



**RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054**  
**(Autonomous Institute, Affiliated to VTU)**

**Department of Computer Science & Engineering**

**Internship Report**

**on**

**Mobile Application Development**

**INT411: Intra Institutional Internship**

**STUDENT NAME :Rifah Balquees**  
**Nandita Nayak K**

**USN :1MS22CS114**  
**1MS22CS088**

---

**Ramaiah Institute of Technology**

**(Autonomous Institute, Affiliated to VTU)**  
**MSR Nagar, MSRIT Post, Bangalore-560054**

**October-November, 2023**



**RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054**  
**(Autonomous Institute, Affiliated to VTU)**

**Department of Computer Science & Engineering**

**CERTIFICATE**

This is to certify that Mr./Ms. \_\_\_\_\_,  
a student of Bachelor of Engineering, bearing USN: \_\_\_\_\_, has successfully  
completed, 20 Hours: from 25.10.2023 to 8.11.2023 Intra Institutional Internship in Mobile  
Application Development from the Department of Computer Science & Engineering, M S Ramaiah  
Institute of Technology, Bangalore.

| SL No.      | Component             | Maximum Marks | Marks Obtained |
|-------------|-----------------------|---------------|----------------|
| 1           | Continuous Evaluation | 50            |                |
| 2           | Presentation          | 20            |                |
| 3           | Report                | 30            |                |
| Total Marks |                       | 100           |                |

**Signature of the Student with Date**

**Signature of the Faculty Co-Ordinator**

**Signature of Head of the Department**

## **OVERVIEW OF INTERNSHIP ACTIVITIES**

| <b>DATE</b> | <b>DAY</b> | <b>NAME OF THE TOPIC COMPLETED</b> |
|-------------|------------|------------------------------------|
|             | Monday     |                                    |
|             | Tuesday    |                                    |
|             | Wednesday  |                                    |
|             | Thursday   |                                    |
|             | Friday     |                                    |
|             | Saturday   |                                    |



# **INTRODUCTION**

This report provides an overview of the development of a BMI (Body Mass Index) calculator application using the Flutter framework. The BMI calculator is a simple yet useful mobile application designed to help users calculate their BMI, a measure of a person's weight in relation to their height, and assess their general health status.

## **Project Description**

### **Purpose**

The primary goal of this project is to create a user-friendly BMI calculator application that can be used on Android and iOS devices. This application enables users to input their weight and height, calculates their BMI, and provides a brief interpretation of the results.

Technologies Used:

1. Flutter Framework: Flutter is a popular open-source framework for building natively compiled applications for mobile from a single codebase.
2. Dart Programming Language: Dart is the programming language used for building Flutter applications.

### **Features**

The BMI calculator application has the following features:

1. Input Fields: Users can enter their weight (in kilograms) and height (in centimeters) using sliders.
2. Calculate BMI: After inputting their weight and height, users can press a "Calculate BMI" button to perform the BMI calculation.
3. Result Display: The calculated BMI is displayed on the screen along with an interpretation of the result, indicating whether the user is underweight, normal weight, overweight, or obese.
4. Health Information: The application provides brief information about BMI and its significance for the user's health.

## **Development Process**

1. Project Setup: A new Flutter project was created using the Flutter CLI.
2. User Interface Design: The user interface (UI) was designed using Flutter widgets and layout components. It includes input fields for weight and height, a "Calculate BMI" button, and an area to display the result.
3. BMI Calculation: The application logic was implemented to calculate the BMI based on the user's input.
4. Result Interpretation: A function was created to interpret the BMI result and provide a simple health status message.
5. Testing: Extensive testing was conducted to ensure the application functions as expected and handles edge cases gracefully.
6. Deployment: The application was built for Android and iOS platforms and deployed to app stores.

## **Future Enhancements**

While the current version of the BMI calculator is functional, there are several opportunities for improvement:

1. User Profiles: Implement user profiles to allow users to save their BMI results and track changes over time.
2. Improved User Experience: Enhance the application's design and user experience to make it more visually appealing and intuitive.
3. Health Tips: Provide health and fitness tips based on the user's BMI category.

