

Sprint 1 Reflection Report – Team UNT's Best

1. Accomplishments

Our team was able to deliver all the planned deliverables during Sprint 1 and attain a fully functional demo of Polish, the AI integrated document editor. Some of the major achievements are:

- Requirements & Design: Finalized requirements (R1–R10, NF1–NF6) and completed all major design diagrams (architecture, ER, class, information architecture, UI wireframes).
- Frontend Prototype: Built the React and TypeScript single page app featuring file upload, mock AI chat system, editor canvas, and export modal.
- Architecture Integration: Designed Node.js API structure aligned with future Azure services (OpenAI, Blob, SQL, AD B2C).
- Testing and Accessibility: Conducted UI and performance tests (avg 1.3 s load, < 3 s AI response); verified WCAG 2.1 keyboard navigation and contrast compliance.
- Documentation Package: Delivered updated Requirements Doc, Design Doc, Change Doc, and Test Plan with full traceability and feedback corrections.

2. Use of AI Tools

AI tools supported both development and documentation throughout Sprint 1:

- ChatGPT: Used for brainstorming architectural alternatives, generating function stubs, and rephrasing documentation sections for clarity.
- GitHub Copilot: Assisted in auto completing React components and improving TypeScript syntax.
- AI Testing Support: ChatGPT simulated prompt/response cycles for mock APIs before real Azure OpenAI integration.

Artificial intelligence support did not displace original work but made the work process faster. Every recommendation was approved and confirmed by the members of the team to maintain academic integrity and accuracy.

3. What Worked Well

- Strong Collaboration: Weekly meetings and Trello tracking kept everyone aligned and on schedule.
- Clear Role Distribution: Arnav (Requirements and Design), Waleed (Test Plan and GitHub), Mohamed (Reflection and AI Docs and Demo), Matthew (Repo Structure and UI Integration).

- Agile Workflow: Short iterations with review checkpoints allowed rapid UI and architecture progress.
- Effective Communication: Constant Slack updates and pair debugging sessions resolved frontend/backend conflicts quickly.
- Early Prototype Validation: Demonstrated end to end AI workflow locally, proving system design feasibility.

4. Areas for Improvement

- Version Control Consistency: Some commits were pushed late; for Sprint 2 we'll enforce branch naming and daily rebase rules.
- Backend Testing: Unit tests for API routes are still limited to mock data; we plan to expand with real Azure OpenAI endpoints.
- UI Responsiveness: Minor animation timing and Firefox latency issues need optimization.
- Documentation Overhead: Writing and merging multiple reports took time, will use shared OneDrive for real time edits in Sprint 2.

5. Focus for Sprint 2

- Integrate live Azure OpenAI API for real AI editing.
- Implement autosave and version control with Cosmos DB.
- Connect Azure Blob Storage for secure file management.
- Add template generation and structure adjustment features.
- Begin unit and E2E testing using Playwright and Cypress.
- Maintain WCAG 2.1 accessibility and expand performance monitoring via Application Insights.