

Title: Improving a Web-Based Calculator with Aider and Gemini Pro

Goal:

To apply a dark mode theme to the user interface of a basic React calculator using the Aider tool and the Gemini Pro experimental model.

Tools & Technologies:

- **Aider:** A CLI-based AI development assistant
- **Gemini Pro Experimental (gemini-exp):** The LLM used for coding support
- **Git:** For version control and repository management
- **VS Code:** Code editor
- **Node.js / npm:** For running and managing the React app
- **React:** Frontend JavaScript framework
- **PowerShell:** CLI used for setup and execution
- **Playwright (optional):** Used for content scraping in Aider if needed

Implementation Steps:

1. Setup

- Installed Git and ensured it's added to the system PATH
- Created a Python virtual environment
- Installed Aider and its required packages in the virtual environment

2. API Configuration

- Retrieved a Gemini API key
- Configured the key using `setx GEMINI_API_KEY "<key>"` and verified it in Aider

3. Project Setup

- Cloned the calculator project from GitHub: <https://github.com/andrewagain/calculator.git>
- Navigated into the project folder and ran `npm install` to set up dependencies

4. Applying the Dark Theme with Aider

- Added CSS files (App.css, Display.css, Button.css) to Aider
- Prompted Aider to implement a dark mode design
- Reviewed and approved the changes suggested by Aider

5. Testing

- Ran the app locally with `npm start`
- Visually confirmed that the dark theme was applied correctly

Challenges Faced:

- PowerShell's script execution policy initially blocked the virtual environment
- Git was not recognized until the PATH was correctly configured
- Aider experienced rate limiting from the API during prolonged sessions

Result:

The calculator's UI was successfully updated to a dark theme with clean, responsive styling. Aider's support streamlined the code editing and update process, while Git ensured changes were safely tracked and reversible.