Author:

Sarthak Gautam
21f1000864@student.onlinedegree.iitm.ac.in
I am currently pursuing B.Sc. CS from Post Graduate Government College along with this online B.Sc. Degree in Programming and Data Science. I love mathematics and computing.

Description:

In this Project, I created a basic web application that let's user decide the difficulty of the word and later display the score on User Dashboard. The Application provides a basic user login system that separates the decks of different user. A user is allowed to add/update/delete a Card.

Technologies used:

- Flask
- Flask_sqlalchemy
- datetime
- random
- flask_restful

datetime is used to get date for last access. ***random*** is used to generate a random number and then get card associated with the random number.

DB Schema Design:

user Table Schema

Column Name	Column Type	Constraints
user_name	String	Primary Key,Not Null
first_name	String	Not Null
last_name	String	

deck Table Schema

Column Name	Column Type	Constraints
word_id	Integer	Primary Key, Auto Increment
deck_name	String	Not Null
Word	String	Unique, Not Null
Meaning	String	Not Null
score	Integer	Not Null
user_id	String	Not Null

Here, Only two tables are used in order to reduce the complexity/Constraints for add, update, delete. Also the main theme of the app is to separate flash card data of users so by using user_id in deck table(which is primary key of user table) we can easily manipulate the data according to user_id as while querying the database we can use user_name from url itself and it would be different for all user due to it's Unique constraints.

API Design

In this, Basic Api's of Get and post are created.

GET:
Almost every page has some kind of data associated with it like username, word etc. so using Get we can have a json format for the data that is to be used in the html template.

POST:
Many pages sends data to server. To facilitate this POST is used

DELETE:
A word/deck can be deleted so we have DELETE method.

Architecture and Features:

In this project every element occupies it's obvious place, like, all html pages are stored in Folder named templates, CSS and images is stored in Folder named static. The controller and database schema are in app.py . Also a database named tested.sqlite3 is stored along with the app.py itself. API's are stored in api.py that is later imported into app.py

This is a multipage app i.e all views are done using different html pages. The app url carries information such as username, deck name, word etc. that can be used later in @app.route to access data related to specific user. This helps to separate the data of users. If we open a deck then a random word is shown from database and to view it's meaning user has to hover the cursor over card. Also review is mandatory for each card. A User can Add, update, delete a Word/deck. Additional feature: A user can also view/update his/her profile.

Video:

https://drive.google.com/file/d/1t4BBV51c2pViYXrVAT-ncRKYVLYEEid_/view?usp=sharing

App link:

https://cardapp.sarthakgautam2.repl.co/