Author

Chirag Goel
21f2000540
21f2000540@student.onlinedegree.iitm.ac.in
I am a competitive coder and MERN Stack developer and ml enthusiast.

Description

The goal of this project is to create a web application which helps the user to track their activity.
A kanban board is an agile project management tool designed to help visualize work, limit work-in-progress, and maximize efficiency (or flow). In this app you can add task move task to different list and perform CRUD operations on cards and lists

Technologies Used

The application uses Flask, jwt , SQLAlchemy and Vue JS to implement core functionalities. Vue JS is used for templating and reactivity in HTML. Bootstrap is used for UI Design and responsiveness. Vue Components have been used to ensure maximum reactivity and responsiveness. Vue Chart Js is used for making carts on frontend. Redis has been used for Caching along with the Flask Caching library to ensure faster results at optimizable places. Celery has been used to implement background jobs for processes like monthly reports and daily reminders and also for exporting job. SMTP server is used for sending mails. All these async jobs are done by celery.

DB Schema Design

Table User

Column Name	Type	Constraints
User_id	Integer	Primary Key, Auto Increment
Public_id	String	Unique
User_name	String	Not Null
Name	String	Not Null
Password	String	Not Null
Email	String	Not Null

Every username is uniquely identifiable.

Table List

Column Name	Type	Constraints
List_id	Integer	Primary Key, Auto Increment
List_name	String	Not Null
User_id	Integer	Foreign key (User table)

It stores details of all the list

Table Cards

Column Name	Type	Constraints
Card_id	Integer	Primary Key, Auto Increment
List_id	Integer	Foreign key (List table)
User_id	Integer	Foreign key (User table)
Card_title	String	Not Null
Card_content	String	Not Null
Card_completion_date	String	

Card_deadline	String	Not Null
Card_status	String	Not Null

It stores details of all the card of particular list and user.
List table and card table are made separate to minimize redundancy and ease in access of data.

API Design

The application has a robust API with multiple endpoints protected by authorization with JWT for optimal security. Further the API allows for creation, reading, updation and deletion for all List and Cards, along with the ability to fetch statistics for the user. Api end points are also created to trigger async functions of backend like exporting cards or lists.

Architecture and Features

There are 2 folders backend and project. the backend folder consists of all the backend code. Project folder consists of frontend in that all the components are in src folder then component folder and all assets like svg or images are in assets folder and RouterComponent consists of all the routes.
The application has a login system, dashboard management (CRUD on lists and cards. A card review system. Fully responsive UI. Summary page for all lists and particular list also .Cards can be moved between multiple lists.

Video

https://drive.google.com/file/d/1G7KkslQW3qg6VYTX_vXa-loBqQC0xrQ0/view?usp=sharing