Author

ANSH KUSHWAHA
Roll No.: 21f1006019
21f1006019@student.onlinedegree.iitm.ac.in
From Varanasi, UP
Currently pursuing BE CSE from Panjab University, Punjab

Description

To design a login authenticated flashcard web application using python flask with reactive & responsive UI. For each FlashCard User can Review it as Easy, Medium OR Hard. Backend Jobs such as Daily Reminder, Monthly Progress Report. Export Decks & Export User Progress Report.

Technologies used

- Flask
  for running web application
- Flask-SQLAlchemy
  for connecting & querying database
- Flask-Security
  for login authentication
- Flask-RESTful
  for restful api
- Flask-Caching
  for caching
- Flask-SSE
  for server sent events
- Flask-Mail
  for mailing
- Werkzeug.Exceptions
  for HTTP exceptions
- SMTPlib
  for mailing
- Random
  for randomizing flashcard sequence
- Json
  for formatting json
- OS
- Gevent
  for coroutine
- Gunicorn
  for workers
- Redis
  for caching & batch jobs
- Celery
  for batch jobs

DB Schema Design

- users
  for storing users details
  - id (Integer, Primary Key, Not Null, Unique, Auto Increment)
  - email (String(64), Not Null, Unique)
  - password (String(16), Not Null)
  - fs_uniquifier (String(225), Unique)
  - correct (Integer, Not Null, Default=0)
  - incorrect (Integer, Not Null, Default=0)
  - roles (Relationship)
- roles
  for storing roles
  - id (Integer, Primary Key, Not Null, Unique, Auto Increment)
  - name (String(16), Not Null, Unique)
- users_roles
  for storing relationships between users & roles
  - id (Integer, Primary Key, Not Null, Unique, Auto Increment)
  - user_id (Integer, Foreign Key = users.id)
  - role_id (Integer, Foreign Key = users.id, Default=2)

- cards
  for storing cards
  - id (Integer, Primary Key, Not Null, Unique, Auto Increment)
  - question (String(64), Not Null)
  - answer (String(64), Not Null)
  - deck (String(16), Not Null)

API Design

RESTful API is created using Python's Flask-RESTful library.
APIs are implemented for users & cards

Architecture and Features

- main.py
- Project Report.pdf
- applications
  - controllers
    - api.py
    - controllers.py
    - webhooks.py
  - data
    - database.py
    - models.py
  - jobs
    - tasks.py
    - workers.py
  - utils
- database
  - database.sqlite3
- static
  - css
    - style.css
  - js
    - script.js
- templates
  - dashboard.html
  - error.html
  - index.html
  - play.html

- It has two login options (Administrator & User)
- Administrator Login Password is set to "P@ssw0rd"
- Administrator can view all users with their scores
- Administrator can view, create, modify, remove & export decks
- Users can view & review cards
- Users can export their progress report
- One card will be displayed at a time with 4 options
- Card can be reviewed as easy, medium & hard
- Correct response gives +2 points whereas incorrect gives -1
- Scores will reflect after exiting the session

Video

https://drive.google.com/drive/folders/1O6xYCMUs-vdMPYkf0Nr_O78Ss-HeAgZ5?usp=sharing