Author

NIWESH BARAJ
21f1001463
21f1001463@student.onlinedegree.iitm.ac.in

Description

A web app which can help the user(signup/login) to track their day to day activities by letting them add their choice of trackers where they can make their value types and then logging it. They can also see their progress in the form of graphs and delete and edit them. Will get periodic reports through email and also push message features.

Technologies used

VS Code (for writing, running and debugging codes), Chrome Browser (for seeing the output & debugging),GitHub/ Git (for maintaining code), DB Browser for SQLite (for database). **Backend** - Flask, Flask-JWT-Extended (for token/security), Flask-RESTful (for api), Flask-SQLAlchemy (for database initialization, model class), Matplotlib (for backend graphs), Werkzeug (for security-hashing), celery, redis, pdfkit and various others mentioned in the requirements.txt file. **Frontend** - Vue with vuex for managing states, Bootstrap 5 and bootswatch link (for styling), vue-chartjs for displaying charts on frontend.

DB Schema Design

There are mainly three tables defined for managing users (username & password) and their trackers with tracker logs and one extra table to store expired tokens. They are:

1. user [id (Int, auto increment, PK), name (Str, Not Nullable), username (Str, unique, Not Nullable), email, password (Str, Not Nullable), secret question(Str), secret answer(Str), trackers (one to many relationship to tracker table)]
2. tracker [tracker_id (Int, Auto Increment, PK), user_id (Int, ForeignKey to user.id, Not Nullable) tracker_name (Str, Not Nullable), tracker_description (Str), tracker_type (Str, Not Nullable), tracker_settings (Str), user (many to one relationship to user table), tracker_log (one to many relationship to tracker_log table)]
3. tracker_log [log_id (Int, Auto Increment, PK), log_tracker_id (Int, ForeignKey to tracker.tracker_id, Not Nullable), when (Str, Not Nullable), val (Str, Not Nullable), notes (Str), trackers (many to one relationship to tracker table)]
4. revoked_tokens [id (Int, Auto Increment, PK), jti (Str {storing expired tokens created during login})]

API Design

There are various api endpoints created for different functionality. They are :
UserRegistration, Login, CRUD API for Trackers and their Logs, Export CSV, etc.
The api endpoints were implemented by using Flask-RESTful. A YAML file containing various api endpoints is attached.
Project directory structure :

/ProjectFolder
- **/backend**
  - **/application**
    - **/api** – containing api.py file
    - **/data** – containing database.py, model.py and data_access.py
    - **/jobs** – containing task.py and workers.py
    - **/utils** – containing validation.py and calculation.py
    - **/config.py**
  - **/db_directory** – containing quantified_database.sqlite3
  - **/static and /templates** – containing various template html files for backend works
  - **/local_beat.sh** – starts celery beat and **/local_workers.sh** – starts celery
  - **/local_run.sh** and **/local_setup.sh** – setups and runs flask application for venv
  - **/main.py** – main python file for running the backend application
  - **/README.md** and **/requirements.txt**
- **/frontend** – below files are generated by vue cli
  - **/src** – main folder for frontend. All vue stuffs is inside it
  - **/assets** – contains logo.png file.
  - **/components** – All vue components inside it like login page etc.
  - **/router** – contains single file index.js for handling vue routes
  - **/store** – contains index.js and modules directory
  - **/App.vue** and **/main.js** – initializes the vue app
  - **/Package.json** and **/README.md**

Every user can see their trackers list with their logs and graphs on their dashboard by logging after signing up. CRUD features for trackers and their logs. They can log out and can forget password features in case it happens. One's dashboard can't be viewed by others.

Video

https://drive.google.com/file/d/16u8sctgsmuslmx1aKJ1M0g5ZWasntdXQ/view?usp=sharing