

REPORT ON FINAL PROJECT

APPLICATION DEVELOPMENT I

FLASHCARD APP

Author :

NAME: Qureshi Mohd Sameem

ROLL NO.:21f1000868

EMAIL ID.:21f1000868@student.onlinedegree.iitm.ac.in

I am currently pursuing B.TECH in Computer Engineering from Dr. D.Y. Patil Institute of Engineering, Management and Research. Along with B.TECH ,I am pursuing Diploma in Programming offered by IIT MADRAS. Some of my hobbies include reading and playing outdoor games.

Description:

What is a Flash card? Do we need effective tools for memory training? Do we need a tool which may help us remember things in our own understanding? Well , Flash card will provide the solution to all above problems. The User can have multiple decks by which we can easily and effectively manage our memory training. System will automatically present one card at a time, and user needs to select a choice based on how well they know so that they can keep a score along with a card.

Technologies used

Flask

Flask-login

Flask-SQLAlchemy

Flask-Login provides user session management for Flask. It handles the common tasks of logging in, logging out, and remembering your users' sessions over extended periods of time

DB Schema Design :

The database design contains three tables namely card,user and deck table.

The detailed schemas are as given below:

1. user(card_id, front,back,score,deck)
2. Deck table (id,deck_name, user, date_created , score, is_public, last_rev)
3. Card table (user_name* ,deck_name* , card name, card remarks)

The create statements are as follows:

CREATE TABLE card (

```
card_id INTEGER NOT NULL,  
front VARCHAR(512) NOT NULL,  
back VARCHAR(512) NOT NULL,  
score INTEGER,  
deck VARCHAR NOT NULL,  
PRIMARY KEY (card_id),  
FOREIGN KEY(deck) REFERENCES deck (deck_name)  
)
```

```
CREATE TABLE deck (  
    id INTEGER NOT NULL,  
    deck_name VARCHAR(30),  
    user VARCHAR NOT NULL,  
    date_created DATETIME,  
    score INTEGER,  
    is_public BOOLEAN,  
    last_rev DATETIME,  
    PRIMARY KEY (id),  
    FOREIGN KEY(user) REFERENCES user (username)  
)
```

```
CREATE TABLE user (  
    id INTEGER NOT NULL,  
    username VARCHAR(30) NOT NULL,  
    password VARCHAR(300) NOT NULL,  
    date_created DATETIME,  
    score INTEGER,  
    PRIMARY KEY (id),  
    UNIQUE (username)  
)
```

HTML & CSS:

The required HTML codes and CSS codes are written for representing data that will be acquired using the controllers in a well-organized manner and easy to read format. Through these files only the users

will interact with the server.

CONTROLLERS:

The app part is designed in a way such that it will get the requests from the users and provide the users with the required output in the form of HTML along with CSS for styling.

- i. Sign Up – To create a new user
- ii. Login – To enable a user to access his content
- iii. Decks – To show user his cards
- iv. Add Deck – To create a new deck
- v. Edit Deck – To edit an existing deck
- vi. Delete Decks – To delete a deck
- vii. Cards – To show the cards inside a deck
- viii. Add Card – To create a new card
- ix. Edit Card – To edit an existing card
- x. Delete Card – To delete a card
- xii. Import Deck – To import a deck using a CSV file
- xiii. Export Deck – To export a deck in CSV format

API:

The purpose of the API part is it connects with the database file and fetch the data needed. The app part of controller doesn't have the permission to connect to the database directly. Instead, it requests

the API and the API returns the output to the app definitions. Similar to App definitions there are multiple

APIs present. Some of them are:

- i. SignupAPI – To check DB whether user name available
- ii. LoginAPI – To check the DB whether Login details are correct
- iii. DecksAPI – To add deck data into DB

iv. CardsAPI – To add card data into DB

INTRODUCTION VIDEO LINK:

https://drive.google.com/file/d/1lDdLo_sJYl90B9uUoKYRYlWYW6SG2Z1/view?usp=sharing