Kanban App

Author

Kiran Prasath D K
21f1002124
21f1002124@student.onlinedegree.iitm.ac.in
I am pursuing Btech CSE, in VIT chennai. My main field of interest are robotics. And I am the Vice Captain of Dreadnought Robotics a special team of VIT Chennai.

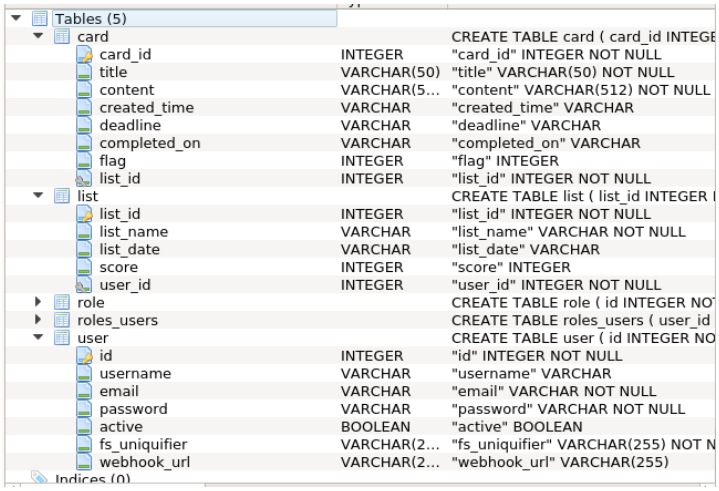
Description

The Kanban app is a list based app where we have to create list based on the requirement, such as reminder. We can then add cards related to reminders to this list, these cards will help the person stay in track of his objective. We need to create Flask and Vue based webapp with options to create,update,delete and export , card/list and also give reminders about the same

Technologies used

Flask: Allow us to create an app based on the flask modules, that makes website making so much easier.
Flask-Restful: To implement API
Flask-CORS: For Cross-Origin Resource Sharing
Flask-Security: For authorisation and authentication
Flask-SQLAlchemy: To connect and update Sqlite database.
Redis – For caching
Celery[Redis] – For workers such as export.
Vite – For setting up and running vue environment
tailwindcss – for styling
pinia – Vue Store for global access of variables from anywhere in the app
Vue Router – for Routing various endpoints to views.

DB Schema Design



The 4 tables are role, user, card, list.
The user table has id(primary key), username, email, password, active, fs_uniquifier based o nthe requirements of the flask security
the Role table has the role assigned to each user.
The list has list_id(primary key), list_name, list_date, score, user_id(foreign key)
The card has card_id(primary key), title, content, deadline, completed_on, flag, list_id(foreign key to list table)

API Design

UserAPI – methods:”GET”,”DELETE”
ListAPI – methods:”GET”,”PUT”,”DELETE”,”POST” for creating, update and deleting list.
CardAPI – methods:”GET”,”PUT”,”DELETE”,”POST” for creating, update and deleting card
UserDataAPI – gives all the details of the user along with the cards and list that belongs to the user.
UserListAPI – returns the lists under a user.
ListCardsAPI – returns the cards under a list.

Architecture and Features

The project has two files backend and frontend.
The backend file has the main.py file in the root directory. Files have been created to organise, such as application(which contains the api, config, controller, database, worker, task, validation) , db_directory(contains database), static, template(for testing backend).
The frontend file has all the configuration files and the requirements for the vue environment. The src forlder has all the vue files. The src file contains main.js(which creates the the app and adds router, pinia) and App.vue. The router folder inside the router has the vue router(that links routes to view), stores(contains auth, data store), views(has all the views such as add card, add list, list and card display and editing).

The features that have been implemented:
1) Flask-Security, token based authorisation and authentication.
2) List Management – Create, update, delete and export lists
3) Card Management – Create, update, delete and export cards
4) API Performance

Video drive link:
https://drive.google.com/file/d/1lWoIjBEUo7uUfSFaPbqV7BtHChWFrd3N/view?usp=share_link