

# Final Report

# Robust and Improved Thoughtful AI

**Team Members:** Alina Sainju, Hannah Yoo, Kyle Houston, Monica Zhang

---

Thoughtful AI is an AI-powered writing tool that aims to help writers to communicate better without off-loading their own thinking. With the recent boom of AI, LLM chat interfaces give one-shot responses that do the thinking for the writer. Thoughtful AI's biggest aim is to make you, the writer, an active participant in your own writing process.

## Vision Statement

Our core objective for this academic year is to enhance the robustness, usability, and integration capabilities of the Thoughtful AI system for the Calvin University community as well as external research collaborators. We aim to achieve this goal through:

- Comprehensive system testing
- Backend pipeline improvement
- Integration of sample external tools as a starting point for future collaboration

By the end of this milestone, we envision a more reliable system that can serve as a valuable resource for the Calvin community as well as a platform that supports other AI researchers' visions. We believe that the Thoughtful AI project represents a meaningful step toward responsible stewardship of artificial intelligence.

---

## Background

Current AI tools either generate entire articles which threaten cognitive skills or only handle mechanics like grammar that are too simplistic for complex tasks. Our tool has 4 different types of AI suggestions, generating suggestions that are based on the context of one's own writing:

1. Reader Perspective Analysis: poses 3 questions from a reader's viewpoint
2. Proposal Advice: suggests strategic next moves
3. Example Sentences: provides what next sentence completions could look like
4. Rewording: provides rewording suggestions for selected text

Here are some relevant, influential prior works that were done.

- <https://arxiv.org/abs/2312.06024>

This paper talks about how current tools either try to generate entire articles (which are threatening to human cognitive skills) or only handle mechanical aspects like grammar (too simplistic for complex journalism tasks). There's no middle ground for sophisticated

editorial support. The problem suggested in this paper resonates with our project goal's baseline. Editors noted issues with the AI-generated rewrites, like paraphrasing secondhand quotes incorrectly or suggesting changes without evidence in the story. The system can identify problems but struggles with appropriate solutions. There's significant skepticism about AI tools for writing these days, and this "missing middle ground" for sophisticated editorial support directly aligns with our project's goal to develop AI that enhances rather than replaces human thinking. We need to ensure Thoughtful AI provides reliable, evidence-based assistance that supports rather than undermines user judgment.

- [https://cplusj2024.github.io/papers/CJ\\_2024\\_paper\\_23.pdf](https://cplusj2024.github.io/papers/CJ_2024_paper_23.pdf)

Users' expectations are shaped by existing conversational AI paradigms. Understanding user expectations is crucial for our usability goals, if Thoughtful AI is to serve the Calvin community effectively, it must align with how users actually want to interact with AI tools. This research informs our integration strategy: by understanding what makes AI assistants trustworthy and useful, we can better design our external tool integrations to support researchers' needs while maintaining the collaborative, thoughtful approach our system aims to provide.

Prior work identifies a gap but doesn't solve the measurement problem of how to evaluate whether AI assistance enhances vs. replaces cognition. As this is an ongoing research work alongside this senior project, our current phase focuses more on strengthening our technical infrastructure and testing frameworks to support more rigorous evaluation.

---

## Normative and Ethical Considerations

**Cultural Appropriateness:** We noticed that the younger generation uses Google Docs more than MS Word. In order to adapt to this cultural shift, we have developed a Google Doc Add-In with the hope of engaging more new users.

**Transparency:** Poor error communication, or no error communication at all, frustrates users and limits growth. To address this, we've been implementing error notifications that inform developers when issues occur while users are using our platform so that we can tackle them immediately.

**Social:** Current AI interfaces like chatbots lead to users delegating the work of thinking to the AI. This makes it a challenge for users to grow their critical thinking skills. It also keeps users from reaching out to peers and their social network for help due to the AI overreliance. Our platform aims to intervene and include the user in the thinking process through an additional interface layer that mediates human-AI interaction in a productive manner.

