**Objective:-** To Study the basics of Dart language and design a basic Flutter App to Create Unit convertor(for length, weight, temperature, Area) app using Flutter.

**Code :-**

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: 'Flutter Demo',

      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 purple 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: ConverterScreen(),

    );

  }

}

class ConverterScreen extends StatefulWidget {

  @override

  \_ConverterScreenState createState() => \_ConverterScreenState();

}

class \_ConverterScreenState extends State<ConverterScreen> {

*// Conversion options with units (from -> to)*

  final List<String> \_conversionTypes = [

    'Meters to Kilometers',

    'Kilometers to Meters',

    'Grams to Kilograms',

    'Kilograms to Grams',

    'Celsius to Fahrenheit',

    'Fahrenheit to Celsius',

    'Square Meters to Hectares',

    'Hectares to Square Meters',

*// New conversions*

    'Miles to Kilometers',

    'Kilometers to Miles',

    'Pounds to Kilograms',

    'Kilograms to Pounds',

    'Liters to Milliliters',

    'Milliliters to Liters',

    'Kilometers per hour to Miles per hour',

    'Miles per hour to Kilometers per hour'

  ];

*// Mapping for input and output units*

  final Map<String, String> \_inputUnits = {

    'Meters to Kilometers': 'meters',

    'Kilometers to Meters': 'kilometers',

    'Grams to Kilograms': 'grams',

    'Kilograms to Grams': 'kilograms',

    'Celsius to Fahrenheit': 'Celsius',

    'Fahrenheit to Celsius': 'Fahrenheit',

    'Square Meters to Hectares': 'square meters',

    'Hectares to Square Meters': 'hectares',

*// New conversions*

    'Miles to Kilometers': 'miles',

    'Kilometers to Miles': 'kilometers',

    'Pounds to Kilograms': 'pounds',

    'Kilograms to Pounds': 'kilograms',

    'Liters to Milliliters': 'liters',

    'Milliliters to Liters': 'milliliters',

    'Kilometers per hour to Miles per hour': 'km/h',

    'Miles per hour to Kilometers per hour': 'mph',

  };

  final Map<String, String> \_outputUnits = {

    'Meters to Kilometers': 'kilometers',

    'Kilometers to Meters': 'meters',

    'Grams to Kilograms': 'kilograms',

    'Kilograms to Grams': 'grams',

    'Celsius to Fahrenheit': 'Fahrenheit',

    'Fahrenheit to Celsius': 'Celsius',

    'Square Meters to Hectares': 'hectares',

    'Hectares to Square Meters': 'square meters',

    'Miles to Kilometers': 'kilometers',

    'Kilometers to Miles': 'miles',

    'Pounds to Kilograms': 'kilograms',

    'Kilograms to Pounds': 'pounds',

    'Liters to Milliliters': 'milliliters',

    'Milliliters to Liters': 'liters',

    'Kilometers per hour to Miles per hour': 'mph',

    'Miles per hour to Kilometers per hour': 'km/h',

  };

*// Input controller*

  final TextEditingController \_inputController = TextEditingController();

*// Selected conversion type and result*

  String \_selectedConversion = 'Meters to Kilometers';

  double \_result = 0.0;

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text(

          'Unit Converter',

          style: TextStyle(

            color: Colors.white,

            fontWeight: FontWeight.bold,

          ),

        ),

        backgroundColor: Colors.deepPurple,

      ),

      body: Padding(

        padding: const EdgeInsets.all(16.0),

        child: Column(

          children: <Widget>[

*// Dropdown for selecting conversion type*

            DropdownButton<String>(

              value: \_selectedConversion,

              onChanged: (String? newValue) {

                setState(() {

                  \_selectedConversion = newValue!;

                });

              },

              items: \_conversionTypes.map((String value) {

                return DropdownMenuItem<String>(

                  value: value,

                  child: Text(value),

                );

              }).toList(),

            ),

            SizedBox(height: 20),

*// Input field with dynamic unit label*

            Row(

              children: [

                Expanded(

                  child: TextField(

                    controller: \_inputController,

                    keyboardType: TextInputType.number,

                    decoration: InputDecoration(

                      labelText: 'Enter value',

                    ),

                  ),

                ),

                SizedBox(width: 10),

                Text(

                  \_inputUnits[

                      \_selectedConversion]!, *// Dynamic unit next to input field*

                  style: TextStyle(fontSize: 16),

                ),

              ],

            ),

            SizedBox(height: 20),

*// Convert button*

            ElevatedButton(

              onPressed: \_convert,

              child: Text('Convert'),

            ),

            SizedBox(height: 20),

*// Display result with unit*

            Text(

              'Result: $*\_result* ${*\_outputUnits[\_selectedConversion]*}',

              style: TextStyle(fontSize: 20),

            ),

          ],

        ),

      ),

    );

