Author

Name: Sejal Anand
Roll No: 21f1002620
Email: 21f1002620@student.onlinedegree.iitm.ac.in
About:
I graduated last year with a Bachelors in Computer Applications. My interest and enthusiasm to learn and make a career in Data Science made me take up this degree. With my dedication, sincerity and perseverance, I hope to learn all this degree has to offer to the best of my abilities.

Description

This project is a Flashcard application for flashcards generation and reviewing. Users can create their decks in which they can create flashcards. The front and back of the flashcard are entered by the user while creating flashcards. The users can review their flashcards through a quiz and their dashboard stores information like their average score on a particular deck, last quiz score and last review time.

Technologies used

- **Flask:** Used for building the web application and backend
- **Flask SQLAlchemy:** For CRUD operations to SQLite database through the flask web app
- **HTML and CSS:** For template coding and minimal styling
- **Jinja2:** For using programming logic in templating
- **Bootstrap:** For layouts and styling of all web pages

DB Schema Design

Tables (5)

Name	Type	Schema
card		CREATE TABLE card (card_id INTEGER NOT card_front VARCHAR(100) NOT NULL, card_b VARCHAR(100) NOT NULL, difficulty VARCHA PRIMARY KEY (card_id), UNIQUE (card_fron
card_id	INTEGER	"card_id" INTEGER NOT NULL
card_front	VARCHAR(100)	"card_front" VARCHAR(100) NOT NULL
card_back	VARCHAR(100)	"card_back" VARCHAR(100) NOT NULL
difficulty	VARCHAR(100)	"difficulty" VARCHAR(100)
card_deck_relation		CREATE TABLE card_deck_relation (card_deck_relation_id INTEGER NOT NULL, "cardCDR_foreignid" INTEGER NOT NULL, "deckCDR_foreignid" INTEGER NOT NULL, PR (card_deck_relation_id), FOREIGN KEY("cardCDR_foreignid") REFERENCES card (card_id), FOREIGN KEY("deckCDR_foreigni REFERENCES deck (deck_id))
card_deck_relation_id	INTEGER	"card_deck_relation_id" INTEGER NOT NULL
cardCDR_foreignid	INTEGER	"cardCDR_foreignid" INTEGER NOT NULL

Name	Type	Schema
deckCDR_foreignid	INTEGER	"deckCDR_foreignid" INTEGER NOT NULL
deck		CREATE TABLE deck (deck_id INTEGER NOT NULL, deck_name VARCHAR(100) NOT NULL, deck_total_score INTEGER, deck_average_score FLOAT, PRIMARY KEY (deck_id), UNIQUE (deck_name))
deck_id	INTEGER	"deck_id" INTEGER NOT NULL
deck_name	VARCHAR(100)	"deck_name" VARCHAR(100) NOT NULL
deck_total_score	INTEGER	"deck_total_score" INTEGER
deck_average_score	FLOAT	"deck_average_score" FLOAT
user		CREATE TABLE user (user_id INTEGER NOT NULL, username VARCHAR(15) NOT NULL, password VARCHAR(15) NOT NULL, PRIMARY KEY (user_id), UNIQUE (username))
user_id	INTEGER	"user_id" INTEGER NOT NULL
username	VARCHAR(15)	"username" VARCHAR(15) NOT NULL
password	VARCHAR(15)	"password" VARCHAR(15) NOT NULL
user_deck_relation		CREATE TABLE user_deck_relation (correct INTEGER, time VARCHAR(100), quiz_count INTEGER, user_deck_relation_id INTEGER NOT NULL, "userUCR_foreignid" INTEGER NOT NULL,

Name	Type	Schema
		"deckUCR_foreignid" INTEGER NOT NULL, PRIMARY KEY (user_deck_relation_id), FOREIGN KEY("userUCR_foreignid") REFERENCES user (user_id), FOREIGN KEY("deckUCR_foreignid") REFERENCES deck (deck_id))
correct	INTEGER	"correct" INTEGER
time	VARCHAR(100)	"time" VARCHAR(100)
quiz_count	INTEGER	"quiz_count" INTEGER
user_deck_relation_id	INTEGER	"user_deck_relation_id" INTEGER NOT NULL
userUCR_foreignid	INTEGER	"userUCR_foreignid" INTEGER NOT NULL
deckUCR_foreignid	INTEGER	"deckUCR_foreignid" INTEGER NOT NULL

Three basic tables have been created namely user, card and deck. User table stores username and password. Deck table stores deck name, deck's total score (to aid in average

score calculation) and deck's average score. Card table stores card front and card back, along with the difficulty of the card as entered by the user. User deck relation is a table that stores information of users relating to a deck. It stores the user id and deck id, the last score of the quiz on the deck, the last review time and the number of quizzes that have been taken for average score calculation. Card Deck Relation table links cards to a deck.

Architecture and Features

The root folder contains the *app.py* file that contains the application and the controllers, *models.py* that contains the database models, *database.sqllite3* is the database file. The folder *templates* contains all the HTML file templates and the *static* folder contains the image for rendering on the website.

- Users can create an account and log in to see their dashboard. Empty fields cannot be submitted on both signup and login forms. If a user is trying to create an account with a username that already exists, it will show an error message. If a user is trying to log in incorrect username or password, it will show the required error message.
- On logging in, the dashboard is visible with a table that contains details: deck name, last review time, last score, the average score on deck, option to edit deck, delete deck and start the quiz on deck.
- If the user has logged in for the first time, this table will be empty. Below the table, there is an option to create a new deck. By clicking on this button, the user will be redirected to create deck page where the user can enter the deck's name and start adding flashcards to the deck.
- When the user doesn't want to add any more cards, the user can go to the dashboard. By clicking on edit deck, the option to change the deck's name or add more cards to the deck is provided. By clicking on delete deck, the deck will be deleted including all its associated cards.
- Finally, by clicking on Start Quiz, the user can start reviewing cards of that particular deck. First, the user is shown only the card's front with a text field asking the user to enter the card's back. The user is also asked to enter the difficulty level of the card. By clicking on submit, the user is shown both the card's front and the back. By clicking on next, the next card is shown. After all cards are done, the score is displayed. By going to the dashboard, now the user can see that score under the last score, the updated average score and the last review time.

Video

https://drive.google.com/file/d/1OP3qT-JklVq8LGNVV6blw7g45ywrDtXK/view?usp=sharing