

Author

Name: SIVA SUBRAMANIAM J

Roll No: 21f1005023

Email : 21f1005023@student.onlinedegree.iitm.ac.in

I am an aspiring Engineering and Undergraduate student. I completed my Foundation Level of the B.Sc Degree in Programming and Data Science by IIT Madras. I am Currently pursuing my Diploma in the same.

Description

This project is about implementing Flashcards using web application. Flashcards are basically cards used for memorizing concepts. The web application is to be created using Flask, Python and SQLite3.

Technologies used

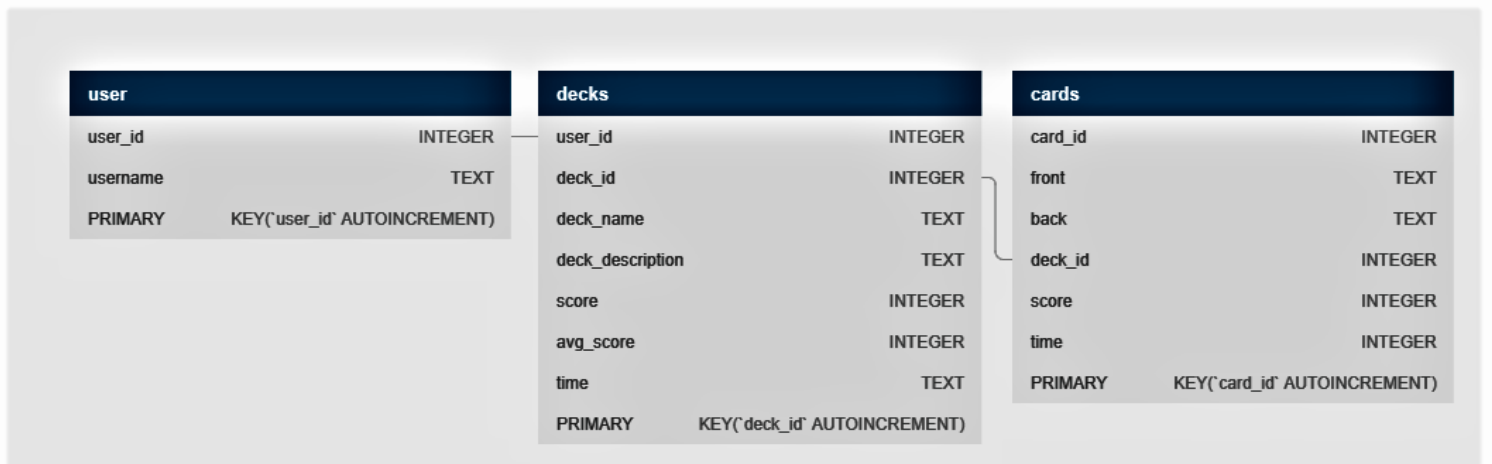
This application uses Python as the main programming language, Flask for implementing the Web Framework and Database is implemented using SQLite3.

Python is used as it provides with variety of choices to choose from various web frameworks and its ease of programming owing to its inbuilt and external packages.

Flask provides with variety of tools and libraries for the development of the application.

SQLite3 is used because of its lightweight structure to store simple data

DB Schema Design



- There are 3 Relations used:- **user, decks, cards.**
- The column attributes are as shown above.
- The column user_id is used as foreign key in the relations user and decks to uniquely retrieve decks associated to user.
- The column deck_id is used as foreign key in the relations decks and cards to uniquely retrieve cards from a specific decks.
- There is one to many relationship from user to decks and from decks to card.
- It is done so because a user can have more than one deck and also a deck can have more than one cards.

Architecture and Features

The Project is created using the standard MVC- Model View Controller Architecture using the web framework Flask. The database technology used is SQLite3 owing to its simple structure and ease of access when the data is simple and primitive.

This is organized in a single "app.py" which contains the Models, View and Controllers.

The html templates are stored in the folder called "templates" in the root directory. The necessary css and images required for the views are stored in a folder called "static" in the root directory.

Core Functionality/Features

- The user login is implemented in a primitive way using global variable in python. Also sessions are maintained using the same
- The Scores are updated as a review gets completed. The last review time is managed by "datetime" module available in python
- For Deck management , The Create, Read, Update are implemented by the controllers itself. I have implemented API only for the Delete part of Deck Management. The Card management is also implemented by the controllers itself.
- The Review is implemented by fetching all the cards corresponding to specific deck only once and saving the card object in a list. The card objects are popped one by one from the deck for review until the list gets empty.

Video

<https://drive.google.com/file/d/1ZPzMhyIwqeyTN5oC4ot9gVKwz5la29dD/view?usp=sharing>