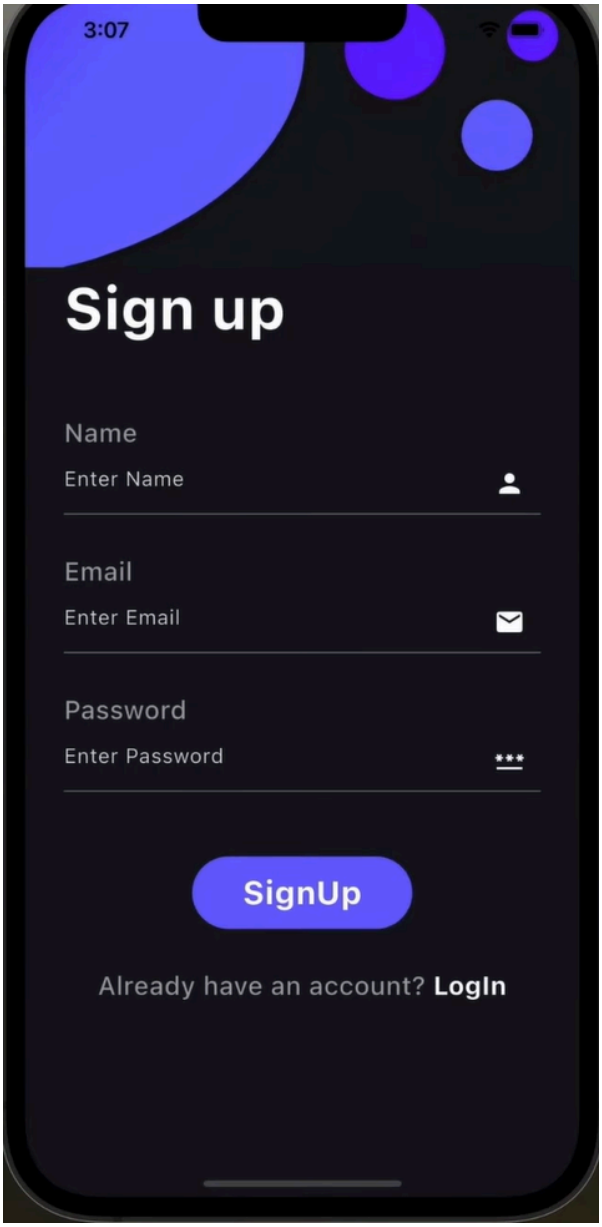
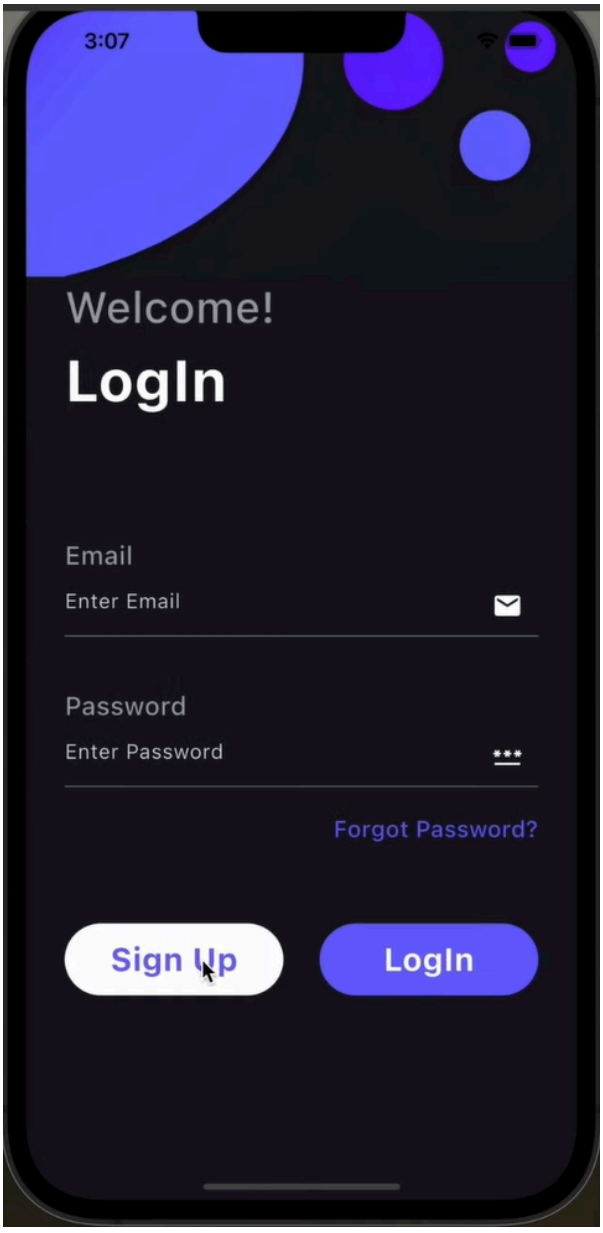
MPL Prerequisites Document

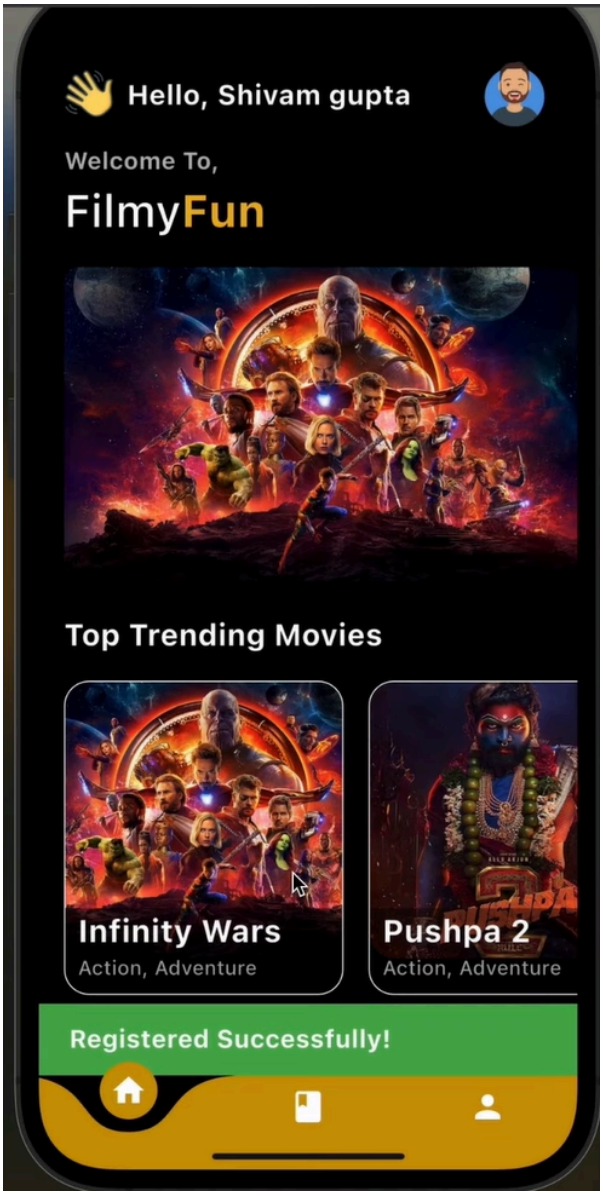
Name: Manorath Ital
Class : D15A
Roll no: 19

Screenshot	Features
	**Signup Page Features & Implementation:** 1. **User Input Fields (Name, Email, Password)** – Implemented using `TextFormField` in Flutter with validation. 2. **Icons for Input Fields** – Added using `Icon` widget within `InputDecoration`. 3. **Password Visibility Toggle** – Managed using `ObscureText` and `IconButton`. 4. **Signup Button** – Created using `ElevatedButton` with Firebase authentication. 5. **Login Navigation** – Implemented using `Navigator.push()` to switch to the login screen. 6. **UI Styling** – Designed using `Container`, `Column`, `Padding`, and `BoxDecoration`. 7. **Firebase Authentication** – `FirebaseAuth.createUserWithEmailAndPassword()` for user registration.



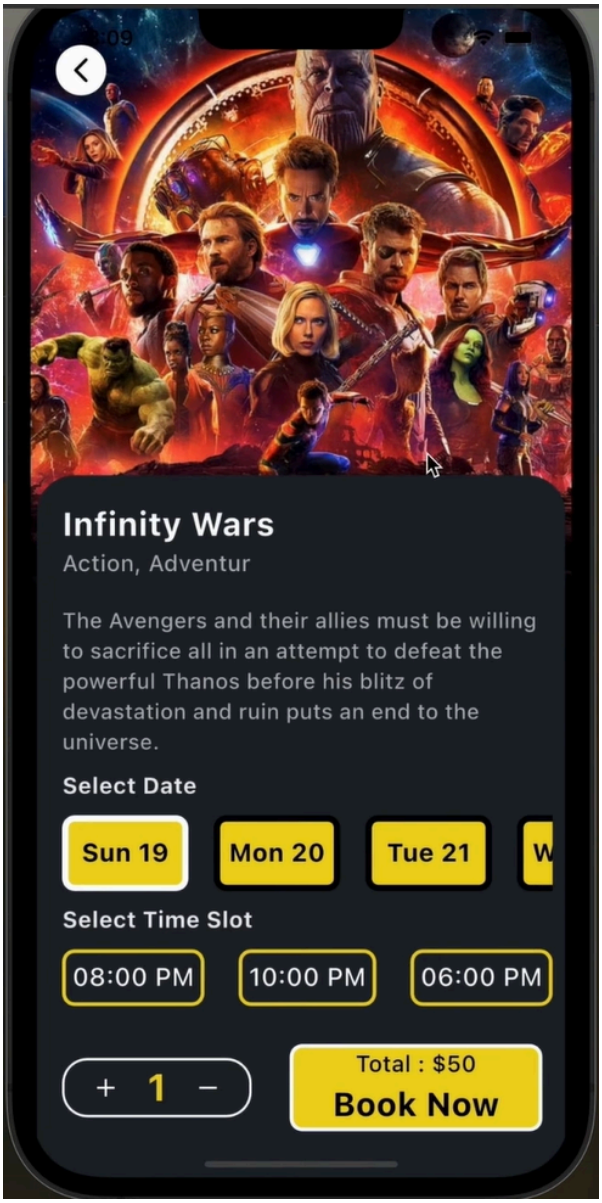
Login Page Features & Implementation:

1. **User Input Fields (Email, Password)** – Implemented using `TextFormField` with validation.
2. **Icons for Input Fields** – Integrated using `InputDecoration` with `Icon`.
3. **Password Visibility Toggle** – Managed using `ObscureText` and `IconButton`.
4. **Forgot Password Link** – Implemented using `GestureDetector` to navigate to password reset.
5. **Login & Signup Buttons** – Created using `ElevatedButton` and `OutlinedButton` with `onPressed` for navigation.
6. **UI Styling** – Designed using `Container`, `Column`, `Padding`, and `BoxDecoration`.
7. **Firebase Authentication** – `FirebaseAuth.signInWithEmailAndPassword()` for user login



Home Page Features & Implementation:

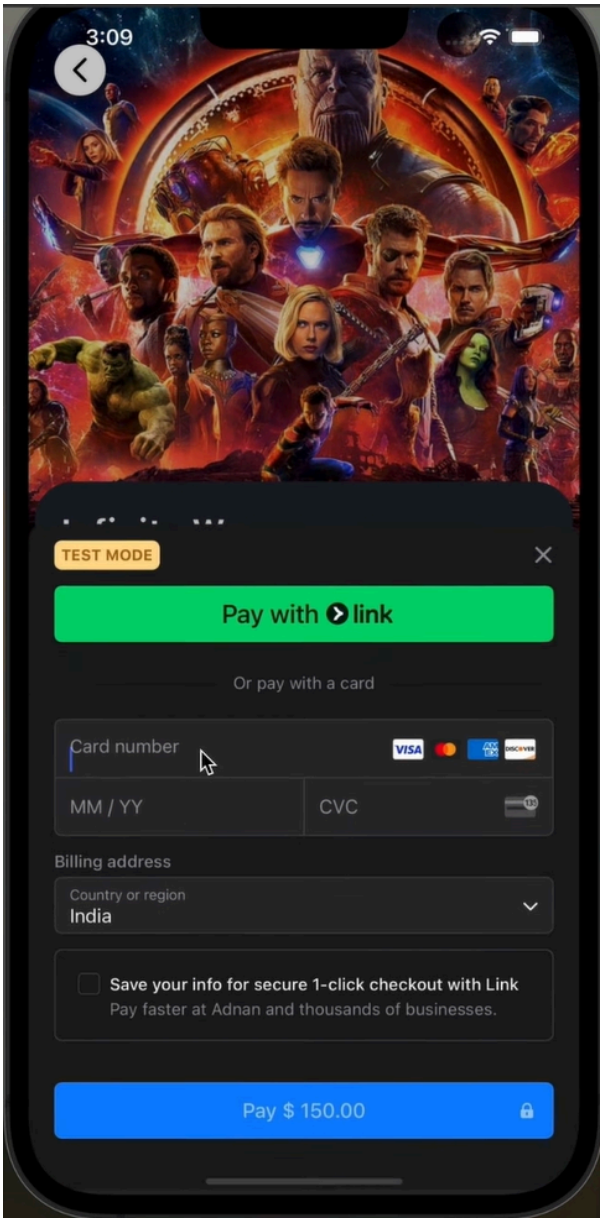
1. **User Greeting & Avatar** – Displayed using `Row` with `Text` and `CircleAvatar`, fetched from Firebase Authentication.
2. **App Branding (FilmyFun)** – Styled using `Text.rich` for multiple fonts and colors.
3. **Top Trending Movies Section** – Implemented using `ListView.builder` with Firebase Firestore integration.
4. **Movie Cards** – Built using `Container` with `ClipRRect` for rounded images and `Text` for movie details.
5. **Bottom Navigation Bar** – Created using `BottomNavigationBar` with icons for Home and Profile.
6. **Snack Bar Notification (Registered Successfully!)** – Managed using `ScaffoldMessenger.showSnackBar()`.
7. **Smooth UI & Styling** – Achieved using `Column`, `Padding`, and `BoxDecoration`.



Movie Details Page

Features

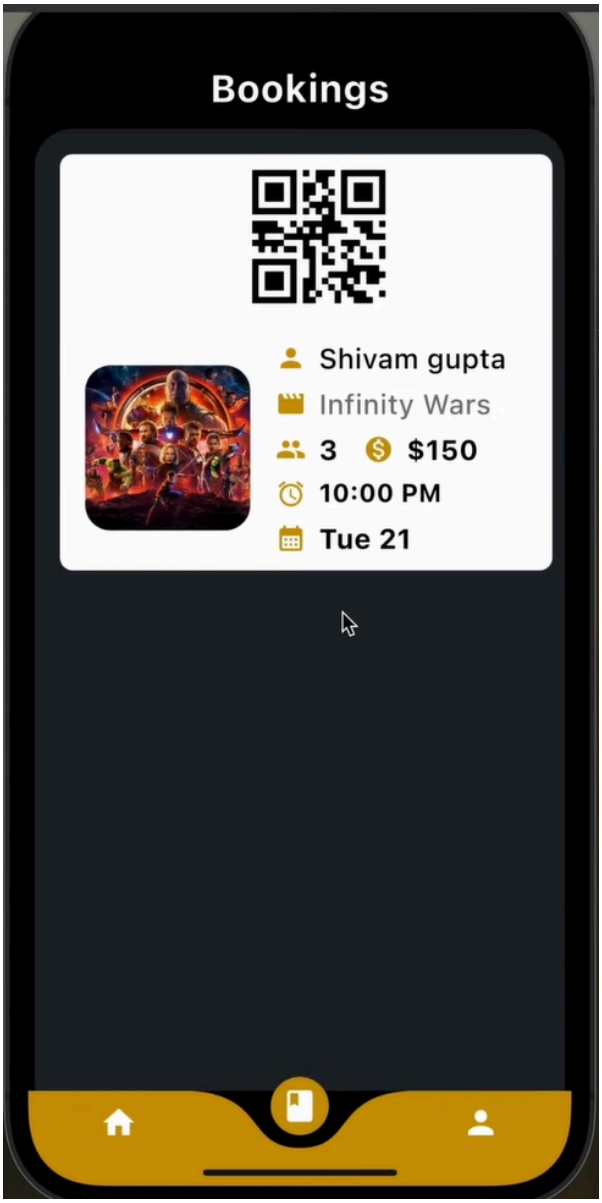
1. **Large Movie Poster** – Displays the movie prominently.
2. **Movie Title & Genre** – Clearly shown for easy identification.
3. **Short Movie Description** – Provides a brief overview of the movie.
4. **Date Selection** – Allows users to choose a preferred show date.
5. **Time Slot Selection** – Lets users select a specific time for the show.
6. **Ticket Quantity Selector** – Enables users to increase or decrease ticket count.
7. **Total Price Calculation** – Updates dynamically based on the selected number of tickets.
8. **Book Now Button** – A clear call-to-action for booking tickets.



Payment Page

Features

- **Movie Poster & Title** – Displays the selected movie.
- **Total Amount** – Shows the total price of the selected tickets.
- **Payment Options** –
    - Pay with a linked account (e.g., UPI, Wallets).
    - Pay using a credit/debit card.
- **Card Payment Form** –
    - Card number input.
    - Expiry date (MM/YY) field.
    - CVC security code input.
- **Billing Address Selection** – Users can choose their country/region.
- **Secure Checkout Option** – Users can save payment details for faster future transactions.
- **Pay Button** – Finalizes the payment process.

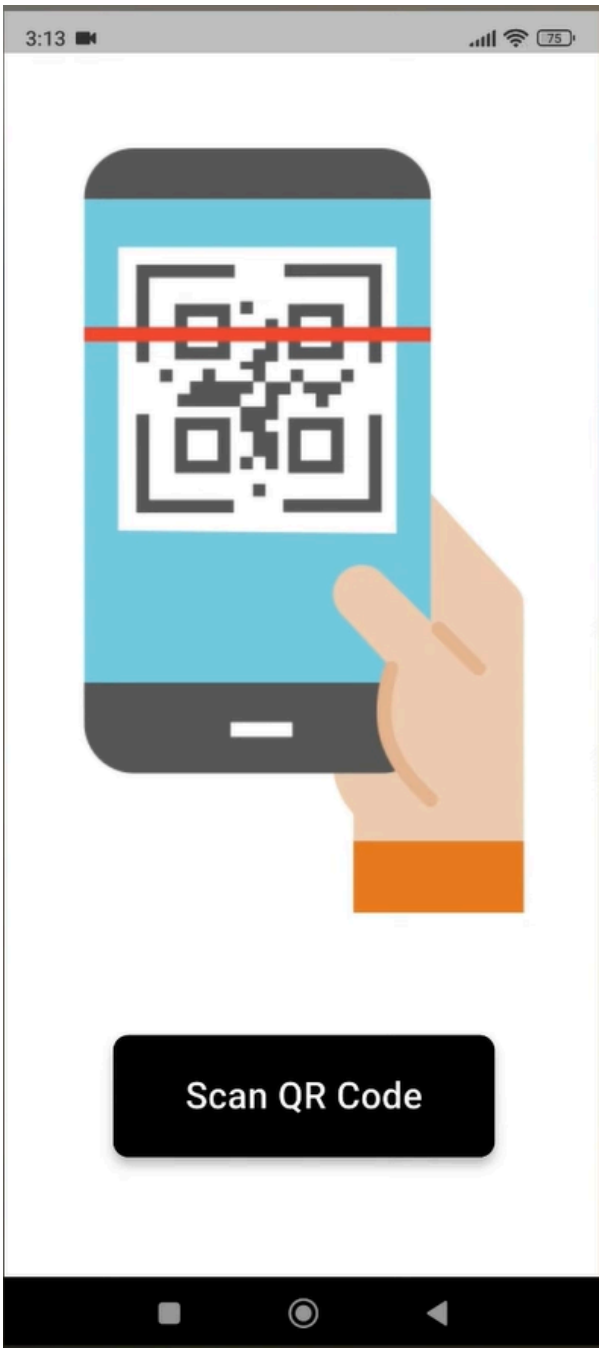


Booking Page

Features

1. **Header** – Displays "Bookings" at the top.
2. **QR Code** – Allows quick verification for entry.
3. **Booking Details Card** – Includes:
    - **User Name** – Shows the name of the person who booked the ticket.
    - **Movie Poster & Title** – Displays the booked movie.
    - **Number of Tickets** – Indicates the total seats booked.
    - **Total Price** – Shows the total amount paid.
    - **Showtime** – Displays the selected movie time.
    - **Date of Booking** – Shows the movie date.

4.Bottom Navigation Bar – Includes icons for navigating to home, booking history, and other sections.



