MAD and PWA Lab

Name: Sarvadnya Awaghad Class: D15A Roll no:04

Aim: To design Flutter UI by including common widgets.

Theory:

We can split the Flutter widget into two categories:

1. Visible (Output and Input)
2. Invisible (Layout and Control)

1. Visible widget

The visible widgets are related to the user input and output data. Some of the important types of
this widget are:

1. Text

A Text widget holds some text to display on the screen. We can align the text widget by using textAlign property, and style property allow the customization of Text that includes font, font weight, font style, letter spacing, color, and many more.

2. Button

This widget allows you to perform some action on click. Flutter does not allow
you to use the Button widget directly; instead, it uses a type of buttons like a FlatButton and a RaisedButton.

3. Image

This widget holds the image which can fetch it from multiple sources like from the asset folder or directly from the URL. It provides many constructors for loading image, which are given below:

o Image: It is a generic image loader, which is used by ImageProvider.

o asset: It load image from your project asset folder.

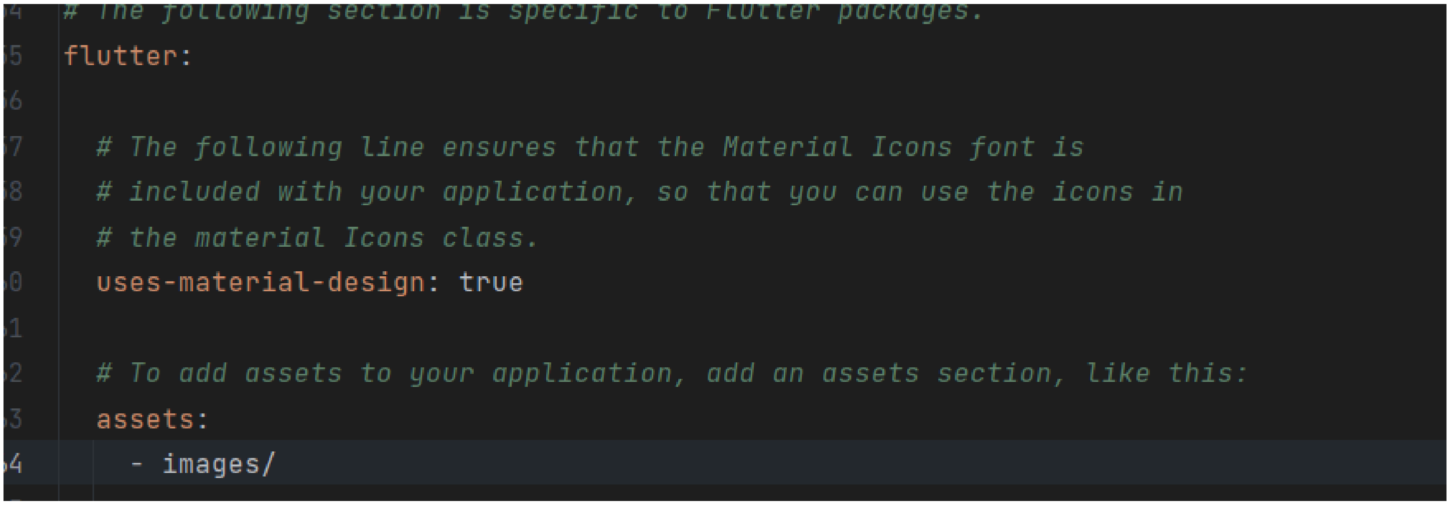
o file: It loads images from the system folder.

o memory: It load image from memory.

o network: It loads images from the network.

