MODERN APPLICATION DEVELOPMENT-1

SELF QUANTIFIED WEB APPLICATION

Project by: Komal Chamyal
Roll No: 21f1001093
Email: 21f1001093@student.onlinedegree.iitm.ac.in

About me: *I'm Komal! I study in Hansraj College, DU. Pursuing a second degree from IITM because I'm very much interested in coding. I had a great time making this project! It really brushed up all my concepts and taught me so many new things.*

Description

Self Quantified Application: This application will enable users to track any metric or activity as per their needs. Users can make Trackers of activities and log its entry multiple times. The application then organises all their Trackers and Logs. It displays them along with creating graphs/trendlines and stats corresponding to the Tracker.

Technologies Used

The application is primarily built on Flask and Flask SQLAlchemy. Other technologies used are:
Flask WTF **-** For rendering and validation of forms
Flask Bcrypt - For password encryption of users in database
Flask Login - For handling user sessions (logging)
Matplotlib - For displaying graphs/trendlines

DB Schema Design

The database comprises of 3 Models of User, Tracker and Log. Schema for each is:

User Model: It stores the usernames, passwords and email of all the users registered through the app.
ID - Primary Key - Stores ID of each user
Username - Unique - Stores Username of all users
Email - Unique
Password - Stores the encrypted passwords of corresponding users

Tracker Model: Keeps the data of all the created trackers.
SNo -Primary Key - Serial Number of the Tracker
Title - Title of the Tracker
Type - Type of Tracker (Numerical type, Multiple Choice type etc)
Desc - Description of the Tracker
Date - DateTime - Stores the date on which the tracker was created/ last updated
Settings - If the tracker is of Multiple Choice Type, this column stores the options of entry values to choose from
User_ID - Foreign Key - Stores the User who created a particular Tracker

Log Model: Used for logging all entries in a tracker.
LogNo - Primary Key
Tracker_ID - Foreign Key - ID of the tracker whose log is being made.
Value - Log Entry Value (Example- 5 miles for a Running tracker, Happy for Mood Tracker etc)
Timestamp - DateTime - Time of log entry (User can change it)
Note - Any notes corresponding to the log entry (Optional)

Note: All columnsused are of Integer, String or DateTime type.

Architecture and Features

Project Architecture:

- ➢ All templates reside in “templates” folder.
- ➢ All images reside in “static” folder.
- ➢ The controllers and views are implemented in the main “app.py” file.
- ➢ Models are organised in “models.py” file.
- ➢ All form validation is done is “forms.py” file.

Application Features:

Core Features

- ✓ CRUD on Trackers
    - I. Users can create trackers of type: NUMERICAL, MULTIPLE CHOICE, TEXT
    - II. Viewing of User’s Trackers on dashboard
    - III. Update and Delete operations provided for Trackers
- ✓ CRUD on Logs
    - I. Create logs based on the Tracker type
    - II. Viewing of a particular tracker’s logs available
    - III. Update and Delete operations available for Logs
- ✓ Graphs and Trendlines - Graph corresponding to each tracker is provided for better understanding of Trackers.

Additional Features

- ➢ Proper Login Framework with User Authentication- Users can Sign up and Log in with Username (unique), Email and password. User Authentication is done and the corresponding dashboards are then loaded.
- ➢ Validation of Inputs- All input fields are validated in all forms.
- ➢ Password Security - With Bcrypt, all passwords are encrypted and hence security of application is increased.

Presentation Video

https://drive.google.com/file/d/1GQ-PSE6KSD9AeoXdY7D16iJU3n7BrIX5/view?usp=sharing

Replit Project Link:

https://self-quantified-application.komalchamyal.repl.co/