**Stewardship:** AI usage carries environmental costs. We track LLM usage through data analytics to ensure efficient resource consumption.

**Delightful Harmony (Aesthetics):** When form doesn't match function, users leave. We're refining our interface styling to make it intuitive and easy to use. We wanted the interface to feel clean and pleasant, not overwhelming, which drove our UI redesign as we moved into Google Doc

**Justice:** Writing assistance should be accessible to all users. While our platform currently works best on desktop, we're working toward mobile responsiveness to ensure equitable access across devices. On top of that in order to meet user needs, we have integrated our tool in Google Docs for wider accessibility.

**Caring:** We built this application to navigate AI usage for writers thoughtfully, ensuring it enhances rather than diminishes human capability. The tool should feel like a supportive writing partner, not a judge, so instead of rewriting your text automatically, it offers suggestions for you to choose from.

**Trust:** We use Posthog to monitor the tool's performance and make sure it behaves consistently.

---

## Product Success Criteria

### *Criterion 1: System Testing and Quality Assurance*

**Timeline:** End of Fall Semester

- **Minimum Success Threshold:** Identify and fix all currently visible bugs and known system vulnerabilities
- **Stretch Goal:** Implement robust testing suite comprehensive enough to confidently present the system to the Calvin community
- **End Of Year Result:** We have successfully integrated Posthog through which we can track user behavior, system errors and LLM analytics as well as evaluation.

### *Criterion 2: Backend Pipeline Enhancement*

**Timeline:** End of Fall Semester

- **Minimum Success Threshold:** Evaluate LLM responses using best practices
- **Stretch Goal:** Implement new LLM prompting based on evaluation
- **End of Year Result:** We added a new suggestion type "Rewording" based on user need analysis and have also updated our backend LLM model to a more robust and newer version while maintaining costs.

