Vivekanand Education Society's Institute of Technology

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

Department of Information Technology

CERTIFICATE

This is to certify that **Aryan Anil Patankar** 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

Course Code : ITL604

Project Title: Spotify **Roll No. 33**

Year/Sem/Class **:** D15A **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.

Project Title: Spotify **Roll No. 33**

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	DOP	DOS	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 ,LO3			
13.	Assignment-2	LO4,LO5 ,LO6			

MAD & PWA Lab
Journal

Experiment No.	01
Experiment Title.	To install and configure the Flutter Environment
Roll No.	33
Name	Aryan Anil Patankar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO1: Understand cross platform mobile application development using Flutter framework
Grade:	

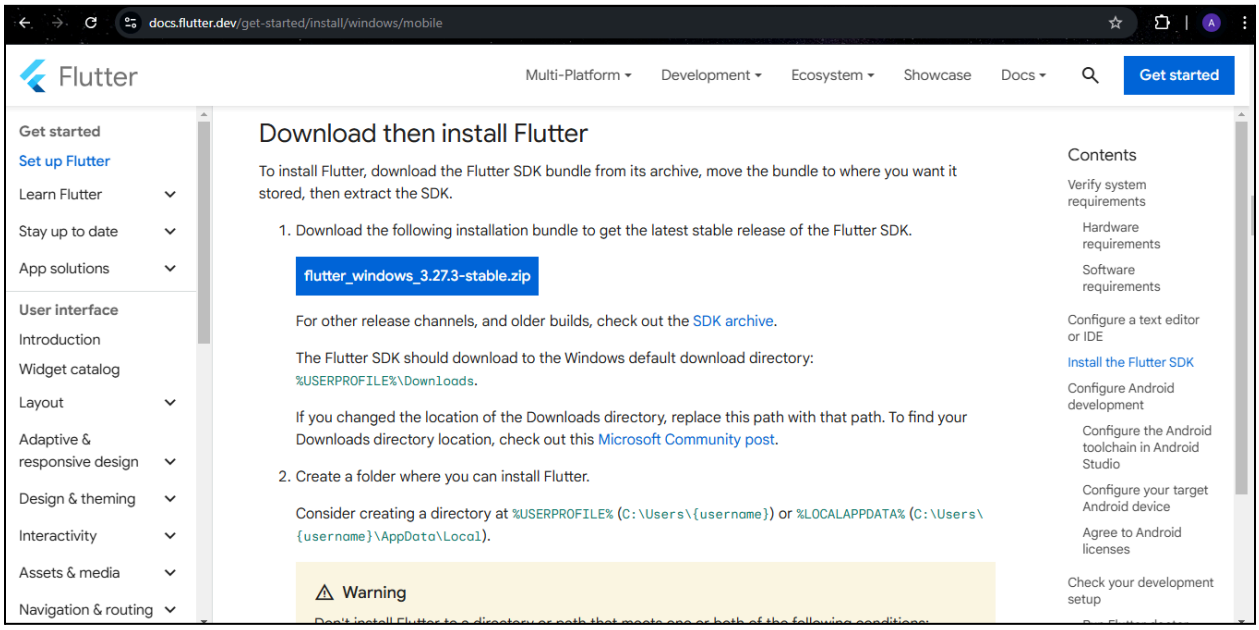
EXPERIMENT NO.1

NAME-ARYAN ANIL PATANKAR
CLASS-D15A
ROLL NO-33

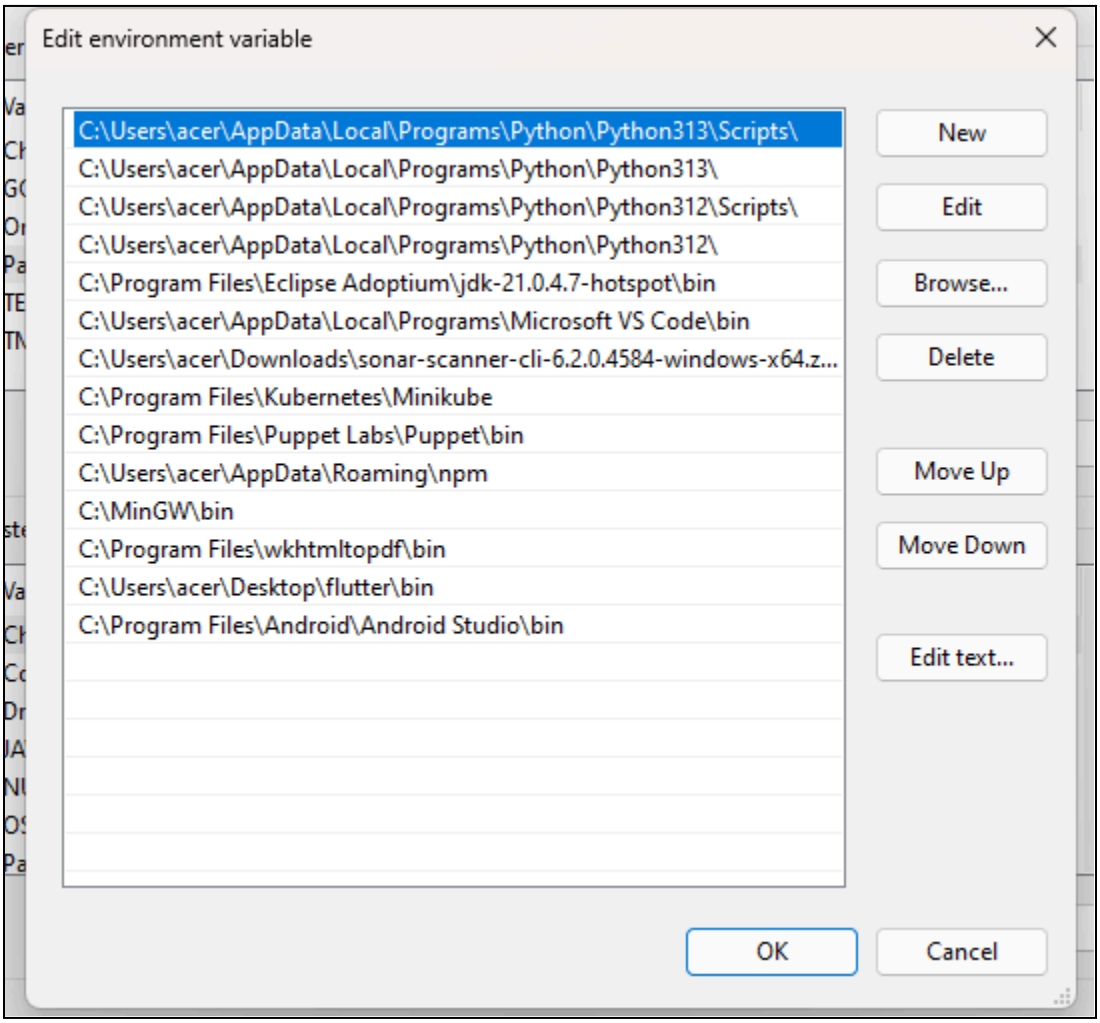
AIM: Installation and Configuration of Flutter Environment.

STEPS:

Install Flutter



Set the environment variables



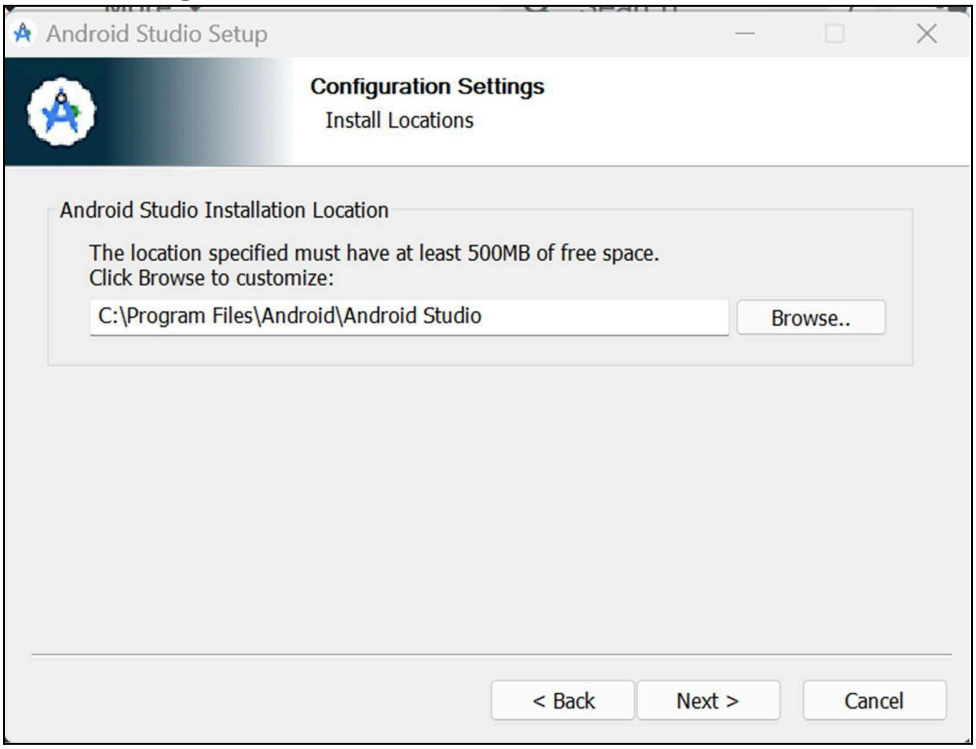
Run flutter command on cmd

```
C:\Users\acer>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:

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

Project
  analyze           Analyze the project's Dart code.
  assemble          Assemble and build Flutter resources.
  build             Build an executable app or install bundle.
  clean             Delete the build/ and .dart_tool/ directories.
  create            Create a new Flutter project.
  drive             Run integration tests for the project on an attached device or emulator.
  gen-l10n          Generate localizations for the current project.
  pub               Commands for managing Flutter packages.
  run               Run your Flutter app on an attached device.
  test              Run Flutter unit tests for the current project.
```

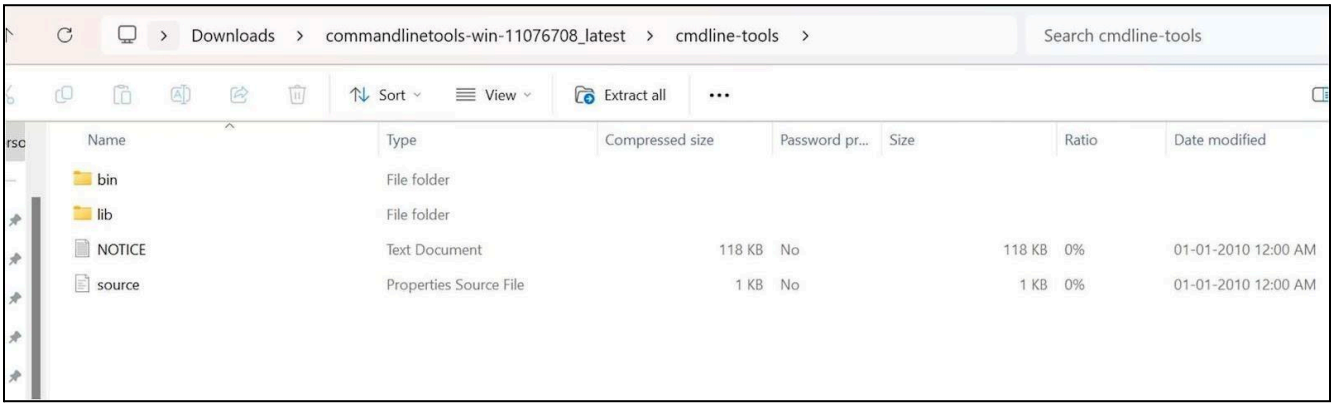
Run flutter doctor

```
C:\Users\HOME\spandan>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.27.3, on Microsoft Windows [Version 10.0.22631.4751], locale
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[X] Android toolchain - develop for Android devices
    X Unable to locate Android SDK.
      Install Android Studio from: https://developer.android.com/studio/index.html
      On first launch it will assist you in installing the Android SDK components.
      (or visit https://flutter.dev/to/windows-android-setup for detailed instructions).
      If the Android SDK has been installed to a custom location, please use
      `flutter config --android-sdk` to update to that location.
```

Downloading Android SDK



Downloaded command line tools of Android SDK



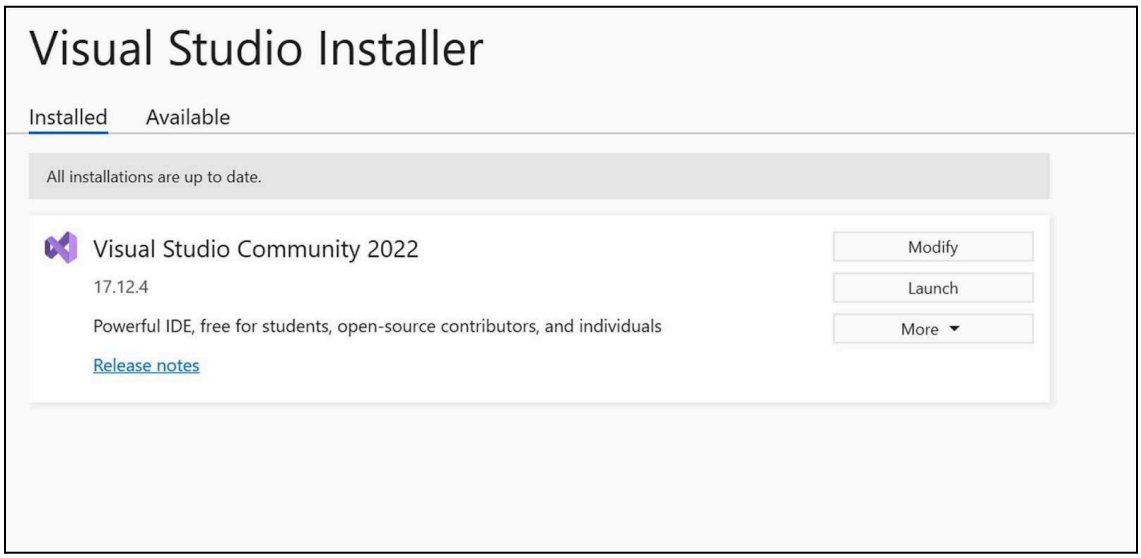
There are two issues according to flutter doctor

```
C:\Users\HOME>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.27.3, on Microsoft Windows [Version 10.0.22631.4751], locale en-IN)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices (Android SDK version 35.0.1)
    ! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses
[✓] Chrome - develop for the web
[X] Visual Studio - develop Windows apps
    X Visual Studio not installed; this is necessary to develop Windows apps.
      Download at https://visualstudio.microsoft.com/downloads/.
      Please install the "Desktop development with C++" workload, including all of its default components
[✓] Android Studio (version 2024.2)
[✓] VS Code (version 1.96.2)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 2 categories.
```

Accept the android licenses

```
C:\Users\HOME>flutter doctor --android-licenses
Warning: This version only understands SDK XML versions up to 3 but an SDK XML file of version 4 was en
you use versions of Android Studio and the command-line tools that were released at different times.
                                                                        Warning: Errors during
Warning: Additionally, the fallback loader failed to parse the XML.ry...
Warning: Errors during XML parse:       ] 66% Fetch remote repository...
Warning: Additionally, the fallback loader failed to parse the XML.
[=======================================] 100% Computing updates...
6 of 7 SDK package licenses not accepted.
Review licenses that have not been accepted (y/N)? y

1/6: License android-googletv-license:
---------------------------------------
Terms and Conditions

This is the Google TV Add-on for the Android Software Development Kit License Agreement.

1. Introduction

1.1 The Google TV Add-on for the Android Software Development Kit (referred to in this License Agreemen
d specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed
```

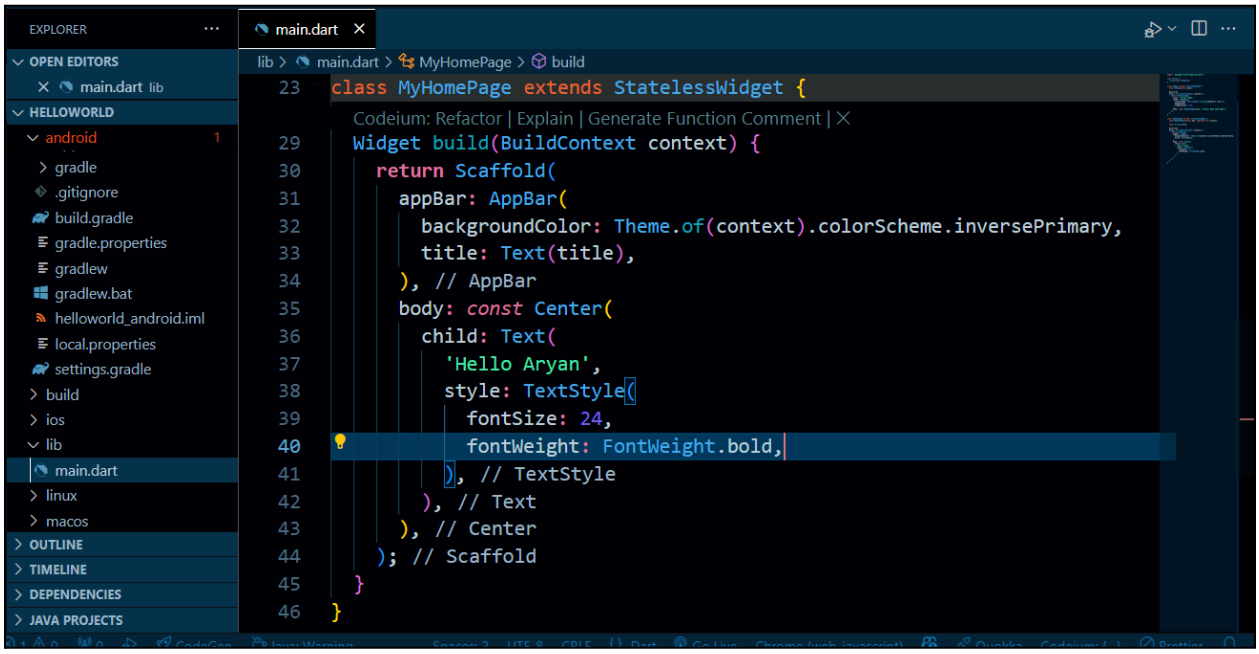
Install Visual Studio



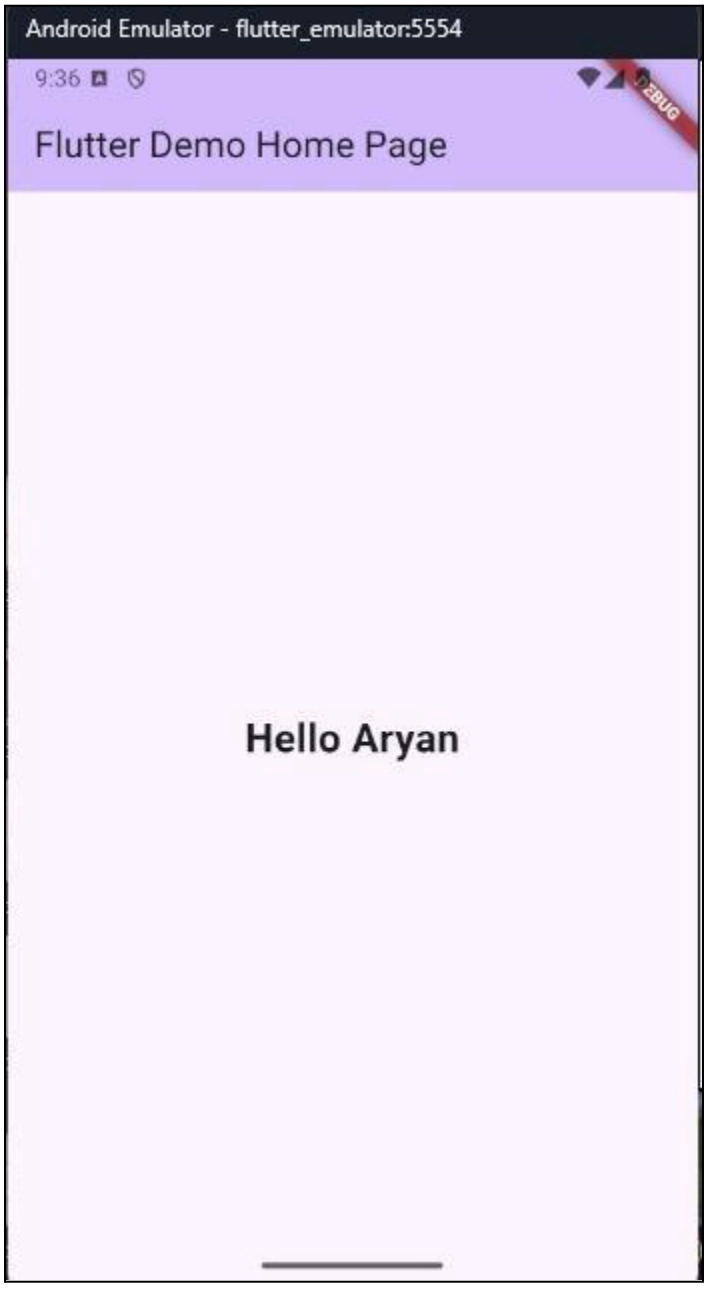
Run flutter doctor command

```
C:\Users\acer>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, 3.27.3, on Microsoft Windows [Version 10.0.22631.4602], locale en-IN)
[√] Windows Version (Installed version of Windows is version 10 or higher)
[√] Android toolchain - develop for Android devices (Android SDK version 35.0.1)
[√] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Build Tools 2022 17.12.3)
    X The current Visual Studio installation is incomplete.
      Please use Visual Studio Installer to complete the installation or reinstall Visual Studio.
[√] Android Studio (version 2024.2)
[√] VS Code (version 1.96.4)
[√] Connected device (3 available)
[√] Network resources
```

Hello world code



Final Output



MAD & PWA Lab
Journal

Experiment No.	02
Experiment Title.	To design Flutter UI by including common widgets.
Roll No.	33
Name	Aryan Anil Patankar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

EXPERIMENT NO.2

NAME-ARYAN ANIL PATANKAR
CLASS-D15A
ROLL NO-33

AIM:To design Flutter UI by including common widgets

1,Introduction

Flutter is an open-source UI framework by Google that enables developers to build cross-platform applications with a single codebase. The UI in Flutter is built using **widgets**, which are the fundamental building blocks of the application.

Widgets in Flutter can be classified into various categories, such as structural widgets, layout widgets, interactive widgets, and styling widgets. These widgets help in designing complex UI components efficiently and responsively.

2.Types of widgets

Flutter widgets are classified into the following types based on their functionality:

1. Structural Widgets (Scaffolding Widgets)

These widgets define the basic structure of an application.

- **Scaffold:** Provides a framework with an app bar, body, floating action button, drawer, etc.
- **AppBar:** Displays the top navigation bar with titles, actions, and menus.
- **Drawer:** A side navigation panel for navigating through different screens.

2. Layout Widgets

These widgets help in arranging other widgets on the screen.

- **Container:** A flexible widget for holding and styling child widgets.
- **Column:** Arranges widgets vertically.
- **Row:** Arranges widgets horizontally.

- **Stack:** Allows widgets to be placed on top of each other.
- **Expanded:** Expands widgets to fill available space.
- **SizedBox:** Provides spacing between widgets.

3. Interactive Widgets

These widgets accept user input and interactions.

- **TextField:** Allows users to enter text input.
- **ElevatedButton:** A clickable button with elevation.
- **OutlinedButton:** A button with an outlined border.
- **IconButton:** A clickable icon button.
- **GestureDetector:** Detects gestures like tap, swipe, and drag.
- **Switch:** A toggle button for enabling/disabling options.

4. Styling & Display Widgets

These widgets help in styling and displaying content.

- **Text:** Displays text content.
- **Image:** Displays an image from assets or a URL.
- **Icon:** Displays an icon from Material or custom icons.
- **Card:** A box with elevation and rounded corners for content display.
- **Chip:** A small, compact element that represents information.

5. Scrolling & List Widgets

These widgets handle large lists and scrolling content.

- **ListView:** Displays a scrollable list of items.
- **GridView:** Displays items in a grid format.
- **SingleChildScrollView:** Makes a child widget scrollable.
- **PageView:** Creates swipeable pages.

3.**Commonly used widgets:**

1. Scaffold (Basic Page Structure)

A *Scaffold* provides a basic structure for a screen, including an **AppBar, Floating Action Button, Drawer, and Body**.

2.**Container (A Flexible Box for UI Design)**

A *Container* is used for styling, padding, and positioning elements.

3.**Row and Column (Arranging Widgets)**

A *Row* arranges widgets horizontally, while a *Column* arranges them vertically.

4.**Stack (Overlapping Widgets)**

A Stack places widgets on top of each other.

5.**ListView (Scrollable List)**

A ListView is used for displaying a scrollable list of items.

6.**ElevatedButton (Clickable Button)**

Buttons provide user interactions in Flutter.

7.**TextField (User Input Field)**

A TextField allows users to enter data.

CODE

Genre_card.dart

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

class GenreCard extends StatelessWidget {
  final String title;
  final Color color;

  const GenreCard({
    Key? key,
    required this.title,
    required this.color,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Container(
      decoration: BoxDecoration(
        color: color,
        borderRadius: BorderRadius.circular(4),
      ),
      child: Center(
        child: Text(
          title,
          style: const TextStyle(
            fontSize: 16,
            fontWeight: FontWeight.bold,
          ),
        ),
      ),
    );
  }
}

width: 48,
```

Now_playing_bar.dart

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


class NowPlayingBar extends StatelessWidget {
  const NowPlayingBar({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
   return Container(
    color: Colors.grey[900],
    padding: const EdgeInsets.all(8),
    child: SafeArea(
     top: false,
     child: Row(
      children: [
       ClipRRect(
        borderRadius: BorderRadius.circular(4),
        child: Image.asset(
         height: 48,
         fit: BoxFit.cover,
        ),
       ),
       const SizedBox(width: 12),
       const Expanded(
        child: Column(
         crossAxisAlignment: CrossAxisAlignment.start,
         mainAxisSize: MainAxisSize.min,
         children: [
          Text(
           'Perfect',
           style: TextStyle(fontWeight: FontWeight.bold),
          ),
          Text(
           'One Direction',
```

Playlist_card.dart

```
            'assets/album_cover.jpg',
              style: TextStyle(color:
Colors.grey, fontSize: 12),
            ),
          ],
        ),
      ),
      IconButton(
        icon: const
Icon(Icons.play_arrow),
        onPressed: () {},
      ),
    ],
  ),
 ),
 );
 }
}
```

```
import 'package:flutter/material.dart';
class PlaylistCard extends
StatelessWidget {
 final String title;
 final String saves;
 const PlaylistCard({
  Key? key,
  required this.title,
  required this.saves,
 }) : super(key: key);

 @override
 Widget build(BuildContext context) {
  return Container(
   padding: const EdgeInsets.all(16),
   decoration: BoxDecoration(
    color: Colors.grey[900],
    borderRadius:
BorderRadius.circular(4),
   ),
 style: const TextStyle(fontWeight:
FontWeight.bold),
```

```
        children: [
          Container(
            width: 48,
            height: 48,
            decoration: BoxDecoration(
              color: Colors.grey[800],
              borderRadius:
BorderRadius.circular(4),
            ),
            child: const
Icon(Icons.music_note),
          ),
          const SizedBox(width: 16),
          Expanded(
            child: Column(
              crossAxisAlignment:
CrossAxisAlignment.start,
              children: [
                Text(
                  title,
                ),
                Text(
                  saves,
                  style: const TextStyle(color:
Colors.grey),
                ),
              ],
            ),
          ),
        ],
      ),
    );
  }
}
```

```
padding: const EdgeInsets.all(12),
        decoration: BoxDecoration(
```

Social_login.dart

```
//
lib/features/auth/widgets/social_login_
button.dart

import 'package:flutter/material.dart';

class SocialLoginButton extends
StatelessWidget {
  final String assetPath;
  final VoidCallback onTap;


  const SocialLoginButton({
    super.key,
    required this.assetPath,
    required this.onTap,
  });


  @override
  Widget build(BuildContext context) {
    return InkWell(
      onTap: onTap,
      child: Container(
        border: Border.all(color:
Colors.grey[300]!),
        borderRadius:
BorderRadius.circular(8),
      ),
      child: Image.asset(
        assetPath,
        height: 24,
        width: 24,
      ),
    ),
  );
 }
}
```

```
bodyLarge: TextStyle(color:
Colors.white),
```

Home_screen.dart

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


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


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


  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Spotify Clone',
      theme: ThemeData(
        scaffoldBackgroundColor:
Colors.black,
        brightness: Brightness.dark,
        textTheme: const TextTheme(
          bodyMedium: TextStyle(color:
Colors.white),
          titleLarge: TextStyle(color:
Colors.white),
          titleMedium: TextStyle(color:
Colors.white),
        ),
      ),
      home: const HomeScreen(),
    );
  }
}


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


  @override
  Widget build(BuildContext context) {
    return Scaffold(
children: [
```

```
body: Stack(
 children: [
  Container(
   decoration: BoxDecoration(
    gradient: LinearGradient(
     begin: Alignment.topCenter,
     end: Alignment.bottomCenter,
     colors: [
      Colors.black.withOpacity(0.8),
      Colors.black,
     ],
    ),
   ),
  ),
  SingleChildScrollView(
   child: SafeArea(
    child: Column(
     crossAxisAlignment: CrossAxisAlignment.start,
      Padding(
       padding: const EdgeInsets.all(16.0),
       child: Row(
        children: [
         const CircleAvatar(
          backgroundColor: Color(0xFF282828),
          child: Text('A', style: TextStyle(color: Colors.white)),
         ),
         const SizedBox(width: 16),
         Container(
          padding: const EdgeInsets.symmetric(horizontal: 16, vertical: 8),
          decoration: BoxDecoration(
           color: const Color(0xFF1DB954),
           borderRadius: BorderRadius.circular(20),
Container(
```

```
                ),
                child: const Text('All',
style: TextStyle(color: Colors.white)),
              ),
              const SizedBox(width:
8),
              Container(
                padding: const
EdgeInsets.symmetric(horizontal: 16,
vertical: 8),
                decoration:
BoxDecoration(
                  color: const
Color(0xFF282828),
                  borderRadius:
BorderRadius.circular(20),
                ),
                child: const
Text('Music', style: TextStyle(color:
Colors.white)),
              ),
              const SizedBox(width:
8),
                padding: const
EdgeInsets.symmetric(horizontal: 16,
vertical: 8),
                decoration:
BoxDecoration(
                  color: const
Color(0xFF282828),
                  borderRadius:
BorderRadius.circular(20),
                ),
                child: const
Text('Podcasts', style: TextStyle(color:
Colors.white)),
              ),
            ],
          ),
        ),
        const PlaylistSection(
          title: 'On Repeat',
          image:
'assets/images/on_repeat.jpg',
          isLarge: true,
children: const [
            MixCard(
```

```
            ),
            const PlaylistSection(
              title: 'Trending Now India',
              image:
'assets/images/trending_india.jpg',
              isLarge: true,
            ),
            const PlaylistSection(
              title: 'Liked Songs',
              image:
'assets/images/liked_songs.jpg',
              isLarge: true,
            ),
            const SectionTitle(title:
'Your top mixes'),
            SingleChildScrollView(
              scrollDirection:
Axis.horizontal,
              padding: const
EdgeInsets.symmetric(horizontal: 16),
              child: Row(
                title: 'Arijit Singh Mix',
                subtitle: 'Pritam,
Vishal-Shekhar and Atif Aslam',
                image:
'assets/images/arijit.jpg',
              ),
              SizedBox(width: 16),
              MixCard(
                title: 'Chill Mix',
                subtitle: 'Kendrick
Lamar, The Weeknd, Future',
                image:
'assets/images/chill.jpg',
              ),
              SizedBox(width: 16),
              MixCard(
                title: 'R&B Mix',
                subtitle: 'Kali Uchis,
SZA',
                image:
'assets/images/rnb.jpg',
type: BottomNavigationBarType.fixed,
      selectedItemColor: Colors.white,
```

```
                                ),
                  ],
                ),
              ),
              const SizedBox(height:
100), // Space for bottom player
            ],
          ),
        ),
      ),
      const Positioned(
        left: 0,
        right: 0,
        bottom: 0,
        child: NowPlayingBar(),
      ),
    ],
  ),
  bottomNavigationBar:
BottomNavigationBar(
    backgroundColor: Colors.black,
    unselectedItemColor: Colors.grey,
    items: const [
      BottomNavigationBarItem(
        icon: Icon(Icons.home_filled),
        label: 'Home',
      ),
      BottomNavigationBarItem(
        icon: Icon(Icons.search),
        label: 'Search',
      ),
      BottomNavigationBarItem(
        icon: Icon(Icons.library_music),
        label: 'Your Library',
      ),
      BottomNavigationBarItem(
        icon: Icon(Icons.person),
        label: 'Profile',
      ),
return Padding(
  padding: const EdgeInsets.all(16.0),
```

```
            ],
      ),
    );
  }
}


class PlaylistSection extends
StatelessWidget {
  final String title;
  final String image;
  final bool isLarge;

  const PlaylistSection({
    Key? key,
    required this.title,
    required this.image,
    this.isLarge = false,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    child: Row(
      children: [
        ClipRRect(
          borderRadius:
BorderRadius.circular(4),
          child: Image.asset(
            image,
            width: isLarge ? 80 : 60,
            height: isLarge ? 80 : 60,
            fit: BoxFit.cover,
          ),
        ),
        const SizedBox(width: 16),
        Expanded(
          child: Text(
            title,
            style: const TextStyle(
              fontSize: 16,


required this.title,
    required this.subtitle,
```

```
                fontWeight:
FontWeight.w600,
              color: Colors.white,
            ),
          ),
        ),
        const Icon(Icons.more_vert,
color: Colors.white),
      ],
    ),
  );
 }
}


class MixCard extends StatelessWidget
{
 final String title;
 final String subtitle;
 final String image;


 const MixCard({
   Key? key,
  required this.image,
 }) : super(key: key);


 @override
 Widget build(BuildContext context) {
  return SizedBox(
   width: 160,
   child: Column(
     crossAxisAlignment:
CrossAxisAlignment.start,
     children: [
      ClipRRect(
       borderRadius:
BorderRadius.circular(4),
       child: Image.asset(
        image,
        width: 160,
        height: 160,
        fit: BoxFit.cover,
),
  );
```

```
    ),
  ),
  const SizedBox(height: 8),
  Text(
    title,
    style: const TextStyle(
      fontWeight: FontWeight.w600,
      color: Colors.white,
    ),
  ),
  const SizedBox(height: 4),
  Text(
    subtitle,
    style: const TextStyle(
      color: Colors.grey,
      fontSize: 12,
    ),
  ),
 ],
    }
}

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

 @override
 Widget build(BuildContext context) {
  return Container(
    color: const Color(0xFF282828),
    padding: const EdgeInsets.all(8),
    child: Row(
     children: [
      Image.asset(
       'assets/images/perfect_1d.jpg',
       width: 40,
       height: 40,
  ),
          ),
```

```
        ),
        const SizedBox(width: 12),
        const Expanded(
          child: Column(
            crossAxisAlignment:
CrossAxisAlignment.start,
            mainAxisSize:
MainAxisSize.min,
            children: [
              Text(
                'Perfect',
                style: TextStyle(
                  fontWeight:
FontWeight.w600,
                  color: Colors.white,
                ),
              ),
              Text(
                'One Direction',
                style: TextStyle(
                  color: Colors.grey,
                  fontSize: 12,
            ],
          ),
        ),
        IconButton(
          icon: const
Icon(Icons.devices_outlined, color:
Colors.white),
          onPressed: () {},
        ),
        IconButton(
          icon: const
Icon(Icons.favorite_border, color:
Colors.white),
          onPressed: () {},
        ),
        IconButton(
          icon: const
Icon(Icons.play_circle_filled, color:
Colors.white),
          onPressed: () {},
        ),
    fontWeight: FontWeight.bold,
        color: Colors.white,
```

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

class SectionTitle extends
StatelessWidget {
  final String title;

  const SectionTitle({Key? key,
required this.title}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(16.0),
      child: Text(
        title,
        style: const TextStyle(
          fontSize: 24,
                                          child: Column(
                                                  crossAxisAlignment:
                                          CrossAxisAlignment.start,
```

Profile_page.dart

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


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


  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.black,
      body: Column(
        children: [
          const Expanded(flex: 2, child: _TopPortion()),
          Expanded(
            flex: 3,
            child: Padding(
              padding: const EdgeInsets.all(16.0),
              children: [
                Text(
                  "Aryan",
                  style: Theme.of(context).textTheme.headlineSmall?.copyWith(
                        color: Colors.white,
                        fontWeight: FontWeight.bold,
                      ),
                ),
                const SizedBox(height: 8),
                const _ProfileInfoRow(),
                const SizedBox(height: 24),
                Text(
                  "Playlists",
                  style: Theme.of(context).textTheme.titleLarge?.copyWith(
                        color: Colors.white,
    );
  }
```

```
                        fontWeight:
FontWeight.bold,
                    ),
                ),
                const SizedBox(height: 16),
                const _PlaylistCard(
                  title: "My playlist #3",
                  saves: "0 saves",
                ),
                const SizedBox(height: 12),
                const _PlaylistCard(
                  title: "CHILL",
                  saves: "0 saves",
                ),
              ],
            ),
          ),
        ),
        const _BottomNavBar(),
      ],
    ),
  }


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


  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const
EdgeInsets.symmetric(vertical: 8.0),
      child: Row(
        children: [
          Text(
            "1 follower",
            style:
Theme.of(context).textTheme.bodyMe
dium?.copyWith(
                color: Colors.grey,
const _PlaylistCard({
    Key? key,
```

```
            ),
        ),
        const Text(" • ", style:
TextStyle(color: Colors.grey)),
        Text(
          "19 following",
          style:
Theme.of(context).textTheme.bodyMe
dium?.copyWith(
              color: Colors.grey,
            ),
        ),
      ],
    ),
  );
 }
}


class _PlaylistCard extends
StatelessWidget {
 final String title;
 final String saves;
 required this.title,
 required this.saves,
}) : super(key: key);


@override
Widget build(BuildContext context) {
 return Container(
  padding: const EdgeInsets.all(12),
  decoration: BoxDecoration(
   color: Colors.grey[900],
   borderRadius:
BorderRadius.circular(8),
  ),
  child: Row(
   children: [
    Container(
     width: 60,
     height: 60,


),
        const SizedBox(height: 4),
```

```
            decoration: BoxDecoration(
            color: Colors.grey[800],
            borderRadius:
BorderRadius.circular(4),
          ),
          child: const
Icon(Icons.music_note, color:
Colors.white),
        ),
        const SizedBox(width: 12),
        Column(
          crossAxisAlignment:
CrossAxisAlignment.start,
          children: [
            Text(
              title,
              style: const TextStyle(
                color: Colors.white,
                fontWeight:
FontWeight.bold,
              ),
                Text(
                  saves,
                  style: TextStyle(color:
Colors.grey[400]),
                ),
              ],
            ),
          ],
        ),
      );
    }
}


