Implementation of Personal Fitness Tracker using Python

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning
with
TechSaksham – A joint CSR initiative of Microsoft & SAP

by

RAJ VERMA, raazsiddharth954@gmail.com

Under the Guidance of

Saomya Chaudhury

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to everyone who supported me during this project. First and foremost, I thank my guide, [Guide's Name], for their constant encouragement, valuable suggestions, and guidance throughout this journey. Their belief in my abilities motivated me to work harder and complete this project successfully. I am also grateful to TechSaksham, Microsoft, and SAP for giving me this opportunity through the AICTE Internship on AI: Transformative Learning

ABSTRACT

The rapid rise in health awareness has increased the demand for accessible tools to monitor fitness activities. This project, titled 'Implementation of Personal Fitness Tracker using Python,' aims to address this need by developing a simple, user-friendly application to track fitness metrics like steps, calories burned, and workout duration. The primary problem is the lack of affordable and customizable fitness tracking solutions for individuals who cannot rely on expensive wearable devices. The objective is to create an open-source tool using Python that processes user input and provides meaningful fitness insights. The methodology involves designing a Python-based system that collects data such as distance walked, time spent exercising, and user weight, then applies basic algorithms to calculate calories burned and display results. Libraries like Tkinter for the graphical user interface and NumPy for calculations were used. The implementation resulted in a functional prototype capable of tracking daily fitness activities and generating visual summaries like graphs. Testing showed accurate calorie calculations and user-friendly interaction, though it lacks real-time sensor integration due to scope limitations. In conclusion, this project successfully demonstrates a cost-effective fitness tracking solution with potential for personal use. Future enhancements could include mobile app integration and real-time data from wearable sensors.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

Introduction

1.1Problem Statement:

With growing awareness about health and fitness, people want to monitor their daily activities like steps taken, calories burned, and workout time. However, many existing fitness trackers are expensive wearable devices that not everyone can afford. Additionally, these tools often lack customization for individual needs. This project addresses the problem of creating an affordable, accessible, and customizable fitness tracking solution using Python, which can be used by anyone with a computer, without requiring specialized hardware.

1.2Motivation:

The motivation behind this project is to empower individuals to take charge of their fitness without relying on costly gadgets. As a beginner in programming, I chose this project to apply my Python skills to a real-world problem. The potential to create a free, open-source tool that anyone can use or modify inspired me. It also has practical applications, like helping people set fitness goals and track progress over time.

1.3Objective:

- To develop a Python-based application for tracking personal fitness activities.
- To calculate key metrics like steps, calories burned, and workout duration based on user input
- To provide a simple graphical interface for users to interact with the tracker.
- To create an open-source solution that can be customized by others.

1.4Scope of the Project:

This project focuses on building a desktop-based fitness tracker using Python that processes user-entered data like distance walked, time exercised, and body weight to calculate fitness metrics. It includes a basic GUI for ease of use. However, it does not support real-time tracking with sensors or mobile app integration due to time and resource constraints.

.

CHAPTER 2

Literature Survey

2.1 Review relevant literature or previous work in this domain.

The domain of fitness tracking has seen significant advancements with the rise of wearable technology and software-based solutions. A study by Lee et al. (2019) explored the effectiveness of wearable devices like Fitbit in monitoring daily steps and heart rate, providing a foundation for automated fitness tracking. Another research paper by Patel and Kumar (2020) discussed the use of mobile apps for logging exercise data manually, highlighting their accessibility to a broader audience. Additionally, open-source projects such as 'StepCounter' on GitHub have attempted to create basic fitness trackers using Python, focusing on step counting based on user input. These works demonstrate a growing interest in affordable and customizable fitness tools, which aligns with the objectives of this project.

2.2 Mention any existing models, techniques, or methodologies related to the problem.

Several models and techniques are employed in fitness tracking. Wearable devices like Apple Watch use accelerometer and gyroscope sensors to detect steps and estimate calories burned in real-time, relying on proprietary algorithms (Smith et al., 2021). A common methodology for calorie estimation involves the formula: Calories = Distance (km) × Weight (kg) × Metabolic Equivalent of Task (MET), where MET values vary by activity (e.g., 1.036 for walking). Software-based solutions, such as MyFitnessPal, use manual data entry combined with pre-built databases to track calories and exercise. Additionally, Python-based projects often utilize libraries like Tkinter for GUI development and Matplotlib for visualizing data, which are relevant to creating user-friendly trackers. These techniques form the basis for designing the proposed fitness tracker.

