

# Habit Tracking Application: Technical Approach and “Making Of” Report

## *Overview:*

As the lead developer of the habit tracking application, the goal was to create a robust and user-friendly system that allows individuals to efficiently track, manage, and maintain their habits. The application is designed to offer users an easy way to monitor their progress, track streaks, and keep them motivated in completing their daily or weekly habits. The design is centered around simplicity and scalability, with the primary focus on ease of use and persistent data storage.

## *Technical Approach:*

The core of the application was divided into three main components:

1. **Habit Class:** This is the foundation of the app, storing the necessary information about each habit such as the habit name, periodicity (daily or weekly), current streak, and dates when the habit was marked as completed. It also includes methods for marking the habit as completed and checking for streak consistency.
2. **Habit Tracker Class:** The Habit Tracker class manages a collection of habits. It is responsible for adding new habits, viewing and deleting habits, and updating their streaks when completed. The Habit Tracker class also interacts with the data storage layer, ensuring data persistence across application sessions.
3. **Data Storage:** Initially, I opted for JSON-based storage to persist the habit data across sessions. This simple approach allowed for rapid prototyping while still ensuring that habit data was saved between sessions. In future iterations, I plan to integrate a more scalable solution like SQLite for more complex data management.

## *Key Features Built:*

- **Add Habit:** The application allows users to easily create new habits by specifying their name and periodicity. This feature was designed to be simple and intuitive, allowing the user to quickly start tracking their habits.
- **Complete Habit:** Users can mark their habits as completed, and the system updates the current streak accordingly. This was one of the key features, as it provided immediate feedback to users and kept them motivated by showing their progress.
- **Streak Tracker:** The app tracks both the current streak (days/weeks the habit has been completed) and the longest streak. This feature was particularly important for keeping users motivated and engaged.
- **Persistent Storage:** Habit data is stored in a JSON file, ensuring that users' habit data is maintained across multiple sessions. This approach provides a simple and effective solution for data storage in the initial phase.
- **Delete Habit:** The ability to delete habits was an essential feature, allowing users to clean up their habit list and remove unnecessary data.

### *Challenges & Pitfalls:*

- **Data Consistency:** One of the unexpected challenges was maintaining consistency in the habit data. While the habit information was stored in the JSON file, ensuring that the data was updated correctly after each interaction (e.g., completing a habit, deleting a habit) required careful handling of file I/O. I had to ensure the file was written to after every change, preventing potential data loss.
- **Streak Tracking:** Managing and correctly updating the streak data was more complex than initially anticipated. I had to carefully track the dates habits were completed and account for streak breaks. This required additional logic to ensure streaks were reset when a user missed a habit, which was not initially foreseen.
- **UI/UX Limitations:** Since the app uses a command-line interface (CLI) for user interaction, I found that user experience was somewhat limited. While this approach was simple and efficient for rapid development, I realized the potential benefits of a graphical user interface (GUI) in enhancing user experience.

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

- **Streak Tracker:** The ability to track both current and longest streaks was a key feature that added significant value to the application. This is what truly differentiates a habit tracker from a simple to-do list, as it keeps users motivated by visualizing their success over time.
- **Persistent Storage:** The use of JSON for habit data storage provided a straightforward solution to ensuring that data is retained across sessions. It enabled users to seamlessly continue tracking their habits, even after closing the application, without the complexity of integrating a full-fledged database in the early stages.
- **Flexible Habit Periodicity:** The flexibility in defining whether a habit is daily or weekly allows the user to customize the app to their own needs. This feature makes the app adaptable to various types of habits, whether they require daily attention or are more suited to a weekly routine.

### *Conclusion:*

The habit tracking application was successfully built with the aim of providing users with an easy-to-use platform for habit management. The key features, including streak tracking and persistent storage, added significant value to the product. However, challenges related to data consistency and streak tracking were encountered, which required additional development effort. Despite the limitations of the CLI interface, the app has shown potential for further expansion, particularly by introducing a graphical user interface (GUI) and more robust data storage solutions.

Moving forward, the next steps involve refactoring the application to use a relational database like SQLite for more complex data management and introducing a more user-friendly GUI.

### *GitHub Repository Link:*

**<https://github.com/SyedMohsinHussain/HabitTrackerProject>**