**PERSONAL REFLECTION**

My reflection on the WeatherWise development and related activities

Working on the WeatherWise application has been both revealing and life-changing. It taught me to use Python while also pushing me to consider my interactions with AI tools when finding solutions for real problems. At the beginning of this work, I build a simple command-line weather assistant before upgrading it to a widget-based system with AI-type answers, charts and a friendly interface for regular use.

At first, I created an app with a menu by using pyinputplus to get user input. Because of this, I planned my program by breaking it down into key functions: get\_weather\_data, parse\_weather\_question and generate\_weather\_response. Even though it worked, the user interface was just straightforward and not creative. It was not enjoyable, as interactivity was missing.

I also looked at how I could use ipywidgets to add a dynamic notebook interface to the application. Because of this, users could now use text fields, buttons and viewing panels and didn’t need to switch between different screens. The previous version’s functions were kept and adjust to be used with event-driven UI logic. For example, I hooked the button clicks to functions that used data and generated visuals and then displayed the results right away on the output area.

AI helped out a lot both by making responses better and by steering the project’s development. I prompted students by getting them to write pseudocode, explaining the question in different ways and suggesting ways to improve modules. In fact, when I asked ChatGPT how to use ipywidgets for my weather assistant project, it provided me with a strong starting point for the interface. I decided to question the AI’s output when it wanted to plot hourly precipitation and made them show the daily total during each hour instead.

The code was managed with GitHub and more than 15 meaningful commits helped track every key task, from setting up the project to working on the UI and adding AI properties. Because of this, the code could be tested again and improved over and over. I also recorded AI exchanges so I could show how inviting feedback helped me improve on both the technical and interactive sides of the solution.

All in all, the project showed me how effective it is to mix AI’s advice with decisions made by humans. I improved my abilities in Python, connecting APIs, designing UI and reviewing AI suggestions. I learned to face complicated applications by first breaking them down and using AI to guide my progress. Now, WeatherWise works well and represents a big step for me in my AI-based development.