## Author

Name: Maheedhar A
Roll number: 21f1006544
student email: 21f1006544@student.onlinedegree.iitm.ac.in
I am a student of SSN College of Engineering , Chennai. I am very enthusiastic full stack developer. My hobbies are listening to music, coding.

## Description

From the problem statement we received from IIT , I understood that we need to make a website which allows the user/client to add and monitor tasks and allows him/her to keep track of activities amidst their busy schedule using Vue.js for the frontend and Flask API for the backend

## Technologies used

In my code, I have used the following technologies:
1) flask, flask-sqlalchemy ,
2) sqlite ,
3) Vuetify ,
4) Chart.JS ,
5) os library from python
6) datetime library from python,
7) HTML,
8) CSS and
9) Bootstrap
10) Vue.js
11) Celery, redis and smtplib
12) Pandas
13) flask-jwt-extended

Their applications:

1) Flask and flask-sqlalchemy for basic routing and database related operations and to CORS access to the Vue.js application
2) Chart.js for plotting the graphs which shows the progress of the user.
3) Os library to set the secret key for our application.
4) HTML,CSS,Bootstrap to design beautiful and interactive web pages.
5) Datetime library to get date and time in format of datetime with respect to sql- to set expiry for javascript web tokens
6) Vue.js, Vuetify for creating frontend and the interface to connect with the flask API.
7) Celery, redis and smtplib to schedule asynchronous tasks and to create reports and the user
8) Pandas to export the trackers in CSV format.
9) flask-jwt-extended to create and manage javascript web tokens.

## DB Schema Design

Database Structure:
There are totally three tables in my database:

i) client
This table stores the data of clients who have created an account in my application.
The attributes of this table are:

1) uid: This is the primary key of this table and has an auto increment feature .
2) uname = this stores the name of the user who registered on our website and has not NULL feature enabled
3) Mail= this stores the email-ID of the user who registered on our website and has not NULL feature enabled
4) password= this stores the password of the user's account who registers on our website, which the user has to input later to login to the website and has not NULL feature enabled
5)date_created= this stores the date of account creation
ii) list

This table stores the information about the lists the user creates to fill with tasks.
The table includes the following attributes:

1) l_id= this stores the id of the tracker the user created. This is the primary key for the list table and has auto increment enabled.
2) u_id= this stores the id of the user who created that particular list, This is a foreign key reference to the uid attribute of the client table.
3) l_name= this stores the name of the lists the user created. It has not NULL enabled.
4) no_of_tasks= this stores the number tasks the user created in that particular list.
5) no_completed= this stores the number of completed tasks in that list.
6) no_crossed_deadline = This stores the number of tasks that has crossed the deadline

iii) task
This table stores the tasks the user adds into a particular list.
The table includes the following attributes:
1) t_id: this stores the id of the task that the user logs into the tracker. This is the primary key for the task table and has auto increment enabled.
2) l_id: this stores the id of the list the user created. This is a foreign key for the log table table that references the tracker_id of trackers table.

4) title = this stores the name of the task .
5) description= this stores the description of the task wants to input.
6) deadline = This stores the deadline in which the task has to be completed. The deadline can be updated
7) status= This stores the status of the task. If the task is completed, status is set to one, else it is set to 0.

## API Design

I have created the API for the user to register and login on the webpage. I have created API to create,update , delete List and to view the summary of his activities .
I have created API for the user to reset his password, create, update, delete tasks.
I have created all these API using flask.

## Architecture and Features

## The Architecture of my project folder:

Project folder
|_ backend

```
        |_ static
        |_ templates
        |_ flask
        |_ application.py
|_ frontend
        |_ node_modules
        |_ public
                |_ index.html
        |_ src
                |_ App.vue
                |_ assets
                |_ components
                |_ views
                |_ router
                |_ store
                |_ utils
                |_ plugins
                |_ main.js
        |_ package.json
```

Features implemented:

1) Register your account in the webpage to track your progress.
2) Login page: This searches the database for your credentials and helps you sign in to the website and provides you with a javascript web token which you require for accessing the application's core features.
3) Dashboard page: A home page that welcomes you to the website and gives a view of your lists and the tasks inside your list.
4) List creation page: This page uses vuetify forms and helps the user create list for his own.
7) List update page: This page allows the user to edit the list he has created and updates it in the database.
8) List delete page: This page allows the user to delete an entire list. Deleting a list deletes all the tasks it has inside the list.
9) task creation page: Users can add tasks to a particular list in this page.
10) Summary page: This page provides information about all the lists the user has created. This page contains the information about deadlines as well as graphs to indicate the status of the activities.
11) task update page: This page allows the user to update the task if the deadline has changed or if the user wishes to edit the details of the task.
12) task delete page: This page allows the user to delete a particular task in the list.
13) I have created asynchronous tasks using celery to send monthly reports to the user and to remind the user if any of his tasks has crossed the deadline.

Additional features:

1) Styling and aesthetics: I have included the styling and aesthetics using CSS and vuetify to make the page stunning and aesthetically pleasing.
2) A working forgot password page: I have worked on making the forgot password page work using the flask API and vue.js.

## Video

Link to the video : [Presentation Link]