Startup 05 Report

Grade using the commit this report was last updated on.

Outstanding Bugs

- [TODO]: search function in navbar currently has mock data, and does not link to anywhere. This will be fixed by having the navbar or search bar contain internal state, with the current set of documents.
- [TODO]: The username is currently hardcoded as 'Kendrick Lamar'. We will have this load from the database, passing in the userID.
- [TODO]: Right now the only functional suggestion category is rhyme. When the other pills are clicked on the suggestion bar or in the popover, the state is updated in Workspace, but the proper suggestions are not passed into the SuggestionBar component to become Suggestion components.
- [TODO]: We have not implemented document specific settings.

Key React Components

- NavBar
  - This is a persistent component in our single page application. Currently static, with stubbed suggestions, although soon it will query the database, and update to show the user's username.
- Workspace
  - Contains the TextEditor component, and SuggestionBar component. Pulls document data from the database, and renders it in the TextEditor. Captures events from the TextEditor, hitting the relevant APIs, and passing the returned data to the SuggestionBar component. It also renders document title, provides a 'Save' button for users to save their documents, and presents a 'last saved' timestamp, when the current document has no edits since the last save.
- SuggestionBar
  - Renders the widget to the right of the text editor on the workspace page where the user will view word suggestions and select which category of suggestion they would like to view
- Suggestion
  - Renders each word that results from an API call on the word that the user has selected, based on the category of suggestion that the user has also selected.
- Settings

- ○ Contains multiple list components that form the Settings page. Pulls user account information and user settings from the database to populate the fields. When the fields are changed, the data is pushed to the database and updated in the user object. A pop-up update password modal also exists which has multiple checks in place to make sure the old password matches and the new passwords match. Once matched, the password can be updated and the data is sent to the database. Currently the password is sent and stored in plaintext, as the raw functionality was the key focus for this stage in the development.

Individual Contributions

- ● Alan
  - ○ I worked on SuggestionBar.js, and tinkered with its parent component, Workspace.js. I wrote functions defined in Workspace.js that are passed into the text editor and the suggestion bar that handle updating the state of the Workspace component to keep record of the current suggestion category in order to call the right API, produce the right kind of suggestions, and highlight the right pill in the suggestion bar. No database coordination was necessary because it's not important to the function of our application that the user's last list of suggestions is remembered.
- ● Hiram
  - ○ I worked on the Settings page and all the components with it. My main task was to get everything on the page functioning with the mock database and editable. The page initially reads from the database, populates the fields, then if the user changes anything it will be pushed and saved back in the database for persistent information. Once functionality was built, I also had to link the page through the Navbar settings link, and fix the css of other files so the settings css wasn't overridden.
- ● Matthieu
  - ○ I worked on Saved Documents which includes adding new documents/collections. From a users collection, I implemented links to the documents in that collection. Currently need to implement the route that brings a user to a new document when they make a new document and the ability to change a documents name in the Saved Document view.
- ● Misha
  - ○ I worked on the WorkSpace. I implemented read/update, allowing users to save their changes to the database, and load their saved documents. I also implemented rendering of saved documents from the database on the Saved Documents page.
- ● Tay

- I worked on the Editor Toolbar. I implemented a functional drop-down menu for each of the buttons in the toolbar, each of which now expand to reveal several different options.

Cut Features

- [Possibly] Document specific settings. We have not implemented a way for users to adjust settings for specific documents yet.