

## **Abstract: Concept of Habit Tracking App in Python**

### **Project Overview**

This project implements a basic habit tracking application using Python (version 3.7+), designed to help users establish and maintain good habits by tracking daily and weekly tasks. The solution focuses on essential backend functionality, including habit definition, tracking, analytics, and data persistence, without any graphical interface.

### **Technical Approach**

The app is built using object-oriented programming for habit modeling and functional programming for analytics. The main modules include:

- `models.py`: Defines the `Habit` class, capturing the core logic for habits, their periodicity, and streak calculation.
- `storage.py`: Handles saving and loading user data using JSON for simple persistence.
- `analytics.py`: Implements required analytics features with functional programming constructs (`map`, `filter`, etc.).
- `cli.py`: Exposes a user-friendly command-line interface for all core features.
- `fixtures.py`: Provides 5 predefined habits (both daily and weekly) and sample data for four weeks as test fixtures.

The project comes with a comprehensive test suite using `pytest`, and all modules are documented with Python docstrings.

### **Key Features and Value**

- **Separation of Concerns**: Clean modular structure ensures maintainability and extensibility.
- **Functional Analytics**: Analytics queries leverage functional programming for performance and readability.
- **Persistence**: User habits are stored between sessions in a simple JSON file, making the app easy to use and test.
- **CLI**: An interactive command line interface for creating, completing, and analyzing habits.

- Testability: Unit tests verify habit tracking logic, analytics correctness, and data persistence reliability.

## **Challenges and Lessons Learned**

- Ensuring correct streak calculation for different periodicities (daily/weekly) required careful handling of date logic.
- Balancing simplicity with extensibility: Keeping the codebase flexible for future features while meeting current requirements.
- Implementing analytics in a purely functional style was challenging but improved code clarity and testability.

## **What Went Well**

- Modular Python package design made development and testing efficient.
- The CLI interface proved intuitive for manual and automated testing.
- Data persistence via JSON was simple yet effective for this scope.

## **Features I'm Most Proud Of**

- The streak calculation logic, which correctly handles consecutive periods for both daily and weekly habits.
- The analytics module, which allows users to easily query their habits and streaks.

## **Repository Link**

<https://github.com/MCK1NG628/Concept-of-Habit-Tracking-App-Python->