## Author

Sarthak Rastogi
21f1000389
21f1000389@student.onlinedegree.iitm.ac.in
Undergrad student working as an NLP intern at Prodigal and researching knowledge-infused language models at University of South Carolina.

## Description

The project involves a Kanban board application that allows users to create, edit, export and delete tasks and the lists containing them. The app also needs to regularly generate progress reports and send reminders.

## Technologies used

- Flask API, Flask-Caching, Flask-Login.
- SQLite3 and Pandas for data management
- Celery workers and beat; Crontab for scheduling reminders
- Matplotlib for generating plots and Base64 for encoding them
- SMTPLib for sending emails and MailHog as a local SMTP server
- VueJS for frontend, Axios for API calls, Bootstrap for all styling

## DB Schema Design

The database contains three tables:

1. Task table:
    a. **id** (primary key), **title** – Title of the card, **content** – Task description, **deadline** – Due date
    b. **completed_flag** – Boolean on whether or not the task has been completed
    c. **list_id** – Name of the list that the task belongs to
    d. **created_at, last_updated_at, completed_at** – Timestamp of when the task was created, last updated and completed respectively
    **e. username**
2. Lists table:
    a. **list_id** (primary key)**, list_title** – Name of the list
    b. **created_at, last_updated_at** – Timestamp of when the list was created and last updated respectively
    **c. username**
3. Users table:
    a. **id** (primary key), **username, password**

## API Design

**/ -** Landing page; **/plots –** Progress Report; **/signin; /signup; /kanban -** kanban board

**Task specific API calls:**
**/insert_task –** Gets all task details and adds a card. Inserts the card into the tasks table.
**/edit_task -** Returns the task information for updation by the update() function below
**/update_task –** Gets the new task info from the HTML form and updates the table with it.
**/delete_task** - Deletes the task from the table using its id
**/export_task-** Calls the async Celery task export_a_task() which retrieves the card's row from the DB table and saves it as a CSV with Pandas.
**/export_all –** Calls the async Celery task export_a_board() which saves the entire board as a CSV.

**List specific API calls:**
**/insert_list** - Get all list details and adds a list into the lists table.
**/edit_list -** Returns the list information for updation by the update_list() function below
**/update_list –** Gets the new litt info from the HTML form and updates the list table with it.
**/delete_list** - Gets the row for the list using its id for deletion in the pre_delete_list() function.
**/pre_delete_list –** Checks the user's choice of whether they want to delete all the tasks in the list or move them to another list. Also takes this other list as text input. Depending on the choice, it either deletes all these cards from the table or changes their list name.

## Architecture and Features

**Scipt.js**: Axios calls to API endpoints and VueJS reactive rendering.
**App.py:** Contains config details and above mentioned API call functions.
**Templates:**
**home.html:** Landing page, m**ain.html:** Forms, **a.html**: Report, **signin.html, signup.html**

**User management and security features:** Users can sign up and login using a password, and view only their own Kanban board.
**List management features:** Creating, editing, exporting and deleting lists. Before list deletion the user is prompted to save tasks in another list. Exporting is implemented as as async task. Caching is used on the dashboard.
**Task management features:** Creating, editing, exporting, moving and deleting tasks. Task cards show their title and description, deadline, last updated dates and whether or not it's been completed.
**Summary page:** List-wise progress charts of tasks, and lists of completed and overdue tasks. Implemented as an async task.
**Reminders:** A Google Chat message containing a list of the pending tasks is sent every evening. A monthly report is also sent as HTML via email.

## Video

https://drive.google.com/file/d/1antI3rNrK-CrRQaN4XqEmW2_yg3jhaN6/view?usp=sharing