**SHRI RAMDEOBABA COLLEGE OF ENGINEERING AND MANAGEMENT, NAGPUR**

**Project Name: Ease Fit Aim:**

1. **The user will provide his or her name, weight, and height, as well as whether or not he or she is a vegetarian or non-vegetarian.**
2. **BMI will be shown.**
3. **In order to reach an appropriate BMI, we will present a suitable diet plan particular to vegetarians or non-vegetarians based on BMI, which will be saved in a file.**
4. **We'll keep track of user information on an excel sheet and keep a user database.**
5. **Show a Line graph of the users' Months vs BMI and Months vs Weight (in kg) from the database (.csv file).**

**Group Members:**

|  |  |
| --- | --- |
| Name | Roll No. |
| Avani Kushwaha | 05 |
| Kunal Sharma | 73 |
|  |  |
|  |  |

# Code:

name=input("Enter Name : ") gender=input("Gender : ") height=float(input("Height(in mts) : ")) weight=int(input("Weight(in kg) : "))

vn=input("Enter \'V\' if you are Vegetarian and \'N\' if you are Non- Vegetarian : ")

vn=vn.lower() bmi=weight/(height\*height) print("BMI :",bmi) weights=[]

if(bmi>=18.5 and bmi<=25): if(vn=='v'):

f=open("Healthy Diet (Vegetarian).txt","r") print(f.read())

else:

f = open("Healthy Diet (Non-Vegetarian).txt", "r") print(f.read())

elif(bmi>=25 and bmi<=29.9): if(vn=='v'):

f=open("Over-Weight Diet (Vegetarian).txt","r") print(f.read())

else:

f = open("Overweight Diet (Non-Vegetarian).txt", "r") print(f.read())

elif(bmi>=30): if(vn=="v"):

f=open("obese veg.txt","r") print(f.read())

else:

f = open("Obese Diet (Non-Vegetarian).txt", "r") print(f.read())

else:

if(vn=="v"):

f = open("Underweight (Vegetarian).txt", "r") print(f.read())

else:

f = open("Underweight (Non-Vegeterian).txt", "r") print(f.read())

weights=[]

print("Enter the weights of 12-months in kg:") for i in range(1,13):

print("Month -",i)

w=int(input("Input - ")) weights.append(w)

print("Weights of past 12-months are:") print(weights)

import pandas as pd

b1 = weights[0]/(height\*height) b2 = weights[1]/(height\*height) b3 = weights[2]/(height\*height) b4 = weights[3]/(height\*height) b5 = weights[4]/(height\*height) b6 = weights[5]/(height\*height) b7 = weights[6]/(height\*height) b8 = weights[7]/(height\*height) b9 = weights[8]/(height\*height) b10 = weights[9]/(height\*height) b11 = weights[10]/(height\*height) b12 = weights[11]/(height\*height) Weight = {

'Months': ["1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"],

'Name': name, 'Gender': gender, 'Height(m)': height,

'Weight(kg)': [weights[0], weights[1], weights[2], weights[3], weights[4], weights[5], weights[6], weights[7], weights[8], weights[9], weights[10], weights[11]],

'BMI': [b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12]

}

df = pd.DataFrame(Weight) df.to\_csv("user\_data.csv",*index*=False)

print("\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Data of 12 months \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n") df = pd.read\_csv('user\_data.csv')

print(df) print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")

import matplotlib.pyplot as plt print("Graph of Months VS Weights\n\n") monthList = df['Months'].tolist() weightList = df['Weight(kg)'].tolist()

plt.plot(monthList,weightList,*label*='Weight', *marker*='o', *markerfacecolor*='k',*linewidth*=3)

plt.xticks(monthList) plt.yticks(weightList)

plt.title("Annual progress Report with weight") plt.xlabel("Months")

plt.ylabel("Weights(kg)")

plt.legend(*loc*='upper right') plt.grid()

plt.show()

print("Graph of Months VS BMI\n\n") monthList = df['Months'].tolist() bmiList = df['BMI'].tolist()

plt.plot(monthList,bmiList,*label*='BMI', *marker*='o', *markerfacecolor*='k',*linewidth*=3) plt.xticks(monthList)

plt.yticks(bmiList)

