Project Report of Habit Tracker Application

GitHub Repository:
https://github.com/Zahra-Tohidi-eng/Habit-Tracker

1. Project Overview

This project implements a command-line based Habit Tracker application developed in Python. The goal of the application is to allow users to create habits (daily or weekly), record completions, and analyze performance using meaningful streak-based metrics.

The application follows a modular architecture and uses SQLite as a lightweight persistent storage solution. The design emphasizes separation of concerns, testability, and maintainability.

This application supports creating daily and weekly habits, recording habit completions, deleting habits, persisting all data in SQLite and calculating the habit streaks in four categories:

- Longest streak per habit
- Current active streak
- Longest streak by periodicity
- Overall longest streak

The most valuable aspects of this project are robust streak analytics that gives the clear output. In addition, modular architecture allows future expansion of the program easily. Specially, automated tests validate business logic. This program has clean database schema design and tie-handling in longest streak calculations

The analytics features transform the application from a simple checklist tool into a behavior analysis tool.

2. Technical Concept & Architecture

The system is structured into four main components:

1. **CLI Layer** – Handles user interaction and menu navigation.
2. **Domain Model (Habit class)** – Represents the core business entity.
3. **Persistence Layer (database module)** – Manages SQLite database interactions.
4. **Analytics Layer (analyze module)** – Computes streaks and statistical insights.

This separation ensures clear responsibility boundaries, reusability of analytics logic, easy unit testing and maintainable code structure.

In the persistence layer, SQLite was selected because it requires no server setup, it is suitable for small to medium applications and it integrates easily with Python's standard library.

The project was developed using PyCharm as the primary integrated development environment (IDE). The use of PyCharm contributed to improved productivity, better code readability, and more reliable testing during development.

3. Challenges & Pitfalls

Database Connection Management

The main issue that we have faced in this project was missing the database connection inside functions. This caused error while running program. The fix required clarifying connection ownership responsibilities.

Handling Duplicates in Streak Calculation

Initially, streak calculations could be affected by duplicate completion dates. This was solved by explicitly deduplicating dates before performing comparisons.

4. Possible Future Improvements

If further developed, the following features would enhance the system:

- Graphical user interface (GUI)
- REST API layer
- Export functionality (CSV/PDF)
- Habit editing (rename/change periodicity)
- Visual progress charts
- Logging system instead of print statements
- Refactoring analytics into class methods for richer domain modeling

5. Conclusion

This project demonstrates the implementation of a structured, testable, and maintainable Python application with persistent storage and non-trivial analytics logic.

The most significant achievement is the correct and extensible streak calculation system combined with a modular architecture that supports future growth. The streak calculation algorithm correctly:

- Sorts and deduplicates completion dates
- Differentiates between daily and weekly habits
- Detects streak interruptions
- Handles edge cases

This feature adds real analytical value to the product. In addition, a structured test setup significantly increases reliability and confidence in correctness.