**Reflection**

During the development of the Calculator project, I used **Aider** extensively to implement new features and refine the existing code. Overall, Aider significantly enhanced my workflow compared to traditional coding methods.

**Most Effective Aider Techniques**

* **add** – Allowed me to add new files or updated code so that Aider could read and understand the latest changes. This made it easier to work on multiple parts of the project simultaneously.
* **undo** – Extremely useful for reverting the last edits without manually tracking changes, especially when experimenting with new features.
* **test** – Helped me automatically write and run unit tests for new features, saving a lot of manual effort.
* **explain** – Provided clear explanations for complex parts of the code, which helped me understand logic issues and guided my debugging.

These commands made implementing features like **calculation history, colored operations, and dark mode** faster, more efficient, and less error-prone than coding manually.

**Limitations Encountered**

* **API Key Limit**: The free OpenRouter key had a daily request limit, which slowed some iterations.
* **UI Adjustments**: After adding new features, the layout for buttons like History and Dark Mode did not look optimal initially. While this was not a major issue, it required some manual adjustment.
* **State Awareness**: Aider sometimes required explicit add commands to fully understand the latest state of the project files.

**Comparison to Traditional Coding Workflow**

* **Speed**: Implementing new features was noticeably faster with Aider because it already had access to the full project context.
* **Ease of Use**: Having all code and files loaded in Aider simplified explanations, debugging, and code additions.
* **Error Reduction**: Automated testing and inline suggestions reduced logical errors and unnecessary trial-and-error coding.
* **Learning and Guidance**: Unlike conventional coding, Aider provided instant guidance, explanations, and actionable suggestions, making the development process smoother.

**Suggestions for Improving Aider**

* Better **UI/Layout Awareness** to automatically handle button positioning and styling.
* Higher **API request limits** or more flexible free-tier usage to accelerate testing and iterative improvements.
* Enhanced **multi-file context handling** so Aider can analyze multiple related files simultaneously without explicit add commands.

Overall, Aider has transformed my coding workflow by making feature addition, testing, and debugging faster and more intuitive. It combines the intelligence of AI with the flexibility of traditional coding, making it an indispensable assistant for modern development tasks.