Experiment No. 5

Aim: Construct front end applications using React and back end using Node js

To create a React app with a Node.js backend similar to the structure you provided, here's a step-by-step guide:

Step-by-Step Guide

---

1. Install Required Software

Make sure you have the following installed on your system:

- **Node.js** (comes with npm)

- **MySQL** (to handle the database)

---

2. Set Up the Backend with Node.js and MySQL

Step 2.1: Create a New Directory for Your Project

```bash

mkdir react-node-app

cd react-node-app

```

Step 2.2: Initialize Node.js and Install Dependencies

Run the following commands to initialize Node.js and install required packages:

```bash
npm init -y
npm install express mysql cors body-parser
```

- **`express`**: Web framework for Node.js.

- **`mysql`**: MySQL database integration.

- **`cors`**: To handle Cross-Origin Resource Sharing.

- **`body-parser`**: To parse request bodies.

Step 2.3: Create Backend Server (Node.js)

In the root of your project directory, create a new file `server.js`. This file will contain the backend code for the Node.js server.

```javascript
const express = require("express");
const mysql = require("mysql");
const cors = require("cors");
const bodyParser = require("body-parser");

const app = express();
app.use(cors());
app.use(bodyParser.json());
```

```
// Create connection to MySQL database
const con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "root",
  database: "students",
});


// Simple GET route to fetch all student details
app.get("/", async function (req, res) {
  const data = await getAll();
  res.writeHead(200, {
    "Content-Type": "application/json",
    "Access-Control-Allow-Origin": "*",
  });
  res.end(JSON.stringify(data));
});


// Route to insert a new student email into the database
app.get("/insert", async function (req, res) {
  const email = req.query.email;
  const result = await insertData(email);
  if (result.affectedRows > 0) {
    const updatedData = await getAll();
    res.writeHead(200, {
      "Content-Type": "application/json",
      "Access-Control-Allow-Origin": "*",
    });
    res.end(JSON.stringify(updatedData));
  } else {
    res.status(500).send("Error inserting data");
  }
});


// Start the server on port 8000
const PORT = 8000;
app.listen(PORT, () => {
  console.log(`Server running on http://localhost:${PORT}`);
});


// Helper function to fetch all student data
const getAll = () => {
  return new Promise((resolve, reject) => {
    con.query("SELECT * FROM std_details", (err, result) => {
      if (err) {
        reject(err);
      } else {
        resolve(result);
      }
    });
  });
};


// Helper function to insert a new email
```

```
const insertData = (email) => {
  return new Promise((resolve, reject) => {
    const sql = "INSERT INTO std_details (email) VALUES ?";
    const values = [[email]];
    con.query(sql, [values], (err, result) => {
      if (err) {
        reject(err);
      } else {
        resolve(result);
      }
    });
  });
};
```

---

3. Set Up MySQL Database

1. **Create a MySQL Database**:

   Run the following commands in MySQL:

   ```sql
   CREATE DATABASE students;
   USE students;
   CREATE TABLE std_details (
     id INT NOT NULL AUTO_INCREMENT,
   ```

```
    email VARCHAR(255),
    PRIMARY KEY (id)
  );
  ```

2. **Start MySQL Server**: Make sure your MySQL server is running, and you have access to it with the credentials (`root:root`) used in the Node.js script.

---

4. Set Up the Frontend with React

Step 4.1: Create React App

In your terminal, navigate to the project directory and create a React app:

```bash
npx create-react-app client
```

Step 4.2: Install Axios for HTTP Requests

```bash
cd client
npm install axios
```

---

5. Frontend Code

Step 5.1: Modify `App.js`

Open `client/src/App.js` and replace its contents with the following code:

```javascript
import React, { useState, useEffect } from "react";
import axios from "axios";
import "bootstrap/dist/css/bootstrap.min.css";
import "./App.css";

function App() {
  const [messages, setMessages] = useState([]);
  const [email, setEmail] = useState("");

  useEffect(() => {
    fetchData();
  }, []);

  const fetchData = () => {
    axios.get("http://localhost:8000/")
      .then((res) => {
        setMessages(res.data);
      })
      .catch((err) => console.error("Error fetching data", err));
  };
```

```
const handleChange = (event) => {
  setEmail(event.target.value);
};

const handleSubmit = (event) => {
  event.preventDefault();
  axios.get(`http://localhost:8000/insert?email=${email}`)
    .then((res) => {
      setMessages(res.data);
    })
    .catch((err) => console.error("Error inserting data", err));
};

return (
  <div className="App">
    <section id="footer">
      <section id="banner">
        <div className="container-fluid" id="banner-container">
          <div className="row" id="banner-row">
            <div className="col-md-4" id="footer-col2"></div>
            <div className="col-md-4" id="footer-col2">
              <h3>Store your email address</h3>
              <form onSubmit={handleSubmit}>
                <div className="mb-3">
                  <input
                    type="email"
                    placeholder="Enter Your Email"
                    className="form-control"
                    onChange={handleChange}
                  />
                  <div className="form-text">
                    We'll never share your email with anyone else.
                  </div>
                </div>
                <button type="submit" className="btn btn-primary">
                  Submit
                </button>
              </form>
              <table className="table">
                <thead>
                  <tr>
                    <th scope="col">#</th>
                    <th scope="col">Email</th>
                  </tr>
                </thead>
                <tbody>
                  {messages.map((message, index) => (
                    <tr key={index}>
                      <th scope="row">{message.id}</th>
                      <td>{message.email}</td>
                    </tr>
                  ))}
                </tbody>
```

```
                </table>
              </div>
            </div>
          </div>
        </section>
      </section>
    </div>
  );
}

export default App;
```

Step 5.2: Add CSS (Optional)

Create or modify the `client/src/App.css` file for some basic styling.

```css
/* App.css */
body {
  font-family: 'Roboto Condensed', sans-serif;
}

.container-fluid {
  padding-top: 50px;
}

.table {
```

  margin-top: 30px;

}

.mb-3 {

  margin-top: 20px;

}

---

6. Run the Application

Step 6.1: Start the Backend (Node.js)

In your terminal, go to the root directory of the project and run:

```bash

node server.js

```

This will start your backend server at `http://localhost:8000`.

Step 6.2: Start the Frontend (React)

Open another terminal window, navigate to the `client` folder, and run:

```bash

npm start

```

This will start the React development server at `http://localhost:3000`.

---

7. Testing the Application

- Open your browser and go to `http://localhost:3000`.

- You should see the form to enter an email address.

- After submitting, the entered email should be inserted into your MySQL database and displayed in the table below.

---

8. Conclusion

You've now successfully created a full-stack application with **React** as the frontend and **Node.js** with **MySQL** as the backend.