MAD -2 Project: Quantified Self

Name: GetTracking

Author:

Hari Priya N

21f1000722

21f1000722@student.onlinedegree.iitm.ac.in

I am 2nd year MSc IT student. I love coding and programming. My goal is to become a researcher and plan to pursue my PhD in CSE.

Description:

To create an application that will allow multiple users to login, add their own trackers, log data for trackers, get customized summary report, daily reminders and insight on the summary of their logs.

Technologies Used:

Flask – Web support and APIs
SQLAlchemy – SQLite support
VueJS – User Interface
Celery – Batch Jobs
Redis – Caching

DB Schema design:

Tables used:

- **Login:** To hold details of users

  Primary key: Username: Unique name of users signed in
  Other attributes: Usertitle: Email IDs of the users
  Password: Password
  Active: Indicates whether the account is active or not
  Fs_uniquifier: A unique string
- **Tracker:** To add details about the tracker

  Primary key: TrackID : Auto generated number
  Foreign key: UserName (Unique name of user from login table)
  Other attributes: Trackname: Name of the tracker
  Trackdesc: Description of the tracker
  Trackdate: Last logged in date/ Creation date(First time)
  Tracktype: Type of value taken by racker
- **Logger:** To hold details of log of trackers by various users.

  Primary key: LogID : Auto generated number
  Foreign key: TrackerID (ID of tracker from tracker table)
  UserName (Unique name of user from login table)
  Other attributes: When: Date of log

Value: Score recorded by user
Notes: Details about the log

Architectures:

Functions:

- **Landing_page():** Opens the start.html page for user login
- **Newuser():** Opens newuser.html page for user to add a new login when initiated
- **Dashboard():** Opens the dashboard.html for a particular user to list all trackers
- **Addtrack():** Opens the add tracker form for a particular user
- **Edittrack():** Opens edittracker.html with prepopulated values to edit and saves changes
- **Addlog():** Opens Logtracker.html to adds log when initiated
- **Editlog():**Opens editlog.html with prepopulated values to edit and saves changes
- **Summary():** Opens the summary.html for a particular tracker to list all logs

APIs and Resources:

UserLogin - "/api/newuser": Creates a newuser
UserInfo - "/api/<string:username>": Gets information of all the users
TrackerAdd - "/api/tracker/<string:username>/add": Creates a new tracker
TrackerUpdate - "/api/tracker/<int:trackid>/edit" : Edits a given tracker
TrackerDelete - "/api/tracker/<int:trackid>/delete" : Delete a specified tracker
TrackerInfo - "/api/tracker/<string:username>": Gets tracker information for a username
LogAdd, "/api/log/<int:trackid>/add" : Adds a log for a tracker
LogInfo, "/api/log/<int:trackid>" : Gets log information for a tracker
LogDelete, "/api/log/<int:logid>/delete" : Deletes a specified log
LogUpdate, "/api/log/<int:logid>/edit" : Edits a particular log
GenData, "/api/<int:trackid>/summary" : Gets Summary of the logs for a tracker

Asynchronous and Scheduled Tasks

data_to_csv – Converts log/tracker details to csv when requested by the user
send_reminder – Sends daily reminders if the user has not logged on in mail
send_monthly_report – Sends monthly reports to users about the trackers through mail