MAHATMA EDUCATION SOCIETY'S

PILLAI COLLEGE OF ARTS, COMMERCE & SCIENCE
(Autonomous)

NEW PANVEL

PROJECT REPORT ON

" MAINTENANCE CALORIE CALCULATOR "

IN PARTIAL FULFILMENT OF

BACHELOR OF INFORMATION TECHNOLOGY

SEMESTER IV – 2023-24

PROJECT GUIDE

Name : Prof. Anju Somani Mam

SUBMITTED BY: Jatin Dnyaneshwar Gangare

ROLL NO: 6130

Description :

This Java project presents a Maintenance Calories Calculator, distinguished by its Graphical User Interface (GUI) developed using Swing components. The GUI facilitates user interaction by offering intuitive input fields for crucial personal details like gender, age, weight, and height, as well as providing radio buttons for selecting activity levels. This user-centric design enhances accessibility, ensuring a seamless experience for individuals seeking to manage their caloric intake effectively.

The core functionality of the calculator lies in its ability to leverage the Harris-Benedict equation to compute the Basal Metabolic Rate (BMR). This foundational metric serves as the cornerstone for determining maintenance calories, allowing users to obtain personalized estimates tailored to their physiological characteristics and activity levels. By integrating sophisticated calculations into a user-friendly interface, the calculator empowers individuals to make informed decisions regarding their dietary needs and fitness goals.

Furthermore, the calculator offers a comprehensive approach to goal-setting, prompting users to select their desired objective among maintaining weight, gaining weight, or losing weight. Based on the chosen goal, the program dynamically adjusts the calculated maintenance calories, providing users with actionable insights to support their journey towards achieving optimal health and fitness outcomes. This adaptive feature underscores the calculator's versatility, catering to a diverse range of user preferences and objectives while promoting a holistic approach to nutritional management and well-being.

CODE :

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

class MaintenanceCaloriesCalculatorGUICSV {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            JFrame frame = new JFrame("Maintenance Calories Calculator");
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            JPanel panel = new JPanel();
            panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));

            JLabel genderLabel = new JLabel("M/F: ");
            JLabel ageLabel = new JLabel("Age: ");
            JLabel weightLabel = new JLabel("Weight (kg): ");
            JLabel heightLabel = new JLabel("Height (cm): ");

            JTextField genderField = new JTextField();
            JTextField ageField = new JTextField();
            JTextField weightField = new JTextField();
            JTextField heightField = new JTextField();

            // Create radio buttons for activity level
            JRadioButton sedentaryButton = new JRadioButton("Sedentary");
            JRadioButton lightlyActiveButton = new JRadioButton("Lightly Active");
            JRadioButton moderatelyActiveButton = new JRadioButton("Moderately Active");
            JRadioButton veryActiveButton = new JRadioButton("Very Active");

            // Create a button group to ensure only one radio button is selected at a time
            ButtonGroup activityLevelGroup = new ButtonGroup();
            activityLevelGroup.add(sedentaryButton);
            activityLevelGroup.add(lightlyActiveButton);
            activityLevelGroup.add(moderatelyActiveButton);
            activityLevelGroup.add(veryActiveButton);

            JButton calculateButton = new JButton("Calculate");
            JLabel resultLabel = new JLabel("Maintenance Calories: ");

            calculateButton.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) {
                    try {
                        char gender = genderField.getText().toUpperCase().charAt(0);
                        int age = Integer.parseInt(ageField.getText());
```

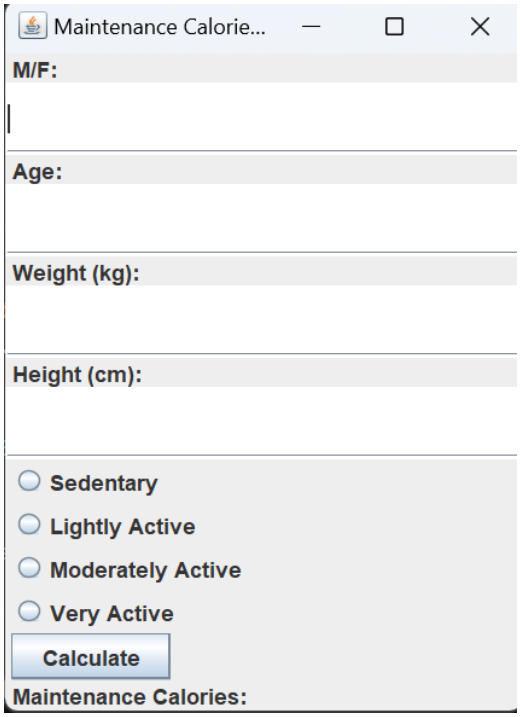
```
                double weight = Double.parseDouble(weightField.getText());
                double height = Double.parseDouble(heightField.getText());

