**MEAN STACK**

Index.html

<!DOCTYPE html>

<html>

<head>

    <title>Student Management System</title>

    <style>

</head>

<body ng-app="studentApp" ng-controller="studentController">

    <h1>Student Management System</h1>

    <form ng-submit="addStudent()">

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

        <input type="text" id="name" ng-model="newStudent.name" required>

        <label for="age">Age:</label>

        <input type="number" id="age" ng-model="newStudent.age" required>

        <label for="dept">Department:</label>

        <input type="text" id="dept" ng-model="newStudent.dept" required>

        <button type="submit">Add Student</button>

    </form>

    <table>

        <tr>

            <th>Name</th>

            <th>Age</th>

            <th>Department</th>

            <th>Actions</th>

        </tr>

        <tr ng-repeat="student in students">

            <td>{{ student.name }}</td>

            <td>{{ student.age }}</td>

            <td>{{ student.dept }}</td>

            <td>

                <button ng-click="deleteStudent(student.\_id)">Delete</button>

            </td>

        </tr>

    </table>

    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

    <script>

        angular.module('studentApp', [])

            .controller('studentController', function (*$scope*, *$http*) {

                $scope.students = [];

                $scope.newStudent = {};

*// Retrieve student data*

                $http.get('http://localhost:4000/students')

                    .then(*response* => {

                        $scope.students = response.data;

                    })

                    .catch(*error* => {

                        console.error(error);

                    });

*// Add a new student*

                $scope.addStudent = () => {

                    $http.post('http://localhost:4000/students', $scope.newStudent)

                        .then(*response* => {

                            $scope.newStudent = {};

                        })

                        .catch(*error* => {

                            console.error(error);

                        });

                        window.location.reload();

                };

*// Delete a student*

                $scope.deleteStudent = (*studentId*) => {

                    $http.delete(`http://localhost:4000/students/${studentId}`)

                        .then(() => {

                            window.location.reload();

                        })

                        .catch(*error* => {

                            console.error(error);

                        });

                };

            });

    </script>

</body>

</html>

Server.js

const express = require('express');

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

const mongoose = require('mongoose');

const cors = require('cors');

const app = express();

app.use(cors());

app.use(bodyParser.json());

const port = 4000;

app.listen(port, () => {

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

});

*// Connect to MongoDB*

mongoose.connect('mongodb://127.0.0.1:27017/studentDB');

*// Create a schema for student data*

const studentSchema = new mongoose.Schema({

    name: String,

    age: Number,

    dept: String

}, { collection: 'student' });

*// Create a model based on the schema*

const Student = mongoose.model('Student', studentSchema);

*// Handle form submission*

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

    const newStudent = new Student({

        name: req.body.name,

        age: req.body.age,

        dept: req.body.dept

    });

    newStudent.save()

        .then(() => {

            res.redirect('/');

        })

        .catch(*err* => {

            console.error(err);

            res.status(500).send('Error saving student data');

        });

});

*// Retrieve student data*

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

    Student.find()

        .then(*students* => {

            res.json(students);

        })

        .catch(*err* => {

            console.error(err);

            res.status(500).send('Error retrieving student data');

        });

});

*// Delete a student*

app.delete('/students/:studentId', (*req*, *res*) => {

    const studentId = req.params.studentId;

    Student.findByIdAndDelete(studentId)

        .then(() => {

            res.sendStatus(204); *// No Content*

        })

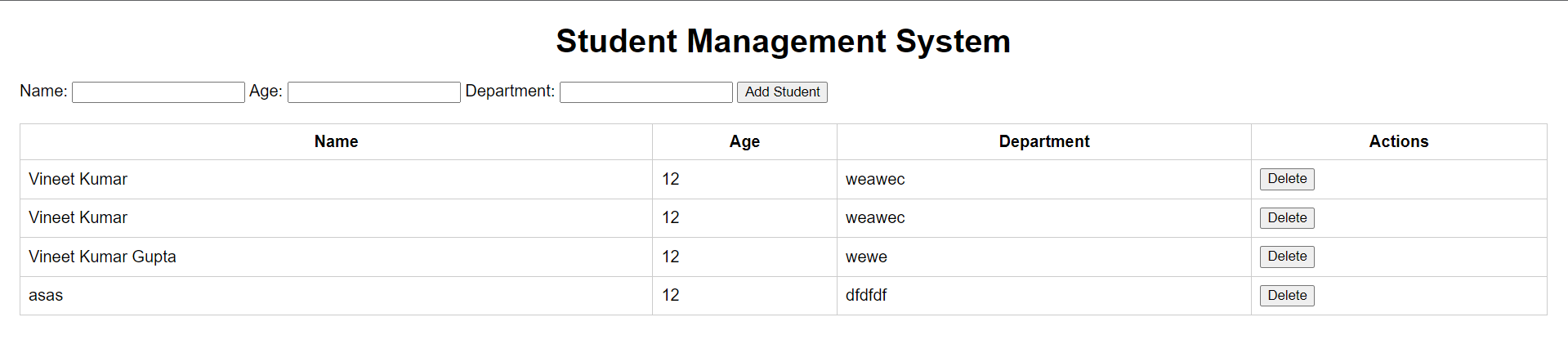
        .catch(*err* => {

            console.error(err);

            res.status(500).send('Error deleting student');

        });

});



**RESTful API Services**

Index.html

<!DOCTYPE html>

<html>

<head>

    <title>RESTful API Calculator</title>

</head>

<body>

    <h1>RESTful API Calculator</h1>

    <form id="calculatorForm">

        <label for="num1">Number 1:</label>

        <input type="number" name="num1" id="num1" required>

        <br>

        <label for="num2">Number 2:</label>

        <input type="number" name="num2" id="num2" required>

        <br>

        <label for="operator">Operator:</label>

        <select name="operator" id="operator" required>

            <option value="add">+</option>

            <option value="subtract">-</option>

            <option value="multiply">\*</option>

            <option value="divide">/</option>

        </select>

        <br>

        <button type="submit">Calculate</button>

    </form>

    <div id="result"></div>

    <script>

        const form = document.getElementById('calculatorForm');

        const result = document.getElementById('result');

        form.addEventListener('submit', (*event*) => {

            event.preventDefault();

            const num1 = form.num1.value;

            const num2 = form.num2.value;

            const operator = form.operator.value;

            fetch(`http://localhost:3000/calculate/${operator}/${num1}/${num2}`)

                .then(*response* => response.json())

                .then(*data* => {

                    if (data.error) {

                        result.textContent = data.error;

                    } else {

                        result.textContent = data.result;

                    }

                })

                .catch(*error* => console.error(error));

        });

    </script>

</body>

</html>

server.js

const express = require('express');

const app = express();

const cors = require('cors');

app.use(express.json());

app.use(cors());

const port = 3000;

app.listen(port, () => console.log(`Listening on port ${port}...`));

app.get('/calculate/:operator/:num1/:num2', (*req*, *res*) => {

    const { operator, num1, num2 } = req.params;

    let result;

    switch (operator) {

      case 'add':

        result = Number(num1) + Number(num2);

        break;

      case 'subtract':

        result = Number(num1) - Number(num2);

        break;

      case 'multiply':

        result = Number(num1) \* Number(num2);

        break;

      case 'divide':

        result = Number(num1) / Number(num2);

        break;

      default:

        return res.status(400).send({ error: 'Invalid operator' });

    }

    res.send({ result });

  });