

# Python Command-Line Fitness Tracker:

## Project statement

### Problem Statement

Many popular fitness tracking solutions are proprietary, relying on closed systems or cloud services that limit user access and control over their own raw health and activity data. Users who prioritize privacy, customizability, and a data-driven approach often lack a lightweight, non-GUI tool that allows them to seamlessly log activities, calculate metrics, and generate detailed, personalized reports directly.

### Scope of the Project

The project scope is focused on delivering a fully functional, command-line interface (CLI) application built in Python.

1. **In-Scope:** Data logging for various activities (running, cycling, weightlifting), essential metric calculations (calories burned, pace), data persistence using SQLite or JSON, and real-time visualization of trends and summaries using Matplotlib.
2. **Out-of-Scope (Initial Phase):** Developing a Graphical User Interface (GUI), integrating with external fitness APIs (Strava, Fitbit), and implementing machine learning models for prediction or suggestion. These are defined as future enhancements.

### Target Users

The primary users of the Python Command-Line Fitness Tracker are:

- **Developers and Engineers:** Individuals comfortable operating in a terminal environment who want a scriptable tool for health tracking.
- **Data Scientists/Analysts:** Users who require easy access to their raw fitness data for complex analysis and custom reporting beyond the capabilities of standard apps.
- **Privacy-Focused Individuals:** Users who prefer to keep their personal health data managed locally and privately, without reliance on third-party servers.

### High-Level Features

The application is structured around four core functional pillars:

1. **Data Logging:** Allows easy input of diverse activities and associated metrics (distance, duration, weight, sets/reps) through simple CLI commands.
2. **Metric Calculation:** Automatically processes raw data to generate critical fitness metrics, including estimated calorie expenditure (via MET values) and performance pace.
3. **Data Persistence:** Securely stores and manages all activity logs and body metrics (weight, BMR) in a reliable data store for long-term tracking.

4. **Visualization and Reporting:** Generates insightful visual outputs (graphs, charts) showing progress over user-defined periods and activity distribution.