**Objective:-** To design Flutter UI by including common widgets, Design BMI Calculator as per given screenshot.

**Code :-**

import 'package:flutter/material.dart';

void main() {

  runApp(const MyApp()); *// Entry point of the app*

}

class MyApp extends StatelessWidget {

  const MyApp({*super*.key});

*// This widget is the root of your application.*

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      theme: ThemeData(

        brightness: Brightness.light, *// Light theme settings*

        primaryColor: Colors.blue,

      ),

      darkTheme: ThemeData(

        brightness: Brightness.dark, *// Dark theme settings*

        scaffoldBackgroundColor:

            const Color(0xFF121212), *// Background for Scaffold in dark mode*

        appBarTheme: const AppBarTheme(

          backgroundColor: Color(0xFF1F1F1F), *// Dark app bar color*

          iconTheme:

              IconThemeData(color: Colors.white), *// Icon color in app bar*

          titleTextStyle: TextStyle(color: Colors.white, fontSize: 20),

        ),

        textButtonTheme: TextButtonThemeData(

          style: TextButton.styleFrom(

            foregroundColor: Colors.white,

            backgroundColor: Colors.blue, *// TextButton background color*

          ),

        ),

        inputDecorationTheme: InputDecorationTheme(

          filled: true,

          fillColor:

              Colors.grey[800], *// Input field background color in dark mode*

          border: OutlineInputBorder(

            borderRadius: BorderRadius.circular(8.0),

            borderSide: BorderSide.none, *// No border for input fields*

          ),

          hintStyle: const TextStyle(color: Colors.grey), *// Hint text style*

        ),

        textTheme: const TextTheme(

          bodyMedium: TextStyle(color: Colors.white), *// Body text color*

          titleLarge: TextStyle(color: Colors.white), *// Large title text color*

        ),

      ),

      themeMode: ThemeMode.dark, *// Sets the dark theme as the default*

      home: const BMICalculator(), *// Home widget is the BMICalculator*

      debugShowCheckedModeBanner: false, *// Disable the debug banner*

    );

  }

}

class BMICalculator extends StatefulWidget {

  const BMICalculator({*super*.key});

  @override

  State<BMICalculator> createState() => \_BMICalculatorState();

}

class \_BMICalculatorState extends State<BMICalculator> {

  int currentindex = 0; *// Track selected gender*

  String result = ""; *// Variable to store BMI result*

  double height = 0.0; *// User height*

  double weight = 0.0; *// User weight*

  TextEditingController heightController =

      TextEditingController(); *// Controller for height input*

  TextEditingController weightController =

      TextEditingController(); *// Controller for weight input*

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: const Text("BMI Calculator"),

        backgroundColor: Theme.of(context).appBarTheme.backgroundColor,

        actions: [

          IconButton(

            onPressed: () {}, *// Placeholder for settings action*

            icon: const Icon(Icons.settings),

          ),

        ],

      ),

      body: SingleChildScrollView(

        child: Padding(

          padding: const EdgeInsets.all(12.0), *// Padding for content*

          child: Column(

            mainAxisAlignment: MainAxisAlignment.start,

            crossAxisAlignment: CrossAxisAlignment.start,

            children: [

              Row(

                children: [

                  radioButton("Man", Colors.blue, 0), *// Male option*

                  radioButton("Women", Colors.pink, 1), *// Female option*

                ],

              ),

              const SizedBox(height: 20.0),

              const Text(

                "Your Height in Centimeter :- ",

                style: TextStyle(fontSize: 18.0), *// Label for height input*

              ),

              const SizedBox(height: 8.0),

              TextField(

                controller: heightController,

                keyboardType: TextInputType.number,

                textAlign: TextAlign.center,

                decoration: InputDecoration(

                  hintText:

                      "Your Height in Centimeter", *// Placeholder for height input*

                  filled: true,

                  fillColor: Theme.of(context).inputDecorationTheme.fillColor,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(8.0),

                    borderSide: BorderSide.none, *// No border for height input*

                  ),

                ),

              ),

              const SizedBox(height: 20.0),

              const Text(

                "Your Weight in Kilogram :- ",

                style: TextStyle(fontSize: 18.0), *// Label for weight input*

              ),

              const SizedBox(height: 8.0),

              TextField(

                controller: weightController,

                keyboardType: TextInputType.number,

                textAlign: TextAlign.center,

                decoration: InputDecoration(

                  hintText:

                      "Your Weight in Kilogram", *// Placeholder for weight input*

                  filled: true,

                  fillColor: Theme.of(context).inputDecorationTheme.fillColor,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(8.0),

                    borderSide: BorderSide.none, *// No border for weight input*

                  ),

                ),

              ),

              const SizedBox(height: 20.0),

              SizedBox(

                width: double.infinity,

                height: 50.0,

                child: TextButton(

                  onPressed: () {

                    setState(() {

*// Parsing the input values*

                      height = double.parse(heightController.value.text);

                      weight = double.parse(weightController.value.text);

                    });

                    calculateBMI(height, weight); *// Trigger BMI calculation*

                  },

                  child: const Text(

                    "Calculate",

                    style: TextStyle(color: Colors.white),

                  ),

                ),

              ),

              const SizedBox(height: 20.0),

              const Center(

                child: Text(

                  "Your BMI is :- ",

                  textAlign: TextAlign.center, *// Label for BMI result*

                  style: TextStyle(

                    fontSize: 24.0,

                    fontWeight: FontWeight.bold,

                  ),

                ),

              ),

              const SizedBox(height: 50.0),

              Center(

                child: Text(

                  result, *// Display calculated BMI*

                  textAlign: TextAlign.center,

                  style: const TextStyle(

                    fontSize: 40.0,

                    fontWeight: FontWeight.bold,

                  ),

                ),

              ),

            ],

          ),

        ),

      ),

    );

  }

*// Function to calculate BMI based on height and weight*

  void calculateBMI(double height, double weight) {

    double finalresult = weight / (height \* height / 10000); *// Formula for BMI*

    String bmi =

        finalresult.toStringAsFixed(2); *// Format result to 2 decimal places*

    setState(() {

      result = bmi; *// Update result in the UI*

    });

  }

*// Function to change the selected gender index*

  void changeIndex(int index) {

    setState(() {

      currentindex = index;

    });

  }

*// Reusable radio button widget for gender selection*

  Widget radioButton(String value, Color color, int index) {

    return Expanded(

      child: Container(

        margin: const EdgeInsets.symmetric(horizontal: 12.0),

        height: 80.0,

        child: TextButton(

          style: TextButton.styleFrom(

            backgroundColor: currentindex == index ? color : Colors.grey[800],

            shape: RoundedRectangleBorder(

              borderRadius: BorderRadius.circular(8.0),

            ),

          ),

          onPressed: () {

            changeIndex(index); *// Update selected gender*

          },

          child: value == "Man"

              ? Icon(

                  Icons.male,

                  color: currentindex == index ? Colors.white : color,

                  size: 40.0,

                )

              : Icon(

                  Icons.female,

                  color: currentindex == index ? Colors.white : color,

                  size: 40.0,

                ),

        ),

      ),

    );

  }

}

**Output:-**