  }

*// Conversion logic based on selected type*

  void \_convert() {

    double inputValue = double.tryParse(\_inputController.text) ?? 0.0;

    switch (\_selectedConversion) {

      case 'Meters to Kilometers':

        \_result = \_convertMetersToKilometers(inputValue);

        break;

      case 'Kilometers to Meters':

        \_result = \_convertKilometersToMeters(inputValue);

        break;

      case 'Grams to Kilograms':

        \_result = \_convertGramsToKilograms(inputValue);

        break;

      case 'Kilograms to Grams':

        \_result = \_convertKilogramsToGrams(inputValue);

        break;

      case 'Celsius to Fahrenheit':

        \_result = \_convertCelsiusToFahrenheit(inputValue);

        break;

      case 'Fahrenheit to Celsius':

        \_result = \_convertFahrenheitToCelsius(inputValue);

        break;

      case 'Square Meters to Hectares':

        \_result = \_convertSquareMetersToHectares(inputValue);

        break;

      case 'Hectares to Square Meters':

        \_result = \_convertHectaresToSquareMeters(inputValue);

        break;

*// New conversions*

      case 'Miles to Kilometers':

        \_result = \_convertMilesToKilometers(inputValue);

        break;

      case 'Kilometers to Miles':

        \_result = \_convertKilometersToMiles(inputValue);

        break;

      case 'Pounds to Kilograms':

        \_result = \_convertPoundsToKilograms(inputValue);

        break;

      case 'Kilograms to Pounds':

        \_result = \_convertKilogramsToPounds(inputValue);

        break;

      case 'Liters to Milliliters':

        \_result = \_convertLitersToMilliliters(inputValue);

        break;

      case 'Milliliters to Liters':

        \_result = \_convertMillilitersToLiters(inputValue);

        break;

      case 'Kilometers per hour to Miles per hour':

        \_result = \_convertKilometersToMilesPerHour(inputValue);

        break;

      case 'Miles per hour to Kilometers per hour':

        \_result = \_convertMilesToKilometersPerHour(inputValue);

        break;

      default:

        \_result = 0.0;

    }

    setState(() {});

  }

*// Conversion functions for each type*

*// Length conversions*

  double \_convertMetersToKilometers(double value) {

    return value / 1000; *// 1 meter = 0.001 kilometers*

  }

  double \_convertKilometersToMeters(double value) {

    return value \* 1000; *// 1 kilometer = 1000 meters*

  }

*// Weight conversions*

  double \_convertGramsToKilograms(double value) {

    return value / 1000; *// 1 gram = 0.001 kilograms*

  }

  double \_convertKilogramsToGrams(double value) {

    return value \* 1000; *// 1 kilogram = 1000 grams*

  }

*// Temperature conversions*

  double \_convertCelsiusToFahrenheit(double value) {

    return (value \* 9 / 5) + 32; *// (C \* 9/5) + 32 = F*

  }

  double \_convertFahrenheitToCelsius(double value) {

    return (value - 32) \* 5 / 9; *// (F - 32) \* 5/9 = C*

  }

*// Area conversions*

  double \_convertSquareMetersToHectares(double value) {

    return value / 10000; *// 1 square meter = 0.0001 hectares*

  }

  double \_convertHectaresToSquareMeters(double value) {

    return value \* 10000; *// 1 hectare = 10000 square meters*

  }

*// New conversion functions*

*// Length: Miles and Kilometers*

  double \_convertMilesToKilometers(double value) {

    return value \* 1.60934; *// 1 mile = 1.60934 kilometers*

  }

  double \_convertKilometersToMiles(double value) {

    return value / 1.60934; *// 1 kilometer = 0.621371 miles*

  }

*// Weight: Pounds and Kilograms*

  double \_convertPoundsToKilograms(double value) {

    return value \* 0.453592; *// 1 pound = 0.453592 kilograms*

  }

  double \_convertKilogramsToPounds(double value) {

    return value / 0.453592; *// 1 kilogram = 2.20462 pounds*

  }

*// Volume: Liters and Milliliters*

  double \_convertLitersToMilliliters(double value) {

    return value \* 1000; *// 1 liter = 1000 milliliters*

  }

  double \_convertMillilitersToLiters(double value) {

    return value / 1000; *// 1 milliliter = 0.001 liters*

  }

*// Speed: Kilometers per hour and Miles per hour*

  double \_convertKilometersToMilesPerHour(double value) {

    return value \* 0.621371; *// 1 km/h = 0.621371 mph*

  }

  double \_convertMilesToKilometersPerHour(double value) {

    return value / 0.621371; *// 1 mph = 1.60934 km/h*

  }

}

**Output:-**