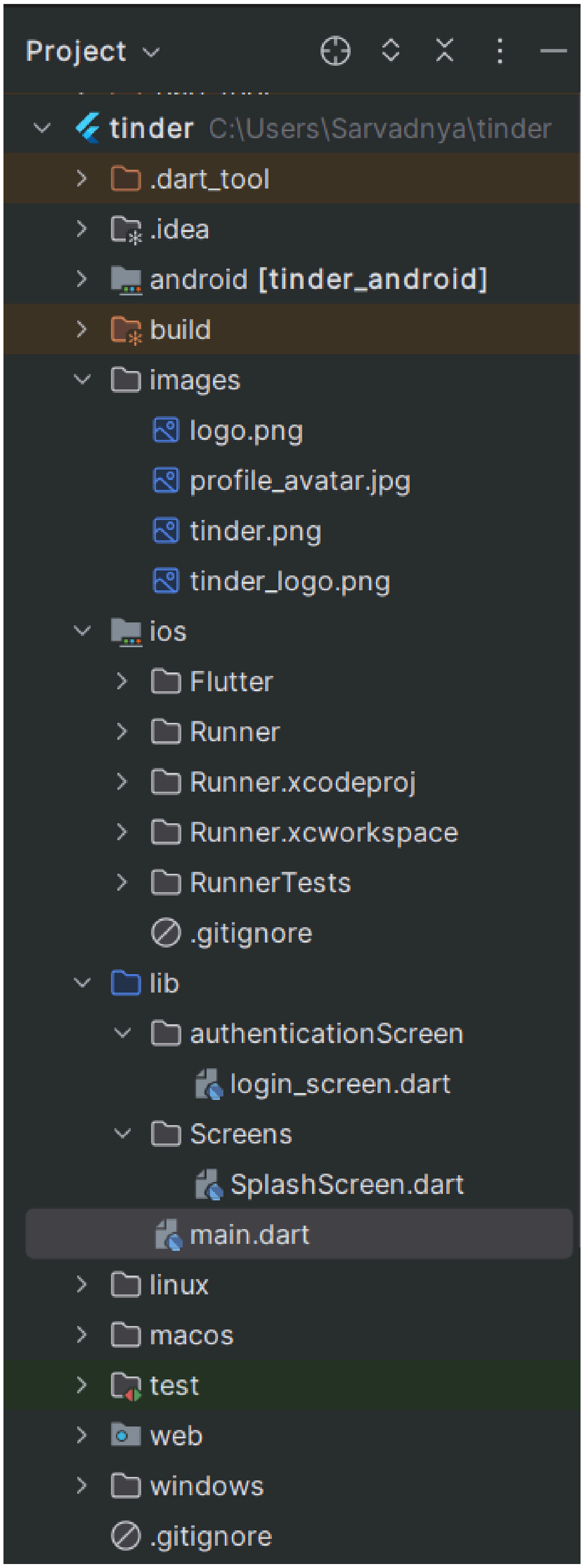
To add an image in the project, you need first to create an assets folder where you keep your images and then add the below line in pubspec.yaml file.

```
assets:
- assets/images
```



```
# 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/
```

File Structure:



```
Project
tinder C:\Users\Sarvadnya\tinder
    .dart_tool
    .idea
    android [tinder_android]
    build
    images
        logo.png
        profile_avatar.jpg
        tinder.png
        tinder_logo.png
    ios
        Flutter
        Runner
        Runner.xcodeproj
        Runner.xcworkspace
        RunnerTests
        .gitignore
    lib
        authenticationScreen
            login_screen.dart
        Screens
            SplashScreen.dart
        main.dart
    linux
    macos
    test
    web
    windows
    .gitignore
```