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

  @override
  Widget build(BuildContext context) {
 IconButton(
```

```
    return Stack(
      children: [
        Container(
          decoration: const
BoxDecoration(
            gradient: LinearGradient(
              begin: Alignment.topCenter,
              end: Alignment.bottomCenter,
              colors: [Color(0xFF9370DB),
Colors.black],
            ),
          ),
        ),
        SafeArea(
          child: Padding(
            padding: const
EdgeInsets.all(16.0),
            child: Column(
              children: [
                Row(
                  children: [
                    icon: const
Icon(Icons.arrow_back, color:
Colors.white),
                    onPressed: () {
                      Navigator.pop(context);
                    },
                  ),
                  const Spacer(),
                  IconButton(
                    icon: const
Icon(Icons.more_vert, color:
Colors.white),
                    onPressed: () {},
                  ),
                ],
              ),
              const SizedBox(height: 20),
              Container(
                width: 100,
                height: 100,
                decoration: BoxDecoration(
);
```

```
                        color:
Colors.purple[200],
                    shape: BoxShape.circle,
                  ),
                  child: Center(
                    child: Text(
                      "A",
                      style:
Theme.of(context).textTheme.displayL
arge?.copyWith(
                            color: Colors.black,
                            fontWeight:
FontWeight.bold,
                          ),
                    ),
                  ),
                ),
              ),
            ],
          ),
        ),
      ),
    ],
  }
}

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

  @override
  Widget build(BuildContext context) {
    return Container(
      padding: const
EdgeInsets.symmetric(vertical: 12),
      color: Colors.black,
      child: Row(
        mainAxisAlignment:
MainAxisAlignment.spaceEvenly,
        children: [
          _NavBarItem(
            icon: Icons.home,
            label: 'Home',
}
```

```
            onTap: () {},
        ),
        _NavBarItem(
          icon: Icons.search,
          label: 'Search',
          onTap: () {},
        ),
        _NavBarItem(
          icon: Icons.library_music,
          label: 'Your Library',
          onTap: () {},
        ),
        _NavBarItem(
          icon: Icons.person,
          label: 'Profile',
          onTap: () {},
        ),
      ],
    ),
  );
}

class _NavBarItem extends StatelessWidget {
  final IconData icon;
  final String label;
  final VoidCallback onTap;

  const _NavBarItem({
    Key? key,
    required this.icon,
    required this.label,
    required this.onTap,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return InkWell(
      onTap: onTap,
      child: Column(
```

```
        mainAxisSize:
MainAxisSize.min,
      children: [
        Icon(icon, color: Colors.white),
        const SizedBox(height: 4),
        Text(
          label,
          style: const TextStyle(
            color: Colors.white,
            fontSize: 12,
          ),
        ),
      ],
    ),
  );
 }
}
```

Search_page:

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

import
'package:flutter_riverpod/flutter_riverp
od.dart';

import
'package:spotify_clone/models/search_
result.dart';

import
'package:spotify_clone/services/youtub
e_service.dart';

import
'package:spotify_clone/providers/audio
_provider.dart';

import
'package:spotify_clone/services/youtub
e_explode_service.dart';



final youtubeServiceProvider =
Provider((ref) => YouTubeService());

final audioPlayerProvider =
StateNotifierProvider<AudioPlayerNot
ifier, AudioPlayerState>((ref) =>
AudioPlayerNotifier());
```

```
class SearchScreen extends
ConsumerStatefulWidget {

  const SearchScreen({Key? key}) :
super(key: key);



  @override

  ConsumerState<SearchScreen>
createState() => _SearchScreenState();

}



class _SearchScreenState extends
ConsumerState<SearchScreen> {

  final TextEditingController
_searchController =
TextEditingController();

  List<SearchResult> _searchResults =
[];

  bool _isLoading = false;



  Future<void> _performSearch(String
query) async {

    if (query.isEmpty) return;
```

```
      _isLoading = true;
    });

    final youtubeService =
ref.read(youtubeServiceProvider);
    final results = await
youtubeService.searchSongs(query);

    setState(() {
      _searchResults = results;
      _isLoading = false;
    });
  }

  Future<void>
_playSong(SearchResult song) async {
    final youtubeService =
ref.read(youtubeServiceProvider);
    final audioUrl = await
youtubeService.getAudioUrl(song.id);

    if (audioUrl != null) {
      setState(() {
await
ref.read(audioPlayerProvider.notifier).p
laySong(audioUrl, song);
      }
    }
  }

  @override
  Widget build(BuildContext context) {
    final audioState =
ref.watch(audioPlayerProvider);

    return Scaffold(
      backgroundColor: const
Color(0xFF1C1B1B),
      body: SafeArea(
        child: Column(
          children: [
            Padding(
              padding: const
EdgeInsets.all(16.0),
              child: TextField(
```

```
            style: const TextStyle(color:
Colors.white),
            decoration: InputDecoration(
              hintText: 'Search for
songs...',
              hintStyle: const
TextStyle(color: Colors.grey),
              filled: true,
              fillColor:
Colors.grey.withOpacity(0.1),
              border: OutlineInputBorder(
                borderRadius:
BorderRadius.circular(12),
                borderSide:
BorderSide.none,
              ),
              suffixIcon: IconButton(
                icon: const
Icon(Icons.search, color: Colors.grey),
                onPressed: () =>
_performSearch(_searchController.text)
,
              ),
            controller: _searchController,
            ),
            onSubmitted:
_performSearch,
          ),
        ),
        if (_isLoading)
          const
CircularProgressIndicator()
        else
          Expanded(
            child: ListView.builder(
              itemCount:
_searchResults.length,
              itemBuilder: (context,
index) {
                final song =
_searchResults[index];
                final isPlaying =
audioState.currentSong?.id == song.id;

                return ListTile(
```

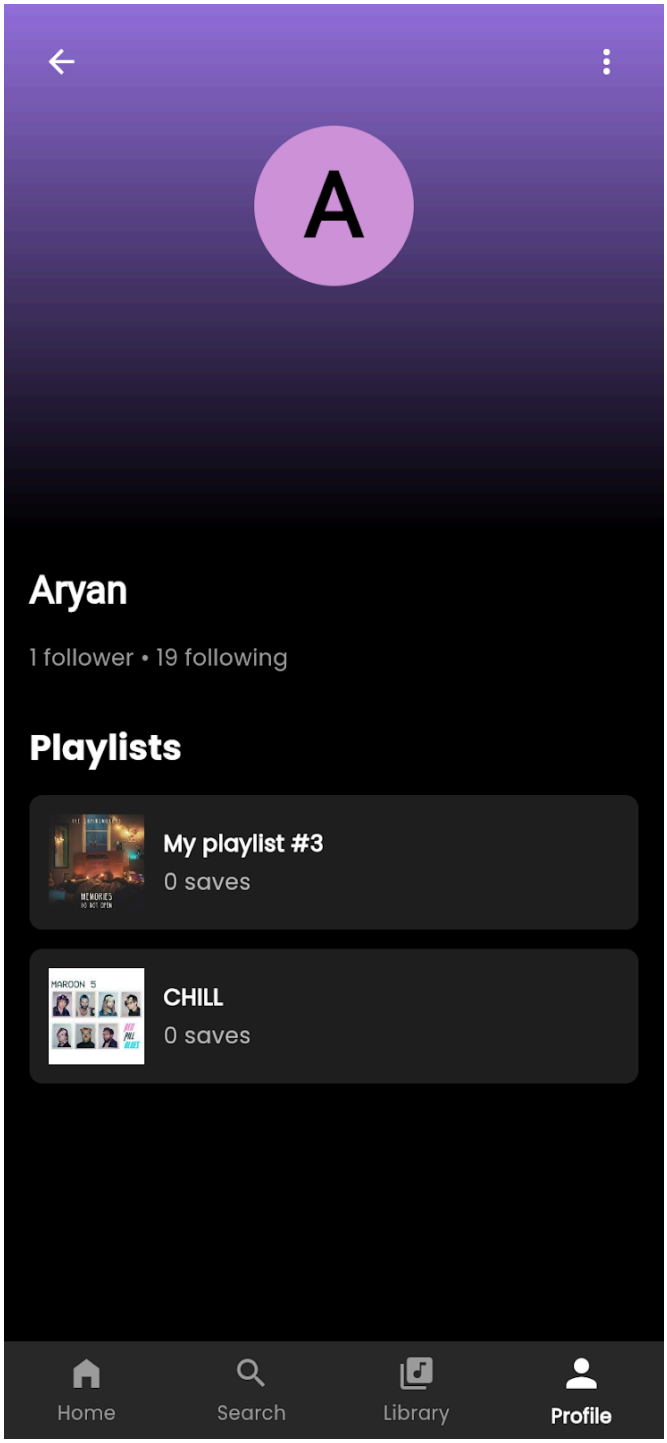
```
                    title: Text(
                  song.title,
                  style: const
TextStyle(color: Colors.white),
                ),
                subtitle: Text(
                  song.channelTitle,
                  style: const
TextStyle(color: Colors.grey),
                ),
                trailing: IconButton(
                  icon: Icon(
                    isPlaying &&
audioState.isPlaying
                        ? Icons.pause
                        : Icons.play_arrow,
                    color: Colors.white,
                  ),
                  onPressed: () {
                    if (isPlaying) {

audioState.currentSong!.thumbnail,
                  leading:
Image.network(song.thumbnail),

ref.read(audioPlayerProvider.notifier).t
ogglePlay();
                    } else {
                      _playSong(song);
                    }
                  },
                ),
              );
            },
          ),
        ),
        if (audioState.currentSong !=
null)
          Container(
            padding: const
EdgeInsets.all(16),
            color: Colors.black,
            child: Row(
              children: [
                Image.network(
```

```
                height: 50,
              ),
              const SizedBox(width: 16),
              Expanded(
                child: Column(
                  crossAxisAlignment:
CrossAxisAlignment.start,
                  children: [
                    Text(

audioState.currentSong!.title,
                      style: const
TextStyle(color: Colors.white),
                      maxLines: 1,
                      overflow:
TextOverflow.ellipsis,
                    ),
                    Text(

audioState.currentSong!.channelTitle,
                      style: const
TextStyle(color: Colors.grey),
                    );
                  ),
                ],
              ),
              IconButton(
                icon: Icon(
                  audioState.isPlaying ?
Icons.pause : Icons.play_arrow,
                  color: Colors.white,
                ),
                onPressed: () {

ref.read(audioPlayerProvider.notifier).t
ogglePlay();
                },
              ),
            ],
          ),
        ),
      ),
    ),
```

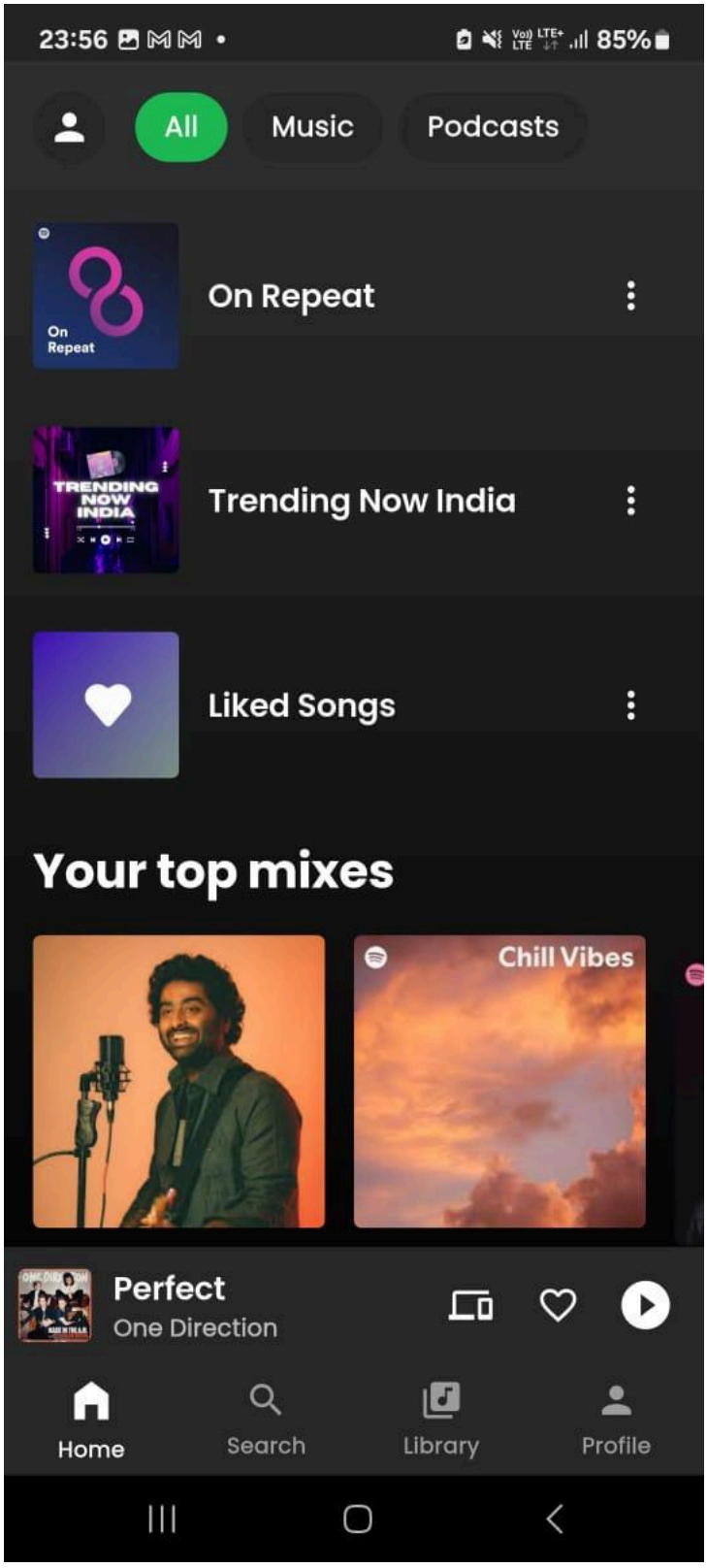
```
}
```

OUTPUT

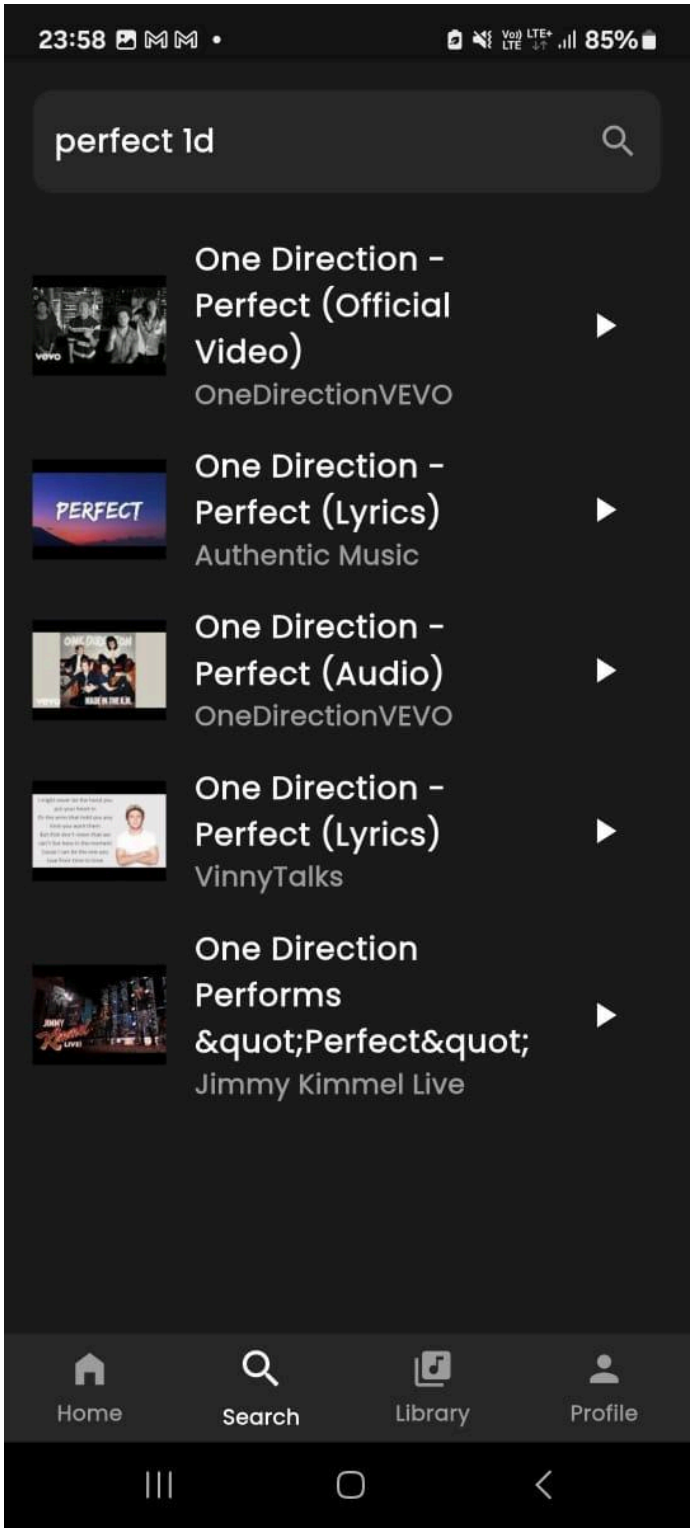
Profile Page



Homepage


23:56
85%
All
Music
Podcasts
On Repeat
Trending Now India
Liked Songs
Your top mixes
Chill Vibes
Perfect
One Direction
Home
Search
Library
Profile

Search Page


23:58
85%
perfect 1d
One Direction - Perfect (Official Video)
OneDirectionVEVO
One Direction - Perfect (Lyrics)
Authentic Music
One Direction - Perfect (Audio)
OneDirectionVEVO
One Direction - Perfect (Lyrics)
VinnyTalks
One Direction Performs "Perfect"
Jimmy Kimmel Live
Home
Search
Library
Profile

MAD & PWA Lab

Journal

Experiment No.	03
Experiment Title.	To include icons, images, fonts in Flutter app
Roll No.	33
Name	Aryan Anil Patankar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

EXPERIMENT NO.3

NAME:ARYAN ANIL PATANKAR
CLASS:D15A
ROLL NO:33

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

Theory: -

Flutter is a versatile open-source UI framework , which allows developers to build natively
compiled applications for mobile, web, and desktop platforms from a single codebase. One of the key strengths of Flutter is its flexibility in creating highly customizable UIs. This practical focuses
on incorporating essential visual elements—icons, images, and custom fonts—into a Flutter application. These elements enhance the visual appeal and usability of the app, providing an engaging experience for users.
A flutter app when built has both assets (resources) and code. Assets are available and deployed
during runtime. The asset is a file that can include static data, configuration files, icons, and
images. The Flutter app supports many image formats, such as JPEG, WebP, PNG, GIF, animated
WebP/GIF, BMP, and WBMP.
Visual elements play a significant role in app development.
• **Enhanced User Experience**: Images and icons make your app visually appealing and user
friendly.
• **Information Conveyance**: They convey information quickly and intuitively. A well chosen icon can replace lengthy text.
• **Branding**: Custom icons and images reinforce your app's branding, making it memorable.

➢ **Adding Icons in Flutter**

Flutter provides built-in material design icons through the Icons class. Custom icons can also

be added using third-party packages such as flutter_launcher_icons and font_awesome_flutter.

```
Icon(
  Icons.home,
  size: 40,
);
```

➢ **Adding Images in Flutter**

Flutter supports images from three sources:

1. **Assets (Stored locally in the project)**

• Place the image inside the assets/images folder in the project.
• Declare the image in pubspec.yaml

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

• Display the image in the app

```
Image.asset('assets/images/sample.png');
```

2. **Network (Fetched from the internet)**

Displaying images from the internet or network is very simple. Flutter provides a built-in method Image.network to work with images from a URL. The Image.network method also
allows you to use some optional properties, such as height, width, color, fit, and many more.

```
Image.network('https://example.com/sample.jpg');
```

3. **Memory or File (Stored on the device)**

➢ Adding Custom Fonts in Flutter

By default, Flutter uses the Roboto font, but custom fonts can be added for a unique UI.

• Download the font and place it in the assets/fonts/ folder.
• Declare the font in pubspec.yaml
• Use the font in the app

```
Text(
  'Custom Font Example',
  style: TextStyle(fontFamily: 'CustomFont', fontSize: 24),
);
```

CODE:

Homepage.dart

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

// First, add this typography class at the
top of your file
class AppTypography {
  static final TextStyle displayLarge =
GoogleFonts.poppins(
    fontSize: 24,
    fontWeight: FontWeight.bold,
    letterSpacing: -0.3,
    color: Colors.white,
  );

  static final TextStyle titleMedium =
GoogleFonts.poppins(
    fontSize: 16,
    fontWeight: FontWeight.w600,
    letterSpacing: -0.2,
    color: Colors.white,
  );

  static final TextStyle bodySmall =
GoogleFonts.poppins(
    fontSize: 12,
class HomeScreen extends
StatelessWidget {
  const HomeScreen({Key? key}) :
super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Stack(
        children: [
          Container(
            decoration: BoxDecoration(
              gradient: LinearGradient(
                begin: Alignment.topCenter,
                end: Alignment.bottomCenter,
                colors: [
                  Colors.black.withOpacity(0.8),
                  Colors.black,
                ],
              ),
            ),
          ),
          SingleChildScrollView(
            child: SafeArea(
              child: Column(
                crossAxisAlignment:
CrossAxisAlignment.start,
```

```
    fontWeight: FontWeight.normal,
    letterSpacing: 0.1,
    color: Colors.grey[400],
  );
}

          children: [
                    // Updated CircleAvatar
with specific size and styling
                    const CircleAvatar(
                      radius: 20,
                      backgroundColor:
Color(0xFF282828),
                      child: Icon(
                        Icons.person,

color: Colors.white,
                        size: 24,
                      ),
                    ),
                    const SizedBox(width: 16),
                    Container(
                      padding: const
EdgeInsets.symmetric(
                          horizontal: 16, vertical:
8),
                      decoration:
BoxDecoration(
                        color: const
Color(0xFF1DB954),
                        borderRadius:
BorderRadius.circular(20),
                      ),
                      child: const Text('All',
                          style: TextStyle(color:
Colors.white)),
                    ),
                    const SizedBox(width: 8),
                    Container(
                      padding: const
EdgeInsets.symmetric(
                          horizontal: 16, vertical:
                children: [
                  Padding(
                    padding: const
EdgeInsets.all(16.0),
                    child: Row(

  borderRadius: BorderRadius.circular(20),
                      ),
                      child: const Text('Music',
                          style: TextStyle(color:
Colors.white)),
                    ),
                    const SizedBox(width: 8),
                    Container(
                      padding: const
EdgeInsets.symmetric(
                          horizontal: 16, vertical:
8),
                      decoration:
BoxDecoration(
                        color: const
Color(0xFF282828),
                        borderRadius:
BorderRadius.circular(20),
                      ),
                      child: const
Text('Podcasts',
                          style: TextStyle(color:
Colors.white)),
                    ),
                  ],
                ),
              ),
              const PlaylistSection(
                title: 'On Repeat',
                image:
'assets/images/on_repeat.jpg',
                isLarge: true,
              ),
              const PlaylistSection(
                title: 'Trending Now India',
                image:
```

```
8),
                    decoration:
BoxDecoration(
                     color: const
Color(0xFF282828),

                image:
'assets/images/liked_songs.jpg',
                isLarge: true,
              ),
              const SectionTitle(title: 'Your
top mixes'),
              SingleChildScrollView(
                scrollDirection:
Axis.horizontal,
                padding: const
EdgeInsets.symmetric(horizontal: 16),
                child: Row(
                  children: const [
                    MixCard(
                      title: 'Arijit Singh Mix',
                      subtitle: 'Pritam,
Vishal-Shekhar and Atif Aslam',
                      image:
'assets/images/arijit.jpg',
                    ),
                    SizedBox(width: 16),
                    MixCard(
                      title: 'Chill Mix',
                      subtitle: 'Kendrick Lamar,
The Weeknd, Future',
                      image:
'assets/images/chill.jpg',
                    ),
                    SizedBox(width: 16),
                    MixCard(
                      title: 'R&B Mix',
                      subtitle: 'Kali Uchis,
SZA',
                      image:
'assets/images/rnb.jpg',
                    ),
'assets/images/trending_india.jpg',
                      isLarge: true,
                    ),
                    const PlaylistSection(
                      title: 'Liked Songs',
                    ),
                  ),
                ),
                const Positioned(
                  left: 0,
                  right: 0,
                  bottom: 0,
                  child: NowPlayingBar(),
                ),
              ],
            ),
          );
        }
      }

class PlaylistSection extends
StatelessWidget {
  final String title;
  final String image;
  final bool isLarge;

  const PlaylistSection({
    Key? key,
    required this.title,
    required this.image,
    this.isLarge = false,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(16.0),
      child: Row(
        children: [
          ClipRRect(
            borderRadius:
BorderRadius.circular(4),
```

```
                ],
              ),
            ),
            const SizedBox(height: 100),
          ],
            ),
        ),
        const SizedBox(width: 16),
        Expanded(
          child: Text(
            title,
            style:
AppTypography.titleMedium, // Updated
typography
            overflow: TextOverflow.ellipsis,
// Added overflow handling
            maxLines: 2,
          ),
        ),
        Material(
          color: Colors.transparent,
          child: IconButton(
            icon: const Icon(
              Icons.more_vert,
              color: Colors.white,
              size: 24,
            ),
            onPressed: () {},
          ),
        ),
      ],
    ),
  );
 }
}

class MixCard extends StatelessWidget {
  final String title;
  final String subtitle;
  final String image;

  const MixCard({
            child: Image.asset(
              image,
              width: isLarge ? 80 : 60,
              height: isLarge ? 80 : 60,
              fit: BoxFit.cover,
@override
  Widget build(BuildContext context) {
    return SizedBox(
      width: 160,
      child: Column(
        crossAxisAlignment:
CrossAxisAlignment.start,
        children: [
          ClipRRect(
            borderRadius:
BorderRadius.circular(4),
            child: Image.asset(
              image,
              width: 160,
              height: 160,
              fit: BoxFit.cover,
            ),
          ),
          const SizedBox(height: 8),
          Text(
            title,
            style:
AppTypography.titleMedium, // Updated
typography
            overflow: TextOverflow.ellipsis, //
Added overflow handling
            maxLines: 1,
          ),
          const SizedBox(height: 4),
          Text(
            subtitle,
            style: AppTypography.bodySmall,
// Updated typography
            overflow: TextOverflow.ellipsis, //
Added overflow handling
            maxLines: 2,
          ),
```

```
    Key? key,
    required this.title,
    required this.subtitle,
    required this.image,
  }) : super(key: key);

  }

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

  @override
  Widget build(BuildContext context) {
    return Container(
      color: const Color(0xFF282828),
      padding: const EdgeInsets.all(8),
      child: Row(
        children: [
          ClipRRect(
            borderRadius:
BorderRadius.circular(4),
            child: Image.asset(
              'assets/images/perfect_1d.jpg',
              width: 40,
              height: 40,
              fit: BoxFit.cover,
            ),
          ),
          const SizedBox(width: 12),
          Expanded(
            child: Column(
              crossAxisAlignment:
CrossAxisAlignment.start,
              mainAxisSize:
MainAxisSize.min,
              children: [
                Text(
                  'Perfect',
                  style:
AppTypography.titleMedium, // Updated
          ],
        ),
      );
    }

      'One Direction',
                style:
AppTypography.bodySmall, // Updated
typography
                overflow:
TextOverflow.ellipsis,
              ),
            ],
          ),
        ),
        Material(
          color: Colors.transparent,
          child: IconButton(
            icon: const Icon(
              Icons.devices_outlined,
              color: Colors.white,
              size: 24,
            ),
            onPressed: () {},
          ),
        ),
        Material(
          color: Colors.transparent,
          child: IconButton(
            icon: const Icon(
              Icons.favorite_border,
              color: Colors.white,
              size: 24,
            ),
            onPressed: () {},
          ),
        ),
        Material(
          color: Colors.transparent,
          child: IconButton(
            iconSize: 32,
            icon: const Icon(
```

```
typography
                overflow:
TextOverflow.ellipsis,
              ),
              Text(
                  ),
        ],
      ),
    );
  }
}

class SectionTitle extends StatelessWidget
{
  final String title;

  const SectionTitle({Key? key, required
this.title}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(16.0),
      child: Text(
        title,
        style: AppTypography.displayLarge,
// Updated typography
      ),
    );
  }
}
```

```
              Icons.play_circle_filled,
              color: Colors.white,
            ),
            onPressed: () {},
          ),
```

Register.dart

```
import 'package:flutter/material.dart';
import
'package:spotify_clone/features/auth/scree
ns/sign_in_screen.dart';
import 'package:spotify_clone/main.dart';

class Register extends StatefulWidget {
  const Register({Key? key}) : super(key:
key);

  @override
  State<Register> createState() =>
_RegisterState();
}

class _RegisterState extends
State<Register> {
  bool _obscurePassword = true;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const
Color(0xFF1C1B1B),
      body: SafeArea(
        child: SingleChildScrollView(
          child: Padding(
            padding: const
EdgeInsets.all(24.0),
            child: Column(
              crossAxisAlignment:
CrossAxisAlignment.start,
              children: [
                Row(
                  children: [
```

```
Navigator.pop(context),
                ),
                const SizedBox(width: 10),
                Image.asset(
'assets/images/spotify_logo.png',
                  height: 150,
                ),
              ],
            ),
            const SizedBox(height: 32),
            const Text(
              'Register',
              style: TextStyle(
                color: Colors.white,
                fontSize: 28,
                fontWeight: FontWeight.bold,
              ),
            ),
            const SizedBox(height: 8),
            Row(
              children: [
                const Text(
                  'If You Need Any Support ',
                  style: TextStyle(color:
Colors.grey),
                ),
                TextButton(
                  onPressed: () {},
                  child: const Text(
                    'Click Here',
                    style: TextStyle(color:
Colors.green),
                  ),
                ),
              ],
            ),
                  IconButton(
                    icon: const
Icon(Icons.arrow_back, color:
Colors.white),
                    onPressed: () =>
hintText: 'Full Name',
                  hintStyle: const
TextStyle(color: Colors.grey),
                  filled: true,
                  fillColor:
Colors.grey.withOpacity(0.1),
                  border: OutlineInputBorder(
                    borderRadius:
BorderRadius.circular(12),
                    borderSide:
BorderSide.none,
                  ),
                ),
              ),
              const SizedBox(height: 16),
              TextField(
                style: const TextStyle(color:
Colors.white),
                decoration: InputDecoration(
                  hintText: 'Enter Email',
                  hintStyle: const
TextStyle(color: Colors.grey),
                  filled: true,
                  fillColor:
Colors.grey.withOpacity(0.1),
                  border: OutlineInputBorder(
                    borderRadius:
BorderRadius.circular(12),
                    borderSide:
BorderSide.none,
                  ),
                ),
              ),
              const SizedBox(height: 16),
              TextField(
                obscureText:
_obscurePassword,
```

```
            const SizedBox(height: 32),
            TextField(
              style: const TextStyle(color:
Colors.white),
              decoration: InputDecoration(
TextStyle(color: Colors.grey),
                filled: true,
                fillColor:
Colors.grey.withOpacity(0.1),
                border: OutlineInputBorder(
                  borderRadius:
BorderRadius.circular(12),
                  borderSide:
BorderSide.none,
                ),
                suffixIcon: IconButton(
                  icon: Icon(
                    _obscurePassword
                        ? Icons.visibility_off
                        : Icons.visibility,
                    color: Colors.grey,
                  ),
                  onPressed: () {
                    setState(() {
                      _obscurePassword =
!_obscurePassword;
                    });
                  },
                ),
              ),
            ),
            const SizedBox(height: 32),
            SizedBox(
              width: double.infinity,
              height: 56,
              child: ElevatedButton(
                onPressed: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (context) =>
const MainScreen()),
                style: const TextStyle(color:
Colors.white),
                decoration: InputDecoration(
                  hintText: 'Password',
                  hintStyle: const
Color(0xFF42C83C),
                  shape:
RoundedRectangleBorder(
                    borderRadius:
BorderRadius.circular(30),
                  ),
                ),
                child: const Text(
                  'Create Account',
                  style: TextStyle(
                    fontSize: 16,
                    fontWeight:
FontWeight.bold,
                  ),
                ),
              ),
            ),
            const SizedBox(height: 32),
            Row(
              mainAxisAlignment:
MainAxisAlignment.center,
              children: [

Image.asset('assets/images/google_logo.pn
g', height: 40),
                const SizedBox(width: 20),

Image.asset('assets/images/apple_logo.png
', height: 40),
              ],
            ),
            const SizedBox(height: 32),
            Row(
              mainAxisAlignment:
MainAxisAlignment.center,
              children: [
                const Text(
```

```
                  );
                },
                style:
ElevatedButton.styleFrom(
                  backgroundColor: const
TextButton(
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                          builder: (context) =>
const SignInScreen()),
                    );
                  },
                  child: const Text(
                    'Sign In',
                    style: TextStyle(color:
Colors.white),
                  ),
                ),
              ],
            ),
          ],
        ),
      ),
    ),
  );
 }
}
```

```
                  'Do You Have An Account?
',
                  style: TextStyle(color:
Colors.grey),
                ),
```

Sign_in.dart

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

import 'register.dart';

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

 @override
 State<SignInScreen> createState() =>
_SignInScreenState();
}

class _SignInScreenState extends
State<SignInScreen> {
 bool _obscurePassword = true;

 @override
 Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: const
Color(0xFF1C1B1B),
    body: SafeArea(
      child: SingleChildScrollView(
        child: Padding(
          padding: const
EdgeInsets.all(24.0),
          child: Column(
            crossAxisAlignment:
CrossAxisAlignment.start,
            children: [
              Row(
                children: [
```

```
Navigator.pop(context),
                ),
                const SizedBox(width: 12),
                Image.asset(

'assets/images/spotify_logo.png',
                  height: 150,
                ),
              ],
            ),
            const SizedBox(height: 32),
            const Text(
              'Sign In',
              style: TextStyle(
                color: Colors.white,
                fontSize: 28,
                fontWeight: FontWeight.bold,
              ),
            ),
            const SizedBox(height: 8),
            Row(
              children: [
                const Text(
                  'If You Need Any Support ',
                  style: TextStyle(color:
Colors.grey),
                ),
                TextButton(
                  onPressed: () {},
                  child: const Text(
                    'Click Here',
                    style: TextStyle(color:
Colors.green),
                  ),
                ),
              ],
            ),
                  IconButton(
                    icon: const
Icon(Icons.arrow_back, color:
Colors.white),
                    onPressed: () =>
                  hintText: 'Enter Username Or
Email',
                  hintStyle: const
TextStyle(color: Colors.grey),
                  filled: true,
                  fillColor:
Colors.grey.withOpacity(0.1),
                  border: OutlineInputBorder(
                    borderRadius:
BorderRadius.circular(12),
                    borderSide:
BorderSide.none,
                  ),
                ),
              ),
              const SizedBox(height: 16),
              TextField(
                obscureText:
_obscurePassword,
                style: const TextStyle(color:
Colors.white),
                decoration: InputDecoration(
                  hintText: 'Password',
                  hintStyle: const
TextStyle(color: Colors.grey),
                  filled: true,
                  fillColor:
Colors.grey.withOpacity(0.1),
                  border: OutlineInputBorder(
                    borderRadius:
BorderRadius.circular(12),
                    borderSide:
BorderSide.none,
                  ),
                  suffixIcon: IconButton(
                    icon: Icon(
                      _obscurePassword
```

```
            const SizedBox(height: 32),
            TextField(
              style: const TextStyle(color:
Colors.white),
              decoration: InputDecoration(
 setState(() {
                    _obscurePassword =
!_obscurePassword;
                  });
                },
              ),
            ),
            ),
            const SizedBox(height: 16),
            Align(
              alignment:
Alignment.centerRight,
              child: TextButton(
                onPressed: () {},
                child: const Text(
                  'Recovery Password',
                  style: TextStyle(color:
Colors.grey),
                ),
              ),
            ),
            const SizedBox(height: 32),
            SizedBox(
              width: double.infinity,
              height: 56,
              child: ElevatedButton(
                onPressed: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (context) =>
const MainScreen()),
                  );
                },
                style:
ElevatedButton.styleFrom(
                  backgroundColor: const
                    ? Icons.visibility_off
                    : Icons.visibility,
                  color: Colors.grey,
                ),
                onPressed: () {
                          ),
              ),
              child: const Text(
                'Sign In',
                style: TextStyle(
                  fontSize: 16,
                  fontWeight:
FontWeight.bold,
                ),
              ),
            ),
          ),
          const SizedBox(height: 16),
          const Center(
            child: Text(
              'Or',
              style: TextStyle(color:
Colors.grey),
            ),
          ),
          const SizedBox(height: 16),
          Row(
            mainAxisAlignment:
MainAxisAlignment.center,
            children: [

Image.asset('assets/images/google_logo.pn
g', height: 40),
              const SizedBox(width: 20),

Image.asset('assets/images/apple_logo.png
', height: 40),
            ],
          ),
          const SizedBox(height: 32),
          Row(
            mainAxisAlignment:
```

```
Color(0xFF42C83C),
                  shape:
RoundedRectangleBorder(
                    borderRadius:
BorderRadius.circular(30),
Colors.grey),
                ),
                TextButton(
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                        builder: (context) =>
const Register()),
                      );
                  },
                  child: const Text(
                    'Register Now',
                    style: TextStyle(color:
Colors.blue),
                  ),
                ),
              ],
            ),
          ],
        ),
      ),
    ),
  );
 }
}
```

Profile_screen.dart

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

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

 @override
 Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.black,
    body: Column(
      children: [
       const Expanded(flex: 2, child:
_TopPortion()),
        Expanded(
          flex: 3,
          child: Padding(
            padding: const
EdgeInsets.all(16.0),
            child: Column(
              crossAxisAlignment:
CrossAxisAlignment.start,
              children: [
                Text(
                  "Aryan",
                  style:
Theme.of(context).textTheme.headlineSm
all?.copyWith(
                      color: Colors.white,
                      fontWeight:
FontWeight.bold,
                    ),
                ),
                const SizedBox(height: 8),
                const _ProfileInfoRow(),
MainAxisAlignment.center,
                children: [
                  const Text(
                    'Not A Member? ',
                    style: TextStyle(color:
```

```
copyWith(
                    color: Colors.white,
                    fontWeight:
FontWeight.bold,
                  ),
              ),
              const SizedBox(height: 16),
              const _PlaylistCard(
                title: "My playlist #3",
                saves: "0 saves",
                imageUrl:

