Project Documentation: AI Personalized Workout & Diet Planner

Project Title: AI Personalized Workout & Diet Planner **Submitted by:** [Your Name] **Course:** [Your Course Name] **Date:** [Submission Date]

1. Introduction

This project is designed to assist individuals in creating a personalized fitness plan based on age, weight, height, gender, activity level, fitness goals, and diet preference. It uses Python programming to generate personalized workout routines and diet plans. The project provides step-by-step guidance for users to plan their workouts and diets efficiently.

2. Objectives of the Project

1. Provide personalized workout plans.
2. Generate daily diet plans.
3. Calculate BMR (Basal Metabolic Rate) and TDEE (Total Daily Energy Expenditure).
4. Track user progress and suggest calorie adjustments.
5. Educate users about healthy lifestyle habits.

3. Features of the Project

- User Input Collection: Name, Age, Gender, Weight, Height, Activity Level, Goal, Diet Preference.
- BMR & Calorie Calculation: Uses Mifflin-St Jeor equation.
- Workout Plan Generator: Weekly workout routines based on user goal.
- Diet Plan Generator: Randomly selects 3 meals per day.
- Progress Tips: Adjust calories based on user progress every two weeks.

4. Uses of the Project

- Personal Fitness Planning.
- Diet Guidance.
- Time Saving.
- AI Assistance with adaptive recommendations.
- Motivation for consistent fitness routines.

5. Technologies and Tools Used

Component	Purpose
Python	Programming language
Random module	To select random meal options
Conditional Statements	Create logic for goals, diet, activity levels
Functions	Structure code for workouts and meals
Console I/O	Interact with user and display results

---

6. Artificial Intelligence Aspect

- Adaptive Recommendations based on user input.
- Randomized Meal Selection.
- Decision Logic for goals (gain, lose, maintain).

---

7. Benefits to Users

1. Tailored Fitness Plans.
2. Ease of Use.
3. Nutritional Guidance.
4. Flexible and Adjustable.
5. Health Awareness.

---

8. Special Features Added

- Complete Weekly Workout Plan.
- Multiple Diet Options.
- Calorie & BMR Calculation.
- Randomization for Variety.
- User-Friendly Messages.

---

9. Full Code Explanation (Line by Line)

9.1 Import Libraries

```
import random
```

- Imports the Python `random` module to allow selection of random meals from the list.

9.2 Display Program Title

```
print("\n===========================================")
print("💪 AI PERSONALIZED WORKOUT & DIET PLANNER 💪")
print("===========================================\n")
```

- Prints a header for the program for better user interaction.

9.3 Collect User Information

```
name = input("Enter your name: ").capitalize()
age = int(input("Enter your age: "))
gender = input("Enter your gender (Male/Female): ").lower()
weight = float(input("Enter your weight (kg): "))
height = float(input("Enter your height (cm): "))
activity = input("Activity level (low / medium / high): ").lower()
goal = input("Goal (gain / lose / maintain): ").lower()
diet_type = input("Diet preference (veg / nonveg): ").lower()
```

- Collects all necessary user information. - `.capitalize()` ensures the first letter of the name is uppercase. - `.lower()` standardizes string inputs for easier conditional checks. - `int()` and `float()` convert numerical inputs to the correct types.

9.4 Calculate BMR

```
if gender == "male":
    bmr = 10 * weight + 6.25 * height - 5 * age + 5
else:
    bmr = 10 * weight + 6.25 * height - 5 * age - 161
```

- Uses the Mifflin-St Jeor equation to calculate BMR. - Adjusts for gender: +5 for male, -161 for female.

9.5 Adjust Calories for Activity Level

```
if activity == "low":
    calories = bmr * 1.2
elif activity == "medium":
    calories = bmr * 1.55
else:
    calories = bmr * 1.75
```

- Multiplies BMR by an activity factor to get TDEE.

9.6 Adjust Calories for Goal

```
if goal == "gain":
    calories += 300
elif goal == "lose":
    calories -= 300
```

- Adjusts calorie intake according to user fitness goal.

9.7 Workout Plan Function

```
def workout_plan(goal):
    if goal == "gain":
        return { ... }  # Weekly strength plan
    elif goal == "lose":
        return { ... }  # Weekly fat loss plan
    else:
        return { ... }  # Balanced maintenance plan
```

- Returns a dictionary containing daily exercises for the week.

9.8 Diet Plan Lists

```
veg_foods = [ ... ]
nonveg_foods = [ ... ]
```

- Lists of sample vegetarian and non-vegetarian foods.

9.9 Diet Plan Function

```
def meal_plan(diet, calories):
    if diet == "veg":
        foods = random.sample(veg_foods, 3)
    else:
        foods = random.sample(nonveg_foods, 3)
    print("\n--- Sample Daily Diet Plan ---")
    print(f"Target Calories: {int(calories)} kcal/day")
    for i, food in enumerate(foods, 1):
        print(f"Meal {i}: {food}")
```

- Randomly selects 3 meals. - Displays each meal and total calories.

9.10 Display Results

```
print(f"Hello {name} 💪 Here’s your Personalized Plan!")
print(f"Estimated BMR: {int(bmr)} kcal/day")
print(f"Recommended Daily Calories: {int(calories)} kcal/day\n")
```

- Greets the user and displays BMR and daily calorie recommendation.

```
print("   Workout Plan:")
for day, routine in workout_plan(goal).items():
    print(f"{day}: {routine}")
```

- Prints daily workout plan.

```
meal_plan(diet_type, calories)
```

- Calls the diet function to display meals.

```
print("\n❤️ Tip: Track your progress every 2 weeks and adjust your calories by
±200 kcal if needed!")
```

- Provides advice for tracking and adjusting progress.

---

10. Conclusion

The AI Personalized Workout & Diet Planner is an effective tool for fitness improvement, weight management, and muscle gain. It demonstrates the application of Python and basic AI logic in real-world solutions and can be further enhanced in future versions with GUI, database tracking, or ML-based predictive algorithms.