Author

Rohit Singh
21f1003875@student.onlinedegree.iitm.ac.in
21f1003875
I am Rohit, currently a diploma student in IIT Madras online degree program. I am passionate about mathematics and data science machine learning.

Description

Project requires us to design a tracker app. It should support multiple types of tracking systems and provide the user with a reliable log of tracks and features such as trendline .

Technologies used

- Flask
- DB browser
- Matplotlib and datetime library from python

The application is built on flask while the model of the application(database) was built using DB browser.
Matplotlib and other libraries are used to design various features of the application for the convenience of the user.

DB Schema Design

The database supporting our application consists of 3 tables that are:

1. User
2. Trackers
3. Tracker_logs

Our User table consists of two columns

- Username(text,unique,not null)
- Password(text,not null)

This table contains Username password for the login process of application

The tracker consists of seven columns

- Id (Int,Primary key,autoincrement)
- Name(text,unique,not null)
- Description(text,not null)
- Tracker_Type(text,not null)
- Last_tracked(text)
- Settings(text)
- User(text,not null, foreign_key(Username from user table))

This table stores the information about the tracker created by the user. Hence we have added the username to identify trackers of a particular user and created one to many relationships that help us in cascading deletion.
Settings column takes comma separated values to create optional type tracker

Other fields are for storing the details of the tracker
Most of the fields are in text format due to ease of working with text.

The tracker_logs table consists of 5 columns:

- T_id(Primary key,autoincrement)
- Timestamp(Text,not null)
- Tracker_name(Text,not null,foreing_key(Name from Trackers table)
- Values(Text,not null)
- Notes(Text)

Tracker_logs table stores of all the logs generated by trackers. Most of the field are string to allow versatility while handling multiple tracker_type. We have used Tracker_name as foreign key to allow cascading delete and identifying the association of log with a particular tracker

API Design

No api implemented yet

Architecture and Features

Project consists of app.py file and template and static folders. All the html templates are in templates folder while css and plots are in static folder.In addition there is a database file called projectdb which consists of the model for application. Rest of all the things are written inside app.py.

Features that we have implemented in our app are:

- Multiple tracker support
- Trendline
- Logs viewer
- Language support

We have given user CRUD functionality on both tracker and logs. Our multiple tracker system is implemented in the app by using the field tracker type from Trackers table using which we change the output for different tracker type
Trendline is implemented through matplotlib which generates a graph of logs of a particular tracker which is later saved in static folder and used in a template which renders it
Language support is a default feature which is a result of the framework used to develop the app which supports this feature

Video

https://drive.google.com/file/d/1v9ZI3zP4EITC3ShlV3yZCZ376HuRI9-o/view?usp=sharing