

Author

Rishika Singh

Roll no- 21f1002631

21f1002631@student.onlinedegree.iitm.ac.in

I am a student and my enthusiasm to learn new skills has drawn me towards this degree.

Apart from this degree I'm pursuing BTech and currently am in 4th year in information technology.

Description

I have created a multiuser task tracking web application for uploading task cards, their deadline and related information. User will also get regular updates on deadline and monthly report which will help the user in planning their tasks accordingly.

Technologies used

- API using python-flask
- VueJS advanced with CLI for user interface
- sqlite for database
- redis for caching
- celery and redis for batch jobs
- CSS and bootstrap for styling
- Matplotlib for plots and graphs
- Pandas for exporting

DB Schema Design

In the Kanban application 3 DB Models are used to store the data which are as follows:-

- **USER (DB Model)-**
The USER model table comprises of these columns- **user_id** which is primary key used to store unique id for individual user, **name** it is used to store the username of the user, **email** it is used to store the email id of the user and is a login credential, **password** which is used to store the password of the user login.
- **LIST (DB Model)-**
The BLOG model table comprises of these columns- **lid** which is primary key used to store unique id for individual list, **uid** is also imported from the users table as a foreign key to keep track record of the lists added by different users, **lname** is used to store the title of the list, **description** is used to store the description of the list, **trend** is used to store the relative url of the plot image for respective user.
- **CARD (DB Model)-**
The follow model table comprises of these columns- **cid** is primary key here used to uniquely determine each card, **lid** imported from the list table as a foreign key, **title** is used to store the title of the card, **content** is used to store the description of the list, **deadline** is used to store deadline of the task, **completed_flag** is used to store whether

the task is completed or not and **deadline_passed** is used to store whether deadline has passed or not.

API Design

In the Kanban application following classes were used to implement API:-

- **Login and Signup** - Used for user login and signup
- **Dashboard** – After login user lands on dashboard where all the lists and cards entered by user are listed.
- **List** – Contains API for entering new list, updating the existing list, deleting any existing list or exporting the list. Contains methods PUT, POST and DELETE.
- **Card** – API for viewing specific card details, entering new card, updating the existing card or deleting any existing card. Also contains option for exporting the card. Contains methods GET, PUT, POST and DELETE.
- **Summary** – Contains API for summary page which contains list and their respective bar graph depicting trend.

Architecture and Features

The project folder has sub folder project for Vue CLI, resource.py file containing API, models.py file for implementing model, jwt_trial.py for tokens, worker.py and tasks.py for implementing batch job using celery and redis.

The Features of the project are:-

- **Login/Sign up page** : For login and registration of new users
- **Dashboard** : Displaying all the list and cards added by respective user
- **List** : CRUD operation on list
- **Card** : CRUD operation on card
- **Summary Page** : Containing the brief summaries of list along with their bar graphs
- **Token** : Token implemented for security during login and signup
- **Export** : Exporting details of cards, list and dashboard
- **Caching** : Caching of data
- **Daily reminder** : Batch job for daily reminders for deadline and task completion via mail
- **Monthly progress report** : Batch job for sending monthly progress report via mail in HTML form

Video Link

https://drive.google.com/file/d/1RN1IzISpRrnWjpPX_2mnTI7EJx0_s96h/view?usp=sharing