Flashcard App

Project Report

Author

Nitish Agarwal
21F1001802
21F1001802@student.onlinedegree.iitm.ac.in
I am a sophomore and an altruistic person. Ready to help others with all the strength. Skillful. Creative. Full of enthusiasm and geared to work at any hour of the day.

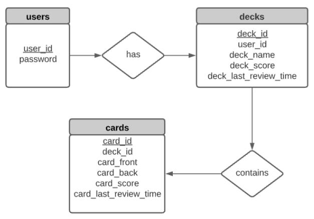
Description

Flashcard App is a RESTful web app designed with the help of SQLite database and Flask framework using REST API. The APIs will connect to a database to keep track of the records inserted into the app. Python language along with HTML and CSS is used in the development of this RESTful web service.
This web app provides an interface which allows the users to access, create an account and maintain a database for the users accordingly so as to reduce the inconvenience of logging in and out repeatedly. The system will also allow users to maintain their decks along with cards where they can put relevant content for memorization and future use. They will also be able to review their decks and cards as well as get scores during the course of revision if need be.

Technologies used

Python: The basic programing language in which the code of the app is written
Flask: A web framework written in Python and provides us with tools, libraries, and technologies that allow us to build a web application.
SQLite: Used for creating databases and its management.
SQLAlchemy: Library that facilitates the communication between Python programs and databases.
HTML: Used for structuring the webpages.
CSS:Used for adding aesthetics to the website.
Javascript: Used for making the webpages interactive.

DB Schema Design



User_id is the primary key of table users and the password should be unique for all users. All the fields for all tables should be a non-empty entry. Deck_id of table decks is the primary key and the user_id of table decks is foreign key for the table users. Card_id of table cards is the primary key and the deck_id of table decks is foreign key for the table decks.

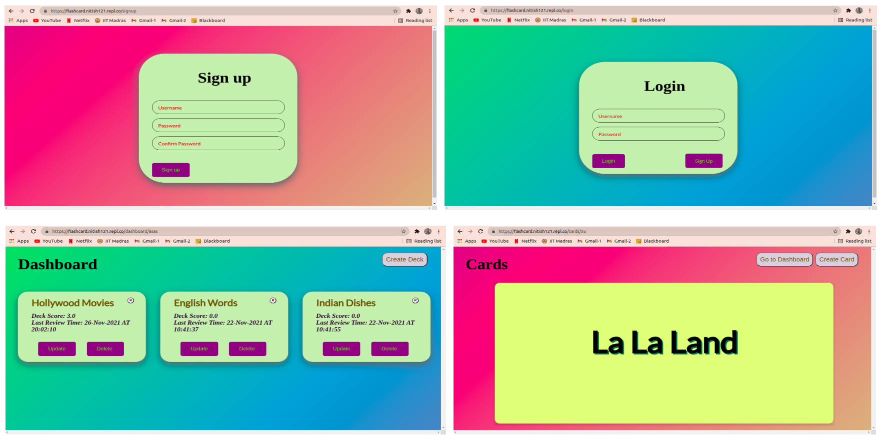
The schema has been designed as such after seeing the requirements of the flashcard app. The basic functionality of the flashcard app should have users, decks and cards, which are represented as tables.

API Design

Suitable APIs have been created for each web page of the app. API for the signup and the login pages have been created which redirect the user to the login page and the dashboard respectively after filling the required details. For the dashboard page, APIs are created to enable the user to perform CRUD operations, the cards page for each deck follows the same.

Architecture and Features

The project is divided into 4 subfolders and a main.py file. The main method is written in the main.py file where the app is initialized. The folder templates contain all the HTML files which are required for the structuring of the webpages. The static folder contains a subfolder which has the js file and a css file for the aesthetics of the website. The db_directory folder contains the sqlite database. The applications folder contains 4 python files: config.py contains the configuration for the server, models.py contains the python version of code for the schemas, database.py contains the code to initialize the database and the controllers.py file contains all the APIs specified along with their functionalities.
Additionally, I have added a reset button present at the top right corner of each deck.
Home page: The first page of the flashcard app opens up the Login page. If the user has an account, they can fill in the details and click on login, else click on signup and create a new account.
Dashboard: After successfully logging in, the app opens up the user dashboard, where all the decks created by the user are present. Here the user can manage their decks and create new decks as well. Additionally, a Reset button is present at the top right corner of every deck, after clicking on which the deck score as well as the deck review time will be set to default values.
Cards: If the user clicks on any deck name, the corresponding cards of that particular deck will be opened. The user can manage their cards and create new cards if needed. If the user hovers over the card, it flips and shows the information the card contains. The Users can follow from the description and choose the difficulty accordingly.



Deck Score: 3.0
Last Review Time: 26-Nov-2021 AT 20:02:10

Deck Score: 0.0
Last Review Time: 22-Nov-2021 AT 10:41:37

Deck Score: 0.0
Last Review Time: 22-Nov-2021 AT 10:41:55

Video

https://drive.google.com/file/d/1p7pKDd_C__I9tV1txo-thaM9dFms7ZVl/view?usp=sharing