Documentation: Imprint

DBMS Semester Project

Team 8:
Vibhaas Nirantar Srivastava
Jayanth P
J Glen Enosh
Saksham Saklani
Niranjan Alase

We describe a full-stack web-based application, "Imprint", which aids students to study more effectively by using flashcards and spaced repetition. We integrate LLMs to facilitate easier flashcard creation by the user. User data along with usage statistics are stored in a large RDBMS, which is the core of the application.

App Description

Flashcards are a scientifically proven method to quickly memorize and learn information, especially when proper spacing is taken into account. From Wikipedia,

> "**Spaced repetition** is an evidence-based learning technique that is usually performed with flashcards. Newly introduced and more difficult flashcards are shown more frequently, while older and less difficult flashcards are shown less frequently in order to exploit the psychological spacing effect. The use of spaced repetition has been proven to increase the rate of learning."

Our application, "Imprint", is designed to do exactly this — provide an easy, intuitive, and effective method to leverage the power of spaced repetition to study and memorize.

Each user can create multiple "decks" of flashcards they wish to study, which can be sub-categorised by tags. Entering the data for the cards can be manual or using the LLM. The decks and the cards within are shown in intervals decided by a spaced repetition algorithm. Also, the user is provided with statistics, a heatmap, and other information and customisation options so that they can maximize the efficiency of their study.

Uniqueness of Imprint compared to other SRS

Imprint is unique when compared to other popular SRS software like Anki, because it natively connects to an LLM which can query its database and generate card suggestions for the user. The LLM can:

1. Suggest alternate phrasing and structure for a card.
2. Convert a given card type from one format to another automatically, for example, from Question-Answer to MCQ format, and even generate the wrong answers for the MCQ format.
3. Generate alternate cards based on subtle shifts of the original card.
4. Fetch relevant information related to the subject matter of the card, enhancing the card.

And much more! This greatly enhances the ease of use for the user, and allows the user to focus on what is important: Learning, and not spending time making flashcards.

Database Description

The project is very database intensive; we need to store thousands of flashcards, categorized into decks and organized by tags — these flashcards are also grouped by user, so we need to store user authentication. Along with this, since we are implementing spaced repetition, a lot of user statistics on cards must be collected: the times they have reviewed the card, the card ease, their learning rate, etc.

We design our database so that it may be easily queried by an LLM. Along with this, we implement a way to Import and Export flashcard decks, which involve advanced database applications.

Please refer to the attached ER diagram and schemas for further description of the database.

Technologies and Frameworks Used

Broadly, we are using a MySQL database, with React for the frontend and a Python+Flask backend.