### *Criterion 3: External Tool Integration and Collaboration*

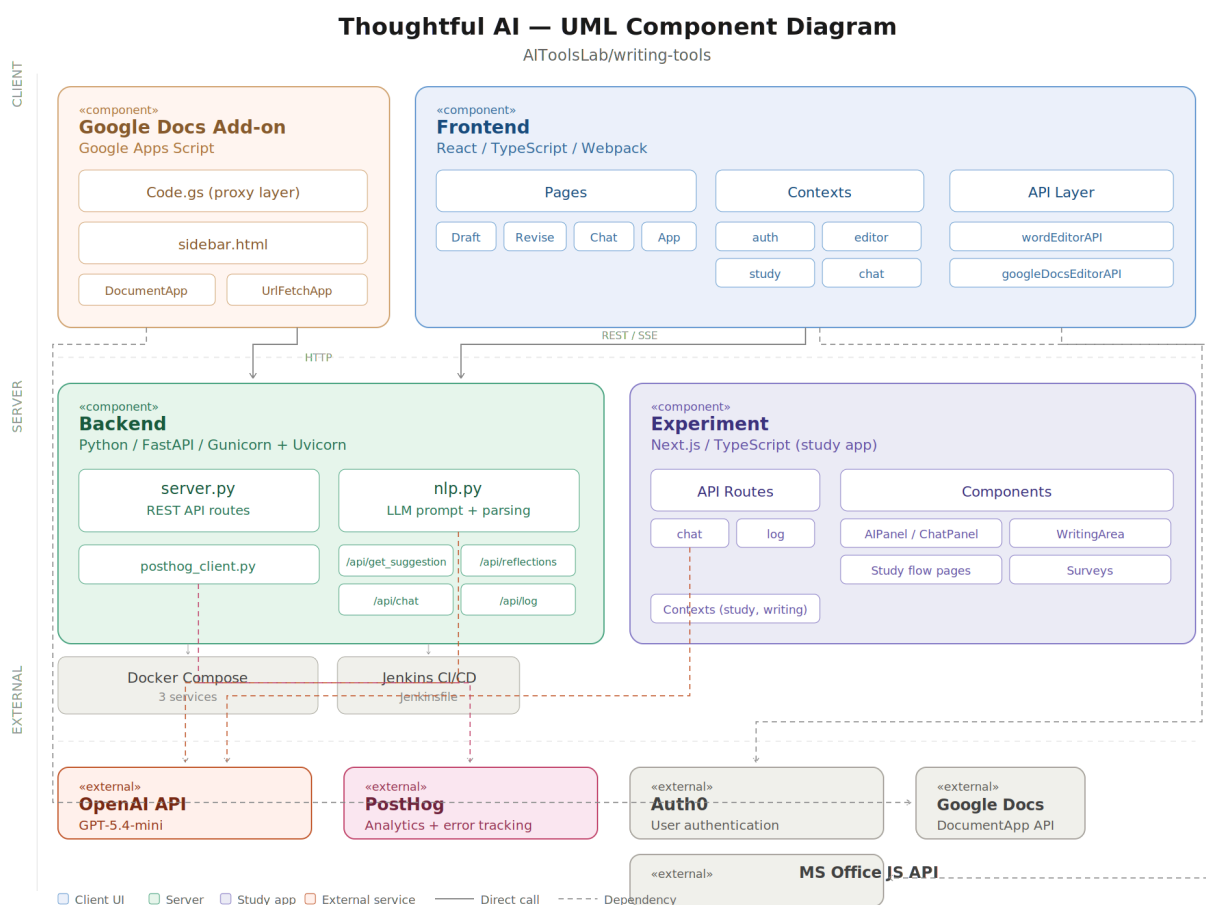
**Timeline:** Spring Semester

- **Minimum Success Threshold:** Integrate a sample external writing tool within our system (Writing Prototype or InkSync)
- **Stretch Goal:** Document and create clean API that we can present to actual collaborators
- **End Of Year Result:** We pivoted from this project and focused on integrating Google Docs Add-In since we saw clearer outcomes and value.

## Approach and Implementation

### System Architecture

- Our system architecture uses Python for custom backend and TS/React for frontend.



### Development Process

### Time Commitment

- Individual Contribution Hours: 5-6 hours per week per team member
- Total Team Hours: 20-24 hours weekly across four team members

### Communication Channels

- MS Teams and WhatsApp

### Task Management

- GitHub Projects for planning, tracking and managing development workflow
- Trello Board for non-development related tasks
- Google Docs for meeting minute notes and storage of senior project documents

### Meeting Schedule

We conducted weekly team coordination meetings as well as faculty consultation meetings.

## Challenges Faced

- When it came down to using coding agents alongside, many of us struggled to properly use it in ways that would be effective. Below, we came up with a few practices we could keep in mind.
- Many of us struggled with debugging or getting stuck on a particular problem, we've now decided to be more vulnerable about mistakes and share with others to hopefully help the other team members not make the same mistake and also to help ourselves identify and solve the problem more clearly.
- One of the problems our team encountered was with the Google Docs add-on, the plan was to conduct internal group testing by creating a shared Google Drive before a public release. However, we ran into a fundamental limitation of Google Apps Script: each script is tied to the individual Google account that created it. Other team members cannot access, test, or run someone else's Apps Script project before it is on the Google marketplace. This means internal beta testing within the team is blocked. There is no straightforward way to share an undeployed Apps Script for other developers to use and test out. Only after the add-in is officially deployed to the Google Marketplace or deployed as a shared web app can it be shared and used by other people. The resolution that we've settled on is to skip the internal testing phase and move directly toward a deployment on the marketplace, at which point the add-in can be shared with the team and broader users.

