**Objective:-** Create node endpoints using express js to :

* store details of the new student
* fetch all students details
* update student by id
* delete the student by id

use MySQL database

**Code :-**

**index.js**

const express = require('express');

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

const cors = require('cors');

const markRoutes = require('./src/marks/marks.routes');

const db = require('./src/config/db.config');

const app = express();

const PORT = 3000;

app.use(cors());

app.use(bodyParser.json());

app.use('/api/students', markRoutes);

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

  res.send('Student Result Management System API');

});

app.listen(PORT, () => {

  console.log(`Server running at  http://localhost:${PORT}`);

});

**Src/config/db.config.js**

const mysql = require('mysql2');

const db = mysql.createConnection({

  host: 'localhost',

  user: 'root',

  password: '',

  database: 'student\_results'

});

db.connect((*err*) => {

  if (err) {

    console.error('Database connection failed: ' + err.stack);

    return;

  }

  console.log('Connected to MySQL database.');

});

module.exports = db;

**Src/marks/marks.controller.js**

const studentService = require('./marks.services');

exports.createStudent = (*req*, *res*) => {

  const student = req.body;

  studentService.createStudent(student, (*err*, *result*) => {

    if (err) {

      res.status(500).send({ message: 'Error creating student', error: err });

    } else {

      res.status(201).send({ message: 'Student created', data: result });

    }

  });

};

exports.getAllStudents = (*req*, *res*) => {

  studentService.getAllStudents((*err*, *results*) => {

    if (err) {

      res.status(500).send({ message: 'Error fetching students', error: err });

    } else {

      res.status(200).send(results);

    }

  });

};

exports.getStudentById = (*req*, *res*) => {

  const id = req.params.id;

  studentService.getStudentById(id, (*err*, *result*) => {

    if (err) {

      res.status(500).send({ message: 'Error fetching student', error: err });

    } else {

      res.status(200).send(result[0]);

    }

  });

};

exports.updateStudent = (*req*, *res*) => {

  const id = req.params.id;

  const student = req.body;

  studentService.updateStudent(id, student, (*err*, *result*) => {

    if (err) {

      res.status(500).send({ message: 'Error updating student', error: err });

    } else {

      res.status(200).send({ message: 'Student updated successfully' });

    }

  });

};

exports.deleteStudent = (*req*, *res*) => {

  const id = req.params.id;

  studentService.deleteStudent(id, (*err*, *result*) => {

    if (err) {

      res.status(500).send({ message: 'Error deleting student', error: err });

    } else {

      res.status(200).send({ message: 'Student deleted successfully' });

    }

  });

};

**Src/marks/marks.model.js**

const db = require('../config/db.config');

class Student {

  static create(*student*, *callback*) {

    const query = `

      INSERT INTO students (roll\_no, name, math, science, english, geography, computer , history)

      VALUES (?, ?, ?, ?, ?, ?, ?, ?)`;

    db.query(query, [student.roll\_no, student.name, student.math, student.science, student.english, student.geography, student.computer , student.history], callback);

  }

  static findAll(*callback*) {

    const query = 'SELECT \* FROM students';

    db.query(query, callback);

  }

  static findById(*id*, *callback*) {

    const query = 'SELECT \* FROM students WHERE id = ?';

    db.query(query, [id], callback);

  }

  static update(*id*, *student*, *callback*) {

    const query = `

      UPDATE students

      SET roll\_no = ?, name = ?, math = ?, science = ?, english = ?, geography = ?, computer = ?, history = ?

      WHERE id = ?`;

    db.query(query, [student.roll\_no, student.name, student.math, student.science, student.english, student.geography, student.computer, student.history , id], callback);

  }

  static delete(*id*, *callback*) {

    const query = 'DELETE FROM students WHERE id = ?';

    db.query(query, [id], callback);

  }

}

module.exports = Student;

**Src/marks/marks.routes.js**

const express = require('express');

const router = express.Router();

const markController = require('./marks.controller');

router.post('/', markController.createStudent);

router.get('/', markController.getAllStudents);

router.get('/:id', markController.getStudentById);

router.put('/:id', markController.updateStudent);

router.delete('/:id', markController.deleteStudent);

module.exports = router;