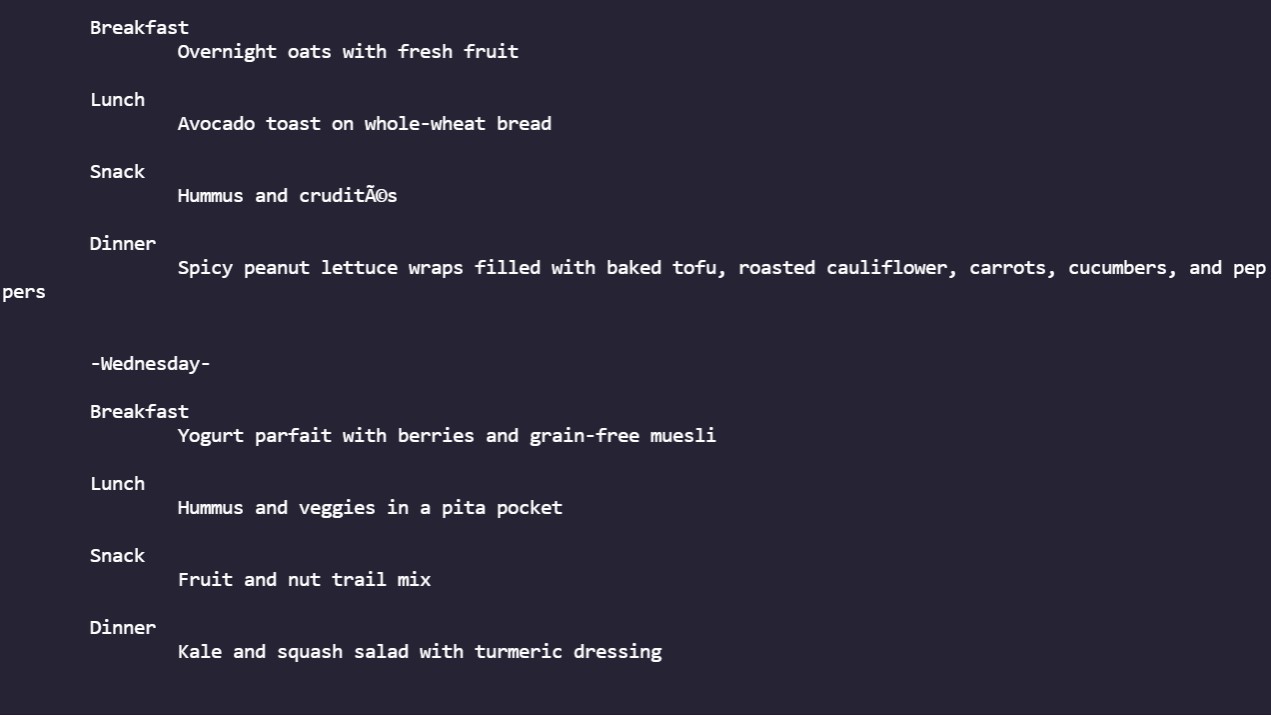
plt.title("Annual progress Report with BMI") plt.xlabel("Months")

