# Claude - Approaches-Reasoning-Journal

PROMPT:

|  |
| --- |
| // Context: I’m going to be working with VS Code and GitHub Copilot on creating a feature, from the idea and docs, specs, to the implementation in C#.  // Objective: I also want to record/save somewhere, in some kind of documentation, the whole process I’m doing, including what I think and I’m going to do and also every prompt I create and use with GHCP, so later I can showcase the process and reasoning I did. What would be the best way to save/write my whole process and reasoning including my own prompts?  // Requirements: Initially open to multiple approaches  // Format: Initially open to multiple approaches  // Constraints: I don't want to over-engineer, I need to be agile. |

For documenting your entire development process including your reasoning and GitHub Copilot prompts, here are several effective approaches:

**Documentation Formats**

**Markdown Documentation** - Create a detailed .md file in your project repository with sections like:

* Project Overview & Goals
* Design Decisions & Reasoning
* Implementation Log (chronological entries)
* Copilot Interactions (prompts and responses)
* Lessons Learned & Reflections

**Development Journal** - Use a tool like **Obsidian, Notion**, or even a simple text editor to maintain a running journal with timestamps, thoughts, and code snippets.

**Capturing Copilot Interactions**

**Manual Documentation** - Copy/paste your Copilot prompts and the generated code into your documentation. Include context about why you chose specific prompts.

**VS Code Extensions** - Look into extensions like:

* "Copilot Labs" for experiment tracking
* "Code Tour" for creating guided walkthroughs of your code
* "Git History" to track your commit progression alongside reasoning

**Screenshots/Screen Recording** - Capture key moments of your Copilot interactions, especially when the suggestions significantly influence your approach.

**Structured Approach**

Consider organizing your documentation with these sections:

1. **Initial Planning** - Requirements, architecture decisions
2. **Implementation Phases** - Break down by features/components
3. **Copilot Collaboration Log** - Specific prompts, results, and iterations
4. **Problem-Solving Sessions** - Challenges encountered and how AI assistance helped
5. **Code Evolution** - Before/after comparisons showing AI's impact

**Pro tip**: Set up a simple template or use Git commit messages strategically to maintain consistent documentation habits throughout your development process.