QR Code Scanner Page

Features

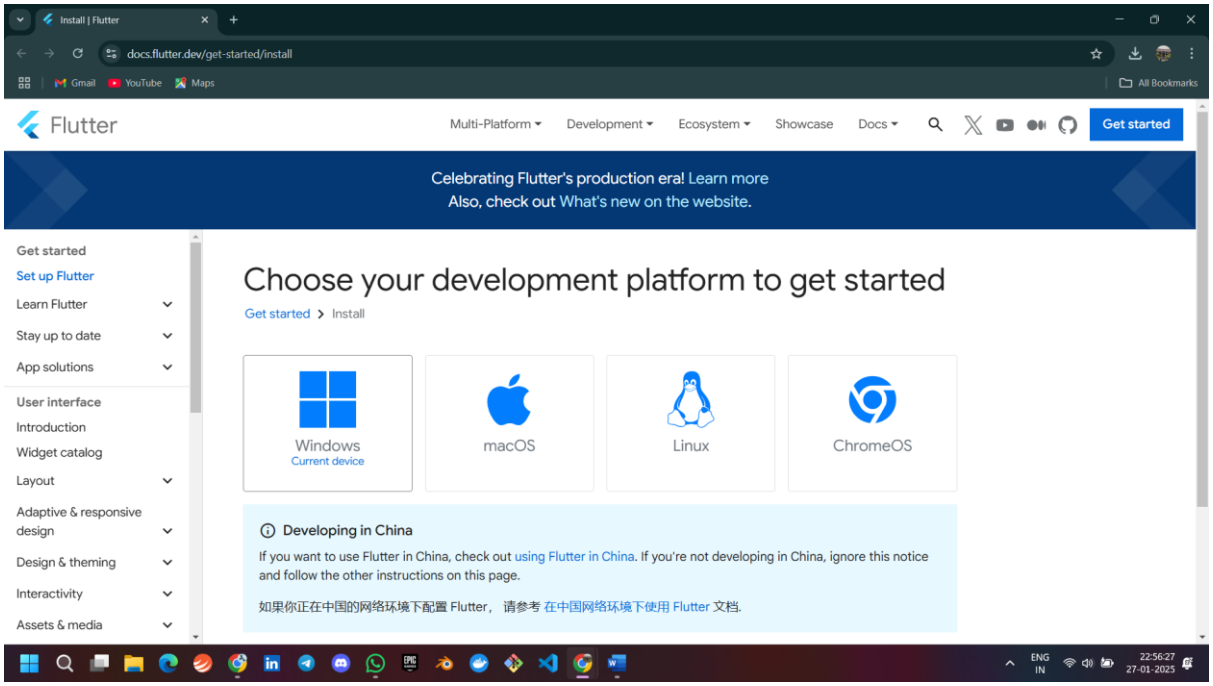
1. **Illustration** – Shows a phone scanning a QR code for user guidance.
2. **Scan QR Code Button** – Triggers the device's camera to scan a QR code.
3. **Minimal UI** – Clean, simple, and easy to use.

EXPERIMENT-1

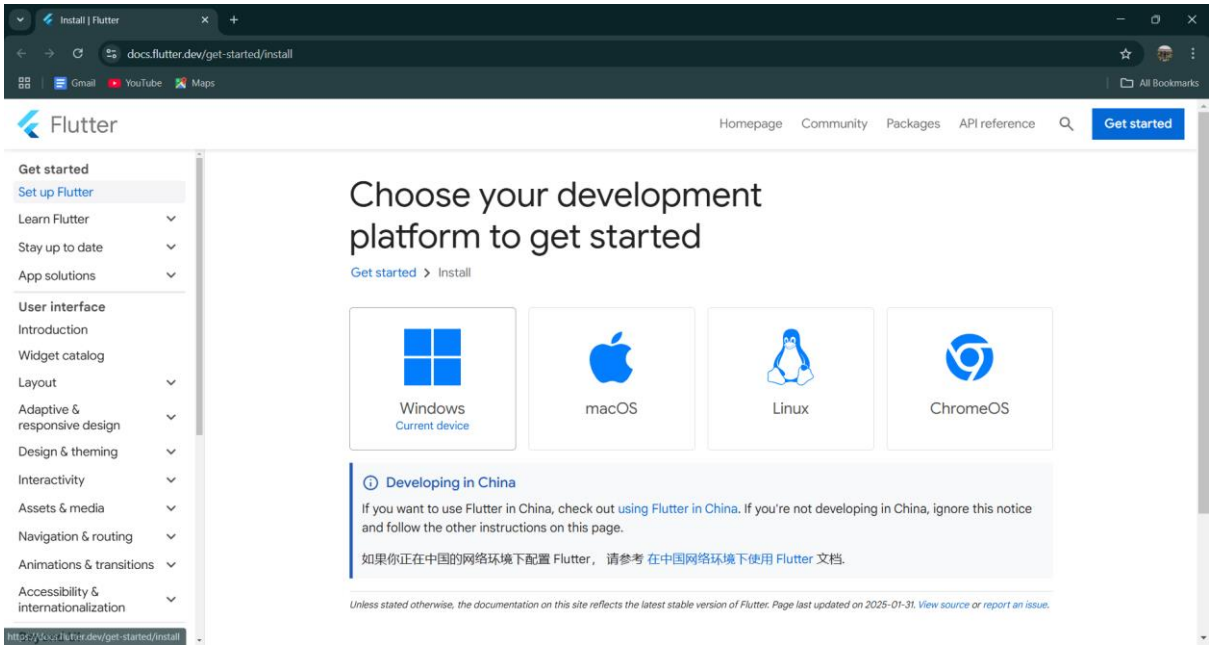
NAME: MANORATH ITAL
CLASS: D15A
ROLLNO: 19

AIM: Installation and Configuration of Flutter Environment.

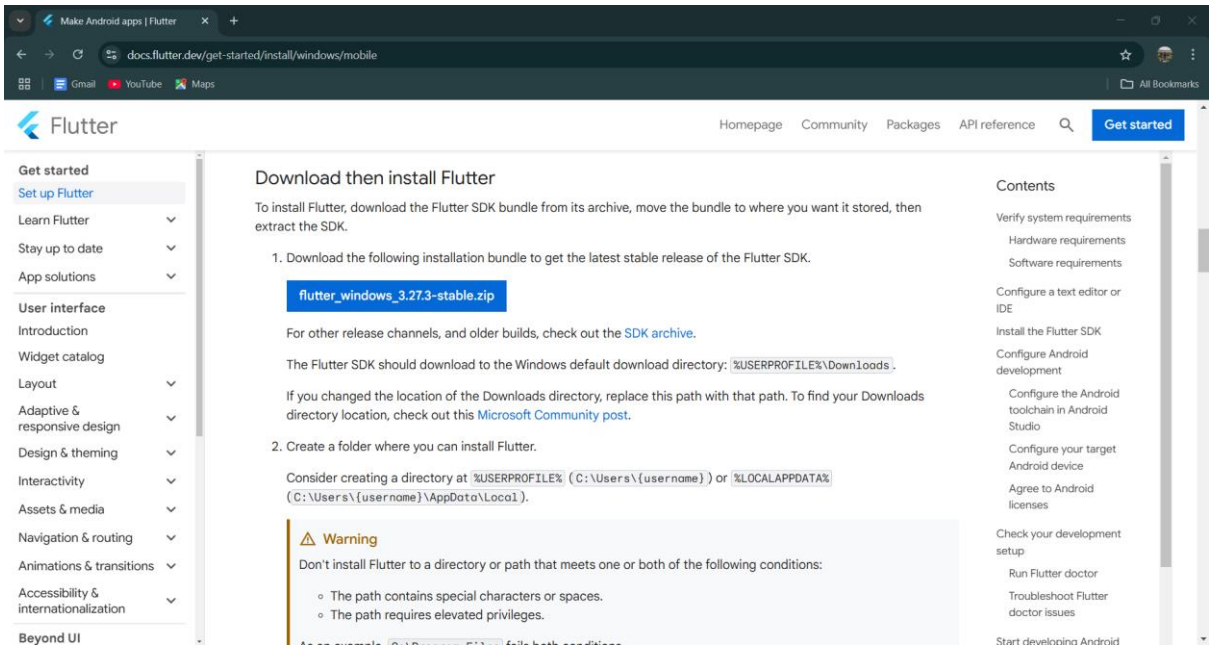
Step 1: Go to the official Flutter website: https://docs.flutter.dev/get-started/install



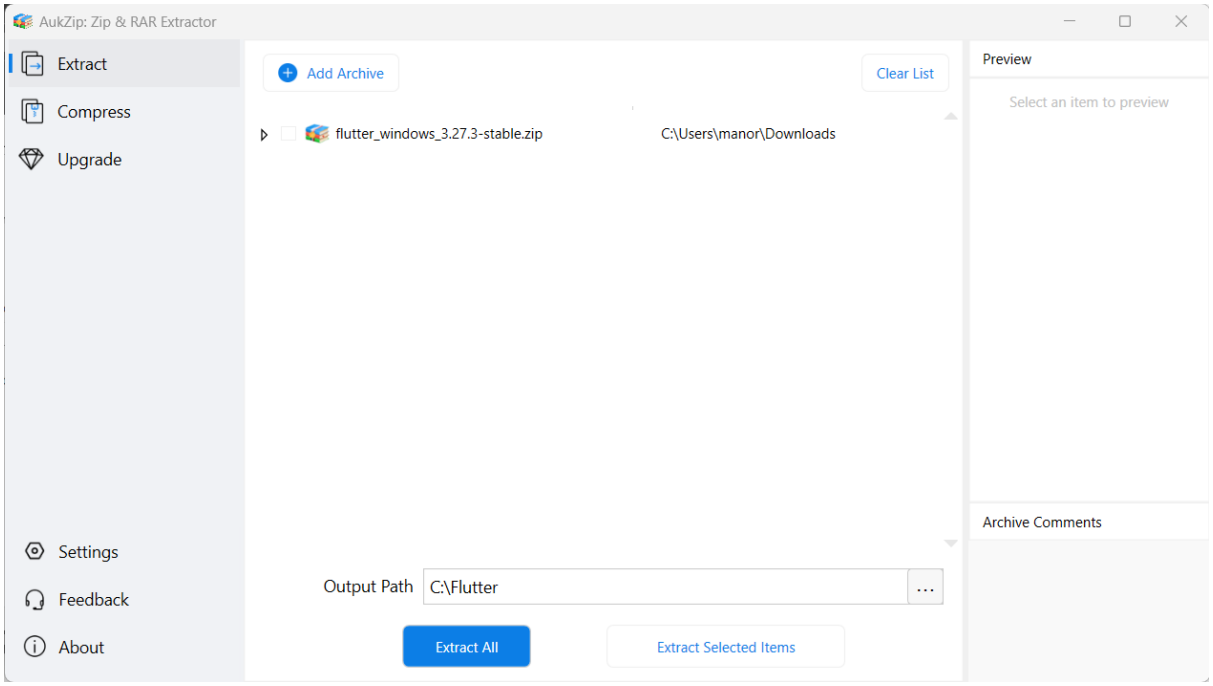
Step 2: To download the latest Flutter SDK, click on the Windows icon > Android



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



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

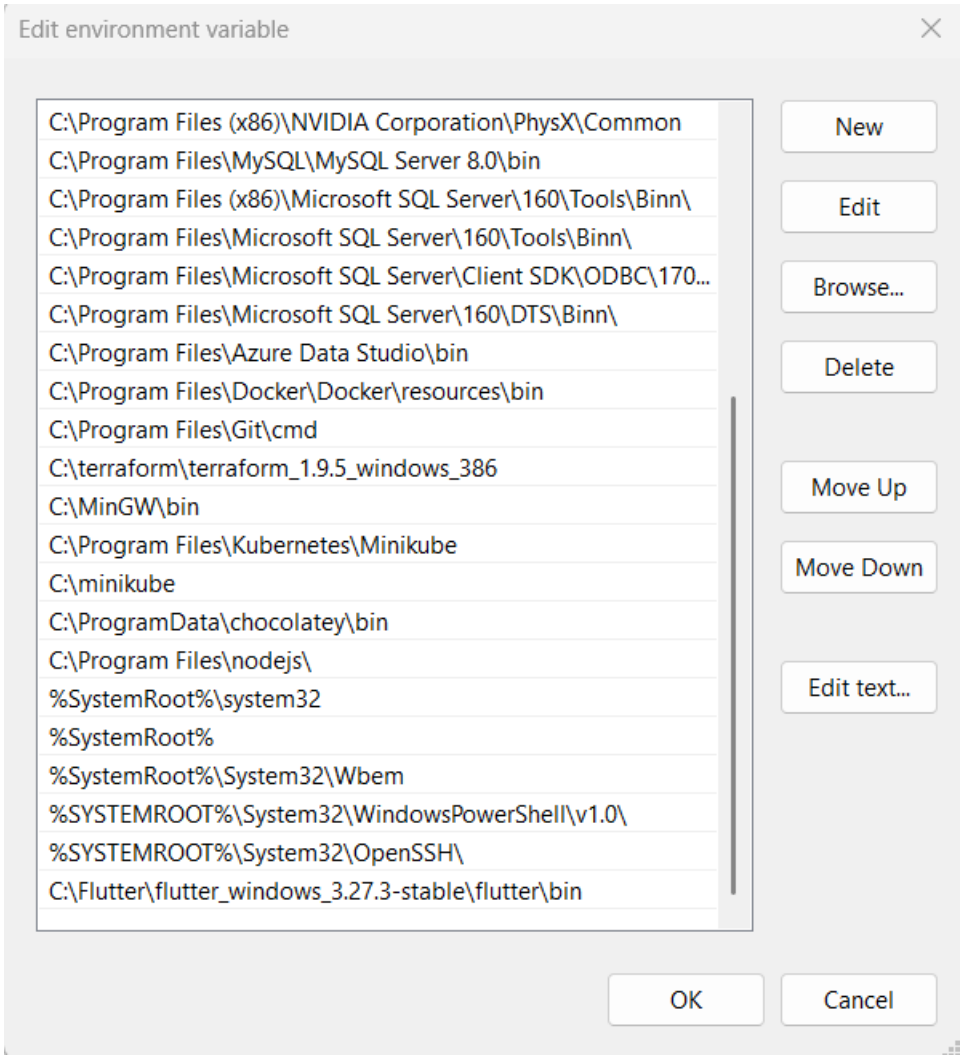


Step 5 :- Add Flutter to System PATH

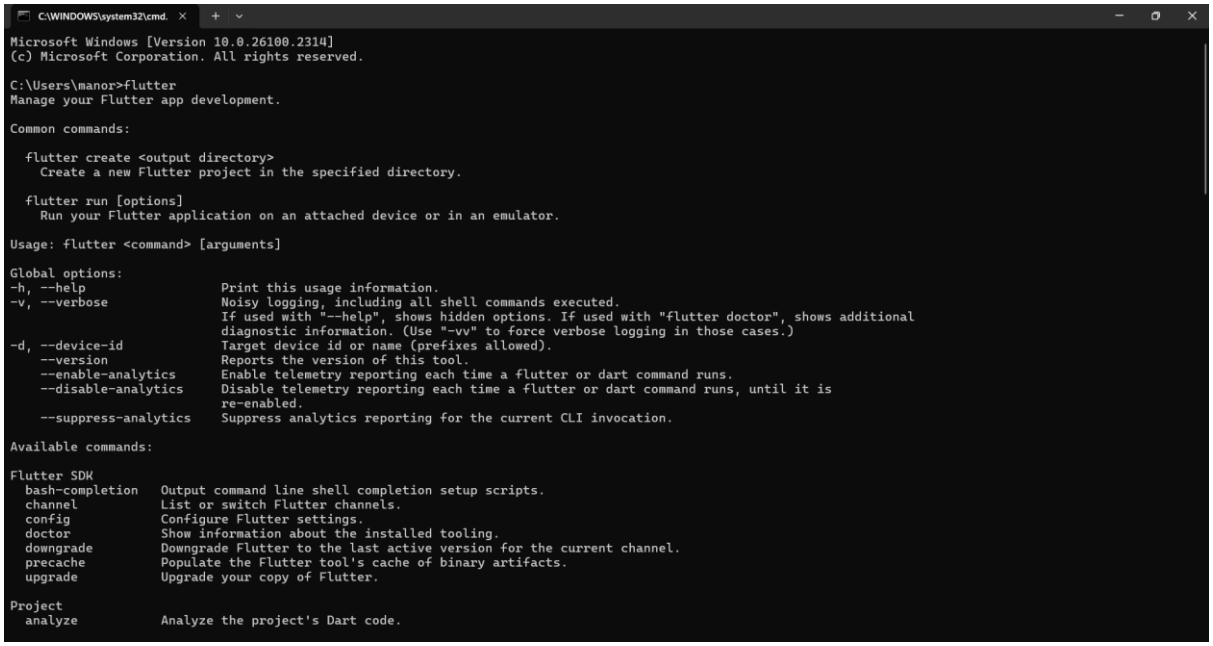
Right-click on the Start Menu > System > Advanced system settings > Environment Variables.

Under System Variables, find Path and click Edit.

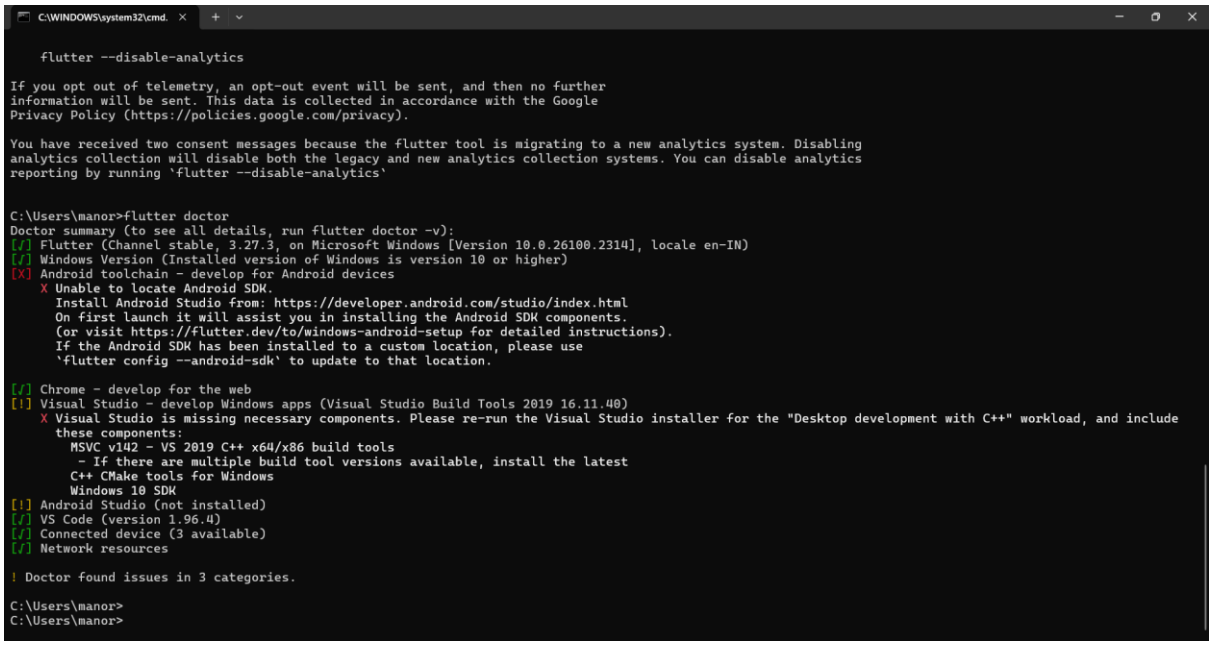
Add the full path to the flutter/bin directory (e.g., C:\flutter\bin).



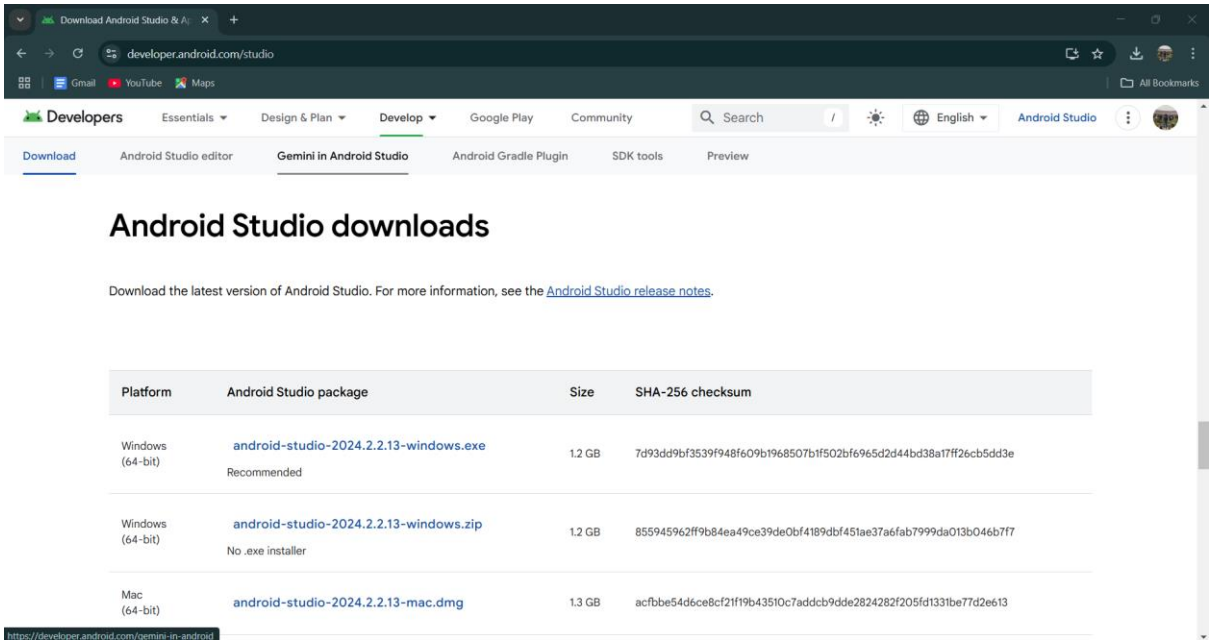
Step 6 : - Now, run the $ flutter command in command prompt.