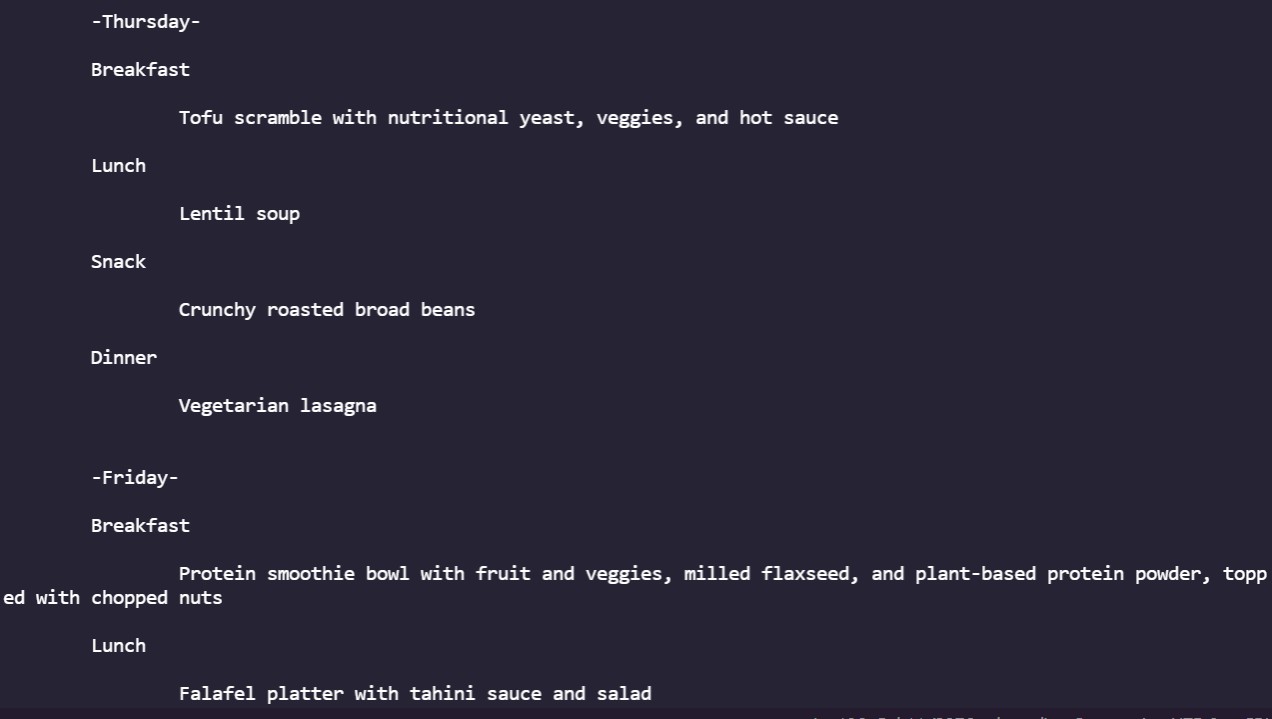
plt.ylabel("BMI") plt.legend(*loc*='upper right') plt.grid()

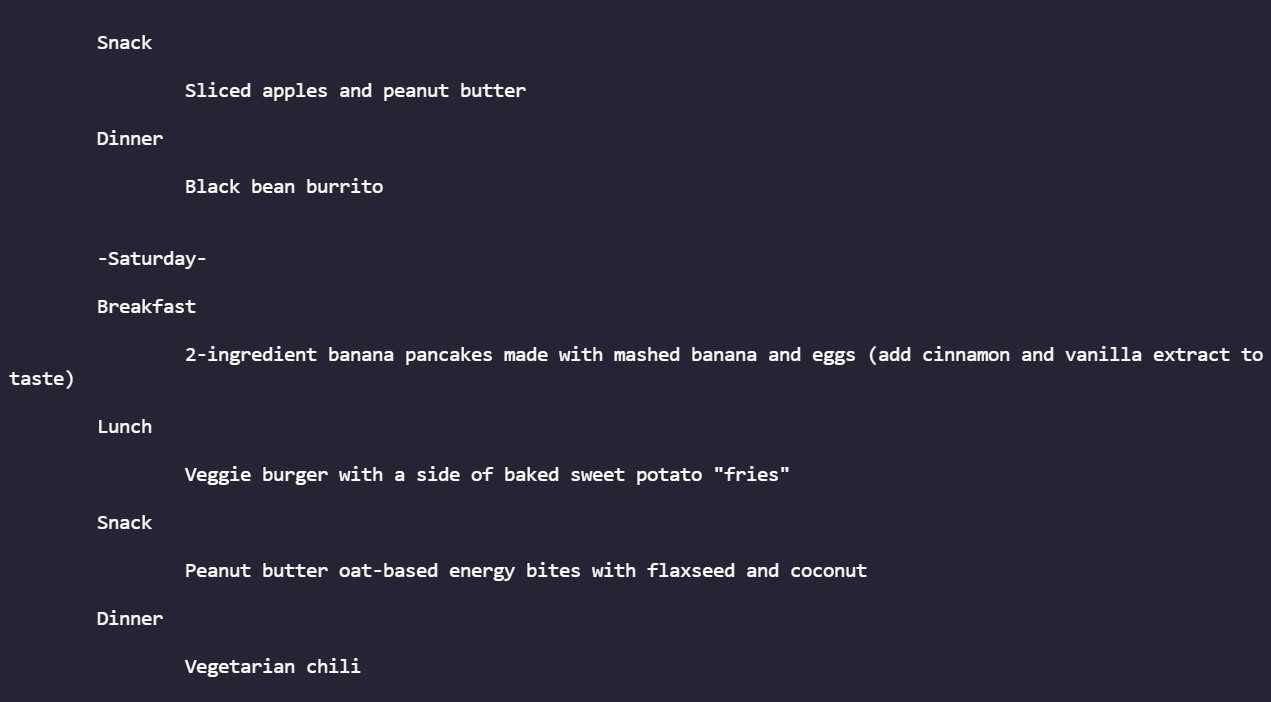
plt.show()

