This is a document stating of the final project of MAD 1.

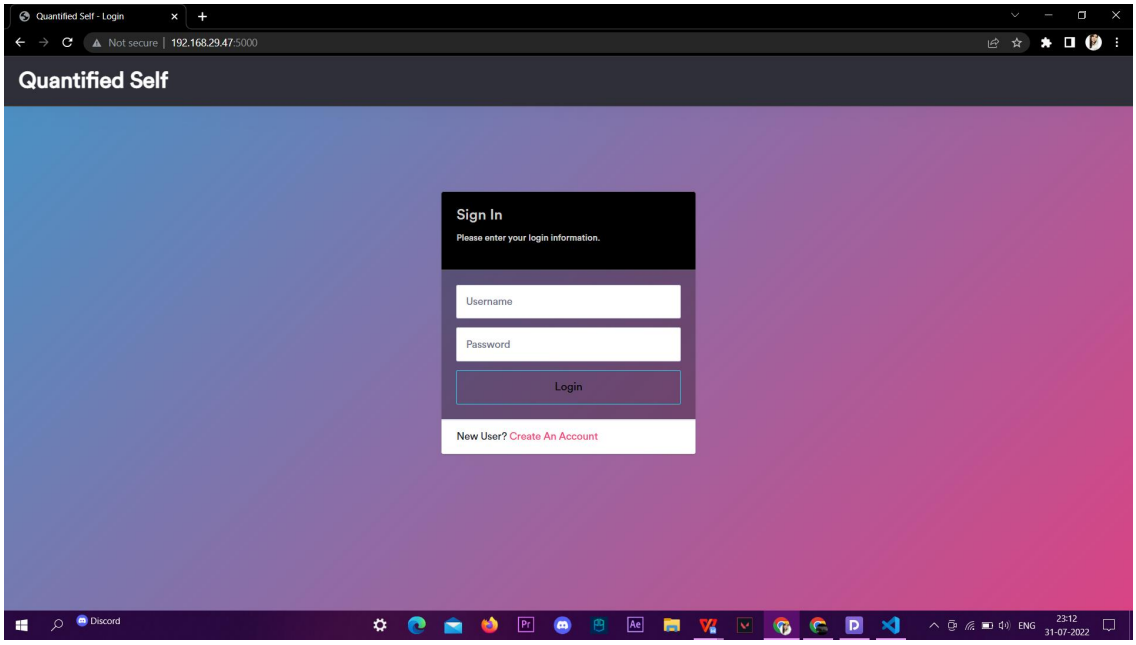
- Flask has been properly implemented and runs successfully.
  - Overall design of the web app is implemented with Bootstrap 5.1.
  - SQLite and its appropriate model are implemented and connected to Flask.
  - Each page has a navigation bar with useful links.
- Creating a user account with username and password is implemented, subsequently storing them in the SQLite database. Password is encrypted with Flask-Bcrypt.
  - Logging in with username and password is implemented with check if the username exists in the database and if the password is correct.
  - Validation of username on account creation and logging in is implemented.
  - Logging out is fully implemented.
- A proper user dashboard called *Home* is implemented.
  - *Home* provides information on what trackers are associated with the profile and value and time of the last logged event of a tracker.
  - *Home* also provides information of last three logged events.
  - *Home* provides quick access to trackers info, ability to log new events, and the ability to go the *trackers dashboard* to create/edit/delete trackers.
- Tracker – CRUD is implemented. Users can create, update, and delete a tracker in the database.
  - Users can add trackers to the database through the *trackers dashboard*.
  - Users can select what trackers to use in their profile and also remove trackers from their profile.
  - Each tracker is divided among 4 tracker types (Numerical, Multiple Choice, Boolean, Time Duration) and user can select one among the four when creating a tracker.
  - For a 'Multiple Choice' type tracker, users are presented with a second page where they can enter Comma Separated Values which will be used when logging with the tracker.
  - Users can edit a tracker's name and description in the database, but the type of a tracker is unchangeable