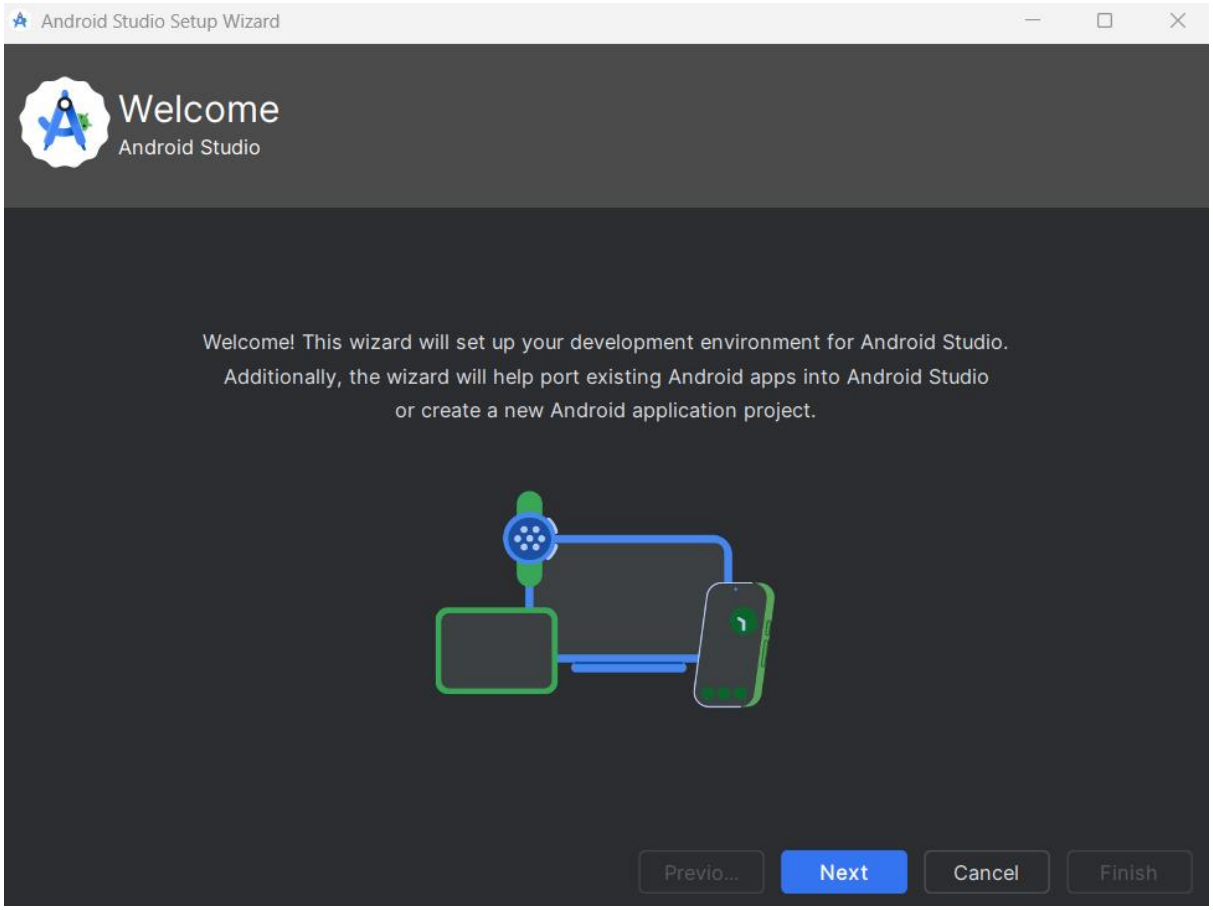
Step 7:- Run the $ flutter doctor command. This command checks for all the requirements of Flutter app development and displays a report of the status of your Flutter installation



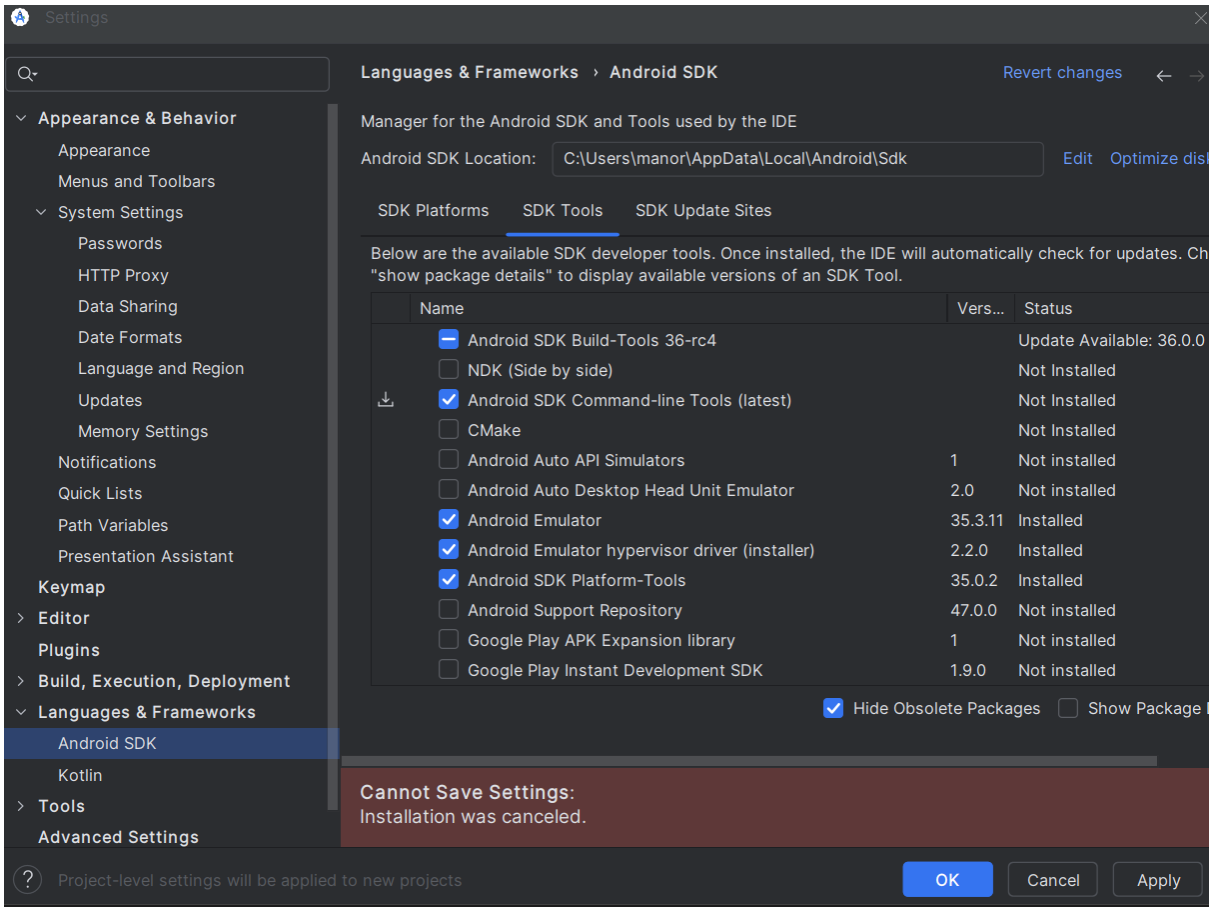
Step 8 : - Go to Android Studio and download the installer.



Step 9: - When the download is complete, open the .exe file and run it. Follow the steps of the installation wizard. Once the installation wizard completes, you will get the following screen.

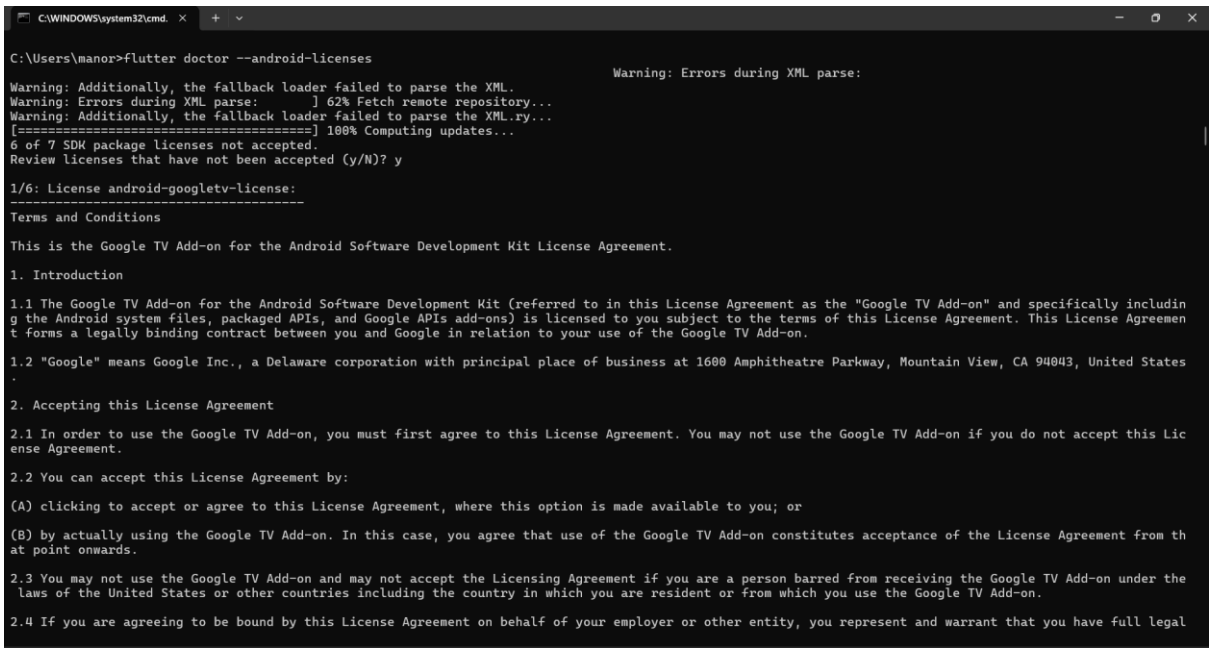


Step 10: - Go to Preferences > Appearance & Behavior > System Settings > Android SDK. Select the SDK Tools tab and check Android SDK Command-line Tools and Install it.

