Author

Gunanidhi Trivedi
21f1001843
21f1001843@student.onlinedegree.iitm.ac.in
Hi, I am a 3-year CSE Undergraduate @Shri Shankaracharya Technical Campus, Bhilai CG.
I am an upcoming data scientist.
I love learning and exploring new topics and fields.

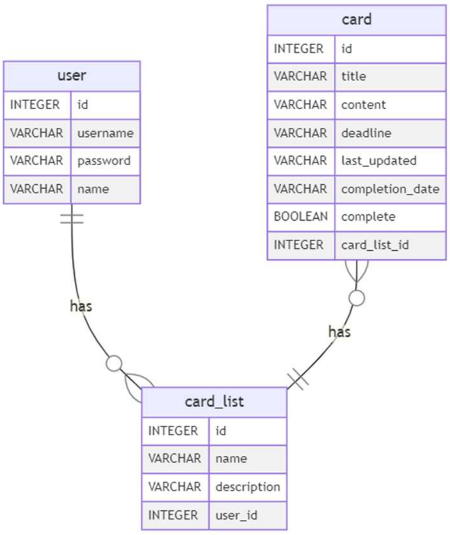
Description

My project is all about list and card management. One can easily use it in many ways example - schedule management, project management, can also work as reminders etc.
for future I can also add a reminder notification for deadlines.

Technologies used

- **flask** - it is a small and lightweight Python web framework that provides useful tools and features for development
- **flask_sqlalchemy** – used it because it helps to accomplish common tasks easily.
- **flask_login** – used for user session management.
- **Matplotlib** – used for creating static visualizations.
- **Datetime** – used for accessing current date.
- **flask_restful** – used for making API.
- **werkzeug.exceptions** – used for error handling.
- **json** – used for formatting API response.

DB Schema Design



user – table for storing user details

card_list – table for storing list information

card – table for storing card details

user and card_list have one to many relationships because one user can have many lists but each list can have only one user.
Card_list and card have one to many relationships because one list can have many cards in it but each card can associate with only one list.

API Design

User operation

- /api/user/{username}
  - **GET** -> *get user detail*
- /api/user
  - **POST** -> *add user*

List operation

- /api/card_list/{user_id}/{card_list_id}
  - **GET** -> *get list detail*
  - **PUT** -> *update list detail*
  - **DELETE** -> *delete list*
- /api/card_list
  - **POST** -> *add* list

Card Operation

- /api/card_list/{user_id}/{card_list_id}/{card_id}
  - **GET** -> *get card details*
  - **PUT** -> *update card details*
  - **DELETE** -> *delete a card*
- /api/card_list/{user_id}/card
  - **POST** -> *add a card in a list*

Architecture and Features

# **Folder Structure**

- `database.sqlite3` it is sqlite DB. It can be anywhere on the machine.
- `main.py` it is the main file which contain all the code (Configuration, Controllers, APIs, etc)
- `static` - default `static` files folder. It serves at '/static' path.
- `static/style.css` Custom CSS. You can edit it.
- `templates` - Default flask templates folder

#Features

- User authorization (login, sign-in).
- Card and List CRUD operation.
- Move the card from one list to another.
- Summary page summarize each list for better understanding.
- Dashboard shows card status as “pending”,” completed”, “deadline passed”.
- APIs for interaction with user, lists and cards.
- User’s username is displayed on the dashboard.
- User cannot access home page and can’t do CRUD operation without login.
- Without login, user can only access login and sign-up page.
- User can add required number of lists and cards in the application.
- A notification is displayed below navbar when an operation is accomplished.

Video

- https://drive.google.com/file/d/1bY5XCvPrLysPf7RUWn0wSNdMaPk4key8/view