import 'package:flutter/material.dart';  
import 'package:untitled6/screens/input\_page.dart';  
  
void main() => runApp(BMICalculator());  
  
class BMICalculator extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 theme: ThemeData.dark().copyWith(  
 primaryColor: Color(0xFF0A0E21),  
 scaffoldBackgroundColor: Color(0xFF0A0E21),  
 ),  
 home: InputPage(),  
 );  
 }

}

Main.Dart

Input\_page.Dart

import 'package:flutter/material.dart';  
import 'package:font\_awesome\_flutter/font\_awesome\_flutter.dart';  
import 'package:untitled6/components/icon\_content.dart';  
import 'package:untitled6/components/reusable\_card.dart';  
import 'package:untitled6/constants.dart';  
import 'package:untitled6/screens/results\_page.dart';  
import 'package:untitled6/components/bottom\_button.dart';  
import 'package:untitled6/components/round\_icon\_button.dart';  
import 'package:untitled6/calculator\_brain.dart';  
  
enum Gender {  
 male,  
 female,  
}  
  
class InputPage extends StatefulWidget {  
 @override  
 \_InputPageState createState() => \_InputPageState();  
}  
  
class \_InputPageState extends State<InputPage> {  
 Gender selectedGender;  
 int height = 50;  
 int weight = 10;  
 int age = 5;  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('BMI CALCULATOR'),  
 ),  
 body: Column(  
 crossAxisAlignment: CrossAxisAlignment.stretch,  
 children: <Widget>[  
 Expanded(  
 child: Row(  
 children: <Widget>[  
 Expanded(  
 child: ReusableCard(  
 onPress: () {  
 setState(() {  
 selectedGender = Gender.male;  
 });  
 },  
 colour: selectedGender == Gender.male  
 ? kActiveCardColour  
 : kInactiveCardColour,  
 cardChild: IconContent(  
 icon: FontAwesomeIcons.*mars*,  
 label: '1 to 50',  
 ),  
 ),  
 ),  
 Expanded(  
 child: ReusableCard(  
 onPress: () {  
 setState(() {  
 selectedGender = Gender.female;  
 });  
 },  
 colour: selectedGender == Gender.female  
 ? kActiveCardColour  
 : kInactiveCardColour,  
 cardChild: IconContent(  
 icon: FontAwesomeIcons.*venus*,  
 label: '1 to 100',  
 ),  
 ),  
 ),  
 ],  
 )),  
 Expanded(  
 child: ReusableCard(  
 colour: kActiveCardColour,  
 cardChild: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Text(  
 'HEIGHT',  
 style: kLabelTextStyle,  
 ),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 crossAxisAlignment: CrossAxisAlignment.baseline,  
 textBaseline: TextBaseline.alphabetic,  
 children: <Widget>[  
 Text(  
 height.toString(),  
 style: kNumberTextStyle,  
 ),  
 Text(  
 'cm',  
 style: kLabelTextStyle,  
 )  
 ],  
 ),  
 SliderTheme(  
 data: SliderTheme.*of*(context).copyWith(  
 inactiveTrackColor: Color(0xFF8D8E98),  
 activeTrackColor: Colors.*white*,  
 thumbColor: Color(0xFFEB1555),  
 overlayColor: Color(0x29EB1555),  
 thumbShape:  
 RoundSliderThumbShape(enabledThumbRadius: 15.0),  
 overlayShape:  
 RoundSliderOverlayShape(overlayRadius: 30.0),  
 ),  
 child: Slider(  
 value: height.toDouble(),  
 min: 1.0,  
 max: 50.0,  
 onChanged: (double newValue) {  
 setState(() {  
 height = newValue.round();  
 });  
 },  
 ),  
 ),  
 ],  
 ),  
 ),  
 ),  
 Expanded(  
 child: Row(  
 children: <Widget>[  
 Expanded(  
 child: ReusableCard(  
 colour: kActiveCardColour,  
 cardChild: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Text(  
 'Addition',  
 style: kLabelTextStyle,  
 ),  
 Text(  
 weight.toString(),  
 style: kNumberTextStyle,  
 ),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 RoundIconButton(  
 icon: FontAwesomeIcons.*minus*,  
 onPressed: () {  
 setState(() {  
 weight--;  
 });  
 }),  
 SizedBox(  
 width: 10.0,  
 ),  
 RoundIconButton(  
 icon: FontAwesomeIcons.*plus*,  
 onPressed: () {  
 setState(() {  
 weight++;  
 });  
 },  
 ),  
 ],  
 ),  
 ],  
 ),  
 ),  
 ),  
 Expanded(  
 child: ReusableCard(  
 colour: kActiveCardColour,  
 cardChild: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Text(  
 'Power',  
 style: kLabelTextStyle,  
 ),  
 Text(  
 age.toString(),  
 style: kNumberTextStyle,  
 ),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 RoundIconButton(  
 icon: FontAwesomeIcons.*minus*,  
 onPressed: () {  
 setState(  
 () {  
 age--;  
 },  
 );  
 },  
 ),  
 SizedBox(  
 width: 10.0,  
 ),  
 RoundIconButton(  
 icon: FontAwesomeIcons.*plus*,  
 onPressed: () {  
 setState(() {  
 age++;  
 });  
 })  
 ],  
 )  
 ],  
 ),  
 ),  
 ),  
 ],  
 ),  
 ),  
 BottomButton(  
 buttonTitle: 'CALCULATE',  
 onTap: () {  
 CalculatorBrain calc =  
 CalculatorBrain(height: height, weight: weight,age: age);  
  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => ResultsPage(  
 bmiResult: calc.calculateBMI(),  
 resultText: calc.getResult(),  
 interpretation: calc.getInterpretation(),  
 ),  
 ),  
 );  
 },  
 ),  
 ],  
 ),  
 );  
 }  
}