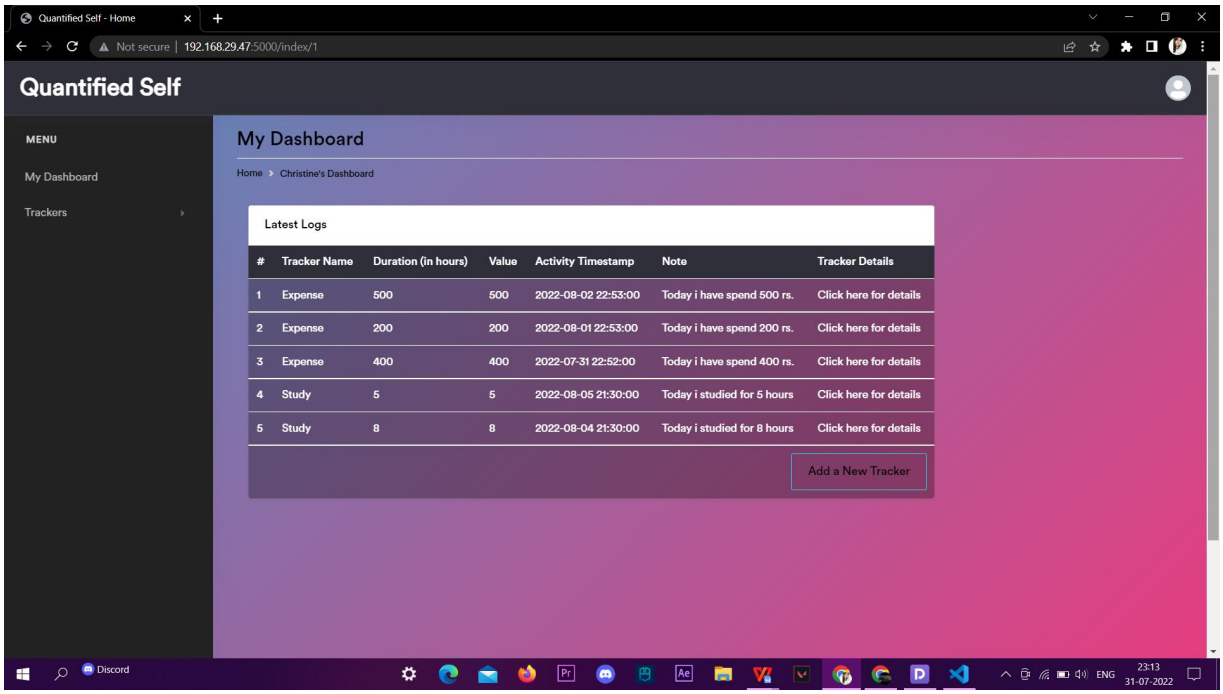
  Users can delete a tracker from the database.
    - Deleting a tracker also deletes every logged event associated with it.
- Logs – CRUD is implemented. Users can add, update and delete logs.
  - Users can log data for a tracker from their *home*.
  - Users are presented with a form when logging and in the form the proper tracker name and tracker type are presented to help while logging a value.
  - Time field in the form is pre-populated with the current time but users can choose to change it.
  - Validation for proper tracker type input is implemented.
  - Users can also edit a logged event through the *Logs* page.
  - When editing a logged event, users can change the value, note and timestamp associated with it.
  - Users can click on a tracker's name in their *home* or *trackers dashboard* to look at the info of a tracker. The info provides the trendlines and previous log events.
    - Users can also log new data through the info page.

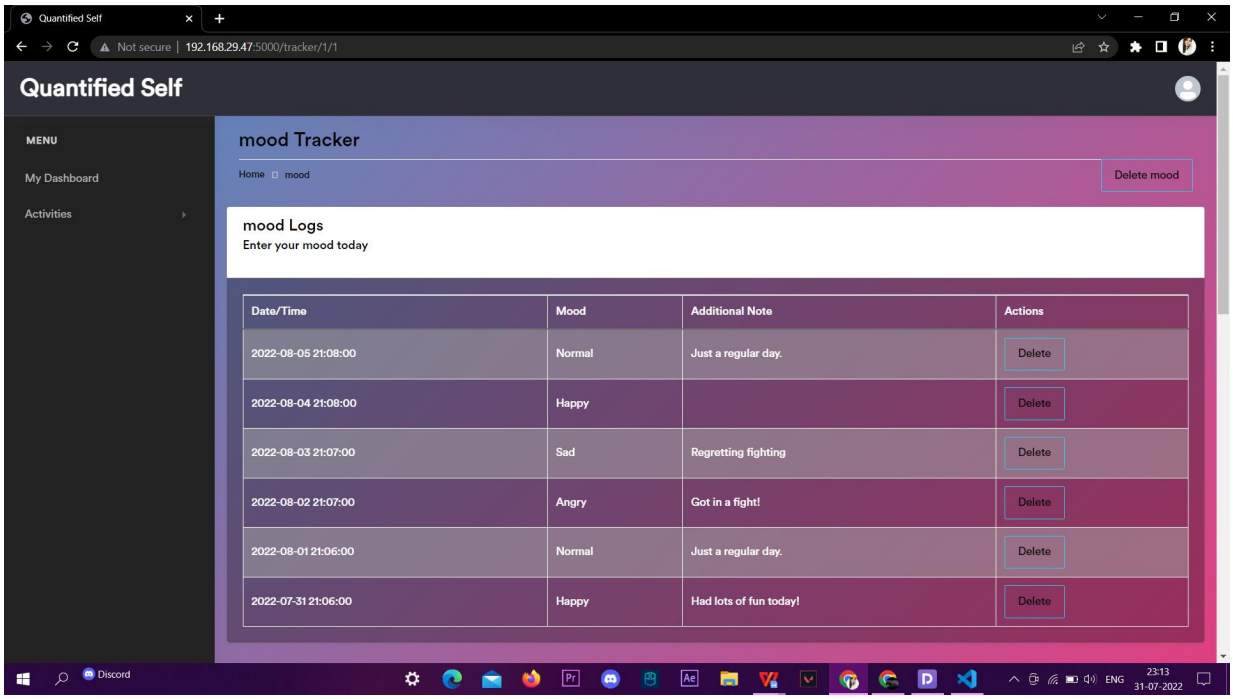
Trendlines are fully implemented for Numerical and Time Duration type trackers with matplotlib.

