

Project Report:

Author

Name: Maheedhar A

Roll number: 21f1006544

student email: 21f1006544@student.onlinedegree.iitm.ac.in

I am a student of SSN College of Engineering, Chennai. I am very enthusiastic about pursuing this course too. My hobbies are listening to music, reading.

Description

From the problem statement we received from IIT , I understood that we need to make a website which allows the user/client to track his day-to-day activities and allows him/her to visibly track her progress over time.

Technologies used

In my code, I have used the following technologies:

- 1) flask, flask-sqlalchemy ,
- 2) sqlite ,
- 3) numpy ,
- 4) matplotlib ,
- 5) os library from python
- 6) datetime library from python,
- 7) HTML,
- 8) CSS and
- 9) Bootstrap

Their applications:

- 1) Flask and flask-sqlalchemy for basic routing and database related operations.
- 2) numpy , matplotlib for plotting the graphs which shows the progress of the user.
- 3) Os library to set the secret key for our application.
- 4) HTML,CSS,Bootstrap to design beautiful and interactive web pages.
- 5) Datetime library to get date and time in format of datetime with respect to sql-alchemy.

DB Schema Design

Database Structure:

There are totally three tables in my database:

i) user

This table stores the data of users who have created an account in my 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

ii) 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_name= this stores the name 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.

iii) log 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_name : 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

A single user can create as many trackers as he wants. So , there exists a one-to-many relationship between the users and trackers table. Similarly, there exists a one-to-many relationship between the log table and trackers table. That's why I created foreign keys in both trackers and log tables. The reason for including the username in the log table is for filtering out the correct tracker to delete or update.

API Design

I have not designed any API for this website. I have used only flask, flask-sqlalchemy to obtain the core features of the application.

Architecture and Features

The project is present in a separate folder in which I have created a virtual environment to run all the technologies I have used in this project. Inside this folder, the application is present with the name app.py. All the css and plots and images are present in the 'static' folder. All the HTML templates are present in the 'templates' folder.

I have created all the templates with the name of the page it's supposed to represent. Inside app.py I have partitioned the code into many parts containing the different parts of the webpage, each part defined using comments.

Features implemented:

- 1) Register your account in the webpage to track your progress.
- 2) Login page: This searches the database for your credentials and helps you sign in to the website.
- 3) Home page: A home page that welcomes you to the website and gives a short description about the website and its use.
- 4) About page: This page gives you a brief description about the QuantifiedSelf application and its ways of working.
- 5) 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.
- 6) Trackers creation page: This page uses HTML forms and helps the user create trackers for his own.
- 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: I have included the styling and aesthetics using CSS and bootstrap to make the page stunning and aesthetically pleasing.
- 2) A working login page: I have worked on making the login page work, without the use of frameworks like flask-login. My login page doesn't allow multiple users with the same name and mail ID to register and you can only login if you correctly enter your credentials. If you fail to enter your credentials correctly, then the webpage prompts a message indicating that the entered credentials might be wrong.

Video

Link to the video: https://youtu.be/aWq_Vlshs8U