

## Author

**Name:** Arvind Raghavendran

**Roll Number:** 21f1005301

**Email ID:** 21f1005301@student.onlinedegree.iitm.ac.in

### About Me:

I am a Mathematics student from Indian Statistical Institute, Bangalore. I have had basic experience in programming in Python before, but this is my first experience in creating a full-fledged app on my own. Despite the obvious lack of experience in programming, I have tried my best to compile and present a working application.

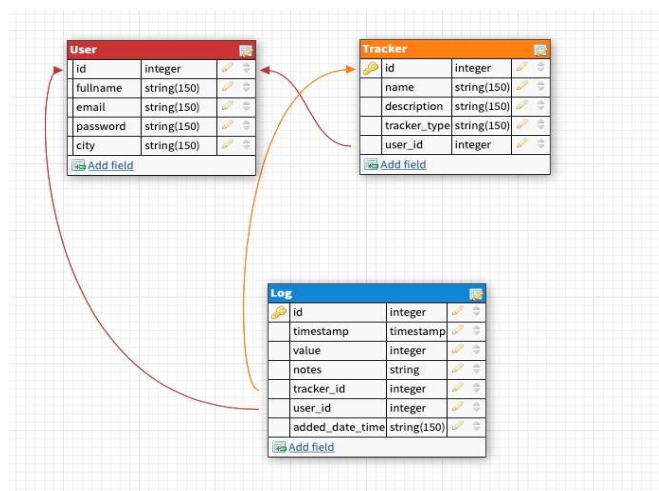
## Description

The aim of the project was to build a Quantified Self App, that would allow one to track and monitor their daily activities, performance, emotions, etc. We were assigned the task of building both the frontend and the backend of the app; the former using HTML styled with CSS, and the latter using Python and its various functionalities. The user should be able to view any/all of his trackers along with all the logs for each tracker. He should also be able to add/remove any of his trackers, and add/remove any of his logs. He also should be able to see a trendline of his tracker logs viz appropriate graphs.

## Technologies used

A variety of Python extensions and modules were used to build the backend of the app. Flask, Flask-Login, SQLAlchemy, Werkzeug, Matplotlib and Jinja2 are a few to name. A complete list of the packages and their versions can be found in the requirements.txt file in the ZIP folder. Each of these packages had their own reasons for being included, for example Flask-Login was used to set up user login, Werkzeug.security was used to provide password authentication to the login process, all the graphs were generated using Matplotlib and NumPy and the model was set up using a database from Sqlite3, which was manipulated using SQLAlchemy. The entire app was built locally on PyCharm, and DB Browser for SQLite3 was used to build the database.

## DB Schema Design



There are 3 main tables in the database: User, Log and Tracker.

User keeps track of all the users who have logged in to the app, with details including id, fullname, email, password and city. Email, fullname, and city are VARCHAR, with the UNIQUE constraint enforced on Email, since it is the username for the user while logging in. The password column is also VARCHAR, and it stores a SHA-256 encoding of the password set by the user. ID is a non-nullable, autoincremented column which is the primary key of this table.

Tracker keeps track of all the trackers set by different users. It has columns id (INTEGER NOT NULL), name, description, tracker type (all VARCHAR). There is a foreign key link to the User, where it references the user\_id. This helps us track which user set which tracker. The primary key of this table is id.

Log keeps track of all the logs done by the user to each tracker. It has columns id (INTEGER NOT NULL), value (INTEGER) and timestamp, notes, added\_date\_time (all VARCHAR). There are 2 foreign key references, one to tracker.id and one to user.id, so we can keep track of which user made the log with respect to which tracker.

All the database schema are stored in models.py file. It was a very intuitive, simple design for the database. The foreign key constraints had to be imposed to ensure that multiple trackers were still identified by their users, and similarly for the logs.

## Architecture and Features

The file hierarchy of the project is very similar to what was explained by Thejesh Sir. There is a config.py file for the configuration settings, an \_\_init\_\_.py file for the initialization of the app and the database, and also the Login Manager, and an auth.py that handles the password authentication of the login page. There is a view.py file that takes care of all the views such as the homepage, sign-up page, login page, adding trackers, viewing logs, etc. A separate file called database.py was made to initialize the database via SQLAlchemy, and as mentioned earlier, models.py has all the data related to the database schema. These 6 files along with the database itself are stored in a folder called Application. The HTML templates are stored in a folder called templates. The graph and the CSS sheet used are stored in a folder called static. Both these folders are subfolders of the Application folder. Along with the Application folder, there is a README.md and a requirements.txt file in the root directory of the ZIP file.

The main features of the application as mentioned before are adding/deleting trackers/logs, and viewing trendlines for each tracker. The app supports 3 main types of trackers: Numerical, MCQ and Boolean. Numerical trackers could involve anything from water consumption to calorie expenditure; MCQ is restricted to the user tracking his/her emotions, and the Boolean is a yes/no tracker, for instance "Was I productive today". There are additional features like editing the user profile and editing the logs and trackers.

## Video

<https://drive.google.com/file/d/1QOawXV3g6wKGv9S5ee2NdhFk9srdWIdI/view?usp=sharing>