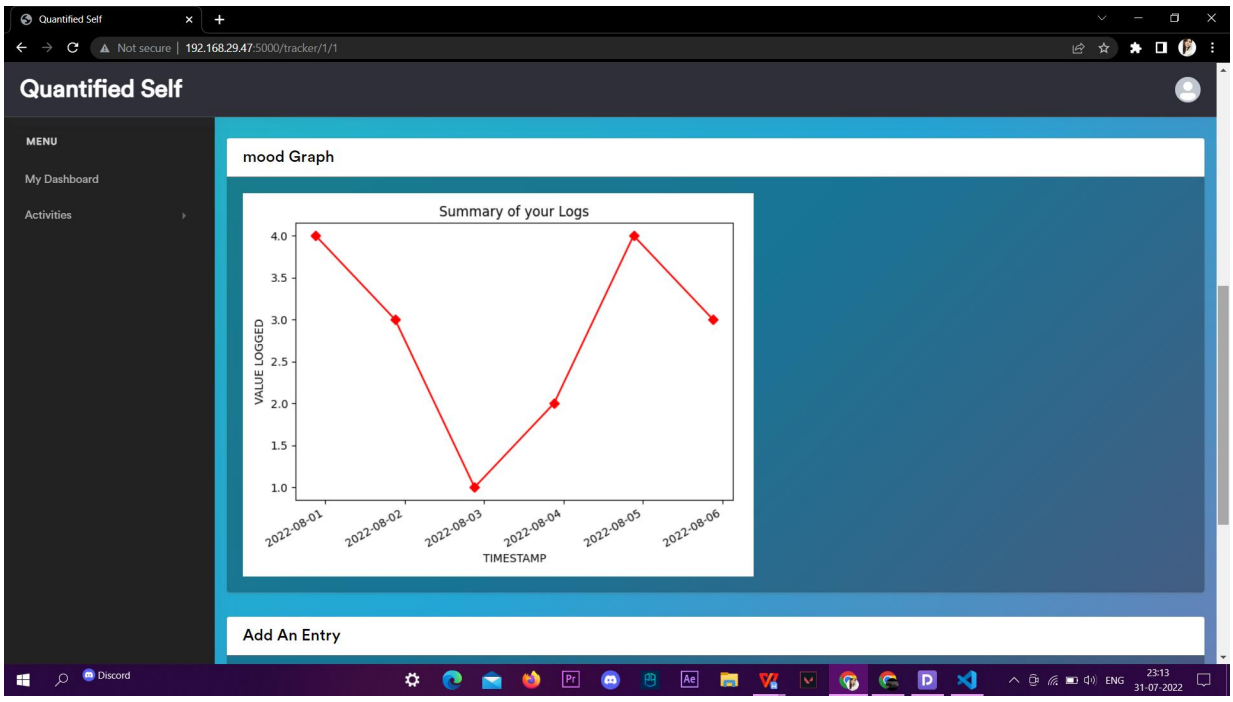
- A Readme is present in the root of the folder which describes thepackages needed to run the app and how to run it.
  - A ‘requirements.txt’ is also provided for easy installation ofpackages.

Here, I have attached few screenshots of login page, user dashboard, tracker page from my website.









Submitted By:

Name: Aayushi Yadav

Roll Number: 21f1006494

Email: 21f1006494@student.onlinedegree.iitm.ac.in