TITLE :- BMI Calculator

STUDENT NAME :- ANJALI KUMARI

REGISTRATION NUMBER :-25BAI10631

COURSE :-BUILD YOUR OWN PROJECT

INSTITUTE :- VELLORE INSTITUTE OF TECHNOLOGY ,BHOPAL

YEAR :-2025

1. Introduction

Body Mass Index (BMI) is a widely used measure to assess whether a person has a healthy body weight relative to their height. It helps categorize individuals into groups such as Underweight, Normal, Overweight, and Obese.

This project focuses on creating a simple and user-friendly BMI Calculator using Python Tkinter GUI.

The application enables the user to input height and weight, calculate BMI instantly, and determine the health category using the standard WHO guidelines.

2. Problem Statement

Users often struggle to quickly determine whether their current weight is healthy based on their height. Manually calculating BMI can be time-consuming.

This project provides an efficient digital solution that automates BMI calculation and categorization.

3. Objectives

ü Build an interactive Python Tkinter GUI application.

ü Accept valid height and weight inputs.

ü Calculate BMI using the standard formula.

ü Provide BMI category classification for better health awareness.

ü Create an easy-to-use, offline desktop health tool.

4. Scope

- Works on any desktop system with Python installed.
- Provides instant BMI calculation.
- No internet, database, or backend required.
- Lightweight and easy for beginners to use.

5. BMI Formula

BMI=weight/height

Height is converted from centimeters to meters within the program.

6. Tools & Technologies Used

1. Python (Core Programming)
2. Tkinter (GUI framework)
3. VS Code (Code editor)
4. Git (Version control)
5. GitHub (Hosting repository)

7. System Architecture

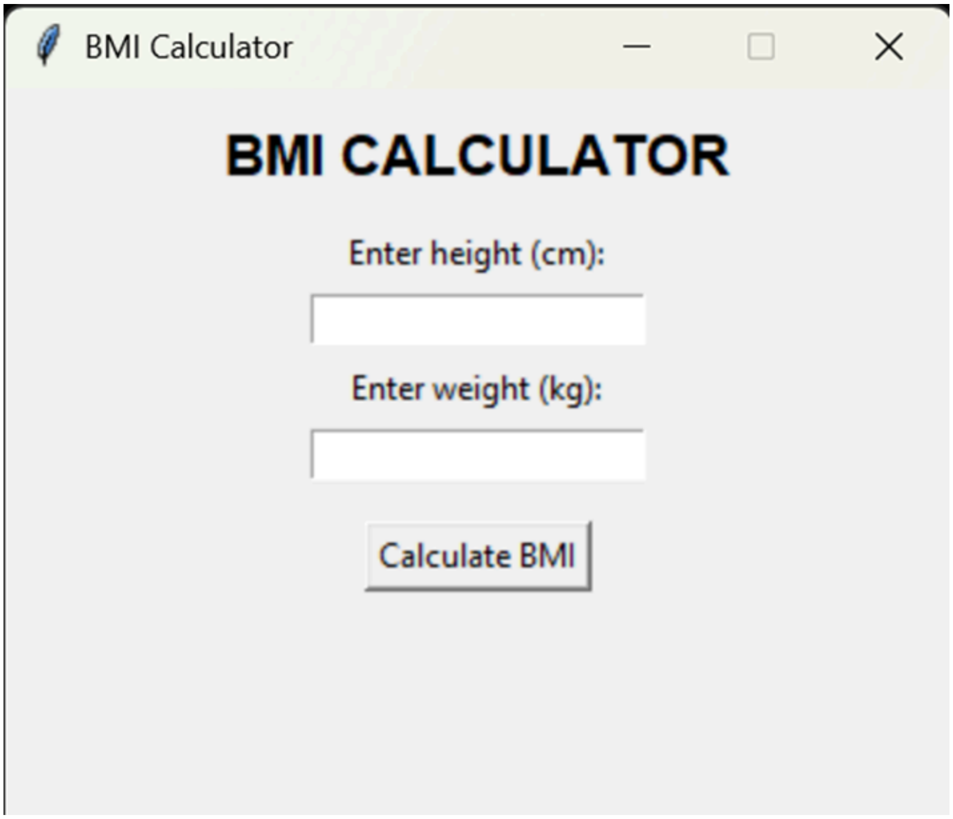
```
  +----------------------+
  |  User Inputs Data      |
  | (Height & Weight)     |
  +----------+-----------+
             |
             v
  +----------------------+
  |   BMI Calculation      |
  +----------+-----------+
             |
             v
  +----------------------+
  | Category Evaluation  |
  +----------+-----------+
             |
          v
+----------------------+
| Display BMI & Status |
+----------------------+
```

