

Experiment 2

Name: Piyush Tilokani

Div: D15A

Roll no: 63

Aim: To design Flutter UI by including common widgets.

Theory: In Flutter, widgets are the building blocks of the user interface, and several common widgets play crucial roles in creating engaging and interactive applications. Here's a brief overview of some fundamental Flutter widgets:

1. **Container:** The most basic building block, a container is a box model that can contain other widgets, allowing you to customize its dimensions, padding, and decoration.
2. **Row and Column:** These widgets help organize children widgets horizontally (Row) or vertically (Column), facilitating the creation of flexible and responsive layouts.
3. **AppBar:** AppBar is a material design widget providing a top app bar that typically includes the app's title, leading and trailing icons, and actions.
4. **ListView:** Used to create scrollable lists of widgets, ListView is versatile for displaying a large number of items efficiently.
5. **TextField:** Enables users to input text, providing a text editing interface with options for validation, styling, and interaction.
6. **ElevatedButton:** ElevatedButton is a Flutter widget used to create a button with a raised appearance. It typically represents the primary action in a user interface. The button has a background color, elevation, and responds to user interactions with visual feedback.
7. **Image:** The Image widget displays images from various sources, supporting both local and network images.
8. **Scaffold:** A top-level container for an app's visual elements, Scaffold provides a structure that includes an AppBar, body, and other optional features like drawers and bottom navigation.
9. **Card:** Representing a material design card, this widget displays information in a compact and visually appealing format, often used for grouping related content.
10. **GestureDetector:** Allows detection of various gestures like taps, drags, and long presses, enabling interactive responses to user input.
11. **Stack:** A widget that allows children widgets to be overlaid, facilitating complex UI designs by layering widgets on top of each other.

12. FutureBuilder: Ideal for handling asynchronous operations, FutureBuilder simplifies the management of UI updates based on the completion of a Future, making it valuable for fetching and displaying data.

These are just a few of the many widgets available in Flutter, each serving a unique purpose in crafting dynamic and user-friendly interfaces.

Code:

```
import 'package:cached_network_image/cached_network_image.dart';
import 'package:faker/faker.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:google_fonts/google_fonts.dart';
```

```
class DetailPage extends StatelessWidget {
  const DetailPage({super.key, required this.imageUrl});
  final String imageUrl;
```

```
  @override
```

```
  Widget build(BuildContext context) {
    return Scaffold(
      body: Column(
        children: [
          Expanded(
            child: Container(
              decoration: BoxDecoration(
                image: DecorationImage(
                  image: CachedNetworkImageProvider(imageUrl),
                  fit: BoxFit.cover,
                ),
              ),
            ),
          child: SafeArea(
            child: Padding(
              padding: const EdgeInsets.symmetric(horizontal: 18),
              child: Column(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                crossAxisAlignment: CrossAxisAlignment.end,
                children: [
                  Row(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,
                    children: [
                      InkWell(
                        onTap: () => Navigator.pop(context),
                        child: CircleAvatar(
```

```

        backgroundColor: Colors.black.withOpacity(0.2),
        child: const Icon(
          CupertinoIcons.chevron_back,
          color: Colors.white,
          size: 28,
        ),
      ),
    ),
  ),
InkWell(
  onTap: () {
    showModalBottomSheet(
      context: context,
      builder: (context) {
        return SizedBox(
          height: 330,
          child: Padding(
            padding: const EdgeInsets.symmetric(
              horizontal: 12, vertical: 16),
            child: Column(
              crossAxisAlignment:
                CrossAxisAlignment.start,
              children: [
                Row(
                  crossAxisAlignment:
                    CrossAxisAlignment.center,
                  children: [
                    const Icon(
                      FontAwesomeIcons.xmark),
                    const SizedBox(
                      width: 12,
                    ),
                    Text(
                      'Options',
                      style: GoogleFonts.notoSans(),
                    ),
                  ],
                ),
                const SizedBox(
                  height: 30,
                ),
                Text(
                  'Follow ${Faker().internet.userName()}',
                  style: GoogleFonts.notoSans(
                    fontSize: 18,
                    fontWeight: FontWeight.w500,
                  ),
                ),
              ],
            ),
          ),
        );
      },
    );
  },
);