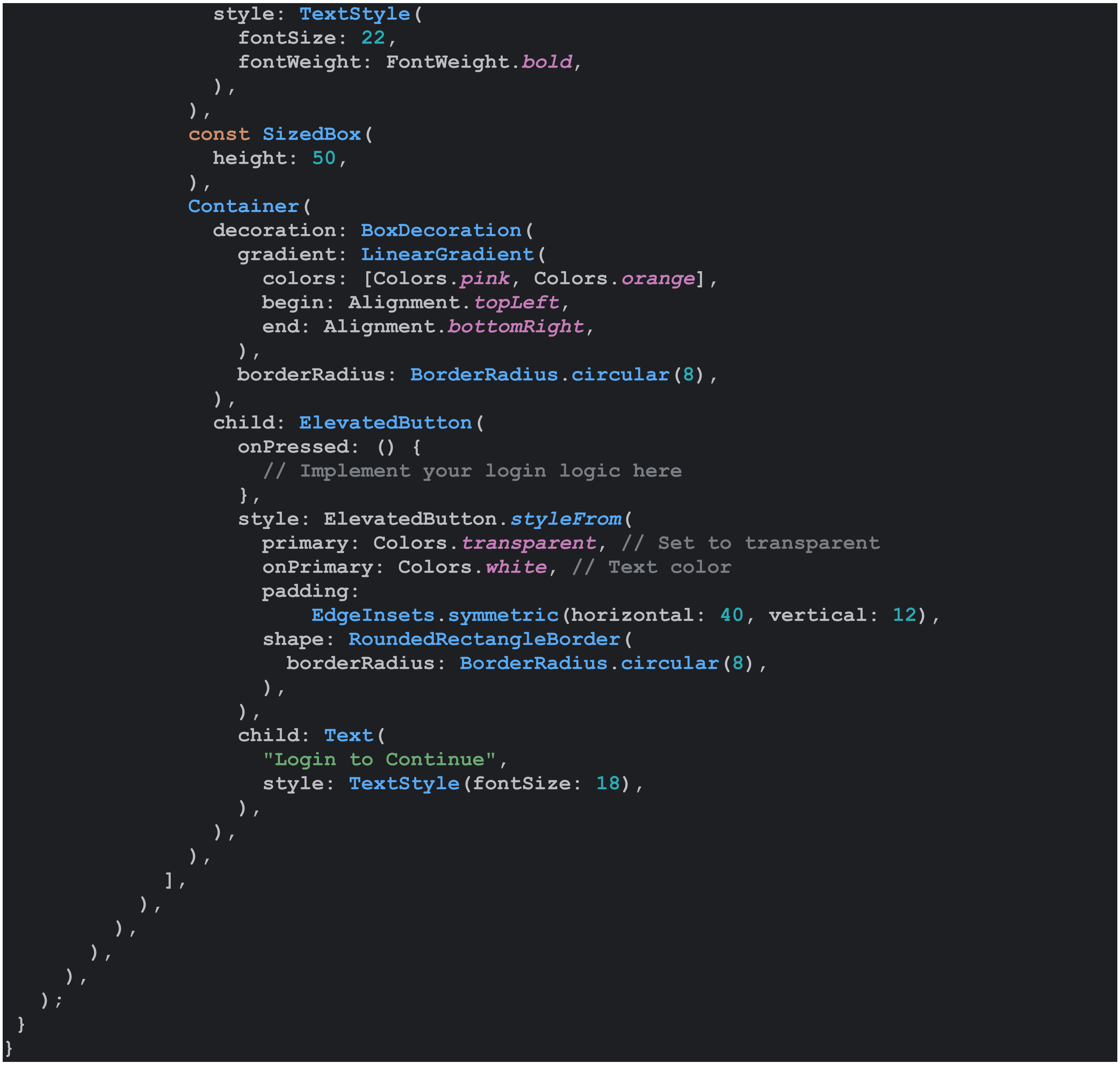
Code:

main.dart :

```
import 'package:flutter/material.dart';
import 'package:tinder/authenticationScreen/login_screen.dart';

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

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

 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     debugShowCheckedModeBanner: false,
     title: 'Tinder Clone',
     theme: ThemeData(
       primaryColor: Colors.white,
       // secondaryHeaderColor: Color.fromRGBO(255, 88, 100, 1.0),
     ),
     home: LoginScreen(),
   );
 }
}
```

