUntil(
caches.keys().then((cacheNames) => { return
Promise.all( cacheNames.map((cacheName)
=> {
if (cacheName !== CACHE_NAME && cacheName !== API_CACHE) {
console.log('[Service Worker] Deleting old cache:',
cacheName);
return caches.delete(cacheName);
}
})
);
});
.then(() => self.clients.claim())
);
});
// =====
// Fetch Event
// =====
self.addEventListener('fetch', (event) => {
const { request } = event;
const url = new URL(request.url);
// 1. Skip non-GET requests and chrome-extension if
(request.method !== 'GET' || url.protocol ===
'chrome-extension:') {
return;
}
// 2. API Requests (Network First with Cache Fallback) if
(url.pathname.startsWith('/api/')) { event.respondWith(
fetch(request)
.then(networkResponse => {
// Cache successful API responses if
(networkResponse.ok) {
const clone = networkResponse.clone();
caches.open(API_CACHE)
```

```
.then(cache => cache.put(request, clone));
}
return networkResponse;
})
.catch(() => {
// Return cached version if available
return caches.match(request)
.then(cachedResponse => cachedResponse || Response.json(
{ error: 'Network error' },
{ status: 503 }
));
})
);
return;
}
// 3. Static Assets (Cache First with Network Fallback)
event.respondWith(
caches.match(request)
.then(cachedResponse => {
// Return cached version if found if
(cachedResponse) {
return cachedResponse;
}
// Otherwise fetch from network return
fetch(request)
.then(networkResponse => {
// Cache successful responses if
(networkResponse.ok) {
const clone = networkResponse.clone();
caches.open(CACHE_NAME)
.then(cache => cache.put(request, clone));
}
return networkResponse;
})
.catch(() => {
// Special handling for HTML pages
if (request.headers.get('accept').includes('text/html')) { return
caches.match('/offline.html');
}
// Return placeholder for images
if (request.headers.get('accept').includes('image')) { return
caches.match('/images/placeholder-product.jpg');
}
});
});
```

```
)  
);  
});  
// =====  
// Background Sync  
// =====  
self.addEventListener('sync', (event) => { if  
(event.tag === 'sync-cart') { event.waitUntil(  
// Get cart data from IndexedDB  
getCartData()  
.then(cartItems => {  
return fetch('/api/cart-sync', {  
method: 'POST',  
headers: { 'Content-Type': 'application/json' },  
body: JSON.stringify(cartItems)  
});  
})  
.then(() => {  
return showNotification('Cart Synced', 'Your cart has been updated');  
})  
.catch(err => { console.error('Sync  
failed:', err);  
})  
);  
}  
});  
// =====  
// Push Notifications  
// =====  
self.addEventListener('push', (event) => { let  
data = {};  
try {  
data = event.data.json();  
} catch (e) {  
data = {  
title: 'New Update',  
body: 'Check out our latest products!', icon:  
'/icons/icon-192x192.png',  
url: '/'  
};  
}  
const options = {
```

```
body: data.body,
icon: data.icon || '/icons/icon-192x192.png',
badge: '/icons/icon-96x96.png',
data: {
url: data.url || '/'
}
};
event.waitUntil(
self.registration.showNotification(data.title, options)
);
});
self.addEventListener('notificationclick', (event) => {
event.notification.close();
event.waitUntil(
clients.matchAll({ type: 'window' })
.then(clientList => {
for (const client of clientList) {
if (client.url === event.notification.data.url && 'focus' in
client) {
return client.focus();
}
}
}
if (clients.openWindow) {
return clients.openWindow(event.notification.data.url);
}
})
);
});
// =====
// Helper Functions
// =====
async function getCartData() {
// In a real app, you would use IndexedDB
return new Promise(resolve => { resolve([]); });
}
async function showNotification(title, body) {
return self.registration.showNotification(title, { body });
}
```

Output:

Fetch event:

The screenshot shows the Zomato homepage with a red header and a search bar. The DevTools Application tab is open, specifically the Service workers panel. It displays the URL `http://127.0.0.1:5501/` and the source file `service-worker.js`. The status shows the service worker is activated and running. A push notification is visible with the message "Your Paneer Tikka Pizza is ready!". The sync section shows a pending sync with the tag "test-tag-from-devtools". The update cycle shows two entries: one for version #4730 labeled "Install" and another for version #4730 labeled "Wait". The console log at the bottom lists several successful fetches from the service worker.

