

Exp No: 2

Aim: To design Flutter UI by including common widgets.

Theory:

Text: Flutter's Text widget is used to display text on the screen. It supports various styles such as font size, font weight, color, alignment, and more. You can use it to display static text as well as dynamic text generated at runtime.

Image: The Image widget is used to display images in a Flutter application. It supports various image sources, including network images, local images, and even memory images. You can customize the image's dimensions, alignment, and more.

Icon: Flutter's Icon widget is used to display graphical icons from an icon library such as Material Icons or FontAwesomeIcons. Icons can be customized with properties like size, color, and alignment. They're commonly used for indicating actions or representing UI elements.

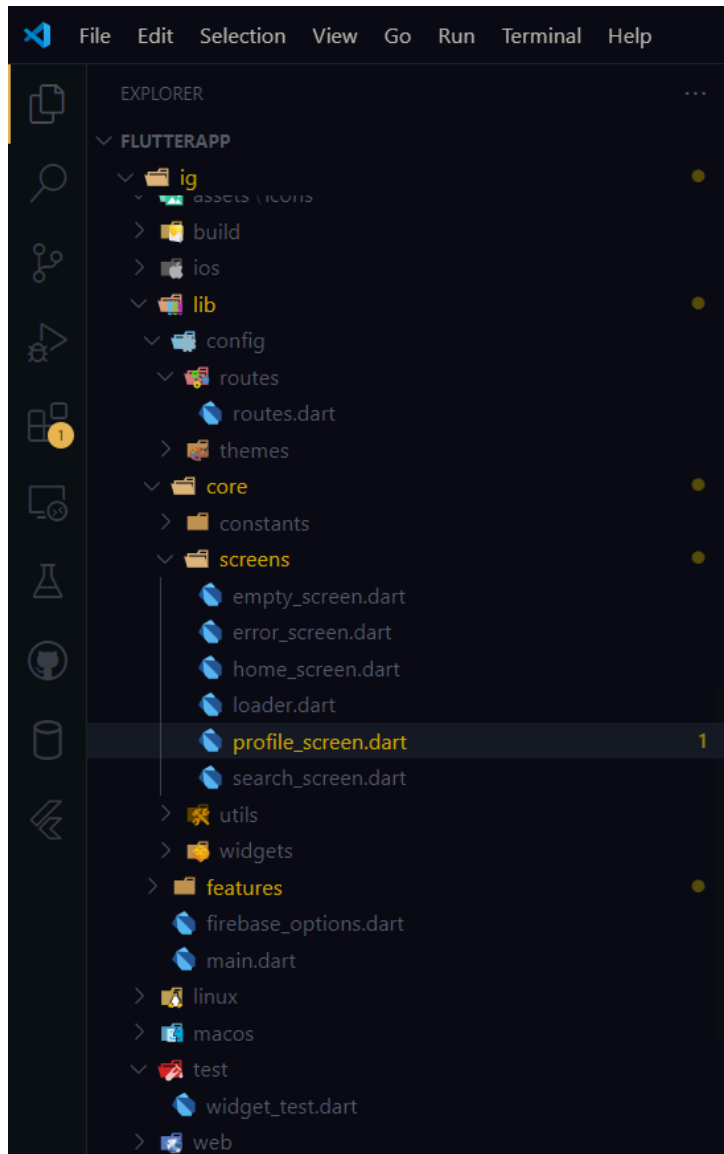
Container: The Container widget is a versatile widget used to create rectangular visual elements. It can contain a single child widget and supports styling options like color, padding, margin, border, and more. Containers are often used for layout purposes and to apply styling to other widgets.

Row: The Row widget arranges its children widgets horizontally in a single line. It's commonly used for laying out multiple widgets side by side. You can control the alignment, spacing, and size of the children widgets within the Row.

Column: Similar to Row, the Column widget arranges its children widgets vertically in a single line. It's useful for creating vertical layouts such as lists or forms. Like Row, you have control over the alignment, spacing, and size of the children widgets within the Column.

ListView: The ListView widget is used to create scrollable lists of widgets. It's particularly useful when dealing with a large number of items that need to be displayed within limited screen space. ListView can display its children vertically or horizontally and supports both static and dynamic lists.

