Project Report

Objective: The objective of the function is to create a quantified self-app, which is an all-in-one self-tracker application.

Approach: For Security and Login framework I have used flask-Security, which also provides a feature of Token based authentication to the web application

I created a backend consisting of three major tables, “user”,” trackers”, and “value”. The 1st table consists of the details of the registered users(username, email, password, active status, etc.).2nd consists of the trackers’ trackers, so this includes the username, tracker, its type, and the time its created. While the last table consists of the values corresponding to different trackers created by the users. It includes 2 details values and a Note.

Then keeping the principle of “Separation of Concern” in mind I used Vue JS components for the frontend part, which were making the API calls to the backend to retrieve information

I have also cached certain APIs to improve the overall performance of the application.

Details: The backend part is connected to the frontend part using the Rest API calls whose documentation can be found in the api folder in the “documentation.yml” file.

The user can choose between different types of trackers for logging the values. A user can perform “CRUD” operations both on the trackers and the values tables. The user also gets the option of exporting any of the tables as a CSV, a confirmation of which you get as a Webhook message

The user gets a daily reminder to log his/her activities using the tracker on Webhook. The users also receive monthly reports to give them details of the trackers they have created in an HTML format. I have used Mail hog to send dummy emails.

The Styling and the aesthetics part of the application are done using the bootstrap CSS

The following link consists of an attachment to the video describing the project

Following command line codes are needed to run all the features of the program

- HTML: python3app.py
- Redis Server: redis-server
- Save celery tasks: celery -A tasks.cel beat --max-interval 2 -l info
- Implementing celery tasks: celery -A tasks.cel worker
- Mail Hog: ~/go/bin/MailHog

Link: https://youtu.be/CE8TTMFjJVM