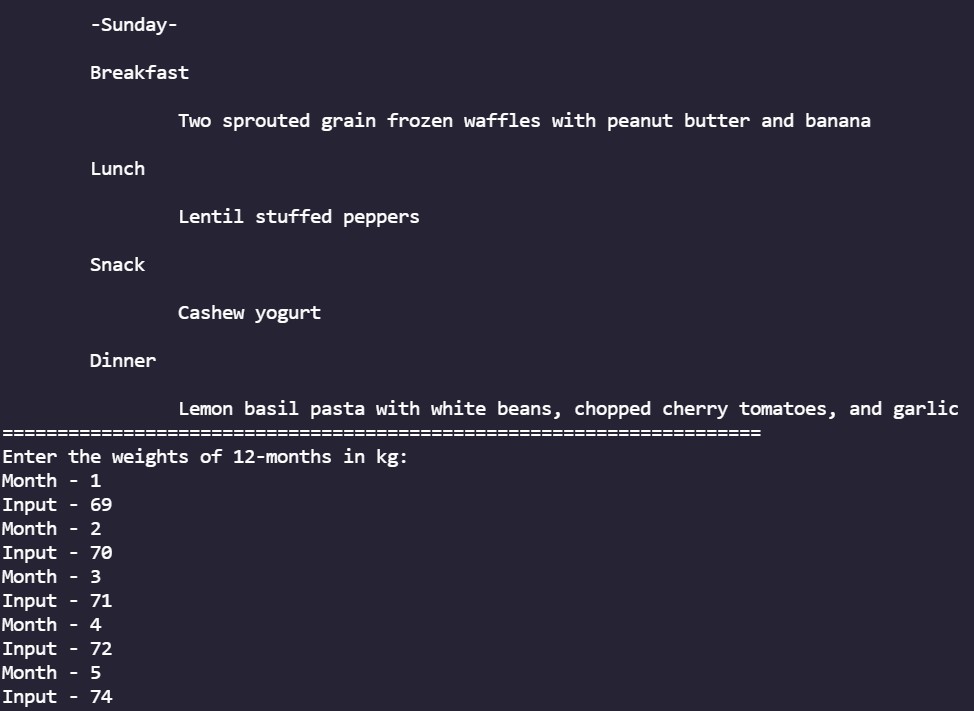
# Output 1:

