MAD FINAL PROJECT

AUTHOR
Name - SHIVAM ARORA
Roll No. - 21f1006965
Email Id - 21f1006965@student.onlinedegree.iitm.ac.in

Introduction - I am pursuing Btech in Computer Science with specialization in Social Sciences along with majors in Economics from IIITD. I am currently working as a Marketing Analyst intern at Goodera.

DESCRIPTION

The aim of this project is to create a flashcard web application which involves a user logging into their account and creating and managing their own flashcards. These flashcards are used for memory training and are quite useful as they use active memory recall. The user can create and add multiple decks of cards and then based on his performance he will be scored for each deck after reviewing all the cards and marking his level of familiarity with the card content.

TECHNOLOGIES USED

1. Flask
2. Jinja2
3. Bootstrap
4. HTML
5. SQLite
6. Replit

DATABASE SCHEMA DESIGN

There are 3 tables in the database of this application. These are :

1. **User -** This table is used for storing the information of the user. The attributes of these table include id, username, password, date of creation, score, userdeck (udeck).
The schema for the table is as follows

ID	USERNAME	PASSWORD	DATE_CREATED	SCORE	UDECK
int	String(30)	String(300)	DateTime	int	relationship

2. **Deck** - This table is used for storing all the information related to the decks of the user. The attributes of this table include id, deck name, user, date of creation, score, is_public, last reviewed, dcard**.** The schema for the table is as follows

ID	DECK_NAME	USER	DATE_CREATED	SCORE	IS_PUBLIC	LAST_REV	DCARD
int	String(30)	String	DateTime	int	boolean	DateTime	Relationship

3. **Card -** This table is used for storing all the information related to the cards of the user. The attributes of these tables include card Id, front, back, score, deck. The schema for the table is as follows:

CARD_ID	**FRONT**	**BACK**	**SCORE**	**DECK**
int	String(512)	String(512)	int	String

API DESIGN

We have created three APIs :

1. User API - This has two functions - POST, GET
   - POST - User Register/login
   - GET - Fetching User Information

2. Card API - This has two functions - POST, GET and PUT
   - POST - Create a new card
   - GET - Fetch that card
   - PUT - Take review score of the card and add it to the overall score

3. Deck API - This has two functions - POST, GET
   - POST - Create a new deck
   - GET - Fetch that deck

ARCHITECTURE AND FEATURES

We have used the MVC architecture for this application.

The following are the main functionalities that are used in the application:

1. User Login
2. Personalised card decks that can be added, deleted or edited
3. Personalised cards inside each deck that can be added, deleted, edited and ranked according to the person's ability (memory)
4. For scoring system we have decided 3 levels - Easy (1 point) , Medium (2 points) and hard (3 points)

VIDEO

https://drive.google.com/file/d/1QQjHOT1RxpD9JpDrqkif9wrtN-6Zpb3j/view?usp=sharing