**Amazone (Team 018-1)**

**TAs: Wyatt & Kanaka**

**CSCI 3008-010**

**October 29, 2021**

**Project Milestone 4**

**GitHub Link:** https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-018-01.git

## Revised List of Features:

AI Generate Prompts
- Use previous content in the document (sentences, paragraphs, etc.) to use an AI to generate further content

AI style transfer
- Gives the user an option to change the current style/diction of their writing into a different style. For example, a user could write something in a more commonly readable style and have the AI transform their writing to match the style of a legal document.

Word processing features
- Simple word processing functionality, similar to other rich text editors. Will give the consumer the ability to change document formatting, font sizes, etc.

Document stats
- Basic analytics about the document such as readability, word count, speaking time, etc.

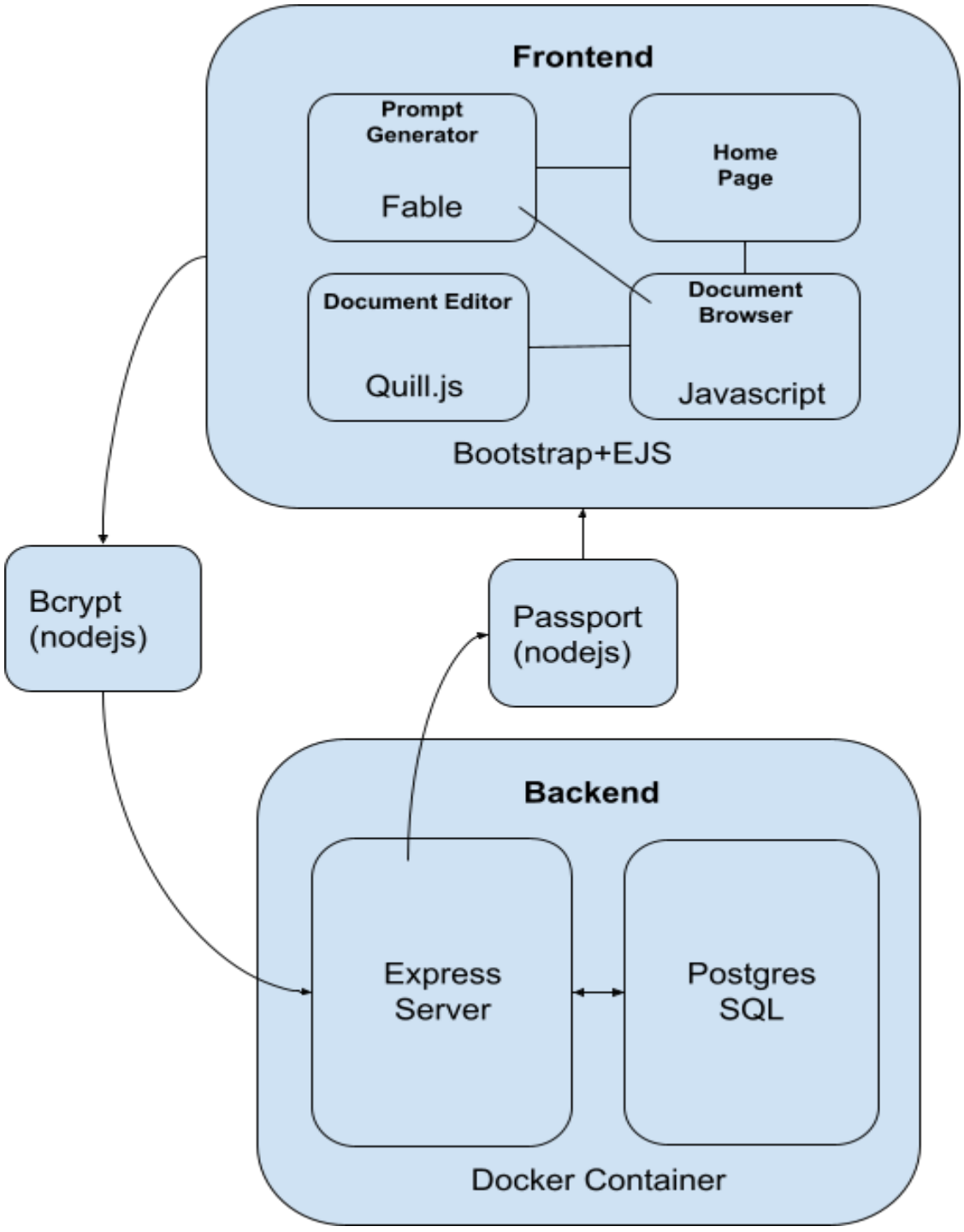
Spell check / Grammar check
- Checks the grammar and spelling of the loaded document (perhaps using LanguageTool or just the in-browser checking capabilities).

