

Diet Planner Project Report

Project Title: Diet Planner Using Python

Course: B.Tech (CSE) – 1st Year

Submitted By: Hemant singh Bhadoriya

Registration Number: 25BCE11032

Submitted To: Dr. Vinesh Kumar

2. Introduction

This project is a simple Python-based Diet Planner that helps users calculate their daily calorie needs and generates a basic meal plan. The program takes inputs like age, gender, activity level, and fitness goal (bulking or shredding). Based on this information, it uses formulas to calculate calories and macros, then suggests meals from a food database.

3. Problem Statement

Many students and beginners struggle to plan their diet properly. They don't know how many calories to eat or what food combinations to choose. Doing it manually is confusing and time-consuming. This project aims to solve this problem by making diet planning simple and automatic.

4. Functional Requirements

- User should be able to enter age, gender, height, weight.
- User should choose their activity level.
- User should select fitness goal: bulking or shredding.
- User should choose veg or non-veg.
- The system should calculate daily calories.
- The system should generate a meal plan based on macros.
- Display results clearly.

5. Non-functional Requirements

- Easy to use and understand.
- Fast calculation.
- Clean and readable output.
- Organized JSON food database.
- Should run on any system having Python installed.

6. System Architecture

A simple architecture is followed:

- **Input Layer:** User provides basic details.
- **Processing Layer:** Calculates BMR, TDEE, macros.
- **Data Layer:** Loads food items from JSON.
- **Output Layer:** Generates final meal plan.

7. Design Diagrams

Use Case Diagram



Workflow Diagram



Sequence Diagram



Display Plan

Class/Component

Components: Input module, Calculation module, Food DB handler, Output module.

ER Diagram (if storage used)

Tables: **Food category → Food items → Macros**

8. Design Decisions & Rationale

- Chose Python because it is easy to use.
- Used JSON for storing food items because it looks clean and organized.
- Broke down the code into functions for readability.
- Kept output simple since project is beginner-level.

9. Implementation Details

- Used formulas for BMR and TDEE.
- Activity multipliers for different lifestyles.
- Macro ratios change based on bulking or shredding.
- Food items selected randomly but according to macro needs.
- Final plan printed in a clean format.

10. Screenshots / Results

```

def get_meal_plan(diet):
    breakfast = "Eggs and toast"
    lunch = "Chicken, rice and vegetables"
    snack = "Fruits or yogurt"
    dinner = "Chicken curry with roti"

    return breakfast, lunch, snack, dinner

def main():
    print("---- DIET PLANNER ----")

    name = input("Enter your name: ")
    gender = input("Gender (male/female): ").lower()
    age = int(input("Age: "))
    height = float(input("Height in cm: "))
    weight = float(input("Weight in kg: "))
    activity = input("Activity level (low/medium/high): ")
    goal = input("Goal (shred/bulk/maintain): ").lower()
    diet = input("Diet type (veg/non-veg): ").lower()

    bmr = calculate_bmr(gender, weight, height, age)
    tdee = calculate_tdee(bmr, activity)
    target_calories = adjust_calories(tdee, goal)

    breakfast, lunch, snack, dinner = get_meal_plan(diet)
    water = weight * 0.035

    print("\n----- RESULT -----")
    print("Name: " + name)

```

```

1  def calculate_bmr(gender, weight, height, age):
2      if gender == "male":
3          s = 5
4      else:
5          s = -161
6      return (10 * weight) + (6.25 * height) - (5 * age) + s
7
8
9  def calculate_tdee(bmr, activity):
10     if activity == "low":
11         return bmr * 1.2
12     elif activity == "medium":
13         return bmr * 1.55
14     elif activity == "high":
15         return bmr * 1.9
16     else:
17         return bmr
18
19
20 def adjust_calories(tdee, goal):
21     if goal == "shred":
22         return tdee - 300
23     elif goal == "bulk":
24         return tdee + 300
25     else:
26         return tdee
27
28

```

```

43
44  def main():
45      print("---- DIET PLANNER ----")
46
47      name = input("Enter your name: ")
48      gender = input("Gender (male/female): ").lower()
49      age = int(input("Age: "))
50      height = float(input("Height in cm: "))
51      weight = float(input("Weight in kg: "))
52      activity = input("Activity level (low/medium/high): ").lower()
53      goal = input("Goal (shred/bulk/maintain): ").lower()
54      diet = input("Diet type (veg/non-veg): ").lower()
55
56      bmr = calculate_bmr(gender, weight, height, age)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\python tutorials> & C:/Users/heman/AppData/Local/Programs/Python/Python314/python.exe "d:/python tutorials/f
---- DIET PLANNER ----
Enter your name: ATHARV DATAR
Gender (male/female): male
Age: 18
Height in cm: 185
Weight in kg: 85
Activity level (low/medium/high): medium
Goal (shred/bulk/maintain): shred
Diet type (veg/non-veg): veg

----- RESULT -----
Name: ATHARV DATAR
BMR: 1921
Daily Calorie Requirement: 2678

```

```
---- DIET PLANNER ----
Enter your name: ATHARV DATAR
Gender (male/female): male
Age: 18
Height in cm: 185
Weight in kg: 85
Activity level (low/medium/high): medium
Goal (shred/bulk/maintain): shred
Diet type (veg/non-veg): veg

----- RESULT -----
Name: ATHARV DATAR
BMR: 1921
Daily Calorie Requirement: 2678

Daily Meal Plan:
Breakfast: Oats with milk
Lunch: Dal, rice and vegetables
Snack: Fruits or nuts
Dinner: Paneer with roti

Recommended Water Intake (liters): 2.98
PS D:\python tutorials> █
```

11. Testing Approach

- Tested with different ages and genders.
- Verified calorie results manually.
- Checked both veg and non-veg.
- Tried extreme inputs to make sure program doesn't crash.

12. Challenges Faced

- Understanding calorie formulas.
- Managing a large food database.
- Formatting the output neatly.
- Writing clean JSON.

13. Learnings & Key Takeaways

- Learned how to structure a Python project.
- Understood JSON data handling.
- Improved knowledge of diet and nutrition basics.
- Practiced writing cleaner and modular code.

14. Future Enhancements

- Adding a GUI using Tkinter.
- Adding weekly meal plans.
- Storing user data in a database.
- Adding graphs for calories.
- Creating a mobile app version.

15. References

- Python official documentation
- Fitness and nutrition formula resources
- Articles on TDEE and BMR
- Online calorie calculator examples