Author

Shashwat
21f1001237
21f1001237@student.onlinedegree.iitm.ac.in
Pursuing B.Sc in Programming & Data Science & 2nd year student of B.Tech in Computer Science & Engineering
From Lucknow, Uttar Pradesh

Description

- FlashCard web application should allow a user to create a deck of cards for memorization.
- Each card should have a front, back part and a rating which is given at the time of review.
- Users should be able to review a deck to check their understanding of the cards.

Technologies used

Flask (framework)
Flask SQLAlchemy (ORM) (for support of SQL)
flask_restful (to design rest APIs)
Bootstrap (for styling)
FontAwesome (for icons)

DB Schema Design

user(user_id(PK), user_name , user_fullname, user_email, user_password)
deck(d_id(PK), d_name, d_score, d_review_time)
udecks(ud_id(PK), d_id(FK deck.d_id), u_id(FK user.user_id))
card(c_id(PK), c_front, c_back, c_rating(FK rating.r_id))
dcards(dc_id(PK), d_id(FK deck.d_id), c_id(FK card.c_id))
rating(r_id(PK), level, point)

Each user is a base entity which has multiple decks associated with it. Each card has a rating which is given at the time of review which is taken from the "rating" table. "udecks" and "dcards" maps the user with deck and deck with cards respectively.

API Design

User APIs- GET API for fetching user details like (user_name,user_email,user_decks) from user_id
POST API for creating a new user

Deck APIs- CRUD APIs on deck
Get details of deck like (deck name, cards in the deck) , update the deck name, delete the deck by giving user_id & d_id as parameter
Create a new deck for a user by giving user_id as parameter
APIs for getting deck score & deck last review time by giving user_id & d_id as parameter

Card APIs- CRUD APIs on card
Get details of card like (card front, card back), update the card details, delete a card from the deck by giving user_id, d_id, c_id as parameter

Create a new card for a deck by giving user_id & deck_id as parameters.
API for retrieving the last rating of the card by giving user_id & d_id as parameter.

Architecture and Features

Structure of the app is as follows :-

```
|-----application
          |------api.py                  All APIs are written here
          |------config.py               Development Configuration is done here
          |------controllers.py          All controllers are written here
          |------database.py             Setting up SQLAlchemy base & db
          |------deckService.py          To preserve sequence of cards when reviewed
          |------loginService.py         For Session Management
          |------model.py                Database Models
          |------notification.py         For success alerts
|------db_directory                      Database file(.sqlite) is stored here
|------static                            Contains css,js,images,fontawesome
          |------
|------templates                         Contains all the templates of web pages
          |------
```

Application contains following features :-

- A user can be created from the “Register here” link in the login page with username, email & password which is stored in the “user” table in dB. Passwords can also be updated if forgotten by providing required details.
- After login a deck can be created with a name and then cards can be populated to it. A deck can consist of a maximum of 10 cards (limit the no. of cards for scoring purposes) With cards containing a front and the information to memorise at back.
- Decks & Cards can be edited and deleted as per user’s choice. It is done by filling up the respective forms. Several constraints are implemented to prevent redundancy of data like two decks can’t have the same name for a user or two cards can’t have the same front in a deck.
- User Dashboard contains all the decks of the user with last reviewed time. From there a deck can be reviewed.

  Review Sequence- Cards in the deck are first filtered into 4 groups Unrated, Easy,Medium & Hard on the basis of previous rating. These groups are then shuffled individually and then the final sequence is generated as first Unrated-Hard-Medium-Easy. This list of lists is preserved in “deckService.DATA”.
- During review, cards are displayed in the above mentioned sequence one by one and a rating is to be provided by the user as to how he finds the particular card at that time.After rating a new card is shown till deck is finished.
- Deck Score and Review Time is updated accordingly and stored in the dB.
- A higher deck score means the user found the deck hard while reviewing.
- A deck can be reviewed any number of times.
- In case of invalid login session, the user will be redirected to the login page.

Demo username:- “vins” password-=”awdasda”

Video

https://drive.google.com/file/d/1wTk1zF9gRkoTcDVvJDBewdwZ_4kfR1gq/view?usp=sharing