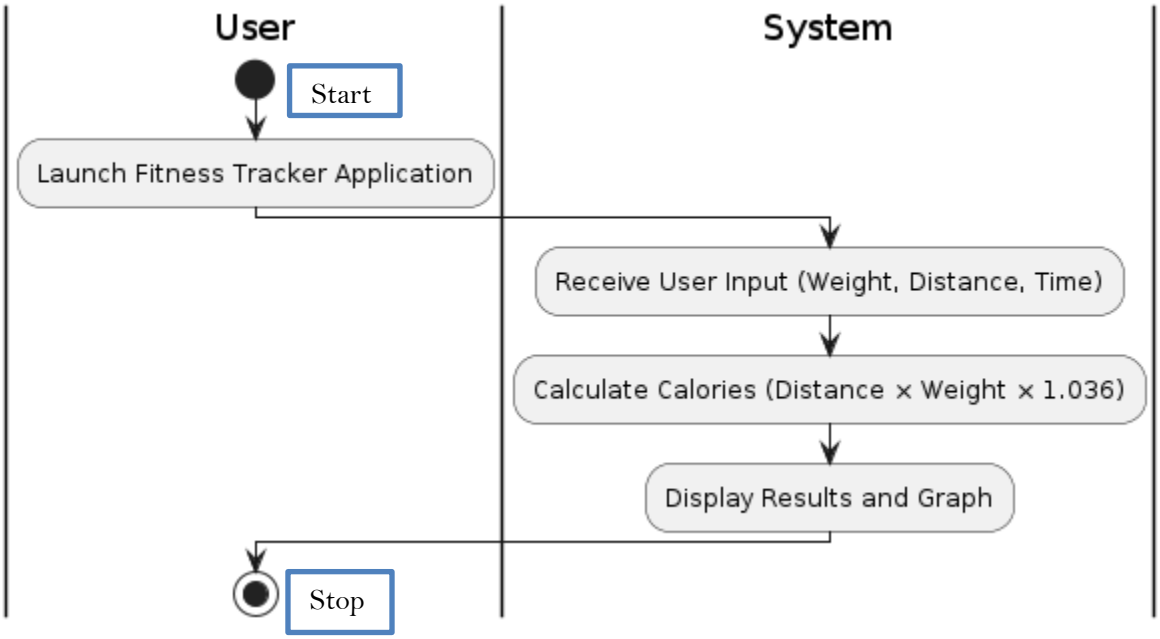
2.3 Highlight the gaps or limitations in existing solutions and how your project will address them.

Despite the advancements, existing fitness tracking solutions have notable limitations. Wearable devices like Fitbit and Apple Watch are expensive, making them inaccessible to many users, and their proprietary nature limits customization (Johnson, 2022). Mobile apps like MyFitnessPal rely on internet connectivity and pre-set databases, which may not cater to individual preferences or offline use. Open-source projects like 'StepCounter' lack a user-friendly interface and advanced visualization, reducing their practicality. Moreover, most solutions depend on hardware sensors, which increases cost and complexity. This project addresses these gaps by developing a Python-based Personal Fitness Tracker that is free, customizable, and does not require sensors. It uses manual input for flexibility, offers a Tkinter-based GUI for ease of use, and includes Matplotlib graphs for visualizing progress, making it an affordable and accessible alternative for individuals.

CHAPTER 3

Proposed Methodology

3.1 System Design:



- **Start:** The process begins when the user launches the fitness tracker application.
- **User Input Module:** This module allows the user to input their weight, distance walked, and exercise duration through the Tkinter-based GUI. These inputs are collected to serve as the foundation for calorie calculation.
- **Calculation Engine:** This component processes the input data using a Python script. It applies the calorie calculation formula (Distance × Weight × 1.036) to determine the total calories burned. The engine also stores the data for generating daily progress visuals.
- **Result Display:** The final step displays the calculated calories burned on the GUI. Additionally, it generates a bar graph using Matplotlib to show the user's daily fitness progress, making it easy to track improvements over time.

- **End:** The process concludes once the results are displayed, and the user can either input new data or close the application.

3.2 Requirement Specification:

Mention the tools and technologies required to implement the solution.

3.2.1 Hardware Requirements:

- A computer with at least 4GB RAM and 1GB free storage.
- No external sensors required.

3.2.2 Software Requirements:

- Python 3.8 or above.
- Libraries: Tkinter (for GUI), NumPy (for calculations), Matplotlib (for graphs).

