<u>Productivity Assistant Bot Documentation</u>

[Link to video showcasing bot](https://youtu.be/6BlUNlx6dA4) - https://youtu.be/6BlUNlx6dA4

Introduction

This document outlines the development and implementation of the Productivity Assistant Bot, a sophisticated tool designed to boost daily productivity through smart task management and AI-driven assistance. Created using Python and integrating cutting-edge APIs such as Anthropic, this bot is a comprehensive solution for enhancing personal efficiency by managing and tracking daily tasks systematically.

Use Case

The Productivity Assistant Bot is tailored for individuals seeking to optimise their daily task management. It provides robust support by enabling users to add, complete, and remove tasks from a daily planner efficiently. Additionally, it features a chat interface for processing natural language queries and includes functionality to export tasks to Excel for in-depth analysis, thereby catering to a broad spectrum of productivity needs.

Motivations

The inception of this project occurred at the beginning of my final year at university, a period that demanded exceptional time management and efficiency due to my academic and personal commitments. The primary motivation behind the development of this bot was to create a tool that would not only streamline my daily tasks but also accelerate my progress towards academic and personal goals. Initially, I recorded my activities without the AI assistant for six weeks, focusing on capturing baseline data which included tasks such as gym visits, library sessions, and seminar attendance. This bot was designed to enhance this baseline by facilitating more structured and optimised daily schedules.

Efficiency and Data

Data collection spanned a 12-week period, divided equally into phases with and without the AI assistant. Utilising the `pandas` library, I processed these datasets to compare productivity metrics effectively. This analysis was visualised through a comparative bar chart created with `matplotlib`, which illustrated significant shifts in activity patterns. For instance, library visits increased markedly during the period with AI assistance, demonstrating more strategic time allocation which allowed for shorter, more frequent study sessions as opposed to the

longer, infrequent sessions previously recorded. This optimisation not only bolstered my study efficiency but also enhanced my gym attendance, underscoring the bot's success in refining daily time management.

Comparison of Activity Counts
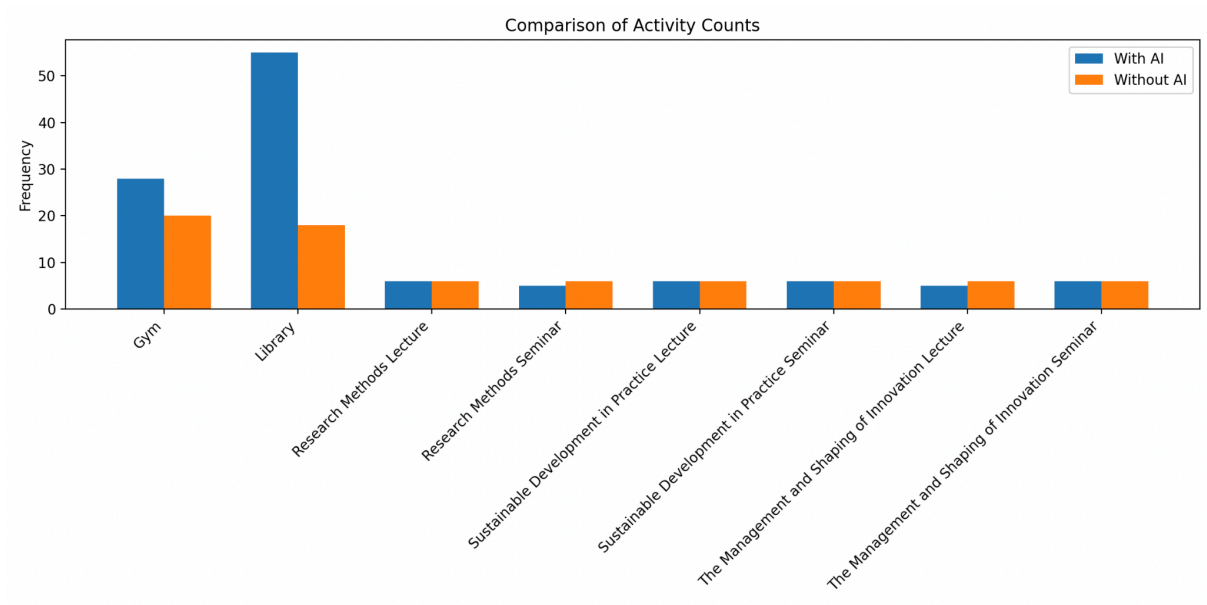
Skills Developed

Throughout this project, I developed several key skills:

- **Advanced Python Programming**: Leveraging libraries such as Tkinter for GUI development and Pandas for data manipulation.
- **API Integration**: Implementing and integrating the Anthropic API to infuse AI capabilities into the bot.
- **Concurrency in Python**: Employing threading to ensure seamless API calls and GUI updates, maintaining a responsive and user-friendly interface.
- **GUI Design**: Crafting an intuitive and aesthetically pleasing user interface that enhances user experience and interaction.

Challenges

This project presented multiple challenges, notably in integrating the Anthropic API and managing concurrent operations within the Python environment. These challenges were met with rigorous testing and iterative debugging to ensure robust performance. The real-time data synchronization between the GUI elements and the backend operations also required meticulous attention to detail and a deep understanding of event-driven programming.