Page formatting
- Planning to implement columns and separate pages instead of one giant text box to enable users to style documents as they please

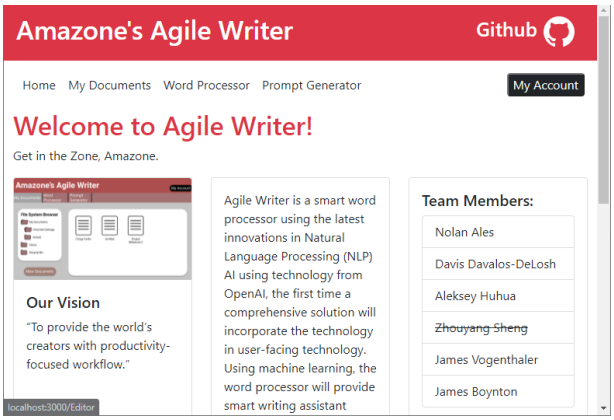
Document statistics
- Planning to implement features that will keep track of user statistics such as word count, words generated by AI, creation and modified date/time

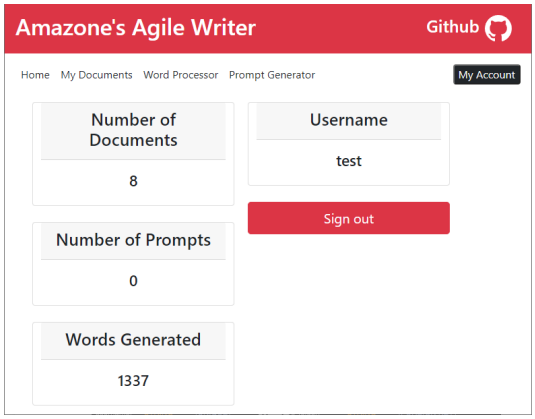
~~Autocompletion~~
- If we have time, we wish to implement auto-completion by AI, but it currently doesn't seem possible with the AI we are using. The AI may be limited to just prompt generation currently.