"
gABAQAAAQABAAD/2wCEAAkGBxI
TEhUSEhMVFRUVGBcXFRUYFxUWF
RUVFxcXFhUVFRUYHSggGBolHRUW
ITEhJSkrLi4uFx8zODMtNygtLisBCgoK
Dg0OGhAQGi0lHyUtLS0tKy0tLS0tLS0t
LS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS
0tLS0tLS0tLS0tLf/AABEIAOEA4QMBI
gACEQEDEQH/xAAbAAABBQEBAAA
AAAAAAAAAAAAAAgMEBQYBB//E
AD8QAAEEAAQDBQUGBQMDBQAA
AAEAAgMRBBIhMQVBURMiYXGRBj
KBobEUI3LB0fBCUmKS4RWy8Qcz0iR
TVIKT/8QAGgEAAwEBAQEAAAAAA
AAAAAAAAAECAwQFBv/EACwRAAI
CAQMDAwMDBQAAAAAAAAABAhE
DEhMhBDFBIjJRYXGRFNHwBSNCge
H/2gAMAwEAAhEDEQA/APLEIQmmE
oLgCUEAASggBKAUjOAJYCKV/wD6b
C7E4eJoexkrIHOt4eQOQUBmrY
7fBJsdFGAnYm6jzCvcJwmAYmSOR2a
KOHtM1vAJ7Fkl21hdltx0y3VaKoaBm02
vTnpemqTZSRpXw/dO+H5qJhwKWighb
2Er3AmixoANauDzZNHbLt4qBwbh7Xw
Pkc4gsBofzfdvf06tG5GiXTS7i6xJtEI11T
```

```
                const SizedBox(height: 24),
                Text(
                  "Playlists",
                  style:
Theme.of(context).textTheme.titleLarge?.
QALuGb3lf4ngLGYpkBc4t7Nr5DfetsZfK
1vTVjwLutN13AcFZJiYowSxk0bXizmM
ZdG40XUMwDm+Gnqs80vSbYV6iJgOH
Z3KXj+BFnLdNcFha7GGMl7Iw6UkFxD
gI43vAe5gvdosAdVt4HsnLmn+EaDNl1L
2taM2UnUO0010XnZHJPg74c+ODzDE4
eioro1rRgoXTPa4ksN9lbuzzuzANDnZX
ZbbZ1FXV0Fn5cOWkggggkEHcEGiCto
T4JlDkrnMSCxT3RJpzFopGTiQ8i4WKU
WJJYqsVEbKklqlZEgsRYaSPlXE/kXUC
ogLqAugahaEHAV0OT5i1pOPj01/wAKd
Q9LIof4JbZfBLyJMrdQjgQ+ArOHiRE0
M2XWJsTQ2/e7JjWA3Wl5bUANTjQob
NUiyg4kO1dI+PMHxCFzQ/KaETYrDi00
e5ex3UKNiA1OxhS2UkbrDytEbw5mYE
sNZsurc1Wa27xsCvMKk4ZxAxxOZlvO
N7qrjcy6/wDtfwV29v3bh+H81lI3aDyT6
PmzLrXyv58FoMeez7IAVRF2dzJHJm9I
mtXIeIEOhdlswgCr96pHv16e/XwVadOv
LXVEY1PLb1XfSPPbZZ4/G9pK18jQ7K
1jHCyMwY3Ld8id+evVNN4qWTGZjQD
lDWi7yhrWsZZrUgMHS6Tco7x81ELLU
JKkU29T+5Mk4w98zcQWglt5gLpwcX5
gTysPI8FY8NlzPZI1tNEeRrScxytaY9XU
LJsnYbrPxNoLTezEVxxE9Hf7iuTrHpgq
OzpFcnZBe4Nxj5iyy6w8B1Zi+IxvcDRon
MXbHUqbg84zuiHeIAGuwbIyQa8z3KB
8UnGRjtn+f5BaHguHC83LlaVnpwgkikn
wzcxmMWZji45A4tLO9dXWlVWo2vzV
HioXPc95Gr3OcfNxJP1Xrc3Bc7Gua4Cjb
xQOYUdPpr4LPcSwEYOUAAnl18lnHPJ
Va7gtM+x5y+BRzAtZxDhBAsBVjIeVc7
XVHLaIeLko3wrhgWhdhLsgb0ky4XfTd
Xui2TOmFNOjV79jJ5JmXAFUsiJeJlLkX
VYfZChXrRG2zNBKA1C4E5G3vN810
M5USnx18QCnYSW94GiNQVIkgOl9KT
rcLbT5LBy4OjQ74IrJGuPeHxaANfLZRc
```

UoFbq2Zw8HCvnLjmbIGgaVlptmt7t7P2
UYTh7HS4dry7LKAX1VjvyN7ummjAuz
UcGlfJnXBMPC0fEOGRxYoRSOOSgX
GyO8Y73ANNL+YBIB5pjB8EEmIlhNtp
kjmAODu8G5ohnoBzTbdaFg8lKlwXXJ
+BXb/d+V+H+5yVi+H/P9GgfwqcuPcPh
3mfqmzwTECz2Z/ub+qgTvIe4WfePPxS
HP8T6pR3aXK/D/cJbep9yazguIG8Z9W/
qtFwDCuY1jXaENOmmneJ5LH9oSPePr
8lr/Z5/cj/AfP3iuPrVPQtVHV0mjU9NkHi
Dqmf5q04bjKVDxST7+T8S7BOuKcLR6
UXwb2DiWlWkyxiTRwBHis5w/EarTYN
2i4pw0sbVD0XDbFbjx39eazp4UwYjIQc
utkgtAdVgAnRw8ltsHiGtFEE3zAJr02SuJ
RMlp1HbauWu4K1gqhqT5+DDdalT7GO
x8TI3MZQAeaBN9FnpsQ7tDG+MNIcRd
+h15LZ47hUbiC5oFEEEEij4C1l4uGySSj
O1wBJBe7MdKqzXwKvHJeTpx0xzA4M
a5volzYJpB0rzB19AtWzCxta0NYG0AD
zsj+LXmo2JKjd54JU7Mh/pg8PR36IWl7
VCrcZZ4S1WnAeGnEYiOEPYzOT33m
miml2p+FeZCq2q24HMGTxPJoBwJO9C
9TS9mb4dHkwVtI0D8IctHXKacRqLHirf
h3BiWBxaaeHAGuY2Ws4LG2aKaPK50
Bc6TKxoY7PpWpI00vr9E/wiWaKFseIjD
o33kDstOcNQTVkaAei8yWRtHoN03xye
NiGpCmOLN1Z5rT+1s+FdK37NG6PKM
smY3meDqdys1xg+55/ou3G7o86fuHHB
KaEFdambnU7BuPMfVNp2AajzCTGjaP
FNk/CVjY1s5vdf+ErFQizWnxV9F2Zz9Z
3Qp6Zz805JGd77tkB3iPBNsY6QFo2bZJ
6DQE/Rdjao5VElYl3fcfFIkk08+aRiXd8+
ZTLpU49kKS9TFNk0W19nndyP8B/3FY
JrluvZn/txfgP+4rj672L7nX0i9T+xTcVd9/
J+IpuN67xU/fyfiKjh9Lmrg7FKi2wuIpaL
A8T0WPLXNa15HdfeUggg1odjoR0KfgN
OmqxyYbNYzTPQMPxWtnEeRSMZxd4
ohxN206WRp3XDyKyrJa2eDQ+gJ/LdS8
Lc4LNO6Q7V2UVqD+Sw2q+w0o3yOS4
HFySRmSQERkOoNOpB6b6ilohNW+ios
U+YsDI3xtlDW6iS2911HvEcxSjfaZmN+
+cHOJJFODu7TRy21tKUHIruad+I06qn4
dHRZqDfTl4FPYZneISOIN1b5px7nO3y
O5UtoSsqU0Is6DtJ2MajzCRSdhGo8x9V
LGjeMZbH9corzWXh4VNXufNv6rTuPc
kH9H6rCRtPIn1S6TVzpMuq08ai3fwmY
7MPq39UmLhM+vc3/qb+qr9ep9Sm6JG5
Yakz0VJNJlFj5Lkef63fVQuKOsM80/ij33
fiP1UXiOzfNdsF2PNk/UTRsE4wJHIJ1q
TOlAnIdx5hIXWnUealjNnM7R3kVich3t
v9zf1WuxUnveRWIzLTo06Zl1XdD8WNc
HkOoB13RsAHevgkQ4ssPd0zEDL/ADN
Pj00TD0y8c6Om589qHwW7jXBkn5LB7
DZsi7N25t/Mpt0JP8v97f1TWId3imlaboh
pWPiB39P97P1W79mGERx3uGnY3/Eei
89tb32Vd91H+E/7iuXrL0I6Om9z+xTcX
NTy/iKrzIda+Cncb1xEn4vyCrZ2ULWUF
wjWRacXxn/pogAwGNmXtC1zSDmDjfe
IfevLTNpSrOEYt7sweQaqiKr1Cqs4LySb
FdA6tWjQHTmrDhcgLjR0yt5AVRIogaX
+q2lHgxi6dIu8NDC0lwMxdTqvJls3v4ap
ztEhsa6WLFqzdTocM6akxNJt4Ucvogm6
BvTf4KdsreD7e1w0d8DYI9VGlmsaGx4
WjGVldlc4ucW254a0UAdnA6nbYclXRj
KCC7NrY0NDrv8ADlyWigiZZCTnQo2d
CrSRqK1qXXeb5pLE4G25g6kfVbeTA0n
DSC011NqU094eaThcOGMAHx81wuog
+K5cnLZ04+EinxQp7vxH6qLj/wCHzUrF
++78RUXiH8HmtYeDkfuJ5S2FITMkhB0
cR8ApOolFAOo81BMh/mKSXHqfUood
G7xkZo+R+iwbX2nHTPO7nH4lMdoAq
w3AWWOsWSlPcTfj+WyQHjqjtB1Wu59
DLZXyLmOpTaO2b1XWytOyW4ytqPyC
03BOPwxMY1+e2gg00EWSTpr4rN2EE
LPI9aplwhodotcZxBj5HvF05xIsa14qPLi
QdvmLHxCiNCUAoSSNKsZ7F2YuBbre
mXSibqvgpOFaWuLtCXdBWvX5pIClR
DbbcKnJme2i54c7Nup7ok1AzorSBljUfF
TjalKmYdWpY4ao+CmmgVbio+XX0+K
002GVDxTCEWV3/pjysX9Q1OjLYvFkP
cBRAJFjW6NWPApeIOUkWCWizRDh/c
N1bYmJroIo3U0sMhDnuDYwHlpvM3v
WLJrXyVRG6Npk7oe0ZdGl2VwDtadlB
DT8Fi4pOjvhlbVkf7QfBcUf4/JCmi9TH

hxABVc2Mste2Rpa1rrYXFjhe7hfvV0Cp
cXjSVUMAcItv9T8ULM/akLfZFuoyTV
Mwb6kZz7w8t1Dal3qF6TVnlp0ewcH4o9
gDXO7zqEhadCee3KqTPEOKuM0YzHK
ywBZoXzpZrh82VjdbO58D0ThnJeD4rzJ
DBtXySmwt3Pz0/4VaydgqB8a8N1zs75F
XZjArTc0ukfojcK/T/ACynDQOVHyUuK
yNR+/BSsw/dLjgFLkVHHXkRkRlT7ma
nzK5lWeo10DIapUQ28x9UhrU8zcX1H1
RZLgXcT1asxoa0czmbofxC1TtIUXiUtM
G/vDbfYqcfuRl1eLVjkvobSIt30N7eqg+0
DgIHuG4GmwN7b8lhXYo9X+v+VGnxB
OhL/LNp9V6u+q7Hy+P+lyU1Jy8/A7x02
GuF6sbm5jNlBOo8vkVn7U15/EmXAeK
wcrPZjDSqGq8vUIS6QkPQhTE4PfZ+If
UJEaW9p0I3BtIuuDbRu0TE6z44rP8Ayt9
P8oPEpz/AP38VlKFsqDpUyRjW94lQMd
/B5hLfi5juwJp4keW22qITiqJauVlio726lS
Qkhmqzbo6oxtjbWJwMKmxwp3sdFm5n
YsPBAibrrtqnnt/L9/NLazwXToixqFDLIj
d6cvl4fFLbAb3+PMDoEpxIHLw86tPFrht
rfX9AP0TbGooafhjyrQADw1BP0CR9m0
q/lqfMqxDNCTyXAw1qBe3xU62W8aK
mWPwXGRm9uis34YBN5BonqM9rkbd
GLO+56dfNJLB4qZ2Wp8z9UkxLOzbQ
RWsHQ+v+E6IhppzHPx8k4I065mh9eS
WoWgajcmuJvGTfWxp6qPHOo0shIN3u
P3a0guTlytaWhhxTLilOemXOXWjz2hL
kzmtOEpslUSJXVy0IAVGpUa7h2h8fi38k
mIqbsvTVEhqdamWp1pUstDzGnoV2im
MW8gMAcRodiRzK4O1od93qpoLJAT8
TASm5C4DUn5KKJiHaLOSs3xySfJo2Y
dE0ahwcWAGoNpD+Jk/wj1XPplZ6W5C
iQ9opMiO/ySmSF9KygwppVdE8Mhx4fr
r5p/sxz/MfRPuipR3kD3iAPFK7K4SFsDf
E+p+qe03oqtjx2ZwbE3MTz2Hn1rQrbcB
4E2VjjI8A1dA5R10O9/opn6e5DyxSsyG
KxLW7gqukxjeR581ZcZ4VT3BrrAJAN7
0VmZNCtscYyRjlyTiaPC94X+ikDDlUfC
MXkcNdD12C2kWHcRdt/tj/APFZZfSzo
wzUo2VBwx8fmkywnKd/mrPEYY88vo
wfQKo4gzI1zhVt1G24Wads1lwim7F4Flr
WJ6P32fiH1CaYnW++z8Q+oQV4No1uij
OZbgpjNlFl0Nrll3N8b4KTFjvu/EVFx49z
zUzFjvu8yomM/g/EFrDwc79xNKiTjVS1
FlHeSR1UJC7SW1qV2aLNFEbpQZG3rv
+9FZ5ENgHRClQpY2yqbH4Wu9lR/Qq2
K0x49NvTdasFt6g2Nbv0pVfE8Y6b/ALj
KdZLXAmqIFto8rF+FldWNT8mWTF8FI
4JBVvw7hsckgZJKYw4OObLmAygk3qK
FA6qNxLAxxn7qUSt6hrmn0OhHja1OW
cXF0yvXVxdQQTuGwEZvIj4pmHp0T8
GOYL10KMU0WHjZyzTd8m7S0qvB1p
TgKZaUvMmI7i3aj8I/X809E41uUjCxsfO
BK/JH3Q9w3Arkm+KyxiRwgc4xg9y9yP
FLu6FdKx+SYnTUpH2Z4IL2PDTsfdHq
RSe4DxdrDlmAc3lQANmxq6tvMH5LU8
Y45hpYGxsgMZPvFryQ70ABB8f8rKTc
XVG8VFq7MdJI3Zua/Fwr5BID38r9N1M
M8LfdjvxdWnxOy7Lxwn3b8acdfCzqR5
BPnwgtLuybw2Oc1lieddTlcd9tANF6JJD
hzA1sTvvRo4EVlo63dHrpa8oZxbEPcXN
LgT/IXBo5bE0Oim4Z51zuDnc7JIN8qsZt
tb0891lkwtmsct1VltjsYWPLcwkI2YxpOb
Tm7cVppV+SqHDESOLSyWjyyO+GzV
YYTHj3HEtvRvZhrQD5VShaAkl8vaA6
myNOg72muv5JxVeCpTb8ljgYjhyS9jm
WTlLmkbUQKO/P0ScX7QTA5Wksq7A
OvqNxRSv9WY8NbK1xA0zl7i8dCTWov
zrkqzjkAzZoszg0d8ktNHlqK0rwpSo2/Ui
pZfRUSdBiJSwucddmCrJvckb9FRYwkH
vCj+/RRBiXjQOPly9E0+ck66rohjpnNkz
qUUixwjqO2YXt1C1M3tMIGhkMoe0bB
0ZLmg6gF11pdc9lhhM46D0CWcO4HvG
jV19AlLEpP1CjnaVI0x9s5ibdkI5jIASOm
bcefyKZxfH7NgOO9BxFDTQ0BrWh+H
TRZuRhHiOqR2yNiHhB+pn2bNFhuFdr
A6YSW9tnIN8oBJ035fNan2JmhfI1kgHd
AY1zRZ5Eg6kHnrX6LzvDYksdmaa/Md
COa2nBfaoQtikY1xe1zhIM9MeDWSubC
KO4N9Vllxyr5LhOMl9ST7XYFkcrg0Z
Wh1AOpp0vWv4dK3pZuWqIsjQ/vVXnE
vaUzOcXhr72J7rh1DSaOipWVNJkDRZt
ziHk0BuTR8VeKUox5R0LJ/j3KyF+SQP

h4kEfUJ7DcPMjXHM1rW6lzyQ0eFgEk+
ABVjxPjbuyY0PLXFv3jP4bI7wrpd6G91
zhj3WHsc0O1DW5byu3NnRrL2vXfqt05
VdHmyavSVx9nZXNLm5SbIyElrzXMB
4Ao68702VNicM9hp7S09CKO5H1BHw
4S3oPRGOlGWvHT4Kvje/WjslQanvajV
YrHyDy8UPNmFWmJJSfJEoqx0SmQl1
WQPNXSRCuXCEiStgPPmkF6VPHlNW
HDqP8ppNCdi7TjZiNASPimbQigsde+9y
SpEETRqd+Q3F/mo8I5qUFLKiSGSknQ
8/TyRDJSRAO8gBRRpY+0k/VWreKgBp
c3VoqzRzAdbH71VFNJlHmNR+YUayTb
vRJ41LuUsrj2NJJ7RM2ZCy/wAI/PZNw
SMLmyTGzVZAzu3yDje9dR9FStOU2N
RzbyThkfl7jnFnNt+75jmPFTtJdilmfktDi
QXBuVuW+8K212sedJmWOLvBo2rvne
y4DQcgBfj4qMx5DQTu7U+Q0b+ajvfYI
H8RHy/5QoBLJ8lhgmNsuaNBep6bD4k3
8Ao077IJ52fmlOlphaNt/Tb6fNR3TbX0+t
lUlzZMpKqJuTM089PkqmSMchqnhitCB
z5+m3omrrU0qiqM5STGS0hLimI1HkkPl
vb/ACmldGd12JHbnXx3CdZii0ENJbm3r
S/NQkAo0oam0SBrzUp2MGQAaHZ39X
jqq7MluOlpSjY4za7D32h3Urij5kI0C1k2
OJPjDjoEiMqS1y3SRzyk0NjDDp4JBwI8
QpYclKtKI3GiCMGOYtVtHdXuKd3D+
+arbvfVZT4fBtibkm2Q0BTOxHgPQpuU
AbV8FGo000dbyTxIUZbH/pRBHJjnsmA
cw4eUODgCKJjF0fAlJhZmI5KPl9FybEg
bEWdhepW0PCxBwPEmRoMwxWXMQ
M4bFPFh3U7cDNm9Sj2u4k4YPDQjGv
GfB4UnCdjbXggHtDOdvd2/o8Uq5HqM
OWaWeaC7u30XqMnB4nScFkytymJjZtB
RMOHbiW5xztodv0VVNG2HHcWxDGt
BwrSYBlaWslmcxjJA0irbmcRysoCzDNe
K3SYpbPdI8TuPELbYaU4mfg2IlAdJPJIy
d2Vv3v2fENYxzwBRcWuAJ50E97ZTFu
FeMVPBiJJnB2CMUQa6NjJcs2aVsbBW
UFuU2c1IDUYiXEgj5JqJwJBvQfVb/wB
nnCPF4dwd2Q/0dsjpGtstd2Di6XKPeIIzd
SWqs4fxyP8A1ETTYozgROZHinwZOzk
LD2chhrUMc48jvaB6rMrJLoa56fv0SMZ
zg1y30qiMtixryKsncHa4B2GkbJz7MnJJ5
DMAHfA34FR8ZH2ZLWvscmmnsPgW
mwn+HSQOY4Sh0NkWRFmaCObXWC
3yNjddUMq7pHDlg06/6RfsGK/+LJ/Y/w
D8UJ/tI/8A3Zv/AMj+qFdx+v5Rnu5Plfgr
4R0of7TomXlJzIKlKi27BaH2KkcJMUW
mnfYcXlPQhgIPyWdTsDHk1GHkkGww
OJLa71huuWrvlW6oR6h7YcUixGGxLYi
MpwmFxJA5Sz40TSAjk63Cx/Usp7YYO
Uswc3ZyGL7BhAZQx3Zh1OGUyVlB1G
l81m8JhJZSRDHJIQMxEbHyEN/mIYDQ
8Up2OlLBGZZDGKphe8xgDamE0PRKg
PTWcbYzBSMe4BzeFYWaC9+1kglwr66
mnsFKLxBhlxnGMOwZpJoY3xNGpe6I
QyljeriCaHgV55NBLQc9kmXK0tc5r8vZ
k5Gua4islnKCNL0TjcLiGmN4jnaZDcLw
yRpkJ2MLgLedR7t7hKgNpw7Dvhl4FD
K0skbLLI6Nwp7GzYsGPM06tJay6OqrM
We3wOIZu/A4p0jdNfs+JeWPA8pWtcem
ZZvGduyTNN2zZdHXJ2jZb/hdb+9emh8
FzCwzOEjomTOaG/fGNsjmhl396Wig223
3tO7fJOgN5w8F2Iw7GglzuCBoaAS5zjh
pKDQNyeiyHDODTSYhuFLDHM66ZK
1zHWIzIGlrhYLg2hprYUeKPFBoxDW4
kMaMrZ2iYMY0d3K2YaNA1FA+CXNh
cWckz48STKW9lK5kxMriLj7OQjvuoW
MpJoaJUBa+00To8Fw+KVjmSAYt+RwL
XtjfM3JmadRmLXEX0WZT+NdM52aY
yl5sZpS8uOUlpGZ+pohzfAgjkmE0IEIQ
mAIXEIAELiEACEITAF1cQgBdrhcuISA
EIQgARaEIAF21xCAO2rn2UmcyWSRh
LXx4bFvY4bte3DyZXDxFqlUzhPE5MPI
JYsuYNc3vND209pa62nQ6HmgDY8UH
YumMPcc/ieDzhulHsXz5PACV79PAdFk
vaAAYvFAbDETgDoBM8Aeid4Z7QTQu
kfUcple2V/bM7QGZjnOZMNRTwXv12
OY2CqyWQucXONucS5xO5c424nzJKQ
HpPFssnA9+9DhMMD1yS4qDKfg7Dzeq
pvbCZ1cVBJ+6xeG7IWe5lZi425P5e4xg0
/lCzzuOTmJ0GYdm+JkDm1vHHKZma/z
Bzna9DSVxbj02Ij7OQRiy10j2MyyTPYw
xsfM6+84Nc4CgB3iatFASvbl5PEMVZJq

```
INGjkBauvbP7R2kb8S+KUvjuOeHLkna
HEZi4NbmeD3TYBFDzObc6zaaQmxRlP
LRIQhWSCEIQAIXEIECLQuIA7aFxCY
E1r062VQ8y7nTTJcbJ4mXRiRtf781Vue
SuWjWydpFriJO6f3zUBrk0152tdzKZOy
xouBo1pop2GxY+2cHZkYC1nDyZAX53
Zg1uUguyUL5NB8VjZcU50cURrLC1zW
UNafI6U2efeefhSnO47IThnZIg/C9l2cgY
7tHCGuzbIS6nAUNAAigND/1KykYKVo
oTxSzV0MsnavHhT5XhYpTMdxOWVk
UchBbCHiPSiBI/O4E8xe3RQk0gO2hcQ
gDq4hCABCEJgCEIQAIQhAAhdQhAC
EIQBxdQhIDhXUIQBxCEJgdQhCQAhC
EAC4uoQAIQhAHEIQmAIQhAAhCEA
CEIQAIQhAAhCEAf//Z",

            ),
            const SizedBox(height: 12),
            const _PlaylistCard(
              title: "CHILL",
              saves: "0 saves",
              imageUrl:
"
gABAQAAAQABAAD/2wCEAAkGBxI
TEhUSEhMWFhIXFhcYFxcWFhgXGBc
WGBgWGBcXFRcZHiggGB0lGxUWIjE
hJikrLi4uGB8zODMsNygtLisBCgoKDg0
OGxAQGy8lICY1LS0tLS0vLS0tKystLS
0tLS0tLS0tLS0tMi0tLS0tLS0tLS8tL
S0tLS0tNS0tK//AABEIAOEA4QMBIgA
CEQEDEQH/xAAcAAEAAgMBAQEAA
AAAAAAAAAAABQYDBAcCAQj/xAB
JEAACAQIDBAYFCQUGBgIDAAABA
gMAEQQSIQUGMVETIkFhcYEHMpGx
wRQVI0JScpKh0TNiorLhU2NzgpPwFy
Q0VMLSdPEWNUP/xAAaAQEAAwEB
AQAAAAAAAAAAAAAAAQIDBAUG/
8QAKREAAgICAgIABgEFAAAAAAA
AAAECEQMhEjEEQRNRYYGR8MEyQ
mJxof/aAAwDAQACEQMRAD8A7jSlK
AUpSgFKUoBSlKAUpSgFKUoBSlKAU
pSgFKUoBSlKAUpSgFKUoBSlKAUpSg
VwHgBsB4K49kXlo4cGnR/EJu0o8
MwA9Rkkfp/WeqyfEsc+eV80lZ5HFzqF
CzvQ5BTOE+0E2HYWRiM94yMc9mZ0
UpZ2ZkiN912UAa2O6NNEUBat4wIMB
g2sjBdJh8UwSF8lNZJiMRG4dkCGPOV
dk7WgxMfSwSB47kXFxYjjcGxHnQi0bt
Krbb+bND9H8qXMGy+q9r3t62W1r9t7V
L7T2rBh4ummkVItOsdb34WAuT5UFo3a
VC7H3swWKfo4J1d7Xy2ZTYcswF/Kt/A
7SimMixvmMTmOQWIyuOK6jXyoLNul
amH2nE8skCveWIIZFseqHF11IsbjlW3Q
kUqJ23vJhcIVGJlEZcErdWNwtr+qDzHtr
Swe/WzpZFijxKtI7BVXJILsdANVtQi0W
OlKUJFKUoBSlKAUpSgFKUoBSlKAU
pSgFKUoCF31//AF+L/wDjzfyNXPdl7Kx
5ghwizMMFNhlxBkyMTGGQM0Af1Rdi
bDlXSt5cG82ExEMdi8kMiLc2GZlIFz2a
mvOxsE8eChgYDpEw8cbAG4zLGFNjyu
ONSVatnNMLi77P2XhXmEOHxBxHTvc
L1Y3LBc5ICgk28xUjtA4PCYnCS7Nniz
STxYeaKKRHV4nNi7KpuCCB1uZHgdn
AbmTdHsuKeON0w/yjp1YhltIbpYH1tbG
pPeHcyInCtg8PDG8eLhkkKqqHokJLC4
GuuU27qkrTorib0QYLFY/p8PJIHxWjKi
FQQnqkuRrYE25VmXCTfJ9qY3oWwsc
0TdHE10cFA15GX6pY/wCyLGrJs7djM2
PTFIrQ4mbOovclcoAOmqsCLg8RYGsW
yt3cSIMRs+eTPhSmSCW46QIwIyMOH
VFrfpahNM2Nn7Jg+a0TokynChiMo1JT
OT+Ik1UNkxiU7BSTroVxTFW1BZFzIS
DxIIFqmosPtpIBhBHhyoXohNm1yDqhuj
4epbSs+O3RmiiwTYRw0+CzBRJYLIsthI
Cfqm3A8r9tjQMb94WNJtnSIiq/y6FMyix
yNcstx2Gwqv7P2nicJicfilXpMGuLkWdF
9ZO3pV7gCL+HZxqwHZe0MXPh2xixQ
w4eQTARNnLyJ6oJPAam/d32qV3W2RL
DJjWlAyzYl5Esb3RgAL8jpwqBVsiNzsd
HPtPaE0TB43jwhVh2joj5gg3BB1BBBq8
VVN1t2fkmMxjxoqYaURdEFN7EBjIMv
1RmY2HC3DQVa6MtHoqm92wMTPiM
NiMM0IaFZRaYMynpAo4Aa6A/lWhBjc
ZhcTAmNjw8kc7iNHgTKUlPC4bUjw4fl
```

```
FKUoBSsOMxSRRvLIcsaKWY6myqLk2
GvAV8w+LR41mRrxsgdWsdUIzA2OvA
0BnpWps7aUU8SzxOGiYEhrECwJBPW
AI1Brxsra8GJDNBIsgVirWvoR3Hs7+Bo
DepVdx2/Gz4ZGilxAWRDZlKSaEdlwtql
),
              ],
            ),
          ),
        ),
        //const _BottomNavBar(),
      ],
    ),
  );
 }
}

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

 @override
 Widget build(BuildContext context) {
  return Padding(
   padding: const
EdgeInsets.symmetric(vertical: 8.0),
   child: Row(
    children: [
     Text(
      "1 follower",
      style:
Theme.of(context).textTheme.bodyMediu
m?.copyWith(
            color: Colors.grey,
           ),
     ),
     const Text(" • ", style:
TextStyle(color: Colors.grey)),
     Text(
      "19 following",
      style:
U1vTgsYxinwctpIm60LG0UytYMH7wO
B8e21RSbL2hi8RBJjFihiw7iVVibOXkH
C5PAWoQ+y60pSoLClKUApSlAKUpQC
lKUApSlAKUpQClKUB8drC9aj7SRTZm
UHkWAPsNbE/qnwNc33lhzYlj3L/KKlEl
],
   ),
  );
 }
}

class _PlaylistCard extends
StatelessWidget {
 final String title;
 final String saves;
 final String? imageUrl; // New optional
parameter

 const _PlaylistCard({
  Key? key,
  required this.title,
  required this.saves,
  this.imageUrl, // Make it optional
 }) : super(key: key);

 @override
 Widget build(BuildContext context) {
  return Container(
   padding: const EdgeInsets.all(12),
   decoration: BoxDecoration(
    color: Colors.grey[900],
    borderRadius:
BorderRadius.circular(8),
   ),
   child: Row(
    children: [
     Container(
      width: 60,
      height: 60,
      decoration: BoxDecoration(
       color: Colors.grey[800],
       borderRadius:
```

```
Theme.of(context).textTheme.bodyMediu
m?.copyWith(
                color: Colors.grey,
              ),
          ),
 : const Icon(Icons.music_note, color:
Colors.white),
          ),
          const SizedBox(width: 12),
          Column(
            crossAxisAlignment:
CrossAxisAlignment.start,
            children: [
              Text(
                title,
                style: const TextStyle(
                  color: Colors.white,
                  fontWeight: FontWeight.bold,
                ),
              ),
              const SizedBox(height: 4),
              Text(
                saves,
                style: TextStyle(color:
Colors.grey[400]),
              ),
            ],
          ),
        ],
      ),
    );
  }
}

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

  @override
  Widget build(BuildContext context) {
    return Stack(
BorderRadius.circular(4),
            ),
            child: imageUrl != null
                ? Image.network(imageUrl!, fit:
BoxFit.cover)
                end:
Alignment.bottomCenter,
              colors: [Color(0xFF9370DB),
Colors.black],
            ),
          ),
        ),
        SafeArea(
          child: Padding(
            padding: const
EdgeInsets.all(16.0),
            child: Column(
              children: [
                Row(
                  children: [
                    IconButton(
                      icon: const
Icon(Icons.arrow_back, color:
Colors.white),
                      onPressed: () {
                        Navigator.pop(context);
                      },
                    ),
                    const Spacer(),
                    IconButton(
                      icon: const
Icon(Icons.more_vert, color:
Colors.white),
                      onPressed: () {},
                    ),
                  ],
                ),
                const SizedBox(height: 20),
                Container(
                  width: 100,
                  height: 100,
                  decoration: BoxDecoration(
```

```
      children: [
        Container(
          decoration: const BoxDecoration(
            gradient: LinearGradient(
              begin: Alignment.topCenter,
 "A",
                    style:
Theme.of(context).textTheme.displayLarg
e?.copyWith(
                        color: Colors.black,
                        fontWeight:
FontWeight.bold,
                      ),
                  ),
                ),
              ),
            ],
          ),
        ),
      ],
    );
  }
}
```

Sign_in_screen.dart

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

import 'register.dart';

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

  @override
  State<SignInScreen> createState() =>
_SignInScreenState();
}

class _SignInScreenState extends
State<SignInScreen> {
  bool _obscurePassword = true;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const
Color(0xFF1C1B1B),
      body: SafeArea(
        child: SingleChildScrollView(
          child: Padding(
            padding: const
EdgeInsets.all(24.0),
            child: Column(
              crossAxisAlignment:
CrossAxisAlignment.start,
              children: [
                Row(
                  children: [
                    IconButton(
                      color: Colors.purple[200],
                      shape: BoxShape.circle,
                    ),
                    child: Center(
                      child: Text(
```

```
 ),
                const SizedBox(width: 12),
                Image.asset(

'assets/images/spotify_logo.png',
                  height: 150,
                ),
              ],
            ),
            const SizedBox(height: 32),
            const Text(
              'Sign In',
              style: TextStyle(
                color: Colors.white,
                fontSize: 28,
                fontWeight: FontWeight.bold,
              ),
            ),
            const SizedBox(height: 8),
            Row(
              children: [
                const Text(
                  'If You Need Any Support ',
                  style: TextStyle(color:
Colors.grey),
                ),
                TextButton(
                  onPressed: () {},
                  child: const Text(
                    'Click Here',
                    style: TextStyle(color:
Colors.green),
                  ),
                ),
              ],
            ),
            const SizedBox(height: 32),
                  icon: const
Icon(Icons.arrow_back, color:
Colors.white),
                  onPressed: () =>
Navigator.pop(context),
                Email',
                hintStyle: const
TextStyle(color: Colors.grey),
                filled: true,
                fillColor:
Colors.grey.withOpacity(0.1),
                border: OutlineInputBorder(
                  borderRadius:
BorderRadius.circular(12),
                  borderSide:
BorderSide.none,
                ),
              ),
            ),
            const SizedBox(height: 16),
            TextField(
              obscureText:
_obscurePassword,
              style: const TextStyle(color:
Colors.white),
              decoration: InputDecoration(
                hintText: 'Password',
                hintStyle: const
TextStyle(color: Colors.grey),
                filled: true,
                fillColor:
Colors.grey.withOpacity(0.1),
                border: OutlineInputBorder(
                  borderRadius:
BorderRadius.circular(12),
                  borderSide:
BorderSide.none,
                ),
                suffixIcon: IconButton(
                  icon: Icon(
                    _obscurePassword
                        ? Icons.visibility_off
```

