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. I am a bike enthusiast. I love traveling and exploring places.

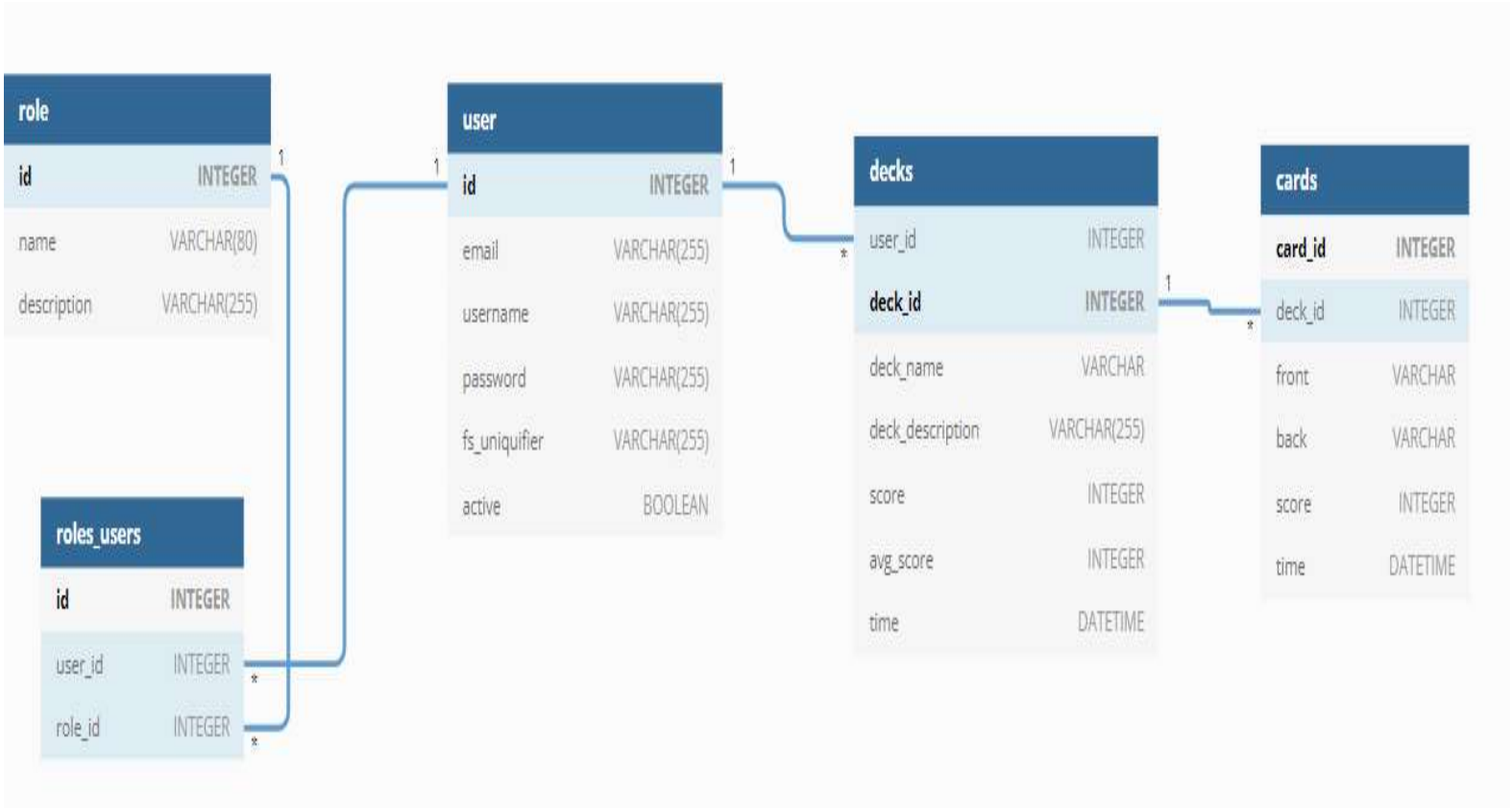
Description

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,Vue and SQLite3.

Technologies used

This application uses Python as the main programming language, Vue for implementing the Web Framework, Flask for implementing API 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.
Vuejs provides reactivity for application when needed.

DB Schema Design



.

- The database contains 4 Tables **User, Role, Decks, Cards, Roles_users**
- The mentions tables contains columns as seen in the above diagram
- User Table maintains the login details of the User
- Role Table is not being currently used but it is been created to facilitate the Flask-Security.
- Decks Table contains the details of the decks. There is one to many relations from user to decks.
- Cards Table contains the details of the cards.There is one to many relations from Decks to cards

Architecture and Features

The Project is created using Vue framework. All the logics are implemented in the frontend that is in Vue and data is fetched and transferred to and fro from the database completely through the flask API.
There is single html template in the template folder in root directory, other necessary templates are dynamically changed when needed with the help of **Vue Router** using **Router-links and Router-views.**

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

Core Functionality/Features

- The user login is implemented using Flask Security using Cookie based authentication.
- 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 **CRUD** operations are handled with the help of vue and its fetch API. The frontend when required to perform the operations call the specific API appropriate for the operation and update the database
- .For Card management:-Same as for Deck Management, but for cards
- Review of deck is completely manage by JavaScript by fetching the required cards of the deck and review it and update the score and again use fetch api to "**PUT"** the updated score of each card
  After the each deck review is completed, the total score of the whole deck is updated using the fetch api to "**PUT"** updated score of the deck

Video

https://drive.google.com/file/d/1s4l6cqcFRvPJkGJd2BcVBS6idbMpGT14/view?usp=sharing