Constants.Dart

import 'package:flutter/material.dart';  
  
const kBottomContainerHeight = 80.0;  
const kActiveCardColour = Color(0xFF1D1E33);  
const kInactiveCardColour = Color(0xFF111328);  
const kBottomContainerColour = Color(0xFFEB1555);  
  
const kLabelTextStyle = TextStyle(  
 fontSize: 18.0,  
 color: Color(0xFF8D8E98),  
);  
  
const kNumberTextStyle = TextStyle(  
 fontSize: 50.0,  
 fontWeight: FontWeight.*w900*,  
);  
  
const kLargeButtonTextStyle = TextStyle(  
 fontSize: 25.0,  
 fontWeight: FontWeight.*bold*,  
);  
  
const kTitleTextStyle = TextStyle(  
 fontSize: 50.0,  
 fontWeight: FontWeight.*bold*,  
);  
  
const kResultTextStyle = TextStyle(  
 color: Color(0xFF24D876),  
 fontSize: 22.0,  
 fontWeight: FontWeight.*bold*,  
);  
  
const kBMITextStyle = TextStyle(  
 fontSize: 100.0,  
 fontWeight: FontWeight.*bold*,  
);  
  
const kBodyTextStyle = TextStyle(  
 fontSize: 22.0,  
);

Calculator\_brain.dart

import 'dart:math';  
  
  
class CalculatorBrain {  
 CalculatorBrain({this.height, this.weight,this.age});  
  
 final int height;  
 final int weight;  
 final int age;  
 double \_bmi;  
  
 String calculateBMI() {  
 //\_bmi = weight / pow(height / 100, 2);  
 \_bmi=pow((height + weight),age);  
  
 return \_bmi.toStringAsFixed(1);  
 }  
  
 String getResult() {  
 if (\_bmi >= 500000) {  
 return 'Heigh result';  
 } else if (\_bmi > 50000) {  
 return 'Normal';  
 } else {  
 return 'Underweight';  
 }  
 }  
  
 String getInterpretation() {  
 if (\_bmi >= 25) {  
 return 'You have a higher than normal body weight. Try to exercise more.';  
 } else if (\_bmi >= 18.5) {  
 return 'You have a normal body weight. Good job!';  
 } else {  
 return 'You have a lower than normal body weight. You can eat a bit more.';  
 }  
 }  
}

Result\_page.dart

import 'package:flutter/material.dart';  
import 'package:untitled6/constants.dart';  
import 'package:untitled6/components/reusable\_card.dart';  
import 'package:untitled6/components/bottom\_button.dart';  
  
class ResultsPage extends StatelessWidget {  
 ResultsPage(  
 {@required this.interpretation,  
 @required this.bmiResult,  
 @required this.resultText});  
  
 final String bmiResult;  
 final String resultText;  
 final String interpretation;  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('BMI CALCULATOR'),  
 ),  
 body: Column(  
 mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
 crossAxisAlignment: CrossAxisAlignment.stretch,  
 children: <Widget>[  
 Expanded(  
 child: Container(  
 padding: EdgeInsets.all(15.0),  
 alignment: Alignment.*bottomLeft*,  
 child: Text(  
 'Your Result',  
 style: kTitleTextStyle,  
 ),  
 ),  
 ),  
 Expanded(  
 flex: 5,  
 child: ReusableCard(  
 colour: kActiveCardColour,  
 cardChild: Column(  
 mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: <Widget>[  
 Text(  
 resultText.toUpperCase(),  
 style: kResultTextStyle,  
 ),  
 Text(  
 bmiResult,  
 style: kBMITextStyle,  
 ),  
 Text(  
 interpretation,  
 textAlign: TextAlign.center,  
 style: kBodyTextStyle,  
 ),  
 ],  
 ),  
 ),  
 ),  
 BottomButton(  
 buttonTitle: 'RE-CALCULATE',  
 onTap: () {  
 Navigator.*pop*(context);  
 },  
 )  
 ],  
 ),  
 );  
 }  
}

