

## Author

Chirag Goel

21f2000540

[21f2000540@student.onlinedegree.iitm.ac.in](mailto:21f2000540@student.onlinedegree.iitm.ac.in)

Presently I am also pursuing Btech in computer science from MAIT, Delhi. I am a competitive coder and MERN stack developer and ml enthusiast.

## 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 styling html and manipulating HTML elements.
7. flask\_bcrypt for hashing the password.

## DB Schema Design

I have used 4 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)

I have used 4 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 anomaly's.

## API Design

List of Resources created::

- Tracker: Has 2 functions:
  1. get: which will return all all trackers
  2. post: will add a new tracker
- Tracker\_manipulate: Has 2 functions:
  1. put: update an existing tracker
  2. Delete: delete an existing tracker
- Tracker\_logs: Has 2 functions:
  1. get: will list all logs corresponding to a particular tracker.
  2. Post: will add a new log to the tracker
- Log\_manipulate: Has 2 functions:
  1. Put: Will update an existing log
  2. Delete: will delete an existing log.

## Architecture and Features

In my project all the html files are in templates folder 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 controllers are in app.py. All database models are in models.py . all api's are in api.py file. Backend validation for password is also there and password hashing is also done.

## Video

<https://drive.google.com/file/d/1XAV3yWXF9U6OFpSbD5s1vtwp5BxvkW5T/view?usp=sharing>