Topic: Databases MongoDB and MySQL
Todo list app using MySQL

Github Repository Link: https://github.com/Trupti0406/todo-mysql
Working video Link: Click Here

Approach:

Instructions in this assignment were pretty straightforward, these are the steps that I followed while creating this application:

1) I created 2 folders inside my root directory, one for “frontend” and another for “backend”.
2) In my frontend folder, I created a React app and made the UI for the application.
3) Then in the backend folder, I created my node app. And installed all the required packages.
4) Then I created my database first, with the database name: **“todolist_database”**, table name: **“todolist_table”** and column names: **“todo_id”** and **“todo_name”**.
5) The instructions mention that we have to add tasks (POST request) and then display them in the form of a list (GET) request. And then add a delete functionality on each todo list item using (DELETE) request.
6) All the requests are simple and easy to code because, in the last assignment, we performed similar operations, just the delete functionality is add on here.
7) While sending a task, traditionally we also send an id. Here I’ve used **AUTO INCREMENT** feature of SQL to increase the ID automatically as soon as we add tasks.
8) Further, we’ve utilized this id while creating a delete functionality in our app.
9) I’ve created a separate component to render the app in React, the component is called Todo.jsx.
10) In my Todo component, I’ve used useState and UseEffect hooks to achieve the desired output.
11) The submitHandler in said component handles the fetching and displaying of tasks.
12) The deleteHandler in the same component handles the delete task action in our app.
13) The deleteHandler is responsible for clearing the storage after refreshing the page for which I’ve called it in a useEffect hook with empty array dependency.
14) This app is fully responsive.

Learning Outcomes:

- Learned the basic syntax of SQL.
- I got familiar with using MySQL as a database in my app.
- Learned to connect the backend to the frontend using REST APIS and MySQL.
- Got comfortable with sending API requests to MySQL database and testing them.