Quantified Self App – Project Report – 23rd July 2022

Author

Shriram G
21f2000668
21f2000668@student.onlinedegree.iitm.ac.in
Currently a Student in 2nd Term Diploma in IITM B.Sc. Degree.

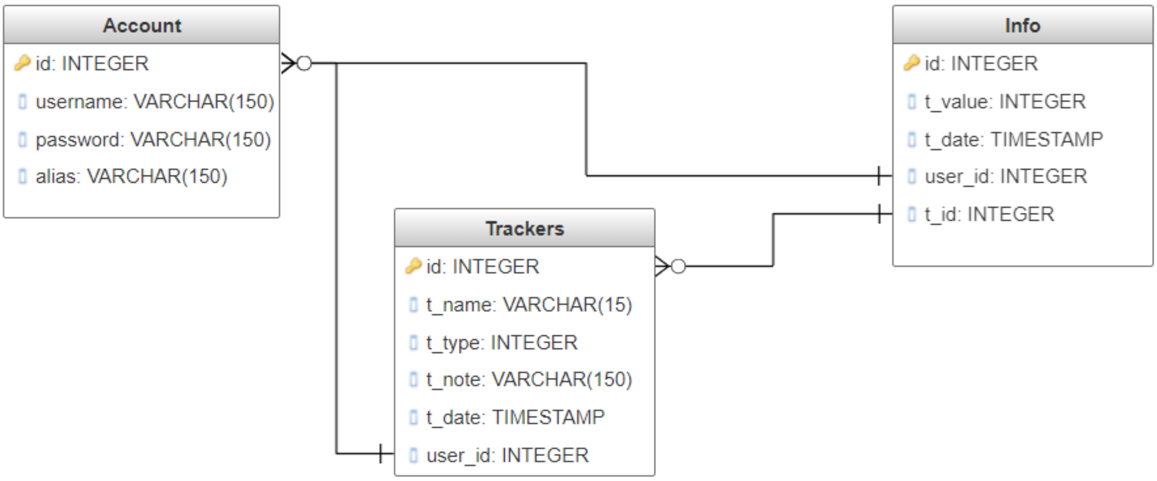
Description

The app functions as a tracking app to maintain logs of periodic occurrences of habits, events, health factors etc. in the user's life.

Technologies used

1. **Flask – The python micro-framework**
2. **Flask-Login – For User Authentication**
3. **SQLiteDB – For Database Management**
4. **Flask-SQLAlchemy – Object Relational Mapper**
5. **Jinja – Markup Templating Engine**
6. **Bootstrap – For Ease of Styling (Through CDN)**

DB Schema Design



A Separate User Table Is Used To Keep Record Of User Records.

The Trackers and Info of the Trackers are kept in separate tables such that there is a separate record of trackers with foreign key of user's id to identify the records and there is a separate table for tracker information where it is given a unique tracker id and user id to easily query records with ease

API Design

This app purely implements the app using Flask's inbuilt Blueprints functionality and does not use Flask-Restful for a REST Architecture.

Architecture and Features

The directory tree structure of the project is given below

```
--app
    --static
    --templates
    --__init__.py
    --views.py
    --sign.py
    --setup.py
    --models.py
    --{{DBname}}.db
--main.py
```

The root folder contains the app folder with the main file which runs the app. Root folder also holds the requirements file, project report etc.

Much of the code base goes into the app folder to establish a package structure to avoid circular importing issues.

The static and templates folder contain the static files and templates according to standard Flask Coding Conventions.

Setup file contains configuration details and __init__ file contains the app constructor.

Sign and views python files contain the authorization view blueprints and other blueprints for other views respectively.

Models.py is dedicated to the DB model made using SQLAlchemy ORM

Extra Structural Features :

- Uses Blueprints for flexibly extending the coding for usage in multiple apps.
- Uses specialized class to serialize the app objects for handling multiple apps at a time

Trackers :

The app provides Numeric and Boolean trackers primarily.

The Boolean trackers may also be used for time stamp tracking as well and functions with dual purpose which is apparent in the views of the app.

The app supports all CRUD operations for trackers.
Users may Create new, update existing, and delete old trackers and all existing trackers are available on the dashboard.

Video

Project Video Presentation :
https://drive.google.com/file/d/145HQ_zhbMnIhPAIBTKKPyWmjBRh22cUK/view?usp=sharing