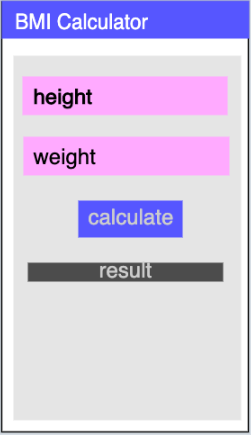
PRACTICAL 10

**AIM:** Flutter UI widgets Simple BMI Calculator App.

**Scenario:** Create a Flutter App which implements UI widgets to calculate BMI given height and weight as input.

****

**Part B (to be completed by students)**

**(Students must submit the soft copy as per the following segments. The soft copy must be uploaded on the Blackboard. The filename should be Batch\_RollNo\_Exp\_No)**

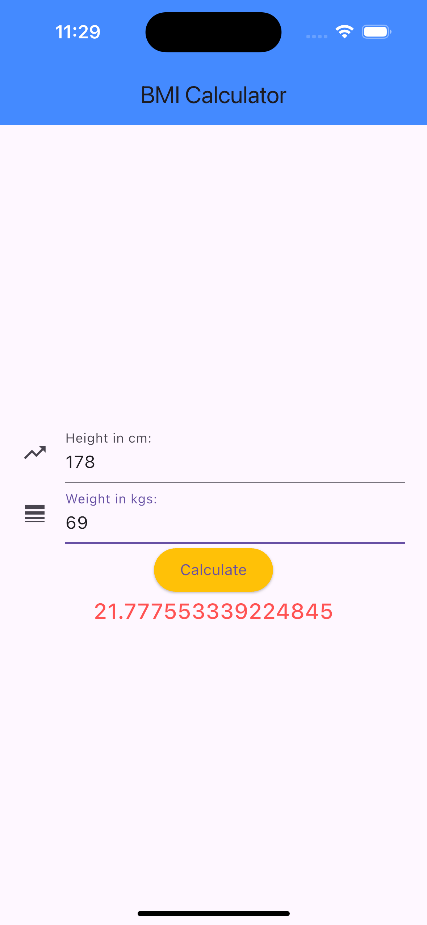
|  |  |
| --- | --- |
| **Roll No.: K005** | **Name: Jal Bafana** |
| **Prog/Yr/Sem: Cyber Security/2nd/4th** | **Batch: K1** |
| **Date of Experiment: 09.04.2025** | **Date of Submission: 09.04.2025** |

1. **Program Scenario and Program code:** (Write Scenario and Paste your program code (Java, xml resource and layout)).

import 'package:flutter/material.dart';  
import 'demo.dart';  
import 'package:flutter/material.dart';  
  
void main() {  
 runApp(const BMIApp());  
}  
  
class BMIApp extends StatelessWidget {  
 const BMIApp({super.key});  
  
 // This widget is the root of your application.  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: 'BMI App',  
 debugShowCheckedModeBanner: false,  
 theme: ThemeData(  
 colorScheme: ColorScheme.fromSeed(seedColor: Colors.*deepPurple*),  
 useMaterial3: true,  
 ),  
 home: Demo(),  
 );  
 }  
}

import 'package:flutter/cupertino.dart';  
import 'package:flutter/material.dart';  
class Demo extends StatefulWidget{  
 \_DemoState createState()=>\_DemoState();  
  
}  
class \_DemoState extends State {  
 final TextEditingController \_heightController = TextEditingController();  
 final TextEditingController \_weightController = TextEditingController();  
 double \_result = 0.0;  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('BMI Calculator'),  
 backgroundColor: Colors.*blueAccent*,  
 ),  
 body: Container(  
 padding: EdgeInsets.symmetric(horizontal: 20.0),  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 TextField(  
 controller: \_heightController,  
 keyboardType: TextInputType.*number*,  
 decoration: InputDecoration(  
 labelText: 'Height in cm:',  
 icon: Icon(Icons.*trending\_up*),  
 ),  
 ),  
 TextField(  
 controller: \_weightController,  
 keyboardType: TextInputType.*number*,  
 decoration: InputDecoration(  
 labelText: 'Weight in kgs:',  
 icon: Icon(Icons.*line\_weight*),  
 ),  
 ),  
 ElevatedButton(  
 child: Text('Calculate'),  
 style: ElevatedButton.*styleFrom*(  
 backgroundColor: Colors.*amber*,  
 textStyle: TextStyle(color: Colors.*white*),  
 ),  
 onPressed: calculateBMI,  
 ),  
 Text(  
 \_result == 0.0 ? "Enter value" : "${\_result}",  
 style: TextStyle(  
 color: Colors.*redAccent*,  
 fontSize: 20,  
 fontWeight: FontWeight.*w500*,  
 )  
 )  
 ]  
 )  
 )  
 )  
 }  
 void calculateBMI(){  
 double height = double.*parse*(\_heightController.text) / 100;  
 double weight = double.*parse*(\_weightController.text);  
 double heightsquare = height \* height;  
 double result = weight / heightsquare;  
 \_result = result;  
 setState(() {});  
 }  
}

1. **Output:** (Paste your program input and output screen shots).

****

1. **Observations:** A brief description of the design aspects and working of the code in your own words.

The BMI Calculator app takes height and weight inputs via two TextField widgets. When the user presses the ElevatedButton, the BMI is calculated and displayed. The app uses a StatefulWidget to update the result. However, it lacks input validation for error handling.

1. **Questions:** Draw & Explain with respect to layouts the Scene Graph of the experiment.

Scaffold

├── AppBar

│ └── Text("BMI Calculator")

├── Body (Container)

│ └── Padding

│ └── Column

│ ├── TextField (Height input)

│ ├── TextField (Weight input)

│ ├── ElevatedButton (Calculate)

│ └── Text (BMI Result)

1. **Conclusion (Learning Outcomes):** How were the outcomes defined for the experiment in Part A fulfilled through the scenarios?

The app calculates BMI using Flutter widgets, with dynamic UI updates through StatefulWidget. It’s simple, but can be enhanced with input validation for better user experience.