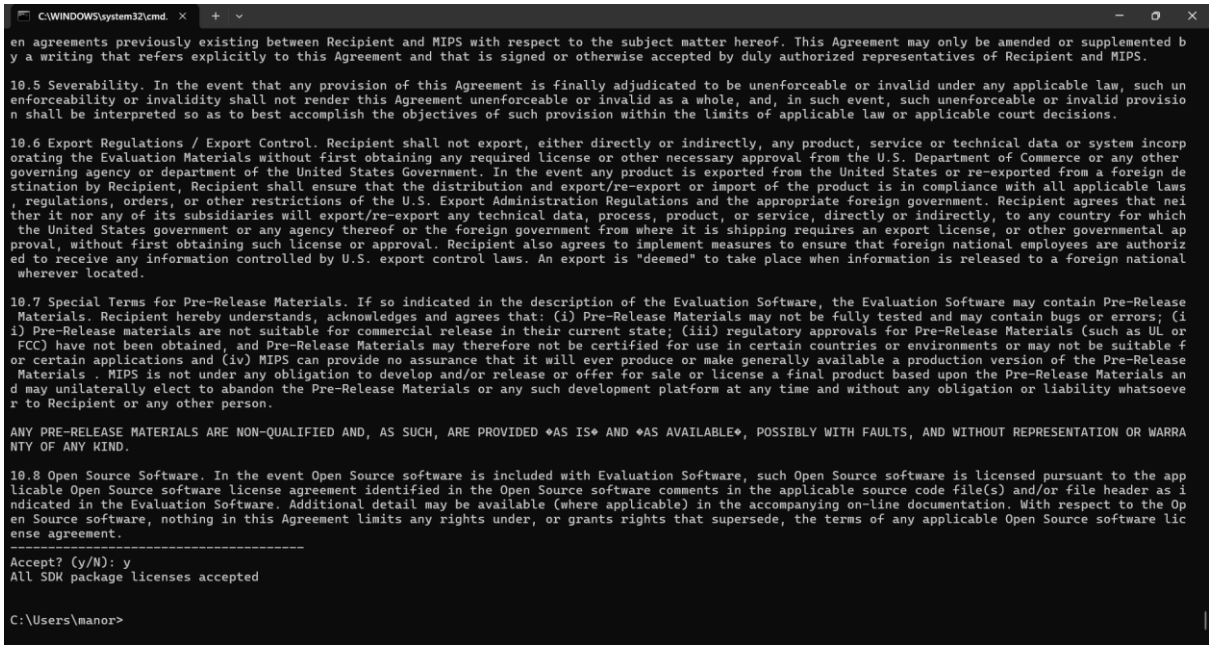


Step 9: - Open a terminal and run the following command

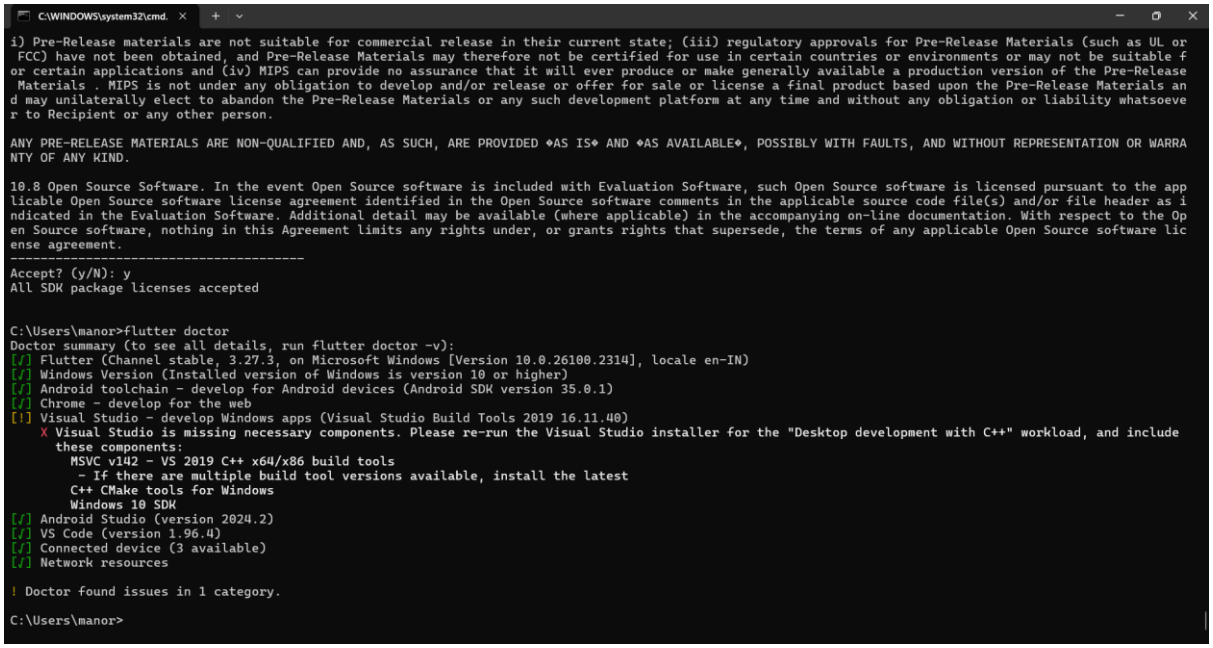
1)flutter doctor --android licenses



Accept all the licenses

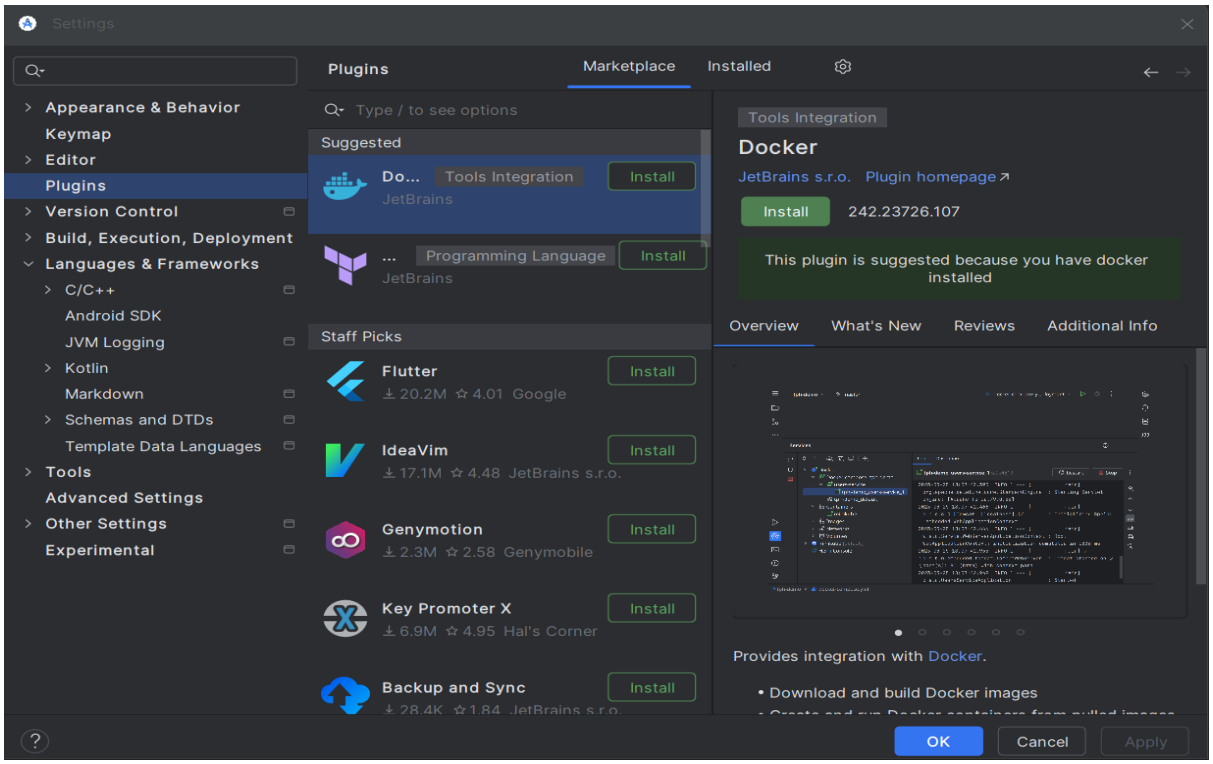


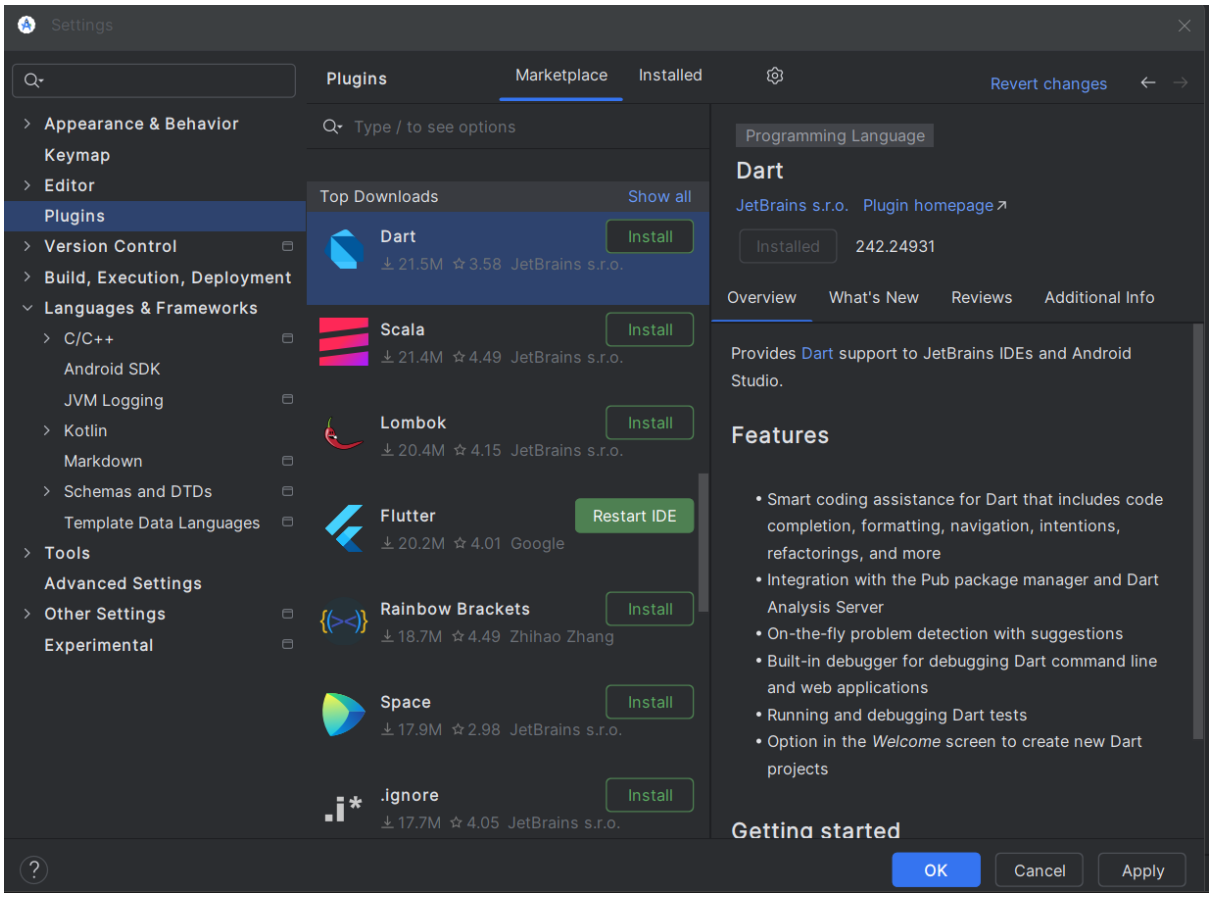
2)flutter doctor



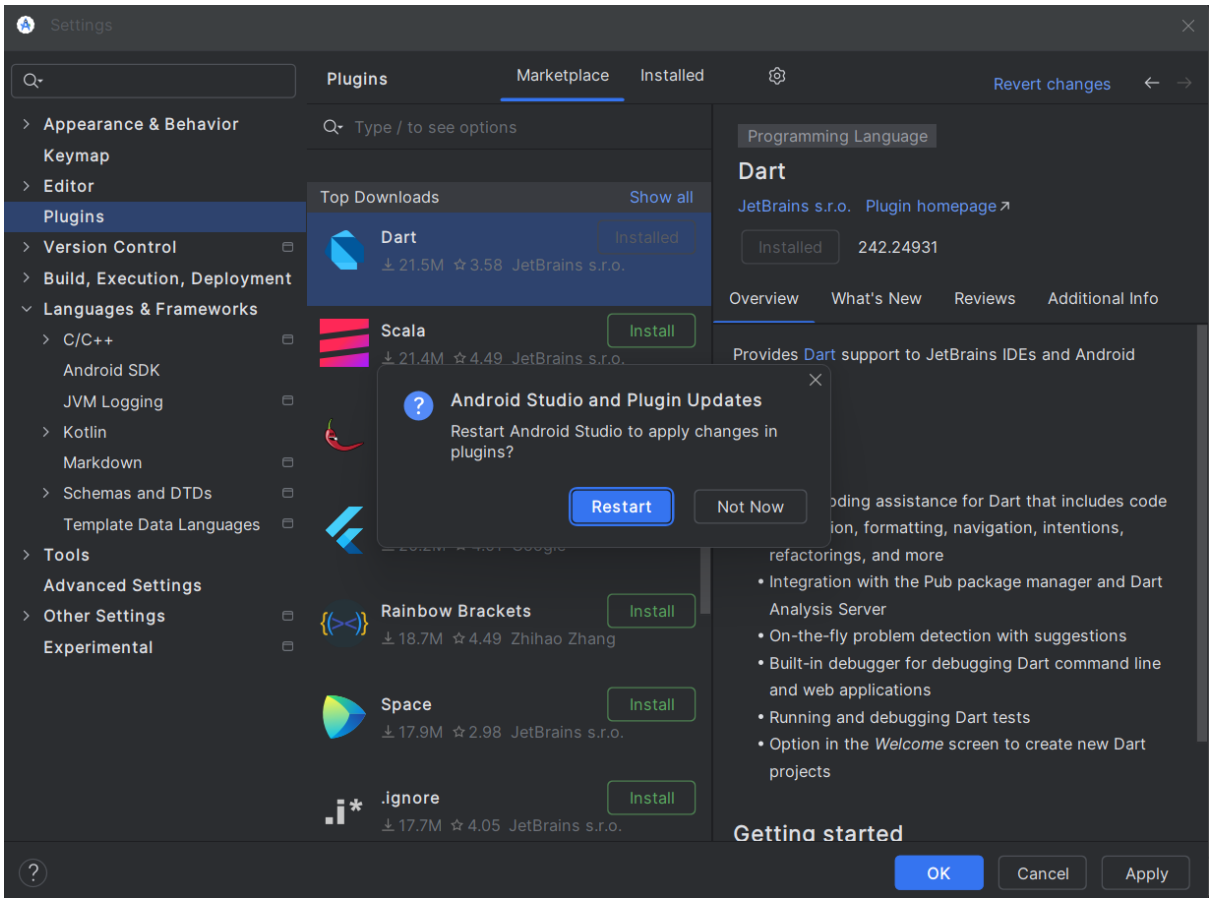
Step 10: - Next, you need to set up an Android emulator. It is responsible for running and testing the Flutter application

Step 11: - Now, install Flutter and Dart plugin for building Flutter application in Android Studio. These plugins provide a template to create a Flutter application, give an option to run and debug Flutter application in the Android Studio itselfOpen the Android Studio and then go to File->Settings->Plugins. Now, search the Flutter plugin. If found, select Flutter plugin and click install

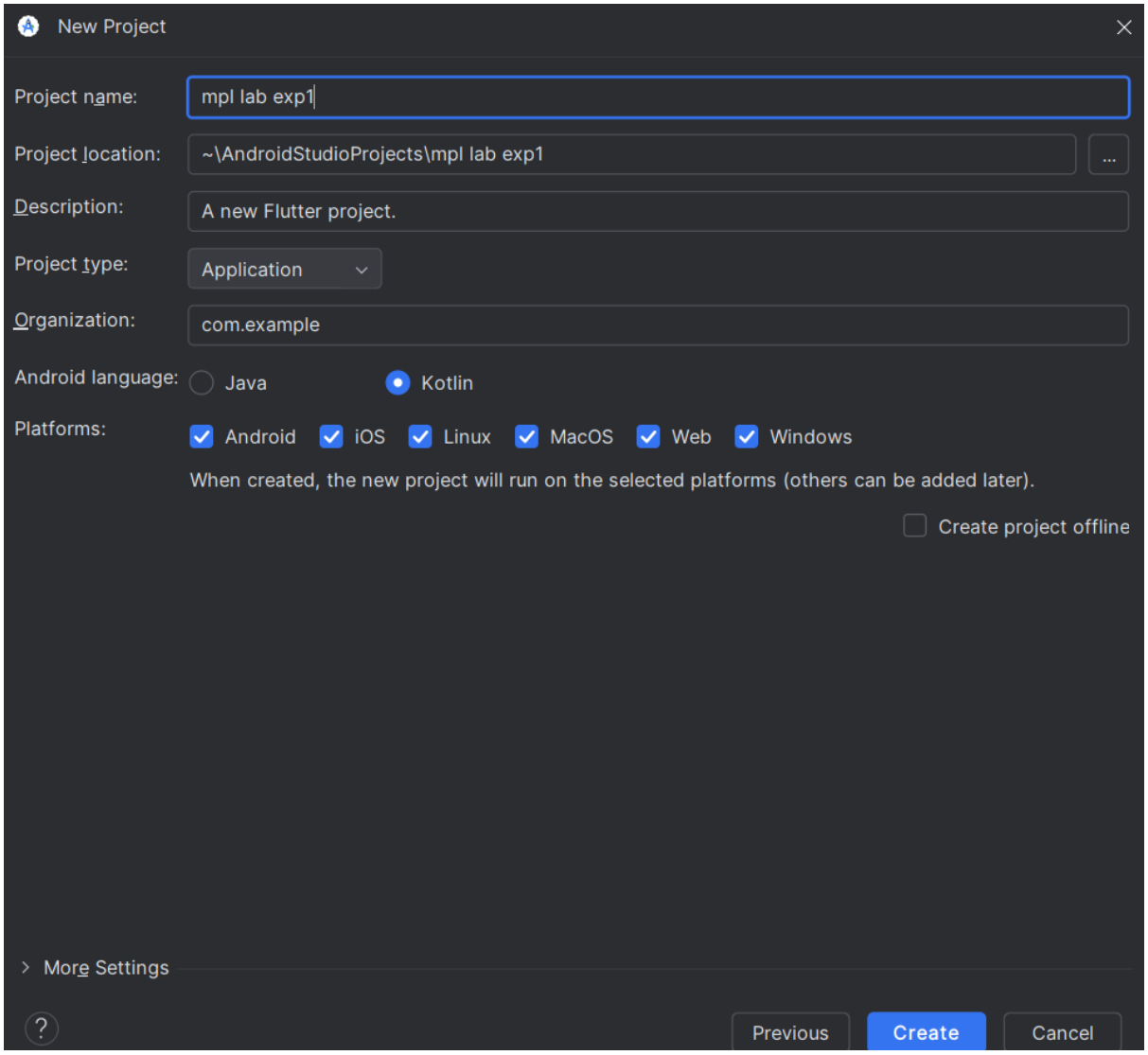


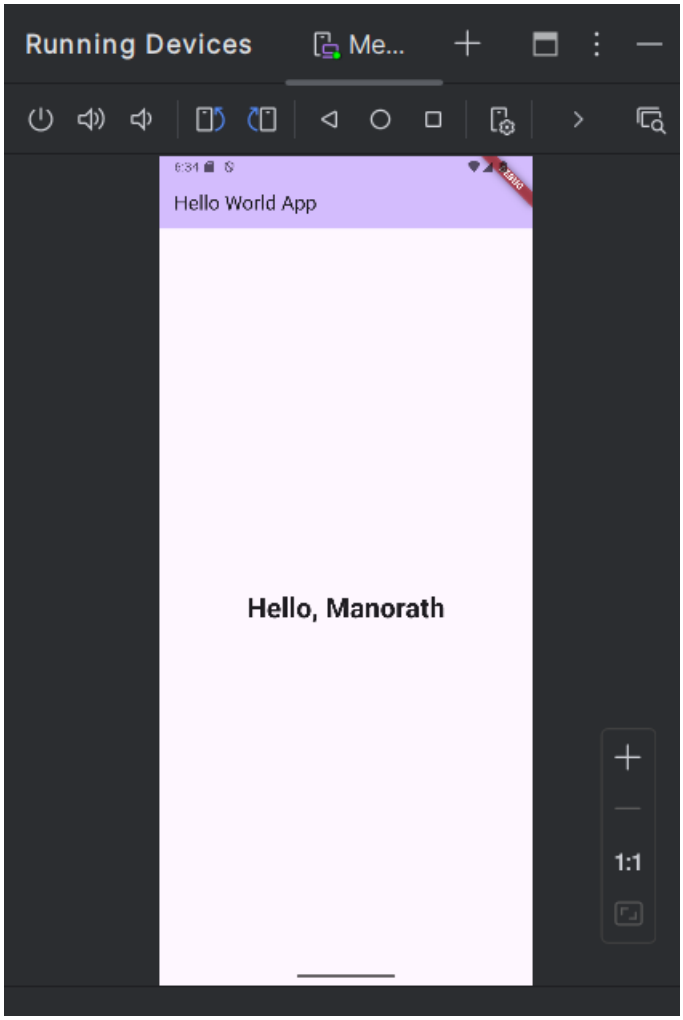


Step 12: - Restart the Android Studio



Step 13: - Go to File > New Project > Create Flutter Project, then select the project name and location, and click Next to proceed.


New Project
Project name:
mpl lab exp1
Project location:
~\AndroidStudioProjects\mpl lab exp1
...
Description:
A new Flutter project.
Project type:
Application
Organization:
com.example
Android language:
Java
Kotlin
Platforms:
Android
iOS
Linux
MacOS
Web
Windows
When created, the new project will run on the selected platforms (others can be added later).
Create project offline
More Settings
Previous
Create
Cancel


Running Devices
Me...
Hello World App
Hello, Manorath
1:1

Experiment-2

Name: Manorath Ital
Class : D15A
Roll no: 19

Aim:
To design Flutter UI by including common widgets.

Theory:
Flutter follows a widget-based approach where everything in the UI is a widget. Widgets can be classified into two main types:

- **Stateless Widgets:** Do not change their state once built (e.g., Text, Container).
- **Stateful Widgets:** Can update dynamically based on user interaction (e.g., TextField, Checkbox).

Commonly Used Widgets in Flutter

(a) Scaffold Widget

The `Scaffold` widget provides the basic structure for a Flutter app, including an `AppBar`, `Drawer`, `FloatingActionButton`, and `BottomNavigationBar`. It is a fundamental widget used to create a standard screen layout in Flutter.

(b) Container Widget

A `Container` is a box model widget that can hold other widgets. It is commonly used for adding padding, margins, borders, and background decorations.

(c) Row and Column Widgets

- **Row:** Arranges widgets horizontally.
- **Column:** Arranges widgets vertically.
  These two widgets are fundamental for designing layouts in Flutter.

(d) ListView Widget

The `ListView` widget is used for displaying a scrollable list of items. It is useful for showing large amounts of data dynamically.

(e) Stack Widget

The `Stack` widget is used to place widgets on top of each other. This is useful for creating overlapping UI elements such as banners, profile images, or layered designs.

(f) ElevatedButton Widget

The `ElevatedButton` widget is used for clickable buttons with a raised effect. It is a commonly used button in Flutter applications.

(g) TextField Widget

The `TextField` widget is used to take user input, such as entering a name, email, or password. It is commonly used in forms and authentication screens.

Home Page

This home page is designed using Flutter and consists of various widgets to create an

engaging UI for a movie app called **FilmyFun**.

Widgets Used:

1. **Scaffold Widget** – Provides the basic structure of the app.
2. **Container Widget** – Used for layout styling and decorations.
3. **Row & Column Widgets** – Organize UI elements horizontally and vertically.
4. **Text Widget** – Displays textual content.
5. **Image.asset Widget** – Loads images from local assets.
6. **CarouselSlider Widget** – Implements an image slider for featured movie posters.
7. **ListView Widget** – Displays a scrollable list of trending movies.
8. **Stack Widget** – Used for overlaying text and effects on movie images.

UI Features:

1. Header Section:

- Displays a welcome message: *"Hello, Manorath"*.
- Includes a wave emoji icon and a user profile picture.

2. App Title:

- The app name is displayed in a stylish way:
    - "Filmy" (White text)
    - "Fun" (Golden yellow text)

3. Carousel Image Slider:

- Showcases featured movie posters in a **carousel slider**.
- Users can swipe through movie images.

4. Trending Movies Section:

- Displays popular movies in a horizontal scrollable **ListView**.
- Each movie has:
    - A movie poster.
    - A title (e.g., *Infinity Wars*).
    - A genre (e.g., *Action, Adventure*).
    - A semi-transparent overlay for better readability.

Color Scheme:

- **Background:** Black for a cinematic feel.
- **Text Colors:** White and golden yellow for contrast.
- **Overlay Effects:** Black with transparency for better readability.

home.dart
Code:

```
import
'package:carousel_slider/carousel_slider.dar
t';

import 'package:flutter/material.dart';

class home extends StatefulWidget {

const home({super.key});

@override

State<home> createState() =>
_homeState();

}

class _homeState extends State<home> {
```

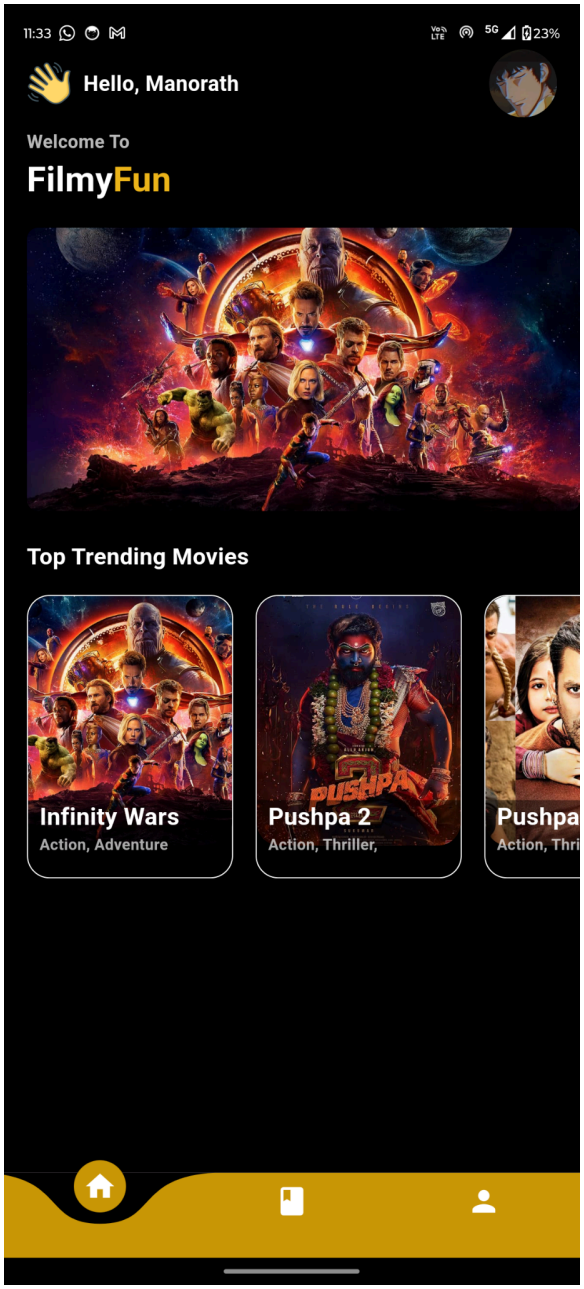
```
final List<String>imageUrls = [
"images/infinity.jpg",
"images/salman.jpg",
"images/shahrukhmovies.png",
];
@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.black,
body: Container(
margin: EdgeInsets.only(top: 40.0,left:
20.0),
child: Column(
crossAxisAlignment:
CrossAxisAlignment.start,
children: [
Row(children: [
Image.asset("images/wave.png", height: 40,
width: 40, fit: BoxFit.cover,),
SizedBox(width: 10.0,),
Text("Hello, Manorath",
style: TextStyle(
color: Colors.white,
fontSize: 22.0,
fontWeight: FontWeight.bold),
),
Spacer(),
Padding(
padding: const EdgeInsets.only(right: 20.0),
child: ClipRRect(
borderRadius: BorderRadius.circular(60),
child: Image.asset(
"images/Screenshot (55).png",
height: 60,
width: 60,
fit: BoxFit.cover,),
),
)
],
),
SizedBox(height: 10.0),
Text("Welcome To",
style: TextStyle(
color: Color.fromARGB(186, 255, 255, 255),
fontSize: 19.0,
fontWeight: FontWeight.bold),),
Row(children: [
Text("Filmy",
style: TextStyle(
color: Colors.white,
```

```
fontSize: 36.0,
fontWeight: FontWeight.bold),),
Text("Fun",
style: TextStyle(
color: Color(0xffedb41d),
fontSize: 36.0,
fontWeight: FontWeight.bold),)
],
),
SizedBox(height: 20.0,),
Center(child:
CarouselSlider(
items: imageUrls.map((url) {
return Builder(builder: (context) {
return Container(
width: MediaQuery.of(context).size.width, // Fixed incorrect comma
child: ClipRRect(
borderRadius: BorderRadius.circular(10),
child: Image.asset(url, fit: BoxFit.cover),
),
);
});
}).toList(), options: CarouselOptions(
height: 250,
autoPlay: false,
enlargeCenterPage: true,
aspectRatio: 16/9,
viewportFraction: 1.0
),// Fixed missing .toList()
)
),
SizedBox(height: 25.0,),
Text("Top Trending Movies",
style: TextStyle(
color: Colors.white,
fontSize: 24.0,
fontWeight: FontWeight.bold),),
SizedBox(height: 20.0,),
Container(
height: 250,
child: ListView(
scrollDirection: Axis.horizontal,
children: [
Container(
decoration: BoxDecoration(border: Border.all(color: Colors.white),borderRadius: BorderRadius.circular(20)),
child: Stack(
children: [
```

```
ClipRRect(
borderRadius: BorderRadius.circular(20),
child: Image.asset("images/infinity.jpg",
height: 220,
width: 180,
fit: BoxFit.cover,
),
),
Container(
padding: EdgeInsets.only(left: 10.0),
margin: EdgeInsets.only(top: 180),
height: 220,
width: 180,
decoration:
BoxDecoration(color:
Colors.black45,
borderRadius:
BorderRadius.only(
bottomRight: Radius.circular(20),
bottomLeft: Radius.circular(20))),
child: Column(
crossAxisAlignment:
CrossAxisAlignment.start,
children: [
Text("Infinity Wars", style: TextStyle(color:
Colors.white, fontSize: 25.0, fontWeight:
FontWeight.bold),),
Text("Action, Adventure", style:
TextStyle(color: Color.fromARGB(173, 255,
255, 255), fontSize: 16.0, fontWeight:
FontWeight.bold),),
],
),
),
],
),
),
SizedBox(width: 20.0),
Container(
decoration: BoxDecoration(border:
Border.all(color:
Colors.white),borderRadius:
BorderRadius.circular(20)),
child: Stack(
children: [
ClipRRect(
borderRadius: BorderRadius.circular(20),
child: Image.asset("images/pushpa.jpg",
height: 220,
width: 180,
fit: BoxFit.cover,
),
```

```
),
Container(
padding: EdgeInsets.only(left: 10.0),
margin: EdgeInsets.only(top: 180),
height: 220,
width: 180,
decoration:
BoxDecoration(color:
Colors.black45,
borderRadius:
BorderRadius.only(
bottomRight: Radius.circular(20),
bottomLeft: Radius.circular(20))),
child: Column(
crossAxisAlignment:
CrossAxisAlignment.start,
children: [
Text("Pushpa 2", style: TextStyle(color:
Colors.white, fontSize: 25.0, fontWeight:
FontWeight.bold),),
Text("Action, Thriller,", style: TextStyle(color:
Color.fromARGB(173, 255, 255, 255),
fontSize: 16.0, fontWeight:
FontWeight.bold),),
],
),
),
],
),
),
SizedBox(width: 20.0),
Container(
decoration: BoxDecoration(border:
Border.all(color:
Colors.white),borderRadius:
BorderRadius.circular(20)),
child: Stack(
children: [
ClipRRect(
borderRadius: BorderRadius.circular(20),
child: Image.asset("images/salman.jpg",
height: 220,
width: 180,
fit: BoxFit.cover,
),
),
Container(
padding: EdgeInsets.only(left: 10.0),
margin: EdgeInsets.only(top: 180),
height: 220,
width: 180,
decoration:
BoxDecoration(color:
```

```
Colors.black45,
borderRadius:
BorderRadius.only(
bottomRight: Radius.circular(20),
bottomLeft: Radius.circular(20))),
child: Column(
crossAxisAlignment:
CrossAxisAlignment.start,
children: [
Text("Pushpa 2", style: TextStyle(color:
Colors.white, fontSize: 25.0, fontWeight:
FontWeight.bold),),
Text("Action, Thriller,", style: TextStyle(color:
Color.fromARGB(173, 255, 255, 255),
fontSize: 16.0, fontWeight:
FontWeight.bold),),
],
),
),
],
),
)
],
),
)
]),
));
}
}
```



