**Objective:-** To create an interactive Form using the form widget, Create a Loan Calculator(EMI based on simple as well as compound.

**Code :-**

import 'package:flutter/material.dart';

import 'dart:math'; *// Import math package to use pow() for compound interest calculation*

void main() {

  runApp(const MyApp());

}

*// Main class for the app*

class MyApp extends StatelessWidget {

  const MyApp({*super*.key});

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      theme: ThemeData(

        brightness: Brightness.light,

        primaryColor: Colors.blue, *// Light theme with blue as primary color*

      ),

      darkTheme: ThemeData(

        brightness: Brightness.dark,

        scaffoldBackgroundColor:

            const Color(0xFF121212), *// Dark theme background color*

        appBarTheme: const AppBarTheme(

          backgroundColor: Color(0xFF1F1F1F),

          iconTheme: IconThemeData(color: Colors.white),

          titleTextStyle: TextStyle(color: Colors.white, fontSize: 20),

        ),

        textButtonTheme: TextButtonThemeData(

          style: TextButton.styleFrom(

            foregroundColor: Colors.white,

            backgroundColor: Colors.blue,

          ),

        ),

        inputDecorationTheme: InputDecorationTheme(

          filled: true,

          fillColor: Colors.grey[800],

          border: OutlineInputBorder(

            borderRadius: BorderRadius.circular(8.0),

            borderSide: BorderSide.none,

          ),

          hintStyle: const TextStyle(color: Colors.grey),

        ),

        textTheme: const TextTheme(

          bodyMedium: TextStyle(color: Colors.white),

          titleLarge: TextStyle(color: Colors.white),

        ),

      ),

      themeMode: ThemeMode.dark, *// Set dark mode by default*

      home: const EMICalculator(), *// Set the EMI calculator screen as the home*

      debugShowCheckedModeBanner: false, *// Hide the debug banner*

    );

  }

}

*// Stateful widget for EMI Calculator*

class EMICalculator extends StatefulWidget {

  const EMICalculator({*super*.key});

  @override

  State<EMICalculator> createState() => \_EMICalculatorState();

}

class \_EMICalculatorState extends State<EMICalculator> {

  int currentindex = 0; *// 0 for Simple Interest, 1 for Compound Interest*

  String result = ""; *// To hold the calculated EMI result*

  double principal = 0.0; *// Principal amount*

  double rate = 0.0; *// Annual interest rate*

  double tenure = 0.0; *// Loan tenure in years*

  double totalPayment = 0.0; *// Total payment calculated*

  double totalInterest = 0.0; *// Total interest calculated*

*// Controllers for TextField inputs*

  TextEditingController principalController = TextEditingController();

  TextEditingController rateController = TextEditingController();

  TextEditingController tenureController = TextEditingController();

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: const Text("EMI Calculator"), *// AppBar title*

        backgroundColor: Theme.of(context)

            .appBarTheme

            .backgroundColor, *// AppBar color based on theme*

        actions: [

          IconButton(

            onPressed: () {},

            icon:

                const Icon(Icons.settings), *// Settings icon (currently unused)*

          ),

        ],

      ),

      body: SingleChildScrollView(

*// Enables scrolling for the screen*

        child: Padding(

          padding: const EdgeInsets.all(12.0), *// Screen padding*

          child: Column(

            mainAxisAlignment: MainAxisAlignment.start,

            crossAxisAlignment: CrossAxisAlignment.start,

            children: [

              Row(

                children: [

*// Radio buttons for selecting Simple/Compound interest*

                  radioButton("Simple Interest", Colors.blue, 0),

                  radioButton("Compound Interest", Colors.pink, 1),

                ],

              ),

              const SizedBox(height: 20.0),

              const Text(

                "Loan Amount (Principal) :- ",

                style: TextStyle(fontSize: 18.0),

              ),

              const SizedBox(height: 8.0),

*// TextField for entering principal amount*

              TextField(

                controller: principalController,

                keyboardType: TextInputType.number,

                textAlign: TextAlign.center,

                decoration: InputDecoration(

                  hintText: "Enter Principal Amount",

                  filled: true,

                  fillColor: Theme.of(context)

                      .inputDecorationTheme

                      .fillColor, *// TextField styling based on theme*

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(8.0),

                    borderSide: BorderSide.none,

                  ),

                ),

              ),

              const SizedBox(height: 20.0),

              const Text(

                "Annual Interest Rate (%) :- ",

                style: TextStyle(fontSize: 18.0),

              ),

              const SizedBox(height: 8.0),

*// TextField for entering interest rate*

              TextField(

                controller: rateController,

                keyboardType: TextInputType.number,

                textAlign: TextAlign.center,

                decoration: InputDecoration(

                  hintText: "Enter Interest Rate",

                  filled: true,

                  fillColor: Theme.of(context).inputDecorationTheme.fillColor,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(8.0),

                    borderSide: BorderSide.none,

                  ),

                ),

              ),

              const SizedBox(height: 20.0),

              const Text(

                "Loan Tenure (in Years) :- ",

                style: TextStyle(fontSize: 18.0),

              ),

              const SizedBox(height: 8.0),

*// TextField for entering tenure in years*

              TextField(

                controller: tenureController,

                keyboardType: TextInputType.number,

                textAlign: TextAlign.center,

                decoration: InputDecoration(

                  hintText: "Enter Loan Tenure",

                  filled: true,

                  fillColor: Theme.of(context).inputDecorationTheme.fillColor,

                  border: OutlineInputBorder(

                    borderRadius: BorderRadius.circular(8.0),

                    borderSide: BorderSide.none,

                  ),

                ),

              ),

              const SizedBox(height: 20.0),

              SizedBox(

                width: double.infinity, *// Button fills full width*

                height: 50.0, *// Button height*

                child: TextButton(

                  onPressed: () {

*// Parsing values from TextField inputs*

                    setState(() {

                      principal =

                          double.tryParse(principalController.text) ?? 0.0;

                      rate = double.tryParse(rateController.text) ?? 0.0;

                      tenure = double.tryParse(tenureController.text) ?? 0.0;

