Spaced Repetition System
Fall 2023 Project Proposal

Group Membership

Name	Skill Sets	Credit Hours	Responsibility
Prasad Shetye	Python, Java, C, C++, SQL, HTML/CSS, Javascript Algorithms	3-4 hours a week/ 1 credit hour	Team 1: Creating initial flashcard framework/Implementing repetition algorithm
Chaz Edmond	Java, C, Python, Basic SQL, HTML/CSS	3-4 hours a week/ 1 credit hour	Team 2: Brainstorm/Researching repetition algorithm/Implementing repetition algorithm
Sattwik Mallavaram	Python, Java, Researching, Writing, SQL, Swift	3-4 hours a week/ 1 credit hour	Team 1: Creating initial flashcard framework/Implementing repetition algorithm
Vinayak Ramasubramanian	Python, Java, SQL, HTML, Algorithms	3-4 hours a week/ 1 credit hour	Team 2: Brainstorm/Researching repetition algorithm/Implementing repetition algorithm
Ani Vedartham	Python, Java, SQL, Algorithms, HTML/CSS	3-4 hours a week/ 1 credit hour	Team 1: Creating initial flashcard framework/Implementing repetition algorithm

Project Goals

- Use a tool to extract keywords from a section of the textbook.
- Have a method (maybe ChatGPT) for defining the word based on the textbook.
- Turn the keyword and definition into front and back flashcards (eventually add the ability to type in and check your answer).
- Come up with a spaced repetition algorithm to determine the review interval for a card.
- Find a method for determining the difficulty of a card as an addition so a card already has an interval modifier before someone starts learning it.
- Implement it into the ChatBot Recommender

Timeline

Week 1-3: Introductions, Team formation, Project Logistics
Week 4-5: Research spaced repetition algorithms and plan out the implementation in our project
Week 6-7: Create the framework flashcard program
Week 8-9: Brainstorm/research methods to determine the difficulty of cards and spaced repetition algorithms
Week 10-11: Implement the spaced repetition algorithms
Week 12-13: Researching previous implementations of the keyword extraction and possible adaptations to own project
Week 14-15: Optimize the system and add final features before testing
Week 15-16: Interact with system to find and fix apparent flaws and make final adjustments
Week 17: Final Presentation

Project Description

Problem to be Solved/Research Potential

- A method for effectively using your time to study important terms within a textbook
- Manually picking out terms and creating flashcards is tedious so we are looking to do this automatically
- Research methods of determining the difficulty of a term/concept

Proposed Solution

By using a similar keyword extraction method as the ChatBot Recommender, we would be able to pick out important vocabulary words from a textbook. We would then have ChatGPT come up with a definition that is derived from the textbook. Words and their definitions will be turned into flashcards that can have a typed answer. Afterward, we will implement a spaced repetition system, which allows users to review newer or difficult cards more often than old or easy cards. A difficulty statistic will be the basis for the algorithm determining the interval between reviews. Eventually, this can be tied in as a feature of the ChatBot Recommender, providing more tools to someone looking for a simpler and personalized textbook.

Foreseeable Problems

It might be difficult to automatically produce definitions for the flashcards based on textbook material. The standard and correctness of the words may vary, which would give the students a wrong impression of certain concepts.

It could be difficult to implement the algorithm to determine the difficulty of each flashcard. This is because the accuracy of the algorithm will be dependent on the quality of the data collected. Limited data for the algorithm could pose a significant issue with how well these difficulties are measured.

Another problem would be if the algorithm encounters very obscure or unique concepts/words. The data collected will have a factor in how well these topics can be converted into flashcards. On top of this, if the data collected is skewed in any way, one example being we only have data on mostly expert/proficient learners, then the data will not be representative of beginners at all which will cause them to have a tougher time. The overarching complication with most of these situations is the quality of data collected for the algorithms to implement.

Implementation Tools

We plan to use the textbook database to help train our model to generate the definitions. For this, we will draw inspiration from existing AI and ML models already out there, including the OpenAI API. To store the various users' information and definitions of words, we will need to have a database. For this, we plan to use SQL. To create the backend of the platform, we plan to use Python, which will allow us to code our spaced repetition and difficulty algorithms as well as access our database from SQL. For version control and collaborative software development, we plan to use Github so that updates that each member contributes can be saved and added to the project.

Future Plans

In the coming semesters, we hope to explore the possibility of hosting this program online so that it can be accessible through the web. To accompany this, we will create a login feature to help users save their progress and the algorithm that has been specifically tailored towards them. In addition, we hope to either integrate an existing front end or create our own front end for this program so that it is easy to use and also aesthetically pleasing for users. Lastly, we plan to implement an input for users so that they can adjust the priority of any specific card, so that they can manually adjust how often they see a specific card depending on their preferences.