FLASHCARDS APP:MAD1 PROJECT DOCUMENTATION

The Flashcard App is used as a learning tool as it helps a person memorise much needed ideas and facts.As the final project for the MAD1 course, the Flashcard App is made using Flask and sqlite.The database consists of 3 tables:user_info,deck_info,card_info

The App consists of 4 major parts:

- A login and signup system that updates and keeps track of users vis a user database
- User Dashboard displaying all the Decks created by him
- A Deck management system to add,update and delete cards and to remove the entire Deck from the system
- A Deck review system that assesses how well the User knows the information from the cards and scores the User based on his/her performance.

1. Login/Signup System

The app uses a basic login system where every user has a unique username which is stored along with his name and password upon signup. Upon login the details of the corresponding User is identified from the database and opens his/her dashboard.

Also both frontend and backend validations have been set up to check for and prevent invalid data,non existing user and multiple signups from the same username.

2. User Dashboard

The User Dashboard displays all the Decks made by the user and provides links to its corresponding Deck management and review systems.It also displays the scores received upon Completing review and time of the last review.

Also an Add Deck feature is present where you are allowed to create an empty Deck with its name and description.

3. Deck Management System

Deck management system displays all the cards present in its corresponding Deck and allows the option to edit or delete cards from the deck.Also an option to completely delete the deck has been provided which deletes all the info corresponding to the deck and its cards from the database in a cascading process.

4. Deck Review System

Deck Review System forms the major part of the application as it displays to the user each card in the deck one after the other and obtains the level of difficulty of the card depending on the users opinion from:Easy,Medium and Hard.the system also calculates the score of the deck based on this user response and store it in the database so as to display it in the dashboard.Also the system updates the last_review time depending on when the last card of the deck was answered.

- **DATABASE**
  Database used for the application mainly consisted of 3 tables:

1. **User_info** :This table consisted of the name ,username and password of the users and was connected to the deck_info table via the username field.

2. **Deck_info** :This table stored the deck related information such as the deck_id,name of the deck,its short description,the score obtained, last reviewed time and the username corresponding to the user who made the Deck.It is also connected to the User_info table via username field and to the card_info table via the deck_id field.

3. **Card_info** :This table stores the information regarding the individual cards and is connected to the deck_info table via the deck_id field.The table also stores the individual card ids,front and back of the card(question and answer in the card respectively),the individual card score obtained and the deck_id to find which deck the card belongs to.

SCORING:

The review scoring pattern is based on a percentage system,ie,the score is computed considering the maximum possible marks to be hundred.While reviewing the deck ,the user is given 3 options-Easy,Medium and Hard and the options are given the score 1,2, and 3 respectively.

Once the User leaves the review the score is calculated by dividing the obtained score with maximum possible score(3 x no.of attempted questions) and multiplied with 100 to bring the score out of hundred.

Along with the computation of score,the system also computes the time when the User submitted their last response and saves it as the last review time which is available in the User dashboard from then.

The working overall uses basics of flask,sqlalchemy and sqlite to create a system which maintains records of users,their decks and cards help to manipulate ,review and score them based on their relative understanding of the cards.

APIs

Custom APIs were made using flask_restful and yaml documentation was done using swagger.editor tool. The API was able to extract a user given their username and extract a deck or card given their deck or card id respectively.Also API were added that could add new users and delete decks and cards from the system.The documentation also includes the schema to the various formats of output jsons.'The documentation now is linked to the replit url on which the app is running