                // Get the selected radio button for activity level
                int activityLevel = 0;
                if (sedentaryButton.isSelected()) {
                    activityLevel = 1;
                } else if (lightlyActiveButton.isSelected()) {
                    activityLevel = 2;
                } else if (moderatelyActiveButton.isSelected()) {
                    activityLevel = 3;
                } else if (veryActiveButton.isSelected()) {
                    activityLevel = 4;
                }

                double bmr = calculateBMR(gender, age, weight, height);
                double maintenanceCalories = calculateMaintenanceCalories(bmr,
activityLevel);

                String goal = askUserForGoal();
                double adjustedCalories = adjustCaloriesForGoal(maintenanceCalories, goal);

                resultLabel.setText("Calories per day for " + goal + ": " +
Math.round(adjustedCalories));

                // Write data to CSV file
                try (PrintWriter writer = new PrintWriter(new FileWriter("calories_data.csv",
true))) {
                    writer.println(gender + "," + age + "," + weight + "," + height + "," +
activityLevel + "," + goal + "," + adjustedCalories);
                } catch (IOException ex) {
                    resultLabel.setText("Failed to write data to CSV file.");
                }
            } catch (NumberFormatException ex) {
                resultLabel.setText("Invalid input. Please enter numeric values.");
            } catch (IllegalArgumentException ex) {
                resultLabel.setText(ex.getMessage());
            }
        }
    });

    panel.add(genderLabel);
    panel.add(genderField);
    panel.add(ageLabel);
    panel.add(ageField);
    panel.add(weightLabel);
    panel.add(weightField);
    panel.add(heightLabel);
    panel.add(heightField);
    panel.add(sedentaryButton);
```

```
        panel.add(lightlyActiveButton);
        panel.add(moderatelyActiveButton);
        panel.add(veryActiveButton);
        panel.add(calculateButton);
        panel.add(resultLabel);

        frame.getContentPane().add(panel);
        frame.setSize(300, 400);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    });
  }

  private static double calculateBMR(char gender, int age, double weight, double height) {
    if (gender == 'M' || gender == 'F') {
      if (gender == 'M') {
        return 88.362 + (13.397 * weight) + (4.799 * height) - (5.677 * age);
      } else {
        return 447.593 + (9.247 * weight) + (3.098 * height) - (4.330 * age);
      }
    } else {
      throw new IllegalArgumentException("Invalid gender. Use 'M' or 'F'.");
    }
  }

  private static double calculateMaintenanceCalories(double bmr, int activityLevel) {
    switch (activityLevel) {
      case 1:
        return bmr * 1.2;
      case 2:
        return bmr * 1.375;
      case 3:
        return bmr * 1.55;
      case 4:
        return bmr * 1.725;
      default:
        throw new IllegalArgumentException("Invalid activity level.");
    }
  }

  private static String askUserForGoal() {
    String[] options = {"Maintain Weight", "Gain Weight", "Lose Weight"};
    int choice = JOptionPane.showOptionDialog(null, "Select your goal", "Choose Goal",
JOptionPane.DEFAULT_OPTION,
        JOptionPane.QUESTION_MESSAGE, null, options, options[0]);

