Project title: Fitness tracker and goal setter

Project description: a fitness tracker and goal setter program that allows users to set fitness goals, track their progress, and receive motivational messages.

# List out modules/ functions of project:

Modules:

1. `tkinter`: This module is used for creating graphical user interfaces.
2. `tkinter.messagebox`: This submodule is specifically used for creating message boxes for displaying information or prompting the user.

Functions:

1. `calculate\_progress()`: This function calculates the progress of various fitness goals based on user input and displays the progress in a message box.
2. `get\_motivational\_message(progress)`: This function returns a motivational message based on the progress percentage passed as an argument.
3. `clear\_data()`: This function clears all the data entered by the user in the input fields.
4. `set\_goals()`: This function clears existing data and sets default goals.
5. `set\_default\_goals()`: This function sets default goals in the input fields.

Additionally, there are several tkinter widget creation functions used such as `Label`, `Entry`, and

`Button`.

# Explain in short different predefined modules required for project.:

1. tkinter (`import tkinter as tk`):
   * This module is a standard GUI (Graphical User Interface) toolkit for Python.
   * It provides various widgets like buttons, labels, entry fields, etc., for creating desktop applications with graphical elements.
   * In the project, tkinter is used to create the main window, labels, entry fields, and buttons for user interaction.
2. tkinter.messagebox (`import tkinter.messagebox as messagebox`):
   * This submodule of tkinter provides methods for creating different types of message boxes, such as info boxes, warning boxes, and error boxes.
   * It's used in the project to display progress reports and error messages to the user.

These modules are essential for creating a GUI-based fitness and goal tracking application. Tkinter provides the necessary tools for building the user interface, while the `messagebox` submodule adds functionality for displaying messages to the user.

# What are the inputs/outputs of project: Code:

# import tkinter as tk

# from tkinter import messagebox

# def open\_registration\_window():

# registration\_window = tk.Toplevel(root)

# registration\_window.title("User Registration")

# registration\_window.geometry("300x300")

# # Create labels and entries for user registration

# tk.Label(registration\_window, text="UserID:").grid(row=0, column=0, padx=5, pady=5)

# user\_id\_entry = tk.Entry(registration\_window)

# user\_id\_entry.grid(row=0, column=1, padx=5, pady=5)

# tk.Label(registration\_window, text="Name:").grid(row=1, column=0, padx=5, pady=5)

# name\_entry = tk.Entry(registration\_window)

# name\_entry.grid(row=1, column=1, padx=5, pady=5)

# tk.Label(registration\_window, text="Age:").grid(row=2, column=0, padx=5, pady=5)

# age\_entry = tk.Entry(registration\_window)

# age\_entry.grid(row=2, column=1, padx=5, pady=5)

# tk.Label(registration\_window, text="Gender:").grid(row=3, column=0, padx=5, pady=5)

# gender\_entry = tk.Entry(registration\_window)

# gender\_entry.grid(row=3, column=1, padx=5, pady=5)

# tk.Label(registration\_window, text="Height:").grid(row=4, column=0, padx=5, pady=5)

# height\_entry = tk.Entry(registration\_window)

# height\_entry.grid(row=4, column=1, padx=5, pady=5)

# tk.Label(registration\_window, text="Weight:").grid(row=5, column=0, padx=5, pady=5)

# weight\_entry = tk.Entry(registration\_window)

# weight\_entry.grid(row=5, column=1, padx=5, pady=5)

# tk.Label(registration\_window, text="Email:").grid(row=6, column=0, padx=5, pady=5)

# email\_entry = tk.Entry(registration\_window)

# email\_entry.grid(row=6, column=1, padx=5, pady=5)

# # Button to submit registration

# submit\_button = tk.Button(registration\_window, text="Submit Registration", command=lambda: register\_user(registration\_window))