File structure:



Code:

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

```

import 'package:ig/features/auth/presentation/screens/login_screen.dart';
import
'package:ig/features/auth/providers/get_user_info_by_id_provider.dart';
import
'package:ig/features/posts/presentation/widgets/add_friend_button.dart';
import '/core/constants/app_colors.dart';
import '/core/constants/constants.dart';
import '/core/constants/extensions.dart';
import '/core/screens/error_screen.dart';
import '/core/screens/loader.dart';
import '/core/widgets/round_button.dart';
import '/features/posts/presentation/widgets/icon_text_button.dart';

```

```

class ProfileScreen extends ConsumerWidget {
  const ProfileScreen({
    Key? key,
    this.userId,
  }) : super(key: key);

  final String? userId;
  static const String routeName = '/profile'; // Define the route name
  @override
  Widget build(BuildContext context, WidgetRef ref) {
    final myUid = FirebaseAuth.instance.currentUser!.uid;
    final uid = userId ?? myUid;
    final userInfo = ref.watch(getUserInfoAsStreamByIdProvider(uid));

    return userInfo.when(
      data: (user) {
        return SafeArea(
          child: Scaffold(
            appBar: userId != myUid ? AppBar(
              title: const Text('Profile'),
            ) : null,
            backgroundColor: AppColors.whiteColor,
            body: Padding(
              padding: Constants.defaultPadding,

```

```
child: Column(  
  crossAxisAlignment: CrossAxisAlignment.start,  
  children: [  
    Row(  
      children: [  
        CircleAvatar(  
          radius: 50,  
          backgroundImage: NetworkImage(user.profilePicUrl),  
        ),  
        const SizedBox(width: 20),  
        Column(  
          crossAxisAlignment: CrossAxisAlignment.start,  
          children: [  
            Text(  
              user.fullName,  
              style: const TextStyle(  
                fontWeight: FontWeight.w400,  
                fontSize: 21,  
              ),  
            ),  
            const SizedBox(height: 5),  
            Row(  
              children: [  
                Text(  
                  '1000 ',  
                  style: TextStyle(fontWeight: FontWeight.bold),  
                ),  
                Text('followers'),  
                const SizedBox(width: 10),  
                Text(  
                  '1000 ',  
                  style: TextStyle(fontWeight: FontWeight.bold),  
                ),  
                Text('following'),  
              ],  
            ),  
          ],  
        ),  
      ],  
    ),  
  ],  
),
```

```

        ),
      ],
    ),
    const SizedBox(height: 20),
    // userId == myUid
    //   ? _buildAddToStoryButton()
    //   : AddFriendButton(
    //     user: user,
    //   ),
    const SizedBox(height: 10),
    RoundButton(
      onPressed: () async {
        await _signOut(context);
      },
      label: 'Log out',
      color: Colors.transparent,
    ),
    const SizedBox(height: 20),
    _buildProfileInfo(
      email: user.email,
      gender: user.gender,
      birthday: user.birthday,
    ),
  ],
),
),
),
);
},
error: (error, stackTrace) {
  return ErrorScreen(error: error.toString());
},
loading: () {
  return const Loader();
},
);
}

```

```

Future<void> _signOut(BuildContext context) async {
  try {
    await FirebaseAuth.instance.signOut();
    Navigator.pushNamedAndRemoveUntil(
      context,
      LoginScreen.route, // Use the route getter
      (route) => false,
    );
  } catch (e) {
    print('Error signing out: $e');
    // Handle error here
  }
}

```

```

_buildAddToStoryButton() => RoundButton(onPressed: () {}, label: 'Add to
Story');

```

```

_buildProfileInfo({
  required String email,
  required String gender,
  required DateTime birthday,
}) =>
  Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      IconButton(
        icon: gender == 'male' ? Icons.male : Icons.female,
        label: gender,
      ),
      const SizedBox(height: 10),
      IconButton(
        icon: Icons.cake,
        label: birthday.yMMMEd(),
      ),
      const SizedBox(height: 10),
      IconButton(

```

```
        icon: Icons.email,  
        label: email,  
    ),  
],  
);  
}
```

Output:

