Name: Abujaid Ansari **Batch** – A

Class: D15B **Roll No.** 02

Aim: To design flutter UI by including common widgets.

Theory:

Introduction to flutter:

Flutter is Google's SDK for crafting beautiful, fast user experiences for mobile, web, and desktop from a single codebase. Flutter works with existing code, is used by developers and organizations around the world, and is free and open source.

Flutter apps are written in the Dart language and make use of many of the language's more advanced features.

For better performance, release versions of Flutter apps on all platforms use ahead-of-time (AOT) compilation,[19] except for on the Web, where code is transpiled to JavaScript.

Flutter inherits Dart's Pub package manager and software repository, which allows users to publish and use custom packages as well as Flutter-specific plugins.

Flutter's engine, written primarily in C++, provides low-level rendering support using either Google's Skia graphics library or the custom "Impeller" graphics layer. Additionally, it interfaces with platform-specific SDKs such as those provided by Android and iOS to implement accessibility, file and network I/O, native plugin support, and more.

Code:

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

void main() {
  runApp(BMICalculatorApp());
}


class BMICalculatorApp extends StatelessWidget {
```

```
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'BMI Calculator',
      theme: ThemeData(
        primaryColor: Colors.blue,
        scaffoldBackgroundColor: Colors.white,
        textTheme: TextTheme(
          bodyText2: TextStyle(color: Colors.grey[800]),
        ),
      ),
      home: BMICalculatorScreen(),
    );
  }
}

class BMICalculatorScreen extends StatefulWidget {
  @override
  _BMICalculatorScreenState createState() => _BMICalculatorScreenState();
}

class _BMICalculatorScreenState extends State<BMICalculatorScreen> {
  // Variables to store user input data
  double _height = 170.0; // Initial height in centimeters
  double _weight = 70.0; // Initial weight in kilograms

  @override
  Widget build(BuildContext context) {
```

```
return Scaffold(
  appBar: AppBar(
    title: Text('BMI Calculator'),
  ),
  body: Padding(
    padding: EdgeInsets.all(16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.center,
      children: [
        SizedBox(height: 20.0),
        Text(
          'Calculate Your BMI',
          style: TextStyle(
            fontSize: 24.0,
            fontWeight: FontWeight.bold,
          ),
        ),
        SizedBox(height: 20.0),
        Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text('Height (cm): '),
            SizedBox(width: 10.0),
            Container(
              width: 100.0,
              child: TextFormField(
                keyboardType: TextInputType.number,
                initialValue: _height.toString(),
                onChanged: (value) {
```