Round\_icons button.dart

import 'package:flutter/material.dart';  
  
class RoundIconButton extends StatelessWidget {  
 RoundIconButton({@required this.icon, @required this.onPressed});  
  
 final IconData icon;  
 final Function onPressed;  
  
 @override  
 Widget build(BuildContext context) {  
 return RawMaterialButton(  
 elevation: 0.0,  
 child: Icon(icon),  
 onPressed: onPressed,  
 constraints: BoxConstraints.tightFor(  
 width: 56.0,  
 height: 56.0,  
 ),  
 shape: CircleBorder(),  
 fillColor: Color(0xFF4C4F5E),  
 );  
 }  
}

Reuseable\_card.dart

import 'package:flutter/material.dart';  
  
class ReusableCard extends StatelessWidget {  
 ReusableCard({@required this.colour, this.cardChild, this.onPress});  
  
 final Color colour;  
 final Widget cardChild;  
 final Function onPress;  
  
 @override  
 Widget build(BuildContext context) {  
 return GestureDetector(  
 onTap: onPress,  
 child: Container(  
 child: cardChild,  
 margin: EdgeInsets.all(15.0),  
 decoration: BoxDecoration(  
 color: colour,  
 borderRadius: BorderRadius.circular(10.0),  
 ),  
 ),  
 );  
 }  
}

Icon\_content.dart

import 'package:flutter/material.dart';  
import 'package:untitled6/constants.dart';  
  
class IconContent extends StatelessWidget {  
 IconContent({this.icon, this.label});  
  
 final IconData icon;  
 final String label;  
  
 @override  
 Widget build(BuildContext context) {  
 return Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Icon(  
 icon,  
 size: 80.0,  
 ),  
 SizedBox(  
 height: 15.0,  
 ),  
 Text(  
 label,  
 style: kLabelTextStyle,  
 )  
 ],  
 );  
 }  
}

Buttom\_Button.dart

import 'package:flutter/material.dart';  
import 'package:untitled6/constants.dart';  
  
class BottomButton extends StatelessWidget {  
 BottomButton({@required this.onTap, @required this.buttonTitle});  
  
 final Function onTap;  
 final String buttonTitle;  
  
 @override  
 Widget build(BuildContext context) {  
 return GestureDetector(  
 onTap: onTap,  
 child: Container(  
 child: Center(  
 child: Text(  
 buttonTitle,  
 style: kLargeButtonTextStyle,  
 ),  
 ),  
 color: kBottomContainerColour,  
 margin: EdgeInsets.only(top: 10.0),  
 padding: EdgeInsets.only(bottom: 20.0),  
 width: double.*infinity*,  
 height: kBottomContainerHeight,  
 ),  
 );  
 }  
}

PUBSPEC.Yaml

name: untitled6  
description: A new Flutter project.  
  
*# The following line prevents the package from being accidentally published to  
# pub.dev using `flutter pub publish`. This is preferred for private packages.*publish\_to: 'none' *# Remove this line if you wish to publish to pub.dev  
  
# The following defines the version and build number for your application.  
# A version number is three numbers separated by dots, like 1.2.43  
# followed by an optional build number separated by a +.  
# Both the version and the builder number may be overridden in flutter  
# build by specifying --build-name and --build-number, respectively.  
# In Android, build-name is used as versionName while build-number used as versionCode.  
# Read more about Android versioning at https://developer.android.com/studio/publish/versioning  
# In iOS, build-name is used as CFBundleShortVersionString while build-number used as CFBundleVersion.  
# Read more about iOS versioning at  
# https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html*version: 1.0.0+1  
  
environment:  
 sdk: ">=2.1.0 <3.0.0"  
  
dependencies:  
 flutter:  
 sdk: flutter  
  
 cupertino\_icons: ^0.1.2  
 font\_awesome\_flutter: ^8.4.0  
  
dev\_dependencies:  
 flutter\_test:  
 sdk: flutter  
  
flutter:  
 uses-material-design: true

Result

A screenshot of a computer

Description automatically generated with medium confidence

BMI

Graphical user interface, application, website

Description automatically generated