## Code:

Main.dart

```
import 'package:mad2/pages/show_details.dart';
import 'package:flutter/material.dart';

class HomePage extends StatefulWidget {
  static const String routeName = '/';
  const HomePage({Key? key}) : super(key: key);

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {

  double sliderWeight = 50;
  double sliderHeight= 1.5;
  double bmi = 0;
  String status = '';

  @override
  Widget build(BuildContext context) {
    return Scaffold(

      appBar: AppBar(
        title: Text('Welcome'),
        centerTitle: true,
      ),
      bottomNavigationBar: Container(
        height: 50,
        color: Color.fromARGB(255, 1, 19, 34),
        alignment: Alignment.bottomCenter,
        child: Padding(
          padding: const EdgeInsets.all(8.0),

        ),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          children: [
            const Text('Weight',style: TextStyle(fontSize: 25)),
            Slider(
              value: sliderWeight,
              min: 20,
              max: 120,
              divisions: 100,
              label: '${sliderWeight.round()}',
              onChanged: (value){
```

```

        setState((){
            sliderWeight = value;
            calculateBmi();
        });
    )),
    Text('${sliderWeight.toStringAsFixed(2)} kg', style:
TextStyle(fontSize: 25),),
    SizedBox(height: 30,),
    const Text('Height', style: TextStyle(fontSize: 25),),
    Slider(
        value: sliderHeight,
        min: 1.2,
        max: 2.2,
        divisions: 10,
        label: '${sliderHeight.round()}',
        onChanged: (value){
            setState((){
                sliderHeight = value;
                calculateBmi();
            });
        )),
    Text('${sliderHeight.toStringAsFixed(1)} m', style:
TextStyle(fontSize: 25),),
    SizedBox(height: 30,),
    Container(
        alignment: Alignment.center,
        height: 150,
        width: 150,
        decoration: BoxDecoration(
            color: Colors.blue,
            shape: BoxShape.circle
        ),
        child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
                Text('BMI', style: TextStyle(fontSize: 40, color:
Colors.white),),
                Text('${bmi.toStringAsFixed(1)}', style: TextStyle(fontSize:
40, color: Colors.white),),
            ],
        ),
    ),
    SizedBox(height: 20,),
    Center(child: Text(status, style: TextStyle(fontSize: 20, fontWeight:
FontWeight.bold),),),
    SizedBox(height: 20,),
    SizedBox(
        height: 40,
        child: ElevatedButton(
            //onPressed: () => Navigator.pushNamed(context,
SecondPage.routeName),

```



```

        //onPressed: () => Navigator.pushReplacementNamed(context,
SecondPage.routeName),
        onPressed: () => Navigator
            .pushNamed(context, ShowDetails.routeName, arguments: bmi),
        child: const Text('View Details'),
    ),
),
    SizedBox(height: 20,),
],
),
),
);
}

```

```

void calculateBmi() {
    bmi = sliderWeight/(sliderHeight*sliderHeight);
    if(bmi<16){
        status = 'Underweight (Severe thinness)';
    } else if(bmi>=16 && bmi <=16.9){
        status = 'Underweight (Moderate thinness)';
    }else if(bmi>=17 && bmi <=18.4){
        status = 'Underweight (Mild thinness)';
    }else if(bmi>=19 && bmi <=24.9){
        status = 'Normal';
    }else if(bmi>=25 && bmi <=29.9){
        status = 'Overweight (Pre-obese)';
    }else if(bmi>=30 && bmi <=34.9){
        status = 'Obese (Class I)';
    }else if(bmi>=35 && bmi <=39.9){
        status = 'Obese (Class II)';
    } else {
        status = 'Obese (Class III)';
    }
}
}

```

```

import 'package:flutter/material.dart';
class ShowDetails extends StatefulWidget {
  static const String routeName = '/show_details';
  const ShowDetails({Key? key}) : super(key: key);
  @override
  State<ShowDetails> createState() => _ShowDetailsState();
}
class _ShowDetailsState extends State<ShowDetails> {
  @override
  void initState() {
    print('InitStake Call');
    super.initState();
  }
  @override
  void didChangeDependencies() {
    bmi = ModalRoute.of(context)!.settings.arguments as double;
    print('BMI is: $bmi');
    super.didChangeDependencies();
  }
  late double bmi;
  @override
  Widget build(BuildContext context) {
    print('Build Call');
    return Scaffold(
      appBar: AppBar(
        title: const Text('Show Details'),
        centerTitle: true,
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.start,
            children: [

              SizedBox(height: 10,),
              DataTable(
                headingRowColor:
                  MaterialStateColor.resolveWith((states) => Colors.blue),
                columns: [
                  DataColumn(
                    label: Text('Category',style: TextStyle(fontSize: 20,,)),
                  ),
                  DataColumn(
                    label: Text('BMI',style: TextStyle(fontSize: 20,,)),
                  ),
                ],
                rows: [
                  DataRow(
                    color: MaterialStateColor.resolveWith((states) {
                      return bmi<16 ? Colors.green : Colors.white;}),