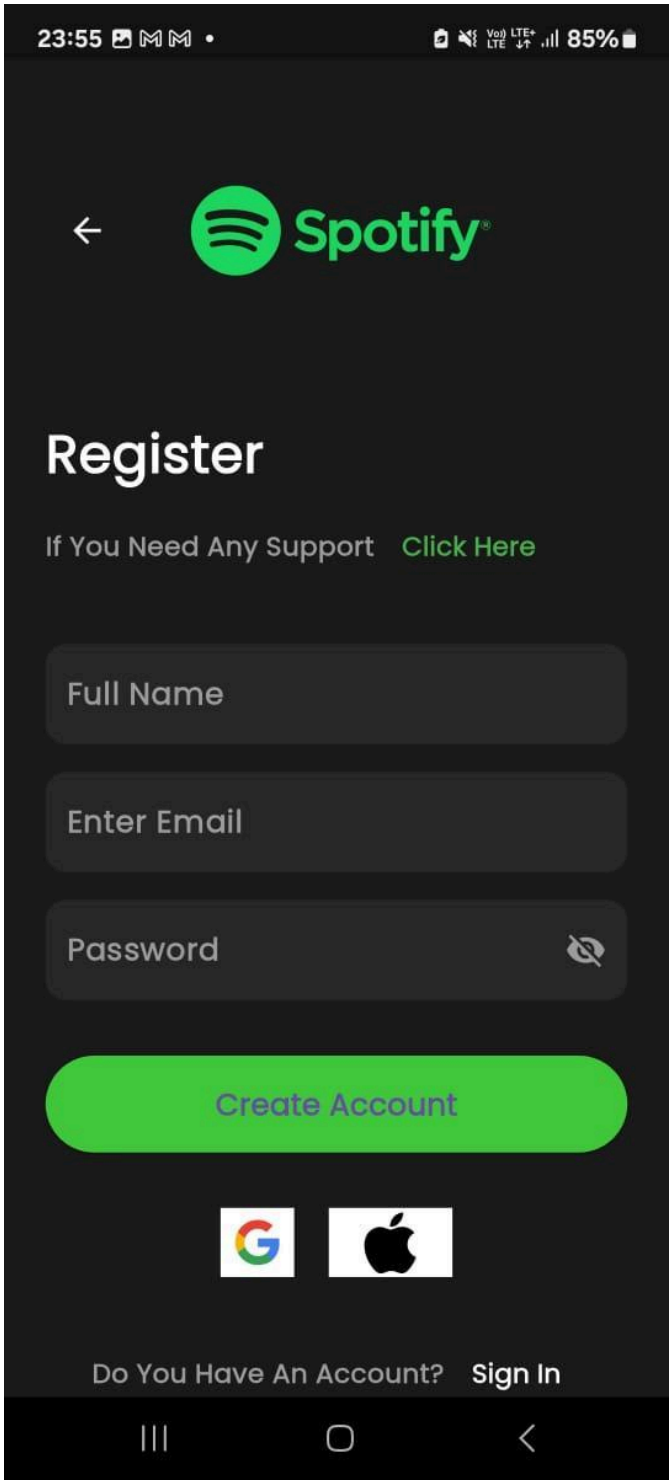
```
            TextField(
              style: const TextStyle(color:
Colors.white),
              decoration: InputDecoration(
                hintText: 'Enter Username Or
_obscurePassword = !_obscurePassword;
                  });
                },
              ),
            ),
          ),
          const SizedBox(height: 16),
          Align(
            alignment:
Alignment.centerRight,
            child: TextButton(
              onPressed: () {},
              child: const Text(
                'Recovery Password',
                style: TextStyle(color:
Colors.grey),
              ),
            ),
          ),
          const SizedBox(height: 32),
          SizedBox(
            width: double.infinity,
            height: 56,
            child: ElevatedButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder: (context) =>
const MainScreen()),
                );
              },
              style:
ElevatedButton.styleFrom(
                backgroundColor: const
Color(0xFF42C83C),
                shape:
                    : Icons.visibility,
                color: Colors.grey,
              ),
              onPressed: () {
                setState(() {
                          ),
              child: const Text(
                'Sign In',
                style: TextStyle(
                  fontSize: 16,
                  fontWeight:
FontWeight.bold,
                ),
              ),
            ),
          ),
          const SizedBox(height: 16),
          const Center(
            child: Text(
              'Or',
              style: TextStyle(color:
Colors.grey),
            ),
          ),
          const SizedBox(height: 16),
          Row(
            mainAxisAlignment:
MainAxisAlignment.center,
            children: [

Image.asset('assets/images/google_logo.pn
g', height: 40),
              const SizedBox(width: 20),

Image.asset('assets/images/apple_logo.png
', height: 40),
            ],
          ),
          const SizedBox(height: 32),
          Row(
            mainAxisAlignment:
MainAxisAlignment.center,
```

```
RoundedRectangleBorder(
                    borderRadius:
BorderRadius.circular(30),
                  ),

  ),
                TextButton(
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                          builder: (context) =>
const Register()),
                      );
                    },
                    child: const Text(
                      'Register Now',
                      style: TextStyle(color:
Colors.blue),
                    ),
                  ),
                ],
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

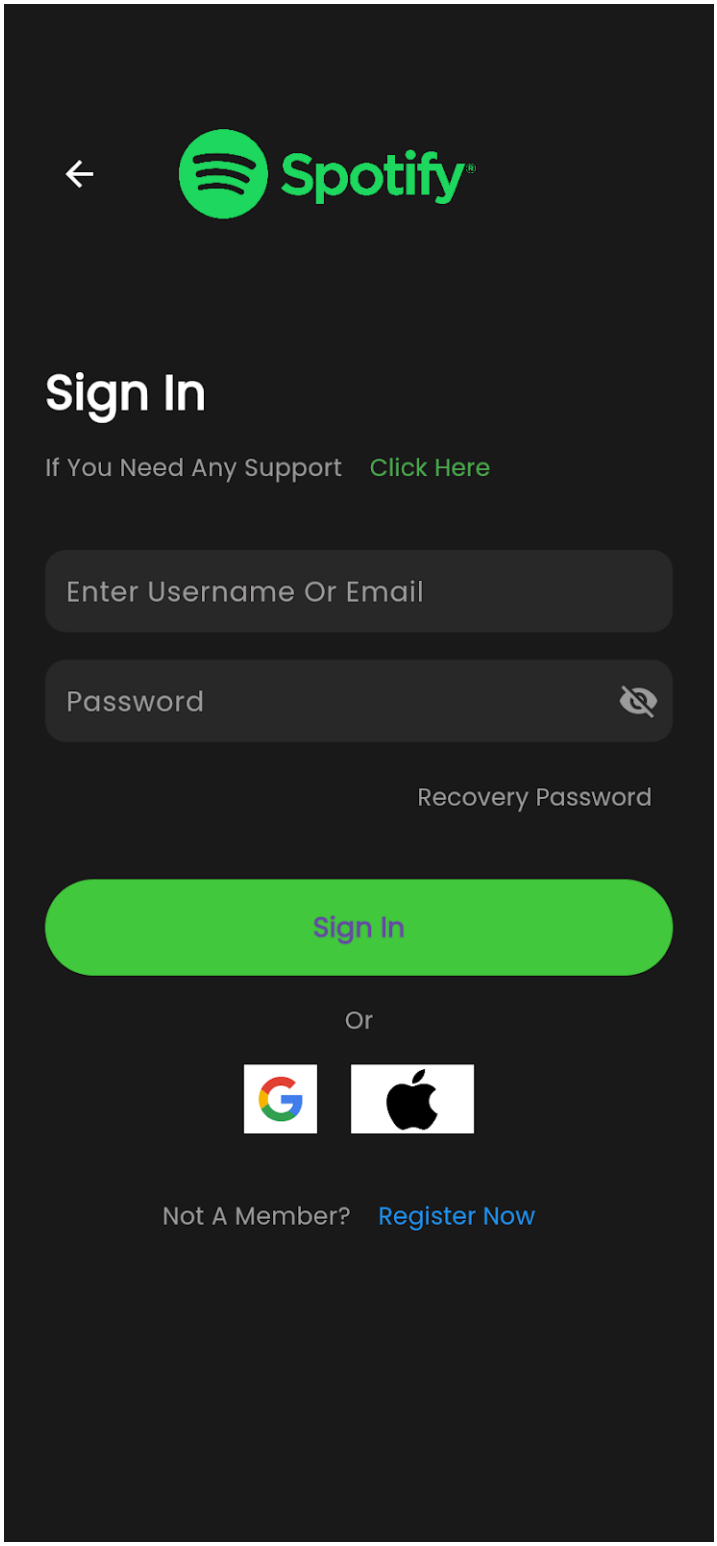
```
                    children: [
                      const Text(
                        'Not A Member? ',
                        style: TextStyle(color:
Colors.grey),
```

OUTPUT:

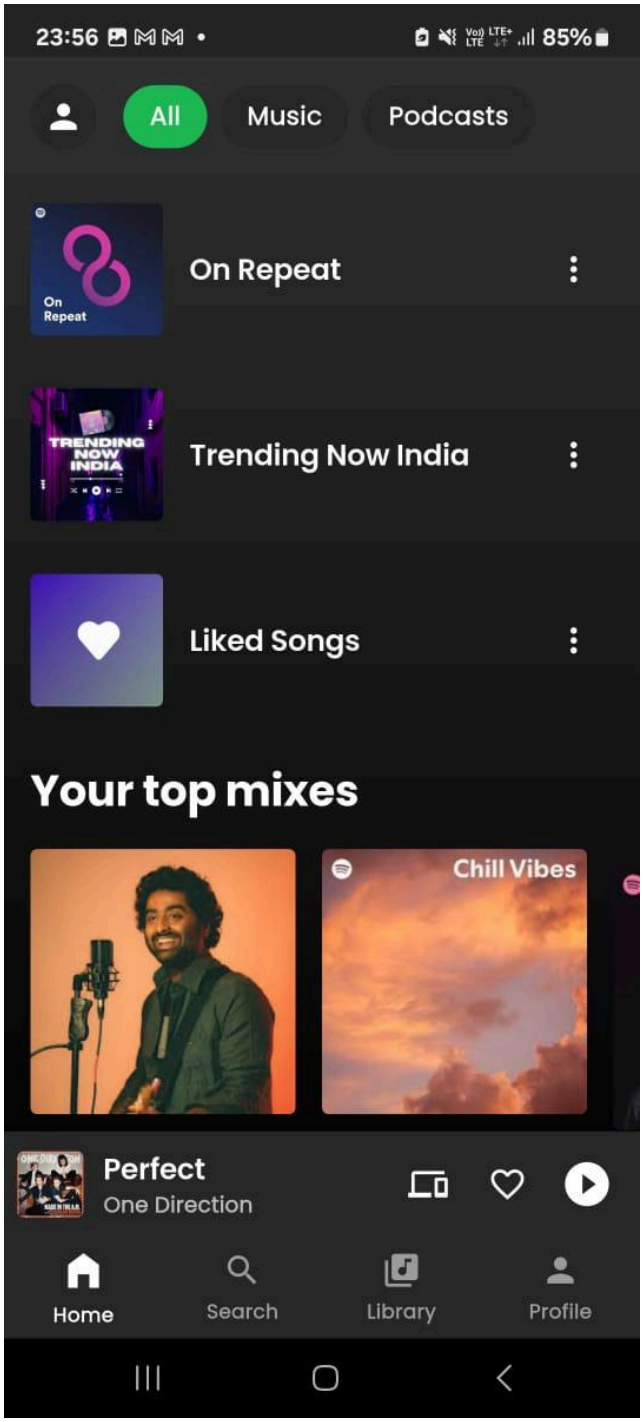
Register.dart


23:55
85%
Spotify
Register
If You Need Any Support Click Here
Full Name
Enter Email
Password
Create Account
Do You Have An Account? Sign In

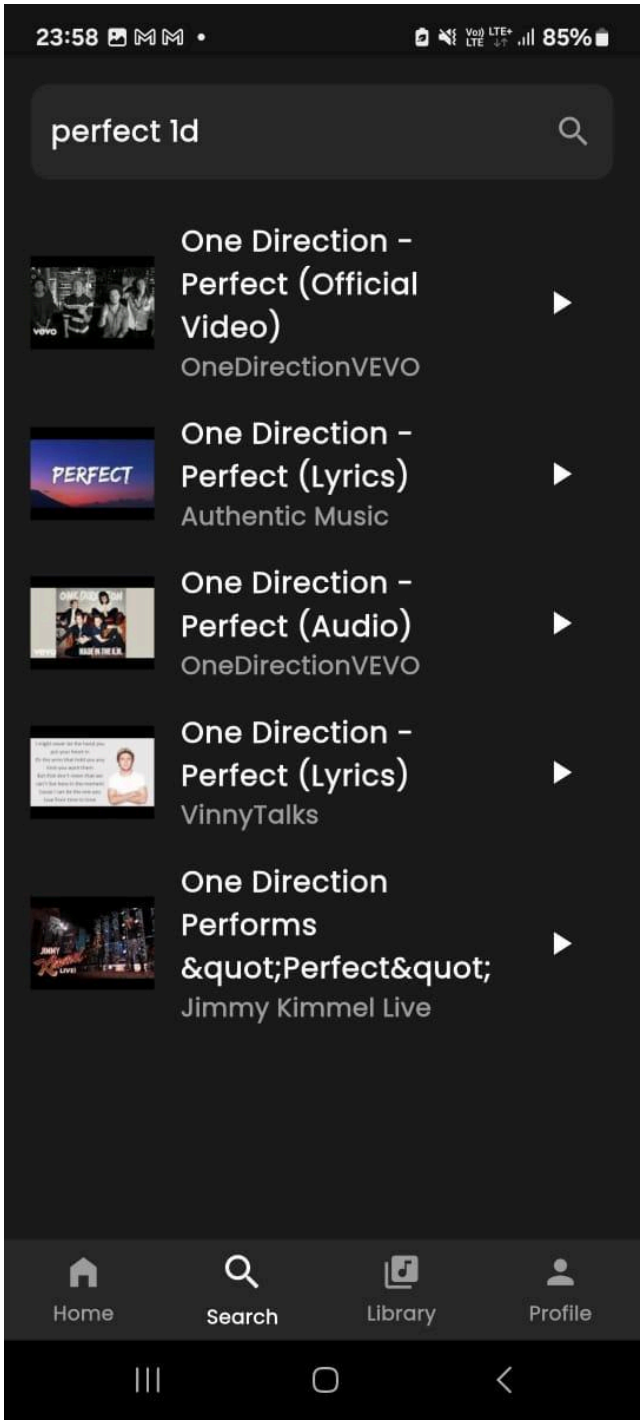
Sign_in.dart



Home_page.dart

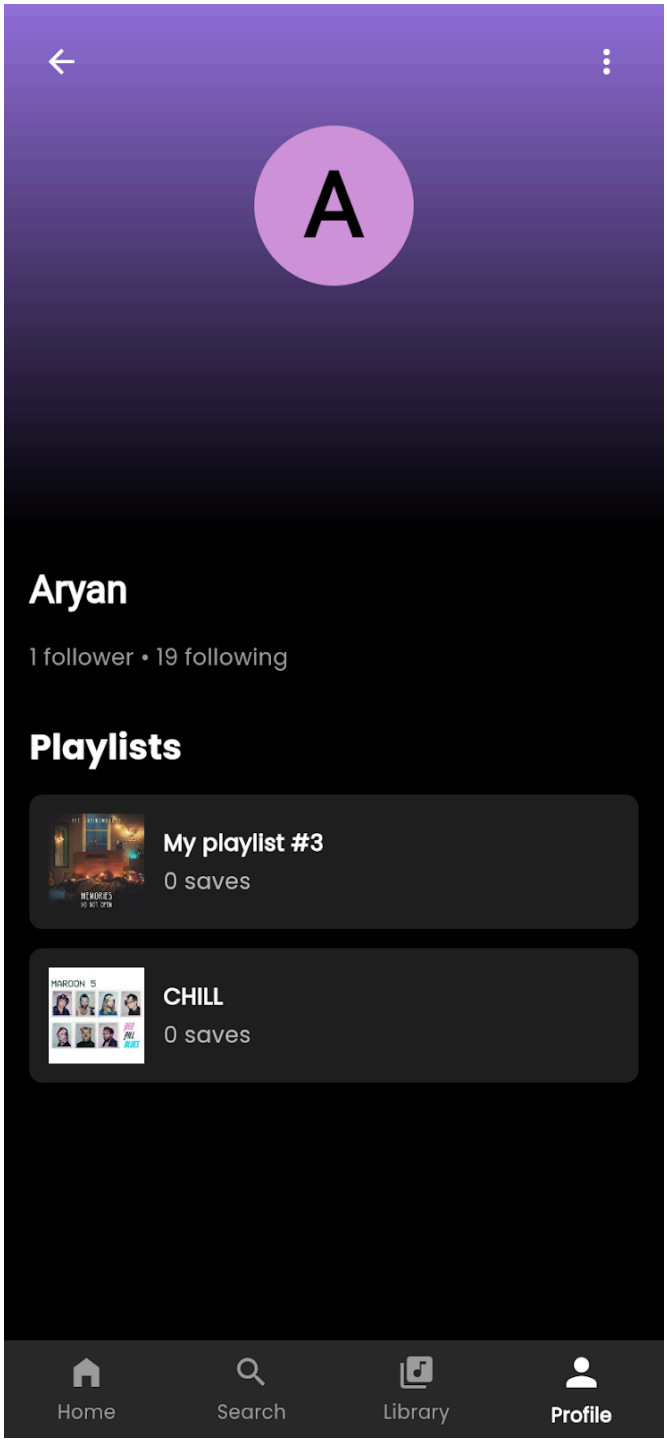


Search_page.dart



Profile_page.dart

Project Title: Spotify **Roll No. 33**



MAD & PWA Lab
Journal

Experiment No.	04
Experiment Title.	To create an interactive Form using form widget
Roll No.	33
Name	Aryan Anil Patankar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

EXPERIMENT NO.4

NAME-ARYAN PATANKAR
CLASS-D15A
ROLL NO-33

AIM:To create an interactive Form using form widget.

Theory:
1.**Introduction**
Forms are a crucial component in mobile applications for collecting user input. Flutter provides the **Form** widget, which helps manage multiple input fields efficiently and allows validation before submission.

The **Form** widget works in combination with **TextFormField** and a **GlobalKey<FormState>** to track the form state, validate inputs, and process the collected data.

2.Form Widget Overview

The **Form** widget in Flutter acts as a container for form fields and manages their state. It provides built-in validation and submission handling.

Key Components of a Form:

1. **Form Widget:**
    - Wraps multiple input fields (TextFormFields) together.
    - Requires a *GlobalKey<FormState>* to track and validate form fields.
2. **TextFormField Widget:**
    - A special type of text field used in forms.
    - Supports validation and user input.
3. **Validator Function:**
    - Ensures correct user input and prevents invalid data.
    - Example: Ensuring an email field contains a valid email format.
4. **Submit Button:**

- Triggers form validation and submission.

3.Working Principle of Form widget

The **Form** widget allows developers to:

- Group multiple **TextFormField** widgets together.
- Validate input before proceeding.
- Save user input after validation.

The **GlobalKey<FormState>** is used to control the form, providing methods like:

- *formKey.currentState!.validate()* → *Checks all validators in the form fields.*
- *formKey.currentState!.save()* → *Saves the form input values.*
- *formKey.currentState!.reset()* → Clears the form fields.

CODE:

Register_signup page

```
import 'package:flutter/material.dart';
import 'package:spotify_clone/features/auth/screens/sign_in_screen.dart';
import 'package:spotify_clone/features/auth/screens/register.dart';

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

  @override
  Widget build(BuildContext context) {
body: SafeArea(
      child: Column(
        children: [
          Padding(
            padding: const EdgeInsets.all(16.0),
            child: Row(
              children: [
                IconButton(
                  icon: const Icon(Icons.arrow_back, color: Colors.white),
                  onPressed: () => Navigator.pop(context),
                ),
              ],
```

```
    return Scaffold(
      backgroundColor: const
Color(0xFF1C1B1B),
      child: SingleChildScrollView(
              child: Padding(
                padding: const
EdgeInsets.symmetric(horizontal:
24.0),
                child: Column(
                  crossAxisAlignment:
CrossAxisAlignment.center,
                  children: [
                    Image.asset(

'assets/images/spotify_logo.png',
                      width: //235
                          500,
                      height: 150,
                    ),
                    const SizedBox(height:
32),
                    const Text(
                      'Enjoy Listening To
Music',
                      style: TextStyle(
                        color: Colors.white,
                        fontSize: 24,
                        fontWeight:
FontWeight.bold,
                      ),
                    ),
                    const SizedBox(height:
16),
                    const Text(
                      'Spotify is a proprietary
Swedish audio\nstreaming and media
services provider',
                      textAlign:
TextAlign.center,
                    ),
                  ),
                ),
                Expanded(
                  ),
                      ),
                      const SizedBox(height:
32),
                      SizedBox(
                        width: double.infinity,
                        height: 56,
                        child: ElevatedButton(
                          onPressed: () {
                            Navigator.push(
                              context,
                              MaterialPageRoute(
                                  builder: (context)
=>
                                      Register()), //
Navigate to ThirdScreen
                            );
                            // Handle register
                          },
                          style:
ElevatedButton.styleFrom(
                            backgroundColor:
const Color(0xFF42C83C),
                            shape:
RoundedRectangleBorder(
                              borderRadius:
BorderRadius.circular(30),
                            ),
                          ),
                          child: const Text(
                            'Register',
                            style: TextStyle(
                              color: Colors.white,
                              fontSize: 16,
                              fontWeight:
FontWeight.bold,
```

```
                  style: TextStyle(
                    color: Colors.grey,
                    fontSize: 16,
                                ),
                const SizedBox(height:
16),
                TextButton(
                  onPressed: () {
                    Navigator.push(
                        context,
                        MaterialPageRoute(
                            builder: (context)
=>

SignInScreen()), // Navigate to
ThirdScreen
                          );
                    // Handle sign in
                  },
                  child: const Text(
                    'Sign in',
                    style: TextStyle(
                      color: Colors.white,
                      fontSize: 16,
                    ),
                  ),
                ),
              ],
            ),
          ),
        ),
      ],
    ),
  ),
 );
 }
}
```

Register.dart

```
import 'package:flutter/material.dart';
import
'package:spotify_clone/features/auth/sc
reens/sign_in_screen.dart';
import
'package:spotify_clone/main.dart';

class Register extends StatefulWidget {
  const Register({Key? key}) :
super(key: key);

  @override
  State<Register> createState() =>
_RegisterState();
}

class _RegisterState extends
State<Register> {
  bool _obscurePassword = true;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const
Color(0xFF1C1B1B),
      body: SafeArea(
        child: SingleChildScrollView(
          child: Padding(
            padding: const
EdgeInsets.all(24.0),
            child: Column(
              crossAxisAlignment:
CrossAxisAlignment.start,
              children: [
                Row(
```

```
Icon(Icons.arrow_back, color:
Colors.white),
                onPressed: () =>
Navigator.pop(context),
              ),
              const SizedBox(width: 10),
              Image.asset(

'assets/images/spotify_logo.png',
                height: 150,
              ),
            ],
          ),
          const SizedBox(height: 32),
          const Text(
            'Register',
            style: TextStyle(
              color: Colors.white,
              fontSize: 28,
              fontWeight:
FontWeight.bold,
            ),
          ),
          const SizedBox(height: 8),
          Row(
            children: [
              const Text(
                'If You Need Any Support
',
                style: TextStyle(color:
Colors.grey),
              ),
              TextButton(
                onPressed: () {},
                child: const Text(
                  'Click Here',
                children: [
                  IconButton(
                    icon: const
),
                  ),
                ],
              ),
              const SizedBox(height: 32),
              TextField(
                style: const TextStyle(color:
Colors.white),
                decoration:
InputDecoration(
                  hintText: 'Full Name',
                  hintStyle: const
TextStyle(color: Colors.grey),
                  filled: true,
                  fillColor:
Colors.grey.withOpacity(0.1),
                  border:
OutlineInputBorder(
                    borderRadius:
BorderRadius.circular(12),
                    borderSide:
BorderSide.none,
                  ),
                ),
              ),
              const SizedBox(height: 16),
              TextField(
                style: const TextStyle(color:
Colors.white),
                decoration:
InputDecoration(
                  hintText: 'Enter Email',
                  hintStyle: const
TextStyle(color: Colors.grey),
                  filled: true,
                  fillColor:
```

```
                    style: TextStyle(color:
Colors.green),
  borderRadius:
BorderRadius.circular(12),
                    borderSide:
BorderSide.none,
                  ),
                ),
              ),
              const SizedBox(height: 16),
              TextField(
                obscureText:
_obscurePassword,
                style: const TextStyle(color:
Colors.white),
                decoration:
InputDecoration(
                  hintText: 'Password',
                  hintStyle: const
TextStyle(color: Colors.grey),
                  filled: true,
                  fillColor:
Colors.grey.withOpacity(0.1),
                  border:
OutlineInputBorder(
                    borderRadius:
BorderRadius.circular(12),
                    borderSide:
BorderSide.none,
                  ),
                  suffixIcon: IconButton(
                    icon: Icon(
                      _obscurePassword
                          ? Icons.visibility_off
                          : Icons.visibility,
                      color: Colors.grey,
                    ),
                    onPressed: () {
                      setState(() {
Colors.grey.withOpacity(0.1),
                  border:
OutlineInputBorder(
                                        });
                    },
                  ),
                ),
              ),
              const SizedBox(height: 32),
              SizedBox(
                width: double.infinity,
                height: 56,
                child: ElevatedButton(
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                        builder: (context) =>
const MainScreen()),
                    );
                  },
                  style:
ElevatedButton.styleFrom(
                    backgroundColor: const
Color(0xFF42C83C),
                    shape:
RoundedRectangleBorder(
                      borderRadius:
BorderRadius.circular(30),
                    ),
                  ),
                  child: const Text(
                    'Create Account',
                    style: TextStyle(
                      fontSize: 16,
                      fontWeight:
FontWeight.bold,
                    ),
                  ),
```

```
                    _obscurePassword =
!_obscurePassword;
Row(
              mainAxisAlignment:
MainAxisAlignment.center,
              children: [

Image.asset('assets/images/google_logo
.png', height: 40),
                const SizedBox(width: 20),

Image.asset('assets/images/apple_logo.
png', height: 40),
              ],
            ),
            const SizedBox(height: 32),
            Row(
              mainAxisAlignment:
MainAxisAlignment.center,
              children: [
                const Text(
                  'Do You Have An
Account? ',
                  style: TextStyle(color:
Colors.grey),
                ),
                TextButton(
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                          builder: (context) =>
const SignInScreen()),
                    );
                  },
                  child: const Text(
                    'Sign In',
                    style: TextStyle(color:
Colors.white),
                  ),
                ),
                const SizedBox(height: 32),
              ],
            ),
          ],
        ),
      ),
    );
  }
}
```

```
                ),
```

Sign_in_screen.dart

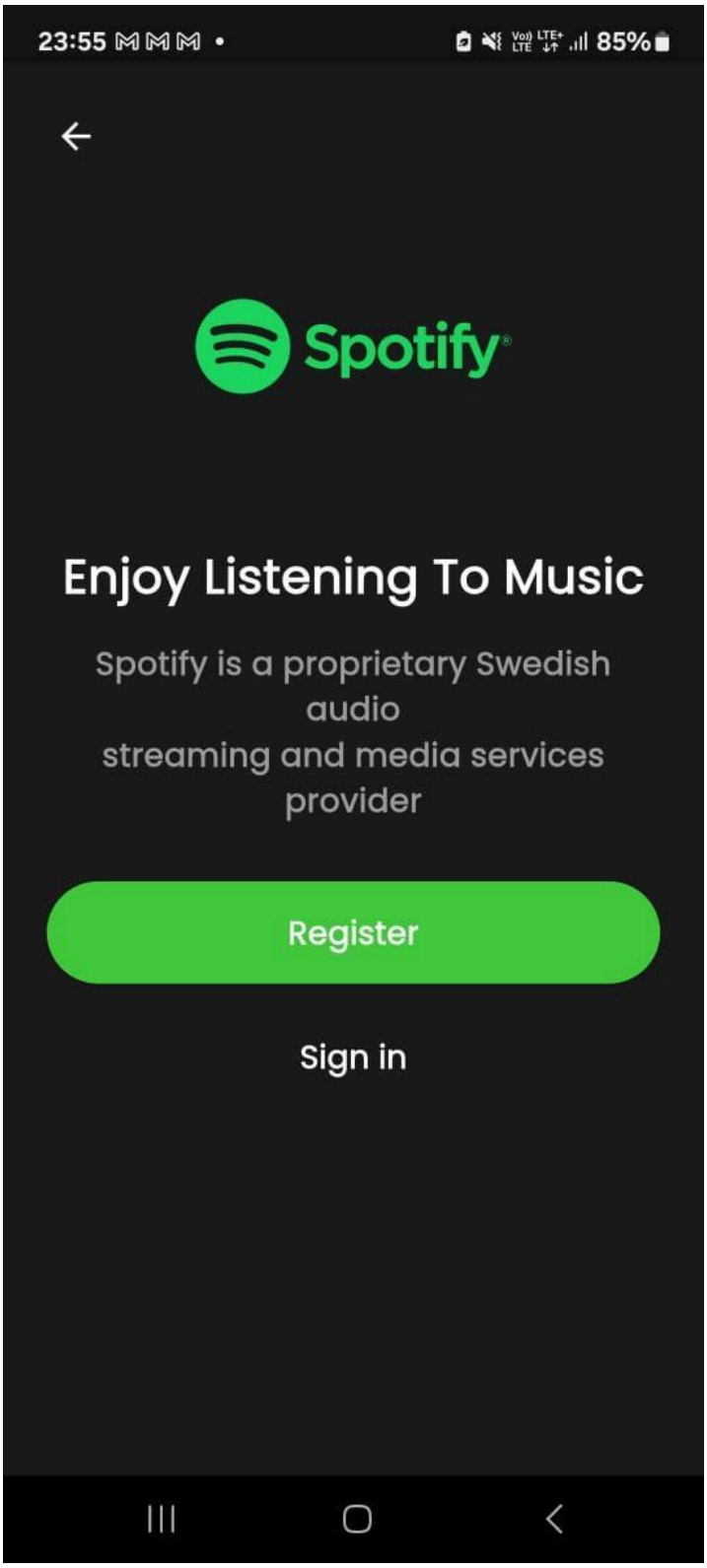
```
import 'package:flutter/material.dart';
import
'package:spotify_clone/main.dart';

import 'register.dart';

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

  @override
  State<SignInScreen> createState() =>
_SignInScreenState();
}

class _SignInScreenState extends
State<SignInScreen> {
  bool _obscurePassword = true;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const
Color(0xFF1C1B1B),
      body: SafeArea(
        child: SingleChildScrollView(
          child: Padding(
            padding: const
EdgeInsets.all(24.0),
            child: Column(
              crossAxisAlignment:
CrossAxisAlignment.start,
              children: [
                Row(
                  children: [
 icon: const Icon(Icons.arrow_back,
color: Colors.white),
                    onPressed: () =>
Navigator.pop(context),
                  ),
                  const SizedBox(width: 12),
                  Image.asset(

'assets/images/spotify_logo.png',
                    height: 150,
                  ),
                ],
              ),
              const SizedBox(height: 32),
              const Text(
                'Sign In',
                style: TextStyle(
                  color: Colors.white,
                  fontSize: 28,
                  fontWeight:
FontWeight.bold,
                ),
              ),
              const SizedBox(height: 8),
              Row(
                children: [
                  const Text(
                    'If You Need Any Support
',
                    style: TextStyle(color:
Colors.grey),
                  ),
                  TextButton(
                    onPressed: () {},
                    child: const Text(
                      'Click Here',
```

```
                IconButton(
                  Colors.green),
                  ),
                ),
              ],
            ),
            const SizedBox(height: 32),
            TextField(
              style: const TextStyle(color:
Colors.white),
              decoration:
InputDecoration(
                hintText: 'Enter Username
Or Email',
                hintStyle: const
TextStyle(color: Colors.grey),
                filled: true,
                fillColor:
Colors.grey.withOpacity(0.1),
                border:
OutlineInputBorder(
                  borderRadius:
BorderRadius.circular(12),
                  borderSide:
BorderSide.none,
                ),
              ),
            ),
            const SizedBox(height: 16),
            TextField(
              obscureText:
_obscurePassword,
              style: const TextStyle(color:
Colors.white),
              decoration:
InputDecoration(
                hintText: 'Password',
                hintStyle: const
TextStyle(color: Colors.grey),
                    style: TextStyle(color:

fillColor: Colors.grey.withOpacity(0.1),
                border:
OutlineInputBorder(
                  borderRadius:
BorderRadius.circular(12),
                  borderSide:
BorderSide.none,
                ),
                suffixIcon: IconButton(
                  icon: Icon(
                    _obscurePassword
                        ? Icons.visibility_off
                        : Icons.visibility,
                    color: Colors.grey,
                  ),
                  onPressed: () {
                    setState(() {
                      _obscurePassword =
!_obscurePassword;
                    });
                  },
                ),
              ),
            ),
            const SizedBox(height: 16),
            Align(
              alignment:
Alignment.centerRight,
              child: TextButton(
                onPressed: () {},
                child: const Text(
                  'Recovery Password',
                  style: TextStyle(color:
Colors.grey),
                ),
              ),
            ),
```

```
            filled: true,
                    SizedBox(
          width: double.infinity,
          height: 56,
          child: ElevatedButton(
            onPressed: () {
              Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context) =>
const MainScreen()),
              );
            },
            style:
ElevatedButton.styleFrom(
              backgroundColor: const
Color(0xFF42C83C),
              shape:
RoundedRectangleBorder(
                borderRadius:
BorderRadius.circular(30),
              ),
            ),
            child: const Text(
              'Sign In',
              style: TextStyle(
                fontSize: 16,
                fontWeight:
FontWeight.bold,
              ),
            ),
          ),
        ),
        const SizedBox(height: 16),
        const Center(
          child: Text(
            'Or',
            style: TextStyle(color:
Colors.grey),
          ),
        ),
        const SizedBox(height: 32),
        Row(
          mainAxisAlignment:
MainAxisAlignment.center,
          children: [
            Image.asset('assets/images/google_logo
.png', height: 40),
            const SizedBox(width: 20),
            Image.asset('assets/images/apple_logo.
png', height: 40),
          ],
        ),
        const SizedBox(height: 32),
        Row(
          mainAxisAlignment:
MainAxisAlignment.center,
          children: [
            const Text(
              'Not A Member? ',
              style: TextStyle(color:
Colors.grey),
            ),
            TextButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder: (context) =>
const Register()),
                );
              },
              child: const Text(
                'Register Now',
                style: TextStyle(color:
```

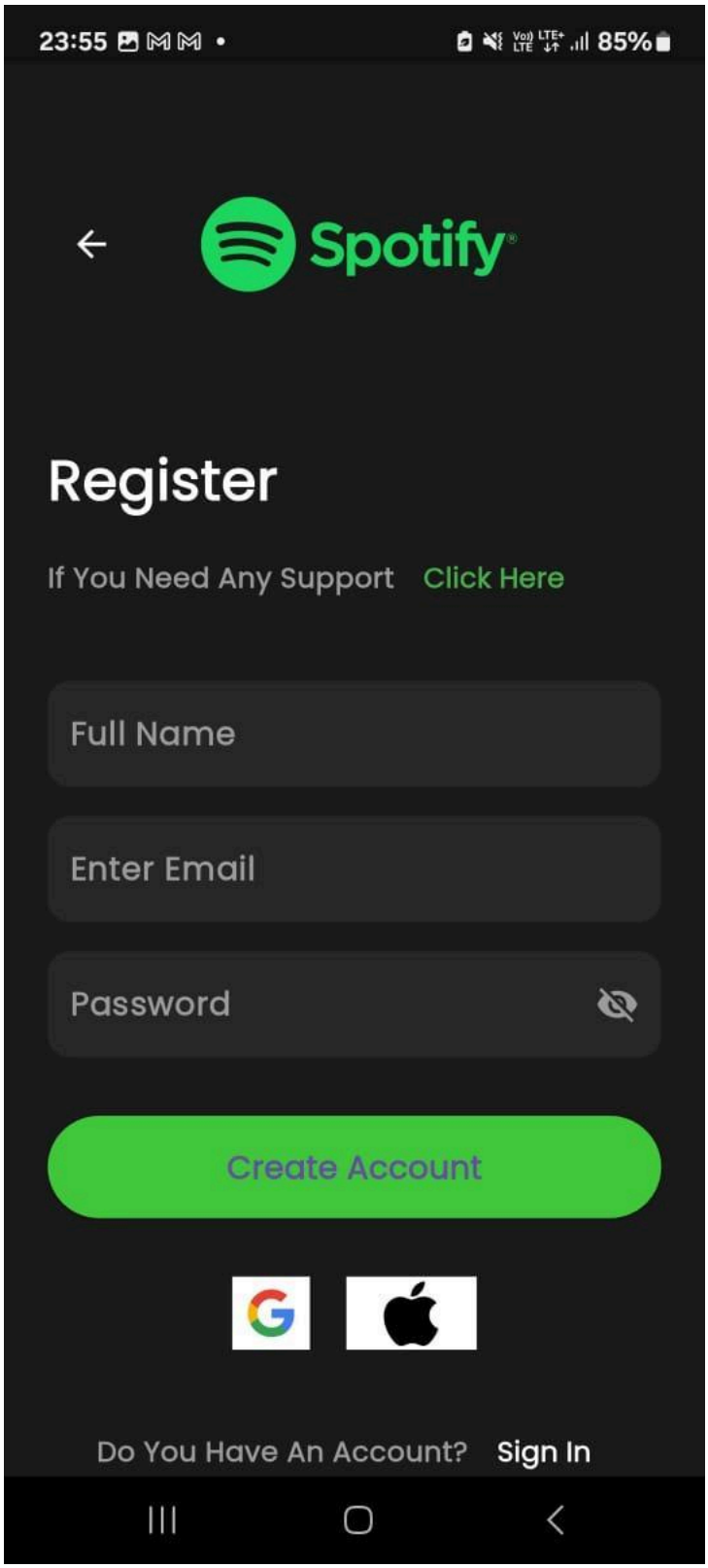
```
                ),                                  Colors.blue),
                                    ),
                  ),
                 ],
               ),
             ],
           ),
         ),
       ),
     ),
   );
  }
}
```

OUTPUT (Register/Signup page)

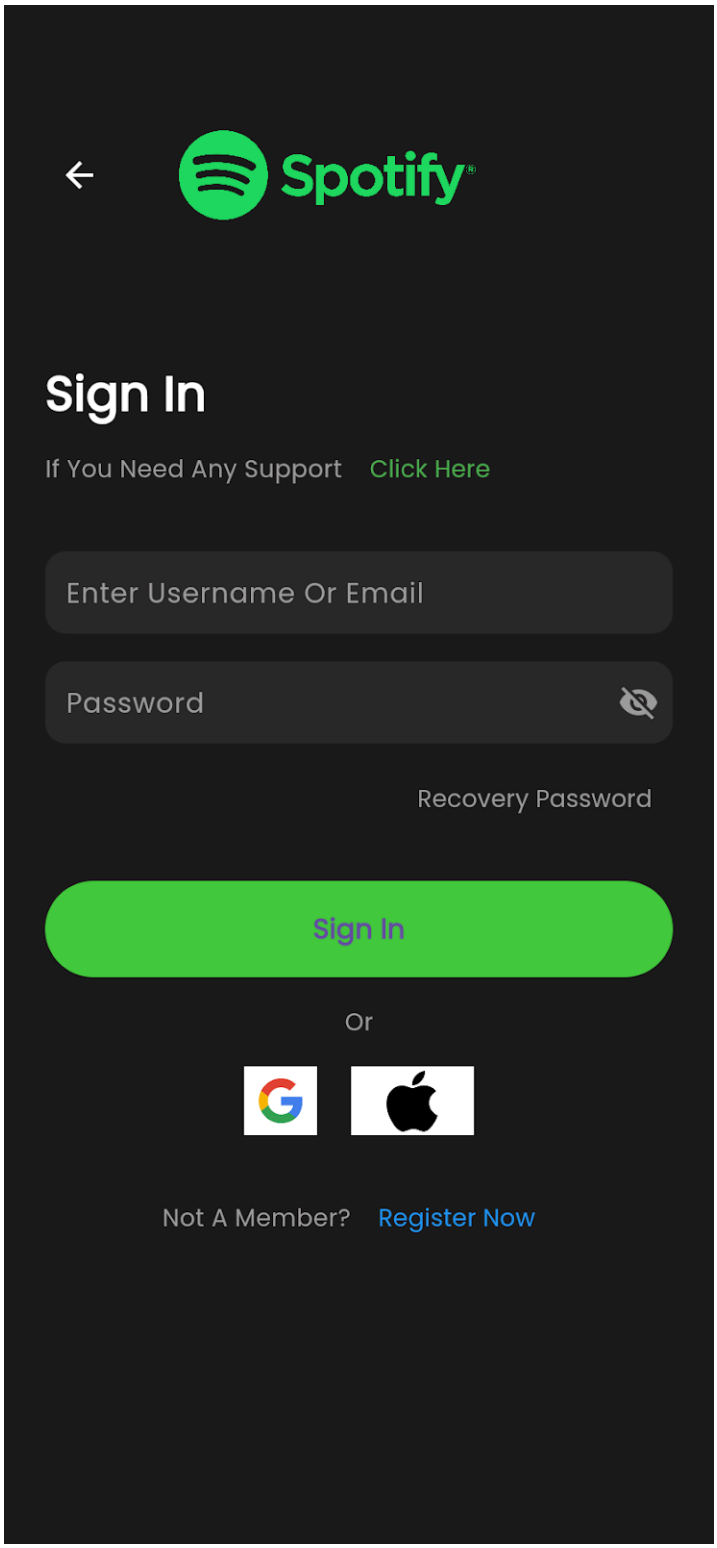


-

Register page



Sign In Page


Spotify
Sign In
If You Need Any Support Click Here
Enter Username Or Email
Password
Recovery Password
Sign In
Or
Not A Member? Register Now

MAD & PWA Lab
Journal

Experiment No.	05
Experiment Title.	To apply navigation, routing and gestures in Flutter App
Roll No.	33
Name	Aryan Anil Patankar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

EXPERIMENT NO.5

NAME:ARYAN ANIL PATANKAR
CLASS:D15A
ROLL NO:33

AIM:To apply navigation, routing and gestures in Flutter app.

Theory:

In Flutter, the screens and pages are known as routes, and these routes are just a widget. In Android, a route is similar to an Activity.
In any mobile app, navigating to different pages defines the workflow of the application, and the way to handle the navigation is known as routing. Flutter provides a basic routing class MaterialPageRoute and two methods Navigator.push() and Navigator.pop() that shows how to navigate between two routes. The following steps are required to start navigation in your application.
Gestures enable the app to respond to user interactions, making the application more dynamic and responsive.

➢ **Navigation and Routing in Flutter**

Navigation is the process of moving between different screens or pages in an app. Flutter provides a simple and effective way to handle this through the use of the Navigator widget and
routes.

1. Using Navigator Widget

The Navigator widget manages a stack of routes, allowing for pushing and popping routes on
the stack.
• **Pushing a Route**: To navigate to a new screen, use Navigator.push().
• **Popping a Route**: To go back to the previous screen, use Navigator.pop().

ElevatedButton(