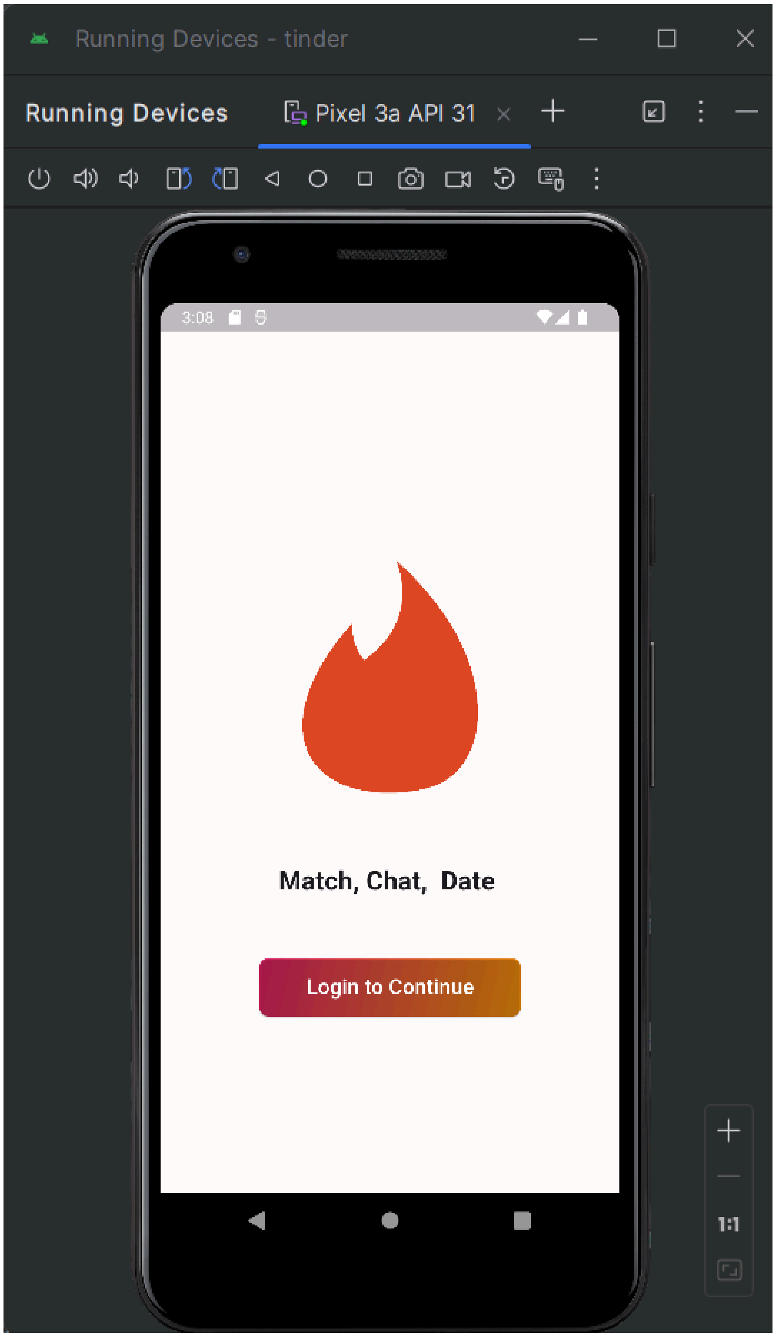
LoginScreen.dart :

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

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

 @override
 State<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
     body: Container(
       child: SingleChildScrollView(
         child: Center(
           child: Column(
             children: [
               const SizedBox(
                 height: 220,
               ),
               Image.asset(
                 "images/tinder_logo.png",
                 width: 150,
               ),
               const SizedBox(
                 height: 60,
               ),
               Text(
                 "Match, Chat,  Date ",
```



```
                 style: TextStyle(
                   fontSize: 22,
                   fontWeight: FontWeight.bold,
                 ),
               ),
               const SizedBox(
                 height: 50,
               ),
               Container(
                 decoration: BoxDecoration(
                   gradient: LinearGradient(
                     colors: [Colors.pink, Colors.orange],
                     begin: Alignment.topLeft,
                     end: Alignment.bottomRight,
                   ),
                   borderRadius: BorderRadius.circular(8),
                 ),
                 child: ElevatedButton(
                   onPressed: () {
                     // Implement your login logic here
                   },
                   style: ElevatedButton.styleFrom(
                     primary: Colors.transparent, // Set to transparent
                     onPrimary: Colors.white, // Text color
                     padding:
                         EdgeInsets.symmetric(horizontal: 40, vertical: 12),
                     shape: RoundedRectangleBorder(
                       borderRadius: BorderRadius.circular(8),
                     ),
                   ),
                   child: Text(
                     "Login to Continue",
                     style: TextStyle(fontSize: 18),
                   ),
                 ),
               ),
             ],
           ),
         ),
       ),
     ),
   );
 }
}
```

Output:



Conclusion:

Flutter’s widget architecture offers great flexibility for building complex UIs. Understanding key widgets and concepts is essential for effective Flutter development.