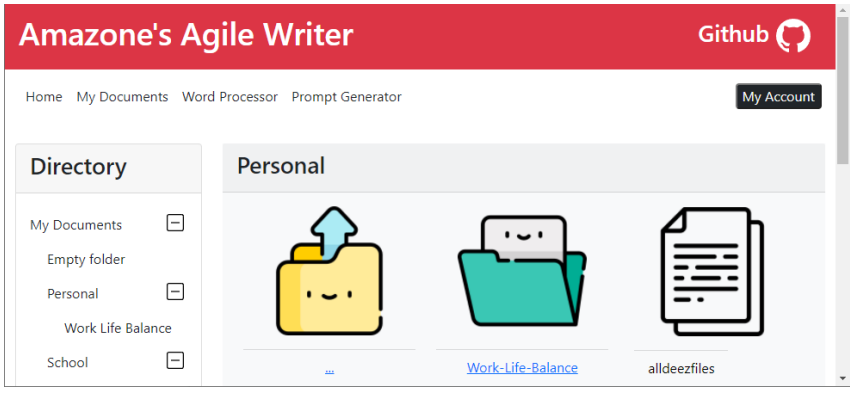
**Architecture Diagram:**

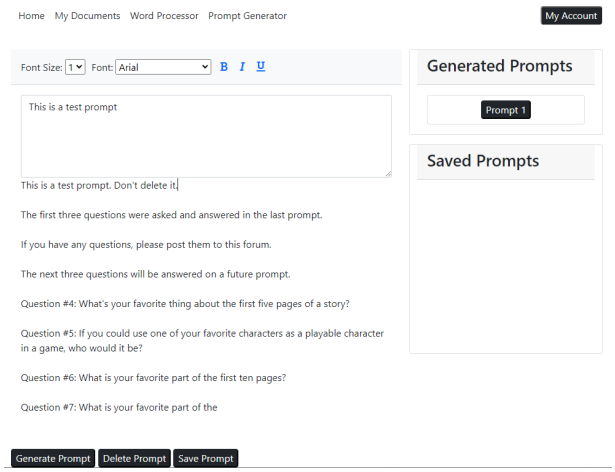


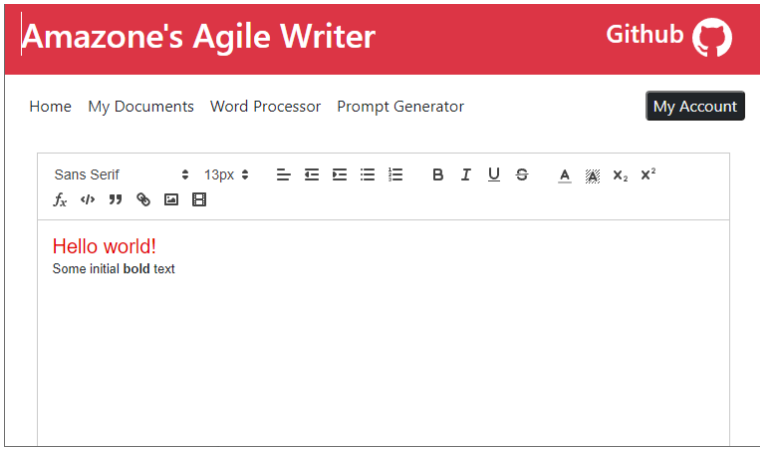
**Front End design:** Our front end design is pretty minimalistic. We have user login/registration, a user account page which displays user statistics (word count, document count, etc), a file management window, an editor, and a page for prompt generation. Users can only access the home page if not logged in, and will be redirected to the login page when trying to access any other page. Once logged in, their information will be loaded into the session, and they will be able to view the other tabs with their relevant documents and statistics. The “My Documents” tab shows users the documents they have saved, and allows the organization and creation of their documents. The “Word Processor” tab is opened through clicking on any of their documents, or can be accessed at any time to create a new document. The prompt generation page is accessible whenever the user is logged in.

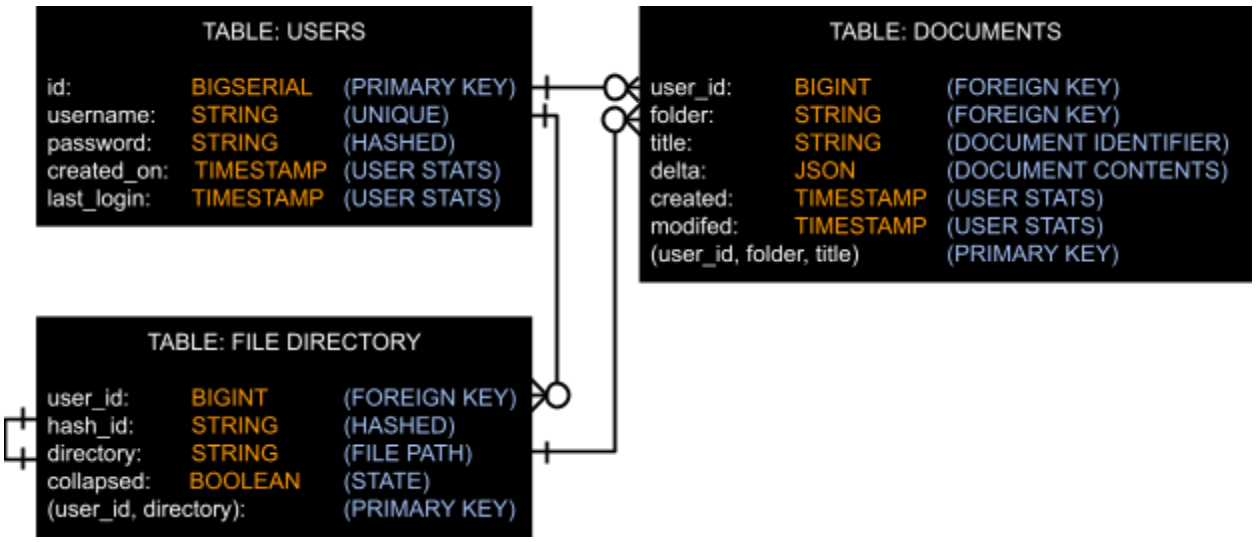



**Web Service Design:** The only web service we're using is the Mesh Transformer JAX library and API, which in turn uses the GPT-J-6B pretrained model underneath. This is how we're implementing all of our "smart" text generation, including in editor as well as the separate prompt generation screen. (Note: This could very well be changed in the upcoming weeks.)

**Database design:**




The user ID is the only data point that is serialized, and the document/file system tables uses the user ID as well as their own data points to ensure uniqueness. This allows separate users to have the same folder and file names.

The user table keeps track of usernames, passwords, and other user statistics.

The file system table keeps track of how users organize their files as well as remembering the state of the file system, such as which folders are collapsed and how folders are ordered (not implemented yet). The directory and hash ID data points refer to the same piece of data. The document's data point keeps track of everything related to files. Particularly, the delta datapoint can store up to 1GB of user text and formatting.