9. Flowchart

```
  +------------------+
  |  Start Program |
  +-------+--------+
          |
          v
+------------------------+
| Enter Height & Weight  |
+-----------+------------+
            |
            v
+--------------------+
| Convert Height          |
| (cm → meters)          |
+---------+----------+
          |
          v
+-----------------------+
| Calculate BMI Formula |
+----------+------------+
            |
            v
+-------------------------+
| Determine BMI Category  |
+----------+--------------+
            |
            v
+---------------------------+
| Display Result to User         |
+-------------+-------------+
            |
            v
       +----------+
       |   End  |
```

□

8.. screenshot

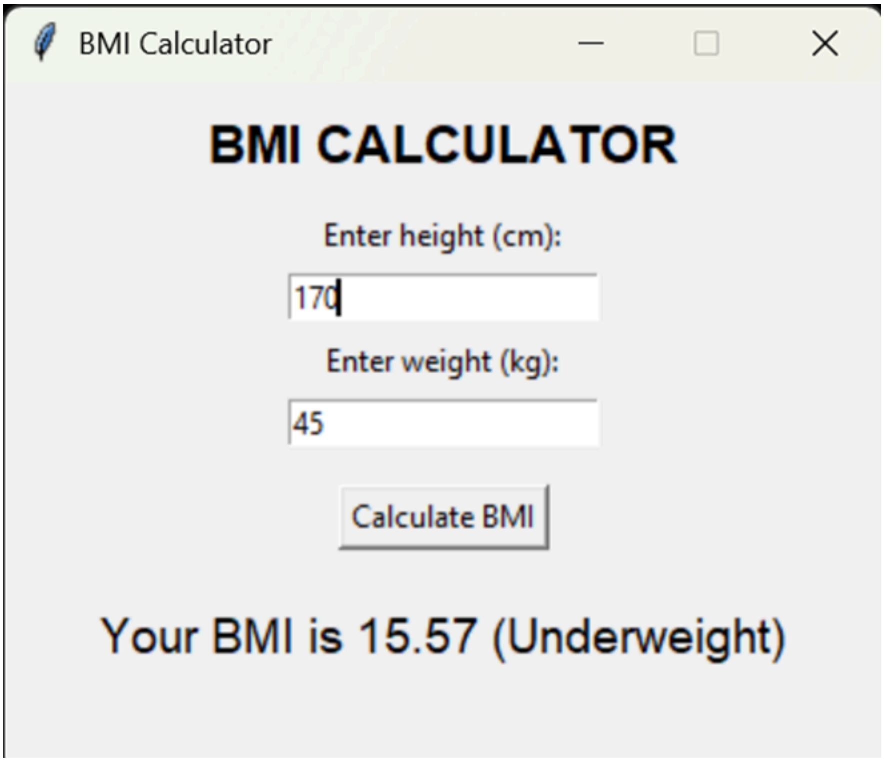


BMI Calculator

BMI CALCULATOR

Enter height (cm):

Enter weight (kg):