    switch (choice) {
      case 0:
        return "Maintain Weight";
      case 1:
```

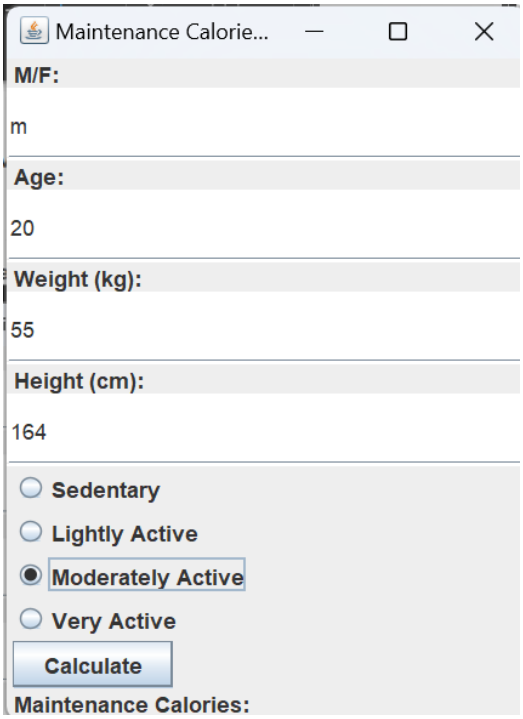
```
            return "Gain Weight";
        case 2:
            return "Lose Weight";
        default:
            return "Maintain Weight";
    }
  }

