

Topic: Module 3 My SQL (Backend development)

Assignment:-

1. Understand the concepts of Sql.
2. Install xampp and start creating the database using sql commands.
3. Create a database (todo-react) , table name todo-tasks and two columns
 - a. id (int , Auto-increment, primary key , varchar)
 - b. tsks(varchar)
4. Start the project by making two folders Frontend and backend.
 - Backend
1. Install the express my sql and cors by using the command `npm I express cors mysql`.
2. Make the main file (sql.js)
3. IN sql.js
 - a. import the express , mysql and cors.
 - b. Initialize the database.
 - c. Create a connection between sql and node.
 - d. Creating APIs
 - e. Make the post request for storing the data from frontend
 - f. Make get request for get the data of from the database.
 - g. Make the delete request for delete the tasks from database with id.
 - h. Starting the server at localhost 5000. With the message “server is listening at port 5000” in console.
4. Check working of Api's by postman .
 - Frontend
1. Create react app using `npx create-react-app frontend`.
2. In app.js
 - 2.1. Import useeffect and usestate from react
 - 2.2. Install the bootstrap by `npm I bootstrap`.
 - 2.3. Import App.css for Apply styling in the page.
 - 2.4. Create a task component for adding a single task in a paragraph
 - 2.5. Create a main app component

- 2.5.1. Inside this component create a state to store the list of tasks.
- 2.6. Create the another function to handle the form submission when user adding the new task
 - 2.6.1. Use event.preventDefault for prevent the default behavior of the form
 - 2.6.2. Fetch the post api with the help of try catch method , inside try method fetch the api of post request for creating a connection between frontend and backend by which we can add the data in sql database in the form of row and columns , and use catch method for display the error in the console.
 - 2.6.3. Fetch the data from backend when the component did mount with the help of useeffect.
 - 2.6.4. Call the function with useeffect .
- 2.7. Create the another function to handle the deletion by id
- 2.8. Now define the structure of the app component using return statement.
 - 2.8.1. Create a structure using html , css and bootstrap for that we import the css and bootstrap.
 - 2.8.2. Create a div inside the body with className container and inside this create a heading for my react application.
 - 2.8.3. Inside the container create a form for add new task and add a event called onsubmit that triggers when the form is submitted.
 - 2.8.4. Inside the form create an input field for adding the new task with classname , type , name, placeholder, value , and an onchange event, below that create a button to submit the form.
- 2.9. Use task.map for render the task component for each task in the task array with two parameter and a callback function.
- 2.10. The onDelete prop is passed to the task component and contains a function called "handle delete task"
- 2.11. At last export the app component
- 3. Testing the working of app , and apply styling to the page in App.css

Learning and conclusions:

learnt how react works with database for post , get and delete the data from frontend and many more that help me to improve my coding journey.