Author

Bhavya Dua
21F1003641
21f1003641@student.onlinedegree.iitm.ac.in
I am a third year B.Tech (CSE) student, with a passion for robotics and public speaking.

Description

A QuantifiedSelf web application which is similar to a journal – entries for different trackers with their logs, which allows the user/client to track his day-to-day activities and allows him/her to visibly track their progress over time

Technologies used

Flask, Flask-SQLAlchemy, SQLite, matplotlib, numpy, pandas, os, datetime, HTML, CSS, Bootstrap, Vue.JS, Celery, Redis, smtplib, flask-jwt-extended

Flask and Flask-SQLAlchemy: Basic routing and database related operations and to provide CORS access to the Vue.js application

Flask-JWT-Extended: Create and manage web tokens

Numpy, matplotlib: Visualization of tracker trends (graphs)

OS library: secret key

HTML, CSS, Bootstrap: Designing webpages

Datetime library: Retrieve date and time, for timestamps and setting timer expiry

Vue.JS: Create frontend and interface to connect with Flask API

Celery, Redis, smtplib: Asynchronous batch jobs

Pandas: Export data as CSV files

DB Schema Design

1. User

   This table stores the data of users who have created an account in the application.
   The attributes of this table are:

   1) User_id: This is the primary key of this table and has an auto increment feature .

2) Username: This stores the name of the user who registered on our website and has not NULL feature enabled

3) Mail: This stores the email-ID of the user who registered on our website and has not NULL feature enabled

4) Password: This stores the password of the user's account who registers on our website, which the user has to input later to login to the website and has not NULL feature enabled

2) Trackers

This table stores the information about the trackers the user creates.

The table includes the following attributes:

1) Tracker_id: This stores the id of the tracker the user created. This is the primary key for the trackers table and has auto increment enabled.

2) u_id: This stores the id of the user who created that particular tracker, This is a

foreign key reference to the username attribute of the user table.

3) tracker_name: This stores the name of the tracker the user created. It has not NULL enabled.

4) tracker_type: This stores the type of the tracker the user created. It has not NULL enabled.

5) tracker_settings: This stores the values/choices of the tracker a user creates if the tracker is not of type numerical.

6) date_created: This stores the date of creation of the trackers

3) Logs Table

This table stores the values the user logs into a particular tracker.

The table includes the following attributes:

1) Log_id: This stores the id of the log that the user logs into the tracker. This is the primary key for the log table table and has auto increment enabled.

2) T_id: This stores the id of the tracker the user created. This is a foreign key for the log table table that references the tracker_id of trackers table.

3) User_id: This stores the name of the user. This is a foreign key which references the username of the user table.

4) Note: This stores the note the user inputs while logging into the table.

5) Value: This stores the value the user wants to input. It has not NULL enabled.

6) Timestamp: This stores the timestamp in which the logging has been done

API Design

APIs have been created and utilized for signing up and logging in to the user's account. They are also used for performing CRUD operations on trackers, as well as trackerlogs.

Architecture and Features

Project folder

|_ backend

|_ static

|_ templates

|_ flask

|_application.py

|_project.db

|_requirement.txt

|_ frontend

|_ node_modules
|_ public

```
    |_ index.html
|_ src
    |_ App.vue
    |_ assets
    |_ components
    |_  views
    |_  router
    |_  store
    |_ utils
    |_ pics
    |_ main.js
```

The templates are in the “templates” folder of the directory. The models are declared in the the Python file, followed by controllers with route functions as decorators in a logical order. A secret key is implemented to create a session which lasts for 1 hour. The user is asked to login using their credentials. If the user does not exist, they are asked to register. If the password is wrong, the same is displayed to the user. Successful logins redirect to the user’s dashboard, where they can see their existing trackers, with the recorded value and timestamp, along with the ability to go to logging view for a particular tracker, update the tracker, delete it, or create a new tracker. Creating a new tracker requires the user to give the tracker’s name, description, and tracker-type, and optional settings. Upon submitting the same via a form, the logging view for a specific tracker type is given. The user, upon filling out the details and submitting them, gets flashed a message regarding the same. They are then given an option to view their list of trackers separately or go back to the dashboard.

Features implemented:

1) Register your account.

2) Login page: This searches the database for your credentials and helps you sign in to the website and provides you with a JavaScript web token which you require for accessing the application’s core features.

3) Home page: A home page that welcomes you to the website.

4) Trackers page: This page searches the database based on the username and filters all the trackers that the user has created and represents it in the form of a table.

5) Trackers Creation page: This page uses Bootstrap forms and helps the user create trackers for themselves.

7) Trackers Update page: This page allows the user to edit the trackers he has created and updates it in the database.

8) Trackers Delete page: This page allows the user to delete an entire tracker. Deleting a tracker deletes all the records it has on the log table.

9) Logging page: Users can log values to a particular tracker in this page. The user is asked to input the value and note that they want to add while adding the value to the tracker.

10) Tracker info page: This page provides information about the tracker to the user. This page contains the logged information as well as graphs to indicate trends in the tracker.

11) Log update page: This page allows the user to update the logged values in case the previously logged values were wrong or if the user wishes to change the values.

12) Log delete page: This page allows the user to delete a particular log in the log table. Finally, whatever changes we do to the log table is dynamically reflected in the trackers info page.

Additional features:

1) Styling and aesthetics: CSS and Bootstrap have been used for styling and aesthetics.

2) A “Forgot Password” page: A page where the password of an account can be reset by using an OTP sent to the mail of the user.

Video

https://drive.google.com/file/d/1KHxC8zZ2jqnMIdtz5SqfhpIga9MlLN-g/view?usp=sharing