  private static double adjustCaloriesForGoal(double maintenanceCalories, String goal) {
    switch (goal) {
        case "Maintain Weight":
            return maintenanceCalories;
        case "Gain Weight":
            // You can adjust this factor based on the desired rate of weight gain
            return maintenanceCalories + 500;
        case "Lose Weight":
            // You can adjust this factor based on the desired rate of weight loss
            return maintenanceCalories - 500;
        default:
            return maintenanceCalories;
    }
  }
}
```

OUTPUT :

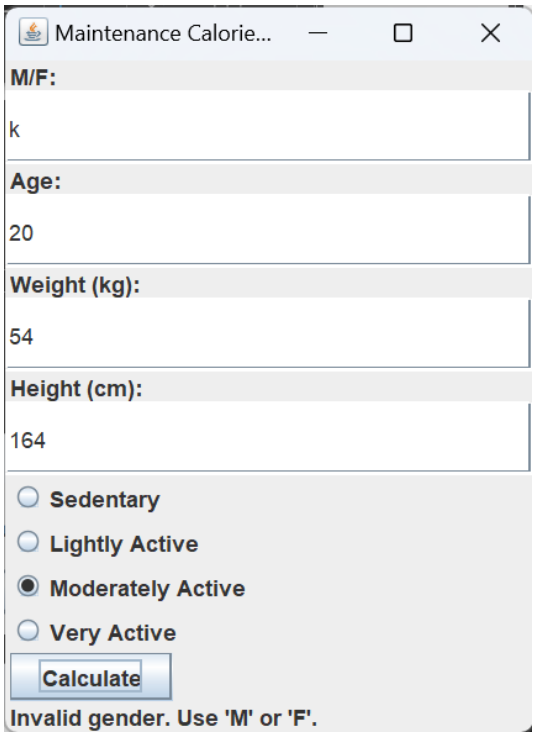
Enter The Values :

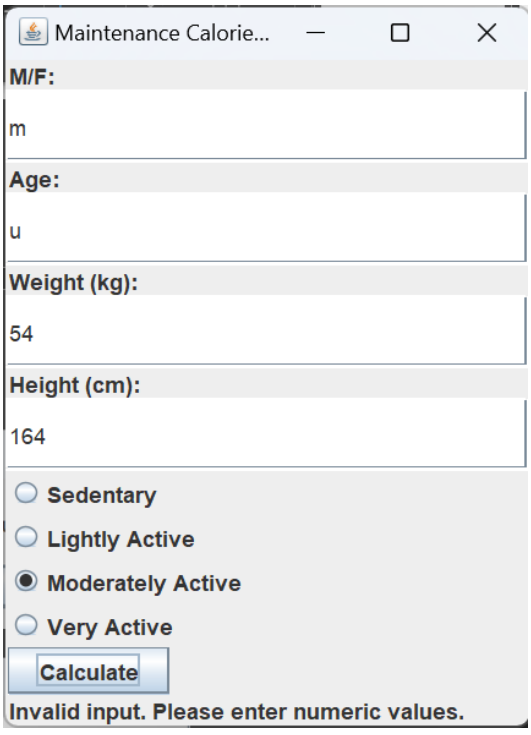


This is what looks after entering values

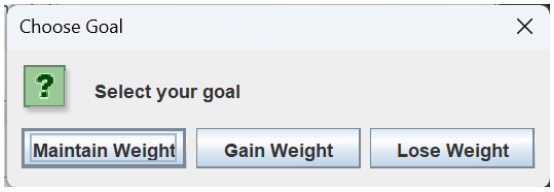


Exception Handling :

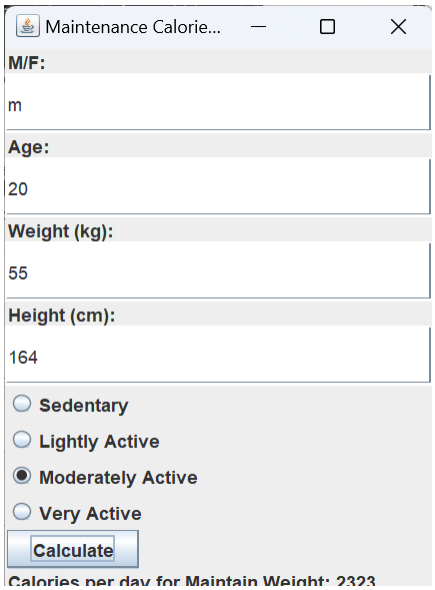


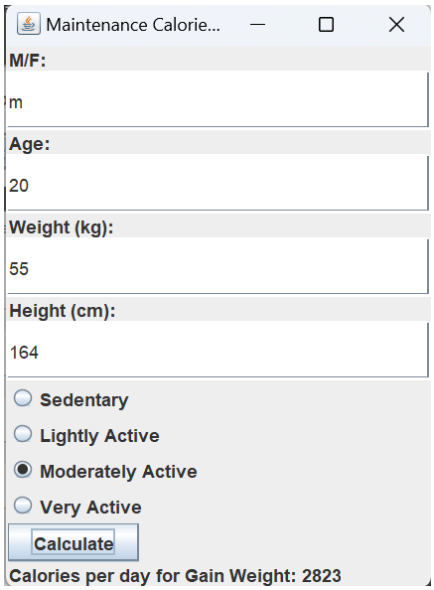


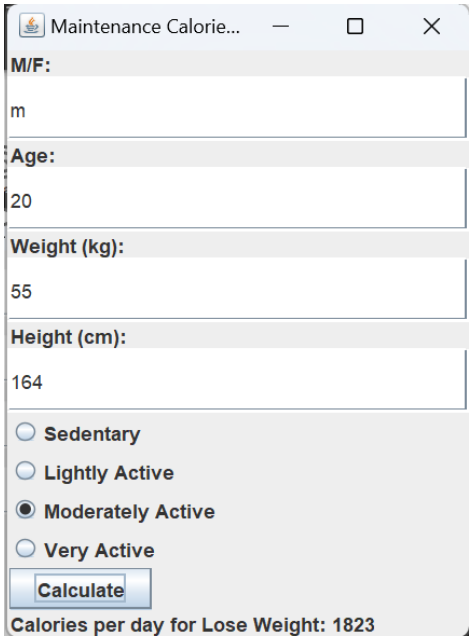
Here you have to select your goal



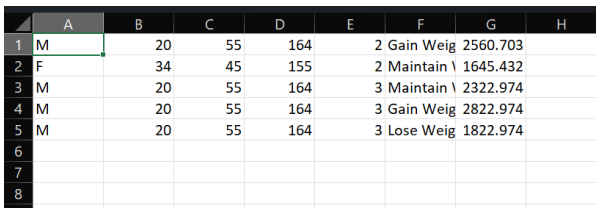
Three different output on specific selection :







Saves entered data into csv file



	A	B	C	D	E	F	G	H
1	M	20	55	164	2	Gain Weig	2560.703	
2	F	34	45	155	2	Maintain \	1645.432	
3	M	20	55	164	3	Maintain \	2322.974	
4	M	20	55	164	3	Gain Weig	2822.974	
5	M	20	55	164	3	Lose Weig	1822.974	
6								
7								
8								

CONCLUSION :

This Core Java project provides a practical solution for calculating maintenance calories through a user-friendly GUI. Leveraging Swing components, the program allows users to input their personal details and activity levels, enabling accurate estimations of daily caloric needs. With functionalities for goal selection and dynamic adjustments based on user objectives, it offers a versatile tool for individuals striving to manage their weight and nutrition effectively. Additionally, the integration of CSV file writing enhances its utility by enabling data persistence for future reference. Overall, this project demonstrates effective utilization of Java programming principles to address a practical health and fitness need with simplicity and efficiency.