```
  onPressed: () {
    Navigator.push(
                                                    Aryan Patankar – D15A / 33
      context,
      MaterialPageRoute(builder: (context) => SecondScreen()),
    );},
);
```

2. **Named Routes**

Flutter also allows the use of named routes to navigate, which can make the routing process
cleaner, especially in larger applications.

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

Navigate to the route using Navigator.pushNamed()

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

3. Handling Gestures in Flutter

Gestures refer to user interactions with the app, such as taps, swipes, pinches, and drags. Flutter provides several widgets and gesture detectors to handle these interactions.

4.Tap Gestures

The most common gesture is the tap, which can be handled using the GestureDetector widget or specific buttons like InkWell or ElevatedButton.

5.Long Press Gesture

For long press gestures, Flutter provides the onLongPress callback in GestureDetector or InkWell.

6.Swipe and Drag Gestures

Flutter also provides swipe and drag gesture handling. The onHorizontalDragUpdate and onVerticalDragUpdate callbacks are used for dragging gestures.

Main.dart

```
import 'package:flutter/material.dart';
import
'features/auth/screens/home_screen.dart';
import
'features/auth/screens/search_screen.dart';
import
'features/auth/screens/profile_page.dart';
import
'features/auth/screens/music_screen.dart';
import
'features/auth/screens/sign_in_screen.dart';
import
'features/auth/screens/loading_screen.dart';
import
'features/auth/screens/register_screen.dart';
import
'package:flutter_riverpod/flutter_riverpod.
dart';
import 'config/theme.dart';


void main() {
  runApp(
    const ProviderScope(
      child: SpotifyClone(),
    ),
  );
}
class SpotifyClone extends
StatelessWidget {
  const SpotifyClone({Key? key}) :
super(key: key);
// Keep instances of screens to maintain
their state
  final List<Widget> _screens = const [
    HomeScreen(),
    SearchScreen(),
    MusicScreen(),
    ProfileScreen(),
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner:
false,
      title: 'Spotify Clone',
      theme: AppTheme.lightTheme,
      home: const LoadingScreen(), //
Changed initialRoute to home
      routes: {
        '/signin': (context) => const
SignInScreen(),
        '/main': (context) => const
MainScreen(),
      },
    );
  }
}

class MainScreen extends StatefulWidget
{
  const MainScreen({Key? key}) :
super(key: key);

  @override
  State<MainScreen> createState() =>
_MainScreenState();
}

class _MainScreenState extends
State<MainScreen> {
  int _selectedIndex = 0;
 type: BottomNavigationBarType.fixed,
          backgroundColor: const
Color(0xFF282828),
          selectedItemColor: Colors.white,
          unselectedItemColor: Colors.grey,
          selectedIconTheme: const
IconThemeData(
```

```
  ];

 void _onItemTapped(int index) {
   setState(() {
     _selectedIndex = index;
   });
 }

  @override
  Widget build(BuildContext context) {
   return WillPopScope(
     // Handle back button press
     onWillPop: () async {
       if (_selectedIndex != 0) {
         setState(() {
           _selectedIndex = 0;
         });
         return false;
       }
       return true;
     },
     child: Scaffold(
       // Use IndexedStack to preserve state
of all screens
       body: IndexedStack(
         index: _selectedIndex,
         children: _screens,
       ),
       bottomNavigationBar: Theme(
         data: Theme.of(context).copyWith(
           canvasColor: Colors.black,
         ),
         child: BottomNavigationBar(
           currentIndex: _selectedIndex,
           onTap: _onItemTapped,

 BottomNavigationBarItem(
               icon: Padding(
                 padding:
EdgeInsets.only(bottom: 4),
                 child:
Icon(Icons.library_music),
                 size: 28,
                 color: Colors.white,
               ),
               unselectedIconTheme: const
IconThemeData(
                 size: 24,
                 color: Colors.grey,
               ),
               selectedLabelStyle: const
TextStyle(
                 fontSize: 12,
                 fontWeight: FontWeight.bold,
               ),
               unselectedLabelStyle: const
TextStyle(
                 fontSize: 12,
               ),
               items: const [
                 BottomNavigationBarItem(
                   icon: Padding(
                     padding:
EdgeInsets.only(bottom: 4),
                     child: Icon(Icons.home_filled),
                   ),
                   label: 'Home',
                 ),
                 BottomNavigationBarItem(
                   icon: Padding(
                     padding:
EdgeInsets.only(bottom: 4),
                     child: Icon(Icons.search),
                   ),
                   label: 'Search',
                 ),
```

Sign_in.dart

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

import 'register.dart';
```

```
            ),
            label: 'Library',
          ),
          BottomNavigationBarItem(
            icon: Padding(
              padding:
EdgeInsets.only(bottom: 4),
              child: Icon(Icons.person),
            ),
            label: 'Profile',
          ),
        ],
      ),
    ),
  ),
 );
 }
}
```

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

 @override
 State<SignInScreen> createState() =>
_SignInScreenState();
}

class _SignInScreenState extends
State<SignInScreen> {
 bool _obscurePassword = true;

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   backgroundColor: const
Color(0xFF1C1B1B),
    body: SafeArea(
     child: SingleChildScrollView(
      child: Padding(
       padding: const
EdgeInsets.all(24.0),
        child: Column(
         crossAxisAlignment:
CrossAxisAlignment.start,
          children: [
           Row(
            children: [
             IconButton(
              icon: const
Icon(Icons.arrow_back, color:
Colors.white),
               onPressed: () =>
Navigator.pop(context),
             ),
             const SizedBox(width: 12),
             Image.asset(

'assets/images/spotify_logo.png',
              height: 150,
          hintText: 'Enter Username Or
Email',
            hintStyle: const
TextStyle(color: Colors.grey),
            filled: true,
            fillColor:
Colors.grey.withOpacity(0.1),
```

```
            ),
          ],
        ),
        const SizedBox(height: 32),
        const Text(
          'Sign In',
          style: TextStyle(
            color: Colors.white,
            fontSize: 28,
            fontWeight: FontWeight.bold,
          ),
        ),
        const SizedBox(height: 8),
        Row(
          children: [
            const Text(
              'If You Need Any Support ',
              style: TextStyle(color:
Colors.grey),
            ),
            TextButton(
              onPressed: () {},
              child: const Text(
                'Click Here',
                style: TextStyle(color:
Colors.green),
              ),
            ),
          ],
        ),
        const SizedBox(height: 32),
        TextField(
          style: const TextStyle(color:
Colors.white),
          decoration: InputDecoration(
 setState(() {
                _obscurePassword =
!_obscurePassword;
              });
            },
          ),
        ),
                border: OutlineInputBorder(
                  borderRadius:
BorderRadius.circular(12),
                  borderSide:
BorderSide.none,
                ),
              ),
            ),
            const SizedBox(height: 16),
            TextField(
              obscureText:
_obscurePassword,
              style: const TextStyle(color:
Colors.white),
              decoration: InputDecoration(
                hintText: 'Password',
                hintStyle: const
TextStyle(color: Colors.grey),
                filled: true,
                fillColor:
Colors.grey.withOpacity(0.1),
                border: OutlineInputBorder(
                  borderRadius:
BorderRadius.circular(12),
                  borderSide:
BorderSide.none,
                ),
                suffixIcon: IconButton(
                  icon: Icon(
                    _obscurePassword
                        ? Icons.visibility_off
                        : Icons.visibility,
                    color: Colors.grey,
                  ),
                  onPressed: () {
                                ),
                ),
                child: const Text(
                  'Sign In',
                  style: TextStyle(
                    fontSize: 16,
                    fontWeight:
```

```
            ),
            const SizedBox(height: 16),
            Align(
              alignment:
Alignment.centerRight,
              child: TextButton(
                onPressed: () {},
                child: const Text(
                  'Recovery Password',
                  style: TextStyle(color:
Colors.grey),
                ),
              ),
            ),
            const SizedBox(height: 32),
            SizedBox(
              width: double.infinity,
              height: 56,
              child: ElevatedButton(
                onPressed: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                       builder: (context) =>
const MainScreen()),
                  );
                },
                style:
ElevatedButton.styleFrom(
                  backgroundColor: const
Color(0xFF42C83C),
                  shape:
RoundedRectangleBorder(
                    borderRadius:
BorderRadius.circular(30),
Colors.grey),
                ),
                TextButton(
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
FontWeight.bold,
                    ),
                  ),
                ),
              ),
              const SizedBox(height: 16),
              const Center(
                child: Text(
                  'Or',
                  style: TextStyle(color:
Colors.grey),
                ),
              ),
              const SizedBox(height: 16),
              Row(
                mainAxisAlignment:
MainAxisAlignment.center,
                children: [

Image.asset('assets/images/google_logo.pn
g', height: 40),
                  const SizedBox(width: 20),

Image.asset('assets/images/apple_logo.png
', height: 40),
                ],
              ),
              const SizedBox(height: 32),
              Row(
                mainAxisAlignment:
MainAxisAlignment.center,
                children: [
                  const Text(
                    'Not A Member? ',
                    style: TextStyle(color:
```

Register_screen.dart

```
import 'package:flutter/material.dart';
import
'package:spotify_clone/features/auth/scree
ns/sign_in_screen.dart';
import
'package:spotify_clone/features/auth/scree
```

```
                    builder: (context) =>
const Register()),
                  );
                },
                child: const Text(
                 'Register Now',
                 style: TextStyle(color:
Colors.blue),
                ),
              ),
            ],
          ),
        ],
      ),
    ),
  ),
 );
 }
}
```

```
child: Column(
              crossAxisAlignment:
CrossAxisAlignment.center,
              children: [
               Image.asset(

'assets/images/spotify_logo.png',
```

```
ns/register.dart';

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

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   backgroundColor: const
Color(0xFF1C1B1B),
   body: SafeArea(
    child: Column(
     children: [
      Padding(
       padding: const
EdgeInsets.all(16.0),
       child: Row(
        children: [
         IconButton(
          icon: const
Icon(Icons.arrow_back, color:
Colors.white),
          onPressed: () =>
Navigator.pop(context),
         ),
        ],
       ),
      ),
      Expanded(
       child: SingleChildScrollView(
        child: Padding(
         padding: const
EdgeInsets.symmetric(horizontal: 24.0),
                                builder:
(context) =>
                    Register()), //
Navigate to ThirdScreen
                );
                // Handle register
               },
```

```
                width: //235
                    500,
                height: 150,
              ),
              const SizedBox(height: 32),
              const Text(
                'Enjoy Listening To Music',
                style: TextStyle(
                  color: Colors.white,
                  fontSize: 24,
                  fontWeight:
FontWeight.bold,
                ),
              ),
              const SizedBox(height: 16),
              const Text(
                'Spotify is a proprietary
Swedish audio\nstreaming and media
services provider',
                textAlign:
TextAlign.center,
                style: TextStyle(
                  color: Colors.grey,
                  fontSize: 16,
                ),
              ),
              const SizedBox(height: 32),
              SizedBox(
                width: double.infinity,
                height: 56,
                child: ElevatedButton(
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
style: TextStyle(
                    color: Colors.white,
                    fontSize: 16,
                  ),
                ),
              ),
            ],
                    style:
ElevatedButton.styleFrom(
                      backgroundColor: const
Color(0xFF42C83C),
                      shape:
RoundedRectangleBorder(
                        borderRadius:
BorderRadius.circular(30),
                      ),
                    ),
                    child: const Text(
                      'Register',
                      style: TextStyle(
                        color: Colors.white,
                        fontSize: 16,
                        fontWeight:
FontWeight.bold,
                      ),
                    ),
                  ),
                ),
              ),
              const SizedBox(height: 16),
              TextButton(
                onPressed: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (context) =>
                        SignInScreen()), //
Navigate to ThirdScreen
                  );
                  // Handle sign in
                },
                child: const Text(
                  'Sign in',
```

Home_screen.dart

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

// First, add this typography class at the
```

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

```
body: Stack(
      children: [
        Container(
          decoration: BoxDecoration(
            gradient: LinearGradient(
              begin: Alignment.topCenter,
              end: Alignment.bottomCenter,
```

```
top of your file
class AppTypography {
  static final TextStyle displayLarge =
GoogleFonts.poppins(
    fontSize: 24,
    fontWeight: FontWeight.bold,
    letterSpacing: -0.3,
    color: Colors.white,
  );

  static final TextStyle titleMedium =
GoogleFonts.poppins(
    fontSize: 16,
    fontWeight: FontWeight.w600,
    letterSpacing: -0.2,
    color: Colors.white,
  );

  static final TextStyle bodySmall =
GoogleFonts.poppins(
    fontSize: 12,
    fontWeight: FontWeight.normal,
    letterSpacing: 0.1,
    color: Colors.grey[400],
  );
}

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
     8),
                    decoration:
BoxDecoration(
                      color: const
Color(0xFF1DB954),
                    borderRadius:
BorderRadius.circular(20),
```

```
            colors: [
              Colors.black.withOpacity(0.8),
              Colors.black,
            ],
          ),
        ),
      ),
      SingleChildScrollView(
        child: SafeArea(
          child: Column(
            crossAxisAlignment:
CrossAxisAlignment.start,
            children: [
              Padding(
                padding: const
EdgeInsets.all(16.0),
                child: Row(
                  children: [
                    // Updated CircleAvatar
with specific size and styling
                    const CircleAvatar(
                      radius: 20,
                      backgroundColor:
Color(0xFF282828),
                      child: Icon(
                        Icons.person,
                        color: Colors.white,
                        size: 24,
                      ),
                    ),
                    const SizedBox(width: 16),
                    Container(
                      padding: const
EdgeInsets.symmetric(
                        horizontal: 16, vertical:
 child: const Text('Podcasts',
                        style: TextStyle(color:
Colors.white)),
                    ),
                  ],
                ),
              ),
                    ),
                    child: const Text('All',
                      style: TextStyle(color:
Colors.white)),
                    ),
                    const SizedBox(width: 8),
                    Container(
                      padding: const
EdgeInsets.symmetric(
                        horizontal: 16, vertical:
8),
                      decoration:
BoxDecoration(
                        color: const
Color(0xFF282828),
                        borderRadius:
BorderRadius.circular(20),
                      ),
                      child: const Text('Music',
                        style: TextStyle(color:
Colors.white)),
                    ),
                    const SizedBox(width: 8),
                    Container(
                      padding: const
EdgeInsets.symmetric(
                        horizontal: 16, vertical:
8),
                      decoration:
BoxDecoration(
                        color: const
Color(0xFF282828),
                        borderRadius:
BorderRadius.circular(20),
                      ),

                    SizedBox(width: 16),
                    MixCard(
                      title: 'Chill Mix',
                      subtitle: 'Kendrick Lamar,
The Weeknd, Future',
                      image:
```

```
              const PlaylistSection(
                title: 'On Repeat',
                image:
'assets/images/on_repeat.jpg',
                isLarge: true,
              ),
              const PlaylistSection(
                title: 'Trending Now India',
                image:
'assets/images/trending_india.jpg',
                isLarge: true,
              ),
              const PlaylistSection(
                title: 'Liked Songs',
                image:
'assets/images/liked_songs.jpg',
                isLarge: true,
              ),
              const SectionTitle(title: 'Your
top mixes'),
              SingleChildScrollView(
                scrollDirection:
Axis.horizontal,
                padding: const
EdgeInsets.symmetric(horizontal: 16),
                child: Row(
                  children: const [
                    MixCard(
                      title: 'Arijit Singh Mix',
                      subtitle: 'Pritam,
Vishal-Shekhar and Atif Aslam',
                      image:
'assets/images/arijit.jpg',
                    ),

const PlaylistSection({
   Key? key,
   required this.title,
   required this.image,
   this.isLarge = false,
  }) : super(key: key);
'assets/images/chill.jpg',
                    ),
                    SizedBox(width: 16),
                    MixCard(
                      title: 'R&B Mix',
                      subtitle: 'Kali Uchis,
SZA',
                      image:
'assets/images/rnb.jpg',
                    ),
                  ],
                ),
              ),
              const SizedBox(height: 100),
            ],
          ),
        ),
        const Positioned(
          left: 0,
          right: 0,
          bottom: 0,
          child: NowPlayingBar(),
        ),
      ],
    ),
  );
 }
}

class PlaylistSection extends
StatelessWidget {
 final String title;
 final String image;
 final bool isLarge;

              size: 24,
            ),
            onPressed: () {},
          ),
        ),
      ],
```

```
  @override
  Widget build(BuildContext context) {
   return Padding(
    padding: const EdgeInsets.all(16.0),
    child: Row(
     children: [
      ClipRRect(
       borderRadius:
BorderRadius.circular(4),
       child: Image.asset(
        image,
        width: isLarge ? 80 : 60,
        height: isLarge ? 80 : 60,
        fit: BoxFit.cover,
       ),
      ),
      const SizedBox(width: 16),
      Expanded(
       child: Text(
        title,
        style:
AppTypography.titleMedium, // Updated
typography
         overflow: TextOverflow.ellipsis,
// Added overflow handling
         maxLines: 2,
       ),
      ),
      Material(
       color: Colors.transparent,
       child: IconButton(
        icon: const Icon(
         Icons.more_vert,
         color: Colors.white,

const SizedBox(height: 8),
      Text(
       title,
       style:
AppTypography.titleMedium, // Updated
typography
       overflow: TextOverflow.ellipsis, //
      ),
     );
   }
  }

class MixCard extends StatelessWidget {
  final String title;
  final String subtitle;
  final String image;

  const MixCard({
   Key? key,
   required this.title,
   required this.subtitle,
   required this.image,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
   return SizedBox(
    width: 160,
    child: Column(
     crossAxisAlignment:
CrossAxisAlignment.start,
     children: [
      ClipRRect(
       borderRadius:
BorderRadius.circular(4),
       child: Image.asset(
        image,
        width: 160,
        height: 160,
        fit: BoxFit.cover,
       ),
      ),
                width: 40,
        height: 40,
        fit: BoxFit.cover,
       ),
      ),
      const SizedBox(width: 12),
      Expanded(
```

```
Added overflow handling
          maxLines: 1,
        ),
        const SizedBox(height: 4),
        Text(
          subtitle,
          style: AppTypography.bodySmall,
// Updated typography
          overflow: TextOverflow.ellipsis, //
Added overflow handling
          maxLines: 2,
        ),
      ],
    ),
  );
 }
}

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

 @override
 Widget build(BuildContext context) {
  return Container(
    color: const Color(0xFF282828),
    padding: const EdgeInsets.all(8),
    child: Row(
     children: [
      ClipRRect(
       borderRadius:
BorderRadius.circular(4),
       child: Image.asset(
        'assets/images/perfect_1d.jpg',
),
       Material(
        color: Colors.transparent,
        child: IconButton(
         icon: const Icon(
          Icons.favorite_border,
          color: Colors.white,
        child: Column(
         crossAxisAlignment:
CrossAxisAlignment.start,
          mainAxisSize:
MainAxisSize.min,
          children: [
           Text(
            'Perfect',
            style:
AppTypography.titleMedium, // Updated
typography
             overflow:
TextOverflow.ellipsis,
           ),
           Text(
            'One Direction',
            style:
AppTypography.bodySmall, // Updated
typography
             overflow:
TextOverflow.ellipsis,
           ),
          ],
         ),
        ),
        Material(
         color: Colors.transparent,
         child: IconButton(
          icon: const Icon(
           Icons.devices_outlined,
           color: Colors.white,
           size: 24,
          ),
          onPressed: () {},
         ),
           style:
AppTypography.displayLarge, // Updated
typography
     ),
    );
  }
}
```

```
            size: 24,
          ),
          onPressed: () {},
        ),
      ),
      Material(
        color: Colors.transparent,
        child: IconButton(
          iconSize: 32,
          icon: const Icon(
            Icons.play_circle_filled,
            color: Colors.white,
          ),
          onPressed: () {},
        ),
      ),
    ],
  ),
 );
 }
}

class SectionTitle extends StatelessWidget
{
 final String title;

 const SectionTitle({Key? key, required
this.title}) : super(key: key);

 @override
 Widget build(BuildContext context) {
  return Padding(
   padding: const EdgeInsets.all(16.0),
   child: Text(
    Title,
```

Search_screen.dart

```
import 'package:flutter/material.dart';
import
'package:flutter_riverpod/flutter_riverpod.
dart';
import
'package:spotify_clone/models/search_res
```

```
Future<void> _performSearch(String
query) async {
  if (query.isEmpty) return;

  setState(() {
   _isLoading = true;
  });
```

```
ult.dart';
import
'package:spotify_clone/services/youtube_s
ervice.dart';
import
'package:spotify_clone/providers/audio_pr
ovider.dart';
import
'package:spotify_clone/services/youtube_e
xplode_service.dart';

final youtubeServiceProvider =
Provider((ref) => YouTubeService());
final audioPlayerProvider =
StateNotifierProvider<AudioPlayerNotifie
r, AudioPlayerState>((ref) =>
AudioPlayerNotifier());

class SearchScreen extends
ConsumerStatefulWidget {
  const SearchScreen({Key? key}) :
super(key: key);

  @override
  ConsumerState<SearchScreen>
createState() => _SearchScreenState();
}

class _SearchScreenState extends
ConsumerState<SearchScreen> {
  final TextEditingController
_searchController =
TextEditingController();
  List<SearchResult> _searchResults = [];
  bool _isLoading = false;

      body: SafeArea(
      child: Column(
        children: [
          Padding(
            padding: const
EdgeInsets.all(16.0),

    final youtubeService =
ref.read(youtubeServiceProvider);
    final results = await
youtubeService.searchSongs(query);

    setState(() {
      _searchResults = results;
      _isLoading = false;
    });
  }

  Future<void> _playSong(SearchResult
song) async {
    final youtubeService =
ref.read(youtubeServiceProvider);
    final audioUrl = await
youtubeService.getAudioUrl(song.id);

    if (audioUrl != null) {
      await
ref.read(audioPlayerProvider.notifier).play
Song(audioUrl, song);
    }
  }

  @override
  Widget build(BuildContext context) {
    final audioState =
ref.watch(audioPlayerProvider);

    return Scaffold(
      backgroundColor: const
Color(0xFF1C1B1B),

_searchResults[index];
                final isPlaying =
audioState.currentSong?.id == song.id;

                return ListTile(
                  leading:
Image.network(song.thumbnail),
```

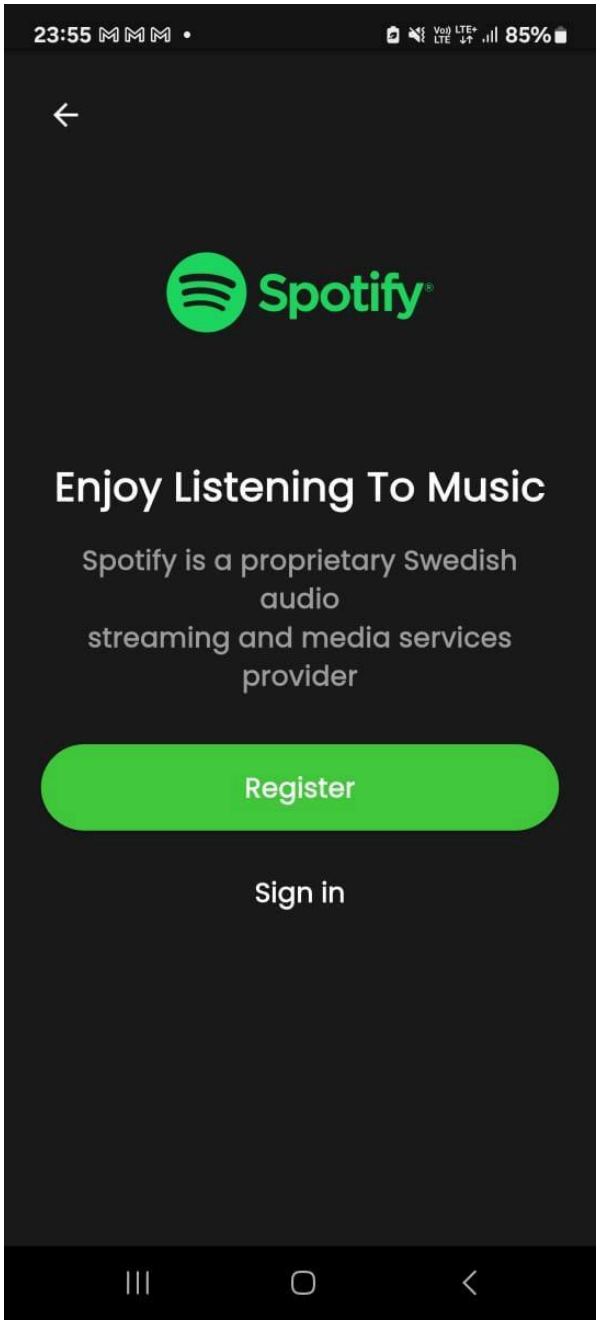
```
            child: TextField(
              controller: _searchController,
              style: const TextStyle(color:
Colors.white),
              decoration: InputDecoration(
                hintText: 'Search for songs...',
                hintStyle: const
TextStyle(color: Colors.grey),
                filled: true,
                fillColor:
Colors.grey.withOpacity(0.1),
                border: OutlineInputBorder(
                  borderRadius:
BorderRadius.circular(12),
                  borderSide: BorderSide.none,
                ),
                suffixIcon: IconButton(
                  icon: const Icon(Icons.search,
color: Colors.grey),
                  onPressed: () =>
_performSearch(_searchController.text),
                ),
              ),
              onSubmitted: _performSearch,
            ),
          ),
          if (_isLoading)
            const CircularProgressIndicator()
          else
            Expanded(
              child: ListView.builder(
                itemCount:
_searchResults.length,
                itemBuilder: (context, index) {
                  final song =
EdgeInsets.all(16),
                color: Colors.black,
                child: Row(
                  children: [
                    Image.network(

audioState.currentSong!.thumbnail,
                    title: Text(
                      song.title,
                      style: const
TextStyle(color: Colors.white),
                    ),
                    subtitle: Text(
                      song.channelTitle,
                      style: const
TextStyle(color: Colors.grey),
                    ),
                    trailing: IconButton(
                      icon: Icon(
                        isPlaying &&
audioState.isPlaying
                            ? Icons.pause
                            : Icons.play_arrow,
                        color: Colors.white,
                      ),
                      onPressed: () {
                        if (isPlaying) {

ref.read(audioPlayerProvider.notifier).togg
lePlay();
                        } else {
                          _playSong(song);
                        }
                      },
                    ),
                  );
                },
              ),
            ),
          if (audioState.currentSong != null)
            Container(
              padding: const
lePlay();
                    },
                  ),
                ],
              ),
            ),
          ],
```

```
                  height: 50,                                  ),
                ),                                           ),
                const SizedBox(width: 16),                  );
                Expanded(                                  }
                  child: Column(                          }
                    crossAxisAlignment:
CrossAxisAlignment.start,
                    children: [
                      Text(

audioState.currentSong!.title,
                        style: const
TextStyle(color: Colors.white),
                        maxLines: 1,
                        overflow:
TextOverflow.ellipsis,
                      ),
                      Text(

audioState.currentSong!.channelTitle,
                        style: const
TextStyle(color: Colors.grey),
                      ),
                    ],
                  ),
                ),
                IconButton(
                  icon: Icon(
                    audioState.isPlaying ?
Icons.pause : Icons.play_arrow,
                    color: Colors.white,
                  ),
                  onPressed: () {

ref.read(audioPlayerProvider.notifier).togg
```

OUTPUT:

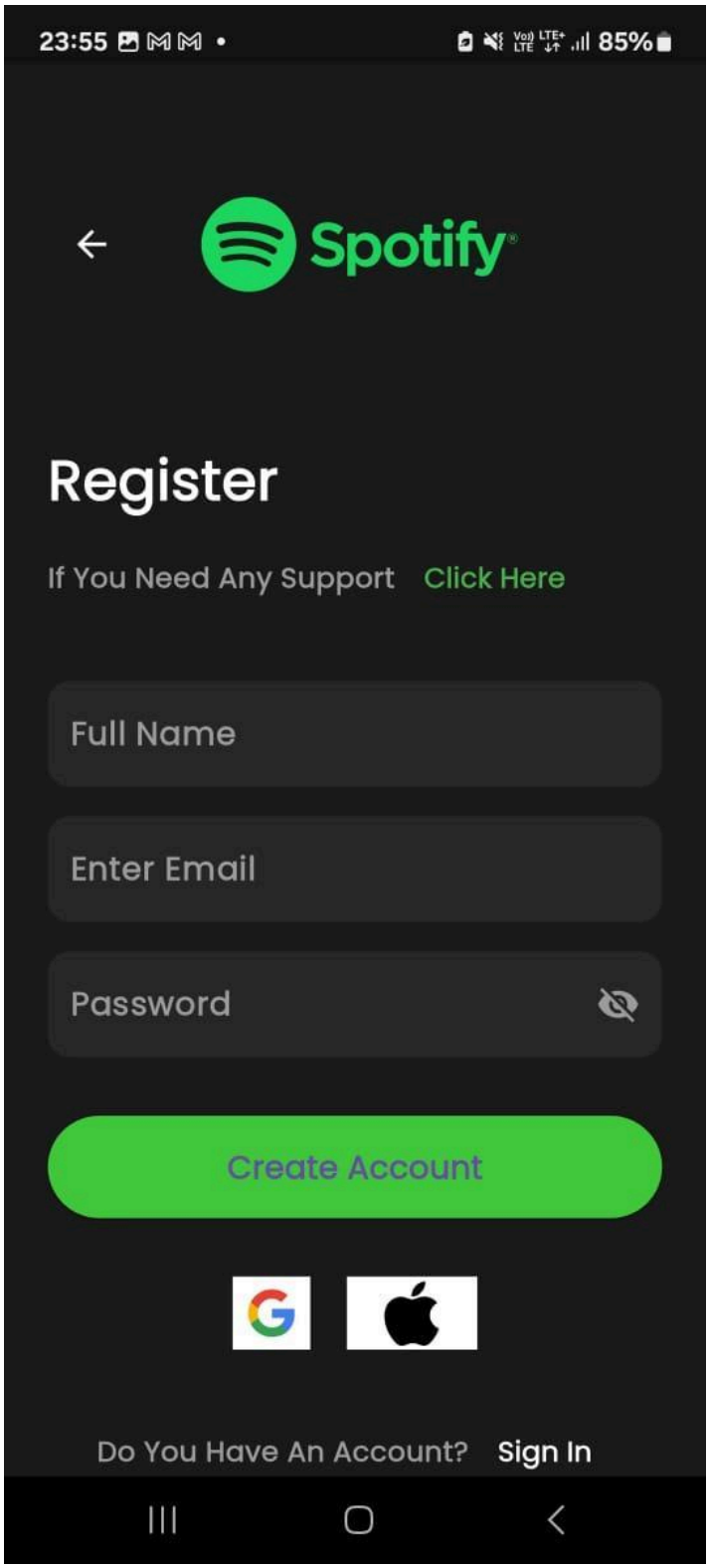
Firstly the user will be met with the splash page

After that, the register or sign in page will appear

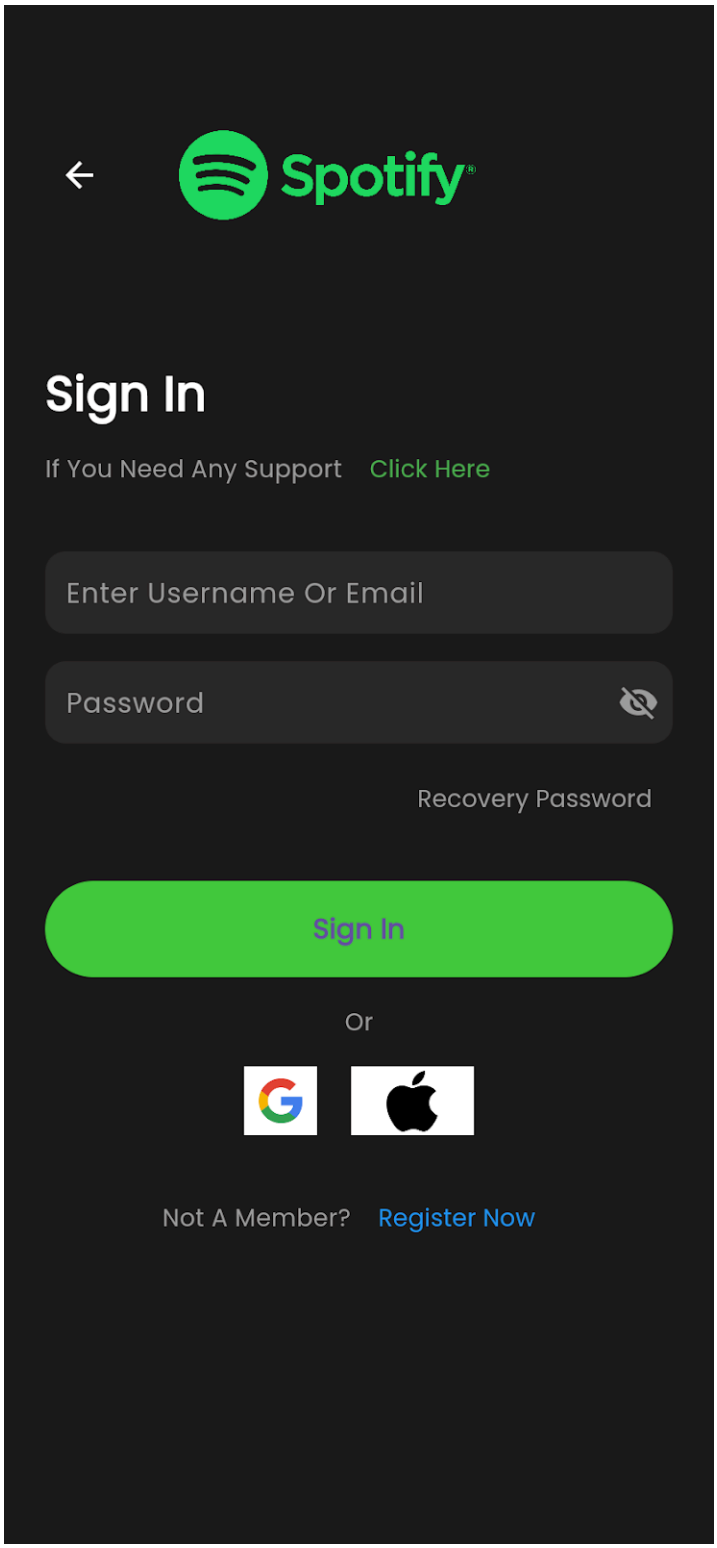


From here, the user can further navigate to either the register or sign in page

(Register)

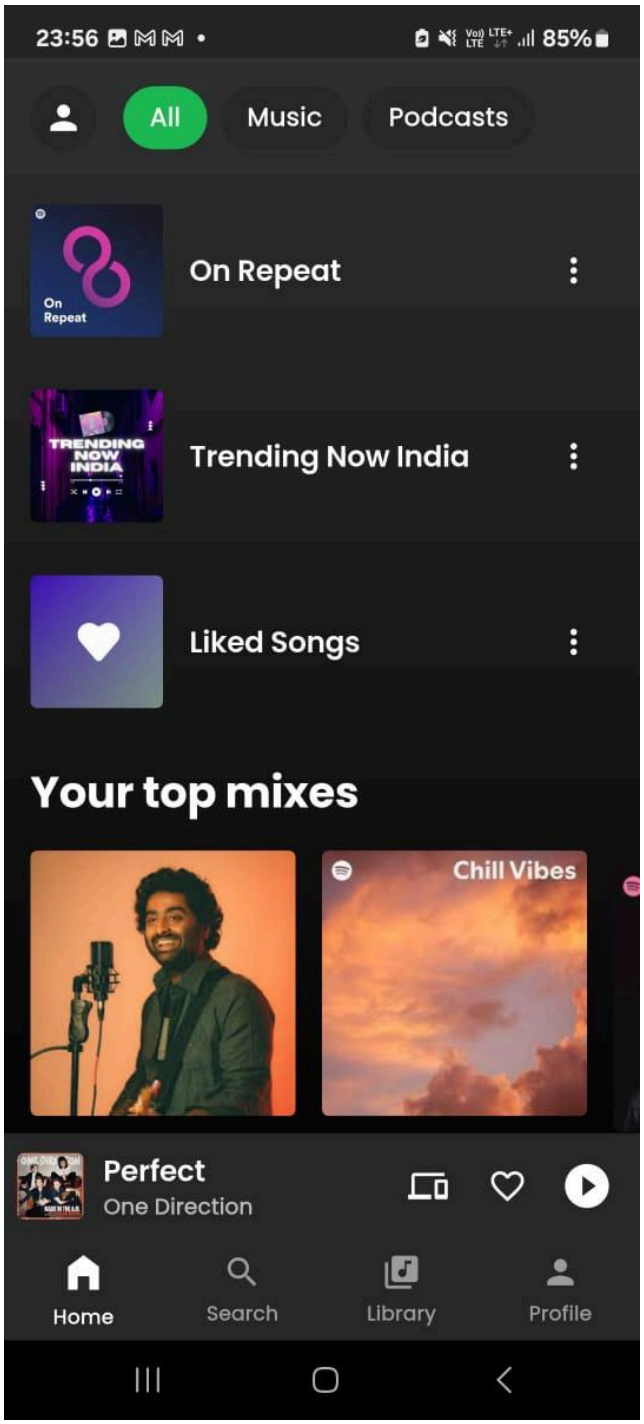

23:55
85%
Spotify
Register
If You Need Any Support Click Here
Full Name
Enter Email
Password
Create Account
Do You Have An Account? Sign In

Sign_in page


Spotify
Sign In
If You Need Any Support Click Here
Enter Username Or Email
Password
Recovery Password
Sign In
Or
Not A Member? Register Now

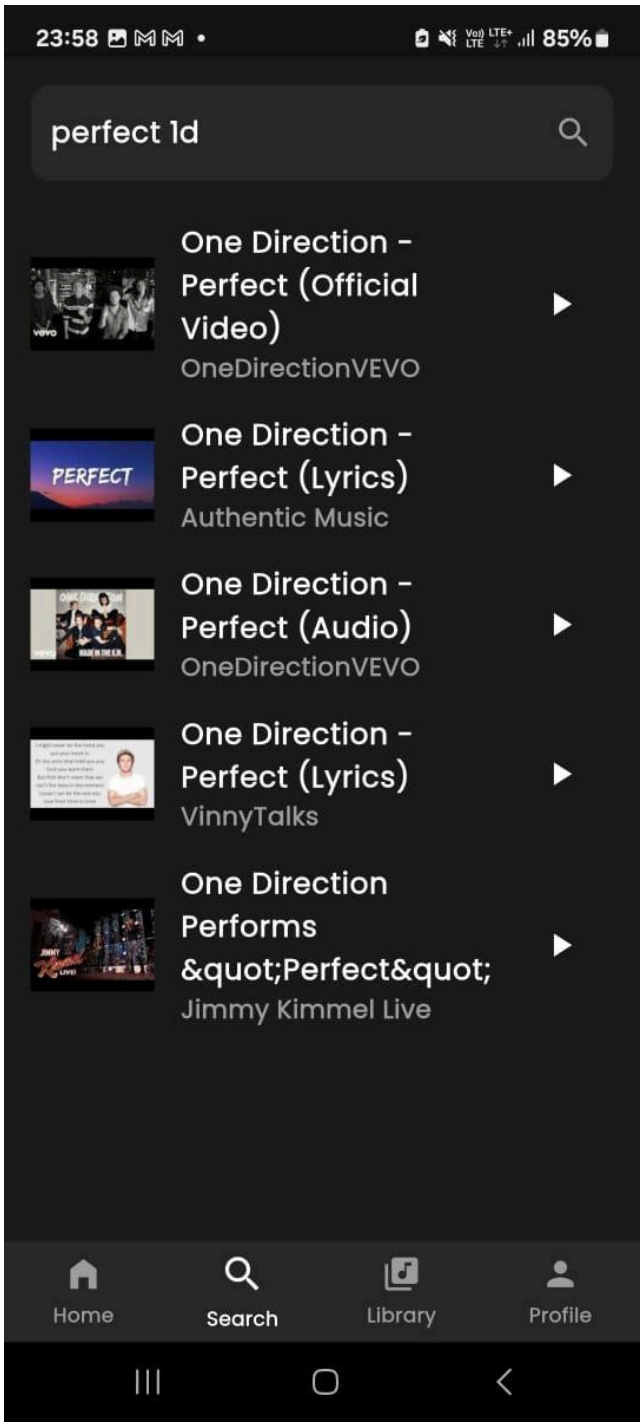
After registering/signup, the user will be navigated to the main home screen of the app

Home_page.dart



Further here we have used bottom navigation to switch between home,search and profile screens

Search_page.dart



MAD & PWA Lab
Journal

Experiment No.	06
Experiment Title.	To Connect Flutter UI with fireBase database
Roll No.	33
Name	Aryan Anil Patankar
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:	

EXPERIMENT NO.6

Name:- Aryan Patankar **Class:-** D15A **Roll:N**

AIM: - To connect Flutter UI with Firebase database.

