

 FITNESS.py > ...

```
1  import tkinter as tk
2  from tkinter import ttk
3  from tkinter import messagebox
4
5  # ----- BMI FUNCTIONS -----
6  def calculate_bmi(weight, height):
7      height_m = height / 100
8      return round(weight / (height_m ** 2), 2)
9
10 def bmi_category(bmi):
11     if bmi < 18.5:
12         return "Underweight"
13     elif 18.5 <= bmi < 25:
14         return "Normal"
15     elif 25 <= bmi < 30:
16         return "Overweight"
17     else:
18         return "Obese"
19
```

```
# ----- WORKOUT PLANS -----  
def beginner_workout_plan(goal):  
    plans = {  
        "Weight Loss": [  
            "Day 1: Cardio + Walk",  
            "Day 2: Upper Body + Core",  
            "Day 3: Lower Body + Cardio",  
            "Day 4: Yoga / Stretching",  
            "Day 5: Full Body Strength"  
        ],  
        "Muscle Gain": [  
            "Day 1: Chest + Triceps",  
            "Day 2: Back + Biceps",  
            "Day 3: Legs",  
            "Day 4: Shoulders + Core",  
            "Day 5: Full Body Strength"  
        ],  
        "General Fitness": [  
            "Day 1: Light Cardio + Stretching",  
            "Day 2: Upper Body Strength",  
            "Day 3: Lower Body Strength",  
            "Day 4: Yoga / Mobility",  
            "Day 5: Mixed Workout"  
        ]  
    }  
    return plans.get(goal, [])
```

```
# ----- DIET PLANS -----
```

```
def beginner_diet_plan(goal):  
    diets = {  
        "Weight Loss": [  
            "Breakfast: Oats + Fruits",  
            "Lunch: Rice + Veg + Salad",  
            "Dinner: Chapati + Veg",  
            "Snacks: Green Tea + Nuts"  
        ],  
        "Muscle Gain": [  
            "Breakfast: Eggs + Banana",  
            "Lunch: Rice + Paneer/Chicken",  
            "Dinner: Chapati + Chicken",  
            "Snacks: Milk + Nuts"  
        ],  
        "General Fitness": [  
            "Breakfast: Idli/Dosa + Sambar",  
            "Lunch: Balanced Rice + Veg + Dal",  
            "Dinner: Chapati + Veg",  
            "Snacks: Fruits"  
        ]  
    }  
    return diets.get(goal, [])
```

```
# ----- WATER INTAKE -----
```

```
def daily_water_intake(weight):  
    return round(weight * 0.033, 2)
```

```
# ----- MAIN FUNCTION -----
```

```
def generate_plan():
    try:
        name = name_entry.get()
        weight = float(weight_entry.get())
        height = float(height_entry.get())
        goal = goal_combobox.get()

        if goal == "":
            messagebox.showerror("Error", "Select a fitness goal!")
            return

        bmi = calculate_bmi(weight, height)
        category = bmi_category(bmi)
        water = daily_water_intake(weight)
        workout = beginner_workout_plan(goal)
        diet = beginner_diet_plan(goal)

        output_text.delete("1.0", tk.END)
        output_text.insert(tk.END, f"Hello {name}!\n\n")
        output_text.insert(tk.END, f"BMI: {bmi} ({category})\n\n")

        output_text.insert(tk.END, "Workout Plan:\n")
        for w in workout:
            output_text.insert(tk.END, f"• {w}\n")

        output_text.insert(tk.END, "\nDiet Plan:\n")
        for d in diet:
            output_text.insert(tk.END, f"• {d}\n")

        output_text.insert(tk.END, f"\nDaily Water Intake: {water} Litres\n")

    except ValueError:
        messagebox.showerror("Error", "Enter valid numbers for weight and height!")
```

```

# ----- GUI SETUP -----
root = tk.Tk()
root.title("Gym Beginner Health Assistant")
root.geometry("600x600")
root.config(bg="#f0f3f5")

title = tk.Label(root, text="Gym Beginner Health Assistant",
                 font=("Arial", 16, "bold"), bg="#f0f3f5")
title.pack(pady=10)

frame = tk.Frame(root, bg="#f0f3f5")
frame.pack(pady=10)

tk.Label(frame, text="Name:", font=("Arial", 12), bg="#f0f3f5").grid(row=0, column=0, sticky="w")
name_entry = tk.Entry(frame, font=("Arial", 12))
name_entry.grid(row=0, column=1, pady=5)

tk.Label(frame, text="Weight (kg):", font=("Arial", 12), bg="#f0f3f5").grid(row=1, column=0, sticky="w")
weight_entry = tk.Entry(frame, font=("Arial", 12))
weight_entry.grid(row=1, column=1, pady=5)

tk.Label(frame, text="Height (cm):", font=("Arial", 12), bg="#f0f3f5").grid(row=2, column=0, sticky="w")
height_entry = tk.Entry(frame, font=("Arial", 12))
height_entry.grid(row=2, column=1, pady=5)

tk.Label(frame, text="Goal:", font=("Arial", 12), bg="#f0f3f5").grid(row=3, column=0, sticky="w")
goal_combobox = ttk.Combobox(frame, values=["Weight Loss", "Muscle Gain", "General Fitness"], font=("Arial", 12))
goal_combobox.grid(row=3, column=1, pady=5)

generate_btn = tk.Button(root, text="Generate Plan", font=("Arial", 14, "bold"),
                        bg="#4CAF50", fg="white", command=generate_plan)
generate_btn.pack(pady=10)

output_text = tk.Text(root, height=20, width=60, font=("Arial", 11))
output_text.pack(pady=10)

root.mainloop()

```