**Documentation**

**Purpose**

My site is a study aid similar to Quizlet. It allows users to add and view subjects, study sets within the subjects, and the questions and answers within the study sets. If a user notices an error with a question or answer, they can edit it by clicking on the text of the question (not the flashcard).

**API**

The API holds all of the data on subjects, study sets, and question/answers. The API handles get requests for subject, study sets within a subject, and questions within a study set. The API also allows for creating or modifying any of these items. If the user inputs a faulty subject/study set request, it

**Reflection**

**Right**

The server side was pretty simple as it constituted mostly just storing the data passed in. The hardest part of the project was the client side, figuring out how to display the information.

**Wrong**

I tried separating the JS code for the client, but I had difficulty since I was bundling it and how I originally structured the code. It is currently too interdependent and I would definitely work more on designing first and implementing later, so I don't have as much overlap. I tried to separate the code on the server as much as I could, but with the JSON responses, it didn't make much sense to separate them further as they all used the template respondJSON and respondJSONMeta functions.

**Future Improvements**

**Styling**

One of my weaknesses is styling and design. If I had more time, I would try to fix the design and add styling for things like flipping the flash card.

**Removing Information**

Another thing I would fix is letting the user remove things like subjects, study sets, and questions.

**Refreshing**

Currently, the way I use the back and forward buttons makes it so that if the user just tries to refresh the page, it will send them to the JSON output rather than the HTML page. Since this is not ideal, if I had time I would try to find a solution for this (but this could easily be avoided by using multiple HTML pages so if working in the future, I would probably just convert to using handlebars).

**Above and Beyond**

**History**

The user can click on the back and forward arrow buttons, and they will work. Every time they change between "pages" (subject to study set to flash cards), the current state is saved in the History and can be returned to.