

## Akshay Shikhar Srivastava

21f1000111

21f1000111@student.onlinedegree.iitm.ac.in

I hail from Ghazipur, Uttar Pradesh, India. Apart from the BS Degree from IITM, I did my undergraduate in Psychology (Hons) from Banaras Hindu University, Varanasi. I enjoy every learning experience and focus more on the journey rather than the destination.

## Description

The LOGIT' is a quantified-self application, which is used to keep a track of users activities via continuous logging. The user makes their own trackers that can be used to log specific value related to that particular tracker only. User can observe the logging history in the form of trend lines. CRUD operations can be performed easily using an aesthetic frontend.

## Technologies used

1. **Flask:** In the project backend is built in Flask.
2. **Flask-restful:** It has been used to deal with everything related to APIs. This is an extension to flask in order to build REST API.
3. **Vuejs (CLI):** It has been used for frontend development.
4. **Vuex:** It has been used in the project for state management purpose.
5. **Vue-Chartjs:** This package of Vuejs is used particularly for generating responsive charts in application dashboard.
6. **Many more modules (in flask) and packages (in Vuejs) are used as required.**

## DB Schema Design

A single database (quantified\_self.sqlite3) is created having multiple tables. It has three tables related to the application.

1. **credentials :** This table stores information about users including name, email and password (hashed)
  - user\_id: [primary key, string]
  - user\_name: [string]
  - user\_email: [string]
  - password: [string]
2. **trackerlist :** This table stores information about trackers and it also captures the relationship between user and their trackers.
  - user\_id: [foreign key, integer]
  - tracker\_id: [primary key ,integer]
  - tracker\_name: [string]
  - tracker\_desc: [string]
  - tracker\_type: [string]
  - tracker\_pri\_question: [string]
3. **logs:** This table stores log values related to every tracker. It captures relationship b/w trackers and their logs.

- tracker\_id: [foreign key, string]
- log\_id: [primary key, integer]
- log\_value: [string]
- remark: [string]
- timestamp: [datetime]

## API Design

The API functionality is written inside 'API.py'. 'GET', 'POST' and 'PUT' API are created and used extensively. Every GET API returns the output in json format back to the function whereas POST and PUT API also receive json data. Appropriate HTTP status code is returned to the function.

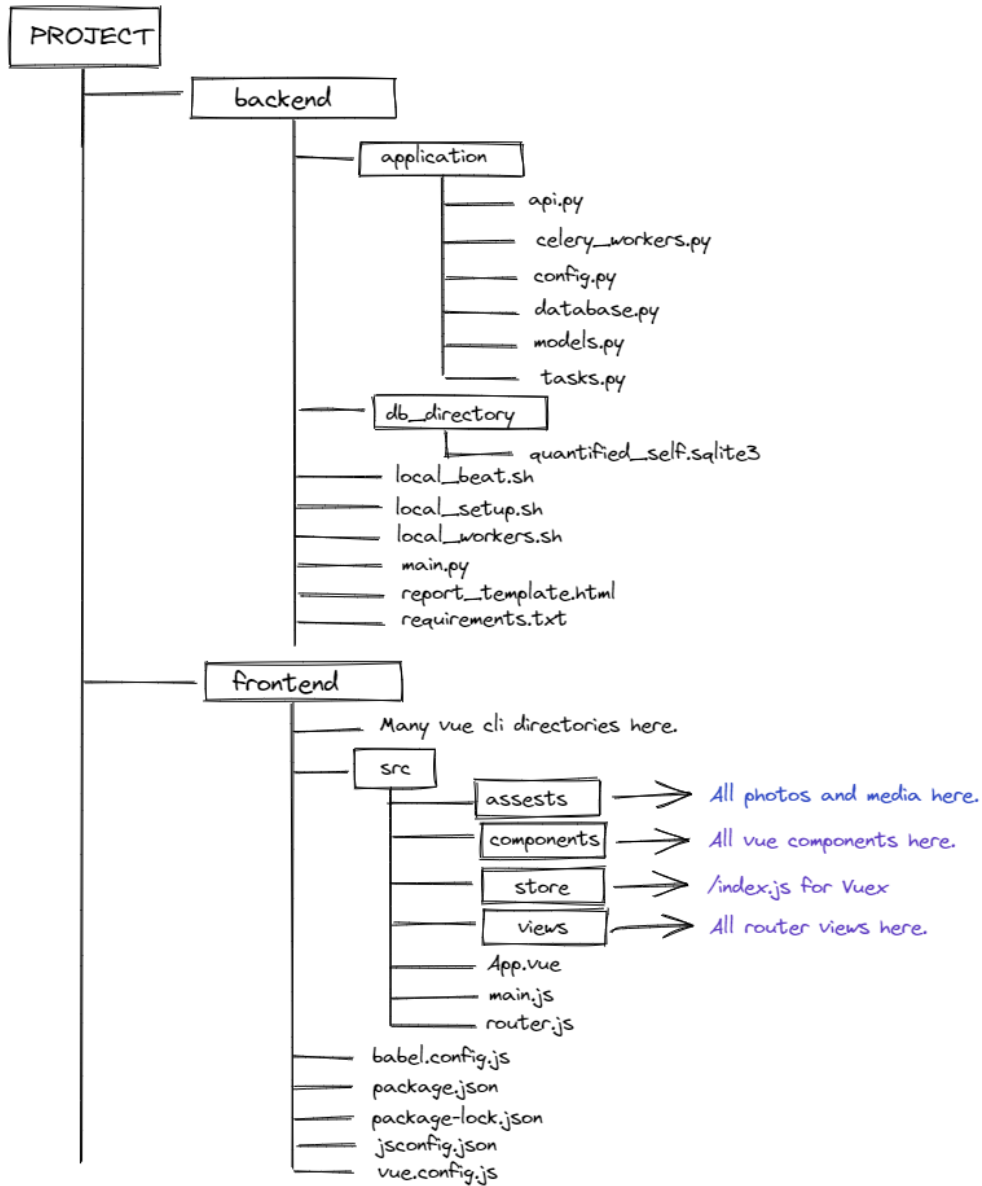
## Architecture and Features

The LOGIT Project has been structured very well and follows standard architecture. Inside the root directory, there exist two directories, backend and frontend separately. Inside backend resides two directories and seven files. The main app is kept inside the backend as 'main.py'.

The main features of this project are:

- A login system with maintenance of the user database which provides flexibility of adding new users. Database is maintained and user inputs are looked up in the database.
- A dashboard with minimalist but informative design with data visualizations.
- Actions like add new tracker, add new logs to the existing tracker or deleting any tracker from the database. Appropriate API is made in accordance with each action.
- An interactive design for logging and changing tracker setting is provided. Vuejs is used to make the page aesthetic and interactive.
- Celery workers and celery beat along with crontab is used to schedule jobs to:
  - Generate monthly report of user and send them via mail.
  - Check every evening if the user has logged any value in any of its trackers in last 24 hours; if not, alert them via mail to log.
- Validations while adding entries. Frontend and backend validations are performed before storing the data.

The overall project structure is as follows:



Video

[https://drive.google.com/file/d/1qhrhnwqFMqH7bo-\\_jX1Ej8dr4deDAJ-F/view?usp=sharing](https://drive.google.com/file/d/1qhrhnwqFMqH7bo-_jX1Ej8dr4deDAJ-F/view?usp=sharing)