# submit\_button.grid(row=7, columnspan=2, padx=5, pady=5)

# def register\_user(window):

# # Perform registration tasks here

# # For demonstration purposes, let's assume the registration is successful

# messagebox.showinfo("Success", "User registered successfully!")

# window.destroy()

# def open\_bmi\_calculator():

# bmi\_window = tk.Toplevel(root)

# bmi\_window.title("BMI Calculator")

# bmi\_window.geometry("300x200")

# # Create labels and entries for BMI calculation

# tk.Label(bmi\_window, text="Enter Height (m):").grid(row=0, column=0, padx=5, pady=5)

# height\_entry\_bmi = tk.Entry(bmi\_window)

# height\_entry\_bmi.grid(row=0, column=1, padx=5, pady=5)

# tk.Label(bmi\_window, text="Enter Weight (kg):").grid(row=1, column=0, padx=5, pady=5)

# weight\_entry\_bmi = tk.Entry(bmi\_window)

# weight\_entry\_bmi.grid(row=1, column=1, padx=5, pady=5)

# # Button to calculate BMI

# calculate\_bmi\_button = tk.Button(bmi\_window, text="Calculate BMI", command=lambda: calculate\_bmi(bmi\_window, height\_entry\_bmi, weight\_entry\_bmi))

# calculate\_bmi\_button.grid(row=2, columnspan=2, padx=5, pady=5)

# # Label to display BMI result

# global bmi\_label\_bmi

# bmi\_label\_bmi = tk.Label(bmi\_window, text="")

# bmi\_label\_bmi.grid(row=3, columnspan=2, padx=5, pady=5)

# # Label to display BMI category

# global category\_label

# category\_label = tk.Label(bmi\_window, text="")

# category\_label.grid(row=4, columnspan=2, padx=5, pady=5)

# def calculate\_bmi(window, height\_entry, weight\_entry):

# try:

# height = float(height\_entry.get())

# weight = float(weight\_entry.get())

# bmi = weight / (height \*\* 2)

# bmi\_label\_bmi.config(text=f"BMI: {bmi:.2f}")

# if bmi < 18.5 or bmi >= 25:

# open\_weight\_management\_window(bmi)

# else:

# category\_label.config(text="Category: Normal")

# except ValueError:

# error\_label = tk.Label(window, text="Please enter valid height and weight values.")

# error\_label.grid(row=3, columnspan=2, padx=5, pady=5)

# def open\_weight\_management\_window(bmi):

# weight\_management\_window = tk.Toplevel(root)

# weight\_management\_window.title("Weight Management")

# if bmi < 18.5:

# tk.Label(weight\_management\_window, text="You are underweight. Choose an option to gain weight:").pack(pady=10)

# tk.Button(weight\_management\_window, text="Weight Gain Plan", command=open\_weight\_gain\_plan).pack(pady=5)

# else:

# tk.Label(weight\_management\_window, text="You are overweight. Choose an option to lose weight:").pack(pady=10)

# tk.Button(weight\_management\_window, text="Weight Loss Plan", command=open\_weight\_loss\_plan).pack(pady=5)

# 

# # Define global variables

# steps\_goal\_entry = None

# calories\_goal\_entry = None

# minutes\_goal\_entry = None

# water\_goal\_entry = None

# sleep\_goal\_entry = None

# steps\_taken\_entry = None

# calories\_burned\_entry = None

# minutes\_active\_entry = None

# water\_intake\_entry = None

# sleep\_duration\_entry = None

# def open\_weight\_gain\_plan():

# global steps\_goal\_entry, calories\_goal\_entry, minutes\_goal\_entry, water\_goal\_entry, sleep\_goal\_entry, steps\_taken\_entry, calories\_burned\_entry, minutes\_active\_entry, water\_intake\_entry, sleep\_duration\_entry

# 

# weight\_gain\_window = tk.Toplevel(root)

# weight\_gain\_window.title("Weight Gain Plan")

# # Create labels and entries for user input

# tk.Label(weight\_gain\_window, text="Enter Daily Steps Goal:").grid(row=0, column=0, padx=5, pady=5)

# steps\_goal\_entry = tk.Entry(weight\_gain\_window)

# steps\_goal\_entry.grid(row=0, column=1, padx=5, pady=5)

# tk.Label(weight\_gain\_window, text="Enter Daily Calories Goal:").grid(row=1, column=0, padx=5, pady=5)

# calories\_goal\_entry = tk.Entry(weight\_gain\_window)

# calories\_goal\_entry.grid(row=1, column=1, padx=5, pady=5)

# tk.Label(weight\_gain\_window, text="Enter Daily Active Minutes Goal:").grid(row=2, column=0, padx=5, pady=5)

# minutes\_goal\_entry = tk.Entry(weight\_gain\_window)

# minutes\_goal\_entry.grid(row=2, column=1, padx=5, pady=5)

# tk.Label(weight\_gain\_window, text="Enter Daily Water Intake Goal (in liters):").grid(row=3, column=0, padx=5, pady=5)

