Author

Name: Sanit Arora
Roll no.: 21f1002820
Student email: 21f1002820@student.onlinedegree.iitm.ac.in

Description

This is a quantified self application, where user can create trackers and log events into them. CRUD operations need to be implemented on trackers and logs. I have used Flask for the backend and Vue JS for the frontend, both on different servers. All calls to the server are being made through api requests and the result is displayed in VueJS templates.
Asynchronous tasks and server side events are also implemented mainly for sending daily log reminders and monthly reports. I have used celery workers and redis database to accomplish this task.

Technologies used

Frontend technologies

- **HTML** - as a markup language for documents
- **CSS** - For styling
- **Vanilla JS** - For reactivity
- **VueJS** - For reactivity and client side computations

Backend technologies

- **Flask** - as a framework for backend logic
- and server
- **Gunicorn** - To handle multiple http requests at a time
- **SQLite** - As a database
- **Flask SQLAlchemy** - As an ORM to interact with the database
- **Celery** - To handle asynchronous tasks
- **Redis** - To handle message queues and store messages

DB Schema Design

Tables

User

Attributes

id(uniques for every user), username, email(uniques for every user), password(used bcrypt to hash and store passwords), trackers(points to the tracker table), report_format(user preference for monthly reports)

Tracker

Attributes

id(unique for every tracker), name, description, tracker_type, settings(points to the settings table), logs(points to the logs table), user_id(foreign key points to user table), last_logged

Logs

Attributes

id(unique for each log), tracker_id(points to tracker table), note, value, user_timestamp, db_timestamp(when log was added to db)

Settings

Attributes

name(name of the setting)

Options

(A many to many table between tracker and settings)

Attributes

Tracker_id, settings_name

Architecture and Features

The code consists of two folders namely backend-code and frontend-code. The backend-code contains all the server side files with main.py as the entry point. Inside the app folder are all the application files which includes api, templates, static files, etc.
The frontend-code contains the vueJS files inside the tracku folder. The main HTML file (index.html) is present inside public folder and the main js file (main.js) and App.vue is present inside the src folder. Components and views are present inside their respective folders.

Features implemented

- JWT authentication (using flask jwt-extended)
- APIs using flask restful

- Charts and graphs using Vue-charttjs
- Add to desktop feature using vue-pwa
- Notifications using vue-notify
- Daily evening reminders using celery, celery beat, redis and google chat api(webhooks)
- Monthly reports using celery, celery beat and redis
- Export csv on every page using event stream

Video link - https://youtu.be/O0F12PAFcuo