

Final Project Report

BY PUSHPAK RUHIL

21f2001180 |Modern Application Development - II | May term, 2022.

# AUTHOR

Name - PUSHPAK RUHIL

Roll No - 21F2001180

Email - 21f2001180@students.onlinedegree.iitm.ac.in

About me – I really enjoy programming from my childhood. Complex problem solving makes me happy.

# DESCRIPTION

Under this project, we are expected to make a version 2.0 of our MAD-I’s WebApp. We have to focus more on the front-end as well async jobs.

# TECHNOLOGIES USED

Here’s are the technologies I used in this project.

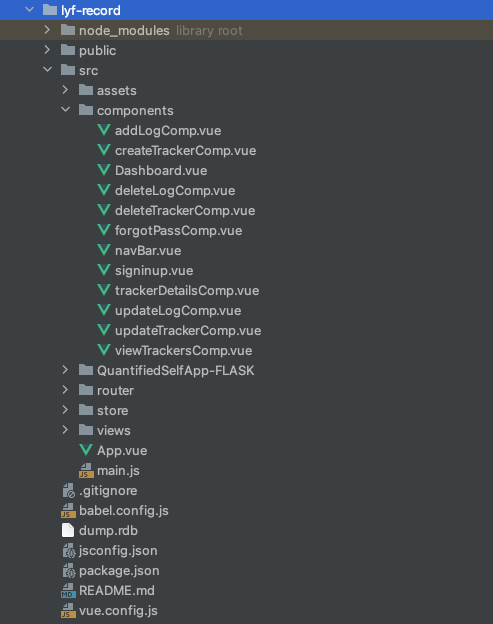
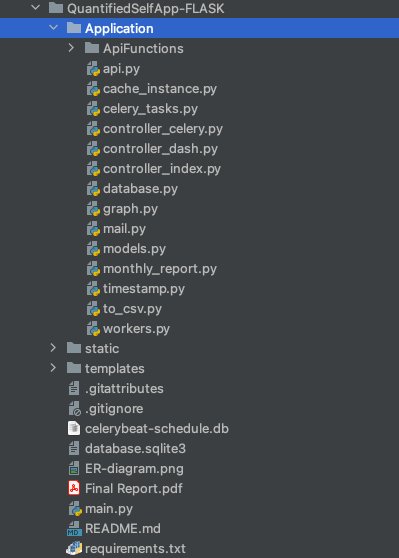
**Existing in previous project:**

Python, HTML/CSS/JS, Jinja, bootstrap, Flask, SQLite, Flask-SQLAlchemy, Matplotlib’s pyplot.

**Additional in this project:**

Celery, Redis, VueJS/Vue CLI, mailhog, pdfkit, flask-restful, flask-caching, flask-JWT-Extended

# ARCHITECTURE

/Component/ folder – all the VueJS components

/QuantifiedSelfApp-FLASK/ folder – Flask backend folder.

/QuantifiedSelfApp-FLASK/Application/ folder – Backend server package. Controllers, API, Async worker, tasks, etc

# FEATURES

Here’s a list of features –

* User login, Dashboard, trendlines
* CRUD operations on tracker - Create, Read, Update, Delete
* CRUD operations on log
* Graphs for each tracker – summarizes all the logs graphically
* APIs
* Backend async jobs – Daily reminders, monthly reports, download trackers/logs
* Server-Side Caching using Flask-Caching

These are the **additional features** that I have added –

* **Streak** – A number representing the current longest login record.
* **Member since** – Represents the number of days since you’ve registered with the application.
* **Sign out option** – signs the user out
* **Forgot Password option** – Lets the user receive his password on mail if he forgets it.
* **Integrated sign in/sign up page** – Interactive page with animation where users can sign in and sign up.
* **Server Sent Events(SSE)**– Every time a user registers, a message is sent to the google chat space.
* **PDF Monthly Reports**
* **Single Page Application**

# VIDEO LINK

Video demonstration of my project is available here.

<https://drive.google.com/file/d/1I8IV73oDG9YzbxBxfPsmNUMnqRkeCMKY/view?usp=sharing>

# DB SCHEMA DESIGN

I have created a total of 9 tables to make my work easy and more organized. I have tried to Normalize the schema.

ER diagram for the schema has been attached by me on the last page, kindly refer to that for more details.

**<<DB>>**: Discussed Before

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Columns** | **details** | **Constraints** |
| User | Username  Password  Email  Creation  Last logged | Username of the user  Password of the user’s account  Email ID of the user, can be used to recover account.  Date of creation  Last logged value to any tracker | Text, Primary key  Text, Not null  Text  Text  Text |
| Tracker | Tracker\_ID  Name  Description  Type  Last\_log | Tracker’s ID number  Name of the tracker  Description of the tracker  Type of the tracker (Numerical, multi choice, time duration, Boolean)  When was the value last logged for the tracker | Integer, Primary key, Autoincrement  Text, Not null  Text  Text, Not null.  Text, Default = ”Not yet logged!” |
| Tracker\_bool  Tracker\_num  Tracker\_mc | Log\_ID  Tracker\_ID  Timestamp  Value  Note | Log event’s unique ID  <<DB>>  Time at which the value was logged  Value input for the log  Note for the particular log | Integer, Primary key, Autoincrement  <<DB>>  Text, Not null  Integer(num)/text(others), Not null  Text |
| Tracker\_TD | Time\_start, time\_end | Start and ending time for the event being logged. In this table, we don’t have a value attribute, but these 2 separate attributes. | Text, Not Null (Time\_start only) |
| Multi\_choices | Tracker\_ID choices | <<DB>>  Stores the choices for trackers with type multi choice! | <<DB>>  Text |
| Streak | Streak\_id  Username  Date  Count | <<DB>>  <<DB>>  Date of last login  Streak count | <<DB>>  <<DB>>  Text, not null  Integer, default=1 |
| User\_Tracker | ID  Username  Tracker\_ID | Just the Primary key  <<DB>>  <<DB>> | Integer, Autoincrement  <<DB>>  <<DB>> |