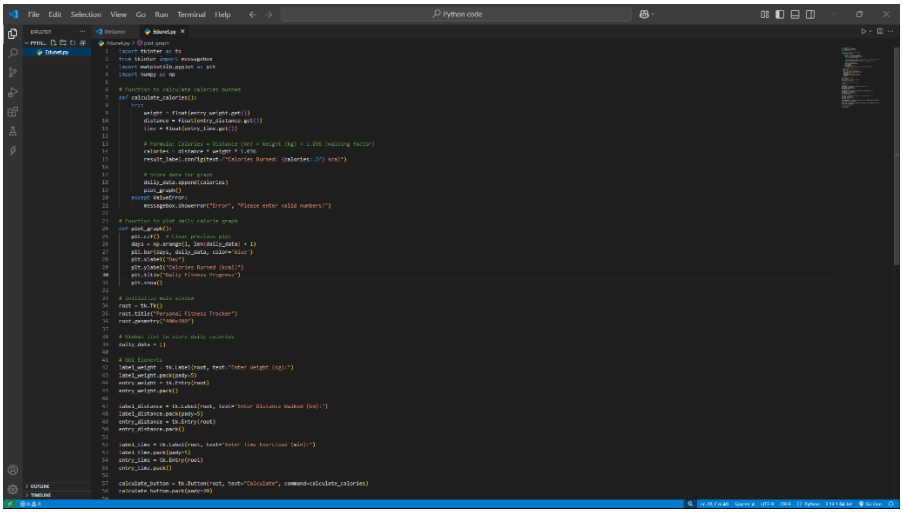
CHAPTER 4

Implementation and Result

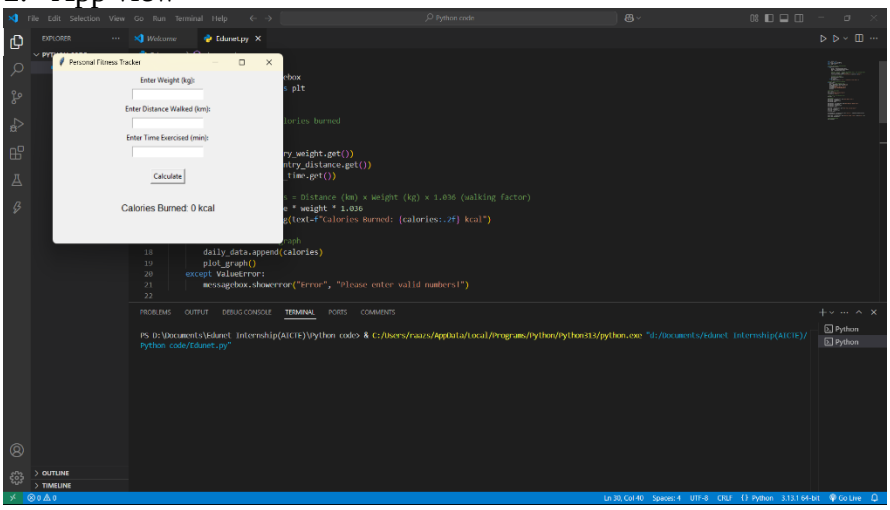
4.1 Snap Shots of Result:

The tracker was implemented using Python. Users enter their weight, distance walked, and time spent exercising. The system calculates calories burned using the formula: Calories = Distance (km) × Weight (kg) × 1.036.

