

Author:

Sarthak Gautam

21f1000864

21f1000864@student.onlinedegree.iitm.ac.in

I am currently pursuing B.Sc. CS from Post Graduate Government College along with this online B.Sc. Degree in Programming and Data Science. I love mathematics and computing.

Description:

In this Project, I created a basic web application that let's user decide the difficulty of the word and later display the score on the User Dashboard. The Application provides a user login System that is used for authentication in api too. A user is allowed to add/update/delete a Card. He can also update his profile and webhook url.

Technologies used:

Technology/Package	Use
Flask	For overall app
Flask_sqlalchemy, SQLite	Database
Vue, Javascript	Frontend
date and time, OS (Python)	To calculate date
flask_restful	API's
Celery	Backend job such as Daily reminders
Insomnia, POSTMAN	Test api's
flask_caching	Caching

DB Schema Design:**Profile_db**

Column Name	Column Type	Constraints
user_name	String	primary_key,unique,Not nullable
first_name	String	Not nullable
email	String	Not nullable
password	String	Not nullable
webhook_url	String	

deck

Column Name	Column Type	Constraints
word_id	Integer	primary_key,auto_increment
deck_name	String	Not nullable
Word	String	Not nullable
Meaning	String	Not nullable
last_access	String	Not nullable
score	Integer	Not nullable
user_name	String	Not nullable

The flask security database tables are kept separate as their are used only for authentication purposes. Also, Here only two tables are used for user and deck management in order to reduce the complexity/Constraints for add, update,delete. Also the main theme of the app is to separate flash card data of users by using user_name.

API Design:

The api's for this project were created from scratch even though some api's were created during previous project so as to meet new project requirements.

Common apis

get_email	loginapi
Signup	dashboardapi
profileapi	specific_deck_words
report_web	update_word
update_score	

Architecture and Features:

The project is largely contained in app.py file i.e. It contains controllers, DB, API's, Celery tasks, Caching etc. CSS, Javascript(including Vue) are present in the static folder. HTML files are present in the templates folder.

The App is a combination of Multi Page app along with single page components using vue-routers. HTML is rendered using controllers and later data/additional html is rendered using Vue. Vue gets/Post this data using fetch api. The user can have CRUD operations and Update is name, webhook url etc.

Video:

<https://drive.google.com/file/d/1Jjr92KPr1sngHsevudT0GbYyixVnEBW8/view?usp=sharing>