**Calculator App**

**BY-Shubhkarman Singh**

**CODE:**

**-Main.dart**

// ignore\_for\_file: use\_key\_in\_widget\_constructors, library\_private\_types\_in\_public\_api  
  
import 'package:flutter/material.dart';  
import 'buttons.dart';  
import 'package:math\_expressions/math\_expressions.dart';  
  
void main() {  
 runApp(const MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 const MyApp({Key? key}) : super(key: key);  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 debugShowCheckedModeBanner: false,  
 home: HomePage(),  
 ); // MaterialApp  
 }  
}  
  
class HomePage extends StatefulWidget {  
 @override  
 \_HomePageState createState() => \_HomePageState();  
}  
  
class \_HomePageState extends State<HomePage> {  
 var userInput = '';  
 var answer = '';  
  
// Array of button  
 final List<String> buttons = [  
 'C',  
 '+/-',  
 '%',  
 'DEL',  
 '7',  
 '8',  
 '9',  
 '/',  
 '4',  
 '5',  
 '6',  
 'x',  
 '1',  
 '2',  
 '3',  
 '-',  
 '0',  
 '.',  
 '=',  
 '+',  
 ];  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: const Text("CALCULATOR \n"  
 "BY Shubhkarman Singh"),  
 ), //AppBar  
 backgroundColor: Colors.*black*,  
 body: Column(  
 children: <Widget>[  
 Expanded(  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
 children: <Widget>[  
 Container(  
 padding: const EdgeInsets.all(20),  
 alignment: Alignment.*centerRight*,  
 child: Text(  
 userInput,  
 style: const TextStyle(fontSize: 18, color: Colors.*white*),  
 ),  
 ),  
 Container(  
 padding: const EdgeInsets.all(15),  
 alignment: Alignment.*centerRight*,  
 child: Text(  
 answer,  
 style: const TextStyle(  
 fontSize: 30,  
 color: Colors.*white*,  
 fontWeight: FontWeight.*bold*),  
 ),  
 )  
 ]),  
 ),  
 Expanded(  
 flex: 3,  
 child: GridView.builder(  
 itemCount: buttons.length,  
 gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(  
 crossAxisCount: 4),  
 itemBuilder: (BuildContext context, int index) {  
 // Clear Button  
 if (index == 0) {  
 return MyButton(  
 buttontapped: () {  
 setState(() {  
 userInput = '';  
 answer = '0';  
 });  
 },  
 buttonText: buttons[index],  
 color: Colors.*deepOrangeAccent*,  
 textColor: Colors.*white*,  
 );  
 }  
  
 // +/- button  
 else if (index == 1) {  
 return MyButton(  
 buttonText: buttons[index],  
 color: Colors.*blue*[50],  
 textColor: Colors.*black*,  
 );  
 }  
 // % Button  
 else if (index == 2) {  
 return MyButton(  
 buttontapped: () {  
 setState(() {  
 userInput += buttons[index];  
 });  
 },  
 buttonText: buttons[index],  
 color: Colors.*blue*[50],  
 textColor: Colors.*black*,  
 );  
 }  
 // Delete Button  
 else if (index == 3) {  
 return MyButton(  
 buttontapped: () {  
 setState(() {  
 userInput =  
 userInput.substring(0, userInput.length - 1);  
 });  
 },  
 buttonText: buttons[index],  
 color: Colors.*blue*[50],  
 textColor: Colors.*black*,  
 );  
 }  
 // Equal\_to Button  
 else if (index == 18) {  
 return MyButton(  
 buttontapped: () {  
 setState(() {  
 equalPressed();  
 });  
 },  
 buttonText: buttons[index],  
 color: Colors.*orange*[700],  
 textColor: Colors.*white*,  
 );  
 }  
  
 // other buttons  
 else {  
 return MyButton(  
 buttontapped: () {  
 setState(() {  
 userInput += buttons[index];  
 });  
 },  
 buttonText: buttons[index],  
 color: isOperator(buttons[index])  
 ? Colors.*blueAccent* : Colors.*white*,  
 textColor: isOperator(buttons[index])  
 ? Colors.*white* : Colors.*black*,  
 );  
 }  
 }),  
 ),  
 ],  
 ),  
 );  
 }  
  
 bool isOperator(String x) {  
 if (x == '/' || x == 'x' || x == '-' || x == '+' || x == '=') {  
 return true;  
 }  
 return false;  
 }  
  
// function to calculate the input operation  
 void equalPressed() {  
 String finaluserinput = userInput;  
 finaluserinput = userInput.replaceAll('x', '\*');  
  
 Parser p = Parser();  
 Expression exp = p.parse(finaluserinput);  
 ContextModel cm = ContextModel();  
 double eval = exp.evaluate(EvaluationType.REAL, cm);  
 answer = eval.toString();  
 }  
}

**-Buttons.dart**

import 'package:flutter/material.dart';  
  
// creating Stateless Widget for buttons  
class MyButton extends StatelessWidget {  
  
// declaring variables  
 final color;  
 final textColor;  
 final String buttonText;  
 final buttontapped;  
  
//Constructor  
 MyButton({this.color, this.textColor, required this.buttonText, this.buttontapped});  
  
 @override  
 Widget build(BuildContext context) {  
 return GestureDetector(  
 onTap: buttontapped,  
 child: Padding(  
 padding: const EdgeInsets.all(0.2),  
 child: ClipRRect(  
 // borderRadius: BorderRadius.circular(25),  
 child: Container(  
 color: color,  
 child: Center(  
 child: Text(  
 buttonText,  
 style: TextStyle(  
 color: textColor,  
 fontSize: 25,  
 fontWeight: FontWeight.*bold*,  
 ),  
 ),  
 ),  
 ),  
 ),  
 ),  
 );  
 }  
}