**Src/marks/marks.servvices.js**

const Student = require('./marks.model');

const createStudent = (*data*, *callback*) => {

  Student.create(data, callback);

};

const getAllStudents = (*callback*) => {

  Student.findAll(callback);

};

const getStudentById = (*id*, *callback*) => {

  Student.findById(id, callback);

};

const updateStudent = (*id*, *data*, *callback*) => {

  Student.update(id, data, callback);

};

const deleteStudent = (*id*, *callback*) => {

  Student.delete(id, callback);

};

module.exports = {

  createStudent,

  getAllStudents,

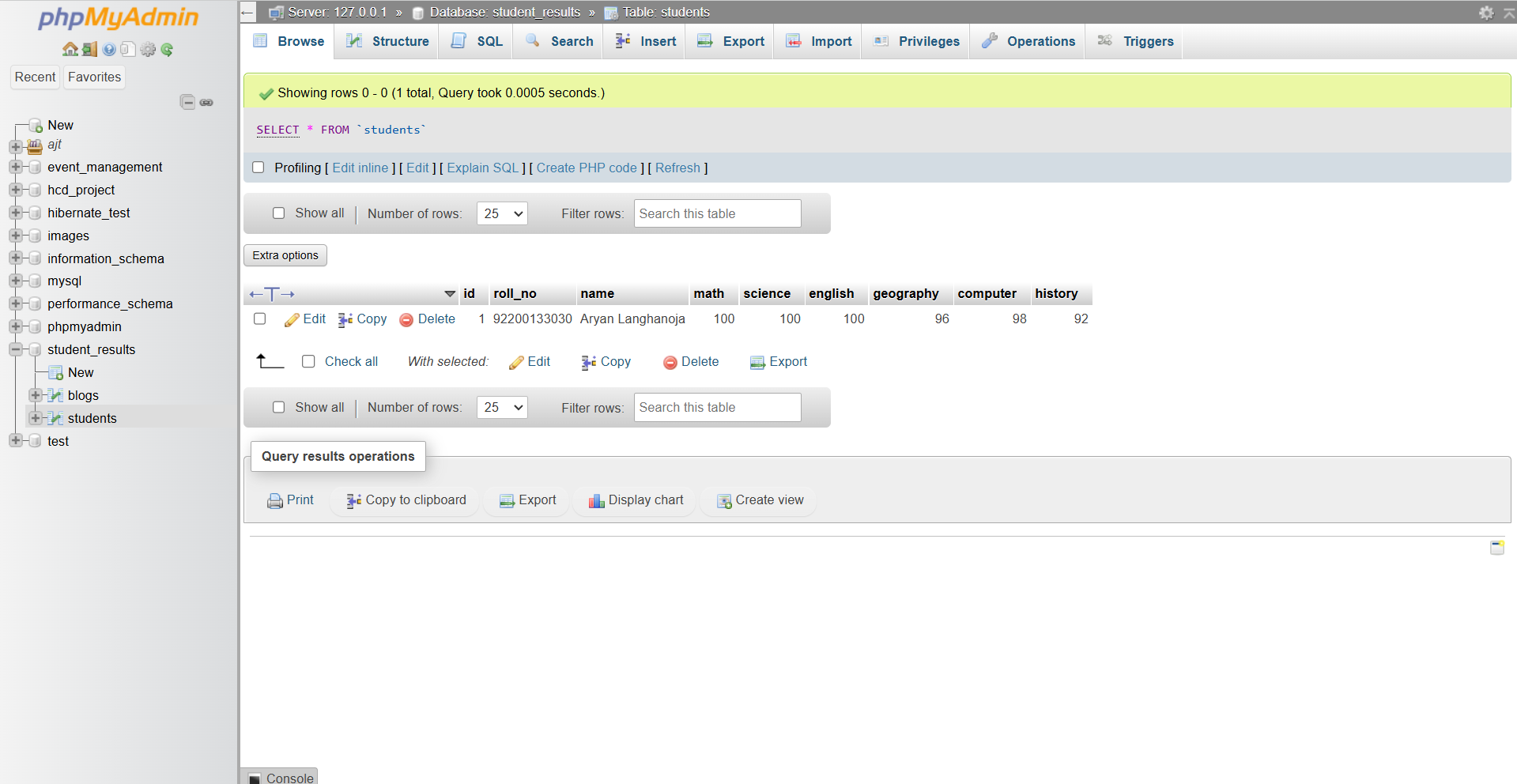
  getStudentById,

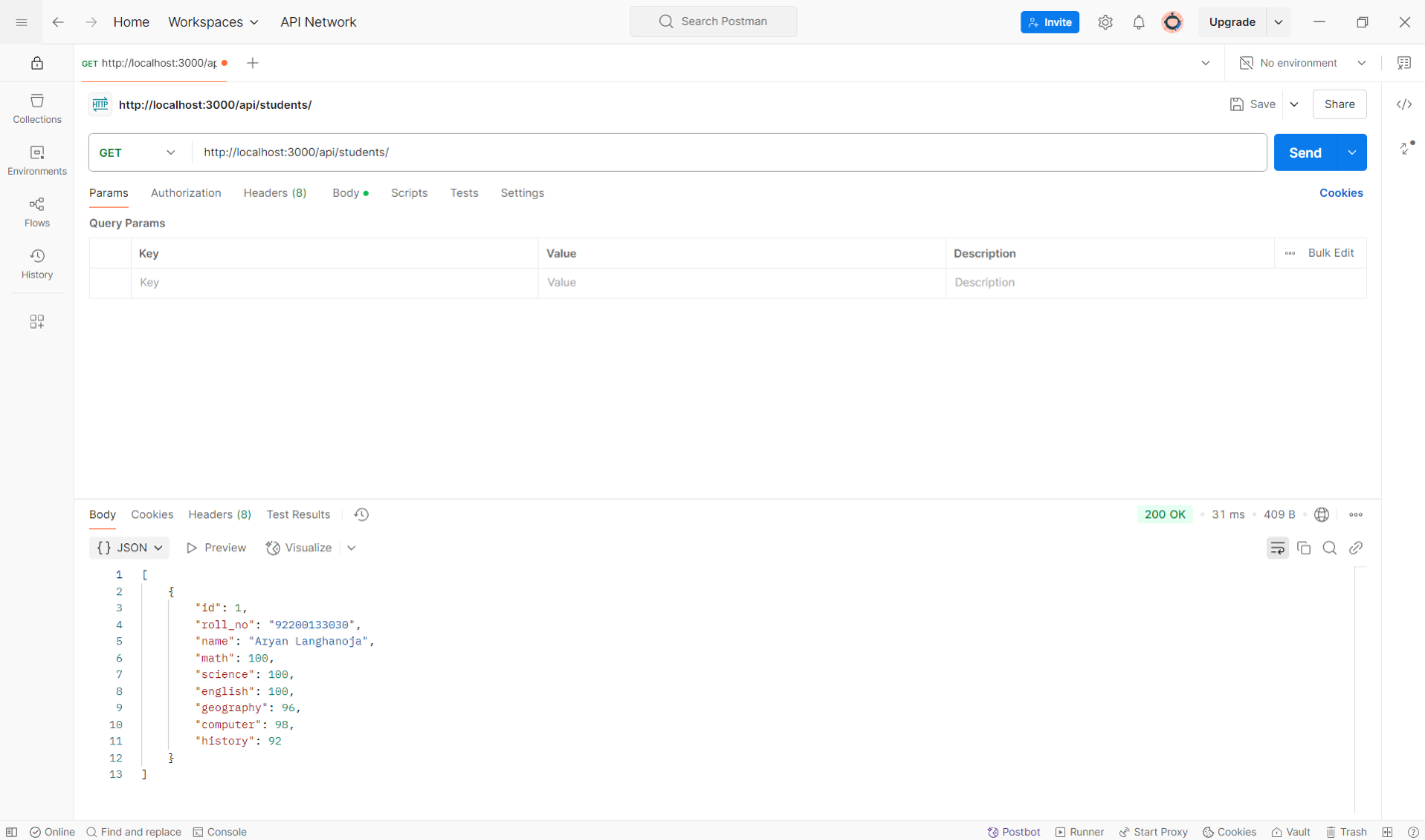
  updateStudent,

  deleteStudent

};

**Output :-**



****

