

## Author

Kunal Chaturvedi

21f1003353

21f1003353@student.onlinedegree.iitm.ac.in

I like programming, design and creating products.

## Description

The goal of this project is to create a web application which helps the user memorise anything the user wants using cards and feedback based on the memorization capability of the user. Further, a scoring system helps guide the feedback system of the application.

## Technologies used

The application uses Flask, FlaskRESTful, SQLAlchemy and Vue JS to implement core functionalities. Vue JS is used for templating and reactivity in HTML. Bootstrap is used for UI Design and responsiveness. Vue Components have been used to ensure maximum reactivity and responsiveness. Redis has been used for Caching along with the Flask Caching library to ensure faster results at optimizable places. Celery has been used to implement background jobs for processes like monthly reports and daily reminders.

## DB Schema Design

**TABLE USER**

Column Name	Type	Constraints
username	TEXT	Primary Key
password	TEXT (Hashed with SHA256)	Not Null

Every username is uniquely identifiable.

**TABLE DECK**

Column Name	Type	Constraints
deck_id	INTEGER	Primary Key, Auto Increment
deck_name	TEXT	Not Null
username	TEXT	Foreign Key "user"("username")
deck_score	INTEGER	Default=0
last_reviewed	TEXT	Default=None
ind_reviews	INTEGER	Default=0

It has a deck score and a card review count for average calculations. A last reviewed time for the deck is also recorded.

#### TABLE CARD

Column Name	Type	Constraints
card_id	INTEGER	Primary Key, Auto Increment
deck_id	INTEGER	Foreign Key "deck"("deck_id")
question	TEXT	Not Null
answer	TEXT	Not Null

Deck ID is referenced from the Card Table.

#### TABLE REVIEW

Column Name	Type	Constraints
review_id	INTEGER	Primary Key, Auto Increment
card_id	INTEGER	Foreign Key "card"("card_id")
deck_id	INTEGER	Foreign Key "deck"("deck_id")
score	INTEGER	Not Null
time	TEXT	Not Null

Each review is counted for each card along with score and time. Each card has a card\_id and belongs to a particular deck.

## API Design

The application has a robust API with multiple endpoints protected by authorization with Flask JWT for optimal security. Further the API allows for creation, reading, updation and deletion for all Decks and Cards, along with the ability to fetch statistics for the user.

## Architecture and Features

The project has a controller and the database files in the root. Static and Template files for HTML are hosted in respective folders.

The application has a login system, deck management (CRUD on decks and cards) along with a scoring system. A card review system. Fully responsive UI. Feedback and recommendations based on scores on decks.

## Video

<https://drive.google.com/file/d/1SpJluBvm5LC-xrgu-pV9zX88cBpzNySC/view?usp=sharing>