

Group NO -16

JackFruit Problem - BMI calculator with categories

Group Members

- 1)Dhyeyaa Palrecha – PES1UG25EC084**
- 2)Dvaipaayan.P.M – PES1UG25EE013**
- 3)Tejas.K – PES1UG25EC283**

```
import csv
import os
import math
from datetime import datetime

FILE_NAME = "health_history.csv"

def calculate_bmi(weight_kg, height_m):
    """Calculates BMI given weight in kg and height in meters."""
    try:
        bmi = weight_kg / (height_m ** 2)
        return round(bmi, 2)
    except ZeroDivisionError:
        return 0.0

def calculate_body_fat_navy(gender, height_cm, waist_cm, neck_cm, hip_cm=0):
    """
    Calculates Body Fat Percentage (BFP) using the U.S. Navy Method.
    """
    try:
        if gender.lower() == 'm':
            if waist_cm - neck_cm <= 0:
                return 0.0

            log_waist_neck = math.log10(waist_cm - neck_cm)
            log_height = math.log10(height_cm)

            bfp = 495 / (1.0324 - 0.19077 * log_waist_neck + 0.15456 * log_height) - 450

        else:
            if waist_cm + hip_cm - neck_cm <= 0:
                return 0.0

            log_waist_hip_neck = math.log10(waist_cm + hip_cm - neck_cm)
            log_height = math.log10(height_cm)

            bfp = 495 / (1.29579 - 0.35004 * log_waist_hip_neck + 0.22100 * log_height) - 450

        return round(max(bfp, 0), 2)
    except Exception:
        return 0.0
```

```
def get_bmi_category(bmi):
    """Returns the health category based on BMI value."""
    if bmi < 18.5:
        return "Underweight"
    elif 18.5 <= bmi < 24.9:
        return "Normal Weight"
    elif 25 <= bmi < 29.9:
        return "Overweight"
    else:
        return "Obese"

def save_record(name, age, gender, weight, height, waist, neck, hip, bmi, bfp, category):
    """Saves the record to a CSV file."""
    file_exists = os.path.isfile(FILE_NAME)

    try:
        with open(FILE_NAME, mode='a', newline='') as file:
            writer = csv.writer(file)
            if not file_exists:
                writer.writerow(["Date", "Name", "Age", "Gender", "Weight (kg)", "Height (m)", "Waist (cm)", "Neck (cm)", "Hip (cm)", "BMI", "Body Fat %", "Category"])

            timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
            writer.writerow([timestamp, name, age, gender.upper(), weight, height, waist, neck, hip, bmi, bfp, category])

        print(f"\n[Success] Record for {name} saved to '{FILE_NAME}' .")
    except Exception as e:
        print(f"\n[Error] Could not save file: {e}")

def main():
    print("*60")
    print("      COMPREHENSIVE HEALTH TRACKER (U.S. NAVY METHOD)      ")
    print("      (Requires: Weight, Height, Waist, Neck, Hip)      ")
    print("*60")

    while True:
        try:
            print("\n--- New Entry ---")
            name = input("Enter Name (or type 'exit' to quit): ").strip()
            if name.lower() == 'exit':
                print("Exiting program. Stay healthy!")
                break
```

```
age = int(input("Enter Age: "))
gender = input("Enter Gender (M/F): ").strip().lower()

while gender not in ['m', 'f']:
    print("Invalid gender. Please enter 'M' for Male or 'F' for Female.")
    gender = input("Enter Gender (M/F): ").strip().lower()

weight = float(input("Enter Weight (in kg): "))
height_cm = float(input("Enter Height (in cm): "))

neck_cm = float(input("Enter Neck Circumference (in cm): "))
waist_cm = float(input("Enter Waist Circumference (in cm): "))

hip_cm = 0.0
if gender == 'f':
    hip_cm = float(input("Enter Hip Circumference (in cm): "))

if weight <= 0 or height_cm <= 0 or age <= 0:
    print("Error: Inputs must be positive numbers!")
    continue

height_m = height_cm / 100
bmi = calculate_bmi(weight, height_m)
bfp = calculate_body_fat_navy(gender, height_cm, waist_cm, neck_cm, hip_cm)
category = get_bmi_category(bmi)

print("-" * 35)
print(f"Results for {name}:")
print(f" > BMI Score : {bmi}")
print(f" > Body Fat % : {bfp}% (U.S. Navy Method)")
print(f" > Health Category: {category}")
print("-" * 35)

save_cmd = input("Do you want to save this result? (y/n): ").lower()
if save_cmd == 'y':
    save_record(name, age, gender, weight, height_m, waist_cm, neck_cm, hip_cm, bmi, bfp, category)

except ValueError:
    print("Invalid Input! Please enter numeric values.")

if __name__ == "__main__":
    main()
```

```
Last login: Mon Dec  1 09:00:22 on ttys000
dhyeyaa@Dhyeyaas-MacBook-Air ~ % cd desktop/dh
dhyeyaa@Dhyeyaas-MacBook-Air dh % python3 p1.py
```

```
=====
COMPREHENSIVE HEALTH TRACKER (U.S. NAVY METHOD)
(Requires: Weight, Height, Waist, Neck, Hip)
=====
```

```
--- New Entry ---
```

```
Enter Name (or type 'exit' to quit): Hari
```

```
Enter Age: 18
```

```
Enter Gender (M/F): M
```

```
Enter Weight (in kg): 77
```

```
Enter Height (in cm): 183
```

```
Enter Neck Circumference (in cm): 38
```

```
Enter Waist Circumference (in cm): 91
```

```
-----
```

```
Results for Hari:
```

```
> BMI Score : 22.99
```

```
> Body Fat % : 20.02% (U.S. Navy Method)
```

```
> Health Category: Normal Weight
```

```
-----
```

```
Do you want to save this result? (y/n): y
```

```
[Success] Record for Hari saved to 'health_history.csv'.
```

```
--- New Entry ---
```

```
Enter Name (or type 'exit' to quit): exit
```

```
Exiting program. Stay healthy!
```

```
dhyeyaa@Dhyeyaas-MacBook-Air dh %
```

health_history

Date	Name	Age	Gender	Weight (kg)	Height (m)	Waist (cm)	Neck (cm)	Hip (cm)	BMI	Body Fat %	Category
2025-12-01 09:10:23	Hari	18	M	77.0	1.83	91.0	38.0	0.0	22.99	20.02	Normal Weight
2025-12-01 09:14:59	Krish	19	M	72.4	1.785	96.52	38.0	0.0	22.72	24.47	Normal Weight
2025-12-01 09:16:13	Tejas	18	M	53.0	1.64	71.0	28.0	0.0	19.71	15.62	Normal Weight
2025-12-01 09:18:27	navneet	17	M	83.0	1.71	101.6	44.0	0.0	28.38	25.19	Overweight
2025-12-01 09:22:02	Ramesh	18	M	60.0	1.8	66.0	30.0	0.0	18.52	6.61	Normal Weight
2025-12-01 09:27:40	Pooja	18	F	50.0	1.62	60.0	26.0	69.0	19.05	8.54	Normal Weight
2025-12-01 09:31:06	Diya	18	F	75.0	1.65	86.0	38.0	90.0	27.55	27.43	Overweight
2025-12-01 09:34:52	A	18	M	80.0	1.79	98.0	30.0	0.0	24.97	30.11	Obese