Theory: -

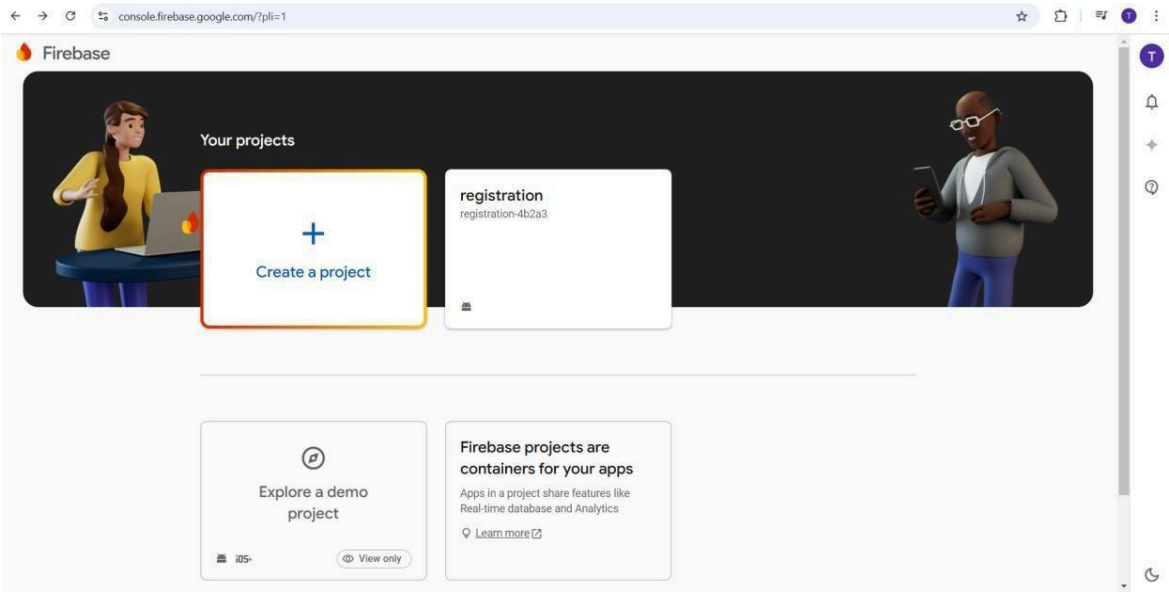
Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. Firebase, a Backend-as-a- Service (BaaS) platform, provides real-time database, authentication, and cloud storage services, making it a powerful backend solution for Flutter applications.

By integrating Firebase with Flutter, developers can store and retrieve data in real time, authenticate users, and manage cloud-based data efficiently. This is particularly useful for applications requiring dynamic content updates and user interactions.

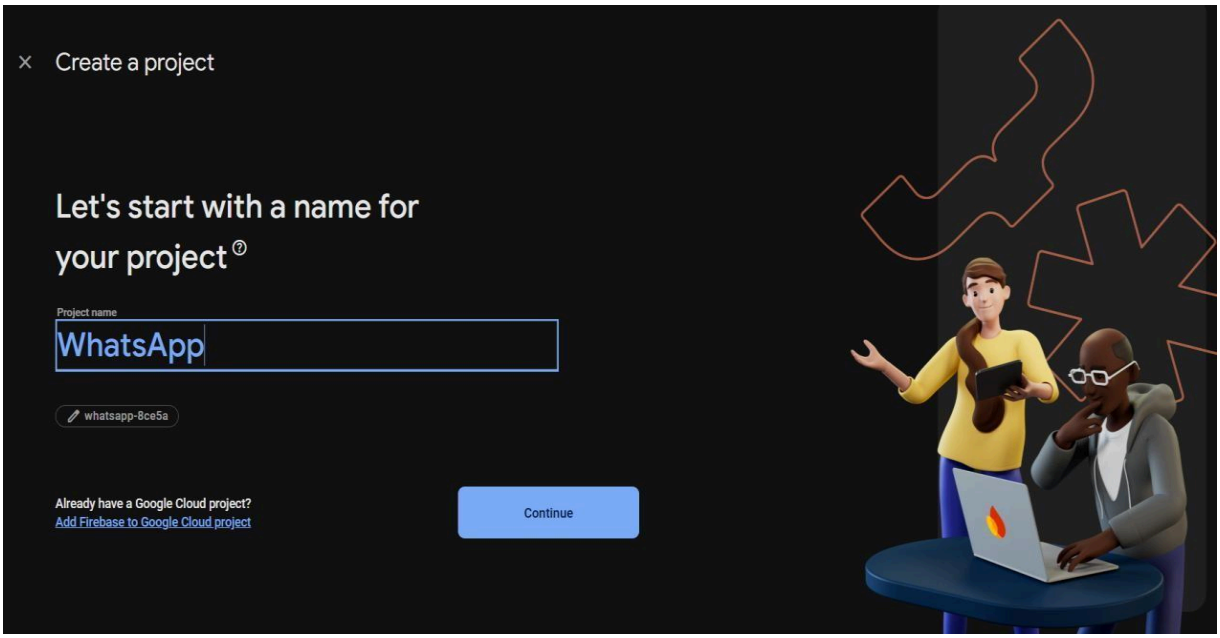
- **Steps to Connect Flutter UI with Firebase Database**

Step 1:

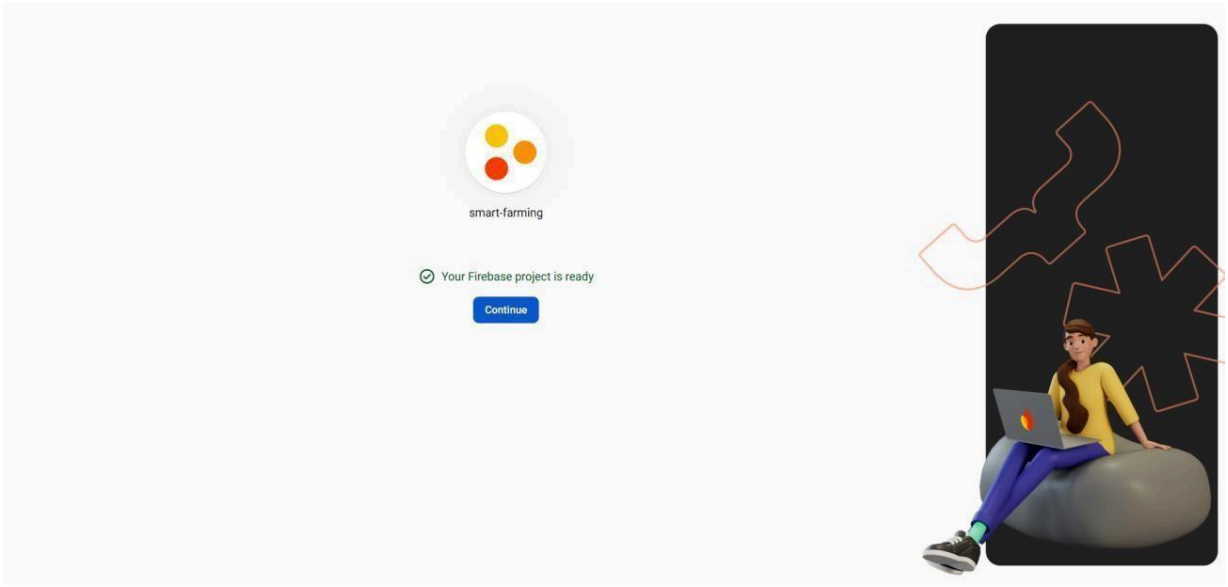
1.1) Go to Firebase Console and Create a Firebase Project



1.2) Click on Create a Project and give it a suitable name.

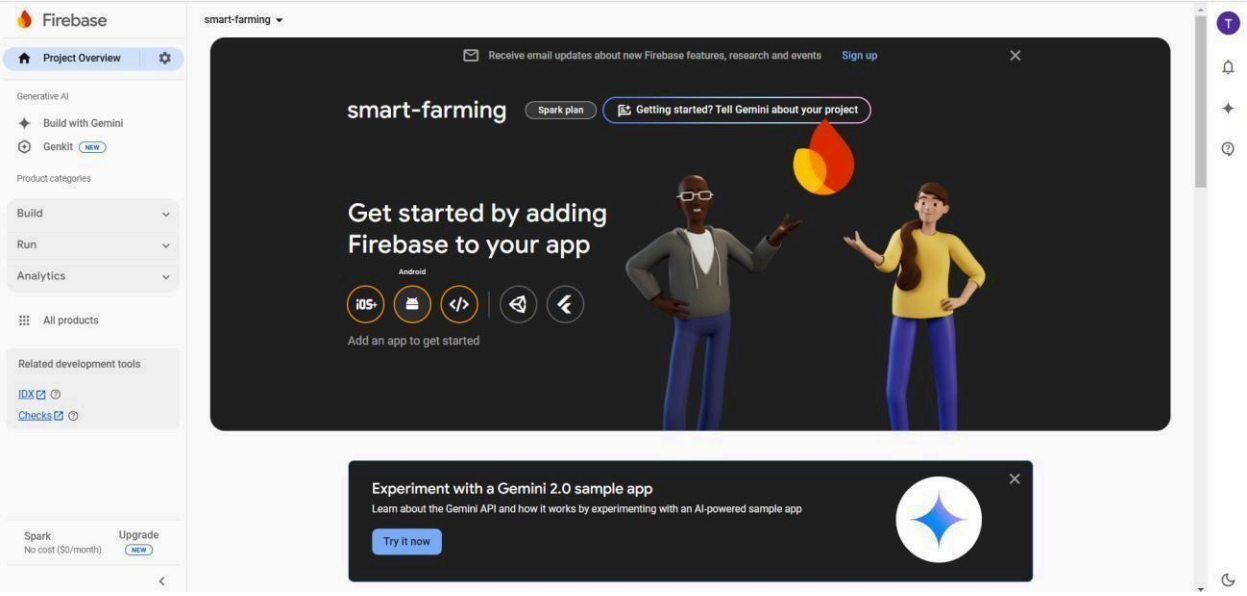


1.3) Enable Google Analytics (optional) & Click continue and complete the setup

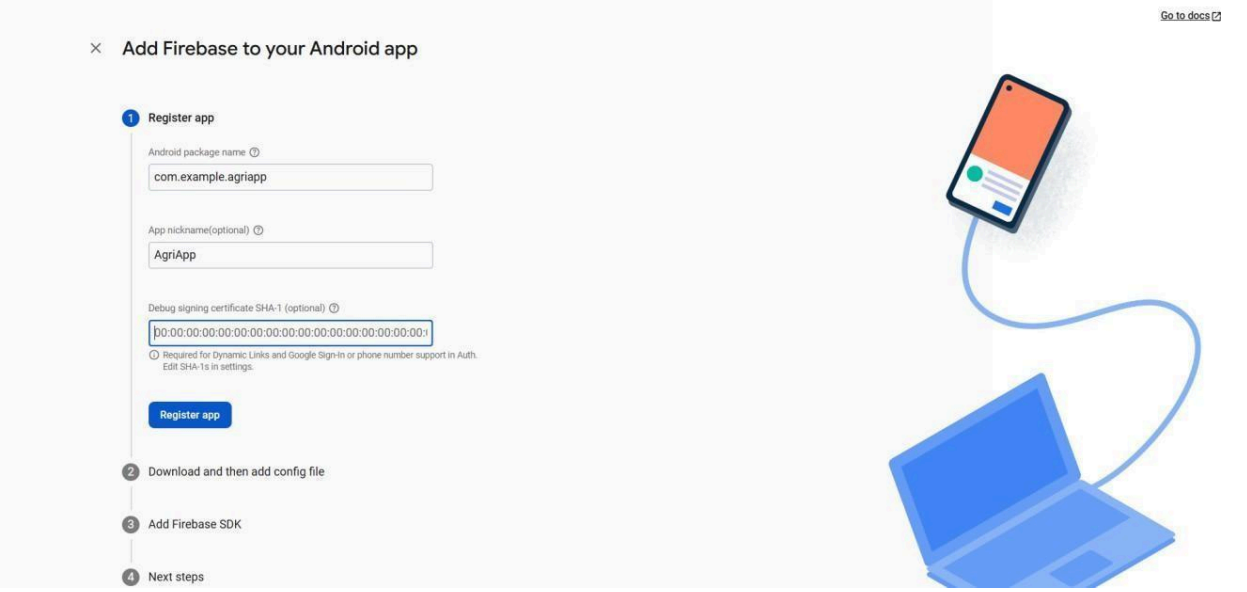


Step 2:- Add Firebase to Your Flutter App

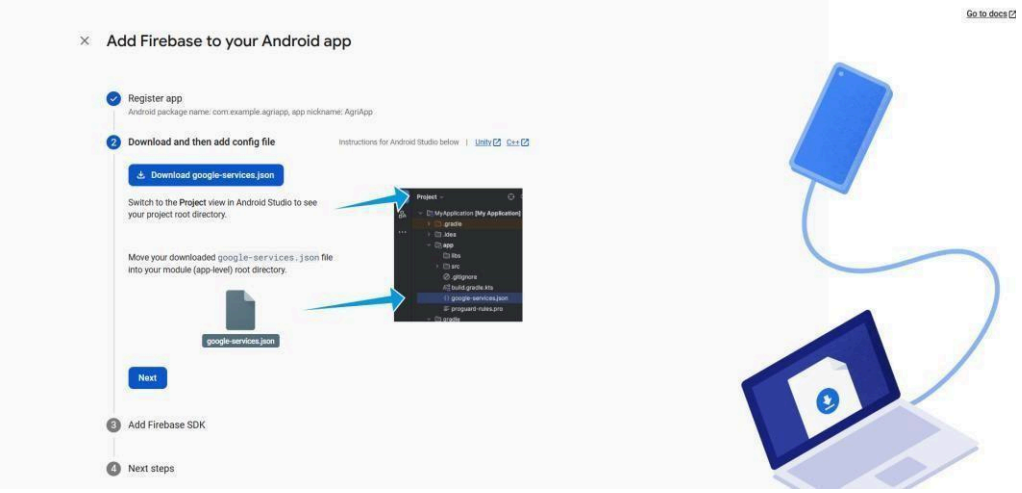
2.1) Click on Android/iOS/Web based on your Flutter application

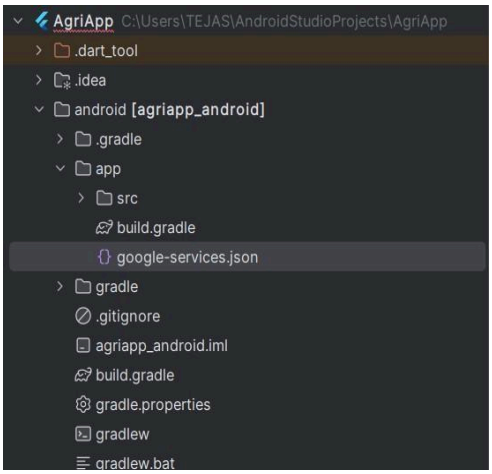


2.2) Register your app with a unique package name (found in android/app/build.gradle for Android).

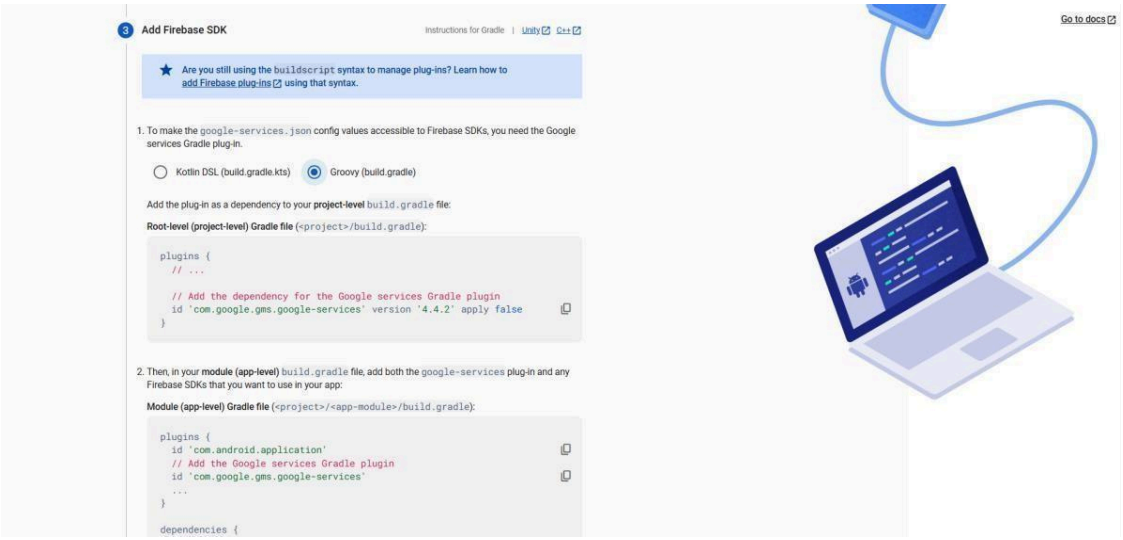


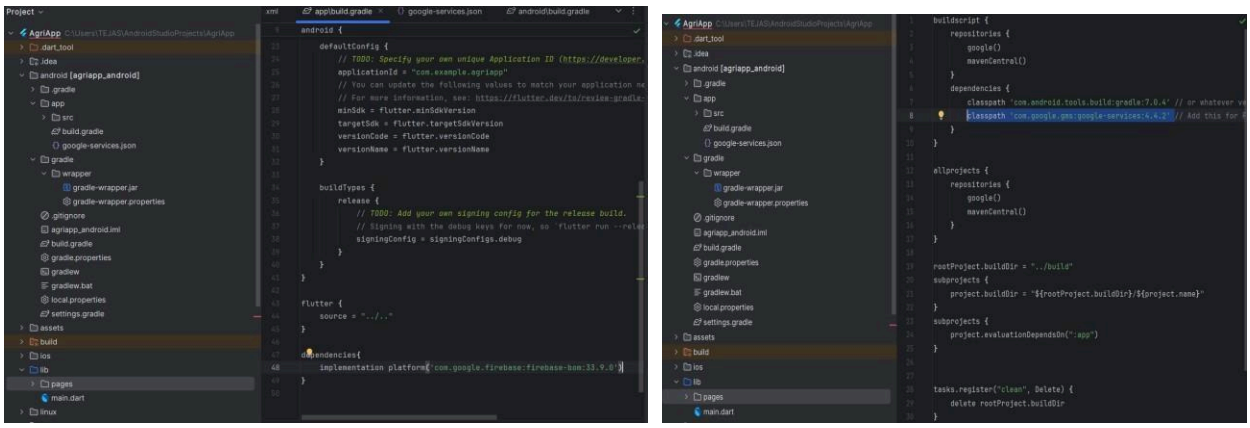
2.3) Download the google-services.json (for Android) & place the JSON file inside android/app/ directory.





2.4) Add Firebase SDK dependencies to android/build.gradle



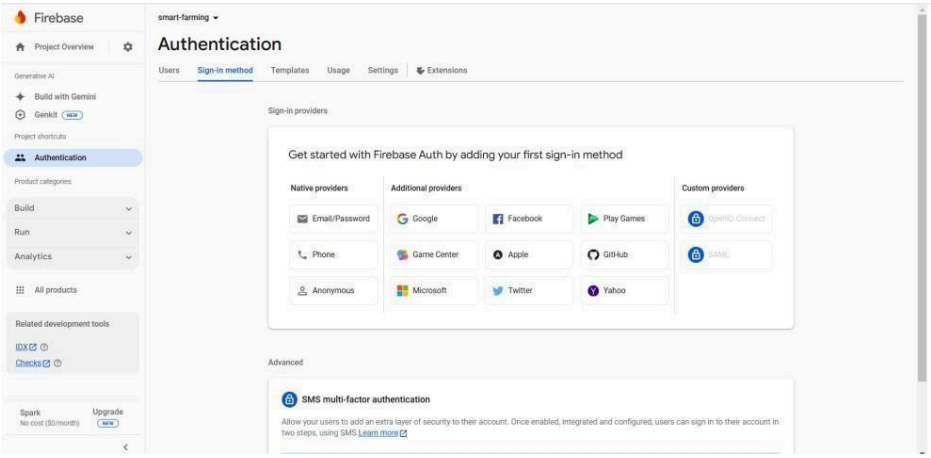


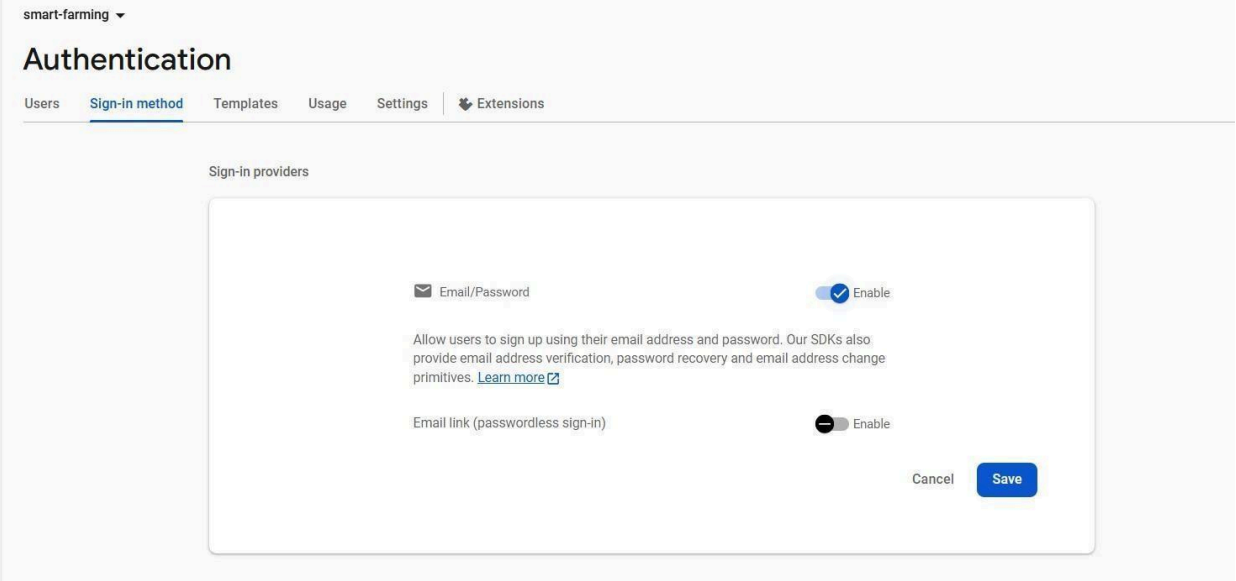
Step 3: - Add Firebase Authentication to Your App

3.1) Add Firebase Authentication Dependencies

```yaml
dependencies:
  flutter:
    sdk: flutter
  firebase_core: ^3.11.0
  firebase_auth: ^5.4.2  # For authentication
  cloud_firestore: ^5.6.3  # For Firestore, if you need it
  firebase_messaging: ^15.2.2
  http: ^0.13.3
  image_picker: ^1.0.4
  tflite_flutter: ^0.11.0
  image: ^3.2.0
  url_launcher: ^6.1.14
```

3.2) Enable Authentication in Firebase Console
Go to **Firebase Console →**
Authentication.
Click on **Sign-in method** and enable **Email/Password** (or any other method like Google).
Click Save





3.3) Implement Authentication in Flutter Modify main.dart

```
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_auth/firebase_auth.dart';

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

Step 4: -Configure Firebase Realtime Database

4.1) Go to Firebase Console → Realtime Database.
4.2) Click **Create Database** → Choose location → Set rules (for development, set read/write to true).
4.3) Click **Publish**.

Code:-

Register_signup page

```
import 'package:flutter/material.dart';
import
'package:spotify_clone/features/auth/sc
 reens/sign_in_screen.dart';
import
'package:spotify_clone/features/auth/sc
 reens/register.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const
Color(0xFF1C1B1B),
body: SafeArea(
        child: Column(
          children: [
            Padding(
              padding: const
EdgeInsets.all(16.0),
              child: Row(
                children: [
                  IconButton(
                    icon: const
Icon(Icons.arrow_back, color:
Colors.white),
                    onPressed: () =>
Navigator.pop(context),
                  ),
                ],
              ),
            ),
            Expanded(

      child: SingleChildScrollView(
              child: Padding(
                padding: const
EdgeInsets.symmetric(horizontal:
24.0),
                child: Column(
                  crossAxisAlignment:
CrossAxisAlignment.center,
children: [
                    Image.asset(

'assets/images/spotify_logo.png',
                      width: //235
                          500,
                      height: 150,
                    ),
                    const SizedBox(height:
32),
                    const Text(
                      'Enjoy Listening To
Music',
                      style: TextStyle(
                        color: Colors.white,
                        fontSize: 24,
                        fontWeight:
FontWeight.bold,
                      ),
                    ),
                    const SizedBox(height:
16),
                    const Text(
                      'Spotify is a proprietary
Swedish audio\nstreaming and media
services provider',
                      textAlign:
TextAlign.center,
                      style: TextStyle(
                        color: Colors.grey,
                        fontSize: 16,
                ),
                    ),
                    const SizedBox(height:
32),
                    SizedBox(
                      width: double.infinity,
                      height: 56,
                      child: ElevatedButton(
                        onPressed: () {
                          Navigator.push(
                            context,
                            MaterialPageRoute(
                              builder: (context)
=>
                                  Register()), //
Navigate to ThirdScreen
```

```
                                style:
ElevatedButton.styleFrom(
                  backgroundColor:
const Color(0xFF42C83C),
                  shape:
RoundedRectangleBorder(
                    borderRadius:
BorderRadius.circular(30),
                  ),
                ),
                child: const Text(
                  'Register',
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 16,
                    fontWeight:
FontWeight.bold,
                  ),
                ),
              ),

                              ),
            const SizedBox(height:
16),
            TextButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder: (context)
=>

SignInScreen()), // Navigate to
ThirdScreen
                );
                // Handle sign in
              },
              child: const Text(
                'Sign in',
                style: TextStyle(
                  color: Colors.white,
                  fontSize: 16,
                ),
              ),
            ),
          ],
        ),
      ),
    ),
                );
                // Handle register
              },
            ],
          ),
        ),
      );
    }
```

Register.dart

```
import 'package:flutter/material.dart';
import
'package:spotify_clone/features/auth/sc
 reens/sign_in_screen.dart';
import
'package:spotify_clone/main.dart';

class Register extends StatefulWidget {
  const Register({Key? key}) :
super(key: key);

  @override
  State<Register> createState() =>
_RegisterState();
}

class _RegisterState extends
State<Register> {
  bool _obscurePassword = true;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const
Color(0xFF1C1B1B),
      body: SafeArea(
        child: SingleChildScrollView(
          child: Padding(
            padding: const
EdgeInsets.all(24.0),
            child: Column(
              crossAxisAlignment:
CrossAxisAlignment.start,
              children: [
                Row(
                  children: [
                    IconButton(
                      icon: const

Icon(Icons.arrow_back, color:
Colors.white),
                      onPressed: () =>
Navigator.pop(context),
                    ),
                    const SizedBox(width: 10),
                    Image.asset(

'assets/images/spotify_logo.png',
height: 150,
                    ),
                  ],
                ),
                const SizedBox(height: 32),
                const Text(
                  'Register',
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 28,
                    fontWeight:
FontWeight.bold,
                  ),
                ),
                const SizedBox(height: 8),
                Row(
                  children: [
                    const Text(
                      'If You Need Any Support
',
                      style: TextStyle(color:
Colors.grey),
                    ),
                    TextButton(
                      onPressed: () {},
                      child: const Text(
                        'Click Here',
                        style: TextStyle(color:
Colors.green),
  borderRadius:
),
                    ),
                  ],
                ),
                const SizedBox(height: 32),
                TextField(
                  style: const TextStyle(color:
Colors.white),
                  decoration:
InputDecoration(
                    hintText: 'Full Name',
                    hintStyle: const
TextStyle(color: Colors.grey),
                    filled: true,
                    fillColor:
Colors.grey.withOpacity(0.1),
                    border:
OutlineInputBorder(
                      borderRadius:
```

```
        BorderRadius.circular(12),

                  borderSide:
BorderSide.none,
                ),
              ),
            ),
            const SizedBox(height: 16),
            TextField(
              style: const TextStyle(color:
Colors.white),
              decoration:
InputDecoration(
                hintText: 'Enter Email',
                hintStyle: const
TextStyle(color: Colors.grey),
                filled: true,
                fillColor:
Colors.grey.withOpacity(0.1),
                border:
OutlineInputBorder(

BorderRadius.circular(12),
                  borderSide:
BorderSide.none,
                ),
              ),
            ),
            const SizedBox(height: 16),
            TextField(
              obscureText:
_obscurePassword,
              style: const TextStyle(color:
Colors.white),
              decoration:
InputDecoration(
                hintText: 'Password',
                hintStyle: const
TextStyle(color: Colors.grey),
                filled: true,
                fillColor:
Colors.grey.withOpacity(0.1),
                border:
OutlineInputBorder(
                  borderRadius:
BorderRadius.circular(12),
                  borderSide:
BorderSide.none,
                ),
                suffixIcon: IconButton(
 icon: Icon(
                    _obscurePassword
                      ? Icons.visibility_off
                      : Icons.visibility,
                    color: Colors.grey,
                  ),
                  onPressed: () {
                    setState(() {
                      _obscurePassword =
!_obscurePassword;
Row(
                                        });
                  },
                ),
              ),
            ),
            const SizedBox(height: 32),
            SizedBox(
              width: double.infinity,
              height: 56,
              child: ElevatedButton(
                onPressed: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (context) =>
const MainScreen()),
                  );
                },
                style:
ElevatedButton.styleFrom(
                  backgroundColor: const
Color(0xFF42C83C),
                  shape:
RoundedRectangleBorder(
                    borderRadius:
BorderRadius.circular(30),
                  ),
                ),
                child: const Text(
                  'Create Account',
                  style: TextStyle(
                    fontSize: 16,
                    fontWeight:
FontWeight.bold,
                  ),
                ),
              ),
            ),
            const SizedBox(height: 32),
```

```
            mainAxisAlignment:
MainAxisAlignment.center,
            children: [

Image.asset('assets/images/google_logo
 .png', height: 40),
              const SizedBox(width: 20),

Image.asset('assets/images/apple_logo.
 png', height: 40),
             ],
           ),
           const SizedBox(height: 32),
           Row(
            mainAxisAlignment:
MainAxisAlignment.center,
            children: [
             const Text(
              'Do You Have An
Account? ',
               style: TextStyle(color:
Colors.grey),
              ),
              TextButton(
               onPressed: () {
                Navigator.push(
                 context,
                 MaterialPageRoute(
                    builder: (context) =>
const SignInScreen()),
                 );
               },
               child: const Text(
                'Sign In',
                style: TextStyle(color:
Colors.white),
               ),
Sign_in_screen.dart

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

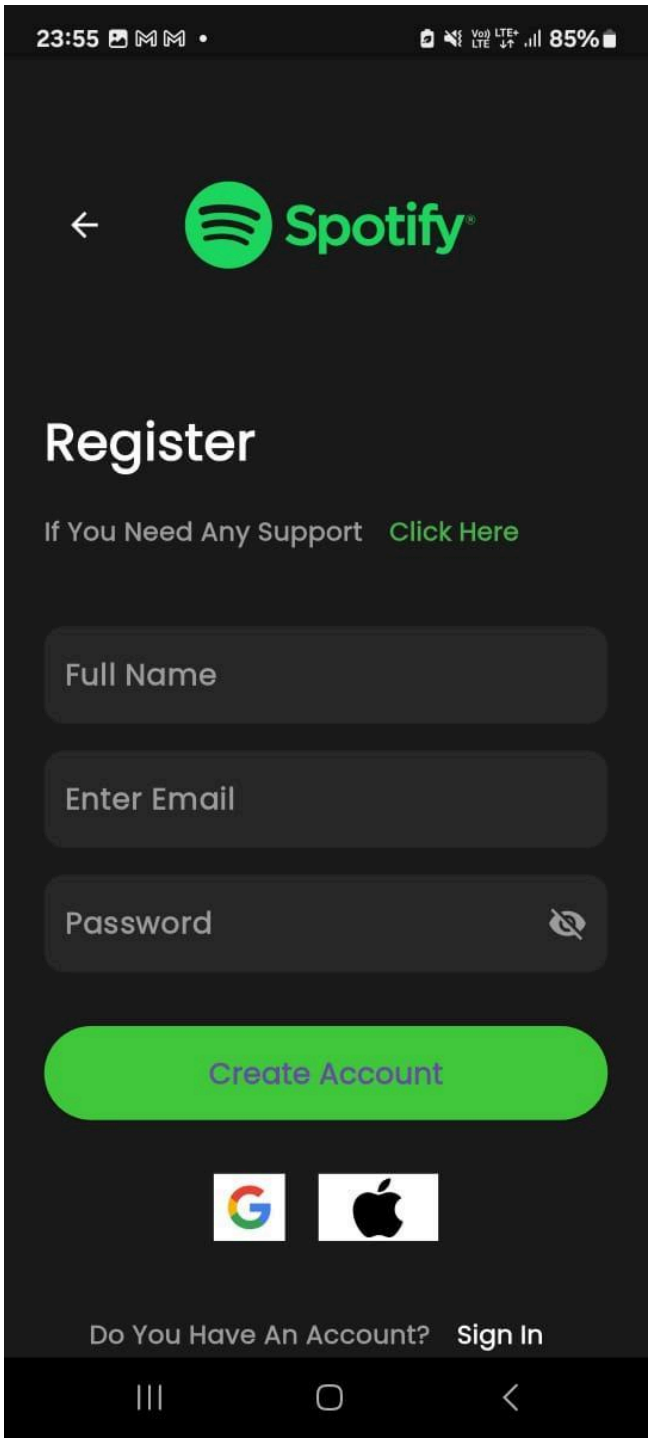
Register.dart

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

import 'register.dart';

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

 @override
 State<SignInScreen> createState() =>
_SignInScreenState();
}

class _SignInScreenState extends
State<SignInScreen> {
 bool _obscurePassword = true;