BMI CALCULATOR

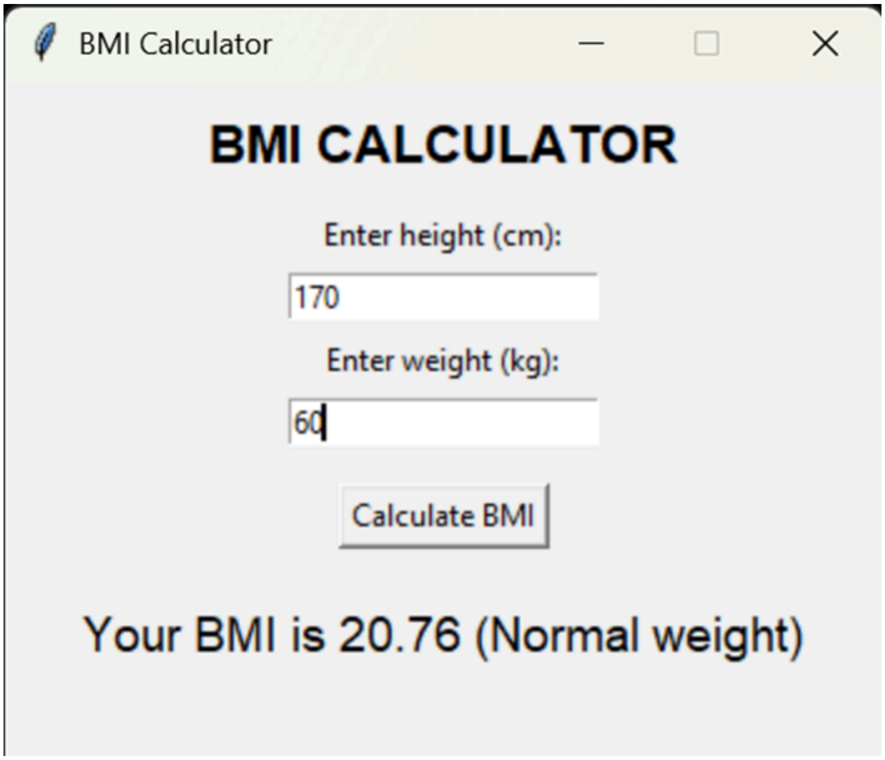
Enter height (cm):

170

Enter weight (kg):

45

Your BMI is 15.57 (Underweight)

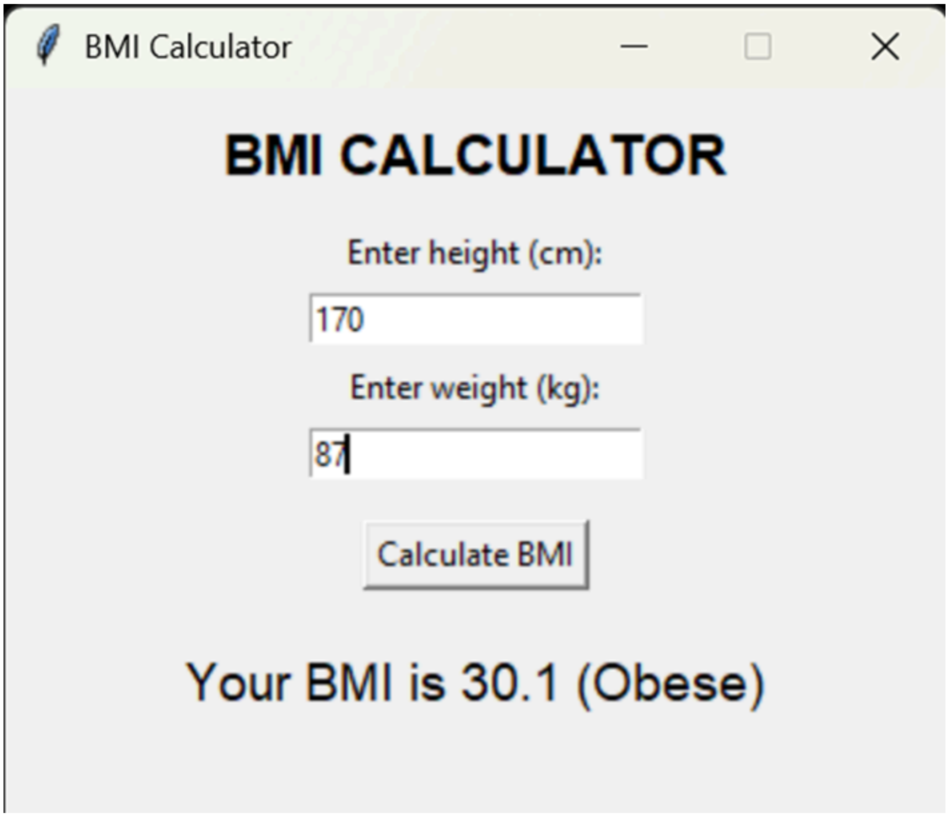


BMI CALCULATOR

Enter height (cm): 170

Enter weight (kg): 60

Your BMI is 20.76 (Normal weight)



