5. Develop Calculator Application

```
import 'package:flutter/material.dart';

void main() {
  runApp(CalculatorApp());
}

class CalculatorApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData.dark(),
      home: Calculator(),
      debugShowCheckedModeBanner: false,
    );
  }
}

class Calculator extends StatefulWidget {
  @override
  _CalculatorState createState() => _CalculatorState();
}

class _CalculatorState extends State<Calculator> {
  String _output = '0';
  String _currentInput = '';
  double _num1 = 0;
  double _num2 = 0;
  String _operator = '';

  void _handleButtonClick(String value) {
    setState(() {
      if (value == 'C') {
        _output = '0';
        _currentInput = '';
        _num1 = 0;
        _num2 = 0;
        _operator = '';
      } else if (value == '+' || value == '-' || value == '×' || value == '÷') {
        _num1 = double.parse(_currentInput);
        _operator = value;
        _currentInput = '';
    } else if (value == '=') {
      _num2 = double.parse(_currentInput);
      if (_operator == '+') {
        _output = (_num1 + _num2).toString();
      } else if (_operator == '-') {
        _output = (_num1 - _num2).toString();
      } else if (_operator == '×') {
        _output = (_num1 * _num2).toString();
      } else if (_operator == '÷') {
        _output = _num2 != 0 ? (_num1 / _num2).toString() : 'Error';
      }
      _currentInput = _output;
      _operator = '';
    } else {
      _currentInput += value;
      _output = _currentInput;
    }
  });
}

Widget _buildButton(String value) {
  return Expanded(
    child: ElevatedButton(
      onPressed: () => _handleButtonClick(value),
      style: ElevatedButton.styleFrom(
        padding: EdgeInsets.all(20),
        shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(8)),
      ),
      child: Text(
        value,
        style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
      ),
    ),
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text('Calculator')),
    body: Column(
      children: [
        Expanded(
          child: Container(
              alignment: Alignment.bottomRight,
              padding: EdgeInsets.all(16),
              child: Text(
                _output,
                style: TextStyle(fontSize: 48, fontWeight: FontWeight.bold),
              ),
            ),
          ),
          Row(children: [_buildButton('7'), _buildButton('8'), _buildButton('9'), _buildButton('÷')]),
          Row(children: [_buildButton('4'), _buildButton('5'), _buildButton('6'), _buildButton('×')]),
          Row(children: [_buildButton('1'), _buildButton('2'), _buildButton('3'), _buildButton('-')]),
          Row(children: [_buildButton('C'), _buildButton('0'), _buildButton('='), _buildButton('+')]),
        ],
      ),
    );
  }
}
```

6.
Develop an application to Check the Weather in Countries Across the world (Weather app)

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

void main() {
  runApp(WeatherApp());
}

class WeatherApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData.dark(),
      home: WeatherScreen(),
      debugShowCheckedModeBanner: false,
    );
  }
}
```

```
class WeatherScreen extends StatefulWidget {
  @override
  _WeatherScreenState createState() => _WeatherScreenState();
}

class _WeatherScreenState extends State<WeatherScreen> {
  final TextEditingController _cityController = TextEditingController();
  String _weather = '';
  String _temperature = '';
  String _description = '';
  String _icon = '';

  Future<void> _fetchWeather(String city) async {
    final apiKey = 'YOUR_OPENWEATHERMAP_API_KEY'; // Replace with your API key
    final url = Uri.parse(
        'https://api.openweathermap.org/data/2.5/weather?q=$city&appid=$apiKey&units=metric');

    try {
      final response = await http.get(url);
      if (response.statusCode == 200) {
        final data = json.decode(response.body);
        setState(() {
          _weather = data['weather'][0]['main'];
          _description = data['weather'][0]['description'];
          _temperature = '${data['main']['temp']} °C';
          _icon =
              'https://openweathermap.org/img/wn/${data['weather'][0]['icon']}@2x.png';
        });
      } else {
        setState(() {
          _weather = 'City not found';
          _description = '';
          _temperature = '';
          _icon = '';
        });
      }
    } catch (e) {
      setState(() {
        _weather = 'Error fetching weather';
        _description = '';
        _temperature = '';
        _icon = '';
      });
    }
```

