**Project of Python**

**Health Data Tracker Report**

Subject: Python Programming(24CAH-606)

Submitted by: Anuj (24MCI10020)

Section\group:24MAM1(A) Branch:MCA(AIML)

Github:- https://github.com/anujdhiman28/Health-Data-Tracker-.git

**1. Introduction:-**

**Aim**

The aim of this project is to develop a user-friendly Health Data Tracker application that allows individuals to input their weight and height in order to calculate their Body Mass Index (BMI). This application is designed using Python and its Tkinter library, providing an accessible way for users to monitor their health metrics.

**Importance of BMI**

Body Mass Index (BMI) is a widely used method to assess body weight relative to height. It serves as an important indicator for determining whether a person is underweight, normal weight, overweight, or obese. Understanding one's BMI can motivate individuals to adopt healthier lifestyle choices.

**2. Objective/Problem Definition:-**

The primary objective of this project is to create a simple tool that helps users calculate their BMI based on their height and weight. This application addresses the following problems:

* **Accessibility**: Many people do not have easy access to health monitoring tools. This application provides a straightforward way for users to check their BMI without needing complicated software.
* **User Engagement**: By simplifying the process of health monitoring, the application encourages users to take an active interest in their health.
* **Awareness**: The application helps raise awareness about the significance of maintaining a healthy weight.

**3. Programming Languages Used:-**

* **Python**: The primary programming language used for developing the Health Data Tracker. Python is known for its simplicity and readability, making it an excellent choice for beginners and experienced programmers alike.
* **Tkinter**: A standard GUI toolkit in Python that allows for the easy creation of graphical user interfaces. It provides various widgets to design interactive applications.

**4. Design Flow:-**

The application operates in a sequential manner, as illustrated in the following block diagram:

1. **Start**
2. **Input Weight and Height**
3. **Validate Input**
4. **Calculate BMI**
5. **Display Results**
6. **End**

**5. Implementation:-**

The application is implemented using Python and Tkinter. Below is the improved code for the Health Data Tracker:

import tkinter as tk

from tkinter import messagebox

def calculate\_bmi():

try:

weight = float(weight\_entry.get())

height = float(height\_entry.get()) / 100 # Convert cm to meters

bmi = weight / (height \*\* 2)

result\_label.config(text=f"BMI: {bmi:.2f}")

# Determine health status

if bmi < 18.5:

status = "Underweight"

elif 18.5 <= bmi < 24.9:

status = "Normal weight"

elif 25 <= bmi < 29.9:

status = "Overweight"

else:

status = "Obesity"

status\_label.config(text=f"Status: {status}")

except ValueError:

messagebox.showerror("Input Error", "Please enter valid numbers.")

# Setting up the main window

root = tk.Tk()

root.title("Health Data Tracker")

root.geometry("300x250")

root.configure(bg="#f0f8ff")

# Title label

title\_label = tk.Label(root, text="Health Data Tracker", font=("Helvetica", 16), bg="#f0f8ff")

title\_label.pack(pady=10)

# Input fields for weight and height

tk.Label(root, text="Weight (kg):", bg="#f0f8ff").pack(pady=5)

weight\_entry = tk.Entry(root)

weight\_entry.pack(pady=5)

tk.Label(root, text="Height (cm):", bg="#f0f8ff").pack(pady=5)

height\_entry = tk.Entry(root)

height\_entry.pack(pady=5)

# Button to calculate BMI

calculate\_button = tk.Button(root, text="Calculate BMI", command=calculate\_bmi, bg="#4CAF50", fg="white")

calculate\_button.pack(pady=15)

# Labels to display the result

result\_label = tk.Label(root, text="", font=("Helvetica", 14), bg="#f0f8ff")

result\_label.pack(pady=5)

status\_label = tk.Label(root, text="", font=("Helvetica", 12), bg="#f0f8ff")

status\_label.pack(pady=5)

# Run the application

root.mainloop()

**Explanation of the Code**

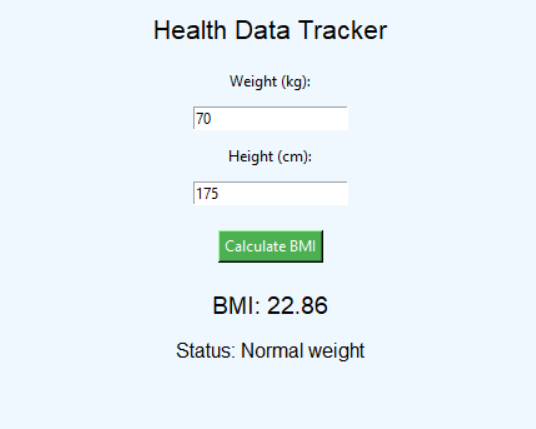
1. **Imports**: The application uses the Tkinter library and the messagebox for error handling.
2. **Function calculate\_bmi()**: This function retrieves the weight and height from user input, calculates BMI, and updates the display with the BMI and health status.
3. **User Interface Setup**: The main window is created, and various widgets are added for input and output.
4. **Main Loop**: The application runs until the user closes the window.

**7. Output**

Upon entering the weight and height, the application calculates and displays the BMI along with a health status message. For example, if a user inputs 70 kg and 175 cm, the output will display:

BMI: 22.86

Status: Normal weight



**8. Conclusion:-**

The Health Data Tracker application successfully provides users with an intuitive way to monitor their BMI. The tool is simple yet effective, making it easy for individuals to keep track of their health metrics. This project demonstrates the capabilities of Python and Tkinter in creating functional applications for everyday use.

**9. Future Framework:-**

Future enhancements for the Health Data Tracker could include:

* **Data Storage**: Allow users to save their BMI history for tracking changes over time.
* **Additional Metrics**: Include features for calculating other health metrics, such as body fat percentage.
* **Visual Representations**: Implement graphs and charts to help users visualize their health data trends.
* **User Profiles**: Create user profiles to store personalized health information and settings.

**10. Learning Outcomes:-**

Through the development of this project, several key learning outcomes were achieved:

* **GUI Development Skills**: Gained experience in creating user interfaces using Tkinter, enhancing understanding of GUI design principles.
* **Health Awareness**: Developed knowledge about BMI and its significance in assessing health.
* **Problem-Solving Abilities**: Improved problem-solving skills through coding challenges and debugging processes.
* **Input Validation Practices**: Learned the importance of validating user input to ensure accurate calculations and improve user experience.