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   backgroundColor: const
Color(0xFF1C1B1B),
   body: SafeArea(
    child: SingleChildScrollView(
     child: Padding(
      padding: const
EdgeInsets.all(24.0),
      child: Column(
       crossAxisAlignment:
CrossAxisAlignment.start,
       children: [
        Row(
         children: [
          IconButton(
           Colors.green),
          ),
 icon: const Icon(Icons.arrow_back,
color: Colors.white),
           onPressed: () =>
Navigator.pop(context),
          ),
          const SizedBox(width: 12),
```

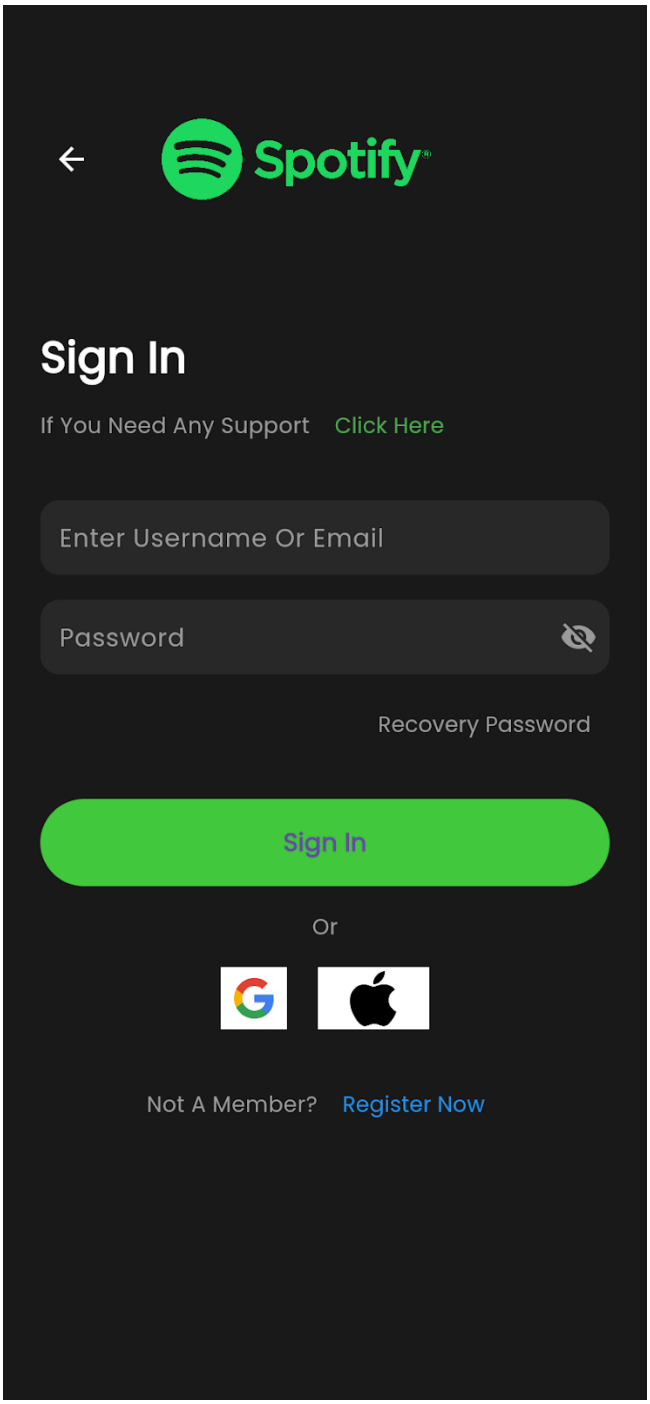
```
'assets/images/spotify_logo.png',
                  height: 150,
                ),
              ],
            ),
            const SizedBox(height: 32),
            const Text(
              'Sign In',
              style: TextStyle(
                color: Colors.white,
                fontSize: 28,
                fontWeight:
FontWeight.bold,
              ),
            ),
            const SizedBox(height: 8),
            Row(
              children: [
                const Text(
                  'If You Need Any Support
',
                  style: TextStyle(color:
Colors.grey),
                ),
                TextButton(
                  onPressed: () {},
                  child: const Text(
                    'Click Here',
                    style: TextStyle(color:

fillColor: Colors.grey.withOpacity(0.1),

                ),
              ],
            ),
            const SizedBox(height: 32),
            TextField(
              style: const TextStyle(color:
Colors.white),
              decoration:
InputDecoration(
                hintText: 'Enter Username
Or Email',
                hintStyle: const
TextStyle(color: Colors.grey),
                filled: true,
                fillColor:
Colors.grey.withOpacity(0.1),
                border:
                    Image.asset(
OutlineInputBorder(
                  borderRadius:
BorderRadius.circular(12),
                  borderSide:
BorderSide.none,
                ),
              ),
            ),
            const SizedBox(height: 16),
            TextField(
              obscureText:
_obscurePassword,
              style: const TextStyle(color:
Colors.white),
              decoration:
InputDecoration(
                hintText: 'Password',
                hintStyle: const
TextStyle(color: Colors.grey),
                filled: true,
                          SizedBox(
              width: double.infinity,
                border:
OutlineInputBorder(
                  borderRadius:
BorderRadius.circular(12),
                  borderSide:
BorderSide.none,
                ),
                suffixIcon: IconButton(
                  icon: Icon(
                    _obscurePassword
                        ? Icons.visibility_off
                        : Icons.visibility,
                    color: Colors.grey,
                  ),
                  onPressed: () {
                    setState(() {
                      _obscurePassword =
!_obscurePassword;
                    });
                  },
                ),
              ),
            ),
            const SizedBox(height: 16),
            Align(
              alignment:
```

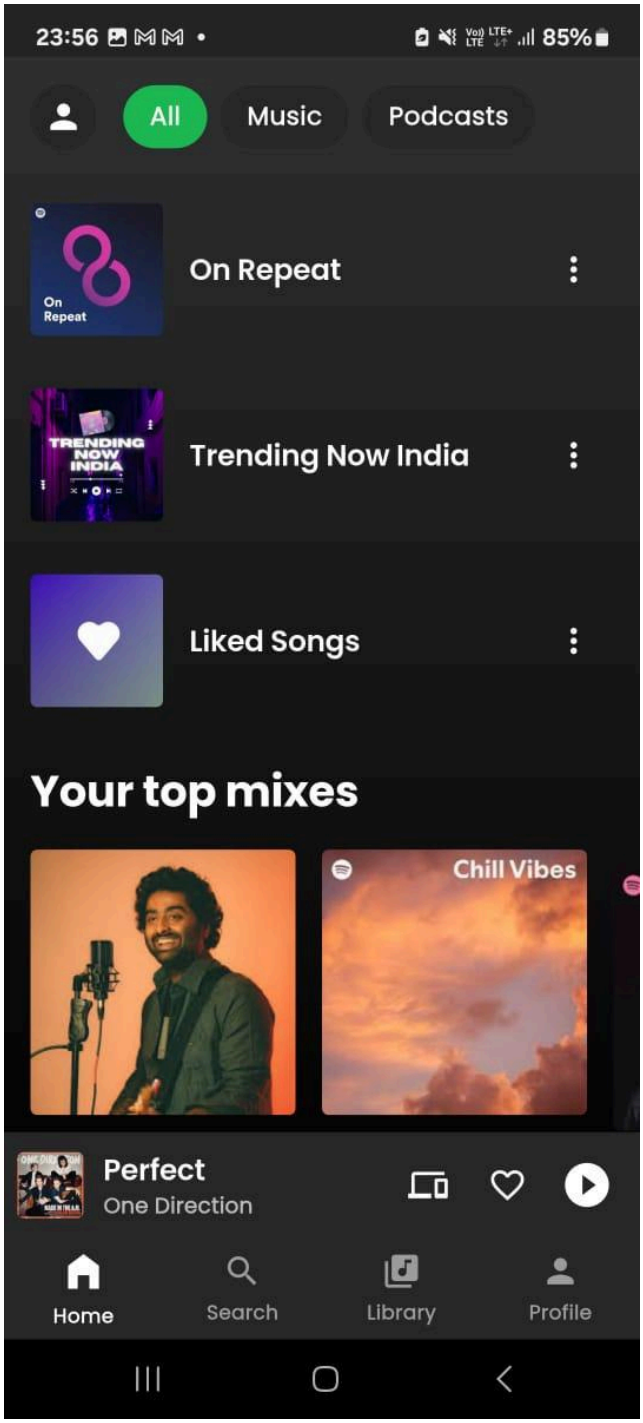
```
Alignment.centerRight,
                child: TextButton(
                  onPressed: () {},
                  child: const Text(
                    'Recovery Password',
                    style: TextStyle(color:
Colors.grey),
                  ),
                ),
              ),
              const SizedBox(height: 32),

),

                height: 56,
                child: ElevatedButton(
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                          builder: (context) =>
const MainScreen()),
                    );
                  },
                  style:
ElevatedButton.styleFrom(
                    backgroundColor: const
Color(0xFF42C83C),
                    shape:
RoundedRectangleBorder(
                      borderRadius:
BorderRadius.circular(30),
                    ),
                  ),
                  child: const Text(
                    'Sign In',
                    style: TextStyle(
                      fontSize: 16,
                      fontWeight:
FontWeight.bold,
                    ),
                  ),
                ),
              ),
              const SizedBox(height: 16),
              const Center(
                child: Text(
                  'Or',
style: TextStyle(color:
Colors.grey),
                ),
                          ),
                ),
              const SizedBox(height: 16),
              Row(
                mainAxisAlignment:
MainAxisAlignment.center,
                children: [

Image.asset('assets/images/google_logo
.png', height: 40),
                  const SizedBox(width: 20),

Image.asset('assets/images/apple_logo.
png', height: 40),
                ],
              ),
              const SizedBox(height: 32),
              Row(
                mainAxisAlignment:
MainAxisAlignment.center,
                children: [
                  const Text(
                    'Not A Member? ',
                    style: TextStyle(color:
Colors.grey),
                  ),
                  TextButton(
                    onPressed: () {
                      Navigator.push(
                        context,
                        MaterialPageRoute(
                            builder: (context) =>
const Register()),
                      );
                    },
                    child: const Text(
                      'Register Now',
                      style: TextStyle(color:
Colors.blue),


                  ],
                ),
              ],
            ),
          ),
        ),
```

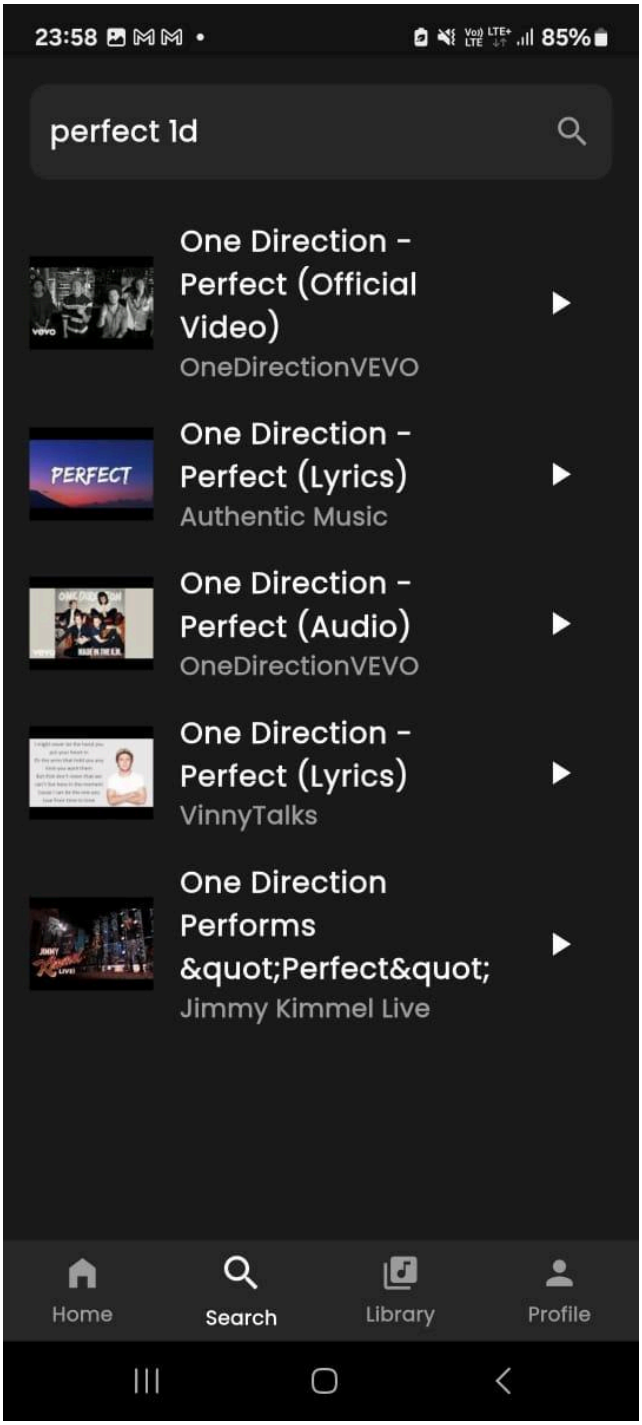
```
                ),
    );
  }
}
```

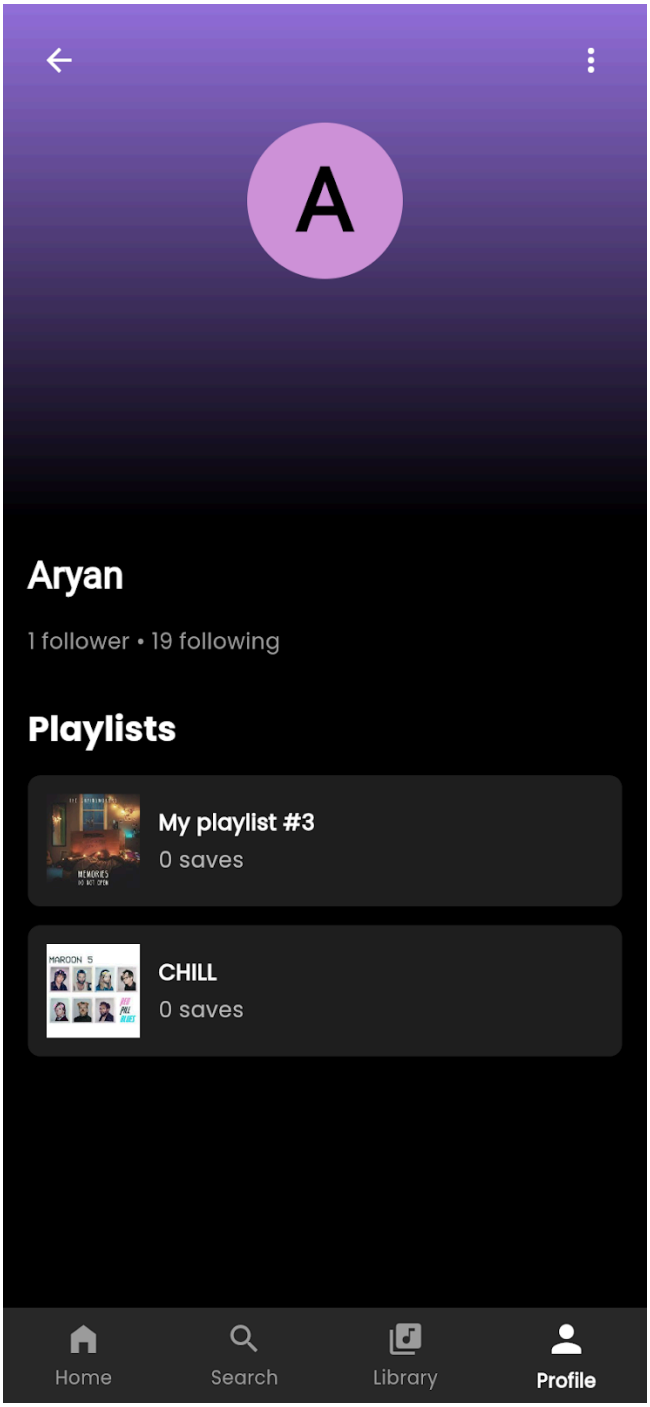

23:55
85%
Spotify
Register
If You Need Any Support
Click Here
Full Name
Enter Email
Password
Create Account
Do You Have An Account?
Sign In

Project Title: Spotify **Roll No. 33**

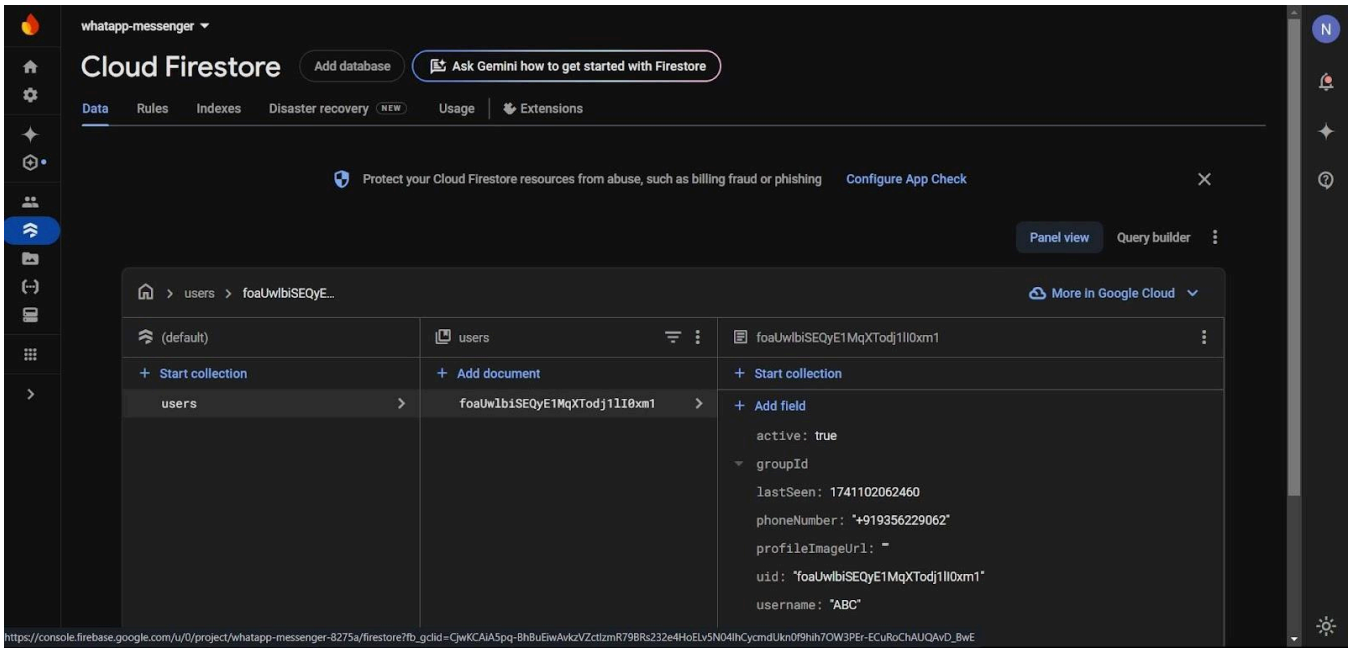



23:56
85%
All
Music
Podcasts
On Repeat
Trending Now India
Liked Songs
Your top mixes
Chill Vibes
Perfect
One Direction
Home
Search
Library
Profile


23:58
85%
perfect 1d
One Direction - Perfect (Official Video)
OneDirectionVEVO
One Direction - Perfect (Lyrics)
Authentic Music
One Direction - Perfect (Audio)
OneDirectionVEVO
One Direction - Perfect (Lyrics)
VinnyTalks
One Direction Performs "Perfect"
Jimmy Kimmel Live
Home
Search
Library
Profile


A
Aryan
1 follower • 19 following
Playlists
My playlist #3
0 saves
CHILL
0 saves
Home
Search
Library
Profile

User details get fetched from the database in the profile page.



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.	33
Name	Aryan Anil Patankar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO4: Understand various PWA frameworks and their requirements
Grade:	

Experiment No. 7

Aryan Patankar
D15A - 33

Aim: To write meta data of your Ecommerce PWA in a Web app manifest file to enable "add to homescreen feature".

Online Reference:
https://developer.mozilla.org/en-US/docs/Web/Manifest
https://www.geeksforgeeks.org/making-a-simple-pwa-under-5-minutes/

Theory:

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

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

Difference between PWAs vs. Regular Web Apps:
A Progressive Web is different and better than a Regular Web app with features like:

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

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

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

4. Engaging Approach
As already shared, the PWAs can employ push notifications and other native device features more efficiently. Their interaction does not depend on the browser user uses. This eventually improves the chances of notifying the user regarding your services, offers, and other options related to your brand and keeping them hooked to your brand. In simpler words, PWAs let you maintain the user engagement and retention rate.

5. Updated Real-Time Data Access
Another plus point of PWAs is that these apps get updated on their own. They do not demand the end-users to go to the App Store or other such platforms to download the update and wait until installed.
In this app type, the web app developers can push the live update from the server, which reaches the apps residing on the user's devices automatically. Therefore, it is easier for the mobile app developer to provide the best of the updated functionalities and services to the end-users without forcing them to update their app.

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

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

Pros and cons 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.

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

Weaknesses refer to:

- IOS support from version 11.3 onwards;
- Greater use of the device battery;
- Not all devices support the full range of PWA features (same speech for iOS and Android operating systems);
- It is not possible to establish a strong re-engagement for iOS users (URL scheme, standard web notifications);
- Support for offline execution is however limited;
- Lack of presence on the stores (there is no possibility to acquire traffic from that channel);
- There is no "body" of control (like the stores) and an approval process;
- Limited access to some hardware components of the devices;
- Little flexibility regarding "special" content for users (eg loyalty programs, loyalty, etc.).

Code:

manifest.json:-

```
{
  "name": "Xtrack",
  "short_name": "Xtrack",
  "start_url": "ani.html",
  "scope": "./",
  "icons": [
    {
     "src": "contract.png",
     "sizes": "192x192",
     "type": "image/png"
    },
    {
```

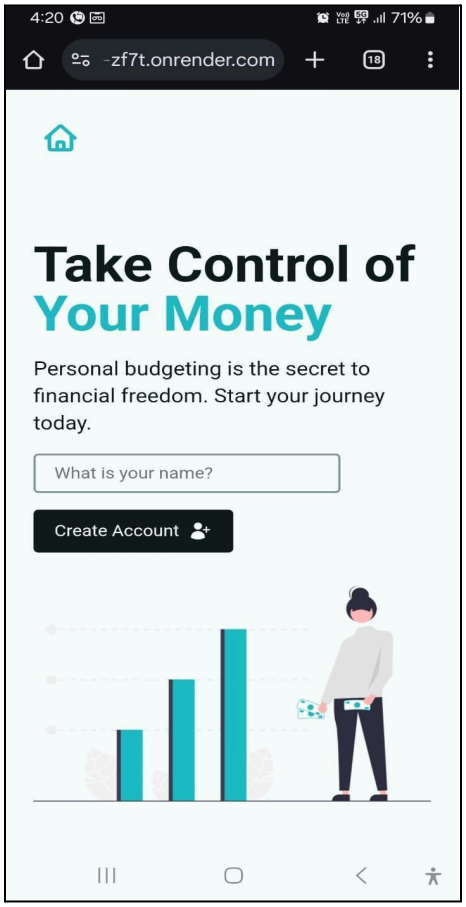
```
        "src": "contract.png",
        "sizes": "512x512",
        "type": "image/png"
       }
     ],
     "theme_color": "#ffd31d",
   "background_color": "#333",
   "display": "standalone"
 }
```

Add the link tag to link to the manifest.json file

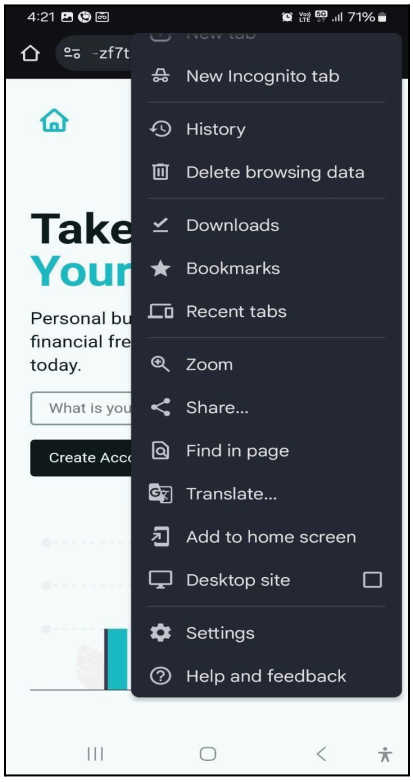
```
<html>
   <head>
      <link rel="manifest" href="manifest.json">
      <script src="myscript.js"></script>
      <title>Xtrack</title>
      <style>
```

Output with Steps:

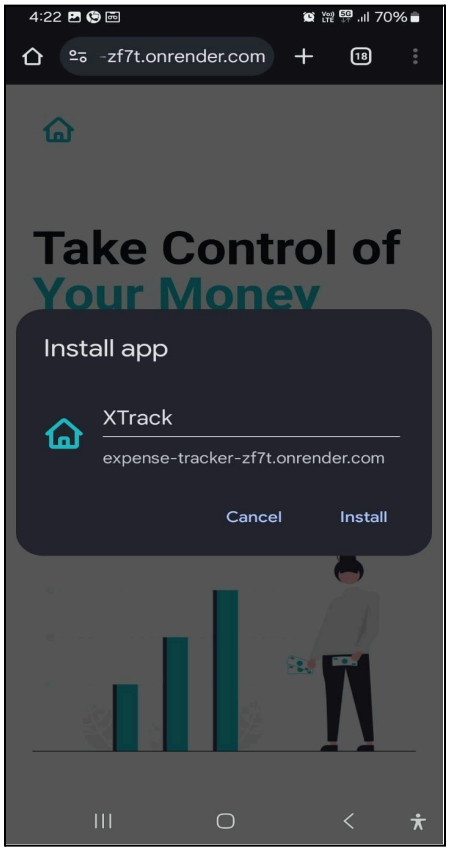
1. Go to google chrome and open your website link:



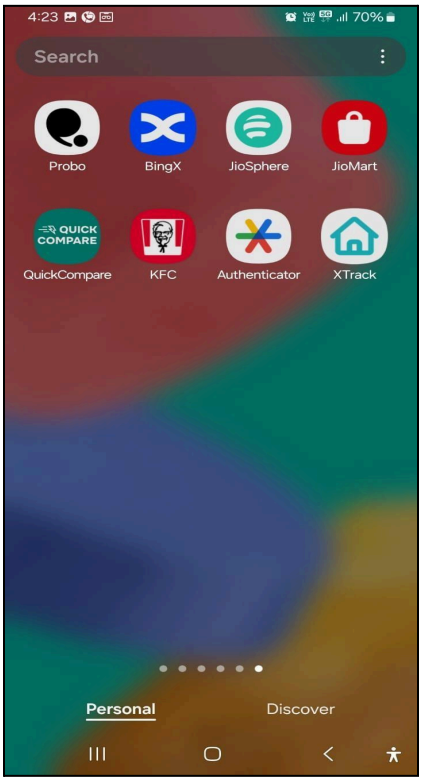
2. Then click on three dots at top right corner, find and click on add to home screen:



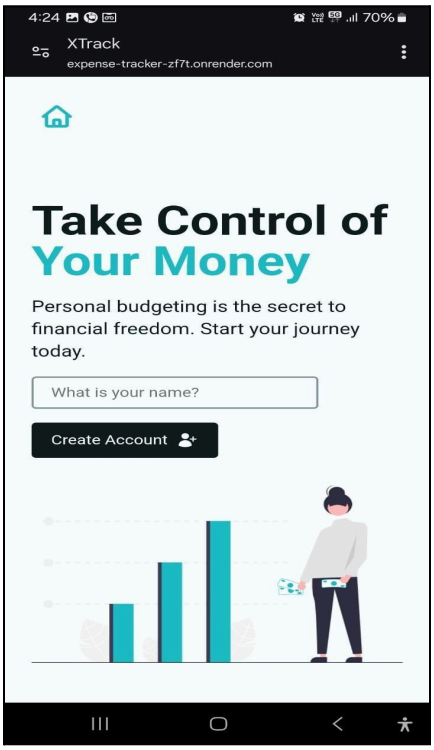
3. Then click on install:



4. Once installed you will see application in your mobile:



5. Final Output:



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.	33
Name	Aryan Anil Patankar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

Experiment No. 8

Name-Aryan Patankar
Class-D15A

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

Theory:

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

Things to note about Service Worker:

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

What can we do with Service Workers?

- You can dominate **Network Traffic**
  You can manage all network traffic of the page and do any

manipulations. For example, when the page requests a CSS file, you can send plain text as a response or when the page requests an HTML file, you can send a png file as a response. You can also send a true response too.

- You can **Cache**
  You can cache any request/response pair with Service Worker and Cache API and you can access these offline content anytime.

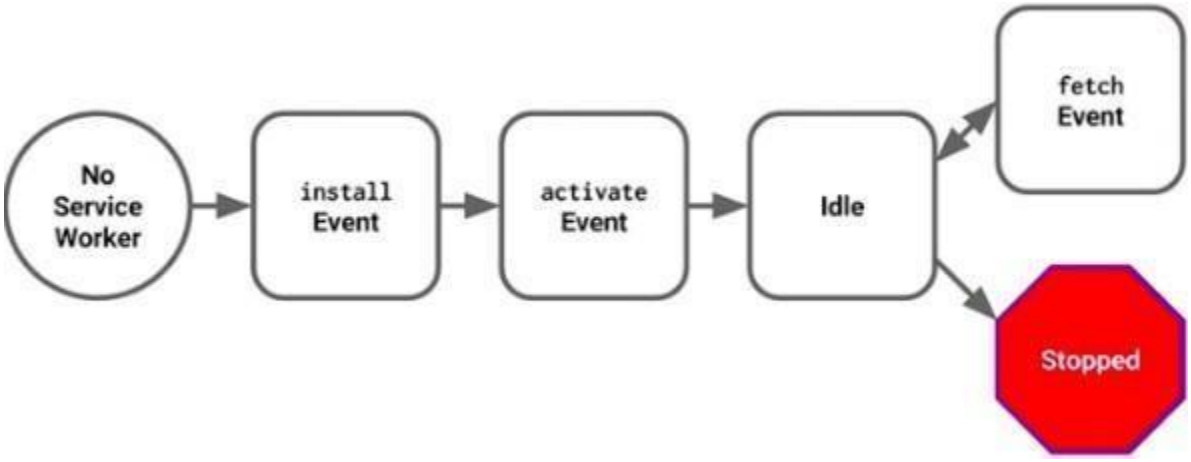
- You can manage **Push Notifications**
  You can manage push notifications with Service Worker and show any information message to the user.
- You can **Continue**
  Although Internet connection is broken, you can start any process with Background Sync of Service Worker.

What can't we do with Service Workers?

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

- You can't work it on **80 Port**
  Service Worker just can work on HTTPS protocol. But you can work on localhost during development.

Service Worker Cycle



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

- Registration
- Installation
- Activation

Registration
To install a service worker, you need to register it in your main JavaScript code.

Registration tells the browser where your service worker is located, and to start installing it in the background. Let's look at an example:

main.js

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

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

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

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

example: main.js

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

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

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

main.js

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

Installation

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

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

service-worker.js

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

Activation

Once a service worker has successfully installed, it transitions into the activation stage. If there are any open pages controlled by the previous service worker, the new service worker enters a waiting state. The new service worker only activates when there are no longer any pages loaded that are still using the old service worker. This ensures that only one version of the service worker is running at any given time.

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

service-worker.js

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

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

Code

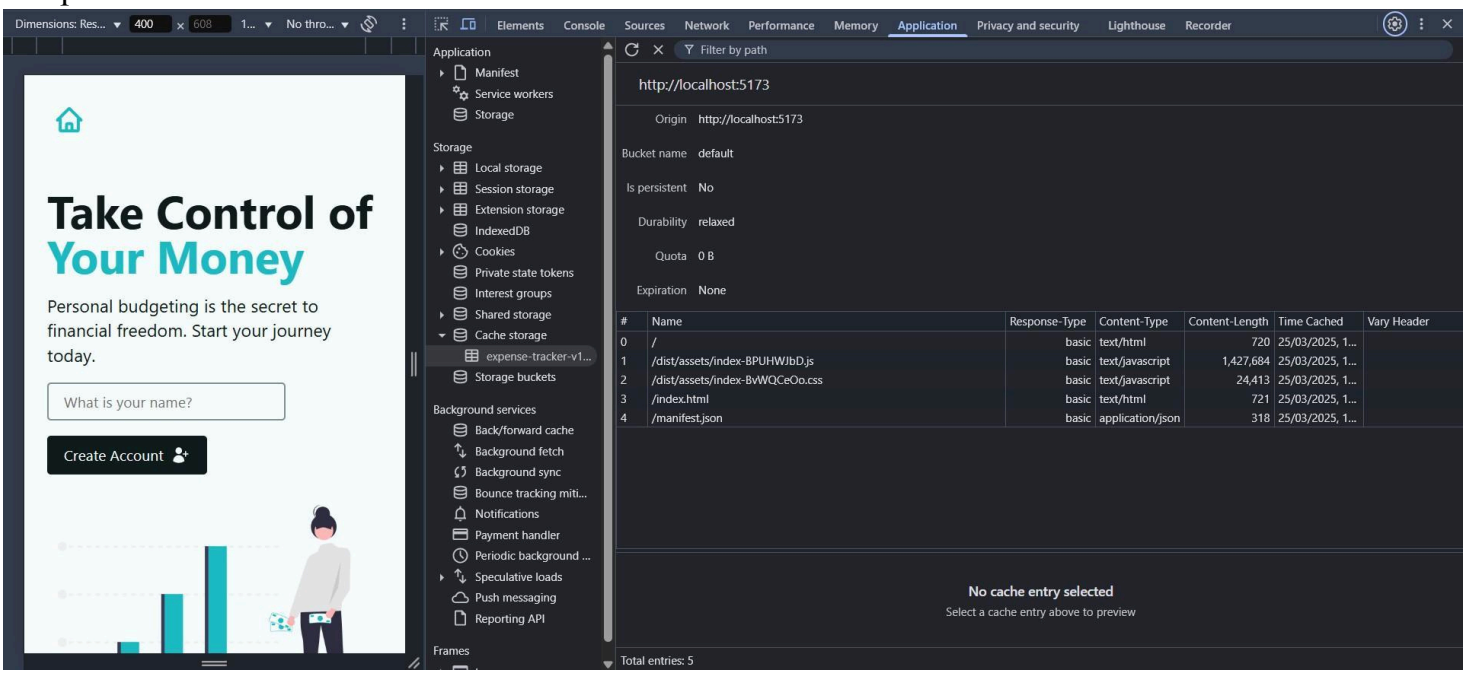
sw.js

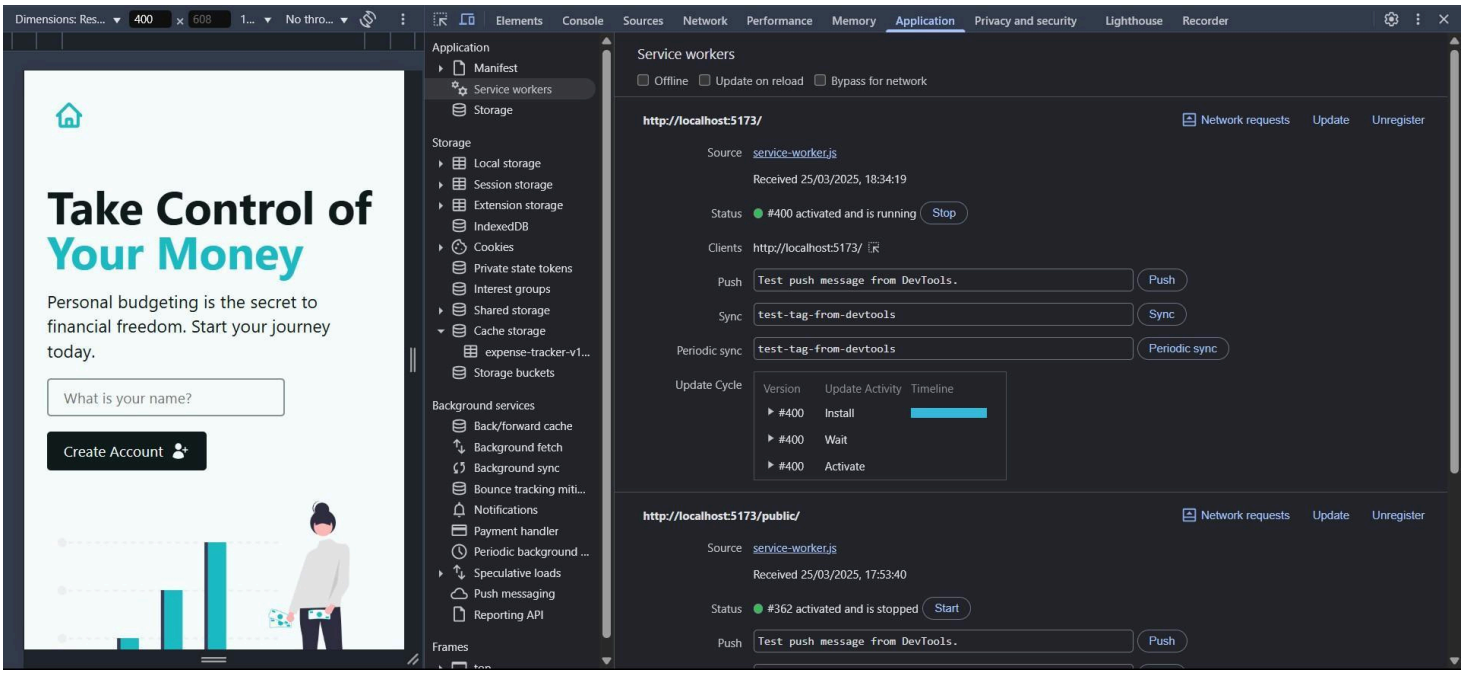
```
self.addEventListener("install", function (event)
   { event.waitUntil(preLoad());
});

var filesToCache = [
   '/',
   '/menu',
   '/contactUs',
   '/offline.html',
];

var preLoad = function () {
   return caches.open("offline").then(function (cache) {
      // caching index and important routes
      return cache.addAll(filesToCache);
   });
};

self.addEventListener("fetch", function (event) {
   event.respondWith(checkResponse(event.request).catch(function
   () {
      return returnFromCache(event.request);
   }));
   event.waitUntil(addToCache(event.request));
});

var checkResponse = function (request) {
   return new Promise(function (fulfill, reject)
   {
      fetch(request).then(function (response)
         { if (response.status !== 404) {
```

```
            fulfill(response)
          } else {
            reject();
          }
        }, reject);
    });
};

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

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

Project Title: Spotify **Roll No. 33**

Output:





Sample Code with Output

Index.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/favicon.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="manifest" href="manifest.json" />
    <title>XTrack</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

main.jsx

```
import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
import App from './App.jsx'
import './index.css'

createRoot(document.getElementById('root')).render(
  <StrictMode>
    <App />
  </StrictMode>,
)

// Service Worker Registration
if ('serviceWorker' in navigator) {
  window.addEventListener('load', () => {
    navigator.serviceWorker.register('public/service-worker.js') // Path to your service worker
file
      .then((registration) => {
        console.log('[Service Worker] Registered with scope:', registration.scope);

        // Check if the service worker is active
        if (registration.active) {
          console.log('[Service Worker] Active');
```

```
        }

        // Check if the service worker is installing
        if (registration.installing) {
          console.log('[Service Worker] Installing');
        }

        // Check if the service worker is waiting
        if (registration.waiting) {
          console.log('[Service Worker] Waiting');
        }

        // Listen for state changes
        registration.addEventListener('updatefound', () => {
          console.log('[Service Worker] Update found');
          const installingWorker = registration.installing;
          if (installingWorker) {
            installingWorker.addEventListener('statechange', () => {
              if (installingWorker.state === 'installed') {
                if (navigator.serviceWorker.controller) {
                  console.log('[Service Worker] New content is available and will be used when
all tabs for this page are closed.');
                  // You can prompt the user to reload here if needed
                } else {
                  console.log('[Service Worker] Content is cached for offline use.');
                }
              }
            });
          }
        });
      })
                                                .catch((error) => {
                                                  console.error('[Service Worker] Registration
                                                  failed:', error);
                                                });
  });

} else {
  console.warn('[Service Worker] Not supported in this browser.');
}
```

service-worker.js

```
// public/service-worker.js

const CACHE_NAME = 'expense-tracker-v1'; // Change this when you update the service worker
const urlsToCache = [
  '/', // Cache the root URL (index.html)
  '/index.html',
  '/manifest.json', // if you have one
  // Correct paths to your built assets
  'dist/assets/index-BPUHWJbD.js', // Replace with the actual path
  'dist/assets/index-BvWQCeOo.css',
];

// Install Event: Cache static assets
self.addEventListener('install', (event) => {
  console.log('[Service Worker] Install');
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then((cache) => {
        console.log('[Service Worker] Caching app shell');
        return cache.addAll(urlsToCache);
      })
      .catch((error) => {
        console.error('[Service Worker] Caching failed:', error);
      })
  );
});

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

```
  );
});

