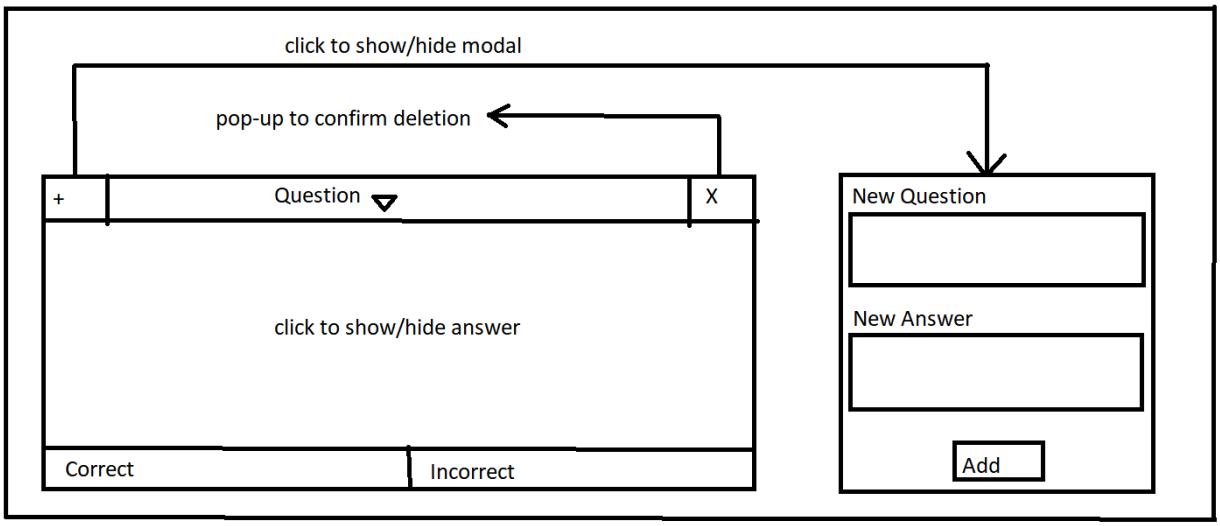
CSC309 Individual Project Proposal - Flashcard Library

Prerak Chaudhari | 1005114760

Introduction to the Flashcard Library

This flashcard library is a **learning utility** that generates a DOM element containing a flashcard deck. Populating this deck with flashcards allows one to utilize it as a learning tool. This library is made more creative/original by utilizing the Leitner system to incorporate spaced repetition into the flashcard shuffling algorithm. This will promote memory retention by more frequently repeating difficult flashcards.

The end user functionality and interactions are visualized in a sketch below:



End users are able to mutate the flashcard deck. New flashcards can be added by clicking on the **+** button, which will toggle a modal containing a question and answer text form. After filling this out, a new flashcard will be added to the deck once the **Add** button is clicked. The current flashcard can also be deleted by clicking on the **X** button, which will trigger a pop-up asking for confirmation. If confirmation is given, the flashcard will be deleted and the deck will automatically switch to the next flashcard. Users can switch to a particular flashcard by selecting the corresponding question prompt from the dropdown menu (see **triangle**). When an element from the dropdown list is selected, the answer area will automatically be updated. By default, this answer area is hidden but the visibility can be toggled on/off by clicking somewhere in the answer area. Users can specify whether they correctly answered the current flashcard or not by clicking on the respective **Correct** or **Incorrect** buttons; the deck will then automatically switch to another flashcard.

This library is beneficial for developers as they do not need to program the DOM interactions and manipulations required for proper UI functionality themselves. Furthermore, they don't need to code a proper shuffling algorithm.

Balancing Specific and General Library Use

This library provides all the specific functionality one would expect from a flashcard deck. As described in the previous section, one can mutate the flashcard deck and shuffle through it intelligently. However, the library is also general enough that it can be used in any web app that could benefit from flashcard utilization. Since developers are presented with a black box that works as intended, they can leverage the library to supplement their web app's application logic.

Three Example Use Cases

One type of web app that could use this library is trivia websites. It would be beneficial for the site to allow users to create and share resources that help them practice trivia topics. Developers can leverage this library to handle the DOM manipulation and shuffling algorithm while they focus on providing functionality that lets end users create, save and share their decks. End users benefit by leveraging available flashcard resources on the site to improve their trivia skills. The decks will frequently revisit difficult flashcards and thus improve the end users' memory retention.

Another type of web app that could use this library is online textbooks. Publishers selling digital copies of their textbooks already allow users to annotate the pages (i.e. highlighting words) so it would be beneficial for the site to provide flashcard functionality. Like with trivia websites, developers benefit from not needing to figure out how to create intelligent flashcard DOM elements. Instead, they can spend resources on linking flashcard decks to various passages in a textbook. End users benefit as they can create flashcard annotations and can refer to the linked section in the textbook if they do not understand the flashcard answer.

A third type of web app that could use this library is educational software. It would be beneficial for educators to be able to create flashcard learning resources for their students on various course materials. Once again, developers benefit by being given a black box DOM element that works as intended. Thus, developers can focus on enabling educators to create and release flashcard decks for their courses. There are 2 end users in this case: the instructor and the students. The instructor benefits from having this flashcard functionality built into the software they are already using to teach the course. The students benefit from being time efficient and only revising the course material they are struggling with.