```

```

        cells: [
            DataCell(Text('Underweight (Severe thinness)'),
                DataCell(Text('< 16.0'))),
        ]),
    DataRow(
        color: MaterialStateColor.resolveWith((states) {
            return bmi>=16 && bmi<=16.9 ? Colors.yellow :
Colors.white;})),
        cells: [
            DataCell(Text('Underweight (Moderate thinness)'),
                DataCell(Text('16-16.9'))),
        ]),
    DataRow(
        color: MaterialStateColor.resolveWith((states) {
            return bmi>=17 && bmi<=18.4 ? Colors.yellow :
Colors.white;})),
        cells: [
            DataCell(Text('Underweight (Mild thinness)'),
                DataCell(Text('17-18.4'))),
        ]),
    DataRow(
        color: MaterialStateColor.resolveWith((states) {
            return bmi>=19 && bmi<=24.9 ? Colors.green :
Colors.white;})),
        cells: [
            DataCell(Text('Normal')),
            DataCell(Text('19-24.9')),
        ]),
    DataRow(
        color: MaterialStateColor.resolveWith((states) {
            return bmi>=25 && bmi<=29.9 ? Colors.orange :
Colors.white;})),
        cells: [
            DataCell(Text('Overweight (Pre-Obese)'),
                DataCell(Text('25-29.9'))),
        ]),
    DataRow(
        color: MaterialStateColor.resolveWith((states) {
            return bmi>=30 && bmi<=34.9 ? Colors.red :
Colors.white;})),
        cells: [
            DataCell(Text('Obese(Class I)'),
                DataCell(Text('30-34.9'))),
        ]),
    DataRow(
        color: MaterialStateColor.resolveWith((states) {
            return bmi>=35 && bmi<=39.9 ? Colors.red :
Colors.white;})),
        cells: [
            DataCell(Text('Obese(Class II)'),
                DataCell(Text('35-39.9'))),

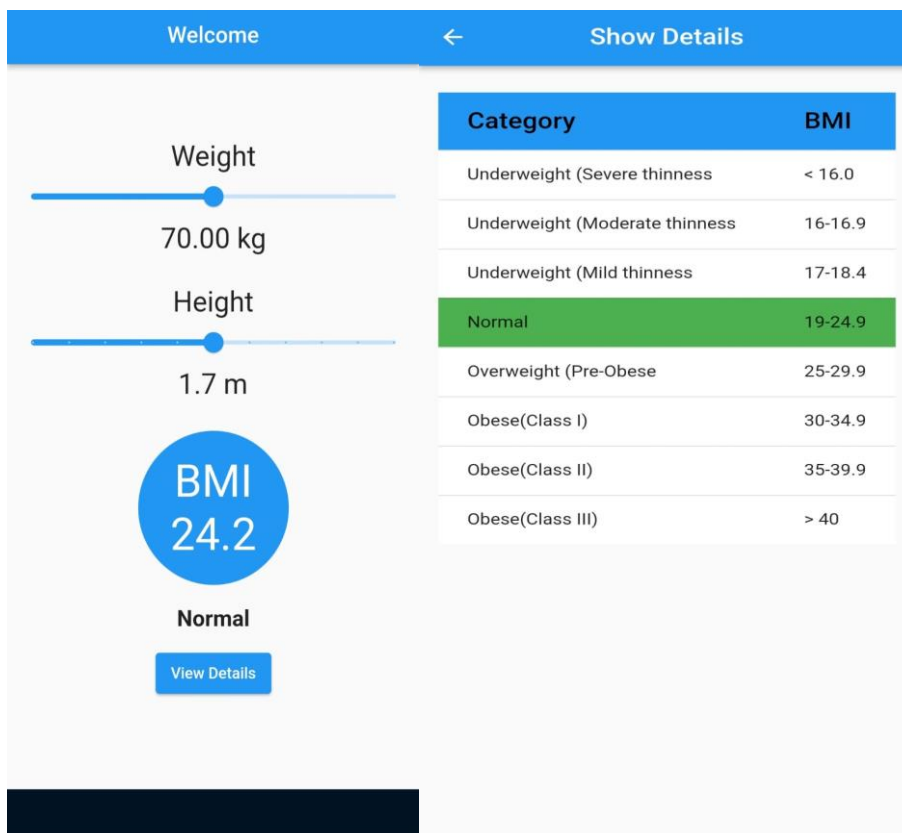
```

```

    ]),
    DataRow(
      color: MaterialStateColor.resolveWith((states) {
        return bmi>40 ? Colors.red : Colors.white;}),
      cells: [
        DataCell(Text('Obese(Class III)'),
          DataCell(Text('> 40')),
        ]),
    ]),
  ],
),
),
),
);
}
}

```

## RESULT SNIPPIT



## Conclusion

The BMI calculator application developed using Flutter is a useful tool for individuals to quickly assess their BMI and understand their general health status. It showcases the capabilities of Flutter in building cross-platform mobile applications efficiently. Further development and enhancements can make this application even more valuable to users interested in tracking their health and fitness.