### *Software Development Practices*

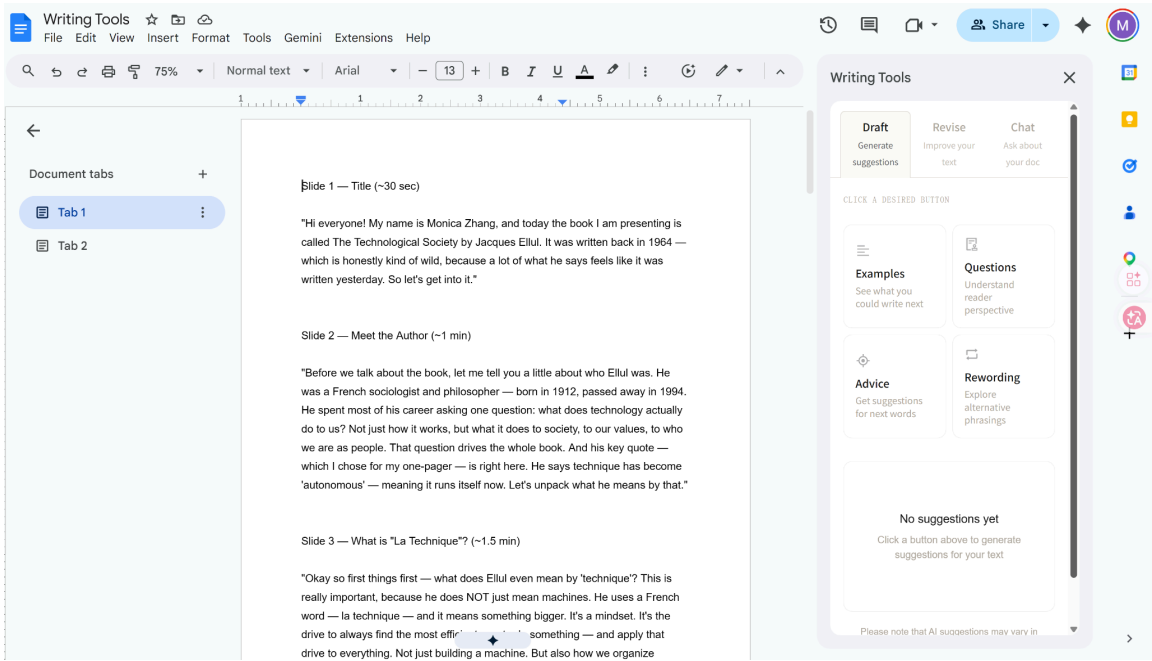
- While practicing software development skills, here are some AI coding practices to keep in mind that we went over as a team together.
  - Avoid over-filling context window ("context poisoning" / degradation)
  - Recognize when AI (or yourself) doesn't understand the problem
  - Dump out a summary of what you're trying to do to a file, review, pass it to a new agent session
  - Understand the problem (avoid the "unknown unknowns").
  - Review the AI's code. Example prompt for getting it to respond to your comments:
- When putting up a PR on GitHub, make sure comments/descriptions are included for others to review for a smoother merge without confusion.