**Challenges:**

> **Challenge 1:** The AI that we are using takes several seconds (up to a few minutes) to generate text, which may hamper our plans on implementing any sort of auto-completion using this AI. We may have to look outward towards using additional AI services to implement this feature.
>
> **Challenge 2:** Time management and forward planning. We all had very little experience in developing in an AGILE type format, and have struggled with keeping

the Jira board up to date and in-line with current progress. We have also struggled to make our meetings meaningful and have ended up overlapping in some areas of code.

**Challenge 3:** Division of labor. Kind of overlaps with the challenge above. This was especially challenging because we had a group member drop out in the middle of the development cycle, and their tasks were never able to be completed by any of the other team members.

**Backup/risk mitigation plan:**

There isn't much we can do for a backup plan currently as much of the work has already been completed. However, as stretch goals, we'd like to look into other AI's we could use to augment our application, and possibly have one or two sessions where we rethink our team organization and planning. This would definitely aid in reducing the challenges above to more manageable levels.

**Individual Contributions:**

**James B:** Implemented user registration and login on the website with bcrypt and passport, and created the associated pages and worked with James V on the database schema. Implemented and integrated bcrypt and passport to enable user registration and login features. Also designed user login, registration, and account pages as well as the basic layout for the filesystem.

**James V:** Implemented Quill.js for rich text editing and also worked with James B on the database schema. Created and integrated the frontend file system with the postgres server enabling document loading. Also migrated the project from basic html to being hosted through docker, express, and node js using ejs and partials. Currently working on the ability to save and format documents.

**Davis:** Is handling implementation and integration of the program's "smart" features. Has researched and laid out the foundation for integrating external AI for use within the program. Currently the AI is used in generating prompts to enable user creativity, and there are efforts being made toward implementing user suggestions while typing documents.

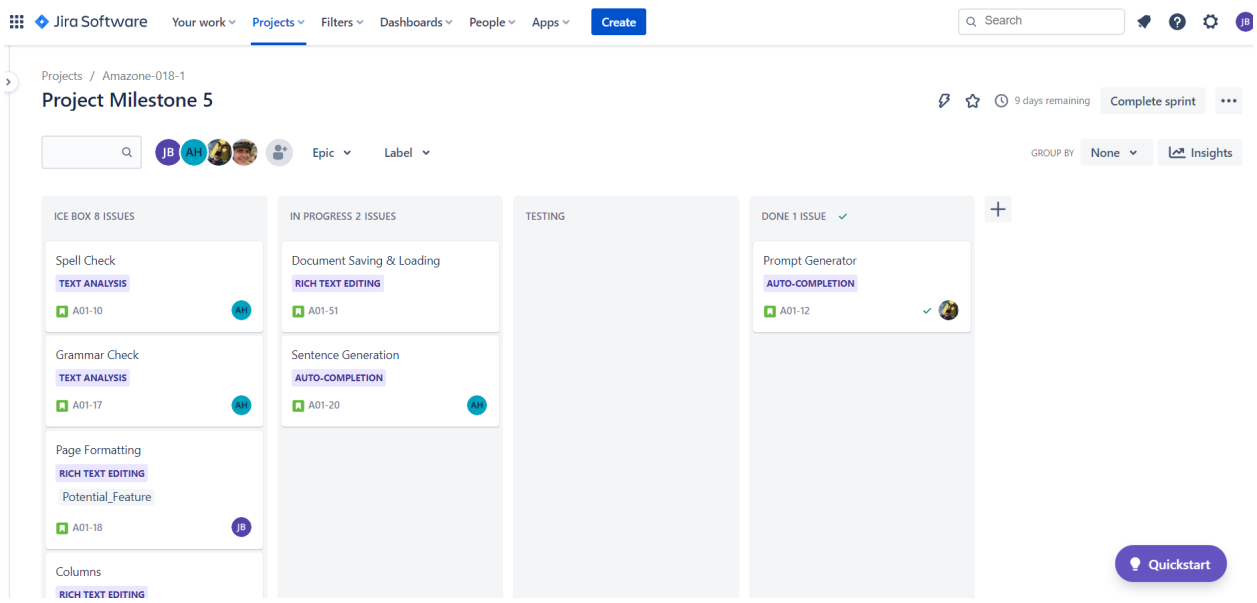
**Nolan:** Finished Python review/cycle.js tutorial and worked on index page; helping with prompt generator page next

**Aleksey:** Created html layout for the prompt generator page**.** Working on the prompt generator page and creating a  functioning database for the page.

**Link to last commits:**

https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-018-01/commits/main

**Screenshot of Agile Board:**




Note: After updating and completing the milestone 4 sprint, I was unable to view it and screen shot it. Therefore I've put the board for our current sprint (being "milestone 5" with its contents as finishing the integral portions of the website, as well as some more advanced optional features).