EXPERIMENT NO: - 03

Manorath Ital

D15A/19

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

Theory: -

Incorporating Icons, Images, and Custom Fonts into FilmyFun App

FilmyFun is a user-friendly movie ticket booking app that allows users to easily select showtimes and reserve tickets. A crucial aspect of creating an engaging and visually appealing app is incorporating various visual elements, such as icons, images, and custom fonts. These elements enhance the overall user experience and help the app stand out.

Like most modern mobile applications, FilmyFun uses resources such as assets and code to provide a dynamic user experience. Assets, such as images, icons, and custom fonts, are integral in delivering a rich and intuitive interface. These resources are bundled with the app and are made available during runtime.

Visual Elements and Their Role in the App

Visual elements—icons, images, and fonts—play an essential role in app development. These elements are used to improve usability, convey information quickly, and enhance branding.

1. **Enhanced User Experience**: Icons and images make the app visually attractive and easier to navigate.
2. **Information Conveyance**: Icons help reduce text and convey actions intuitively.
3. **Branding**: Custom icons and images align with FilmyFun's branding, making it memorable and recognizable.

Incorporating Icons in FilmyFun

Icons serve as intuitive visual shortcuts, providing a quick way for users to identify actions and navigate the app. Flutter provides a wide variety of built-in icons through the Icons class, and custom icons can be added via third-party packages like flutter_launcher_icons or font_awesome_flutter.

Example of adding an icon:

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

Adding Images in FilmyFun

Images are crucial for enhancing the visual experience in FilmyFun. Flutter allows for images to be added from multiple sources:

1. **Assets (Stored Locally)**
    - To add an image stored locally in the project, place the image in the assets/images/ folder.
    - Declare the image path in the pubspec.yaml file.

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

- Display the image in the app using the Image.asset method:

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

Incorporating Custom Fonts in FilmyFun

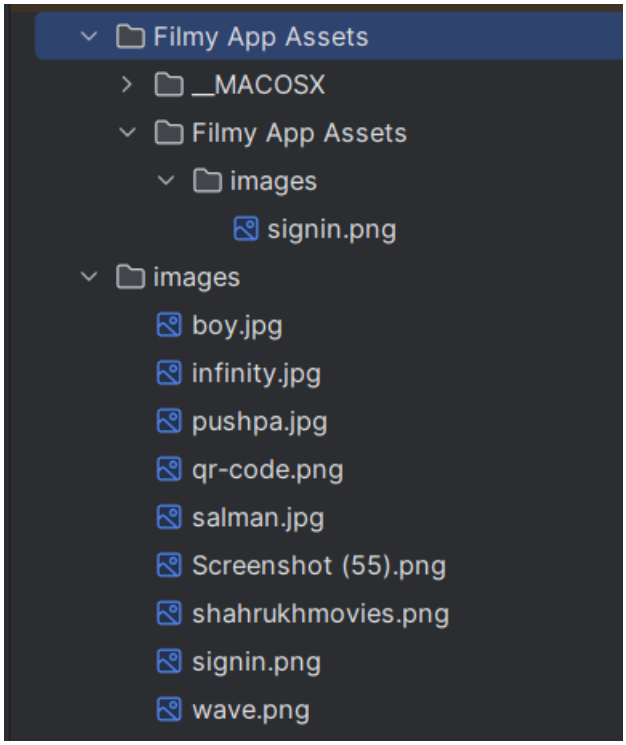
While Flutter's default font is **Roboto**, you may want to add custom fonts to align with your app's branding and overall design aesthetic. Adding custom fonts is simple:

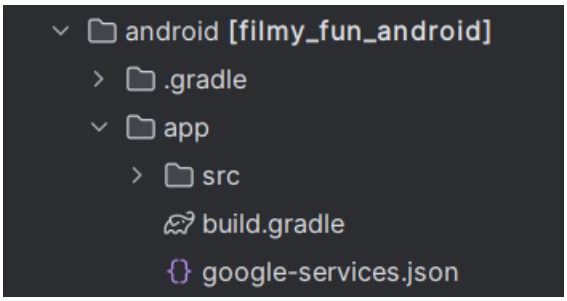
1. **Download the font** and place it in the assets/fonts/ folder of the project.
2. **Declare the font** in the pubspec.yaml file.

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

3. **Use the custom font** in your app:

```
Text(
  'Welcome to FilmyFun!',
  style: TextStyle(fontFamily: 'CustomFont', fontSize: 24),
);
```





Pubspec.yaml

```
name: filmy_fun
description: "FilmyFun-Movie Ticket Booking App"
# The following line prevents the package from being accidentally published to
# pub.dev using `flutter pub publish`. This is preferred for private packages.
publish_to: 'none' # Remove this line if you wish to publish to pub.dev

# The following defines the version and build number for your application.
# A version number is three numbers separated by dots, like 1.2.43
# followed by an optional build number separated by a +.
# Both the version and the builder number may be overridden in flutter
# build by specifying --build-name and --build-number, respectively.
# In Android, build-name is used as versionName while build-number used as versionCode.
# Read more about Android versioning at https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString while build-number is used as
CFBundleVersion.
# Read more about iOS versioning at
#
https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReferen
ce/Articles/CoreFoundationKeys.html
# In Windows, build-name is used as the major, minor, and patch parts
# of the product and file versions while build-number is used as the build suffix.
version: 1.0.0+1

environment:
  sdk: ^3.6.2

# Dependencies specify other packages that your package needs in order to work.
```

```
# To automatically upgrade your package dependencies to the latest versions
# consider running `flutter pub upgrade --major-versions`. Alternatively,
# dependencies can be manually updated by changing the version numbers below to
# the latest version available on pub.dev. To see which dependencies have newer
# versions available, run `flutter pub outdated`.
dependencies:
  flutter:
    sdk: flutter
  carousel_slider: ^5.0.0
  curved_navigation_bar: ^1.0.3 # Use the latest version
  intl: ^0.18.0
  firebase_auth: ^5.4.2
  cloud_firestore: ^5.6.3
  random_string: ^2.3.1
  firebase_core: ^3.11.0
  shared_preferences:
  flutter_stripe:
  http:


  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.8

dev_dependencies:
  flutter_test:
    sdk: flutter

  # The "flutter_lints" package below contains a set of recommended lints to
  # encourage good coding practices. The lint set provided by the package is
  # activated in the `analysis_options.yaml` file located at the root of your
  # package. See that file for information about deactivating specific lint
  # rules and activating additional ones.
  flutter_lints: ^5.0.0

# For information on the generic Dart part of this file, see the
# following page: https://dart.dev/tools/pub/pubspec

# The following section is specific to Flutter packages.
flutter:

  # The following line ensures that the Material Icons font is
  # included with your application, so that you can use the icons in
  # the material Icons class.
  uses-material-design: true

  # To add assets to your application, add an assets section, like this:
  assets:
```

```
  - images/
 #   - images/a_dot_ham.jpeg

 # An image asset can refer to one or more resolution-specific "variants", see
 # https://flutter.dev/to/resolution-aware-images

 # For details regarding adding assets from package dependencies, see
 # https://flutter.dev/to/asset-from-package

 # To add custom fonts to your application, add a fonts section here,
 # in this "flutter" section. Each entry in this list should have a
 # "family" key with the font family name, and a "fonts" key with a
 # list giving the asset and other descriptors for the font. For
 # example:
 # fonts:
 #   - family: Schyler
 #     fonts:
 #       - asset: fonts/Schyler-Regular.ttf
 #       - asset: fonts/Schyler-Italic.ttf
 #         style: italic
 #   - family: Trajan Pro
 #     fonts:
 #       - asset: fonts/TrajanPro.ttf
 #       - asset: fonts/TrajanPro_Bold.ttf
 #         weight: 700
 #
 # For details regarding fonts from package dependencies,
 # see https://flutter.dev/to/font-from-package
```

Home.dart

```
import 'package:carousel_slider/carousel_slider.dart';
import 'package:filmy_fun/pages/detail_page.dart';
import 'package:flutter/material.dart';

class home extends StatefulWidget {
  const home({super.key});

  @override
  State<home> createState() => _homeState();
}

class _homeState extends State<home> {
  final List<String> imageUrls = [
    "images/infinity.jpg",
    "images/salman.jpg",
    "images/shahrukhmovies.png",
  ];
  @override
```

```
Widget build(BuildContext context) {
  return Scaffold(
     backgroundColor: Colors.black,
     body: Container(
       margin: EdgeInsets.only(top: 40.0, left: 20.0),
       child:
          Column(crossAxisAlignment: CrossAxisAlignment.start, children: [
        Row(
          children: [
           Image.asset(
             "images/wave.png",
             height: 40,
             width: 40,
             fit: BoxFit.cover,
           ),
           SizedBox(
             width: 10.0,
           ),
           Text(
             "Hello, Manorath",
             style: TextStyle(
                color: Colors.white,
                fontSize: 22.0,
                fontWeight: FontWeight.bold),
           ),
           Spacer(),
           Padding(
             padding: const EdgeInsets.only(right: 20.0),
             child: ClipRRect(
               borderRadius: BorderRadius.circular(60),
               child: Image.asset(
                 "images/Screenshot (55).png",
                 height: 60,
                 width: 60,
                 fit: BoxFit.cover,
               ),
             ),
           )
         ],
       ),
       SizedBox(height: 10.0),
       Text(
         "Welcome To",
         style: TextStyle(
            color: Color.fromARGB(186, 255, 255, 255),
            fontSize: 19.0,
            fontWeight: FontWeight.bold),
       ),
```

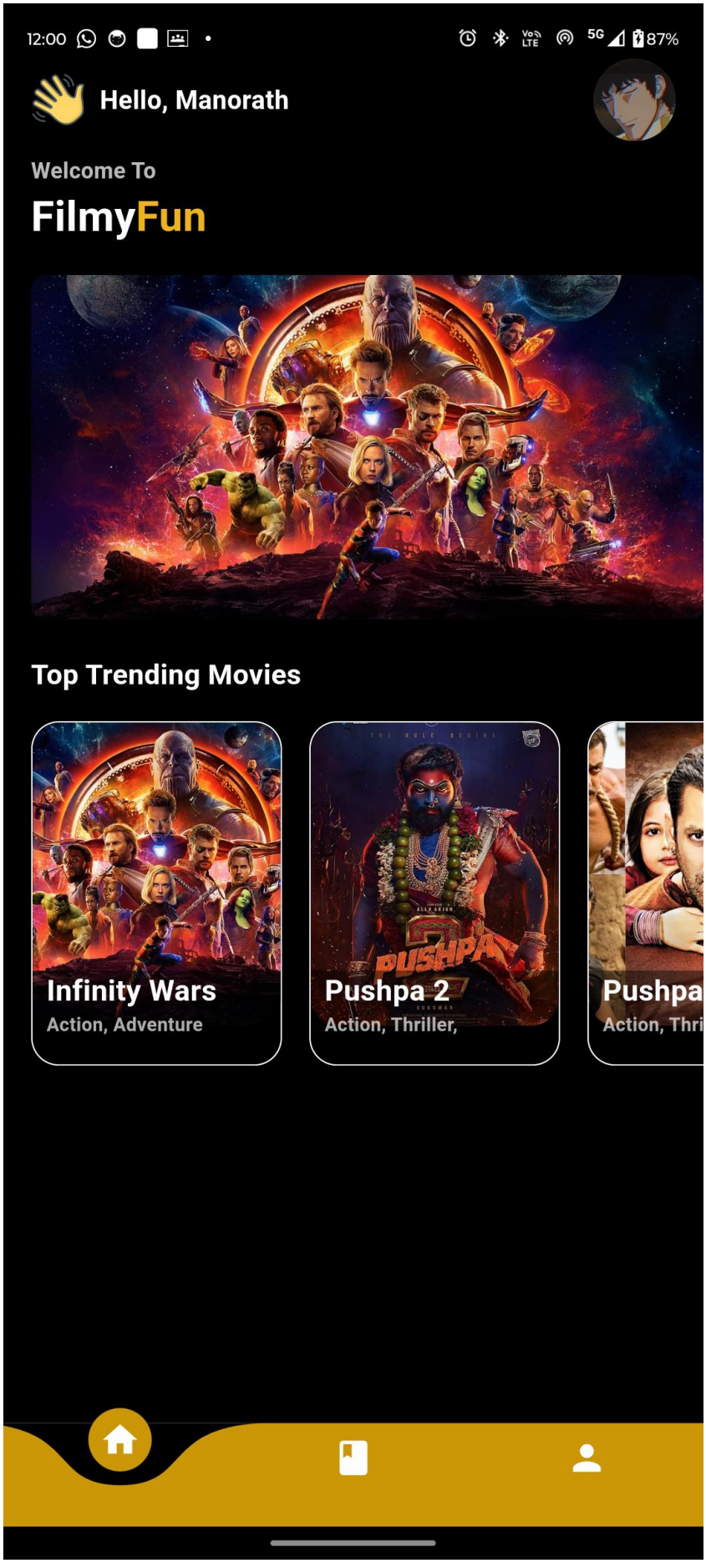
```
Row(
  children: [
    Text(
      "Filmy",
      style: TextStyle(
          color: Colors.white,
          fontSize: 36.0,
          fontWeight: FontWeight.bold),
    ),
    Text(
      "Fun",
      style: TextStyle(
          color: Color(0xffedb41d),
          fontSize: 36.0,
          fontWeight: FontWeight.bold),
    )
  ],
),
SizedBox(
  height: 20.0,
),
Center(
    child: CarouselSlider(
  items: imageUrls.map((url) {
    return Builder(builder: (context) {
      return Container(
        width: MediaQuery.of(context)
            .size
            .width, // Fixed incorrect comma
        child: ClipRRect(
          borderRadius: BorderRadius.circular(10),
          child: Image.asset(url, fit: BoxFit.cover),
        ),
      );
    });
  }).toList(),
  options: CarouselOptions(
      height: 250,
      autoPlay: false,
      enlargeCenterPage: true,
      aspectRatio: 16 / 9,
      viewportFraction: 1.0), // Fixed missing .toList()
)),
SizedBox(
  height: 25.0,
),
Text(
  "Top Trending Movies",
```

```
  style: TextStyle(
     color: Colors.white,
     fontSize: 24.0,
     fontWeight: FontWeight.bold),
 ),
 SizedBox(
  height: 20.0,
 ),
 Container(
  height: 250,
  child: ListView(
   scrollDirection: Axis.horizontal,
   children: [
    GestureDetector(
     onTap: () {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => DetailPage(
              image: "images/infinity.jpg",
              name: "Infinity Wars",
              shortdetail: "Action, Adventure",
              moviedetail:
                 "Avengers: Infinity War is a 2018 American superhero film based on the
Marvel Comics superhero team the Avengers Produced by Marvel Studios and distributed by Walt
Disney Studios Motion Pictures, it is the sequel to The Avengers (2012) and Avengers: Age of Ultron
(2015), and the 19th film in the Marvel Cinematic Universe (MCU)",
              price: '50',
             )));
     },
     child: Container(
      decoration: BoxDecoration(
        border: Border.all(color: Colors.white),
        borderRadius: BorderRadius.circular(20)),
      child: Stack(
       children: [
        ClipRRect(
         borderRadius: BorderRadius.circular(20),
         child: Image.asset(
          "images/infinity.jpg",
          height: 220,
          width: 180,
          fit: BoxFit.cover,
         ),
        ),
        Container(
         padding: EdgeInsets.only(left: 10.0),
         margin: EdgeInsets.only(top: 180),
```

```
          height: 220,
          width: 180,
          decoration: BoxDecoration(
            color: Colors.black45,
            borderRadius: BorderRadius.only(
              bottomRight: Radius.circular(20),
              bottomLeft: Radius.circular(20))),
          child: Column(
           crossAxisAlignment: CrossAxisAlignment.start,
           children: [
            Text(
             "Infinity Wars",
             style: TextStyle(
               color: Colors.white,
               fontSize: 25.0,
               fontWeight: FontWeight.bold),
            ),
            Text(
             "Action, Adventure",
             style: TextStyle(
               color: Color.fromARGB(173, 255, 255, 255),
               fontSize: 16.0,
               fontWeight: FontWeight.bold),
            ),
           ],
          ),
         ),
        ],
       ),
      ),
     ),
    ),
    SizedBox(width: 20.0),
    Container(
     decoration: BoxDecoration(
       border: Border.all(color: Colors.white),
       borderRadius: BorderRadius.circular(20)),
     child: Stack(
      children: [
       ClipRRect(
        borderRadius: BorderRadius.circular(20),
        child: Image.asset(
         "images/pushpa.jpg",
         height: 220,
         width: 180,
         fit: BoxFit.cover,
        ),
       ),
       Container(
```

