**MINISTERUL EDUCAŢIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA  
UNIVERSITATEA TEHNICĂ A MOLDOVEI   
FACULATATEA CALCULATOAREA INFORMATICĂ ȘI MICROELECTRONICĂ  
Departamentul Informatică Și Ingineria Sistemelor**

**Raport**

**la Programarea aplicațiilor mobile**

**Laboratorul 1**

**Efectuat: Popușoi Cătălin IA-221**

**Verificat: Buza Dina**

**Chişinău – 2024**

**SARCINA**

Varianta 1:

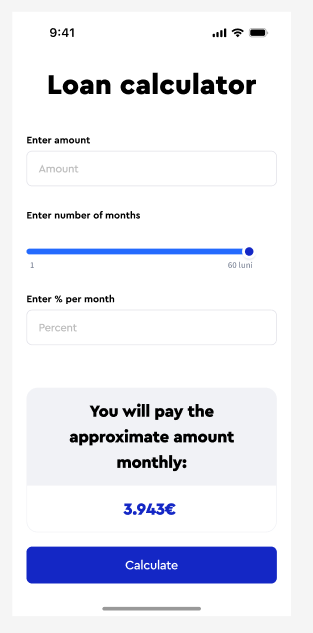


Figura 1: Sarcina pentru laboratorul 1

**Listingul programului**

import 'package:flutter/material.dart';

import 'dart:math';

void main() => runApp(const MaterialApp(

debugShowCheckedModeBanner: false,

home: LoanCalculator(),

));

class LoanCalculator extends StatefulWidget {

const LoanCalculator({super.key});

@override

State<LoanCalculator> createState() => \_LoanCalculatorState();

}

class \_LoanCalculatorState extends State<LoanCalculator> {

double \_sliderValue = 6;

final TextEditingController \_amountController = TextEditingController();

final TextEditingController \_percentageController = TextEditingController();

final TextEditingController \_monthsController = TextEditingController();

String \_monthlyPayment = '';

void \_calculateMonthlyPayment() {

final double? principal = double.tryParse(\_amountController.text);

final double? annualInterestRate =

double.tryParse(\_percentageController.text);

final int months = \_sliderValue.toInt();

if (principal != null &&

annualInterestRate != null &&

principal > 0 &&

annualInterestRate > 0) {

double monthlyInterestRate = (annualInterestRate / 100) / 12;

double monthlyPayment = (principal \*

monthlyInterestRate \*

pow(1 + monthlyInterestRate, months)) /

(pow(1 + monthlyInterestRate, months) - 1);

setState(() {

\_monthlyPayment =

'Approximate Monthly Payment: \$${monthlyPayment.toStringAsFixed(2)}';

});

} else {

setState(() {

\_monthlyPayment = 'Please enter valid values';

});

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('Loan Calculator'),

backgroundColor: const Color(0xFF2980B9),

elevation: 0,

),

body: Container(

decoration: const BoxDecoration(

gradient: LinearGradient(

colors: [Color(0xFF6DD5FA), Color(0xFF2980B9)],

begin: Alignment.topCenter,

end: Alignment.bottomCenter,

),

),

child: Padding(

padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 40),

child: SingleChildScrollView(

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: <Widget>[

\_buildInputField(

controller: \_amountController,

label: 'Enter Loan Amount',

icon: Icons.attach\_money,

),

const SizedBox(height: 20),

\_buildInputField(

controller: \_percentageController,

label: 'Enter Interest Rate (%)',

icon: Icons.percent,

),

const SizedBox(height: 30),

const Text(

"Choose number of months",

style: TextStyle(

fontSize: 16,

fontWeight: FontWeight.bold,

color: Colors.white,

),

),

const SizedBox(height: 10),

Row(

children: [

Expanded(

child: TextField(

controller: \_monthsController,

keyboardType: TextInputType.number,

decoration: InputDecoration(

labelText: "Enter Months",

filled: true,

fillColor: Colors.white,

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(10),

borderSide: BorderSide.none,

),

),

onChanged: (value) {

final int? newMonths = int.tryParse(value);

if (newMonths != null &&

newMonths >= 1 &&

newMonths <= 60) {

setState(() {

\_sliderValue = newMonths.toDouble();

});

}

},

),

),

const SizedBox(width: 10),

Text(

'${\_sliderValue.toInt()} months',

style: const TextStyle(

fontSize: 16,

color: Colors.white,

),

),

],

),

SliderTheme(

data: SliderTheme.of(context).copyWith(

activeTrackColor: Colors.white,

inactiveTrackColor: Colors.white38,

thumbColor: Colors.white,

overlayColor: Colors.white24,

thumbShape:

const RoundSliderThumbShape(enabledThumbRadius: 12),

overlayShape:

const RoundSliderOverlayShape(overlayRadius: 24),

),

child: Slider(

value: \_sliderValue,

min: 1,

max: 60,

divisions: 59,

label: '${\_sliderValue.toInt()} months',

onChanged: (value) {

setState(() {

\_sliderValue = value;

\_monthsController.text =

\_sliderValue.toInt().toString();

});