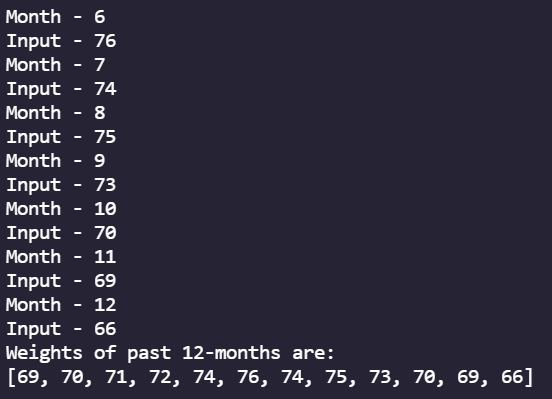


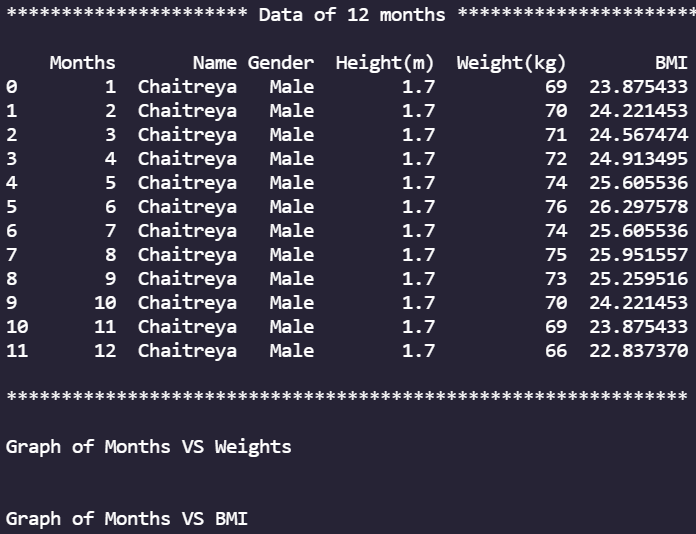


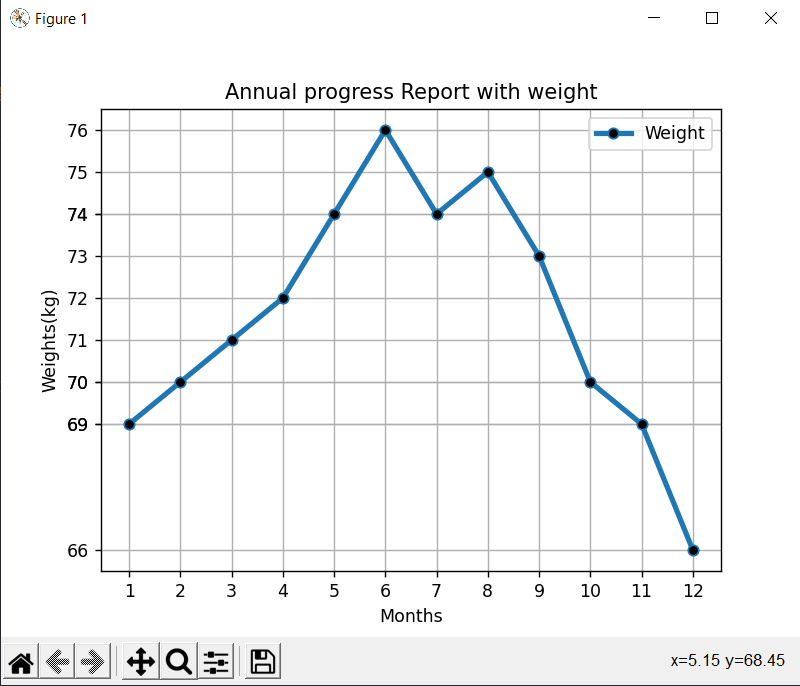


