Project Report

Author

Ninad Aithal
21f1006030
21f1006030@student.onlinedegree.iitm.ac.in

I am a student currently in 3rd year B.Tech in Robotics, AI, & ML studying in Srinivas University, Mukka, Mangaluru. I belong to a village called Perdoor in Udupi, Karnataka

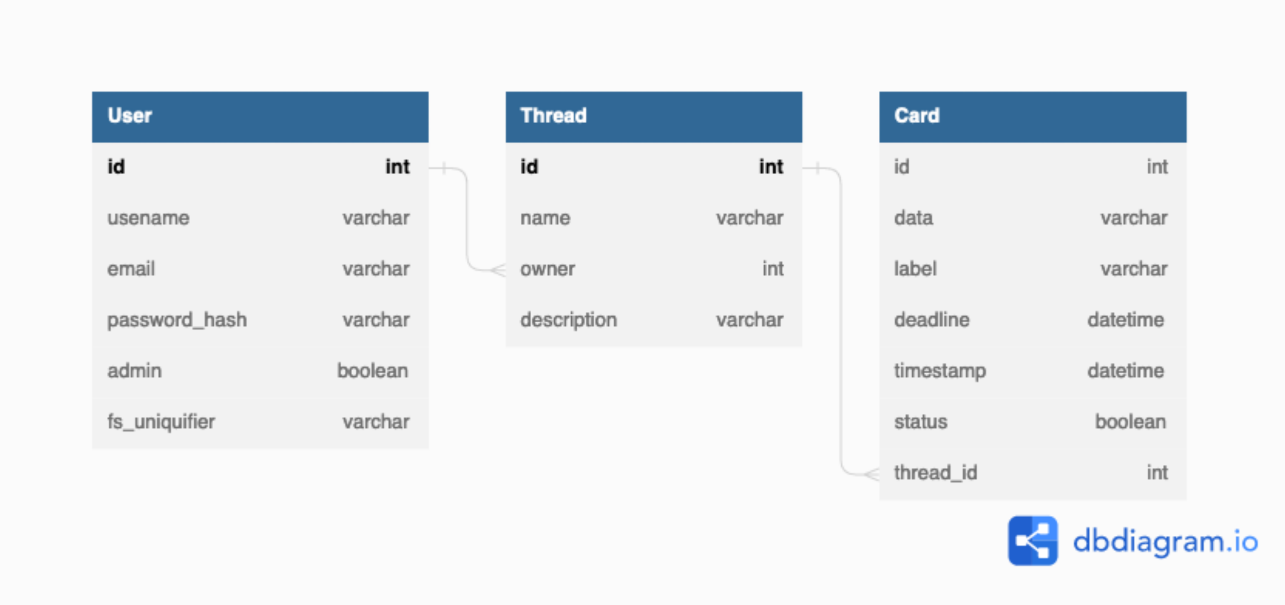
Description

A kanban app to record day to day life activities as a card which can be grouped into threads where a user can register and add threads and card values in it and be provided with a graphical representation of the tasks completed. Also, can manipulate the threads and cards eg: updating it or deleting it., moving a card from one thread to another.

Technologies used

- flask. - framework used to create the web app using python
- flask-sqlalchemy - tools and methods to interact with database from flask application
- flask_bcrypt - enables hashing and related utilities
- flask_login - access control and session management
- flask_restful - construct API’s
- Flask-JWT-Extended - for token based access
- Flask-Celery - to run processes that run the background jobs
- Flask-Caching - for caching
- Jinja2 - for creating monthly reports
- Pandas - for converting database into csv for export

DB Schema Design



User	
id	**int**
usename	varchar
email	varchar
password_hash	varchar
admin	boolean
fs_uniquifier	varchar

Thread	
id	**int**
name	varchar
owner	int
description	varchar

Card	
id	int
data	varchar
label	varchar
deadline	datetime
timestamp	datetime
status	boolean
thread_id	int

3 tables : User , Thread, Card
2 One-to-Many Relationships : User.id < Thread.owner & thread.id < Card.thread_id
If we delete a Thread all the Cards associated with the Thread are deleted due to cascading operation

API Design

Api Classes:

- UserAPI - get, post, put, delete
- ThreadAPI - get, post, put, delete
- CardAPI- get, post, put, delete, patch
- AllThreadAPI - get
- AllCardAPI - get
- CurrentUser - get
- LoginUser - get, post
- TransferCard - put
- TaskComplete - get
- AllThreadCSV
- AllCardCSV
- AllCSV
- Summary
- SummaryThread
- CardStatus

Celery tasks

- all_threads_csv
- all_cards_csv
- all_csv
- summary_data
- summary_thread_data
- monthly_new_thread
- welcome_email
- monthly_report
- daily_remainder

Architecture and Features

my_app folder and main.py are inside the zip folder
To run the web app run the main.py file (refer ReadMe.md)
my_app contains

- ReadMe.md
- Static folder - Contains all the javascript css and images
- Template folder - contains all the templates used for emails
- test.sqlite3 - database
- __init__.py - initialises the app
- data_access.py - for caching
- Routes.py - for flask routes
- Api.py - contains all the api’s
- Tasks.py for implementing celery tasks

Default features like CRUD operations on threads and cards has been implemented and a graphical representation of thread and card is being displayed on summary page.
For creating new threads & cards a cue-router is being called whereas for updating and deleting modals are being used
For rendering of graphs chart.js is being used

Video

https://drive.google.com/file/d/18LKowjyTO8ml_a-bxaIcru-N2G3Ess3e/view?usp=sharing