# water\_goal\_entry = tk.Entry(weight\_gain\_window)

# water\_goal\_entry.grid(row=3, column=1, padx=5, pady=5)

# tk.Label(weight\_gain\_window, text="Enter Daily Sleep Duration Goal (in hours):").grid(row=4, column=0, padx=5, pady=5)

# sleep\_goal\_entry = tk.Entry(weight\_gain\_window)

# sleep\_goal\_entry.grid(row=4, column=1, padx=5, pady=5)

# tk.Label(weight\_gain\_window, text="Enter Steps Taken:").grid(row=5, column=0, padx=5, pady=5)

# steps\_taken\_entry = tk.Entry(weight\_gain\_window)

# steps\_taken\_entry.grid(row=5, column=1, padx=5, pady=5)

# tk.Label(weight\_gain\_window, text="Enter Calories Burned:").grid(row=6, column=0, padx=5, pady=5)

# calories\_burned\_entry = tk.Entry(weight\_gain\_window)

# calories\_burned\_entry.grid(row=6, column=1, padx=5, pady=5)

# tk.Label(weight\_gain\_window, text="Enter Minutes Active:").grid(row=7, column=0, padx=5, pady=5)

# minutes\_active\_entry = tk.Entry(weight\_gain\_window)

# minutes\_active\_entry.grid(row=7, column=1, padx=5, pady=5)

# tk.Label(weight\_gain\_window, text="Enter Water Intake (in liters):").grid(row=8, column=0, padx=5, pady=5)

# water\_intake\_entry = tk.Entry(weight\_gain\_window)

# water\_intake\_entry.grid(row=8, column=1, padx=5, pady=5)

# tk.Label(weight\_gain\_window, text="Enter Sleep Duration (in hours):").grid(row=9, column=0, padx=5, pady=5)

# sleep\_duration\_entry = tk.Entry(weight\_gain\_window)

# sleep\_duration\_entry.grid(row=9, column=1, padx=5, pady=5)

# # Create buttons for actions

# calculate\_button = tk.Button(weight\_gain\_window, text="Calculate Progress", command=calculate\_progress)

# calculate\_button.grid(row=10, columnspan=2, padx=5, pady=5)

# clear\_button = tk.Button(weight\_gain\_window, text="Clear Data", command=clear\_data)

# clear\_button.grid(row=11, columnspan=2, padx=5, pady=5)

# set\_goals\_button = tk.Button(weight\_gain\_window, text="Set Goals", command=set\_goals)

# set\_goals\_button.grid(row=11, column=0, padx=5, pady=5)

# set\_default\_goals\_button = tk.Button(weight\_gain\_window, text="Set Default Goals", command=set\_default\_goals)

# set\_default\_goals\_button.grid(row=11, column=1, padx=5, pady=5)

# tk.Label(weight\_gain\_window, text="Weight Gain Exercises:").pack(pady=10)

# tk.Label(weight\_gain\_window, text="1. Squats").pack()

# tk.Label(weight\_gain\_window, text="2. Deadlifts").pack()

# tk.Label(weight\_gain\_window, text="3. Bench Press").pack()

# def open\_weight\_loss\_plan():

# global steps\_goal\_entry, calories\_goal\_entry, minutes\_goal\_entry, water\_goal\_entry, sleep\_goal\_entry, steps\_taken\_entry, calories\_burned\_entry, minutes\_active\_entry, water\_intake\_entry, sleep\_duration\_entry

