



# Final Project Report

BY PUSHPAK RUHIL

21f2001180 | Modern Application Development - I | Jan, 2022 batch

## AUTHOR

Name - PUSHPAK RUHIL

Roll No - 21F2001180

Email - [21f2001180@students.onlinedegree.iitm.ac.in](mailto:21f2001180@students.onlinedegree.iitm.ac.in)

About me – I love tech! I started my journey with tech when I was in 4<sup>th</sup> grade. Started coding in 6<sup>th</sup> grade. I developed my interest in Data Science when I was in my 12<sup>th</sup> grade.

## DESCRIPTION

In this project, we need to build a WebApp which can be used as an event logger. Users can login to their account, create different sorts of tracker, and log values for that particular event. They should have the CRUD functionality for their trackers and events!

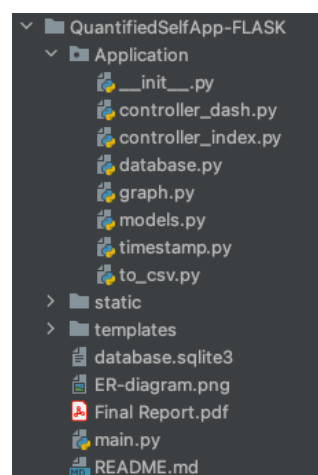
## TECHNOLOGIES USED

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

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

- Python is the core programming language used; Flask is the main framework used for the WebApp.
- SQLite is used for creating a database file for data storage and access.
- Flask-SQLAlchemy is the SQL toolkit used to connect with the database file.

## ARCHITECTURE



Main.py file contains the main code; Application folder is the module that I have created. File names are pretty obvious for their roles.

“static” folder contains all the CSS files, images and GIFs; “templates” folder contains all the HTML files.

“database.sqlite3” contains the database.

## FEATURES

Here's a list of features –

- CRUD operations on tracker - Create, Read, Update, Delete
- CRUD operations on log
- Graphs for each tracker – summarizes all the logs graphically

These are the additional features that I have added –

- Streak – A number representing the current longest login record. Your streak goes +1 if you login daily. If you skip a day, it resets back to 1. Streak updates at 12AM every day.
- Member since – Represents the number of days since you've registered with the application.
- Sign out option – signs the user out
- Integrated sign in/sign up page – Interactive page with animation where users can sign in and sign up. Also checks whether a username already exists or not while signing up. Throws an error if username already exists, incorrect username, incorrect password, etc.
- Export logs to a CSV file – User can export and save their logs for a particular tracker in the CSV format.

Everything has been implemented using basic python libraries and storing the record in the database.

## VIDEO LINK

Video demonstration of my project is available here.

<https://drive.google.com/file/d/1mekyglleZVAztBbuEiW3Qr4KMzPYogSg/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<br>Password<br>Email<br>Creation             | Username of the user<br>Password of the user's account<br>Email ID of the user, can be used to recover account.<br>Date of creation   | Text, Primary key<br>Text, Not null<br>Text<br>Text   |
| Tracker                                   | Tracker_ID<br>Name<br>Description<br>Type<br>Last_log | Tracker's ID number<br>Name of the tracker<br>Description of the tracker<br>Type of the tracker (Numerical, multi choice, time duration, Boolean)<br>When was the value last logged for the tracker | Integer, Primary key, Autoincrement<br>Text, Not null<br>Text<br>Text, Not null.<br>Text, Default = "Not yet logged!" |
| Tracker_bool<br>Tracker_num<br>Tracker_mc | Log_ID<br>Tracker_ID<br>Timestamp<br>Value<br>Note    | Log event's unique ID<br><<DB>><br>Time at which the value was logged<br>Value input for the log<br>Note for the particular log   | Integer, Primary key, Autoincrement<br><<DB>><br>Text, Not null<br>Integer(num)/text(others), Not null<br>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<br>choices                                 | <<DB>><br>Stores the choices for trackers with type multi choice!   | <<DB>><br>Text  |
| Streak                                    | Streak_id<br>Username<br>Date<br>Count                | <<DB>><br><<DB>><br>Date of last login<br>Streak count  | <<DB>><br><<DB>><br>Text, not null<br>Integer, default=1  |
| User_Tracker                              | ID<br>Username<br>Tracker_ID                          | Just the Primary key<br><<DB>><br><<DB>>  | Integer, Autoincrement<br><<DB>><br><<DB>>  |

## ER DIAGRAM

This is the ER diagram for my Database schema that I generated using a web tool.