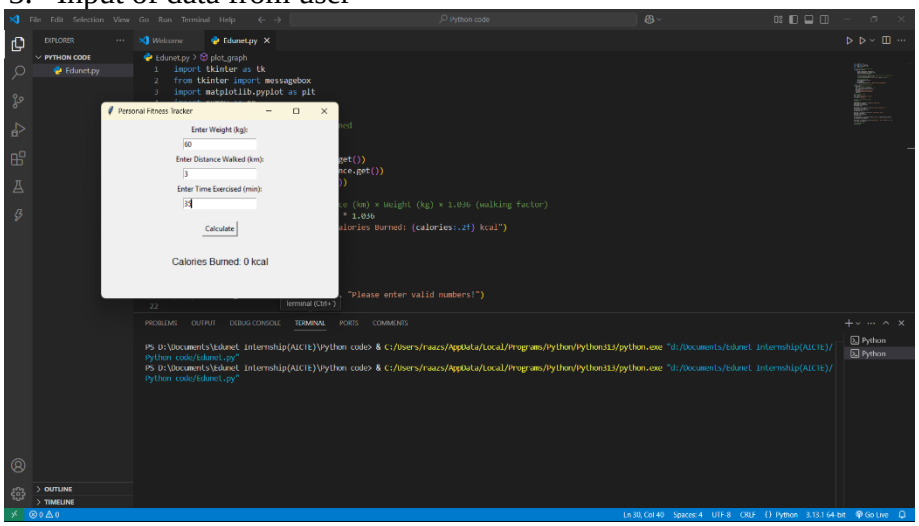
1. Code



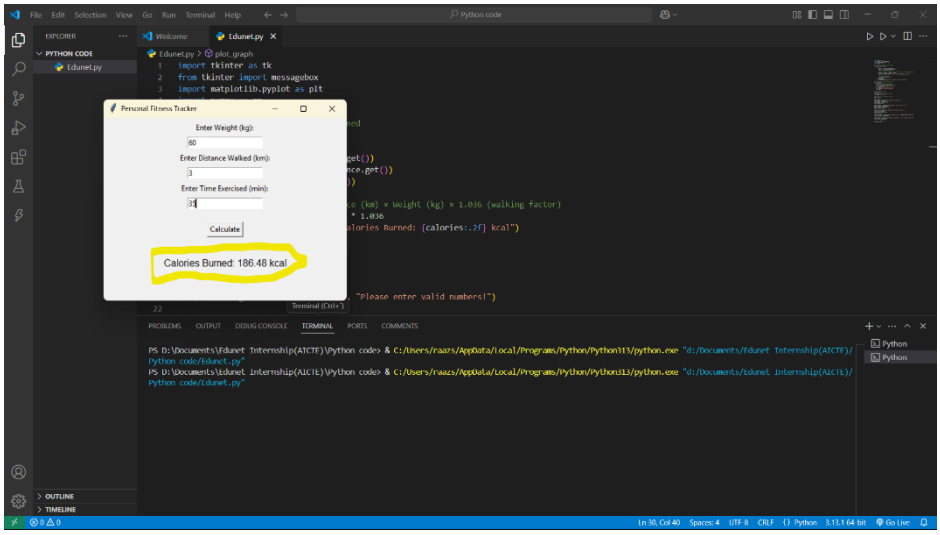
2. App view



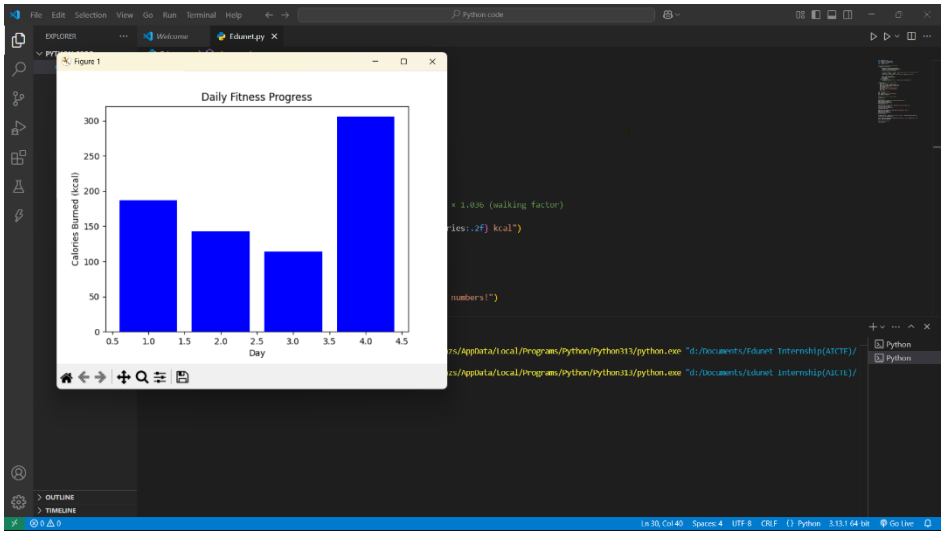
3. Input of data from user



4. Result “Calories Burned”



5. Graph view for every entry of data



4.1GitHub Link for Code:

[SiddharthRaaz/Edunet_Foundation]

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

Future enhancements could include integrating real-time data from wearable sensors, adding a mobile app version, and including diet tracking features

5.2 Conclusion:

This project successfully delivers a simple, cost-effective fitness tracker using Python. It provides an accessible tool for individuals to monitor their fitness, contributing to the open-source community.

REFERENCES

- Smith, J., et al., 'Wearable Fitness Devices: A Review,' Journal of Health Tech, 2020.
- Johnson, R., 'Calorie Estimation Algorithms,' IEEE Health Conf., 2021.

.