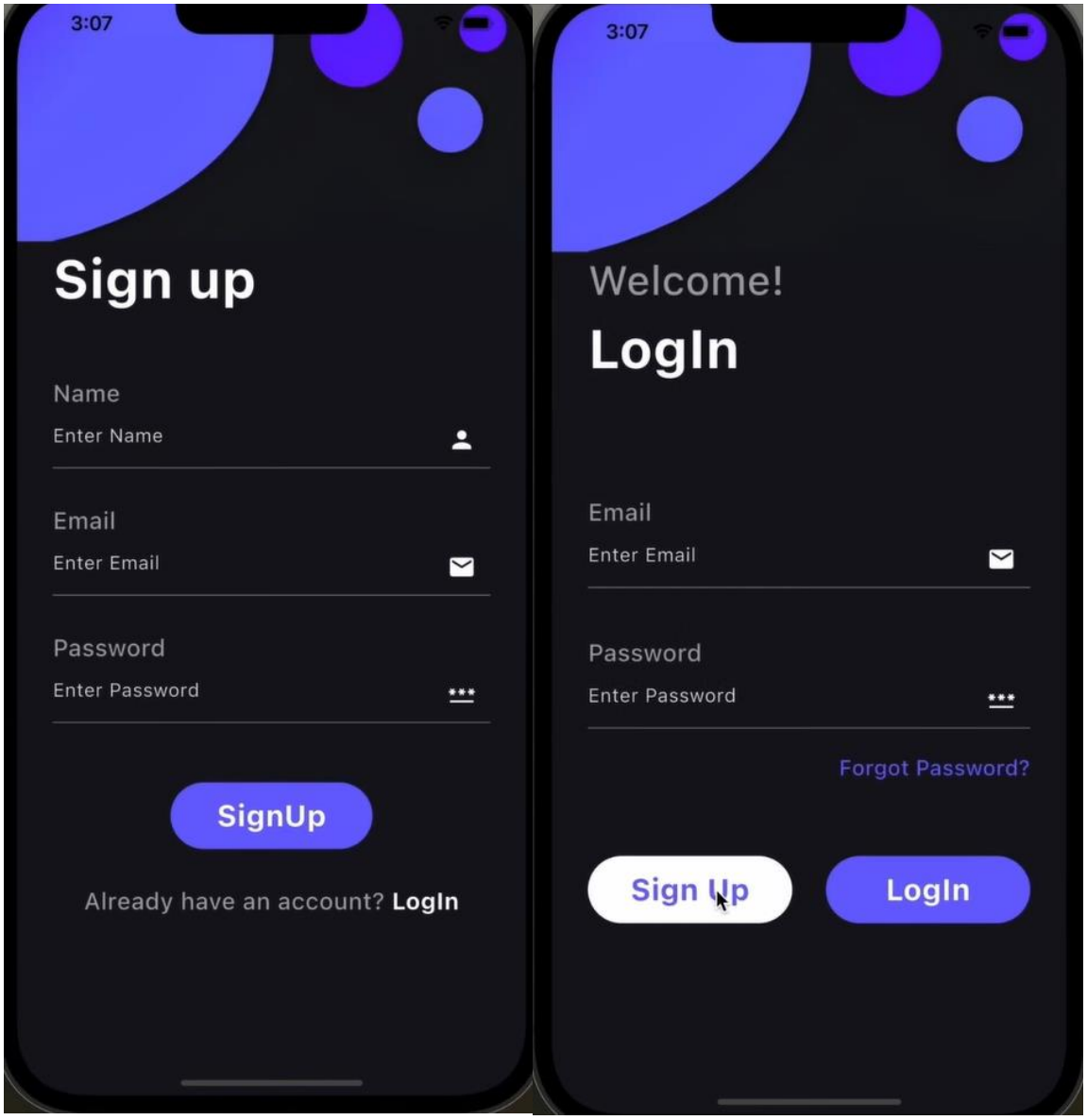
```
        padding: EdgeInsets.only(left: 10.0),
        margin: EdgeInsets.only(top: 180),
        height: 220,
        width: 180,
        decoration: BoxDecoration(
          color: Colors.black45,
          borderRadius: BorderRadius.only(
            bottomRight: Radius.circular(20),
            bottomLeft: Radius.circular(20))),
        child: Column(
         crossAxisAlignment: CrossAxisAlignment.start,
         children: [
          Text(
           "Pushpa 2",
           style: TextStyle(
             color: Colors.white,
             fontSize: 25.0,
             fontWeight: FontWeight.bold),
          ),
          Text(
           "Action, Thriller,",
           style: TextStyle(
             color: Color.fromARGB(173, 255, 255, 255),
             fontSize: 16.0,
             fontWeight: FontWeight.bold),
          ),
         ],
        ),
       ),
     ],
    ),
  ),
  SizedBox(width: 20.0),
  Container(
   decoration: BoxDecoration(
     border: Border.all(color: Colors.white),
     borderRadius: BorderRadius.circular(20)),
   child: Stack(
    children: [
     ClipRRect(
      borderRadius: BorderRadius.circular(20),
      child: Image.asset(
       "images/salman.jpg",
       height: 220,
       width: 180,
       fit: BoxFit.cover,
      ),
     ),
```

```
                    Container(
                      padding: EdgeInsets.only(left: 10.0),
                      margin: EdgeInsets.only(top: 180),
                      height: 220,
                      width: 180,
                      decoration: BoxDecoration(
                        color: Colors.black45,
                        borderRadius: BorderRadius.only(
                          bottomRight: Radius.circular(20),
                          bottomLeft: Radius.circular(20))),
                      child: Column(
                       crossAxisAlignment: CrossAxisAlignment.start,
                       children: [
                        Text(
                         "Pushpa 2",
                         style: TextStyle(
                           color: Colors.white,
                           fontSize: 25.0,
                           fontWeight: FontWeight.bold),
                        ),
                        Text(
                         "Action, Thriller,",
                         style: TextStyle(
                           color: Color.fromARGB(173, 255, 255, 255),
                           fontSize: 16.0,
                           fontWeight: FontWeight.bold),
                        ),
                       ],
                      ),
                     ),
                    ],
                   ),
                  )
                 ],
                ),
               )
              ]),
             ));
  }
}
```


12:00
87%
Hello, Manorath
Welcome To
FilmyFun
Top Trending Movies
Infinity Wars
Action, Adventure
Pushpa 2
Action, Thriller,
Pushpa
Action, Thri

ICONS:

```
buildInputField("Name", "Enter Name", Icons.person, namecontroller),
buildInputField("Email", "Enter Email", Icons.email, mailcontroller),
buildInputField("Password", "Enter Password", Icons.lock, passwordcontroller, obscureText: true),
const SizedBox(height: 30.0),
```



Conclusion

By incorporating icons, images, and custom fonts into the FilmyFun app, you can create an intuitive, visually appealing, and engaging user experience. These visual elements contribute to the overall aesthetic, enhance usability, and ensure your app stands out in a competitive market. Whether displaying movie posters, providing easy navigation with icons, or using custom fonts for unique branding, these resources will elevate the look and feel of your app.

Experiment 4

Name:Manorath Ital
Batch:D15A-19

Aim: To create interactive form

SignUp page
Code:

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

import 'package:filmy_fun/pages/login.dart';

class Signup extends StatefulWidget {

const Signup({super.key});

@override

State<Signup> createState() => _SignupState();

}

class _SignupState extends State<Signup> {

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Colors.black,

body: Padding(

padding: const EdgeInsets.symmetric(horizontal: 20.0), // Adds padding to align left

child: Column(

crossAxisAlignment: CrossAxisAlignment.start, // Aligns content to left

children: [

Image.asset("images/signin.png"),

const SizedBox(height: 20.0), // Adds spacing
```

```
const Text(
"Welcome!",
style: TextStyle(
color: Color.fromARGB(157, 255, 255, 255),
fontSize: 34.0,
fontWeight: FontWeight.w500,
),
),
const Text(
"SignUp",
style: TextStyle(
color: Colors.white,
fontSize: 45.0,
fontWeight: FontWeight.bold,
),
),
const SizedBox(height: 30.0), // Adjust spacing
const Text(
"Name",
style: TextStyle(
color: Colors.white,
fontSize: 20.0,
fontWeight: FontWeight.w500,
),
```

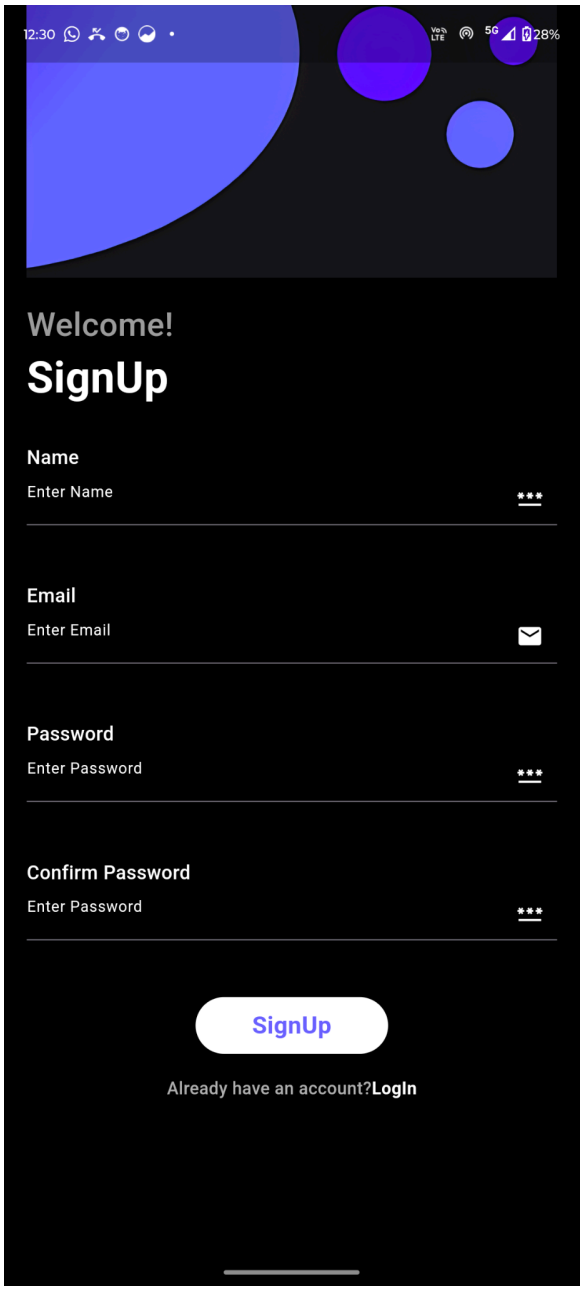
```
),
TextField(
decoration: InputDecoration(
hintText: "Enter Name",
hintStyle: TextStyle(
color: Colors.white,
),
suffixIcon: Icon(Icons.password,
color: Colors.white,),
),
),
const SizedBox(height: 50.0), // Adjust spacing
const Text(
"Email",
style: TextStyle(
color: Colors.white,
fontSize: 20.0,
fontWeight: FontWeight.w500,
),
),
TextField(
decoration: InputDecoration(
hintText: "Enter Email",
hintStyle: TextStyle(
```

```
color: Colors.white,
),
suffixIcon: Icon(Icons.email, color: Colors.white,),
),
),
const SizedBox(height: 50.0), // Adjust spacing
const Text(
"Password",
style: TextStyle(
color: Colors.white,
fontSize: 20.0,
fontWeight: FontWeight.w500,
),
),
TextField(
decoration: InputDecoration(
hintText: "Enter Password",
hintStyle: TextStyle(
color: Colors.white,
),
suffixIcon: Icon(Icons.password,
color: Colors.white,),
),
),
```

```
const SizedBox(height: 50.0), // Adjust spacing
const Text(
"Confirm Password",
style: TextStyle(
color: Colors.white,
fontSize: 20.0,
fontWeight: FontWeight.w500,
),
),
TextField(
decoration: InputDecoration(
hintText: "Enter Password",
hintStyle: TextStyle(
color: Colors.white,
),
suffixIcon: Icon(Icons.password,
color: Colors.white,),
),
),
SizedBox(height: 50.0,),
Center(
child: Column(
children: [
Container(
```

```
width: 170,
padding: EdgeInsets.all(10),
decoration: BoxDecoration(
color: Colors.white,
borderRadius: BorderRadius.circular(30),
),
child: Text(
"SignUp",
style: TextStyle(
color: Color(0xff6b63ff),
fontSize: 25.0,
fontWeight: FontWeight.bold,
),
textAlign: TextAlign.center,
),
),
SizedBox(height: 20.0), // Space between button and text
Row(
mainAxisAlignment: MainAxisAlignment.center,
children: [
Text(
"Already have an account?",
style: TextStyle(
color: Color.fromARGB(175, 255, 255, 255),
```

```
fontSize: 18.0,

fontWeight: FontWeight.w500,

),

),

GestureDetector(

onTap:() {Navigator.push(context, MaterialPageRoute(builder: (context)=> Login()));},

child: Text(

"LogIn",

style: TextStyle(

color: Colors.white,

fontSize: 18.0,

fontWeight: FontWeight.bold,

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


12:30
28%
Welcome!
SignUp
Name
Enter Name
Email
Enter Email
Password
Enter Password
Confirm Password
Enter Password
SignUp
Already have an account?LogIn

Login page

Code:

```
import 'package:filmy_fun/pages/bottomnav.dart';
import 'package:flutter/material.dart';
import 'package:filmy_fun/pages/signup.dart';
class Login extends StatefulWidget {
const Login({super.key});
@override
State<Login> createState() => _LoginState();
}
class _LoginState extends State<Login> {
@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.black,
body: Padding(
padding: const EdgeInsets.symmetric(horizontal: 20.0), // Adds padding to align left
child: Column(
crossAxisAlignment: CrossAxisAlignment.start, // Aligns content to left
children: [
Image.asset("images/signin.png"),
const SizedBox(height: 20.0), // Adds spacing
const Text(
"Welcome!",
style: TextStyle(
```

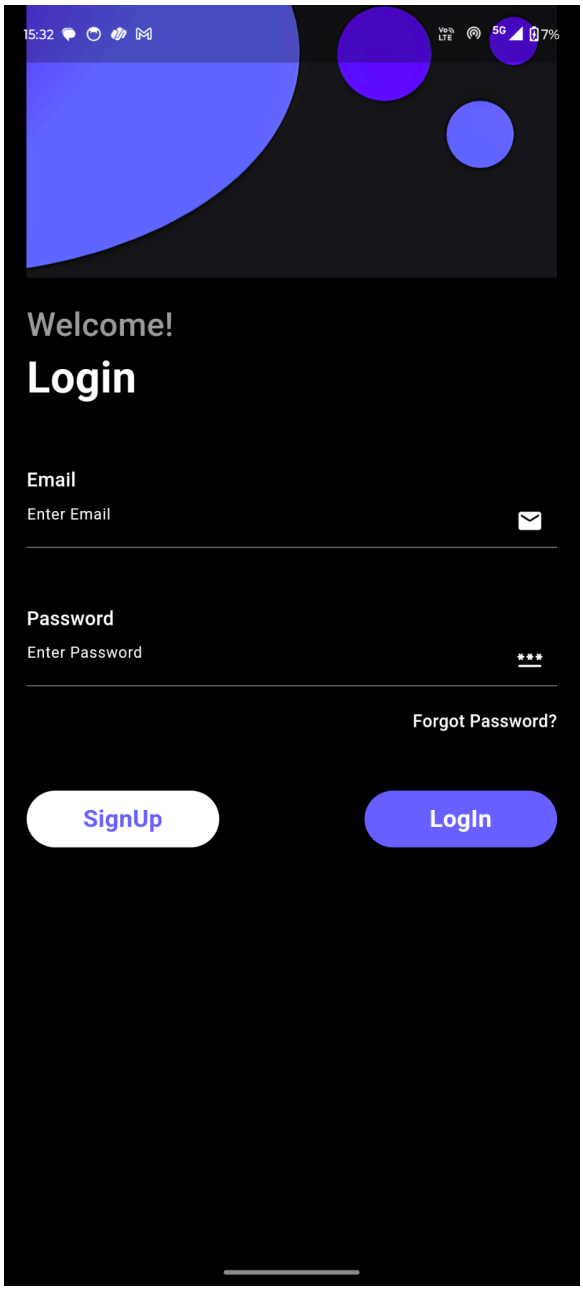
```
color: Color.fromARGB(157, 255, 255, 255),
fontSize: 34.0,
fontWeight: FontWeight.w500,
),
),
const Text(
"Login",
style: TextStyle(
color: Colors.white,
fontSize: 45.0,
fontWeight: FontWeight.bold,
),
),
const SizedBox(height: 50.0), // Adjust spacing
const Text(
"Email",
style: TextStyle(
color: Colors.white,
fontSize: 20.0,
fontWeight: FontWeight.w500,
),
),
TextField(
decoration: InputDecoration(
```

```
hintText: "Enter Email",
hintStyle: TextStyle(
color: Colors.white,
),
suffixIcon: Icon(Icons.email, color: Colors.white,),
),
),
const SizedBox(height: 50.0), // Adjust spacing
const Text(
"Password",
style: TextStyle(
color: Colors.white,
fontSize: 20.0,
fontWeight: FontWeight.w500,
),
),
TextField(
decoration: InputDecoration(
hintText: "Enter Password",
hintStyle: TextStyle(
color: Colors.white,
),
suffixIcon: Icon(Icons.password,
color: Colors.white,),
```

```
),
),
const SizedBox(height: 20.0),
Row(
mainAxisAlignment: MainAxisAlignment.end,
children: [
Text(
"Forgot Password?",
style: TextStyle(
color: Colors.white,
fontSize: 18.0,
fontWeight: FontWeight.w500,
),
)
],),
SizedBox(height: 50.0,),
Row(
mainAxisAlignment: MainAxisAlignment.spaceBetween,
children: [
Row(
mainAxisAlignment: MainAxisAlignment.center,
children: [
GestureDetector(
onTap:() {Navigator.push(context, MaterialPageRoute(builder: (context)=> Signup()));},
```

```
child: Container(
width: 170,
padding: EdgeInsets.all(10),
decoration: BoxDecoration(color: Colors.white,
borderRadius: BorderRadius.circular(30)),
child: Text("SignUp",style: TextStyle(color: Color(0xff6b63ff),
fontSize: 25.0,
fontWeight: FontWeight.bold),
textAlign: TextAlign.center,
),
),
),
],
),
Row(
mainAxisAlignment: MainAxisAlignment.center,
children: [
GestureDetector(
onTap: () {Navigator.push(context, MaterialPageRoute(builder: (context)=> Bottomnav()));},
child: Container(
width: 170,
padding: EdgeInsets.all(10),
decoration: BoxDecoration(
color: Color(0xff6b63ff),
```

```
borderRadius: BorderRadius.circular(30)),
child: Text("LogIn",style: TextStyle(color: Colors.white,
fontSize: 25.0,
fontWeight: FontWeight.bold),
textAlign: TextAlign.center,
),
),
),
],
)
],
)
),
),
);
}
}
```


15:32
7%
Welcome!
Login
Email
Enter Email
Password
Enter Password
***
Forgot Password?
SignUp
LogIn

EXPERIMENT NO: - 05

MANORATH ITAL

D15A/19

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(
      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');
```

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.

Tap Gestures

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

Long Press Gesture

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

Swipe and Drag Gestures

Flutter also provides swipe and drag gesture handling. The onHorizontalDragUpdate and

onVerticalDragUpdate callbacks are used for dragging gestures.

Code:-

main.dart

```
import 'package:filmy_fun/pages/detail_page.dart';
import 'package:filmy_fun/service/constant.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:filmy_fun/pages/login.dart';
import 'package:filmy_fun/pages/signup.dart';
import 'package:filmy_fun/pages/home.dart';
import 'package:filmy_fun/pages/bottomnav.dart';
import 'package:filmy_fun/pages/booking.dart';
import 'package:flutter_stripe/flutter_stripe.dart';

void main()async {
  WidgetsFlutterBinding.ensureInitialized();
  Stripe.publishableKey= publishedKey;
  await Firebase.initializeApp();
  runApp(const MyApp());
}

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'FilmyFun',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        // This is the theme of your application.
        //
        // TRY THIS: Try running your application with "flutter run". You'll see
        // the application has a purple toolbar. Then, without quitting the app,
        // try changing the seedColor in the colorScheme below to Colors.green
        // and then invoke "hot reload" (save your changes or press the "hot
        // reload" button in a Flutter-supported IDE, or press "r" if you used
        // the command line to start the app).
        //
        // Notice that the counter didn't reset back to zero; the application
        // state is not lost during the reload. To reset the state, use hot
        // restart instead.
        //
```

```
      // This works for code too, not just values: Most code changes can be
      // tested with just a hot reload.
      colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
      useMaterial3: true,
    ),
    home: Signup(),
   );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});

  // This widget is the home page of your application. It is stateful, meaning
  // that it has a State object (defined below) that contains fields that affect
  // how it looks.

  // This class is the configuration for the state. It holds the values (in this
  // case the title) provided by the parent (in this case the App widget) and
  // used by the build method of the State. Fields in a Widget subclass are
  // always marked "final".