                    });

*// Call calculation method based on selected interest type*

                    if (currentindex == 0) {

                      calculateEMIWithSimpleInterest(principal, rate, tenure);

                    } else {

                      calculateEMIWithCompoundInterest(principal, rate, tenure);

                    }

                  },

                  child: const Text(

                    "Calculate EMI",

                    style: TextStyle(color: Colors.white),

                  ),

                ),

              ),

              const SizedBox(height: 20.0),

              const Center(

                child: Text(

                  "Your EMI is :- ",

                  textAlign: TextAlign.center,

                  style: TextStyle(

                    fontSize: 24.0,

                    fontWeight: FontWeight.bold,

                  ),

                ),

              ),

              const SizedBox(height: 50.0),

              Center(

                child: Text(

                  result, *// Display calculated EMI*

                  textAlign: TextAlign.center,

                  style: const TextStyle(

                    fontSize: 40.0,

                    fontWeight: FontWeight.bold,

                  ),

                ),

              ),

              const SizedBox(height: 20.0),

              const Center(

                child: Text(

                  "Total Payment Breakdown",

                  textAlign: TextAlign.center,

                  style: TextStyle(

                    fontSize: 20.0,

                    fontWeight: FontWeight.bold,

                  ),

                ),

              ),

              const SizedBox(height: 10.0),

*// Linear progress bar indicating total payment breakdown*

              Container(

                height: 20.0,

                decoration: BoxDecoration(

                  color: Colors.orange, *// Background color of the progress bar*

                  borderRadius: BorderRadius.circular(10.0),

                ),

                child: ClipRRect(

                  borderRadius: BorderRadius.circular(10.0),

                  child: LinearProgressIndicator(

                    value: (principal > 0 && totalPayment > 0)

                        ? principal / totalPayment

                        : 0.0,

                    backgroundColor: Colors.transparent,

                    valueColor:

                        const AlwaysStoppedAnimation<Color>(Colors.green),

                  ),

                ),

              ),

              const SizedBox(height: 10.0),

*// Displaying total principal and interest*

              Row(

                mainAxisAlignment: MainAxisAlignment.spaceBetween,

                children: [

                  Column(

                    crossAxisAlignment: CrossAxisAlignment.start,

                    children: [

                      Text(

                        "Total Principal",

                        style: TextStyle(fontSize: 16.0, color: Colors.green),

                      ),

                      Text(

                        "₹ ${*principal.toStringAsFixed(2)*}",

                        style: const TextStyle(

                          fontSize: 16.0,

                          fontWeight: FontWeight.bold,

                        ),

                      ),

                    ],

                  ),

                  Column(

                    crossAxisAlignment: CrossAxisAlignment.end,

                    children: [

                      Text(

                        "Total Interest",

                        style: TextStyle(fontSize: 16.0, color: Colors.orange),

                      ),

                      Text(

                        "₹ ${*totalInterest.toStringAsFixed(2)*}",

                        style: const TextStyle(

                          fontSize: 16.0,

                          fontWeight: FontWeight.bold,

                        ),

                      ),

                    ],

                  ),

                ],

              ),

            ],

          ),

        ),

      ),

    );

  }

*// Method for radio button widget*

  Widget radioButton(String value, Color color, int index) {

    return Expanded(

      child: Container(

        margin: const EdgeInsets.symmetric(horizontal: 12.0),

        height: 50.0,

        child: TextButton(

          onPressed: () {

            setState(() {

              currentindex = index; *// Update index when radio button is pressed*

            });

          },

          style: TextButton.styleFrom(

            backgroundColor: currentindex == index ? color : Colors.grey[200],

          ),

          child: Text(

            value,

            style: TextStyle(

              color: currentindex == index ? Colors.white : color,

            ),

          ),

        ),

      ),

    );

  }

*// Method to calculate EMI with Simple Interest*

  void calculateEMIWithSimpleInterest(

      double principal, double rate, double tenure) {

    setState(() {

      double interest =

          (principal \* rate \* tenure) / 100; *// Simple interest formula*

      totalPayment = principal + interest; *// Total payment*

      totalInterest = interest; *// Total interest*

      result = (totalPayment / (tenure \* 12))

          .toStringAsFixed(2); *// EMI calculation for months*

    });

  }

*// Method to calculate EMI with Compound Interest*

  void calculateEMIWithCompoundInterest(

      double principal, double rate, double tenure) {

    setState(() {

      double monthlyRate = rate / (12 \* 100); *// Monthly interest rate*

      int n = (tenure \* 12).toInt(); *// Number of months (n)*

      totalPayment =

          principal \* pow(1 + monthlyRate, n); *// Compound interest formula*

      totalInterest = totalPayment - principal; *// Total interest*

      result =

          (totalPayment / n).toStringAsFixed(2); *// EMI calculation for months*

    });

  }

}

**Output:-**



