import React, { useState, useEffect } from 'react';

import axios from 'axios';

function App() {

  const [table, setTable] = useState([]);

  const [selectTable, setSelectTable] = useState('');

  const [student, setStudent] = useState([]);

  const [prn, setPrn] = useState('');

  const [cgpa, setCgpa] = useState('');

  const [course, setCourse] = useState('');

  const [edit, setEdit] = useState(null);

  useEffect(() => {

    fetchTable();

  }, []);

  const fetchTable = async () => {

    try {

      const response = await axios.get('http://localhost:3000/api/tables');

      setTable(response.data);

    } catch (error) {

      console.error('Error Fetched tables', error);

    }

  };

  const fetchStudent = async () => {

    if (!selectTable) {

      return;

    }

    try {

      const response = await axios.get(`http://localhost:3000/api/tables/${selectTable}`);

      setStudent(response.data);

    } catch (error) {

      console.error('Error', error);

    }

  };

  const handleChange = (table) => {

    setSelectTable(table);

    setPrn('');

    setCgpa('');

    setCourse('');

    setStudent([]);

    setEdit(null);

  };

  const handleSubmit = async (e) => {

    e.preventDefault();

    if (

      prn.trim() === '' ||

      cgpa.trim() === '' ||

      course.trim() === '' ||

      !selectTable

    ) {

      alert('Please Fill Correctly');

      return;

    }

    if (edit !== null) {

      try {

        await axios.put(`http://localhost:3000/api/<span class="math-inline">\{selectTable\}/</span>{student[edit].id}`, {

          prn,

          cgpa,

          course,

        });

        setEdit(null);

      } catch (error) {

        console.error('Error Updating data', error);

      }

    } else {

      try {

        await axios.post(`http://localhost:3000/api/${selectTable}`, {

          prn,

          cgpa,

          course,

        });

      } catch (error) {

        console.error('Error', error);

      }

    }

    setPrn('');

    setCgpa('');

    setCourse('');

    fetchStudent();

  };

  const handleEdit = (index) => {

    const studentToEdit = student[index];

    setPrn(studentToEdit.prn);

    setCgpa(studentToEdit.cgpa);

    setCourse(studentToEdit.course);

    setEdit(index);

  };

  const handleDelete = async (index) => {

    try {

      await axios.delete(`http://localhost:3000/api/<span class="math-inline">\{selectTable\}/</span>{student[index].id}`);

      fetchStudent();

    } catch (error) {

      console.error('error', error);

    }

  };

  return (

    <div className="App">

      <h1>Student Form</h1>

      <div className="Table">

        Select Table

      </div>

      <select value={selectTable} onChange={(e) => handleChange(e.target.value)}>

        <option key="" value="">

          Select a table

        </option>

        {table.map((table) => (

          <option key={table} value={table}>

            {table}

          </option>

        ))}

      </select>

      <form onSubmit={handleSubmit}>

        <label>

          PRN:

          <input type="text" value={prn} onChange={(e) => setPrn(e.target.value)} />

        </label>

        <label>

          CGPA:

 <input type="text" value={cgpa} onChange={(e) => setCgpa(e.target.value)} />

</label>

<label>

 Course:

 <input type="text" value={course} onChange={(e) => setCourse(e.target.value)} />

</label>

<button type="submit">{edit !== null ? 'Edit' : 'Add'}</button>

</form>

<table>

<thead>

 <tr>

   <th>PRN</th>

   <th>CGPA</th>

   <th>Course</th>

   <th>Action</th>

 </tr>

</thead>

<tbody>

 {student.map((student, index) => (

   <tr key={index}>

     <td>{student.prn}</td>

     <td>{student.cgpa}</td>

     <td>{student.course}</td>

     <td>

       <button onClick={() => handleEdit(index)}>Edit</button>

       <button onClick={() => handleDelete(index)}>Delete</button>

     </td>

   </tr>

 ))}

</tbody>

</table>

</div>

);

}

export default App;

const express = require('express');

const bodyParser = require('body-parser');

const cors = require('cors');

const mysql = require('mysql');

const app = express();

const port = 3000;

app.use(bodyParser.json());

app.use(cors());

const connection = mysql.createConnection({

  host: 'localhost',

  user: 'root',

  password: 'root123',

  database: 'sourabh',

});

connection.connect((err) => {

  if (err) {

    console.error('Error connecting to MySQL:', err);

    throw err;

  }

  console.log('Connected to MySQL');

});

app.get('/api/tables', (req, res) => {

    connection.query('SHOW TABLES', (err, results) => {

      if (err) {

        console.error('Error fetching tables:', err);

        res.status(500).send('Internal Server Error');

        return;

      }

      const tables = results.map((table) => table.Tables\_in\_sourabh);

      res.json(tables);

    });

  });

app.get('/api/:table', (req, res) => {

  const { table } = req.params;

  connection.query(`SELECT \* FROM ${table}`, (err, results) => {

    if (err) {

      console.error(`Error fetching data from ${table}:`, err);

      res.status(500).send('Internal Server Error');

      return;

    }

    res.json(results);

  });

});

app.post('/api/:table', (req, res) => {

  const { table } = req.params;

  const newData = req.body;

  connection.query(`INSERT INTO ${table} SET ?`, newData, (err, results) => {

    if (err) {

      console.error(`Error inserting data into ${table}:`, err);

      res.status(500).send('Internal Server Error');

      return;

    }

    res.sendStatus(201);

  });

});

app.put('/api/:table/:id', (req, res) => {

  const { table, id } = req.params;

  const updatedData = req.body;

  connection.query(`UPDATE ${table} SET ? WHERE id = ?`, [updatedData, id], (err, results) => {

    if (err) {

      console.error(`Error updating data in ${table}:`, err);

      res.status(500).send('Internal Server Error');

      return;

    }

    res.sendStatus(200);

  });

});

app.delete('/api/:table/:id', (req, res) => {

  const { table, id } = req.params;

  connection.query(`DELETE FROM ${table} WHERE id = ?`, id, (err, results) => {

    if (err) {

      console.error(`Error deleting data from ${table}:`, err);

      res.status(500).send('Internal Server Error');

      return;

    }

    res.sendStatus(200);

  });

});

app.listen(port, () => {

  console.log(`Server is running on port ${port}`);

});