```
            setState(() {
              _height = double.tryParse(value) ?? 0.0;
            });
          },
        ),
      ),
    ],
  ),
  SizedBox(height: 20.0),
  Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Text('Weight (kg): '),
      SizedBox(width: 10.0),
      Container(
        width: 100.0,
        child: TextFormField(
          keyboardType: TextInputType.number,
          initialValue: _weight.toString(),
          onChanged: (value) {
            setState(() {
              _weight = double.tryParse(value) ?? 0.0;
            });
          },
        ),
      ),
    ],
  ),
  SizedBox(height: 40.0),
```

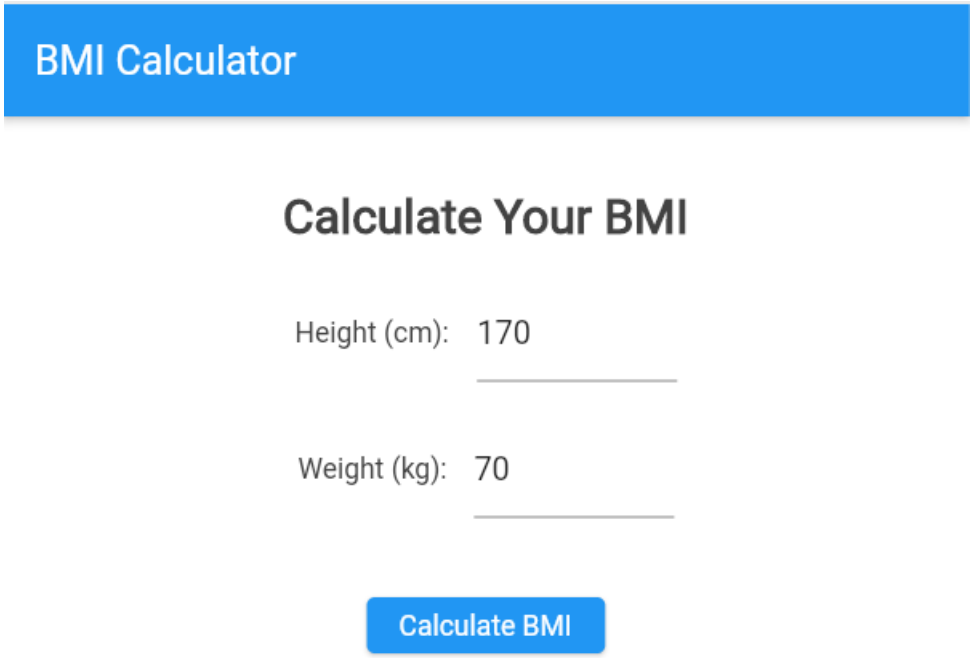
```
            ElevatedButton(
              onPressed: () {
                // Calculate BMI logic here
                double bmi = _weight / ((_height / 100) * (_height / 100));
                // Navigate to BMI result screen
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder: (context) => BMIResultScreen(bmi: bmi),
                  ),
                );
              },
              child: Text('Calculate BMI'),
            ),
          ],
        ),
      ),
    );
  }
}
class BMIResultScreen extends StatelessWidget {
  final double bmi;

  BMIResultScreen({required this.bmi});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
```

```
        title: Text('BMI Result'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              'Your BMI: ${bmi.toStringAsFixed(2)}',
              style: TextStyle(fontSize: 24.0, fontWeight: FontWeight.bold),
            ),
            SizedBox(height: 20.0),
            // Add more UI elements based on BMI result for additional information or recommendations
          ],
        ),
      ),
    );
  }
}
```

Output:



Explanation:

1. Import Statements: The code begins with import statements, importing necessary packages from the Flutter framework.

2. Main Function: The `main()` function is the entry point of the application. It calls the `runApp()` function to start the application by passing an instance of `BMICalculatorApp`.

3. BMICalculatorApp Class: This class extends `StatelessWidget` and represents the entire BMI calculator application. It defines the app's theme and sets the home screen to `BMICalculatorScreen`.

4. BMICalculatorScreen Class: This class extends `StatefulWidget` and represents the screen where users can input their height and weight to calculate their BMI. It contains:

- State Management: `_BMICalculatorScreenState` manages the state of the screen. It stores the user's input for height and weight.

- Scaffold Widget: The `Scaffold` widget provides the basic structure of the screen, including an `AppBar` and a body.

- Padding and Column Widgets: These widgets are used to provide spacing and organize the layout vertically.

- Text Widgets: Display textual content such as the title and labels for height and weight inputs.

- TextFormField Widgets: Allow users to input their height and weight. The `onChanged` callback updates the corresponding state variables when the user enters new values.

- ElevatedButton Widget: Represents a button that triggers the BMI calculation when pressed. It navigates to the `BMIResultScreen` passing the calculated BMI as a parameter.

5. BMIResultScreen Class: This class extends `StatelessWidget` and represents the screen where the BMI result is displayed. It takes the calculated BMI as input and displays it along with any additional information or recommendations.

- Scaffold Widget: Provides the basic structure of the screen, including an `AppBar` and a body.

- Center and Column Widgets: These widgets are used to center the content vertically and organize the layout vertically.

- Text Widget: Displays the calculated BMI with a specified format.

- SizedBox Widget: Adds spacing between elements.

Overall, this code provides a basic BMI calculator app UI with functionality for users to input their height and weight. It utilizes common Flutter widgets to build the user interface.

Conclusion:

Hence, we created a basic GUI for our flutter application which will calculate the BMI. The GUI has labels for height and weight, input fields and a button. We have used different widgets namely text widget,SizedBox, Center and Column widget, etc. to create this GUI.