```
[Service Worker] Fetch successful: http://127.0.0.1:5501/manifest.json
[Service Worker] Manifest: property 'background_color' ignored, '#fffff' is not a valid color.
[Service Worker] Fetch successful: http://127.0.0.1:5501/images/icon-192x192.png
[Service Worker] Orders synced!
[Service Worker] Fetch successful: http://127.0.0.1:5501/
[Service Worker] Fetch successful: http://127.0.0.1:5501/styles.css
[Service Worker] Fetch successful: http://127.0.0.1:5501/manifest.json
```

Sync event:

The screenshot shows the Zomato homepage with a red header and a search bar. The DevTools Application tab is open, specifically the Service workers panel. It displays the URL `http://127.0.0.1:5501/` and the source file `service-worker.js`. The status shows the service worker is activated and running. A push notification is visible with the message "Your Paneer Tikka Pizza is ready!". The sync section shows a pending sync with the tag "test-tag-from-devtools". The update cycle shows three entries: one for version #4730 labeled "Install", one for version #4730 labeled "Wait", and one for version #4730 labeled "Activate". The service workers from other origins section shows no registrations. The console log at the bottom shows a sync event received from the devtools.

```
[Service Worker] Sync event received: test-tag-from-devtools
```

Push:

The screenshot shows the Zomato homepage on the left and the Chrome DevTools Application tab on the right.

Zomato Homepage (Left):

- Search bar: Search...
- Logo: zomato
- Slogan: Discover the best food & drinks in India
- Two images: Order Online (a dish) and Dining (a restaurant interior).
- Text: Order Online Stay home and order to your doorstep
- Text: Dining Visit amazing restaurants and cafes
- Image: A group of people at a concert.
- Text: Simulate Push Notification

Chrome DevTools Application Tab (Right):

- Source: service-worker.js
- Status: #4730 activated and is running
- Clients: http://127.0.0.1:5501/
- Push: Your Paneer Tikka Pizza is ready!
- Sync: test-tag-from-devtools
- Periodic sync: test-tag-from-devtools
- Update Cycle: Version #4730, Update Activity Install, Timeline
- Service workers from other origins: See all registrations
- Console output:
 - [Service Worker] Push received
 - [Service Worker] Simulating push...

MAD & PWA Lab Journal

Experiment No.	10
Experiment Title.	To study and implement deployment of Ecommerce PWA to GitHub Pages.
Roll No.	
Name	
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

PWA EXP10

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:

1. Blogging with Jekyll
2. Custom URL
3. Automatic Page Generator

Reasons for favoring this over Firebase:

1. Free to use
2. Right out of github
3. 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

1. Very familiar interface if you are already using GitHub for your projects.
2. Easy to set up. Just push your static website to the gh-pages branch and your website is ready.
3. Supports Jekyll out of the box.
4. 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

1. The code of your website will be public, unless you pay for a private repository.
2. Currently, there is no support for HTTPS for custom domains. It's probably coming soon though.
3. 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:

1. 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.
2. 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.
3. 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:

1. Realtime backend made easy
2. 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

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

Cons

1. 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.
2. Command-line interface only.
3. No in-built support for any static site generator.

Link to our GitHub repository: <https://github.com/riachaudhari/pwa-zomato>

Github Screenshot:

This screenshot shows the GitHub repository page for 'pwa-zomato' owned by 'riachaudhari'. The repository is public and contains one branch ('main') and no tags. The files listed are: icon-192x192.png, icon-512x512.png, index.html, manifest.json, service-worker.js, and styles.css, all added via upload 2 minutes ago. There is a 'README' file present. The 'About' section indicates no description, website, or topics provided. The 'Activity' section shows 0 stars, 1 watching, and 0 forks. The 'Releases' section shows no releases published, with a link to 'Create a new release'. The 'Packages' section shows no packages published, with a link to 'Publish your first package'.

This screenshot shows the GitHub Settings page for the 'pwa-zomato' repository, focusing on the 'GitHub Pages' section. The left sidebar shows various settings categories like General, Access, Collaborators, Moderation options, Code and automation, and Pages (which is selected). The main area displays information about the GitHub Pages site, stating it is live at <https://riachaudhari.github.io/pwa-zomato/> and was last deployed 2 minutes ago. It includes a 'Visit site' button and a '... More' button. Below this, the 'Build and deployment' section shows the source is set to 'Deploy from a branch' (main) and the root directory is '/ (root)'. A note says the site is currently being built from the main branch. The 'Custom domain' section indicates custom domains are not configured. The right sidebar provides links for adding a Jekyll theme and learning more about deploying to GitHub Pages.

Ria Chaudhari D15A07
