Author

Chirag Goel
21f2000540
21f2000540@student.onlinedegree.iitm.ac.in

Presently I am also pursuing B.tech from MAIT,Delhi. I live in Delhi. I am a competitive coder and MERN stack developer.

Description

The aim of the project was to create a tracker. Basically, an app that use to track your daily activities that you want to trace or want to follow for reason of managing it.

Technologies used

1.FLASK-Flask is an API of python that allows us to build up web-application
2.FLASK-SQLALCHAMY is used to handle the SQL file or the database connections with sqlite
3. matplotlib is used to plot graph and show trend lines
4. Rest api using flask with help of flask_restful module.
5.Bootstrap for designing frontend.
6. CSS and Javascript for for styling html and manipulating it.

DB Schema Design

I have used 5 tables:
1.User -> (user_id,user_name,first_name,last_name,password)
2. Tracker -> (tracker_id,tracker_type,tracker_type,description,settings,user_id(Foreign key))
3. Tracker_Numerical -> (tracker_id(Foreign key,log_id,tracker_timestamp,tracker_value,tracker_note)
4. Tracker_boolean -> (tracker_id(Foreign key,log_id,tracker_timestamp,tracker_value,tracker_note)
5. Tracker_time_durtion -> (tracker_id(Foreign key,log_id,tracker_timestamp,tracker_value,tracker_note)
I have used 5 tables to minimize redundancy and atomicity of tables. Tracker table will store data of tracker and different type of tacker table will store the log values according to the tracker type. This will help in easy retrieval of data and it will also reduce anomality's.

API Design

List of Resources created::

Tracker: Has 2 functions:

get: which will return all all trackers

post: will add a new tracker

Tracker_manipuate: Has 2 functions:

put: update an existing tracker

Delete: delete an existing tracker

Tracker_logs: Has 2 functions:

get: will list all logs corresponding to a particular tracker.

Post: will add a new log to the tracker

Log_manipulate: Has 2 functions:

Put: Will update an existing log

Delete: will delete an existing log.

API Implementation:

The data handling for each endpoint is done using flask-sqlachemy.

Get method endpoints are rendered using querying the database.

Post and put methods are rendered using manipulation of database

Architecture and Features

In my project all the html files are in templates and CSS and JavaScript files are in static folder. The image of graph that is created is also getting stored in static folder. All routes and models are there in mainApp.py.

Video

https://drive.google.com/file/d/1UHcrRvToHsaHzGB9raBT1r-HzEOoaTPo/view?usp=sharing