```

```
),
const SizedBox(
  height: 15,
),
Text(
  'Copy link',
  style: GoogleFonts. notoSans(
    fontSize: 18,
    fontWeight: FontWeight.w500,
  ),
),
const SizedBox(
  height: 15,
),
Text(
  'Download image',
  style: GoogleFonts. notoSans(
    fontSize: 18,
    fontWeight: FontWeight.w500,
  ),
),
const SizedBox(
  height: 15,
),
Text(
  'Hide Pin',
  style: GoogleFonts. notoSans(
    fontSize: 18,
    fontWeight: FontWeight.w500,
  ),
),
const SizedBox(
  height: 15,
),
Text(
  'Report Pin',
  style: GoogleFonts. notoSans(
    fontSize: 18,
    fontWeight: FontWeight.w500,
  ),
),
Text(
  "This goes against Pinterest's community guidelines",
  style: GoogleFonts. notoSans(
    fontSize: 14,
  ),
),
```

```

        ),
      ],
    ),
  ),
);
},
shape: RoundedRectangleBorder(
  borderRadius: BorderRadius.circular(20.0),
),
);
},
child: const Icon(
  CupertinoIcons.ellipsis,
  color: Colors.white,
  size: 28,
),
),
],
),
CircleAvatar(
  backgroundColor: Colors.white.withOpacity(0.8),
  child: const Icon(
    CupertinoIcons.viewfinder,
    color: Colors.black,
  ),
)
],
),
),
),
),
),
),
Container(
  padding: const EdgeInsets.only(
    top: 10,
    bottom: 30,
    left: 18,
    right: 18,
  ),
  color: Colors.white,
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      const Icon(CupertinoIcons.heart_circle_fill),
      Row(
        children: [

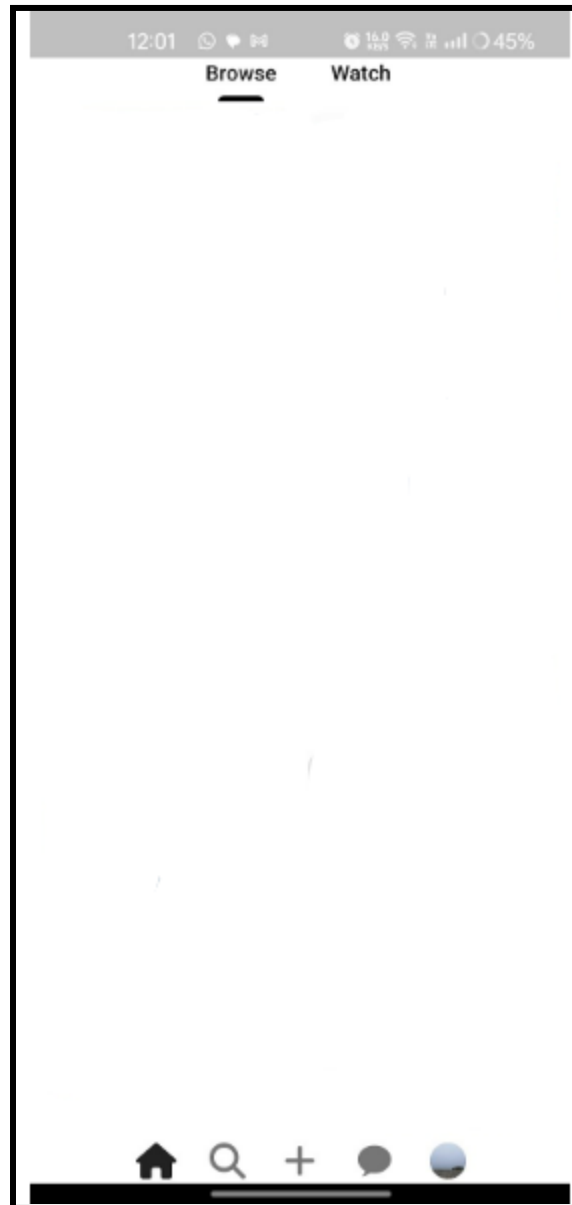
```

```

Container(
  padding: const EdgeInsets.symmetric(
    horizontal: 17, vertical: 15),
  decoration: BoxDecoration(
    color: const Color(0xFFFF1F1F1),
    borderRadius: BorderRadius.circular(30),
  ),
  child: Text(
    'View',
    style: GoogleFonts.notoSans(
      fontWeight: FontWeight.w500,
    ),
  ),
),
const SizedBox(
  width: 6,
),
Container(
  padding: const EdgeInsets.symmetric(
    horizontal: 17, vertical: 15),
  decoration: BoxDecoration(
    color: Theme.of(context).colorScheme.secondary,
    borderRadius: BorderRadius.circular(30),
  ),
  child: Text(
    'Save',
    style: GoogleFonts.notoSans(
      fontWeight: FontWeight.w500,
      color: Colors.white,
    ),
  ),
),
),
),
),
),
),
const Icon(Icons.share),
),
),
),
),
);
}
}

```

App UI:



Widgets used: Icons, font, bottom navigation bar, image

Conclusion: Thus, understood the use of basic common widgets used in Mobile App Development and used some of them to create the login page for the chosen mini project application.