Eesha Chavan D15A08
Shraeyaa Dhaigude D15A15

MAD & PWA Lab Journal

Experiment No.	11
Experiment Title.	To use google Lighthouse PWA Analysis Tool to test the PWA functioning.
Roll No.	
Name	
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO6: Develop and Analyze PWA Features and deploy it over app hosting solution
Grade:	

PWA EXP11

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:

Measures loading speed, content display time, and overall site efficiency (score out of 100).

- PWA Score (Mobile):

Evaluates adherence to Google's Baseline PWA checklist, including Service Worker implementation, offline functionality, and script-disabled performance.

- Accessibility:

Assesses website accessibility features like aria- attributes, screen-reader compatibility, and semantic HTML tags. Scores are pass/fail.

- Best Practices:

Checks adherence to industry standards, including HTTPS usage and security measures.

Output:

a) Performance

The screenshot displays the Zomato homepage on the left and the Lighthouse performance audit results on the right. The Lighthouse report shows a score of 100 for TBT Performance, with scores of 100 for CLS, 82 for LCP, 89 for FCP, and 93 for TTI. The audit details section lists several service worker logs related to order sync and background tasks.

Lighthouse Audit Results:

- CLS: 100
- LCP: 82
- FCP: 89
- TTI: 93

Audit Details:

- Manifest: property 'background_color' ignored, 'ffffff' is not a valid color. manifest.json:1
- [Service Worker] Fetch successful: http://127.0.0.1:5501/images/icon-192x192.png service-worker.js:41
- [Service Worker] Orders synced! service-worker.js:63
- [Service Worker] Sync event received: sync-orders service-worker.js:53
- [Service Worker] Syncing orders in background... service-worker.js:61
- [Service Worker] Orders synced! service-worker.js:63

Console Output:

- First Contentful Paint: 0.7 s
- Largest Contentful Paint: 0.7 s
- Total Blocking Time: 0 ms
- Cumulative Layout Shift: 0
- Speed Index: 0.7 s

Network Requests:

- manifest.json:1
- icon-192x192.png:41
- service-worker.js:41
- service-worker.js:63
- service-worker.js:53
- service-worker.js:61
- service-worker.js:63

b) Accessibility

The screenshot shows the Zomato homepage on the left and the Lighthouse Accessibility report on the right. The report has a score of 82. It includes sections for CONTRAST, BEST PRACTICES, and a CONSOLE tab with a message about orders being synced.

Accessibility Score: 82

Contrast: Background and foreground colors do not have a sufficient contrast ratio.

Best Practices: Touch targets do not have sufficient size or spacing.

Console: [Service Worker] Orders synced!

c) Best Practices

The screenshot shows the Zomato homepage on the left and the Lighthouse Best Practices report on the right. The report has a score of 89. It includes sections for TRUST AND SAFETY and USER EXPERIENCE, along with a CONSOLE tab.

Best Practices Score: 89

Trust and Safety:

- Requests the notification permission on page load
- Ensure CSP is effective against XSS attacks
- Use a strong HSTS policy
- Ensure proper origin isolation with COOP

User Experience:

- Displays images with incorrect aspect ratio

Console: [Service Worker] orders synced!

d) Search Engine Optimization (SEO)

The image shows the Zomato homepage on the left and the Google Lighthouse SEO analysis results on the right.

Zomato Homepage:

- Header: "Search..."
- Main Logo: "zomato"
- Slogan: "Discover the best food & drinks in India"
- Image: A plate of food.
- Text: "Order Online"
- Image: Dining room interior.
- Text: "Dining"
- Text: "Stay home and order to your doorstep. Visit amazing restaurants and cafes"
- Image: People at a concert.
- Image: Various dishes.
- Text: "Simulate Push Notification"

Google Lighthouse Analysis Results:

- Score: 91
- Core Web Vitals: 198, 82, 89, 93
- SEO Score: 91
- Description: "These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on Core Web Vitals. Learn more about Google Search Essentials."
- Content Best Practices:
 - Document does not have a meta description (warning)
- Additional Items to Manually Check (1): "Format your HTML in a way that enables crawlers to better understand your app's content."
- Console Log:
 - [Service Worker] Orders synced!

Conclusion:

Thus we successfully used google Lighthouse PWA Analysis Tool for testing the PWA functioning.

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.	
Name	
Class	D15A/D15B
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:	

MAD & PWA Lab

Journal

Experiment No.	Assignment-2
Assignment 2 Questions	<ol style="list-style-type: none">1. Define Progressive Web App (PWA) and explain its significance in modern web development. Discuss the key characteristics that differentiate PWAs from traditional mobile apps2. Define responsive web design and explain its importance in the context of Progressive Web Apps. Compare and contrast responsive, fluid, and adaptive web design approaches.3. Describe the lifecycle of Service Workers, including registration, installation, and activation phases.4. Explain the use of IndexedDB in the Service Worker for data storage.
Roll No.	
Name	
Class	D15A/D15B
Subject	MAD & PWA Lab
Lab Outcome	LO4:Understand various PWA frameworks and their requirements LO5: Design and Develop a responsive User Interface by applying PWA Design techniques LO6:Develop and Analyze PWA Features and deploy it over app hosting solutions
Grade:	