# 

# weight\_loss\_window = tk.Toplevel(root)

# weight\_loss\_window.title("Weight Loss Plan")

# # Create labels and entries for user input

# tk.Label(weight\_loss\_window, text="Enter Daily Steps Goal:").grid(row=0, column=0, padx=5, pady=5)

# steps\_goal\_entry = tk.Entry(weight\_loss\_window)

# steps\_goal\_entry.grid(row=0, column=1, padx=5, pady=5)

# tk.Label(weight\_loss\_window, text="Enter Daily Calories Goal:").grid(row=1, column=0, padx=5, pady=5)

# calories\_goal\_entry = tk.Entry(weight\_loss\_window)

# calories\_goal\_entry.grid(row=1, column=1, padx=5, pady=5)

# tk.Label(weight\_loss\_window, text="Enter Daily Active Minutes Goal:").grid(row=2, column=0, padx=5, pady=5)

# minutes\_goal\_entry = tk.Entry(weight\_loss\_window)

# minutes\_goal\_entry.grid(row=2, column=1, padx=5, pady=5)

# tk.Label(weight\_loss\_window, text="Enter Daily Water Intake Goal (in liters):").grid(row=3, column=0, padx=5, pady=5)

# water\_goal\_entry = tk.Entry(weight\_loss\_window)

# water\_goal\_entry.grid(row=3, column=1, padx=5, pady=5)

# tk.Label(weight\_loss\_window, text="Enter Daily Sleep Duration Goal (in hours):").grid(row=4, column=0, padx=5, pady=5)

