Author

Parv Pratap Singh
21f1002039
21f1002039@student.onlinedegree.iitm.ac.in
I am a third year B.Tech student working in developing skills in the domain of data science and machine learning. I am also interested in blockchain technology and its applications.

Description

For a good user experience this application should be easy to use and have a simple structure so the user can intuitively start using the application without any training.

Technologies used

The main frameworks used in the project are-

- Flask: For application code
- Flask-RESTful: For making a REST API
- Flask-SQLAlchemy: For connecting to the database
- Jinja2: For making templates
- Matplotlib: For making graphs for summary page
- Bootstrap: For styling
- SQLite: For database management
- Other dependencies for the frameworks

DB Schema Design

User Schema

Column Name	Column Type	Constraints
user_id	Integer	Primary Key, Auto Increment
user_name	String	Unique, Not Null
password	String	Unique, Not Null

The user schema has user_id as primary key with a unique user_name and password (not nullable).

List Schema

Column Name	Column Type	Constraints
list_id	Integer	Primary Key, Auto Increment
list_title	String	Not Null
list_description	String	
list_user_id	Integer	Foreign Key (User.user_id), Not Null

The list schema has list_id as primary key with list_user_id as the foreign key that relates it to the user schema. list_title is not nullable so a list can be easily identified and list_description doesn't have any constraints as a user might not want to give any details about it.

Card Schema

Column Name	Column Type	Constraints
card_id	Integer	Primary Key, Auto Increment
card_title	String	Not Null
card_content	String	
card_create_date	DateTime	Not Null
card_deadline	DateTime	Not Null
card_done	String	
card_done_date	DateTime	
card_list_id	Integer	Foreign Key (List.list_id), Not Null

card_id is primary key card_list_id to relate it to list schema. card_title and card_content are not unique as the user might want same type of cards. card_create_date and card_deadline are not nullable as they are required for the graphs and card_done_date is nullable as a card may not be completed yet. The type for the dates is chosen DateTime as it makes it more precise to compare dates for summary page and as the type Date is not supported by the REST API.

Summary Schema

Column Name	Column Type	Constraints
sum_id	Integer	Primary Key, Auto Increment
sum_cards_total	Integer	
sum_cards_done	Integer	
sum_cards_deadline	Integer	
sum_graphs	Boolean	
sum_list_id	Integer	Foreign Key (List.list_id), Not Null

sum_id is the primary key and sum_list_id relates it to list schema. sum_graphs shows if the list has any graphs made and the rest are for keeping track of the cards.

API Design

CRUD operations implemented for list and cards which take list_id and car_id repectively to return a json object. Create and read operations implemented on user which returns a user object in json format. Finally summary takes sum_id and returns information of lists and their cards in json.

Architecture and Features

The root directory of project folder contains application, templates and static folders along with a setup shell and requirements file to set up the project requirements, a README file for the know-how to use the project, a main.py file to run the application and once the setup is done, an sqlite database is added. The templates folder contains all html files and the static folder contains images and graphs used. The application folder contains models, validation, controllers and api all in different python files.

In the root directory, the docs folder has the pdf file for the project and the api folder has the yaml file.

The application starts with a signup page which redirects it to the login page after the user is signed up successfully, both pages have validation at frontend itself. The password is turned into a more secure string by the hashing algorithm- ‘sha256’. Once signed in, the user is redirected to the index page to create, edit and delete lists and cards. If the user wants to see the progress of his work then he can click on summary to be redirected to the summary page to see the graphs and statistics of the cards of each list.

Video

https://drive.google.com/drive/u/3/folders/1j8oXNFprlQRqpgAvEp8NHdadeUmRR6L9