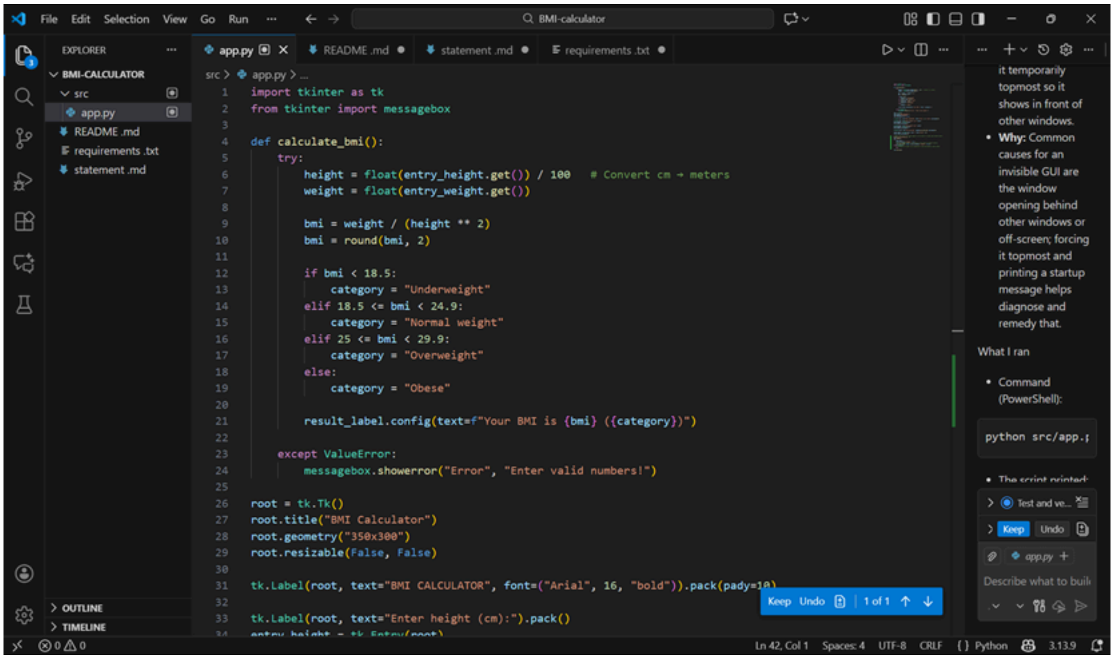
BMI CALCULATOR

Enter height (cm): 170

Enter weight (kg): 87

Your BMI is 30.1 (Obese)

9.. Source Code



```python
import tkinter as tk
from tkinter import messagebox

def calculate_bmi():
    try:
        height = float(entry_height.get()) / 100   # Convert cm → meters
        weight = float(entry_weight.get())

        bmi = weight / (height ** 2)
        bmi = round(bmi, 2)

        if bmi < 18.5:
            category = "Underweight"
        elif 18.5 <= bmi < 24.9:
            category = "Normal weight"
        elif 25 <= bmi < 29.9:
            category = "Overweight"
        else:
            category = "Obese"

        result_label.config(text=f"Your BMI is {bmi} ({category})")

    except ValueError:
        messagebox.showerror("Error", "Enter valid numbers!")

root = tk.Tk()
root.title("BMI Calculator")
root.geometry("350x300")
root.resizable(False, False)

tk.Label(root, text="BMI CALCULATOR", font=("Arial", 16, "bold")).pack(pady=10)

tk.Label(root, text="Enter height (cm):").pack()
```

it temporarily topmost so it shows in front of other windows.