// Fetch Event: Serve from cache or network
self.addEventListener('fetch', (event) => {
  console.log('[Service Worker] Fetch', event.request.url);
  event.respondWith(
    caches.match(event.request)
      .then((response) => {
        // Cache hit - return response
        if (response) {
          console.log('[Service Worker] Found in cache:', event.request.url);
          return response;
        }
        // Not in cache - return fetch request
        console.log('[Service Worker] Not found in cache, fetching:', event.request.url);
        return fetch(event.request);
      })
  );
});
```

Conclusion: Successfully installed and activated the process of new service worker for pwa.

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.	33
Name	Aryan Anil Patankar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

Experiment No. 9

Name-Aryan Patankar

D15A/33

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.

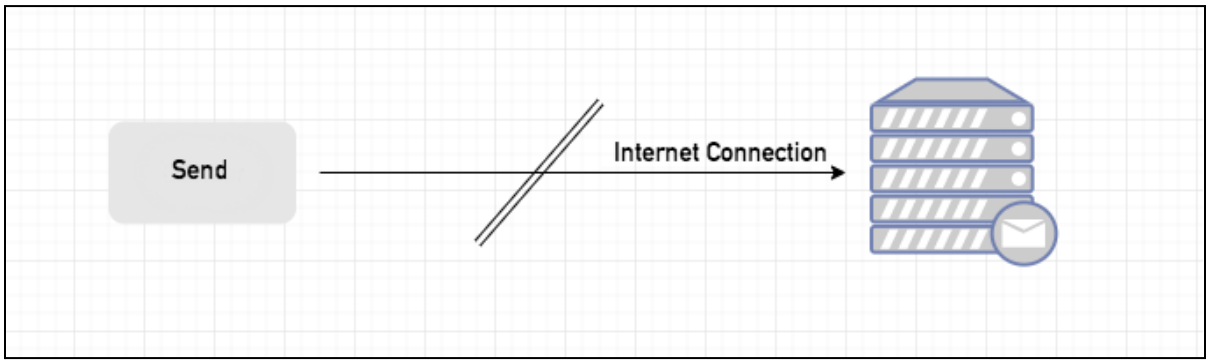
```
self.addEventListener("fetch", function (event) {
    const req = event.request;
    const url = new URL(req.url);

    if (url.origin === location.origin) {
        event.respondWith(cacheFirst(req));
    }
    else {
        event.respondWith(networkFirst(req));
    }
});

async function cacheFirst(req) {
    return await caches.match(req) || fetch(req);
}

async function networkFirst(req) {
    const cache = await caches.open("pwa-dynamic");
    try {
        const res = await fetch(req);
        cache.put(req, res.clone());
        return res;
    } catch (error) {
        const cachedResponse = await cache.match(req);
        return cachedResponse || await caches.match("./noconnection.json");
    }
}
```

Sync Event

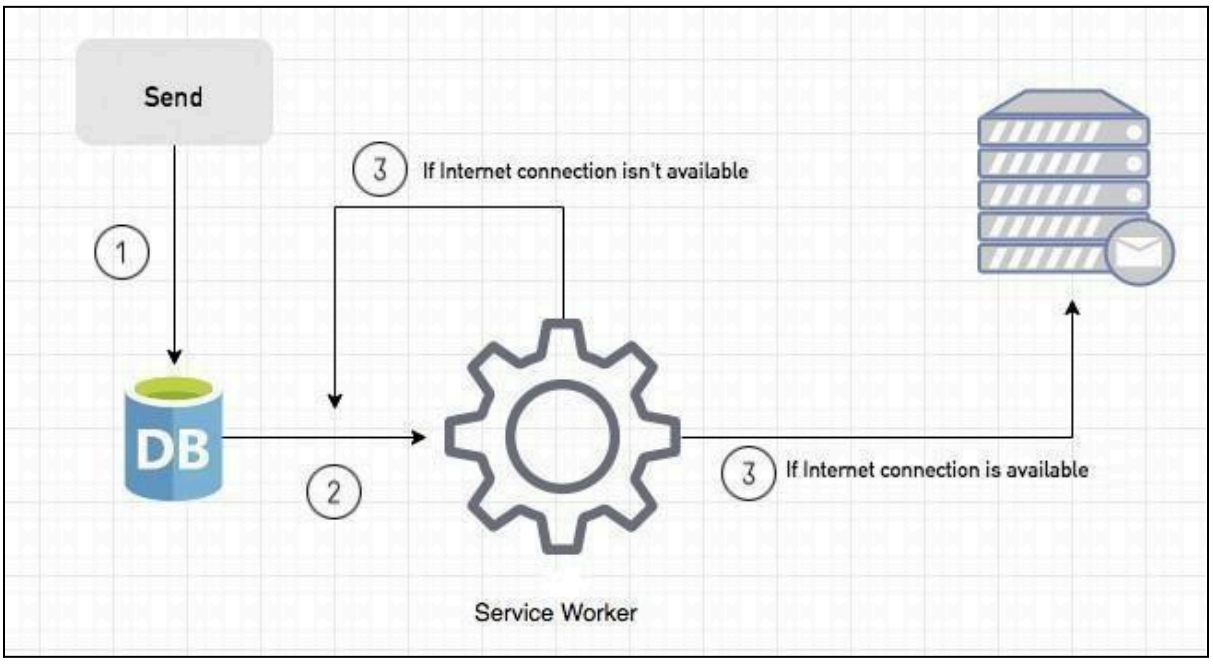
Background Sync is a Web API that is used to delay a process until the Internet connection is stable. We can adapt this definition to the real world; there is an e-mail client application that works on the browser and we want to send an email with this tool. Internet connection is broken while we are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

Here is a job for the Background Sync.

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



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



1. When we click the “send” button, email content will be saved to IndexedDB.
2. Background Sync registration.
3. **If the Internet connection is available**, all email content will be read and sent to Mail Server.

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

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

Event Listener for Background Sync Registration

```
document.querySelector("button").addEventListener("click", async () => {
    var swRegistration = await navigator.serviceWorker.register("sw.js");
    swRegistration.sync.register("helloSync").then(function () {
        console.log("helloSync success [main.js]");
    });
});
```

Event Listener for sw.js

```
self.addEventListener('sync', event => {
    if (event.tag == 'helloSync') {
        console.log("helloSync [sw.js]");
    }
});
```

Push Event

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

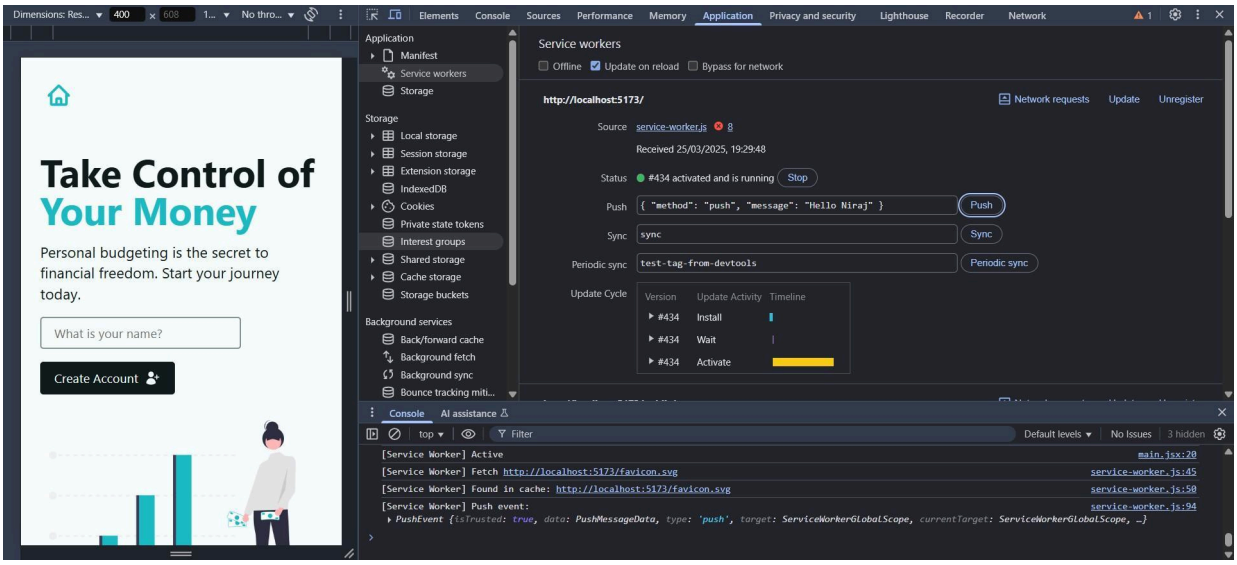
We can check in the following example.
“Notification.requestPermission();” is the necessary line to show notification to the user. If you don’t want to show any notification, you don’t need this line.

In the following code block is in sw.js file. You can handle push notifications with this event. In this example, I kept it simple. We send an object that has “method” and

"message" properties. If the method value is "pushMessage", we open the information notification with the "message" property.

```
self.addEventListener('push', event => {
    if (event && event.data) {
        var data = event.data.json();
        if (data.method === "pushMessage") {
            event.waitUntil(self.registration.showNotification("Test App", {
                body: data.message
            }));
        }
    }
});
```

You can use Application Tab from Chrome Developer Tools for testing push notification.



Code:

service-worker.js

```
// public/service-worker.js


const CACHE_NAME = 'expense-tracker-v1'; // Change this when you update the service
worker
const urlsToCache = [
  '/', // Cache the root URL (index.html)
  '/index.html',
  '/manifest.json', // if you have one
  // Correct paths to your built assets
  'dist/assets/index-BPUHWJbD.js', // Replace with the actual path
  'dist/assets/index-BvWQCeOo.css',
];


self.addEventListener('install', (event) => {
  console.log('[Service Worker] Install');
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then((cache) => {
        console.log('[Service Worker] Caching app shell');
        return cache.addAll(urlsToCache);
      })
      .catch((error) => {
        console.error('[Service Worker] Caching failed:', error);
      })
  );
});


self.addEventListener('activate', (event) => {
  console.log('[Service Worker] Activate');
  event.waitUntil(
    caches.keys().then((cacheNames) => {
```

```
      return Promise.all(
        cacheNames.map((cacheName) => {
          if (cacheName !== CACHE_NAME) {
            console.log('[Service Worker] Deleting old cache:', cacheName);
            return caches.delete(cacheName);
          }
        })
      );
    })
  );
});

// Fetch Event: Cache-first strategy
self.addEventListener('fetch', (event) => {
  console.log('[Service Worker] Fetch', event.request.url);
  event.respondWith(
    caches.match(event.request)
      .then((response) => {
        if (response) {
          console.log('[Service Worker] Found in cache:', event.request.url);
          return response;
        }
        return fetch(event.request)
          .then((response) => {
            if (!response || response.status !== 200 || response.type !== 'basic') {
              return response;
            }
            const responseToCache = response.clone();
            caches.open(CACHE_NAME)
              .then((cache) => {
                cache.put(event.request, responseToCache);
              });
            return response;
```

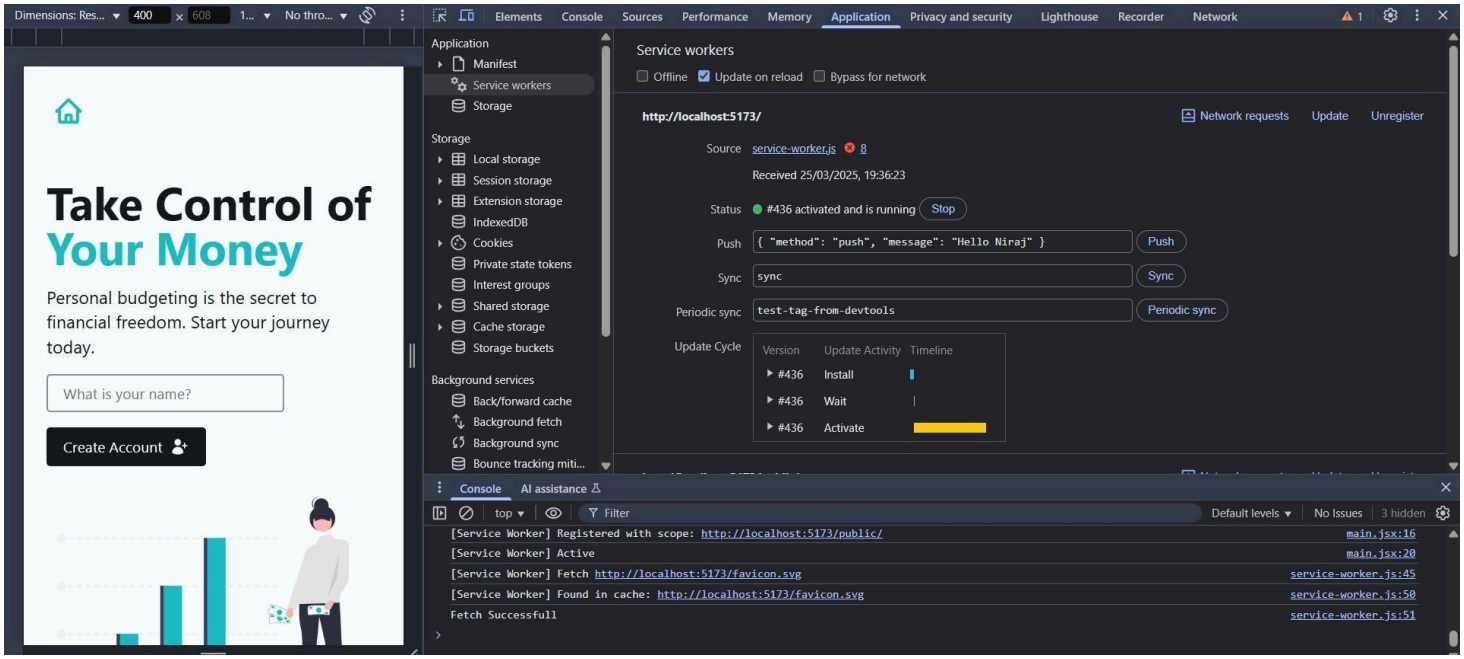
```
                                                    })
                                                    .catch((error) => {
                                                      console.error('[Service Worker] Fetch
                                                      failed:', error);
                                                      // Handle network errors (e.g., show
                                                      offline message)
                                                    });
                                            );  })
});




// Sync Event: Handle background sync
self.addEventListener('sync', (event) => {
  console.log('[Service Worker] Sync event:', event.tag);
  if (event.tag === 'sync-expenses') {
    event.waitUntil(
      // Your background sync logic here (e.g., send expenses to server)
      syncExpenses()
        .then(() => {
          console.log('[Service Worker] Expenses synced successfully');
        })
        .catch((error) => {
          console.error('[Service Worker] Expense sync failed:', error);
          // Handle sync errors (e.g., retry later)
        })
    );
  }
});


// Push Event: Handle push notifications
self.addEventListener('push', function(event) {
  console.log('[Service Worker] Push event:', event);
  let data;
  try {
```

```
    data = event.data ? event.data.json() : {};
  } catch (error) {
    console.error('Error parsing push notification data:', error);
    data = { title: 'Error', message: 'Could not parse notification data.' };
  }

  if (Notification.permission === 'granted') {
    event.waitUntil(
      self.registration.showNotification(data.title || 'Xtrack',
        { body: data.message || 'No message provided.',
        icon: data.icon || '/your-notification-icon.png', // Provide a default icon
      })
    );
  } else {
    console.log('Notification permission not granted. Notification will not be
shown.');
  }
});



//Helper function for sync event
async function syncExpenses() {
  // Replace this with your actual expense syncing logic
  // This example just simulates a delay
  return new Promise((resolve) => {
    setTimeout(resolve, 2000); // Simulate a 2-second delay
  });
}
```

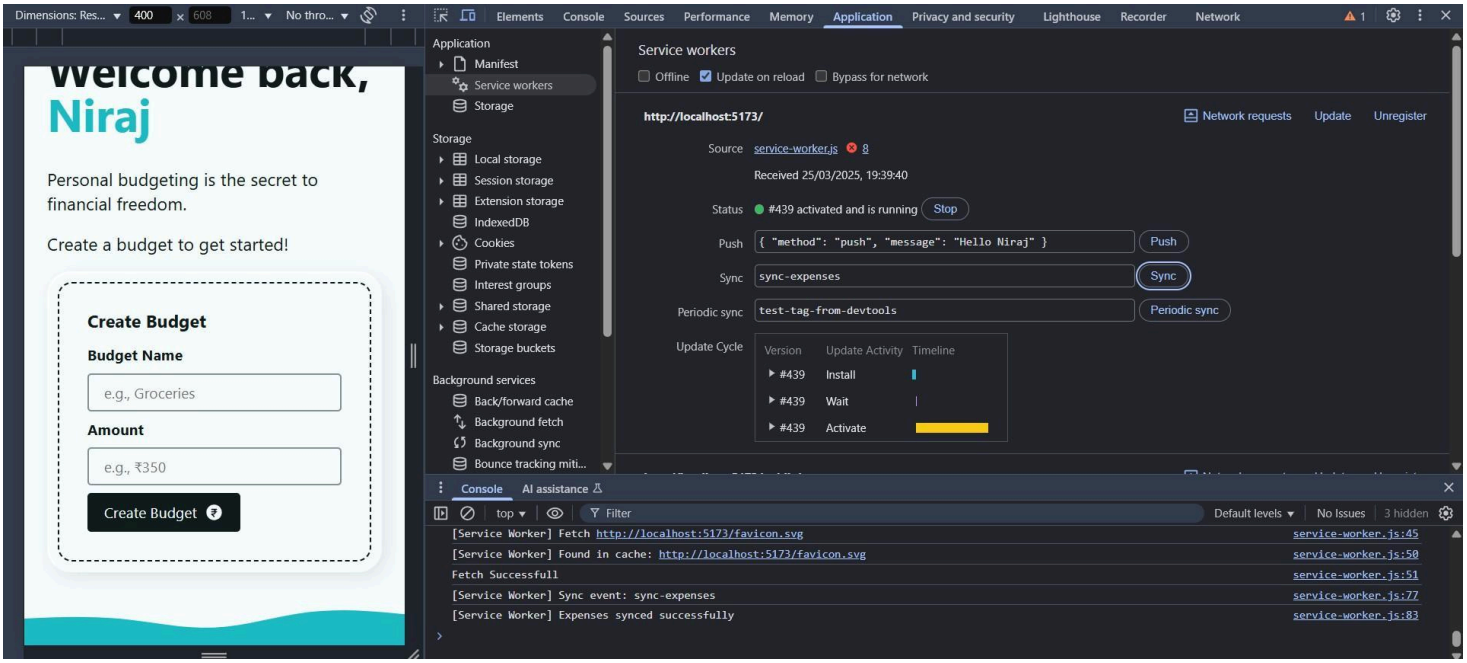
Output:

Project Title: Spotify **Roll No. 33**

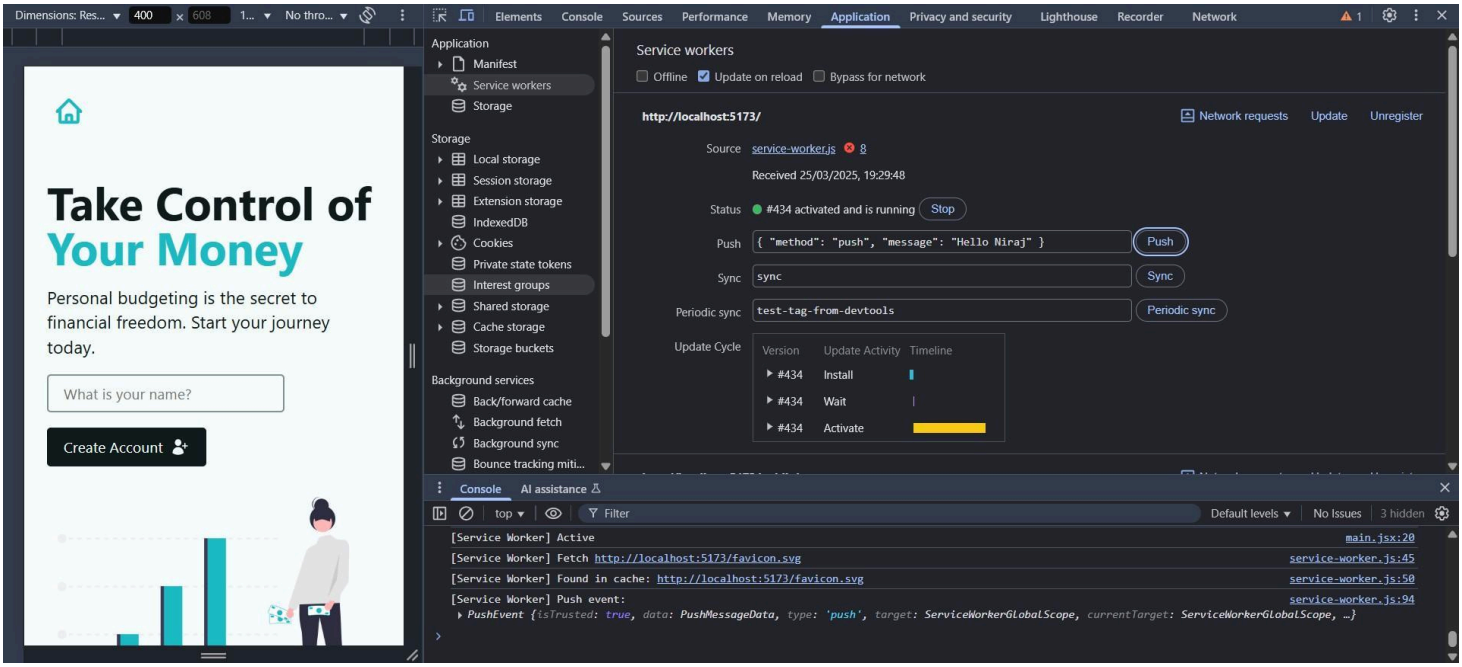
Fetchevent



Sync event



Push event



Conclusion: Successfully implemented Service worker events like fetch, sync and push for PWA.

MAD & PWA Lab
Journal

Experiment No.	10
Experiment Title.	To study and implement deployment of Ecommerce PWA to GitHub Pages.
Roll No.	33
Name	Aryan Anil Patankar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO5: Design and Develop a responsive User Interface by applying PWA Design techniques
Grade:	

Experiment
No. 10 MAD
& PWA Lab

Aryan Patankar
D15A/33

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 developers 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.

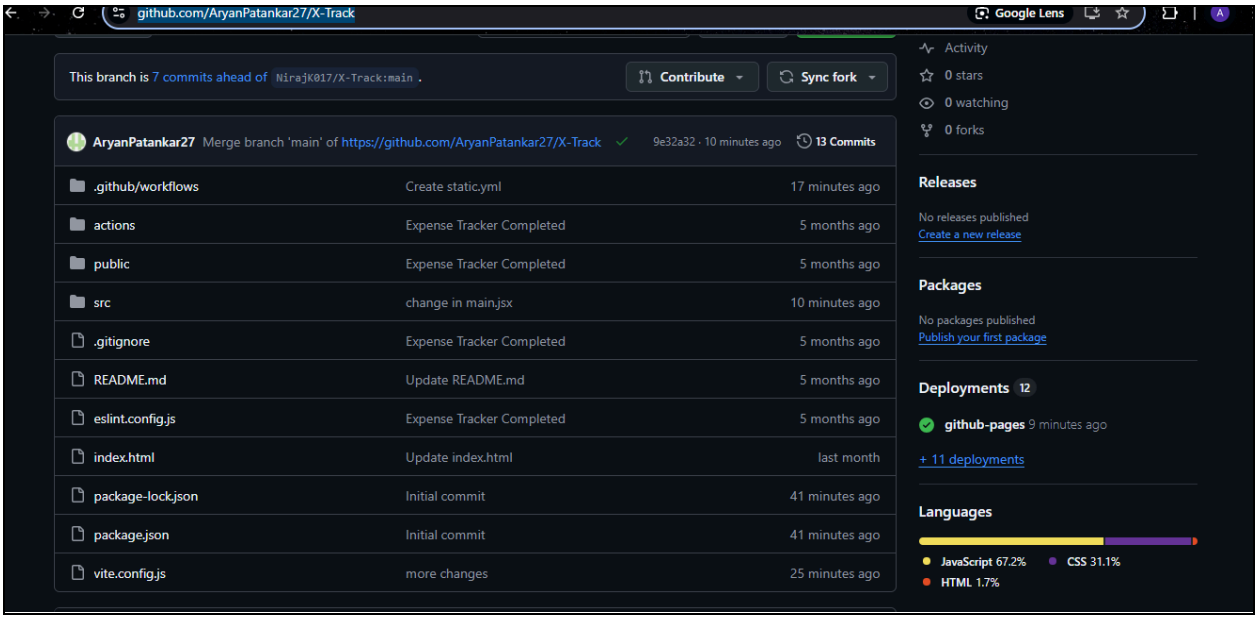
Cons

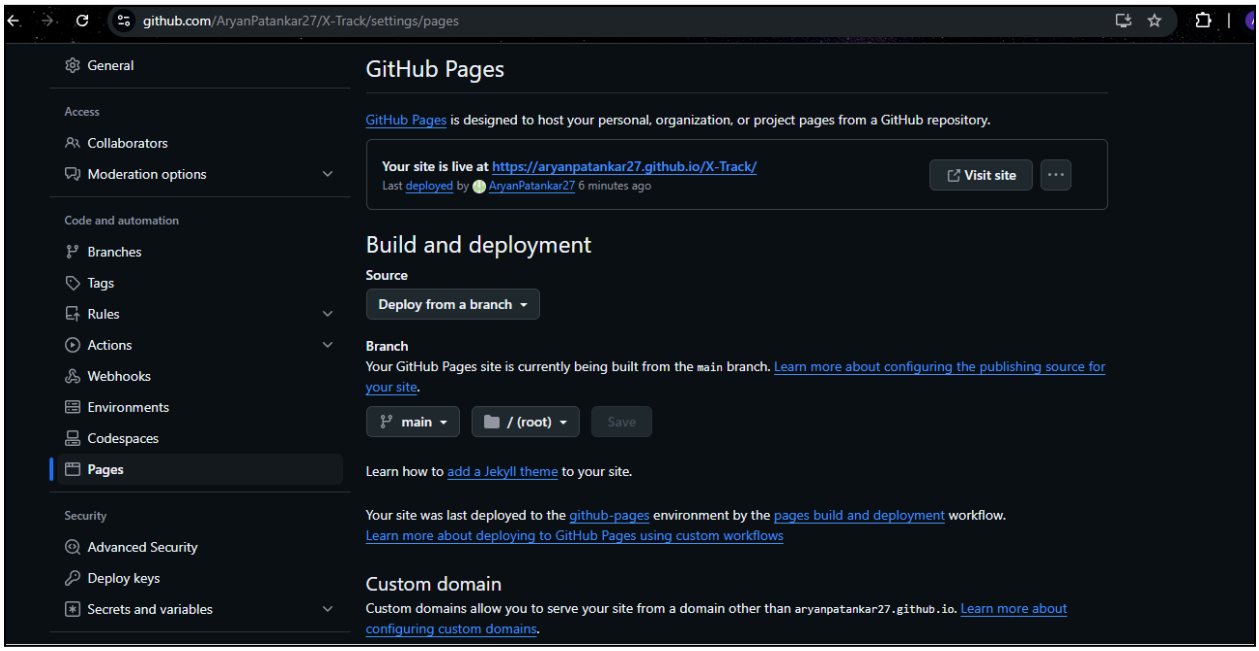
1. Only 10 GB of data transfer is allowed per month. But this is not really a big problem, if you use a CDN or AMP.
2. Command-line interface only.
3. No in-built support for any static site generator.

Link to our GitHub repository: https://github.com/AryanPatankar27/X-Track

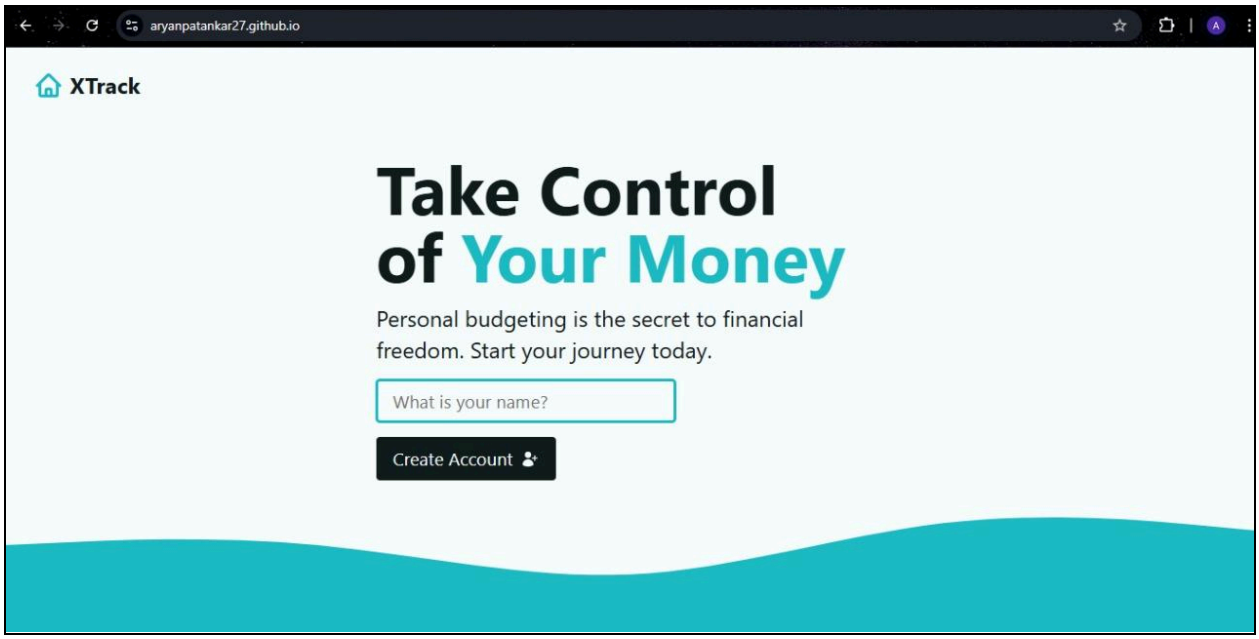
Github Screenshot:

Project Title: Spotify **Roll No. 33**





Project Title: Spotify **Roll No. 33**



MAD & PWA Lab
Journal

Experiment No.	11
Experiment Title.	To use google Lighthouse PWA Analysis Tool to test the PWA functioning.
Roll No.	33
Name	Aryan Anil Patankar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO6: Develop and Analyze PWA Features and deploy it over app hosting solution
Grade:	

Experiment 11

Aryan Patankar
D15A/33

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

Theory :

Google Lighthouse :

Google Lighthouse is a tool that lets you audit your web application based on a number of parameters including (but not limited to) performance, based on a number of metrics, mobile compatibility, Progressive Web App (PWA) implementations, etc. All you have to do is run it on a page or pass it a URL, sit back for a couple of minutes and get a very elaborate report, not much short of one that a professional auditor would have compiled in about a week. The best part is that you have to set up almost nothing to get started. Let's begin by looking at some of the top features and audit criteria used by Lighthouse.

Key Features and Audit Metrics

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

Performance: This score is an aggregation of how the page fared in aspects such as (but not limited to) loading speed, time taken for loading

for basic frame(s), displaying meaningful content to the user, etc. To a layman, this score is indicative of how decently the site performs, with a score of 100 meaning that you figure in the 98th percentile, 50 meaning that you figure in the 75th percentile and so on.

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

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

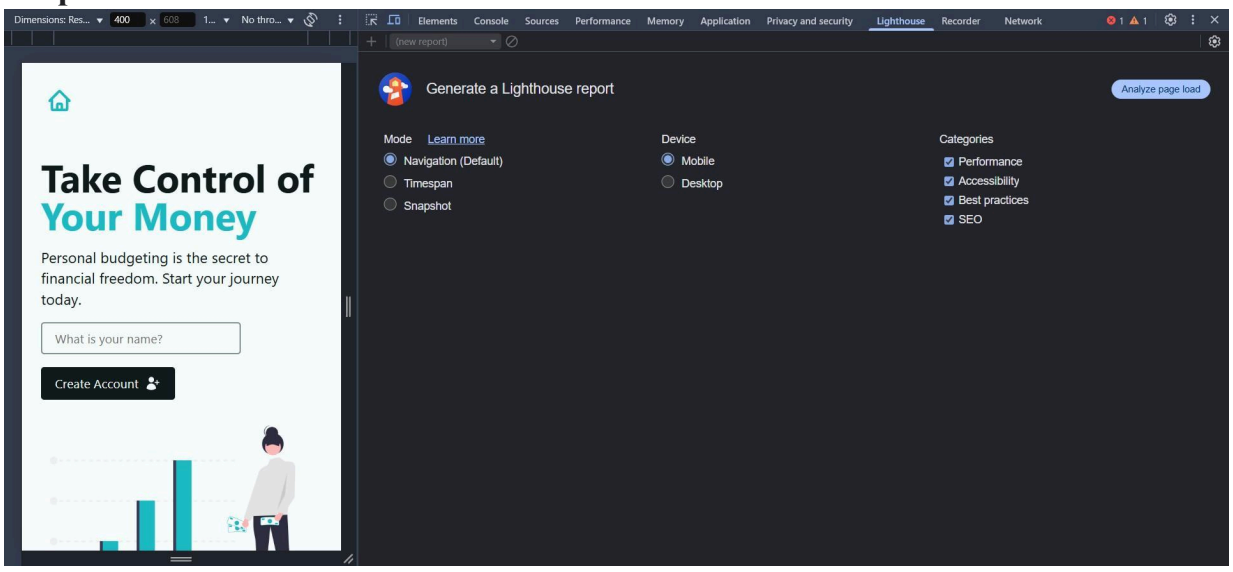
Best Practices: As any developer would know, there are a number of practices that have been deemed 'best' based on empirical data. This metric is an aggregation of many such points, including but not limited to:Use of HTTPS Avoiding the use of deprecated code elements like tags, directives, libraries, etc. Password input with paste-into disabled
Geo-Location and cookie usage alerts on load, etc.

Improvement In Code:

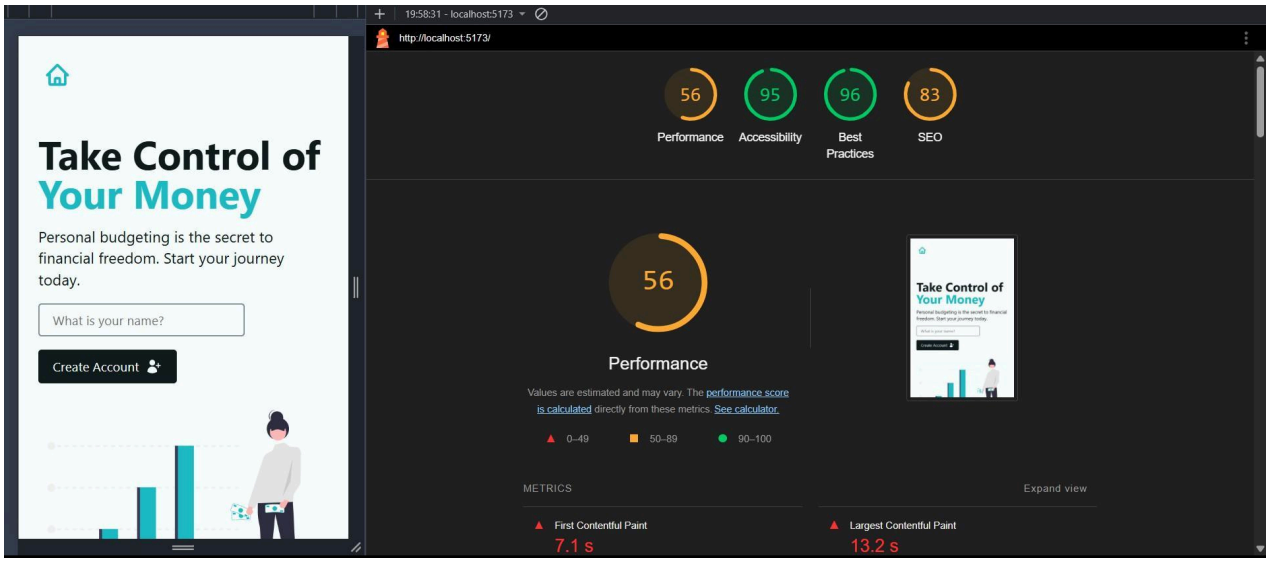
Index.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/favicon.svg" />
     <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
     <meta name="description" content="A concise and accurate description
of your Expense Tracker app. This will appear in search engine results.">
    <link rel="manifest" href="manifest.json" />
    <title>XTrack</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

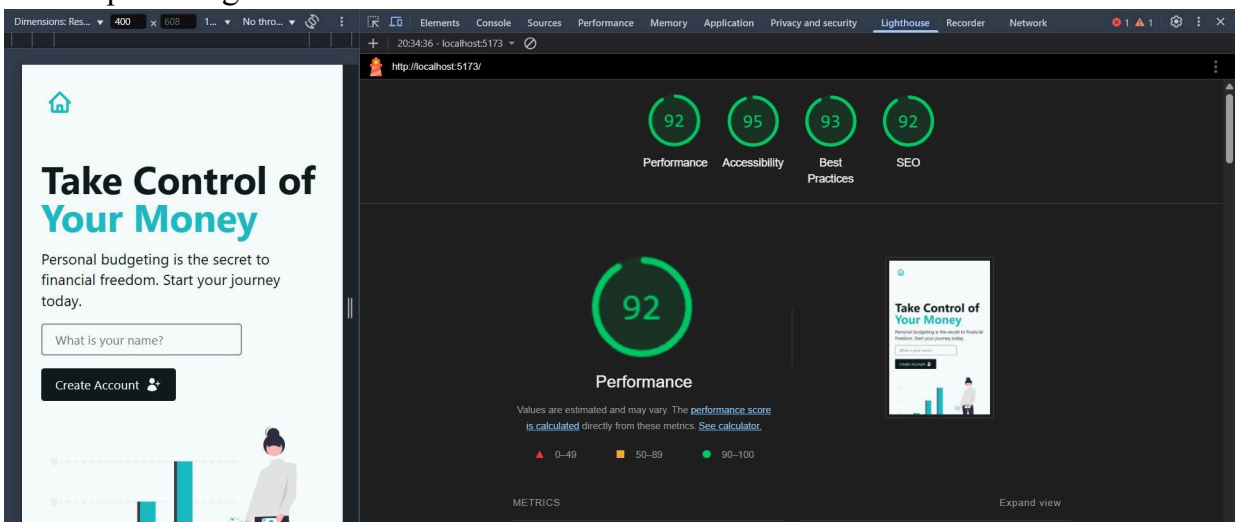
Output:



Before Optimizing Code:



After Optimizing Code:



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	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. 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. 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. 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.
Roll No.	33
Name	Aryan Anil Patankar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO1: Understand cross platform mobile application development using Flutter framework LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation LO3: Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS
Grade:	

MAD & PWA Lab
Journal

Experiment No.	Assignment-2
Assignment 2 Questions	1. Define Progressive Web App (PWA) and explain its significance in modern web development.Discuss the key characteristics that differentiate PWAs from traditional mobile apps 2. Define responsive web design and explain its importance in the context of Progressive Web Apps.Compare and contrast responsive, fluid, and adaptive web design approaches. 3. Describe the lifecycle of Service Workers, including registration, installation, and activation phases. 4. Explain the use of IndexedDB in the Service Worker for data storage.
Roll No.	33
Name	Aryan Anil Patankar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO4:Understand various PWA frameworks and their requirements LO5: Design and Develop a responsive User Interface by applying PWA Design techniques LO6:Develop and Analyze PWA Features and deploy it over app hosting solutions
Grade:	