Author

Raj Rohit Yadav
21f1005377
21f1005377@student.onlinedegree.iitm.ac.in
from Varanasi Uttar Pradesh
Pursuing BS in Data Science and Application from IIT Madras

Description

To create a flask web application having trackers for user to track daily activities and can be created or deleted as per needs of user.

Technologies used

1) Flask
   For running web application.
2) Flask-RESTful
   For building rest API
3) Flask-SQLAlchemy
   For connecting and querying database
4) Matplotlib
   For plotting graph or trendline for trackers
5) Werkzeug
   For HTTP Exceptions

DB Schema design

1) Users
   For saving client or details
   - user_id (Integer, Primary key, Unique, not null)
   - username (String, Unique, not null)
   - password (String, not null)
2) ShadowManager
   For saving client's made trackers
   - User_id (Integer, Foreign key, not null)
   - shadow_id (Integer, Unique, Primary key)
   - shadow_name (String)
   - about (String)
   - shadow_type (String)
   - last_seen (String)
3) Logs
   For saving the logs of tracker
   - user_id (Integer, Foreign key, not null)
   - shadow_id (Integer, Foreign key)
   - log_id (Integer, Unique, Primary key)
   - time (DateTime, not null)
   - value (String, not null)
   - note (String, not null)

API Design

1) ShadowMangerAPI
   End point to read, update, delete and post ShadowManager resources.
   - GET

For getting a tracker detail
- PUT
  For updating tracker detail
- DELETE
  For deleting tracker
- POST
  For adding new tracker

2) LogsAPI
   End point to read, update, delete and post logs
   - GET
     For getting log of tracker
   - PUT
     For updating logs of tracker
   - DELETE
     For deleting logs of tracker
   - POST
     For adding new log to that tracker.

Architecture and Features

1) static
2) templates
   - Boolean.html
   - Dashboard.html
   - editBoolean.html
   - editingShadow.html
   - editMultiple Choice.html
   - editNumerical.html
   - Multiple Choice.html
   - Numerical.html
   - Root.html
   - Shadow.html
   - shadowManagement.html
3) app.py
   having models, controllers, API
4) clientdb.sqlite3
5) validation.py

- First page will be for login authentication
- After successfully login, user will be redirected to their dashboard, where they have all created tracker or create a new one.
- They can log a new event for their tracker.
- On clicking Tracker name, they will be redirected to tracker page where they have all logs for that tracker which can be edited or deleted further, and a trendline.

Video

https://drive.google.com/file/d/1W8OzlCGsv5Ao-mpcoCBD4mtZov-pfOoY/view?usp=sharing

Running Application Link (Replit)

https://quantifiedselfwebapplication.raj-rohitrohit.repl.co/