---

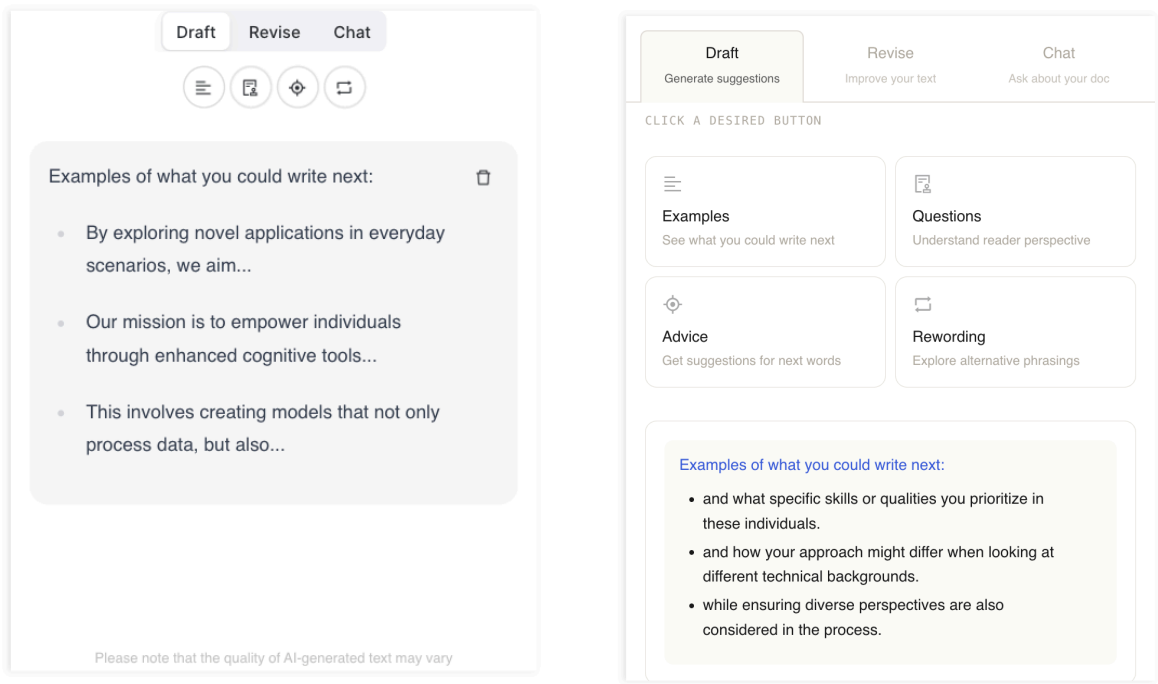
## Results

- Evaluation:
  - For the frontend testing, we made visual regression tests for the demo page and user interaction tests for the draft component. More components still need tests for comprehensive testing to be accomplished.
  - For the backend LLM pipeline, we added a new system to track LLM evaluations and have upgraded the backend LLM model.
  - We have made a good number of UI changes to the system.
  - We have integrated Google Docs Add-In.
- Possible future work:
  - For LLM evaluation work, we have a draft Langfuse PR for inspiration.
  - Guardrails and audience definition features in the Revise tab could be a part of Chat too. Currently those features do not work for Chat yet.
  - Deploy our app to Google Workspace Marketplace.
  - Add Google Doc redirect button in main page (similar to Microsoft Marketplace redirect button)
  - Clean up the main website page to be cleaner and more user friendly.
- Demo Video Link :  
<https://drive.google.com/file/d/1HXvF5W4KPYuHCOmWXpmb3AlolwMHP48K/view?resourcekey>
- Senior Project Presentation [Slides](#)
- Demo screenshots below:

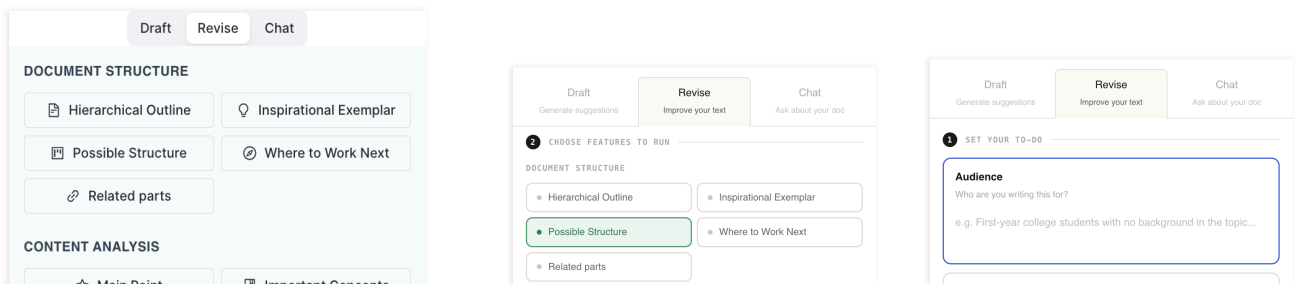
Google Doc Integration:



UI Change on Draft Page:

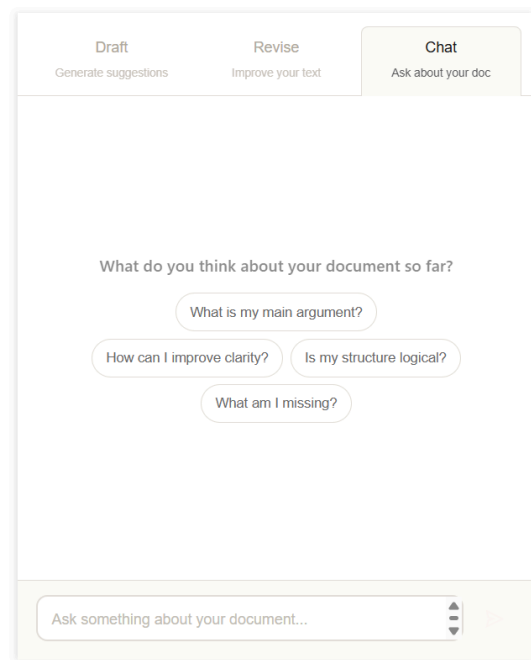
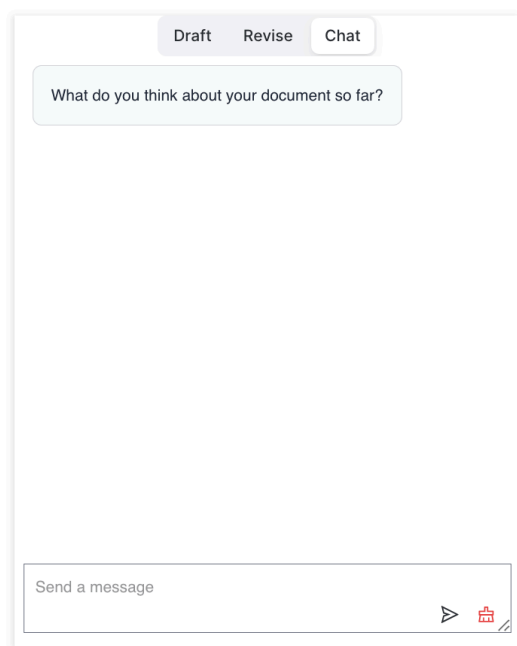


UI Change on Revise Page:



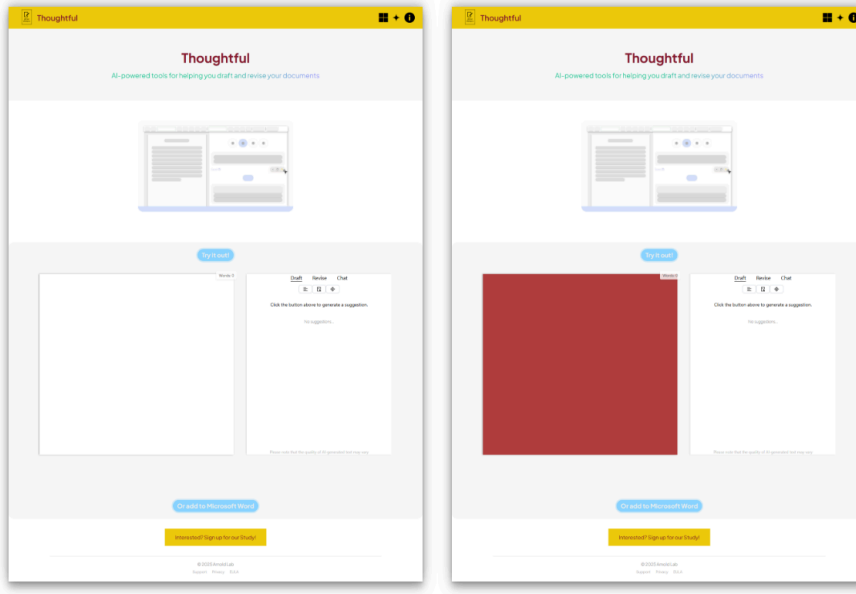


## UI Change on Chat Page:



## Visual Regression Testing with Playwright

Diff Actual Expected Side by side Slider



## Monitoring and evaluating LLM outputs with Langfuse

Langfuse v3.130.0

toy-model Hobby / toy-n

Trace 749a5a20623b6814bfc857a811bcb809

Search 🔍 ⚙️ ⏬ Timeline

generate\_suggestions 5.56s Relevance: 1.00 {}

generate\_suggestions 5.56s

2025-11-11 10:20:18.384

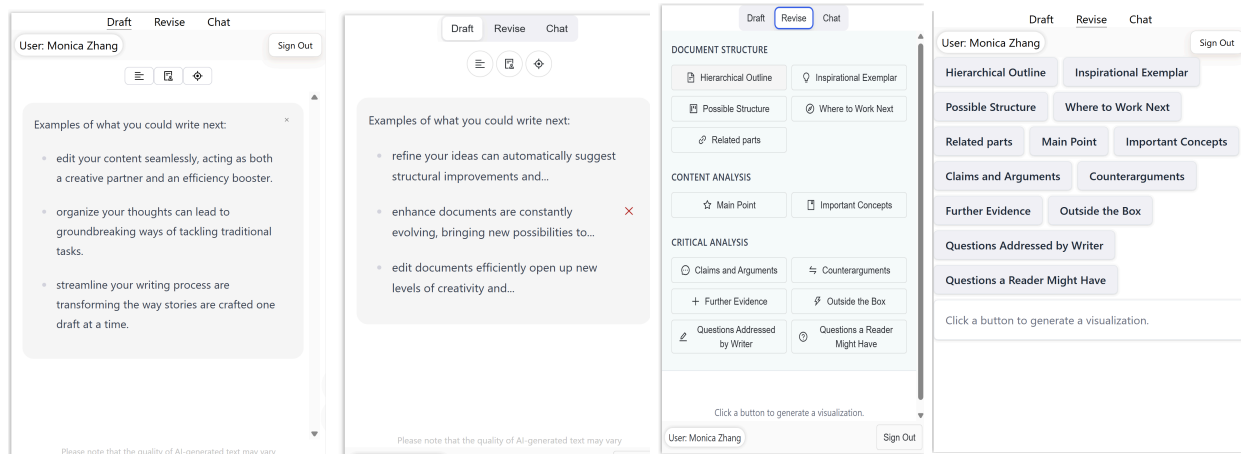
Env: default Latency: 5.56s

Preview Log View Formatted JSON

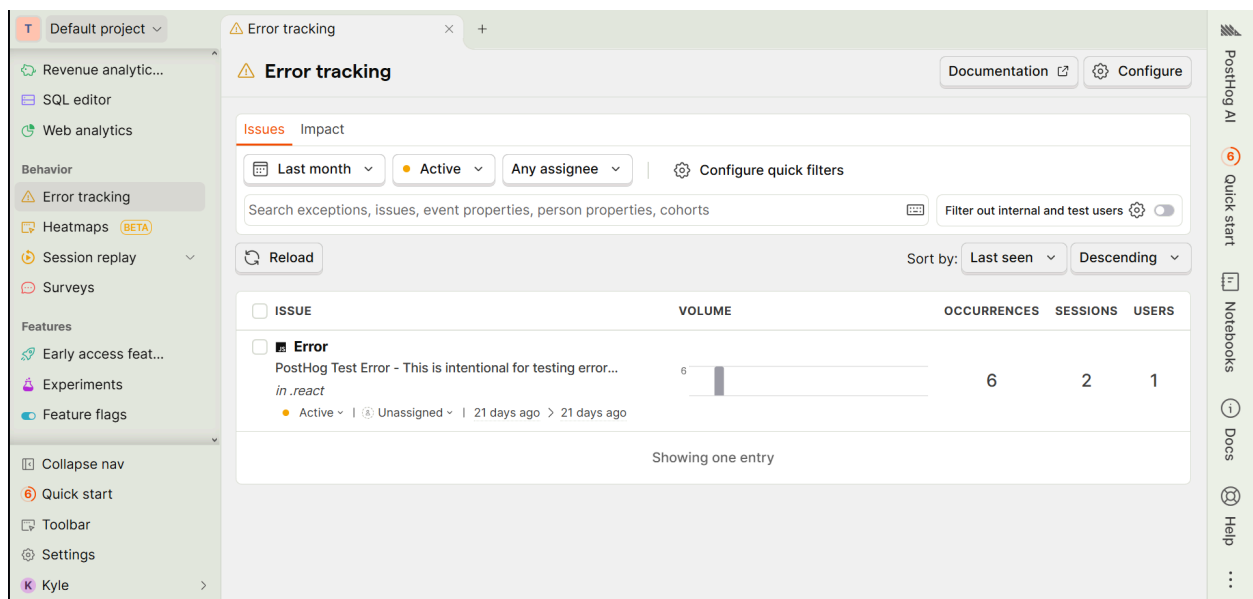
Output

| Path            | Value                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| suggestion_type | "continue"                                                                                                                                                                                                                                                                                                                                                                                    |
| suggestions     | ["Here are three possible next sentences, each taking the narrative in a different direction:", "1. One I chose for myself, and the other was chosen for me. (Hints at a story about identity and personal choice.)", "2. This isn't as strange as it sounds; I assure you there's a perfectly logical explanation. (Promises a rational explanation and invites the reader to learn more.)"] |
| 0               | "Here are three possible next sentences, each taking the narrative in a different direction:"                                                                                                                                                                                                                                                                                                 |

*CSS changes and Refactoring (Outdated images. These were changes we made in our Fall semester. The above pictures reflect the most recent UI view.)*



## Error tracking with Posthog



## LLM Analytics with Posthog

LLM analytics

Docu

Explore end-to-end traces of your LLM interactions.

Dashboard

Traces

Reviews

Generations

Users

Errors

Tools

Sentiment

BETA

Sessions

Last 7 days

