EXP 2

Aim: To design Flutter UI by including common widgets.

Theory:
Widgets:

Each element on a screen of the Flutter app is a widget. The view of the screen completely depends upon the choice and sequence of the widgets used to build the apps and the structure of the code of an app is a tree of widgets.

Category of Widgets:

There are mainly 14 categories in which the flutter widgets are divided. They are mainly segregated on the basis of the functionality they provide in a flutter application.

1. Accessibility: These are the set of widgets that make a flutter app more easily accessible.
2. Animation and Motion: These widgets add animation to other widgets.
3. Assets, Images, and Icons: These widgets take charge of assets such as display images and show icons.
4. Async: These provide async functionality in the flutter application.
5. Basics: These are the bundle of widgets that are absolutely necessary for the development of any flutter application.
6. Cupertino: These are the iOS designed widgets.
7. Input: This set of widgets provides input functionality in a flutter application.
8. Interaction Models: These widgets are here to manage touch events and route users to different views in the application.
9. Layout: This bundle of widgets helps in placing the other widgets on the screen as needed.
10. Material Components: This is a set of widgets that mainly follow material design by Google.
11. Painting and effects: This is the set of widgets that apply visual changes to their child widgets without changing their layout or shape.
12. Scrolling: This provides scrollability of to a set of other widgets that are not scrollable by default.
13. Styling: This deals with the theme, responsiveness, and sizing of the app.
14. Text: This displays text.

Description of few of the widgets are as follows:

- Scaffold– Implements the basic material design visual layout structure.
- App-Bar- To create a bar at the top of the screen.
- Text- To write anything on the screen.
- Container– To contain any widget.
- Center– To provide center alignment to other widgets.

Code:

Code for option buttons:

```
Widget _buildOptionButton(

  BuildContext context,

  String text,

  IconData icon,

  Color color, {

  Color textColor = Colors.white,

  Color? iconColor,

  required Widget destination,

}) {

  return Container(

    width: double.infinity,

    height: 64,

    decoration: BoxDecoration(

      color: color,

      borderRadius: BorderRadius.circular(28),

      boxShadow: [

        BoxShadow(
```

```dart
          color: color.withOpacity(0.5),

          blurRadius: 10,

          offset: const Offset(0, 5),

        ),

      ],

      border: Border.all(

        color: SelectionScreen.textDark.withOpacity(0.5),

        width: 1.5,

      ),

    ),

    child: ElevatedButton(

      onPressed: () {

        Navigator.push(

          context,

          MaterialPageRoute(builder: (context) => destination),

        );

      },

      style: ElevatedButton.styleFrom(

        backgroundColor: color,

        foregroundColor: textColor,

        elevation: 0,

        shape: RoundedRectangleBorder(

          borderRadius: BorderRadius.circular(28),

        ),

      ),
```

```dart
    child: Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Icon(
          icon,
          size: 26,
          color: iconColor ?? textColor,
        ),
        const SizedBox(width: 12),
        Text(
          text,
          style: TextStyle(
            fontSize: 18,
            fontWeight: FontWeight.w600,
            letterSpacing: 0.5,
            color: textColor,
          ),
        ),
      ],
    ),
  ),
);
}
```

Code for card widget:

```
Container(

 padding: const EdgeInsets.symmetric(horizontal: 16, vertical: 14),

 decoration: BoxDecoration(

  color: deepMint.withOpacity(0.8),

  borderRadius: BorderRadius.circular(16),

  boxShadow: [

   BoxShadow(

    color: Colors.black.withOpacity(0.08),

    blurRadius: 6,

    offset: const Offset(0, 3),

   ),

  ],

  border: Border.all(

   color: SelectionScreen.textDark.withOpacity(0.5),

   width: 1.5,

  ),

 ),

 child: Column(

  mainAxisAlignment: MainAxisAlignment.center,

  crossAxisAlignment: CrossAxisAlignment.center,

  children: [

   Text(

    "Did you know?",
```

```dart
          style: TextStyle(

            fontSize: 18,

            fontWeight: FontWeight.w600,

            color: SelectionScreen.textDark.withOpacity(0.8),

          ),

        ),

        const SizedBox(height: 8),

        Text(

          "Journaling daily can improve your wellbeing by 23%",

          style: TextStyle(

            fontSize: 14,

            color: SelectionScreen.textDark.withOpacity(0.85),

            fontWeight: FontWeight.w500,

          ),

        ),

      ],

    ),

)
```

Main.dart code:

```dart
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:timezone/data/latest_all.dart' as tz;
import 'LoginScreen.dart'; // Import your HomeScreen

void main() async {
  WidgetsFlutterBinding.ensureInitialized();

  await Firebase.initializeApp(
```

```
    options: FirebaseOptions(
      apiKey: "AIzaSyA9kxrbh6kBWblj2qeYr-gEgIQ-SBruAhQ",
      appId: "1:264673015498:web:03c5c23b9d471390fb6633",
      messagingSenderId: "264673015498",
      projectId: "moodlog-project",
    ),
  );

  tz.initializeTimeZones(); // ✅ Ensure timezone is initialized correctly
  print("🔥 Firebase Initialized Successfully!");
  print("🌍 Timezones Initialized!");

  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Mood Log',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: LoginScreen(), // Navigate to signup
    );
  }
}
```
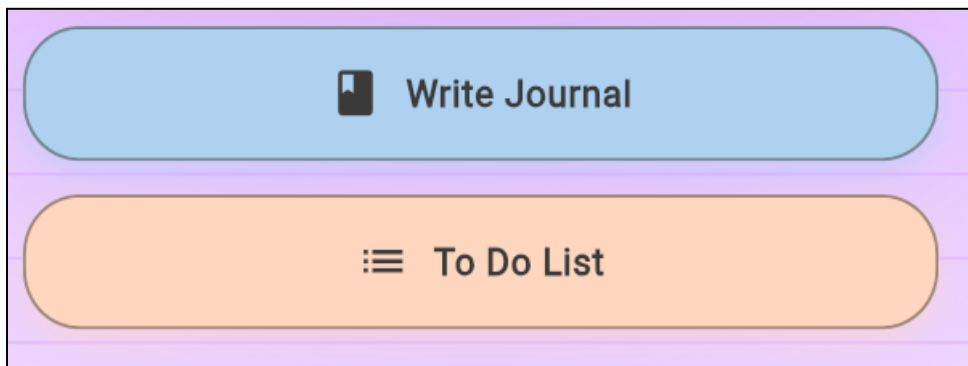
Output:

Elevated buttons:

Text fields:

Checkbox:

Container/card widgets:

Did you know?

Journaling daily can improve your wellbeing by 23%
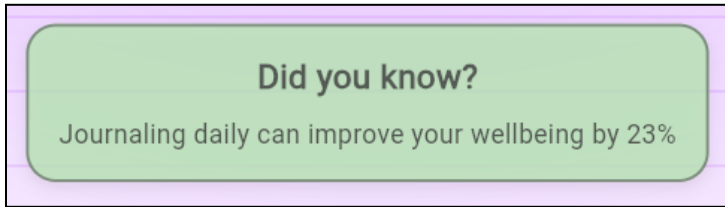
Conclusion:

In conclusion,we can say that Flutter provides a rich set of widgets that are essential for creating intuitive and responsive UIs. Common widgets like TextField, IconButton, ElevatedButton, Card, and Checkbox play a key role in building the user interface of apps. These widgets help users input data, perform actions, display content in an organized manner, and enhance overall user interaction. Understanding and effectively using these widgets allows us to design clean, efficient, and user-friendly mobile applications.