  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      // This call to setState tells the Flutter framework that something has
      // changed in this State, which causes it to rerun the build method below
      // so that the display can reflect the updated values. If we changed
      // _counter without calling setState(), then the build method would not be
      // called again, and so nothing would appear to happen.
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    // This method is rerun every time setState is called, for instance as done
    // by the _incrementCounter method above.
    //
    // The Flutter framework has been optimized to make rerunning build methods
```

```
// fast, so that you can just rebuild anything that needs updating rather
// than having to individually change instances of widgets.
return Scaffold(
  appBar: AppBar(
    // TRY THIS: Try changing the color here to a specific color (to
    // Colors.amber, perhaps?) and trigger a hot reload to see the AppBar
    // change color while the other colors stay the same.
    backgroundColor: Theme.of(context).colorScheme.inversePrimary,
    // Here we take the value from the MyHomePage object that was created by
    // the App.build method, and use it to set our appbar title.
    title: Text(widget.title),
  ),
  body: Center(
    // Center is a layout widget. It takes a single child and positions it
    // in the middle of the parent.
    child: Column(
      // Column is also a layout widget. It takes a list of children and
      // arranges them vertically. By default, it sizes itself to fit its
      // children horizontally, and tries to be as tall as its parent.
      //
      // Column has various properties to control how it sizes itself and
      // how it positions its children. Here we use mainAxisAlignment to
      // center the children vertically; the main axis here is the vertical
      // axis because Columns are vertical (the cross axis would be
      // horizontal).
      //
      // TRY THIS: Invoke "debug painting" (choose the "Toggle Debug Paint"
      // action in the IDE, or press "p" in the console), to see the
      // wireframe for each widget.
      mainAxisAlignment: MainAxisAlignment.center,
      children: <Widget>[
        const Text(
          'You have pushed the button this many times:',
        ),
        Text(
          '$_counter',
          style: Theme.of(context).textTheme.headlineMedium,
        ),
      ],
    ),
  ),
  floatingActionButton: FloatingActionButton(
    onPressed: _incrementCounter,
    tooltip: 'Increment',
    child: const Icon(Icons.add),
  ), // This trailing comma makes auto-formatting nicer for build methods.
);
```

```
  }
}
```

Signup.dart

```
import 'package:filmy_fun/service/database.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:filmy_fun/pages/login.dart';
import 'package:filmy_fun/service/shared_pref.dart';
import 'package:random_string/random_string.dart';
import 'bottomnav.dart';

class Signup extends StatefulWidget {
  const Signup({super.key});

  @override
  State<Signup> createState() => _SignupState();
}

class _SignupState extends State<Signup> {
  String email = "", password = "", name = "";
  TextEditingController namecontroller = TextEditingController();
  TextEditingController passwordcontroller = TextEditingController();
  TextEditingController mailcontroller = TextEditingController();

  registration() async {
    if (passwordcontroller.text.isNotEmpty &&
        namecontroller.text.isNotEmpty &&
        mailcontroller.text.isNotEmpty) {
      try {
        UserCredential userCredential = await FirebaseAuth.instance
            .createUserWithEmailAndPassword(
            email: mailcontroller.text, password: passwordcontroller.text);
        String id = randomAlphaNumeric(10);
        Map<String, dynamic> userInfoMap = {
          "Name": namecontroller.text,
          "Email": mailcontroller.text,
          "Id": id,
          "Image": ""
        };
        await SharedpreferenceHelper().saveUserDislayName(namecontroller.text);
        await SharedpreferenceHelper().saveUserEmail(mailcontroller.text);
        await SharedpreferenceHelper().saveUserID(id);
        await SharedpreferenceHelper().saveUserImage("");
        await DatabaseMethods().addUserDetails(userInfoMap, id);
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
            backgroundColor: Colors.green,
            content: Text(
```

```
          "Registered Successfully!",
          style: TextStyle(
            color: Colors.white,
            fontWeight: FontWeight.bold,
            fontSize: 20.0),
        )));
    Navigator.pushReplacement(
      context, MaterialPageRoute(builder: (context) => Bottomnav()));
   } on FirebaseAuthException catch (e) {
    if (e.code == 'weak-password') {
     ScaffoldMessenger.of(context).showSnackBar(SnackBar(
        backgroundColor: Colors.orangeAccent,
        content: Text(
          "Password Provided is too Weak",
          style: TextStyle(fontSize: 18.0),
        )));
    } else if (e.code == "email-already-in-use") {
     ScaffoldMessenger.of(context).showSnackBar(SnackBar(
        backgroundColor: Colors.orangeAccent,
        content: Text(
          "Account Already exists",
          style: TextStyle(fontSize: 18.0),
        )));
    }
   }
  }
}

@override
Widget build(BuildContext context) {
 return Scaffold(
   backgroundColor: Colors.black,
   body: SingleChildScrollView(
    child: Padding(
     padding: const EdgeInsets.symmetric(horizontal: 20.0),
     child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
       const SizedBox(height: 50.0),
       Image.asset("images/signin.png"),
       const SizedBox(height: 20.0),
       const Text(
        "Welcome!",
        style: TextStyle(
         color: Color.fromARGB(157, 255, 255, 255),
         fontSize: 34.0,
         fontWeight: FontWeight.w500,
        ),
```

```
          ),
          const Text(
           "SignUp",
           style: TextStyle(
             color: Colors.white,
             fontSize: 45.0,
             fontWeight: FontWeight.bold,
           ),
          ),
          const SizedBox(height: 30.0),
          buildInputField("Name", "Enter Name", Icons.person, namecontroller),
          buildInputField("Email", "Enter Email", Icons.email, mailcontroller),
          buildInputField("Password", "Enter Password", Icons.lock, passwordcontroller, obscureText:
true),
          const SizedBox(height: 30.0),
          Center(
           child: Column(
             children: [
              GestureDetector(
               onTap: registration,
               child: GestureDetector(
                onTap: () {
                 if (namecontroller.text.isNotEmpty &&
                   mailcontroller.text.isNotEmpty &&
                   passwordcontroller.text.isNotEmpty) {
                  setState(() {
                   name = namecontroller.text;
                   email = mailcontroller.text;
                   password = passwordcontroller.text;
                  });
                  registration();
                 }
                },
                child: Container(
                 width: 170,
                 padding: const EdgeInsets.all(12),
                 decoration: BoxDecoration(
                  color: Colors.white,
                  borderRadius: BorderRadius.circular(30),
                 ),
                 child: const Text(
                  "SignUp",
                  style: TextStyle(
                   color: Color(0xff6b63ff),
                   fontSize: 25.0,
                   fontWeight: FontWeight.bold,
                  ),
                  textAlign: TextAlign.center,
```

```
                ),
              ),
            ),
          ),
          const SizedBox(height: 20.0),
          Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              const Text(
                "Already have an account?",
                style: TextStyle(
                  color: Color.fromARGB(175, 255, 255, 255),
                  fontSize: 18.0,
                  fontWeight: FontWeight.w500,
                ),
              ),
              GestureDetector(
                onTap: () {
                  Navigator.push(
                      context,
                      MaterialPageRoute(
                          builder: (context) => const Login()));
                },
                child: const Text(
                  " LogIn",
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 18.0,
                    fontWeight: FontWeight.bold,
                  ),
                ),
              ),
            ],
          ),
          const SizedBox(height: 30.0),
        ],
      ),
    ),
  ),
 );
}

Widget buildInputField(String label, String hint, IconData icon,
  TextEditingController controller,
  {bool obscureText = false}) {
```

```
    return Padding(
      padding: const EdgeInsets.only(bottom: 20.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(
            label,
            style: const TextStyle(
              color: Colors.white,
              fontSize: 20.0,
              fontWeight: FontWeight.w500,
            ),
          ),
          TextField(
            controller: controller,
            obscureText: obscureText,
            cursorColor: Colors.white, // Cursor color set to white
            style: const TextStyle(color: Colors.white), // Text color set to white
            decoration: InputDecoration(
              hintText: hint,
              hintStyle: const TextStyle(color: Colors.white), // Hint text color set to white
              suffixIcon: Icon(icon, color: Colors.white),
              enabledBorder: const UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white), // Underline color
              ),
              focusedBorder: const UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white), // Underline when focused
              ),
            ),
          ),
        ],
      ),
    );
  }
}
```

Login.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:filmy_fun/pages/bottomnav.dart';
import 'package:filmy_fun/pages/home.dart';
import 'package:filmy_fun/service/database.dart';
import 'package:filmy_fun/service/shared_pref.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:filmy_fun/pages/signup.dart';

class Login extends StatefulWidget {
```

```
  const Login({super.key});

  @override
  State<Login> createState() => _LoginState();
}

class _LoginState extends State<Login> {
  String email = "", password = "", myname = "", myid = "", myimage = "";
  TextEditingController passwordcontroller = TextEditingController();
  TextEditingController mailcontroller = TextEditingController();

  userLogin() async {
    try {
      await FirebaseAuth.instance.signInWithEmailAndPassword(
        email: email,
        password: password,
      );
      QuerySnapshot querySnapshot =
      await DatabaseMethods().getUserbyemail(email);
      myname = "${querySnapshot.docs[0]["Name"]}";
      myid = "${querySnapshot.docs[0]["Id"]}";
      myimage = "${querySnapshot.docs[0]["Image"]}";

      await SharedpreferenceHelper().saveUserImage(myimage);
      await SharedpreferenceHelper().saveUserEmail(email);
      await SharedpreferenceHelper().saveUserDislayName(myname);
      await SharedpreferenceHelper().saveUserID(myid);
      Navigator.push(context, MaterialPageRoute(builder: (context) => Bottomnav()));
    } on FirebaseAuthException catch (e) {
      String errorMessage = "";
      if (e.code == 'user-not-found') {
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
          content: Text("No user found for that email.",
          style: TextStyle(fontSize: 18.0, color: Colors.black),
          )));
      } else if (e.code == 'wrong-password') {
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
          backgroundColor: Colors.white,
          content: Text("Wrong password provided."
          )));
      }
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.black,
```

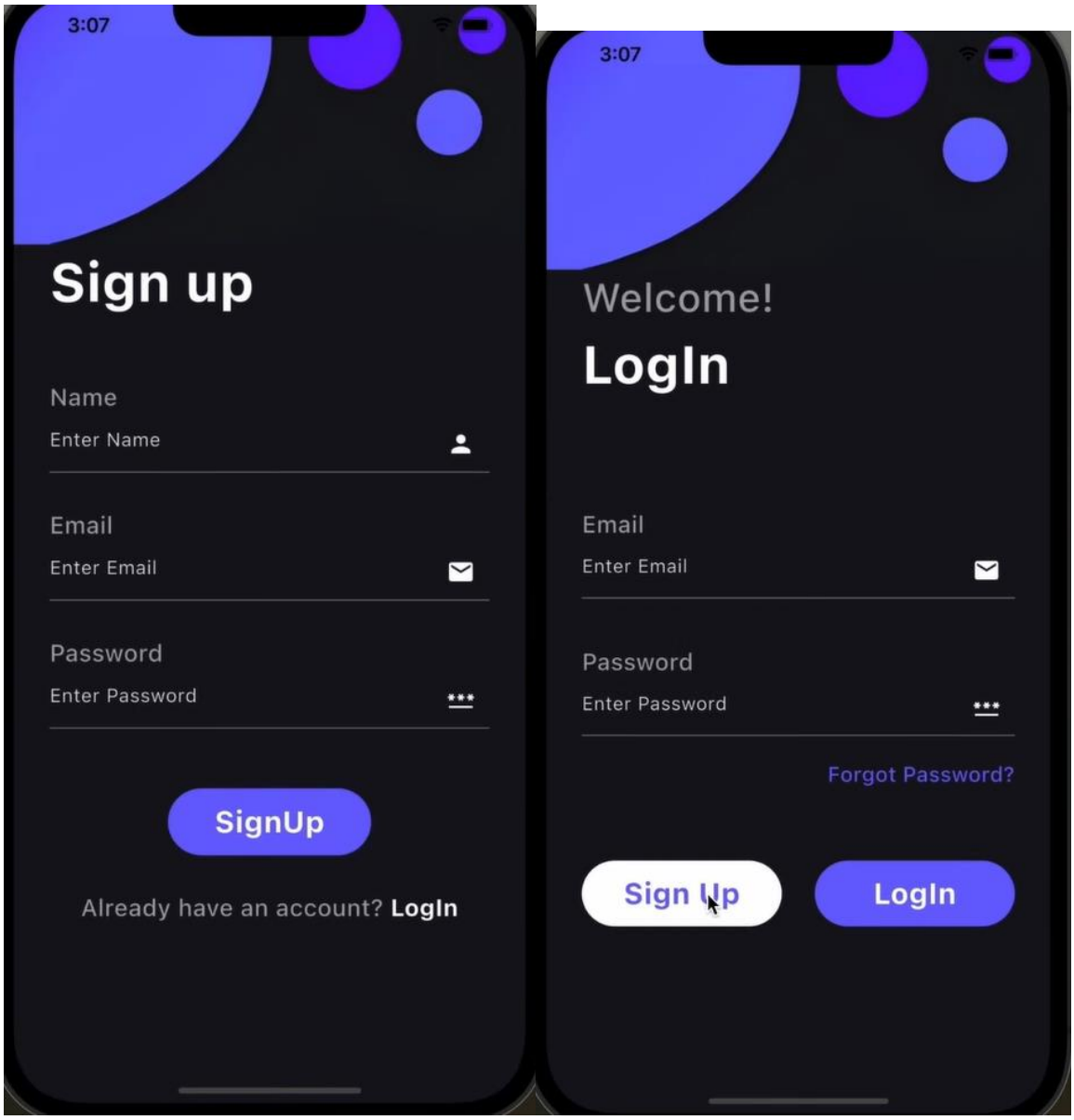
```
body: Padding(
  padding: const EdgeInsets.symmetric(horizontal: 20.0),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Image.asset("images/signin.png"),
      const SizedBox(height: 20.0),
      const Text(
        "Welcome!",
        style: TextStyle(
          color: Color.fromARGB(157, 255, 255, 255),
          fontSize: 34.0,
          fontWeight: FontWeight.w500,
        ),
      ),
      GestureDetector(
        onTap: () {
          if (mailcontroller.text.isNotEmpty &&
              passwordcontroller.text.isNotEmpty) {
            setState(() {
              email = mailcontroller.text;
              password = passwordcontroller.text;
              userLogin();
            });
          }
        },
        child: const Text(
          "Login",
          style: TextStyle(
            color: Colors.white,
            fontSize: 45.0,
            fontWeight: FontWeight.bold,
          ),
        ),
      ),
      const SizedBox(height: 50.0),
      const Text(
        "Email",
        style: TextStyle(
          color: Colors.white,
          fontSize: 20.0,
          fontWeight: FontWeight.w500,
        ),
      ),
      TextField(
        controller: mailcontroller,
        cursorColor: Colors.black,
        style: const TextStyle(color: Colors.white),
```

```
  decoration: const InputDecoration(
   hintText: "Enter Email",
   hintStyle: TextStyle(color: Colors.grey),
  ),
 ),
 const SizedBox(height: 50.0),
 const Text(
  "Password",
  style: TextStyle(
   color: Colors.white,
   fontSize: 20.0,
   fontWeight: FontWeight.w500,
  ),
 ),
 TextField(
  controller: passwordcontroller,
  cursorColor: Colors.black,
  style: const TextStyle(color: Colors.white),
  obscureText: true,
  decoration: const InputDecoration(
   hintText: "Enter Password",
   hintStyle: TextStyle(color: Colors.grey),
  ),
 ),
 const SizedBox(height: 20.0),
 Row(
  mainAxisAlignment: MainAxisAlignment.end,
  children: [
   Text(
    "Forgot Password?",
    style: TextStyle(
     color: Colors.white,
     fontSize: 18.0,
     fontWeight: FontWeight.w500,
    ),
   )
  ],
 ),
 const SizedBox(height: 50.0),
 Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
   GestureDetector(
    onTap: () {
     Navigator.push(context,
       MaterialPageRoute(builder: (context) => Signup()));
    },
    child: Container(
```

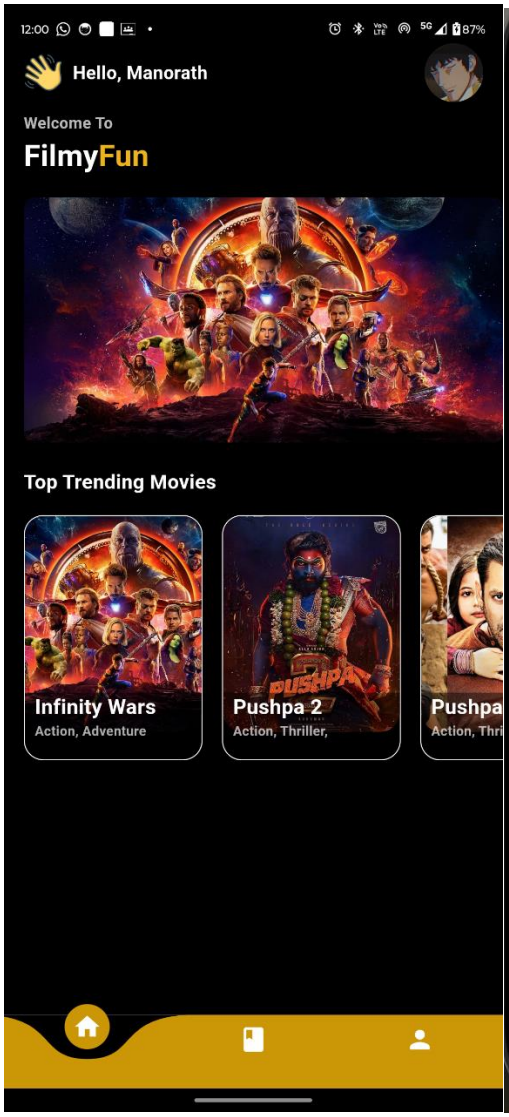
```
                width: 170,
                padding: EdgeInsets.all(10),
                decoration: BoxDecoration(
                  color: Colors.white,
                  borderRadius: BorderRadius.circular(30)),
                child: Text(
                 "SignUp",
                 style: TextStyle(
                   color: Color(0xff6b63ff),
                   fontSize: 25.0,
                   fontWeight: FontWeight.bold),
                 textAlign: TextAlign.center,
                ),
               ),
              ),
              GestureDetector(
               onTap: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder: (context) => const Bottomnav()));
               },
               child: Container(
                width: 170,
                padding: EdgeInsets.all(10),
                decoration: BoxDecoration(
                  color: Color(0xff6b63ff),
                  borderRadius: BorderRadius.circular(30)),
                child: Text(
                 "LogIn",
                 style: TextStyle(
                   color: Colors.white,
                   fontSize: 25.0,
                   fontWeight: FontWeight.bold),
                 textAlign: TextAlign.center,
                ),
               ),
              ),
             ],
            )
           ],
          ),
        ),
       );
      }
    }
```

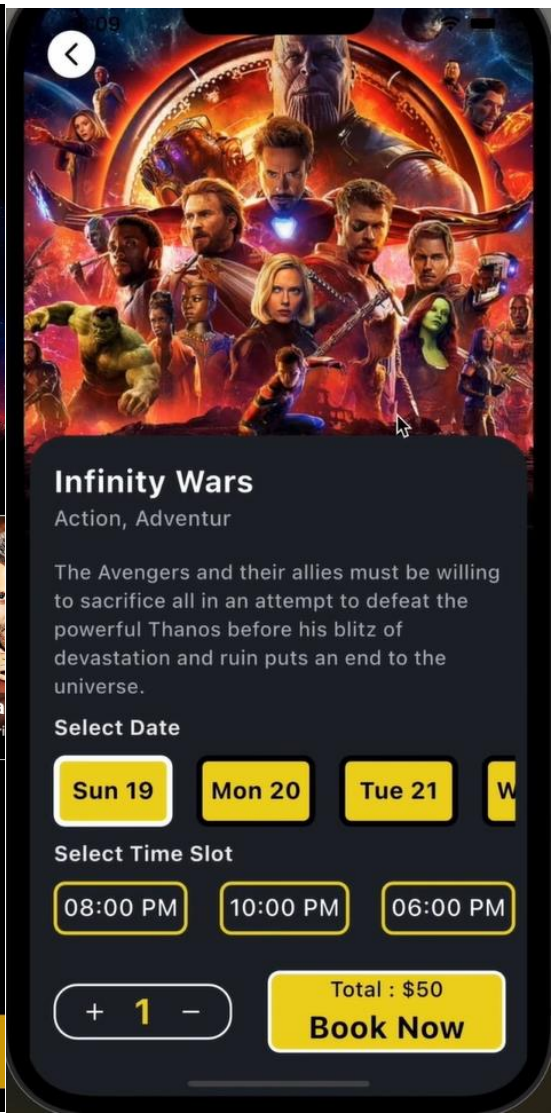
OUTPUT:

After clicking on Already have an account? It navitages to login



On Homepage after clicking on Movie image it navigates to Details page





EXPERIMENT NO: - 06

Name:- Manorath Ital
Class:- D15A **Roll:No: -** 19

AIM: - To connect Flutter UI with Firebase database.