+ Property filters

Saved filters

Filter out internal a

Reload

| ID          | TRACE NAME | PERSON           | SENTIMENT | TOOLS | ERRORS | LATENCY | TOKEN USAGE        | COST     |
|-------------|------------|------------------|-----------|-------|--------|---------|--------------------|----------|
| 2025...e48c | -          | B backend-server |           | -     | -      | 2.03 s  | 117 → 72 (Σ 189)   | \$0.001  |
| 059e...23e3 | -          | B backend-server |           | -     | -      | 1.14 s  | 0 → 0 (Σ 0)        | \$0.00   |
| 7930...dad3 | -          | B backend-server |           | -     | -      | 2.28 s  | 5281 → 23 (Σ 5304) | \$0.0134 |

Conclusion

Alina developed and integrated visual regression testing for the demo page of our app to automatically detect unintended UI changes. This implementation helps catch visual bugs early in the development cycle by comparing screenshots of the current build against baseline images. Alina has added some tests for user interaction flows of the application. Alina has worked on resolving UX bugs and picking up issues on GitHub to fix system bugs. On top of this, Alina has been active in the research side of the project that will help understand how users perceive and use different types of suggestions.

Hannah researched how to monitor and evaluate AI model outputs through open-source platform Langfuse. She built a toy model to test Langfuse's usability which will allow her to observe how Langfuse works in a controlled environment before committing to full integration into our system. This project is on a pause currently since we decided to stick with the features that Posthog already provides but there is space for exploration with Langfuse still.

Monica has researched UI component libraries and made UI changes for draft and revise tabs using the new component library. She has also worked on the Google Docs Add-In.

Kyle has integrated Posthog, an open source analytics platform, for error tracking and LLM analytics in our system. Error tracking lets us monitor, investigate, and resolve any issues our users encounter while using the application. LLM analytics helps us to view the cost analysis of our LLM usage. He has also worked on updating the LLM model as well as adding a smoke test to confirm builds run.