},

),

),

const SizedBox(height: 40),

Center(

child: ElevatedButton(

onPressed: \_calculateMonthlyPayment,

style: ElevatedButton.styleFrom(

padding: const EdgeInsets.symmetric(

horizontal: 40, vertical: 15),

backgroundColor: Colors.white,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(30),

),

),

child: const Text(

'Calculate Payment',

style: TextStyle(

fontSize: 18,

color: Color(0xFF2980B9),

fontWeight: FontWeight.bold,

),

),

),

),

const SizedBox(height: 30),

Center(

child: Text(

\_monthlyPayment,

style: const TextStyle(

fontSize: 20,

fontWeight: FontWeight.bold,

color: Colors.white,

),

),

),

],

),

),

),

),

);

}

Widget \_buildInputField(

{required TextEditingController controller,

required String label,

required IconData icon}) {

return Container(

decoration: BoxDecoration(

color: Colors.white,

borderRadius: BorderRadius.circular(15),

boxShadow: const [

BoxShadow(

color: Colors.black26,

blurRadius: 10,

offset: Offset(0, 5),

)

],

),

child: TextField(

controller: controller,

keyboardType: TextInputType.number,

decoration: InputDecoration(

labelText: label,

prefixIcon: Icon(icon, color: const Color(0xFF2980B9)),

filled: true,

fillColor: Colors.white,

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(15),

borderSide: BorderSide.none,

),

),

),

);

}

}

Clasa principală (LoanCalculator):

* Folosește un StatefulWidget pentru a actualiza valorile dinamice atunci când utilizatorul interacționează cu câmpurile de text și slider-ul.
* Se creează un ecran stilizat care include input-uri pentru sumă, rata dobânzii și numărul de luni pentru împrumut.

Input-uri pentru utilizator:

* Câmpuri de text (TextFields) pentru introducerea sumei împrumutului și a ratei dobânzii, fiecare stilizat cu pictograme (dolari și procente).
* Slider și un câmp de text pentru a seta numărul de luni. Slider-ul permite utilizatorului să aleagă o perioadă între 1 și 60 de luni.

Logica de calcul:

* Funcția \_calculateMonthlyPayment realizează calculul plății lunare folosind formula pentru plata unei ipoteci. Folosește rata dobânzii anuale convertită în rată lunară și numărul total de luni.
* Dacă valorile introduse de utilizator sunt valide, se calculează și afișează valoarea plății lunare.
* În cazul în care valorile sunt invalide, este afișat un mesaj de eroare („Please enter valid values”).

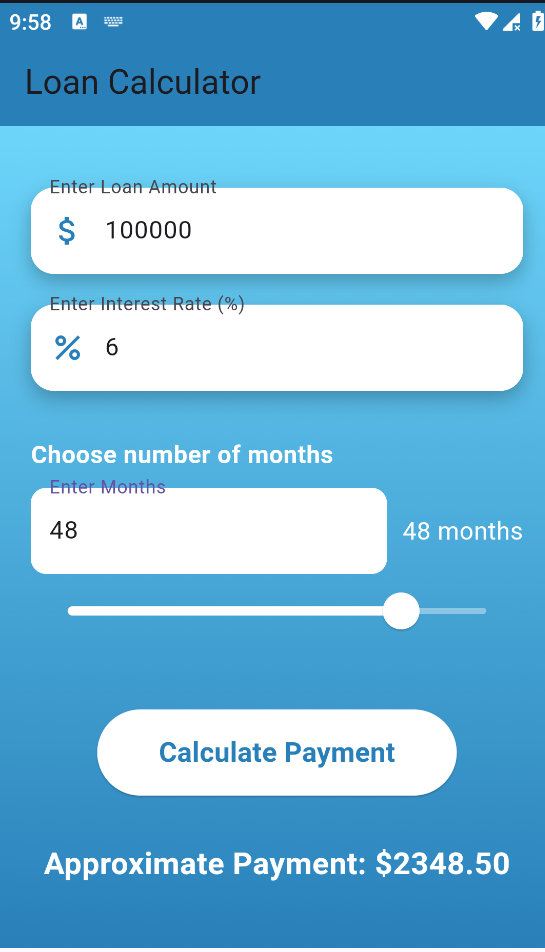
Design:

* Interfața folosește un gradient albastru pentru fundal și include un AppBar cu titlul „Loan Calculator”.
* Butonul „Calculate Payment” este stilizat cu margini rotunjite și este poziționat central.
* Rezultatul este afișat într-un text alb centrat, cu mărime de font mai mare și îngroșat pentru claritate.

Personalizarea slider-ului:

* Slider-ul este personalizat cu culori și un thumb (cerc mic) rotunjit pentru a face selecția numărului de luni mai interactivă și estetică.

Rezultatul:



**Figura 2:** Rezultatul calculelor

**CONCLUZII**

Acest program implementează un **calculator de împrumut** simplu în Flutter, care calculează valoarea aproximativă a ratei lunare pe baza sumei împrumutului, a ratei dobânzii anuale și a perioadei de rambursare în luni. Utilizatorul introduce datele relevante într-o interfață, cu elemente de interacțiune precum câmpuri text și un slider pentru numărul de luni. Programul utilizează o formulă pentru calcularea plății lunare și afișează rezultatul pe ecran într-un format clar.