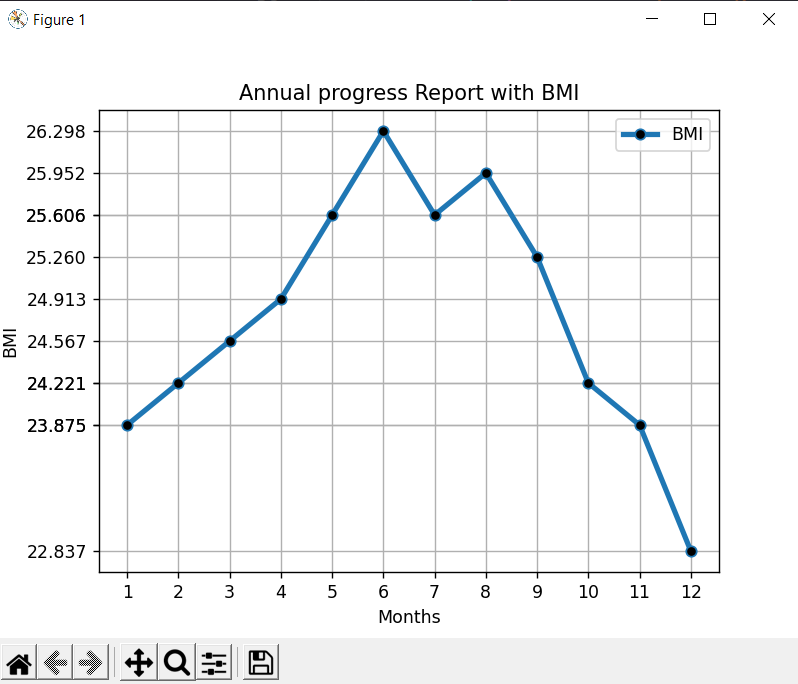


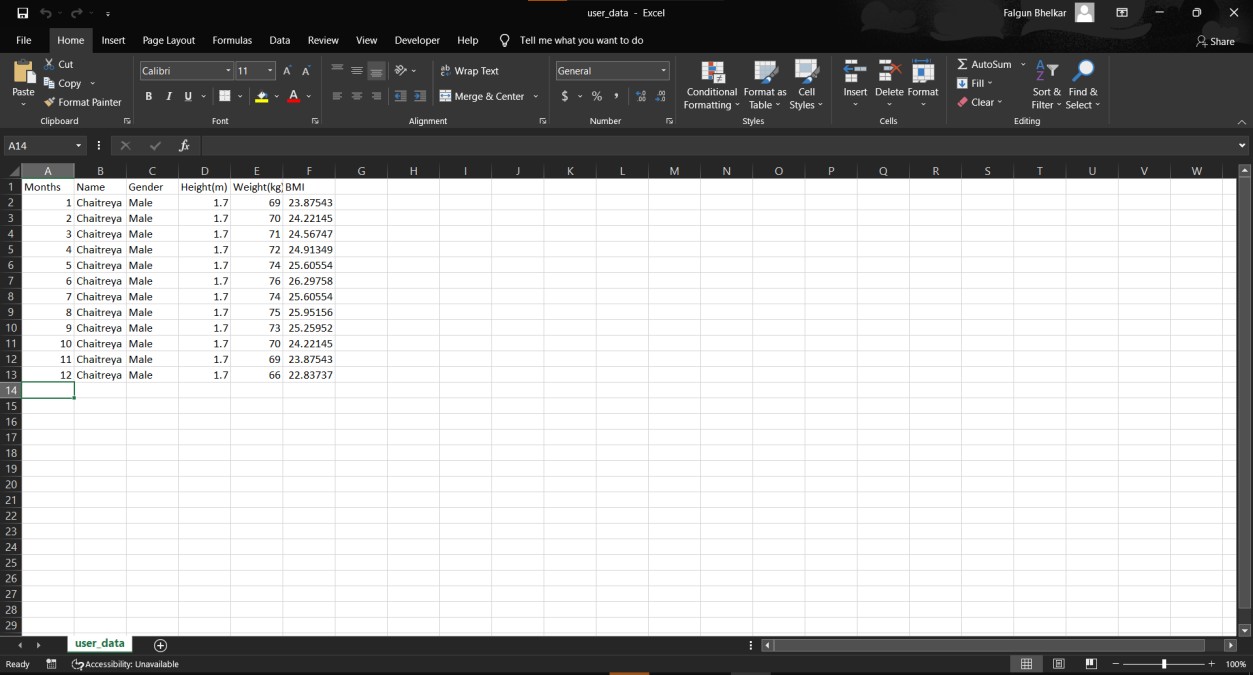












**Output 2:**