# sleep\_goal\_entry = tk.Entry(weight\_loss\_window)

# sleep\_goal\_entry.grid(row=4, column=1, padx=5, pady=5)

# tk.Label(weight\_loss\_window, text="Enter Steps Taken:").grid(row=5, column=0, padx=5, pady=5)

# steps\_taken\_entry = tk.Entry(weight\_loss\_window)

# steps\_taken\_entry.grid(row=5, column=1, padx=5, pady=5)

# tk.Label(weight\_loss\_window, text="Enter Calories Burned:").grid(row=6, column=0, padx=5, pady=5)

# calories\_burned\_entry = tk.Entry(weight\_loss\_window)

# calories\_burned\_entry.grid(row=6, column=1, padx=5, pady=5)

# tk.Label(weight\_loss\_window, text="Enter Minutes Active:").grid(row=7, column=0, padx=5, pady=5)

# minutes\_active\_entry = tk.Entry(weight\_loss\_window)

# minutes\_active\_entry.grid(row=7, column=1, padx=5, pady=5)

# tk.Label(weight\_loss\_window, text="Enter Water Intake (in liters):").grid(row=8, column=0, padx=5, pady=5)

# water\_intake\_entry = tk.Entry(weight\_loss\_window)

# water\_intake\_entry.grid(row=8, column=1, padx=5, pady=5)

# tk.Label(weight\_loss\_window, text="Enter Sleep Duration (in hours):").grid(row=9, column=0, padx=5, pady=5)

# sleep\_duration\_entry = tk.Entry(weight\_loss\_window)

# sleep\_duration\_entry.grid(row=9, column=1, padx=5, pady=5)

# # Create buttons for actions

# calculate\_button = tk.Button(weight\_loss\_window, text="Calculate Progress", command=calculate\_progress)

# calculate\_button.grid(row=10, columnspan=2, padx=5, pady=5)

# clear\_button = tk.Button(weight\_loss\_window, text="Clear Data", command=clear\_data)

# clear\_button.grid(row=11, columnspan=2, padx=5, pady=5)

# set\_goals\_button = tk.Button(weight\_loss\_window, text="Set Goals", command=set\_goals)

# set\_goals\_button.grid(row=11, column=0, padx=5, pady=5)

# set\_default\_goals\_button = tk.Button(weight\_loss\_window, text="Set Default Goals", command=set\_default\_goals)

# set\_default\_goals\_button.grid(row=11, column=1, padx=5, pady=5)

# # Create labels for weight loss exercises using grid

# tk.Label(weight\_loss\_window, text="Weight Loss Exercises:").grid(row=12, column=0, columnspan=2, pady=10)

# tk.Label(weight\_loss\_window, text="1. Cardio (Running, Cycling)").grid(row=13, column=0, columnspan=2)

# tk.Label(weight\_loss\_window, text="2. HIIT Workouts").grid(row=14, column=0, columnspan=2)

# tk.Label(weight\_loss\_window, text="3. Circuit Training").grid(row=15, column=0, columnspan=2)

# def calculate\_progress():

# global steps\_goal\_entry, calories\_goal\_entry, minutes\_goal\_entry, water\_goal\_entry, sleep\_goal\_entry

# try:

# steps\_goal = int(steps\_goal\_entry.get())

# calories\_goal = int(calories\_goal\_entry.get())

# minutes\_goal = int(minutes\_goal\_entry.get())

# water\_goal = int(water\_goal\_entry.get())

# sleep\_goal = int(sleep\_goal\_entry.get())

# steps\_taken = int(steps\_taken\_entry.get())

# calories\_burned = int(calories\_burned\_entry.get())

# minutes\_active = int(minutes\_active\_entry.get())

# water\_intake = int(water\_intake\_entry.get())

# sleep\_duration = int(sleep\_duration\_entry.get())

# steps\_progress = (steps\_taken / steps\_goal) \* 100

# calories\_progress = (calories\_burned / calories\_goal) \* 100

# minutes\_progress = (minutes\_active / minutes\_goal) \* 100

# water\_progress = (water\_intake / water\_goal) \* 100

# sleep\_progress = (sleep\_duration / sleep\_goal) \* 100

# steps\_message = get\_motivational\_message(steps\_progress)

# calories\_message = get\_motivational\_message(calories\_progress)

# minutes\_message = get\_motivational\_message(minutes\_progress)

# water\_message = get\_motivational\_message(water\_progress)

# sleep\_message = get\_motivational\_message(sleep\_progress)

# messagebox.showinfo("Progress Report",

# f"Steps Progress: {steps\_progress:.2f}%\n{steps\_message}\n\n"

# f"Calories Progress: {calories\_progress:.2f}%\n{calories\_message}\n\n"

# f"Minutes Active Progress: {minutes\_progress:.2f}%\n{minutes\_message}\n\n"

# f"Water Intake Progress: {water\_progress:.2f}%\n{water\_message}\n\n"

# f"Sleep Duration Progress: {sleep\_progress:.2f}%\n{sleep\_message}")

# except ValueError:

# messagebox.showerror("Error", "Please enter valid numerical values.")

# def get\_motivational\_message(progress):

# if progress < 25:

# return "You've got this! Keep pushing forward."

# elif progress < 50:

# return "You're making progress! Keep up the good work."

# elif progress < 75:

# return "Great job! You're getting closer to your goal."

# elif progress < 100:

# return "Almost there! Stay focused and keep going."

# else:

# return "Congratulations! You've reached your goal. Time to set new ones!"

# def clear\_data():

# steps\_goal\_entry.delete(0, tk.END)

# calories\_goal\_entry.delete(0, tk.END)

# minutes\_goal\_entry.delete(0, tk.END)

# water\_goal\_entry.delete(0, tk.END)

# sleep\_goal\_entry.delete(0, tk.END)

# steps\_taken\_entry.delete(0, tk.END)

# calories\_burned\_entry.delete(0, tk.END)

# minutes\_active\_entry.delete(0, tk.END)

# water\_intake\_entry.delete(0, tk.END)

# sleep\_duration\_entry.delete(0, tk.END)

# def set\_goals():

# clear\_data()

# set\_default\_goals()

# def set\_default\_goals():

# steps\_goal\_entry.insert(0, "10000")

# calories\_goal\_entry.insert(0, "2000")

# minutes\_goal\_entry.insert(0, "30")

# water\_goal\_entry.insert(0, "8")

# sleep\_goal\_entry.insert(0, "8")

# # Create the main window

# root = tk.Tk()

# root.title("Fitness and Goal Tracker")

# root.geometry("400x400")

# root.configure(bg="#f0f0f0")

# watermark\_label = tk.Label(root, text="FITNESS TRACKER", fg="#4682B4", font=("Arial", 30, "bold italic"))

# watermark\_label.place(relx=0.5, rely=0.1, anchor="center")

# # Button to open user registration window

# registration\_button = tk.Button(root, text="User Registration", command=open\_registration\_window)

# registration\_button.place(relx=0.5, rely=0.3, anchor="center")

# # Button to open BMI calculator window

# bmi\_calculator\_button = tk.Button(root, text="BMI Calculator", command=open\_bmi\_calculator)

# bmi\_calculator\_button.place(relx=0.5, rely=0.4, anchor="center")

# # Run the main event loop

# root.mainloop()

# OUTPUT:

# 

# 

# 

# 

# 

# 