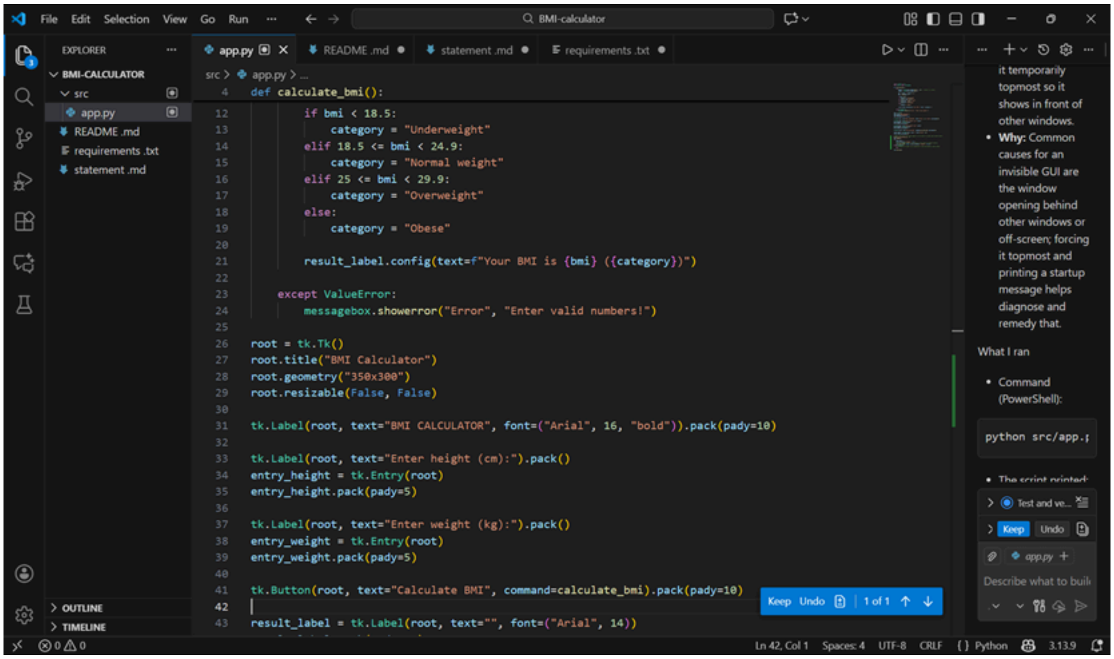
- **Why:** Common causes for an invisible GUI are the window opening behind other windows or off-screen; forcing it topmost and printing a startup message helps diagnose and remedy that.

What I ran

- Command (PowerShell):

```
python src/app.p
```

- The script printed:



```python
def calculate_bmi():
        if bmi < 18.5:
            category = "Underweight"
        elif 18.5 <= bmi < 24.9:
            category = "Normal weight"
        elif 25 <= bmi < 29.9:
            category = "Overweight"
        else:
            category = "Obese"

        result_label.config(text=f"Your BMI is {bmi} ({category})")

    except ValueError:
        messagebox.showerror("Error", "Enter valid numbers!")

root = tk.Tk()
root.title("BMI Calculator")
root.geometry("350x300")
root.resizable(False, False)

tk.Label(root, text="BMI CALCULATOR", font=("Arial", 16, "bold")).pack(pady=10)

tk.Label(root, text="Enter height (cm):").pack()
entry_height = tk.Entry(root)
entry_height.pack(pady=5)

tk.Label(root, text="Enter weight (kg):").pack()
entry_weight = tk.Entry(root)
entry_weight.pack(pady=5)

tk.Button(root, text="Calculate BMI", command=calculate_bmi).pack(pady=10)

result_label = tk.Label(root, text="", font=("Arial", 14))
```

it temporarily topmost so it shows in front of other windows.

- **Why:** Common causes for an invisible GUI are the window opening behind other windows or off-screen; forcing it topmost and printing a startup message helps diagnose and remedy that.

What I ran

- Command (PowerShell):

```
python src/app.p
```

- The script printed:

10. Testing

- Testing performed on:
- Valid numerical inputs
- Invalid inputs
- Boundary BMI values
- GUI responsiveness
- All tests produced correct outputs.

11. Results

The application successfully calculates BMI, shows the category, and handles invalid inputs gracefully.It performs efficiently and offers a user-friendly interface.

12 Conclusion

The BMI Calculator offers a quick and accessible way for users to estimate their Body Mass Index and better understand their general health . I built a complete and functional BMI calculator project which demonstrate practical use of python tkinter .The final interactive version allows user to enter medical measurements and instantly receive a result ,showing how coding can help in real - word healthcare scenarios