```
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text('Weather App')),
    body: Padding(
      padding: EdgeInsets.all(16),
      child: Column(
        children: [
          TextField(
            controller: _cityController,
            decoration: InputDecoration(
              labelText: 'Enter City',
              border: OutlineInputBorder(),
            ),
          ),
          SizedBox(height: 16),
          ElevatedButton(
            onPressed: () {
              _fetchWeather(_cityController.text);
            },
            child: Text('Get Weather'),
          ),
          SizedBox(height: 32),
          if (_weather.isNotEmpty) ...[
            Image.network(_icon),
            Text(
              _weather,
              style: TextStyle(fontSize: 32, fontWeight: FontWeight.bold),
            ),
            Text(
              _description,
              style: TextStyle(fontSize: 20),
            ),
            Text(
              _temperature,
              style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
            ),
          ],
        ],
      ),
    ),
  );
  }
}
```

Add Dependencies

Open pubspec.yaml and add the following dependencies:

yaml

Copy code

```
dependencies:
  flutter:
    sdk: flutter
  http: ^1.2.0
```

Get an OpenWeatherMap API Key

Go to https://openweathermap.org

Sign up and create an API key

7. Develop a “Stopwatch” application using Flutter

```
import 'dart:async';

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';

void main() {
runApp(const MyApp());
}

class MyApp extends StatelessWidget {
const MyApp({super.key});

// This widget is the root of your application.
@override
```

```
Widget build(BuildContext context) {
        return MaterialApp(
        title: 'Stopwatch',
        theme: ThemeData(
                // This is the theme of your application.
                //
                // TRY THIS: Try running your application with "flutter run". You'll see
                // the application has a blue toolbar. Then, without quitting the app,
                // try changing the seedColor in the colorScheme below to Colors.green
                // and then invoke "hot reload" (save your changes or press the "hot
                // reload" button in a Flutter-supported IDE, or press "r" if you used
                // the command line to start the app).
                //
                // Notice that the counter didn't reset back to zero; the application
                // state is not lost during the reload. To reset the state, use hot
                // restart instead.
                //
                // This works for code too, not just values: Most code changes can be
                // tested with just a hot reload.
                colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
                useMaterial3: true,
        ),
        home: const MyHomePage(),
        );
}
}

class MyHomePage extends StatefulWidget {
const MyHomePage({super.key});

@override
State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
late Stopwatch stopwatch;
late Timer t;

void handleStartStop() {
        if(stopwatch.isRunning) {
        stopwatch.stop();
        }
        else {
        stopwatch.start();
```

```
    }
}

String returnFormattedText() {
    var milli = stopwatch.elapsed.inMilliseconds;

    String milliseconds = (milli % 1000).toString().padLeft(3, "0"); // this one for the
miliseconds
    String seconds = ((milli ~/ 1000) % 60).toString().padLeft(2, "0"); // this is for the second
    String minutes = ((milli ~/ 1000) ~/ 60).toString().padLeft(2, "0"); // this is for the minute

    return "$minutes:$seconds:$milliseconds";
}

@override
void initState() {
    super.initState();
    stopwatch = Stopwatch();

    t = Timer.periodic(Duration(milliseconds: 30), (timer) {
    setState(() {});
    });
}

@override
Widget build(BuildContext context) {
    return Scaffold(
    body: SafeArea(
        child: Center(
        child: Column( // this is the column
            mainAxisAlignment: MainAxisAlignment.center,
            children: [

            CupertinoButton(
                onPressed: () {
                handleStartStop();
                },
                padding: EdgeInsets.all(0),
                child: Container(
                height: 250,
                alignment: Alignment.center,
                decoration: BoxDecoration(
                    shape: BoxShape.circle, // this one is use for make the
circle on ui.
```

```
                    border: Border.all(
                    color: Color(0xff0395eb),
                    width: 4,
                    ),
                ),
                child: Text(returnFormattedText(), style: TextStyle(
                    color: Colors.black,
                    fontSize: 40,
                    fontWeight: FontWeight.bold,
                ),),
                ),
            ),

            SizedBox(height: 15,),

            CupertinoButton(       // this the cupertino button and here we perform all
the reset button function
                onPressed: () {
                stopwatch.reset();
                },
                padding: EdgeInsets.all(0),
                child: Text("Reset", style: TextStyle(
                color: Colors.red,
                fontWeight: FontWeight.bold,
                ),),
            ),

            ],
        ),
        ),
    ),
    );
}
}
```