Theory: -

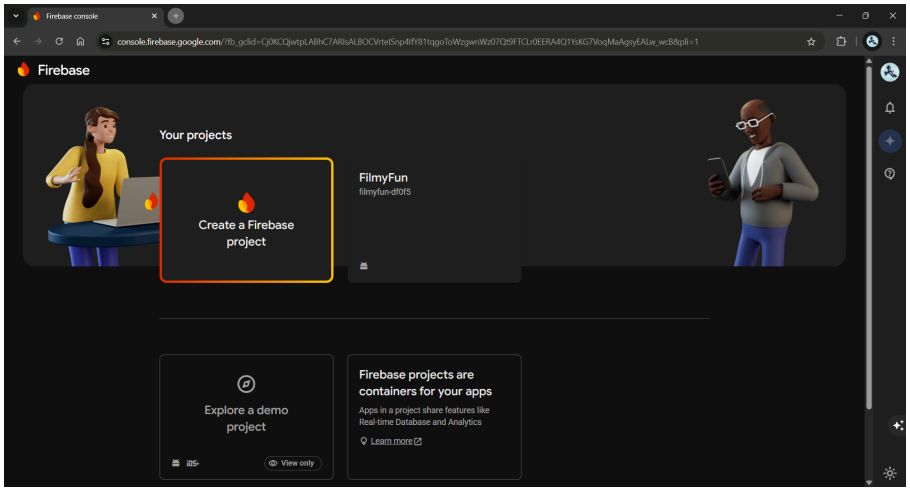
Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. Firebase, a Backend-as-a- Service (BaaS) platform, provides real-time database, authentication, and cloud storage services, making it a powerful backend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

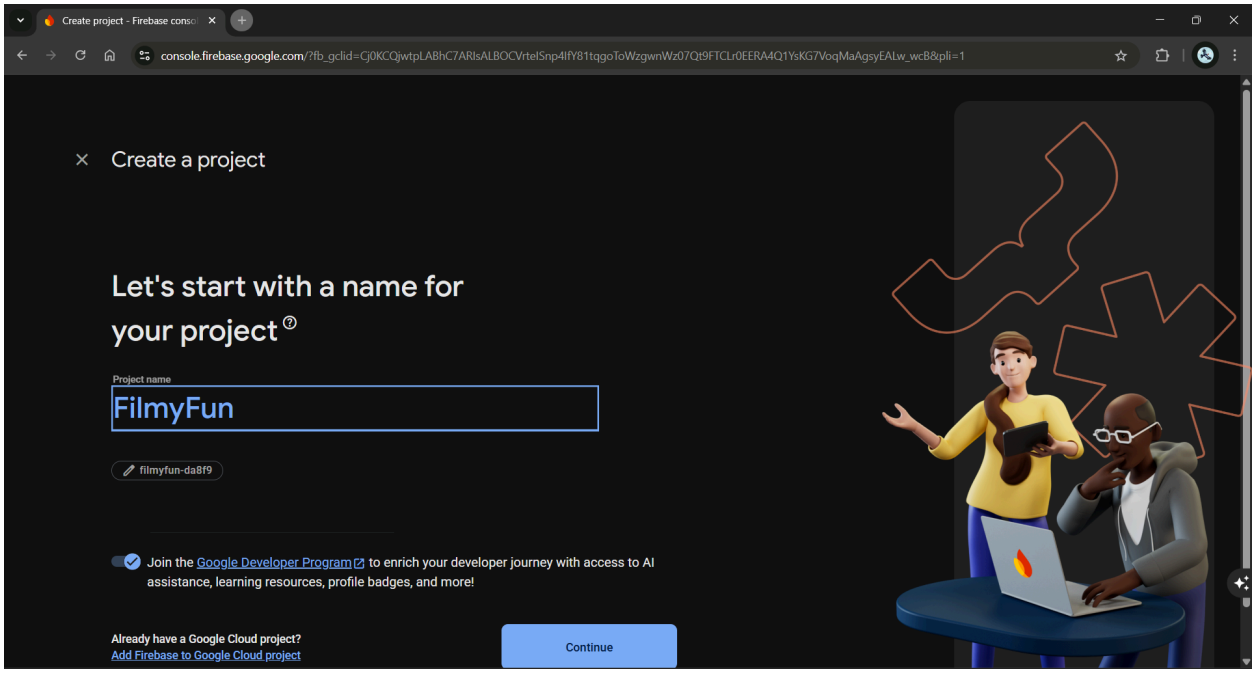
Steps to Connect Flutter UI with Firebase Database

Step 1:

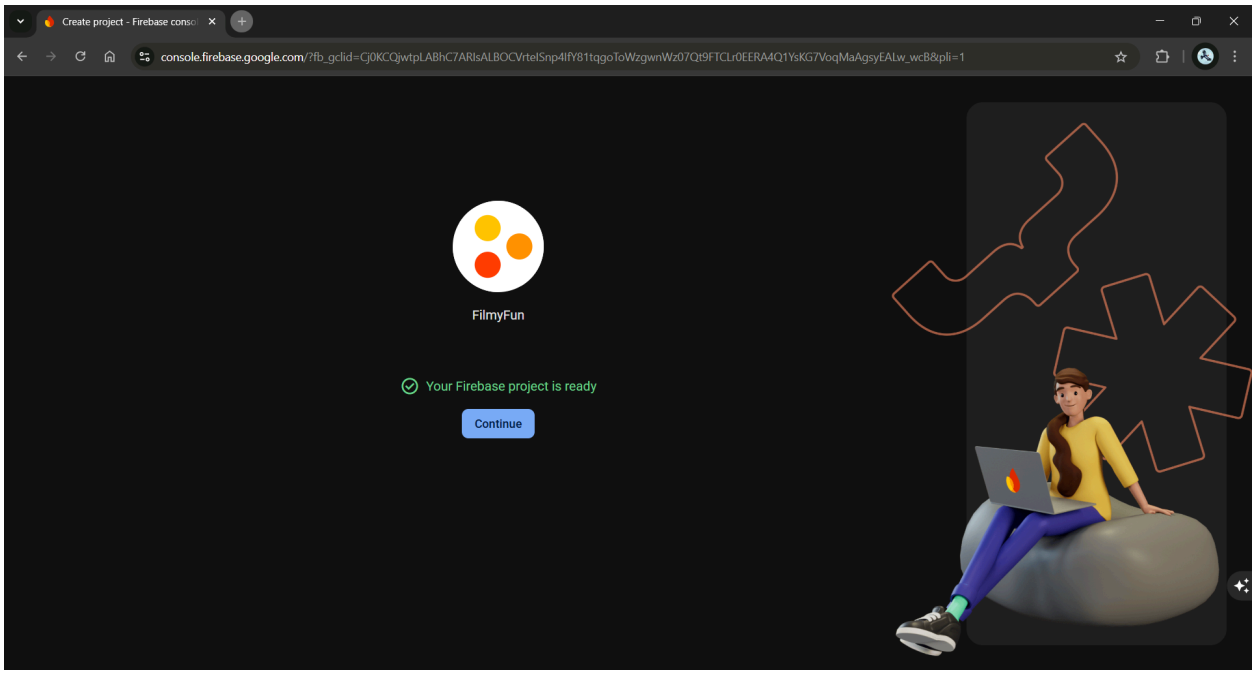
1.1) Go to Firebase Console and Create a Firebase Project



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

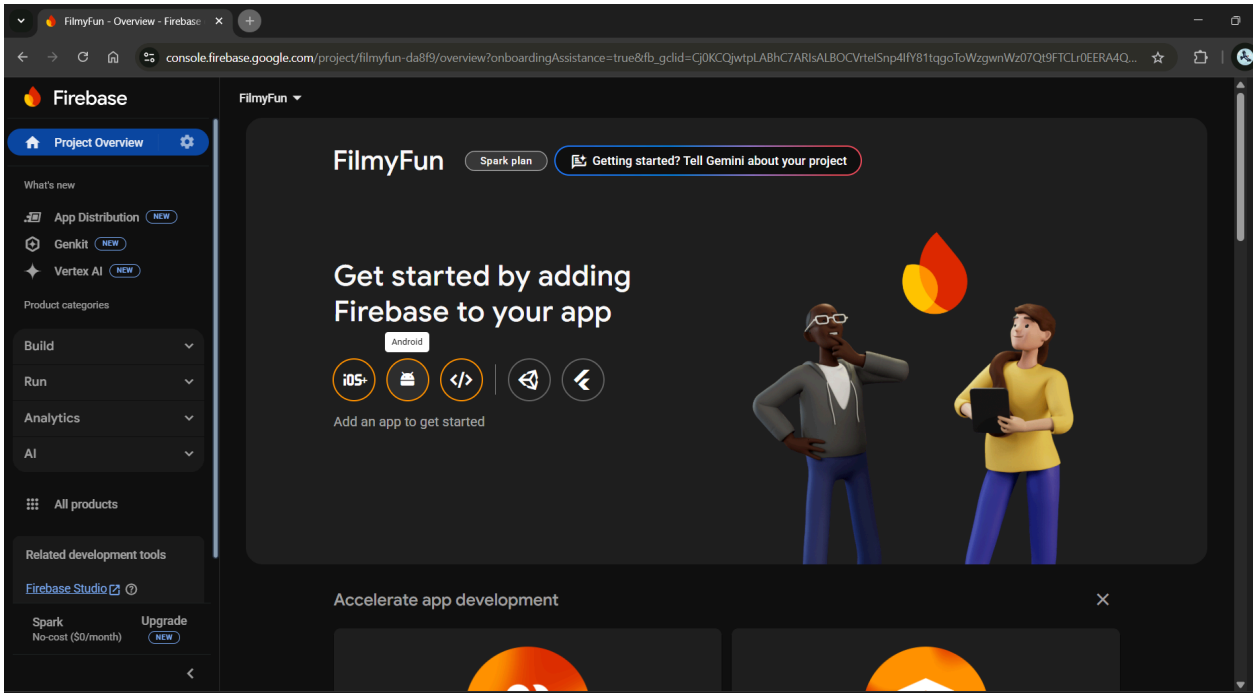


1.3) Enable Google Analytics (optional) & Click continue and complete the setup

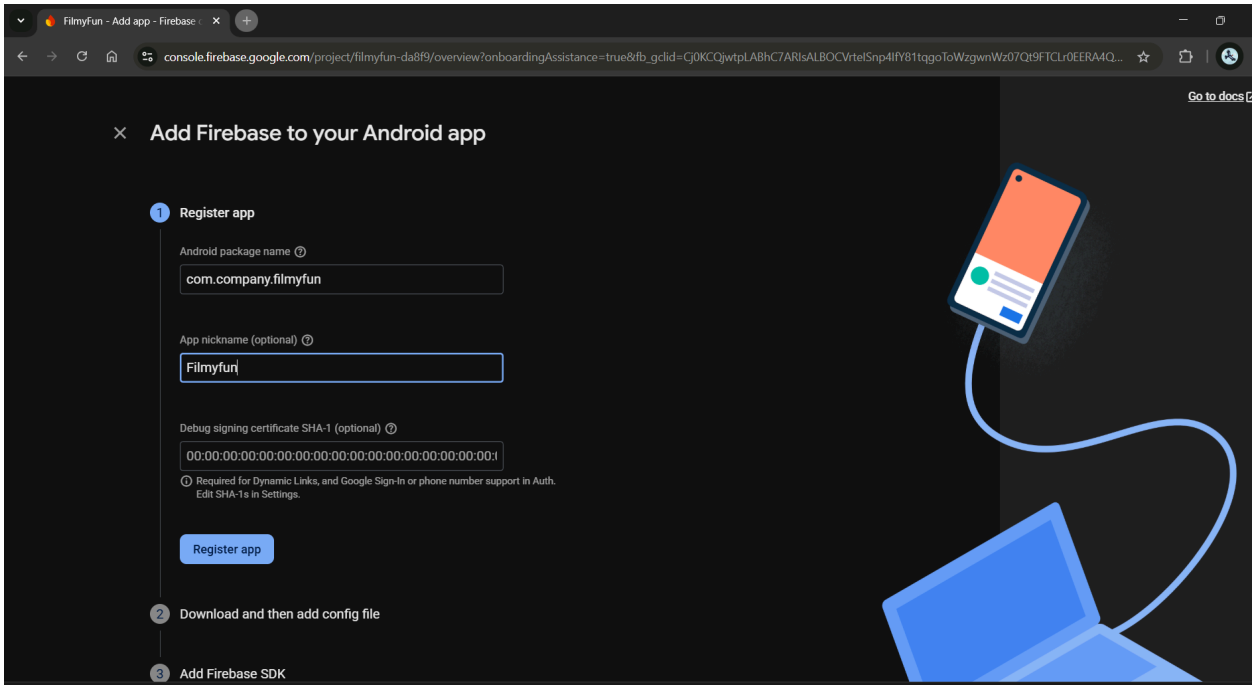


Step 2:- Add Firebase to Your Flutter App

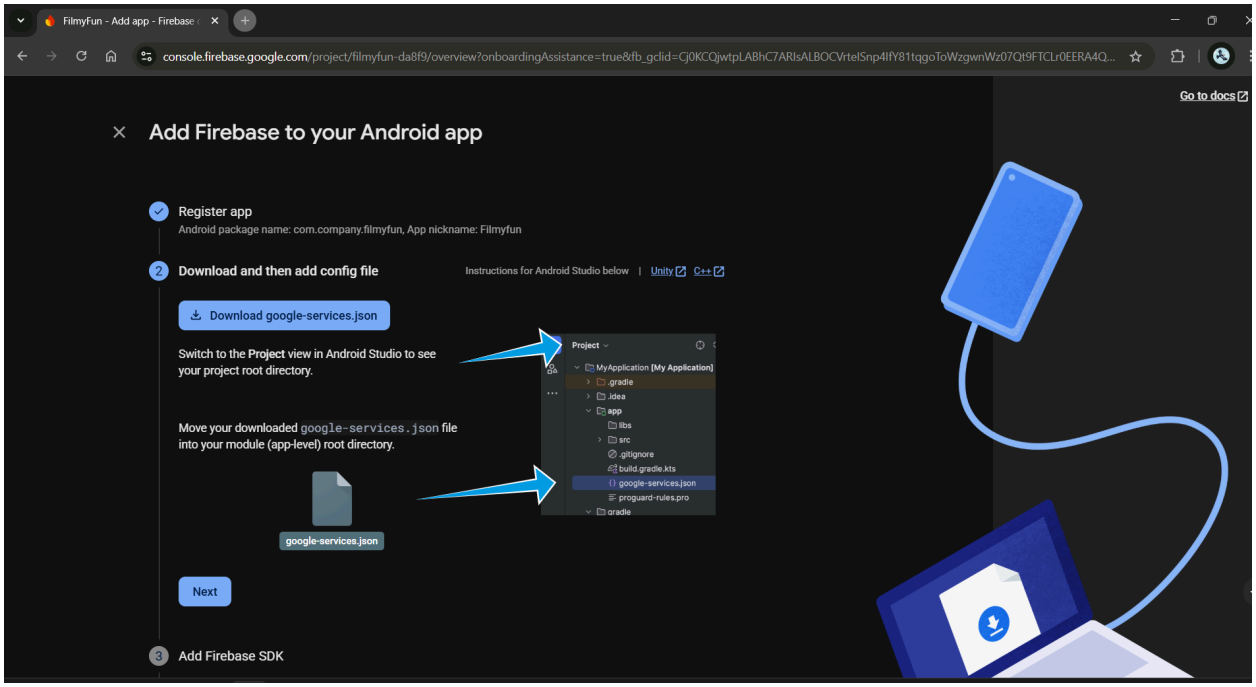
2.1) Click on Android/iOS/Web based on your Flutter application

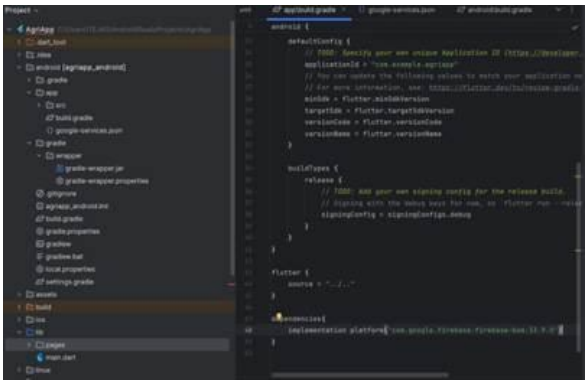


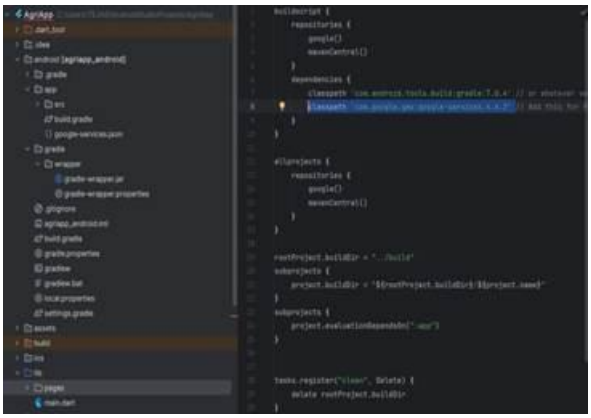
2.2) Register your app with a unique package name (found in android/app/build.gradle for Android).



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







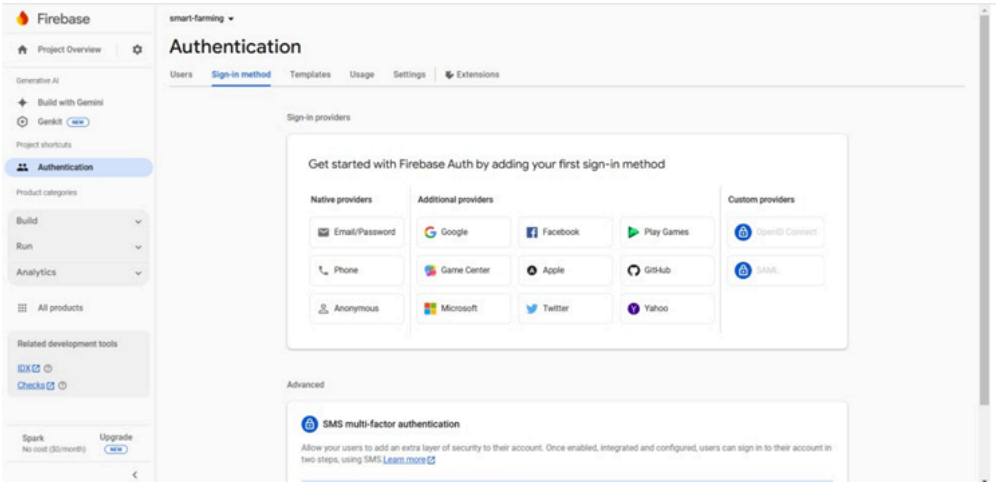
Step 3: - Add Firebase Authentication to Your App

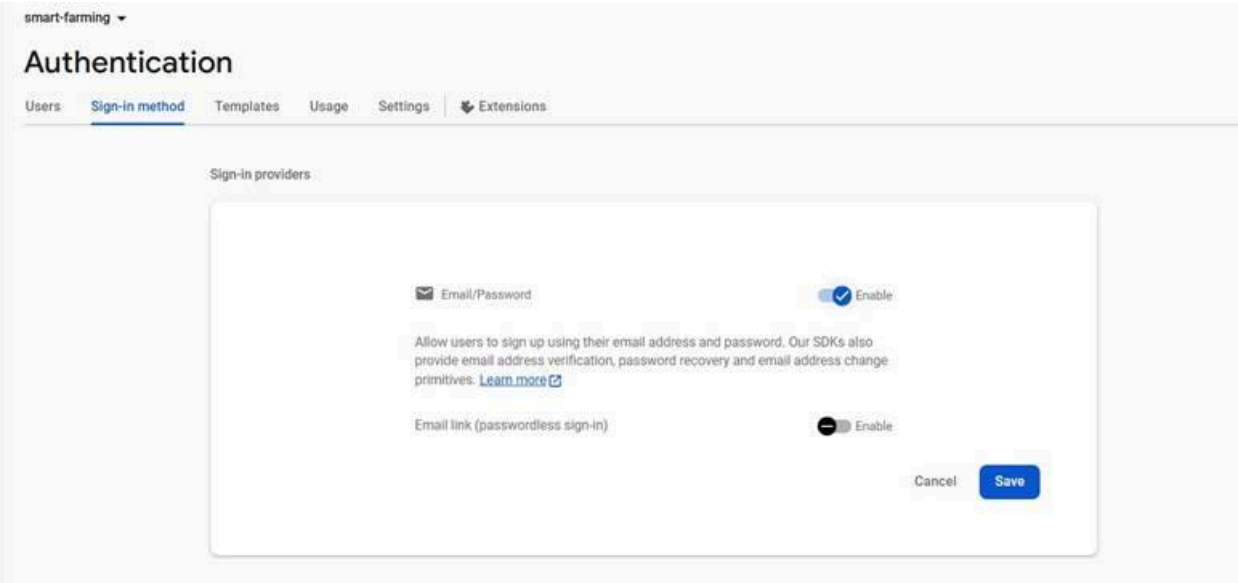
3.1) Add Firebase Authentication Dependencies

```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).





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.

After that data gets stored in database