**Flutter Assignment- 4**

Hitesh Walia, MCA

00311104422

Q10. Write a program to create the Application to perform the following operations on the number

Increment, Decrement, Square, Reset

Check Prime, Check Even/Odd, Check Palindrome, Compute Factorial

import 'package:flutter/material.dart';  
import 'dart:math';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatefulWidget {  
 @override  
 \_MyAppState createState() => \_MyAppState();  
}  
  
class \_MyAppState extends State<MyApp> {  
 int incrementCount = 0;  
 int decrementCount = 0;  
 int squareCount = 0;  
 int primeCount = 0;  
 int evenOddCount = 0;  
 int palindromeCount = 0;  
 int factorialCount = 0;  
 int userInput = 0;  
  
 String resultText = '';  
  
 bool isPrime(int number) {  
 if (number <= 1) return false;  
 if (number == 2) return true;  
 for (int i = 2; i <= sqrt(number); i++) {  
 if (number % i == 0) return false;  
 }  
 return true;  
 }  
  
 bool isEven(int number) {  
 return number % 2 == 0;  
 }  
  
 bool isPalindrome(int number) {  
 String numberStr = number.toString();  
 String reversedNumberStr = numberStr.split('').reversed.join('');  
 return numberStr == reversedNumberStr;  
 }  
  
 int factorial(int number) {  
 if (number == 0 || number == 1) return 1;  
 return number \* factorial(number - 1);  
 }  
  
 int findSquare(int number){  
 return number\*number;  
 }  
  
 void updateResultText(String action, int value) {  
 setState(() {  
 resultText =  
 "You have pushed the $action button this many times: $value\n";  
 });  
 }  
  
 void onButtonPress(String action) {  
 setState(() {  
 switch (action) {  
 case 'Increment':  
 incrementCount++;  
 userInput++;  
 updateResultText('increment', incrementCount);  
 resultText +='Incremented to $userInput\n';  
 break;  
 case 'Decrement':  
 decrementCount++;  
 userInput--;  
 updateResultText('decrement', decrementCount);  
 resultText +='Decremented to $userInput\n';  
 break;  
 case 'Square':  
 squareCount++;  
 updateResultText('square', squareCount);  
 resultText +='Square of $userInput is ${findSquare(userInput)}\n';  
 break;  
 case 'Prime':  
 primeCount++;  
 updateResultText('prime', primeCount);  
 resultText +=  
 '$userInput is ${isPrime(userInput) ? 'prime' : 'not prime'}\n';  
 break;  
 case 'Even/Odd':  
 evenOddCount++;  
 updateResultText('even/odd', evenOddCount);  
 resultText +=  
 '$userInput is ${isEven(userInput) ? 'even' : 'odd'}\n';  
 break;  
 case 'Palindrome':  
 palindromeCount++;  
 updateResultText('palindrome', palindromeCount);  
 resultText +=  
 '$userInput is ${isPalindrome(userInput) ? 'palindrome' : 'not palindrome'}\n';  
 break;  
 case 'Factorial':  
 factorialCount++;  
 updateResultText('factorial', factorialCount);  
 resultText +=  
 'Factorial of $userInput is ${factorial(userInput)}\n';  
 break;  
 case 'Reset':  
 incrementCount = 0;  
 decrementCount = 0;  
 squareCount = 0;  
 primeCount = 0;  
 evenOddCount = 0;  
 palindromeCount = 0;  
 factorialCount = 0;  
 resultText = '';  
 // Reset userInput to its original value  
 userInput =0;  
 break;  
 }  
 });  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: 'Buttons\_Task\_App',  
 debugShowCheckedModeBanner: false,  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text('ButtonApp'),  
 ),  
 body: SingleChildScrollView(  
 padding: EdgeInsets.all(20),  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: <Widget>[  
 // SizedBox(height: 20),  
 // Text(  
 // 'Enter a number:',  
 // style: TextStyle(fontSize: 18),  
 // ),  
 // SizedBox(height: 10),  
 Container(  
 // width: 200, // Set a specific width  
 // child: TextField(  
 // keyboardType: TextInputType.number,  
 // onChanged: (value) {  
 // userInput = int.tryParse(value) ?? 0;  
 // },  
 child: Text(userInput.toString())  
 ),  
 SizedBox(height: 20),  
 Text(  
 resultText,  
 style: TextStyle(fontSize: 16),  
 ),  
 SizedBox(height: 20),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
 children: [  
 buildButton('Increment'),  
 buildButton('Decrement'),  
 buildButton('Square'),  
 ],  
 ),  
 SizedBox(height: 10),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
 children: [  
 buildButton('Prime'),  
 buildButton('Even/Odd'),  
 buildButton('Palindrome'),  
 ],  
 ),  
 SizedBox(height: 10),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 buildButton('Factorial'),  
 SizedBox(width: 10),  
 buildButton('Reset'),  
 ],  
 ),  
 ],  
 ),  
 ),  
 ),  
 );  
 }  
  
 Widget buildButton(String action) {  
 return ElevatedButton(  
 onPressed: () {  
 onButtonPress(action);  
 },  
 style: ButtonStyle(  
 backgroundColor: MaterialStateProperty.*all*<Color>(Colors.*blue*),  
 shape: MaterialStateProperty.*all*<OutlinedBorder>(  
 RoundedRectangleBorder(  
 borderRadius: BorderRadius.circular(30.0),  
 ),  
 ),  
 ),  
 child: Text(action),  
 );  
 }  
}

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |