INTERNATIONAL BURCH UNIVERSITY

FACULTY OF ENGINEERING, NATURAL AND MEDICAL SCIENCES

DEPARTMENT OF INFORMATION TECHNOLOGY

Personal Finance Calculator: Build a calculator for financial calculations, such as loan payments, interest rates, and savings projections.

IBU 008 Programming I

Selma Karasoftić

Sarajevo, 2024

1. Introduction

Personal Finance Calculator came up out of idea to help the people calculate some of the important calculations on an easy way, aiming to help those who doesn't understand finance spectrum very well.

The main objective of this project is to allow users to finish calculations without having to understand the whole process behind the calculation. User enters just the number needed for the calculation and the program does everything else. After it, the user can decide to store the data he calculated and open it later in the future if needed. The program calculates three most important things: Loan payment, total interest paid and projection for savings based on entered values that need to be provided.

The purpose of this report is to explain some main functionalities of the program for the reader to understand the code easier and eventually, at the end, to show visually the output of the calculator.

Final impact that will be made is simplifying the complexness of calculations that need to be made. That way the process will be easy and available to all the users not limiting the usage of the calculator based on the knowledge.

Roadmap:

2. Program Functionalities

1. **Loan Payment Calculation**
   - Calculates the monthly payment for a loan based on the loan amount, interest rate and the time entered in years, this will help the user to trach their monthly expenses

2. **Interest Rate Calculation**
   - Calculates the total interest that the user paid on the loan depending on the information needed for the calculation.

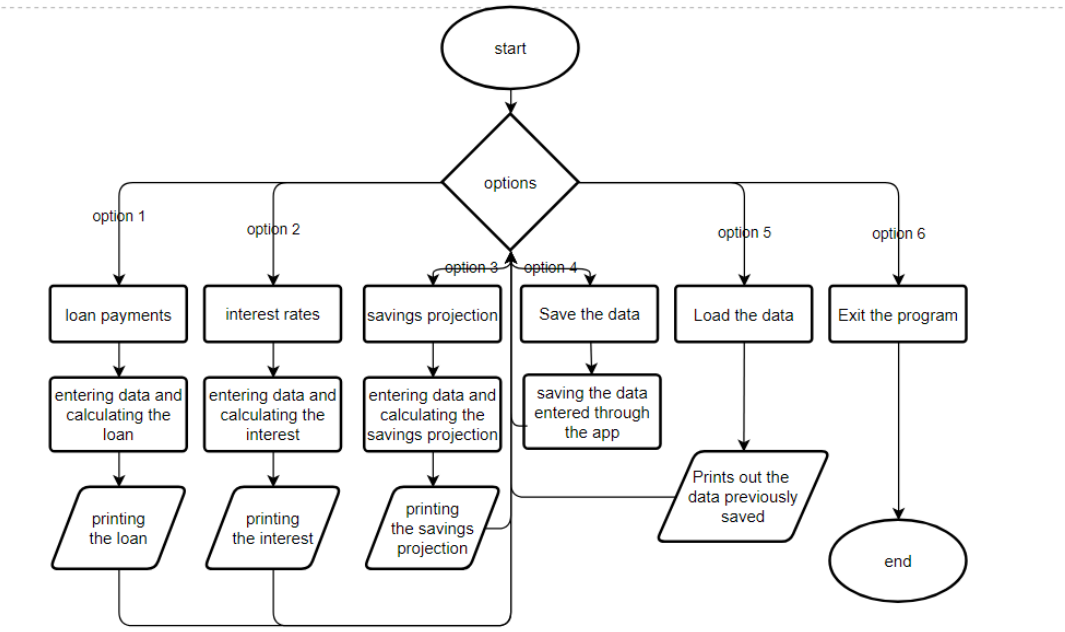
3. **Savings projection**
   - Calculates the future savings amount based on the information needed for proceeding with this request. This helps user to set the realistic goals for the future.

4. **Data saving and loading**
   - Allows users to save calculated financial data and load it later if needed.

5. **Choosing an option**
   - Easily access every feature over the menu just by entering the number of the option you want

6. **Handling of the possible errors**
   - This is important part of the project because in order for program to perform the calculations on the right way, the user needs to enter valid information. For example, of course that the program should not work if the letters are entered instead of numbers or if we try to save the data without having all the information needed to save it.

3. Flowcharts



The program flow consists of the following activities:

- Loan payment calculation
- The application takes input information and based on them calculates and print the loan
- Interest payment calculation
- Similar to the previous step, the application takes necessary input from the user and in dependence to the loan, provides the user with the end result.
- Savings projection
- In accordance to the calculated loan and interest payment as well as entering additional parameters, the application calculates the goal which the user should meet in the terms of budget savings.
- Data saving and loading
- Option for the user to save data in a form of a file as well as the option to load the file and proceed with further operations.
- Program exiting
- Program will stop after the option 6 is chosen which will provide the user with the goodbye message.

4. Code

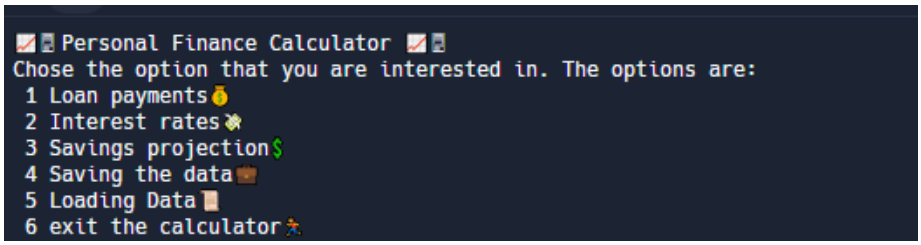
```
from datetime import datetime
#function to calculate monthly payment based on loan amount, interest rate and
time
#using the formula for calculation, the function returns calculated monthly
payment


def calculateMonthlyPayment(loanAmmount, interestRate, time):
  interestRateOneMonth = interestRate / 12 / 100
  timeMonths = time * 12
  monthlyPayment = (loanAmmount * interestRateOneMonth) / (
      1 - (1 + interestRateOneMonth)**-timeMonths)
  return monthlyPayment


#function that calculates total interest based on the loan amount, monthly
payment, and time
#the loop in the dunction calculates interest for each month and updates the
remaining loan amount
#when the loop is completed the
def calculateInterest(loanAmount, monthlyPayment, time):
  totalPayments = monthlyPayment * time * 12  # Total payments made over the loan
duration
  totalInterest = totalPayments - loanAmount  # Total interest paid
  return totalInterest


#class that will represent a savings instance
class savings:

  #initializing the values over the constructor
  def __init__(self, initialAmmount, monthlyPayment, interest, time):
    self.initialAmmount = initialAmmount
    self.monthlyPayment = monthlyPayment
    self.interest = interest
    self.time = time

  #function to calculate the savings using the formula
  def calculateSavingsProjection(self):
    monthlyInterest = self.interest / 100 / 12
    months = self.time * 12
    savings = (self.initialAmmount * (1 + monthlyInterest)**months +
               self.monthlyPayment *
```

```
                ((1 + monthlyInterest)**months - 1) / monthlyInterest)

    return savings


#saving the data to a file
def save_data(data):
  current_time = datetime.today().isoformat()
  with open('C:\\Users\\selma\\Desktop\\history.txt', 'a') as file:
    file.write(current_time)
    file.write("\n")
    for key, value in data.items():
      file.write(f"{key}: {value}\n")
  print("Data saved successfully!")


#the start of the program and listing the options
print("📈💻 Personal Finance Calculator 📈💻")
while True:
  print(
      "Chose the option that you are interested in. The options are: \n 1 Loan
payments💰 \n 2 Interest rates💸 \n 3 Savings projection💲 "
  )
  print(" 4 Saving the data💼 \n 5 Loading Data📜 \n 6 exit the calculator🚴\n")

  userPreference = input("Enter the number (1-6): ")

  if userPreference == '1':
    print(
        "\nYou have chosen Loan Payments💰. You should enter the information
needed!"
    )

    #handling the possible error, for ecample if user enters the letter, it will
print out the except
    try:
      #collecting important information before calling the function
      print("\nEnter three numbers.")
      print("\nloan ammount: ")
      loanAmmount = float(input())
      print("\nInterest rate in percentage: ")
      interestRate = float(input())
      print("\nYears for completing the loan: ")
      time = float(input())
```

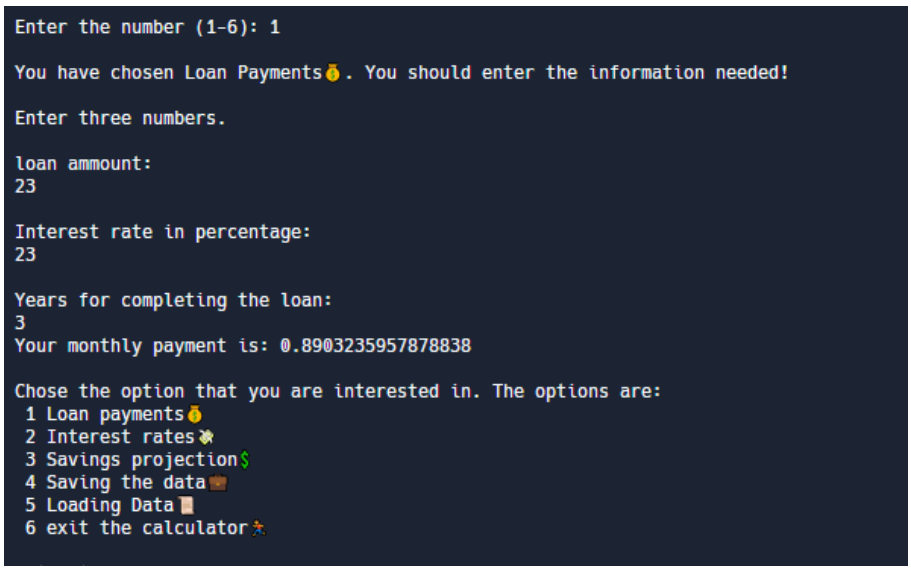
```
      #calling the function to calculate the payment
      monthlyPayment = calculateMonthlyPayment(loanAmmount, interestRate, time)
      print("Your monthly payment is:", monthlyPayment, "\n")
    except ValueError:
      print("Invalid number entered!\n")

#the next part functions almost the same as the previous, only the next one is
calculating
#the interest instead of monthly payment
  elif userPreference == '2':
    print(
        "You have chosen Interest Rates. You should enter the information
needed!"
    )
    try:
      print("\nEnter three numbers.")
      print("\nloan ammount: ")
      loanAmmount = float(input())
      print("\nMonthly payment: ")
      monthlyPayment = float(input())
      print("\nYears for completing the loan: ")
      time = float(input())
      interest1 = calculateInterest(loanAmmount, monthlyPayment, time)
      print("The paid interest is:", interest1, "\n")
    except ValueError:
      print("Invalid number entered!\n")

#the next part also functions almost the same as the first part, only the next
one is calculating
#savings amount instead of monthly payment
  elif userPreference == '3':
    print(
        "You have chosen Savings Projection. You should enter the information
needed!"
    )
    try:
      print("\nEnter four numbers.")
      print("\nInital savings amount: ")
      initialAmmount = float(input())
      print("\nMonthly payment: ")
      monthlyPayment = float(input())
      print("\nInterest: ")
      interest = float(input())
      print("\nYears for completing the loan: ")
      time = float(input())
```

```
      savings_instance = savings(initialAmmount, monthlyPayment, interest,
                                 time)
      savingsProjection = savings_instance.calculateSavingsProjection()
      print("Savings would be: ", savingsProjection, "\n")
    except ValueError:
      print("Invalid number entered!\n")

#saving the data to a file, but only if every calculation is made until this
option
  elif userPreference == '4':
    print("You have chosen Save Data💼.")
    try:
      data = {
          'loan_payment': loanAmmount,
          'total_interest': interest1,
          'savings': savingsProjection
      }
      save_data(data)
    except NameError:
      print("No data calculated yet!\n")

#the saved data will be read only if it exists, if not the exception will be
shown
  elif userPreference == '5':
    print("You have chosen Load Data.📜")
    try:
      print("History: ")

      with open('C:\\Users\\selma\\Desktop\\history.txt', 'r') as file:
        history = file.read()  # Read the entire file content
        # Check if the file has content
        print(history)  # Print the entire content
    except FileNotFoundError:
      print("No saved data found.")

  elif userPreference == '6':
    print("Thank You for using the calculator! Have a nice day.👌")
    break

  elif userPreference != '1' or userPreference != '2' or userPreference != '3' or
userPreference != '4' or userPreference != '5' or userPreference != '6':
    print("Invalid option entered! Try again.\n")
```

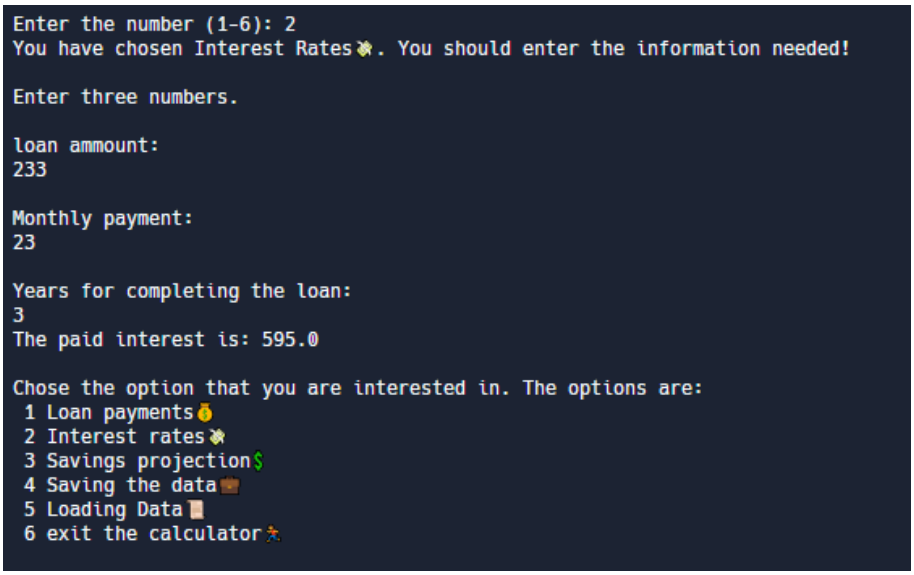
5. Results



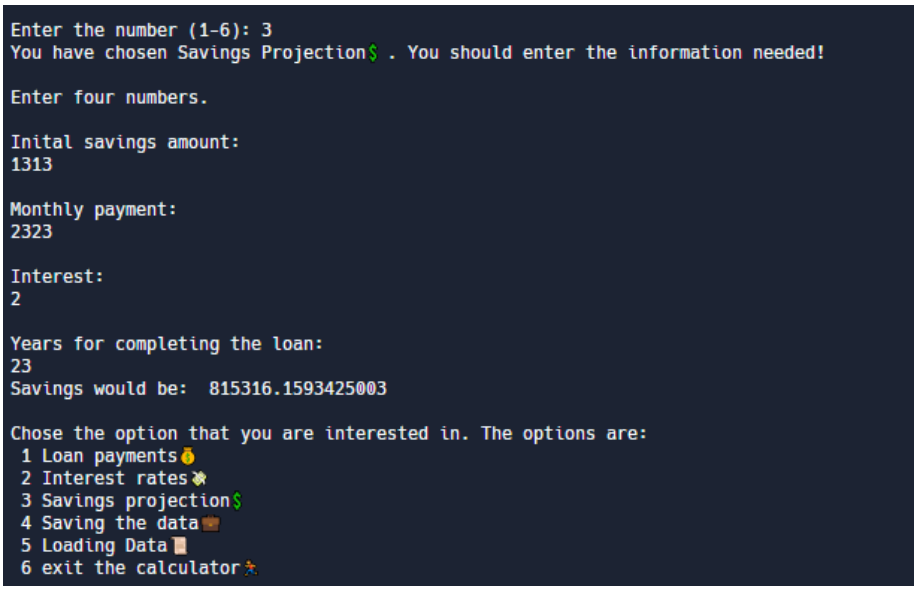
When entering the app, the options will be listed and the user will be let to choose which option is by their preferences.



After the user chooses option 1, he will be asked to enter a few numbers and after entering the requested information, the calculation of the loan will be executed and the result would be printed. Then the program is listing the options again for the user to choose the option again



After the user chooses option 2, he will be asked to enter a few numbers and after entering the requested information, the calculation of the interest will be executed and the result would be printed. Then the program is listing the options again for the user to choose the option again



After the user chooses option 3, he will be asked to enter a few numbers and after entering the requested information, the calculation of the savings projection will be executed and the result would be printed. Then the program is listing the options again for the user to choose the option again.

```
Enter the number (1-6): 4
You have chosen Save Data💼.
Data saved successfully!
Chose the option that you are interested in. The options are:
 1 Loan payments💰
 2 Interest rates📈
 3 Savings projection$
 4 Saving the data💼
 5 Loading Data📜
 6 exit the calculator🏃
```

After choosing the option 4 the user is saving the data, but only if all the previous data is entered. If not, the program will throw an error stating that there is no data due to the lack of it.

```
Enter the number (1-6): 5
You have chosen Load Data.📜
History:
loan_payment: 120.0
total_interest: 1068.0
savings: 23202.576894757636
loan_payment: 1223.0
total_interest: -935.0
savings: 2713.136542043838
2024-01-06T23:32:40.821810
loan_payment: 233.0
total_interest: 595.0
savings: 815316.1593425003

Chose the option that you are interested in. The options are:
 1 Loan payments💰
 2 Interest rates📈
 3 Savings projection$
 4 Saving the data💼
 5 Loading Data📜
 6 exit the calculator🏃
```

Option 5 prints out all the calculations done, also printing the date and time on which the data was entered.

```
Enter the number (1-6): 6
Thank You for using the calculator! Have a nice day.👋
```

Finally, the option 6 exits the program but it firstly prints out the goodbye message.