Node MySql Express Angular (CRUD)

**folder**

🡪sql\_node\_angular

🡪backend

🡪frontend

-npm init

-npm i express

-npm i body-parser

-npm i mysql

-npm i nodemon

-npm i cors

**Connect to the mySsql Database**

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

const express = require('express');

const mysql = require('mysql');

const server = express();

server.use(bodyParser.json());

server.use(express.json());

// must use this for cross api call

let cors = require("cors");

server.use(cors());

// connection

const db =  mysql.createConnection({

    host:"localhost",

    user:"root",

    password:"",

    database:"sql\_node\_angular"

});

db.connect(function (error){

    if(error){

        console.log(`${error}error while connecting to the databse`);

    }else{

        console.log('App connected to the database');

    }

});

server.listen(8080,function check(error){

    if(error) console.log('Error while conneting to the server');

    else console.log("server started on port no 8080");

})

**Crud**

// insert to the database

server.post("/api/student/add",(req,res) =>{

    let details ={

        stdName: req.body.stdName,

        course: req.body.course,

        fee: req.body.fee,

    };

    let sql = "INSERT INTO stdDetail SET ?";

    db.query(sql,details,(error)=>{

        if(error){

            res.send({

                status:false, message:"faild to create student id"

            });

        }else{

            res.send({status:true , message:"student id created successfully "});

        }

    });

});

// all student view

server.get("/api/student/get",(req,res)=>{

    let id = req.body;

    let sql  = "SELECT \* FROM stdDetail";

    db.query(sql,function(error,result){

    if(error){

        console.log("error connecting ")

    }else{

        res.send({status:true,data:result});

    }

});

});

// get student by id

server.get("/api/student/get/:id",(req,res)=>{

    let id = req.params.id;

    let sql  = "SELECT \* FROM stdDetail WHERE id="+id;

    db.query(sql,function(error,result){

    if(error){

        console.log("error connecting ")

    }else{

        res.send({status:true,data:result});

    }

});

});

// update student data

server.put('/api/student/update/:id',(req,res)=>{

    let sql = "UPDATE stdDetail SET stdName ='"+ req.params.stdName +"',course = '"+req.params.course+"',fee = '"+req.params.fee+"' WHERE id='"+req.params.id+"'";

    let query = db.query(sql,(error,result)=>{

        if(error){

            res.send({status:false,message:`${error}`});

        }else{

            res.send({status:true,message:"student updated"});

        }

    });

});

// delete student data

server.delete('/api/student/delete/:id',(req,res)=>{

    // let id = req.params.id;

    let sql = "DELETE FROM stdDetail WHERE id ='"+req.params.id+"'";

    let query  = db.query(sql,(error,result)=>{

        if(error){

            res.send({status:false,message:`${error}`});

        }else{

            res.send({status:true , message:"student deleted"});

        }

    });

});

**Create frontend (frontend)**

-ng new frontend

**App.module.ts**

Import

import { FormsModule } from '@angular/forms';

import { HttpClientModule } from '@angular/common/http';

import { RouterModule,Routes } from '@angular/router';

const appRutes:Routes = [

  {path:'',component:StudentCrudComponent},

]

@NgModule({

  declarations: [

    AppComponent,

    StudentCrudComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    FormsModule,

    HttpClientModule,

    RouterModule.forRoot(appRutes),

  ],

Add bootstrap

-npm i bootstrap

Create Component **studetnComponent**

-ng g c student\_crud

**Ts-file**

import { Component, OnInit } from '@angular/core';

// import { DataService } from '../service/data.service';

import { HttpClient } from '@angular/common/http';

@Component({

  selector: 'app-student-crud',

  templateUrl: './student-crud.component.html',

  styleUrls: ['./student-crud.component.css']

})

export class StudentCrudComponent {

  // DataService: any;

  // constructor(private dataService:DataService, private http:HttpClient){}

  // ngOnInit(): void {

  //   this.getStdData();

  //   throw new Error('Method not implemented.');

  // }

  // std:any;

  // getStdData(){

  //   this.dataService.getData().subscribe((res: any)=>{

  //     this.std = res;

  //     console.log(this.std);

  //   });

  // }

stdArray: any[]=[];

isResult = false;

isupdateformactive=false;

stdName:string="";

course:string="";

fee:string="";

stdId = "";

constructor(private http:HttpClient){

  this.getAllStd();

}

getAllStd(){

  this.http.get("http://127.0.0.1:8000/api/student/get").subscribe((res:any)=>{

    this.isResult=true;

    console.log(res.data);

    this.stdArray=res.data;

  })

}

// single functiion for update and insert

// if request comes with id then it will call update method otherwise insert the new data

save(){

  if(this.stdId ==""){

    this.register();

  }else{

    this.update();

  }

};

register(){

    let bodyData={

      "stdName":this.stdName,

      "course":this.course,

      "fee":this.fee

    };

    this.http.post("http://127.0.0.1:8000/api/student/add",bodyData).subscribe((res:any)=>{

    console.log('data added');

    this.getAllStd();

    });

}

// this is for show the data for update in the form

editDisplay(arr:any){

  // console.log(arr);

  this.stdId = arr.id;

  this.stdName =arr.stdName;

  this.course = arr.course;

  this.fee = arr.fee;

}

update(){

  let updateData={

    "stdName":this.stdName,

    "course":this.course,

    "fee":this.fee

  };

  this.http.put("http://127.0.0.1:8000/api/student/update/"+this.stdId,updateData).subscribe((res:any)=>{

      alert('student update....');

      this.stdId ="";

      this.stdName ="";

      this.course = "";

      this.fee = "";

  });

  this.getAllStd();

}

delete(arr:any){

  this.http.delete("http://127.0.0.1:8000/api/student/delete/"+arr).subscribe((res)=>{

    alert('student deleted...');

  });

  this.getAllStd();

}

}

**Html file**

<div class="col-md-3"></div>

<div class="col-md-6">

<form>

    <div class="form-group">

      <input type="text" [(ngModel)]="stdId" [ngModelOptions]="{standalone:true}" >

      <label for="name">Student Name</label>

      <input type="name" [(ngModel)]="stdName" [ngModelOptions]="{standalone:true}" class="form-control" id="name"  placeholder="Enter student Name">

    </div>

    <div class="form-group">

      <label for="course">Course</label>

      <input type="Course" [(ngModel)]="course" [ngModelOptions]="{standalone:true}" class="form-control" id="course" placeholder="Course">

    </div>

    <div class="form-group">

        <label for="Fee">Fee</label>

        <input type="Fee" [(ngModel)]="fee" [ngModelOptions]="{standalone:true}" class="form-control" id="Fee" placeholder="Fee">

      </div>

    <button type="submit" class="btn btn-primary" (click)="save()">Submit</button>

  </form>

<br>

    <table class="table">

       <thead>

        <h2 \*ngIf="!isResult">Loading.......</h2>

       </thead>

        <tbody>

          <tr \*ngFor="let arr of stdArray">

            <th scope="row">{{arr.id}}</th>

            <td>{{arr.stdName}}</td>

            <td>{{arr.course}}</td>

            <td>{{arr.fee}}</td>

            <td>

              <button type="button" (click)="editDisplay(arr)">Edit</button>

              <button type="button" (click)="delete(arr.id)">Delete</button>

            </td>

            </tr>

        </tbody>

      </table>

</div>

<div class="col-md-3"></div>