 Minimal Kanban

Intro :

    Name : R Sashi Adhithya
    Roll : 21F3000611
    Mail : 21f3000611@ds.study.iitm.ac.in
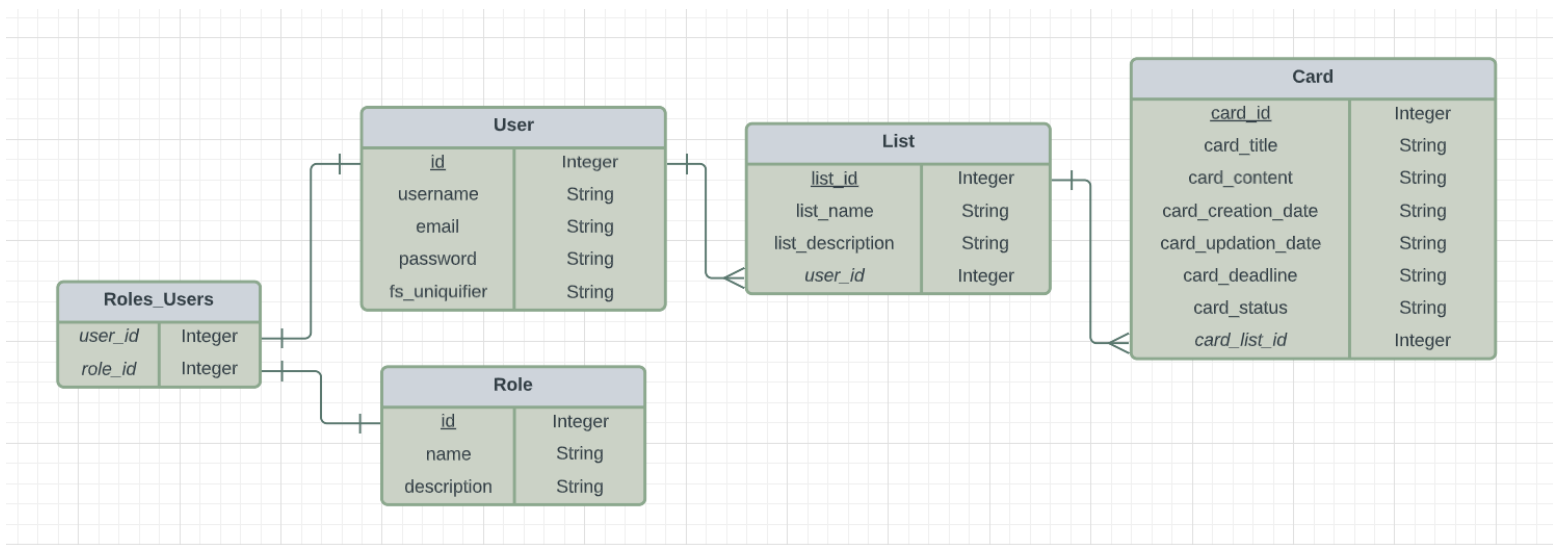
Description :

    Minimal Kanban Board provides a comprehensive platform for
    organizing your to-do lists and tasks, complete with deadlines
    and the ability to update, delete, and rearrange items as needed.
    Utilize this powerful project management solution to effortlessly
    visualize your work in progress and optimize your workflow for
    maximum productivity and efficiency.

Technologies and Frameworks :

- Frontend:
    Vue-CLI, Vue3, Vue-Router@4, FetchAPI, Chartjs, HTML/CSS/JS,
    Adobe Photoshop
- Backend:
    Python, bcrypt, Flask, Flask-RESTful, Flask-Security-Too,
    Flask-Caching, Flask-SQLalchemy, sqlite3, weasyprint
- Development Environment:
    Webstorm, Pycharm/WSL

DB Schema Design :

    One to Many Relationships for Data Access,
    User→List→Card
    User→Role→Roles_Users for Flask-Security-Too

The diagram shows an Entity Relationship Diagram with the following entities:

Roles_Users
| user_id | Integer |
| role_id | Integer |

User
id	Integer
username	String
email	String
password	String
fs_uniquifier	String

Role
id	Integer
name	String
description	String

List
list_id	Integer
list_name	String
list_description	String
user_id	Integer

Card
card_id	Integer
card_title	String
card_content	String
card_creation_date	String
card_updation_date	String
card_deadline	String
card_status	String
card_list_id	Integer

Architecture and Features :

The Minimal Kanban Board consists of a frontend built with Vue3 and a backend using the Flask-RESTful framework. The frontend makes fetch calls to the backend for authentication and CRUD operations using the Fetch API. Hosted as a single page application with Vue-Router@4, the frontend was developed and prototyped using Vue-CLI on Webstorm and served locally. Logos and assets were created and edited using Adobe Photoshop.

The backend handles API calls and implements token-based authentication with Flask-Security-Too. Flask-Caching throughout the app which uses Redis for storing data and asynchronous tasks, such as getting started, regular updating, daily reminders and monthly reports, were processed using Celery and Redis.

Key Features:
- Secure login framework with Token-Based Authentication
- CRUD operations for lists and tasks on the kanban board
- Summary page with illustrative graphs on user's progress
- Redis caching for faster loads with adjustable cache expiry
- Asynchronous tasks through Celery for improved user experience
- Ability to import/export tasks and lists as .csv (csv via user registered email)
- Daily reminders and monthly reports (pdf via user registered email)