**AIM: Develop a web application which involves database operations using Express.**

**CODE:**

// server.js

const express = require('express');

const mysql = require('mysql');

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

const path = require('path');

const app = express();

const port = 3000;

// Middleware

app.use(bodyParser.urlencoded({ extended: false }));

app.use(express.static(path.join(\_\_dirname, 'public')));

// Database connection

const db = mysql.createConnection({

    host: 'localhost',

    user: 'root',

    password: '',

    database: 'student\_db'

});

// Connect to database

db.connect((err) => {

    if (err) {

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

        return; }

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

    // Create table if not exists

    const createTableQuery = `

        CREATE TABLE IF NOT EXISTS students (

            id INT AUTO\_INCREMENT PRIMARY KEY,

            name VARCHAR(100) NOT NULL,

            email VARCHAR(100) NOT NULL,

            course VARCHAR(100) NOT NULL

        ) `;

    db.query(createTableQuery, (err) => {

        if (err) {

            console.error('Error creating table:', err);

        } else {

            console.log('Students table created or already exists');

        }});

});

// Routes

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

    res.sendFile(path.join(\_\_dirname, 'public', 'index.html'));

});

// Get all students

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

    const query = 'SELECT \* FROM students';

    db.query(query, (err, results) => {

        if (err) {

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

            return res.status(500).json({ error: 'Error fetching students' });

        }

        res.json(results);

    }); });

// Add a new student

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

    const { name, email, course } = req.body;

    if (!name || !email || !course) {

        return res.status(400).json({ error: 'All fields are required' }); }

    const query = 'INSERT INTO students (name, email, course) VALUES (?, ?, ?)';

    db.query(query, [name, email, course], (err, result) => {

        if (err) {

            console.error('Error adding student:', err);

            return res.status(500).json({ error: 'Error adding student' });}

        res.redirect('/');

    }); });

// Delete a student

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

    const id = req.params.id;

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

    db.query(query, [id], (err, result) => {

        if (err) {

            console.error('Error deleting student:', err);

            return res.status(500).json({ error: 'Error deleting student' });  }

        res.redirect('/'); });

});

// Update a student

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

    const { id, name, email, course } = req.body;

    if (!id || !name || !email || !course) {

        return res.status(400).json({ error: 'All fields are required' }); }

    const query = 'UPDATE students SET name = ?, email = ?, course = ? WHERE id = ?';

    db.query(query, [name, email, course, id], (err, result) => {

        if (err) {

            console.error('Error updating student:', err);

            return res.status(500).json({ error: 'Error updating student' });}

        res.redirect('/');

    }); });

// Start server

app.listen(port, () => {

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

});

//index.html

<script>

        // Fetch and display students when page loads

        document.addEventListener('DOMContentLoaded', fetchStudents);

        function fetchStudents() {

            fetch('/api/students')

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

                .then(students => {

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

                    studentList.innerHTML = '';

                    students.forEach(student => {

                        const row = document.createElement('tr');

                        row.innerHTML = `

                            <td>${student.id}</td>

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

                            <td>${student.email}</td>

                            <td>${student.course}</td>

                            <td>

                                <button class="edit-btn" onclick="editStudent(${student.id}, '${student.name}', '${student.email}', '${student.course}')">

                                    Edit

                                </button>

                                <a href="/api/students/delete/${student.id}" class="delete-btn"

                                   onclick="return confirm('Are you sure you want to delete this student?')">

                                   Delete

                                </a>

                            </td>`;

                        studentList.appendChild(row); });})

                .catch(error => {

                    console.error('Error fetching students:', error); });}

        function editStudent(id, name, email, course) {

            // Show edit form and hide add form

            document.getElementById('editForm').style.display = 'block';

            document.getElementById('addForm').style.display = 'none';

            // Populate edit form with student data

            document.getElementById('editId').value = id;

            document.getElementById('editName').value = name;

            document.getElementById('editEmail').value = email;

            document.getElementById('editCourse').value = course; }

        function cancelEdit() {

            // Hide edit form and show add form

            document.getElementById('editForm').style.display = 'none';

            document.getElementById('addForm').style.display = 'block'; }

    </script>

<script>

  async function loadStudents() {

    const res = await fetch('/api/students');

    const data = await res.json();

    let html = '<ul>';

    data.forEach(s => {

      html += `<li>${s.name} - ${s.branch}

                <form action="/delete/${s.\_id}" method="POST" style="display:inline;">

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

                </form>

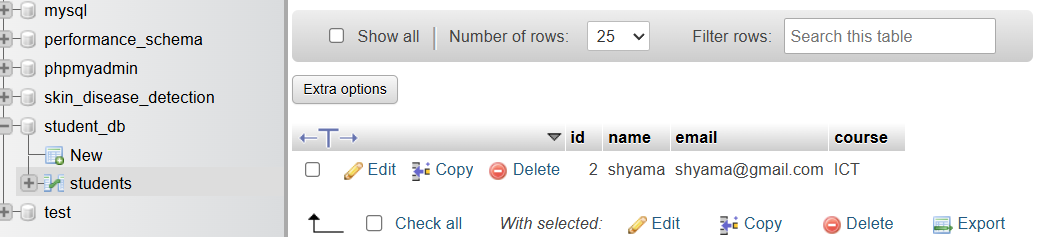
              </li>`;});

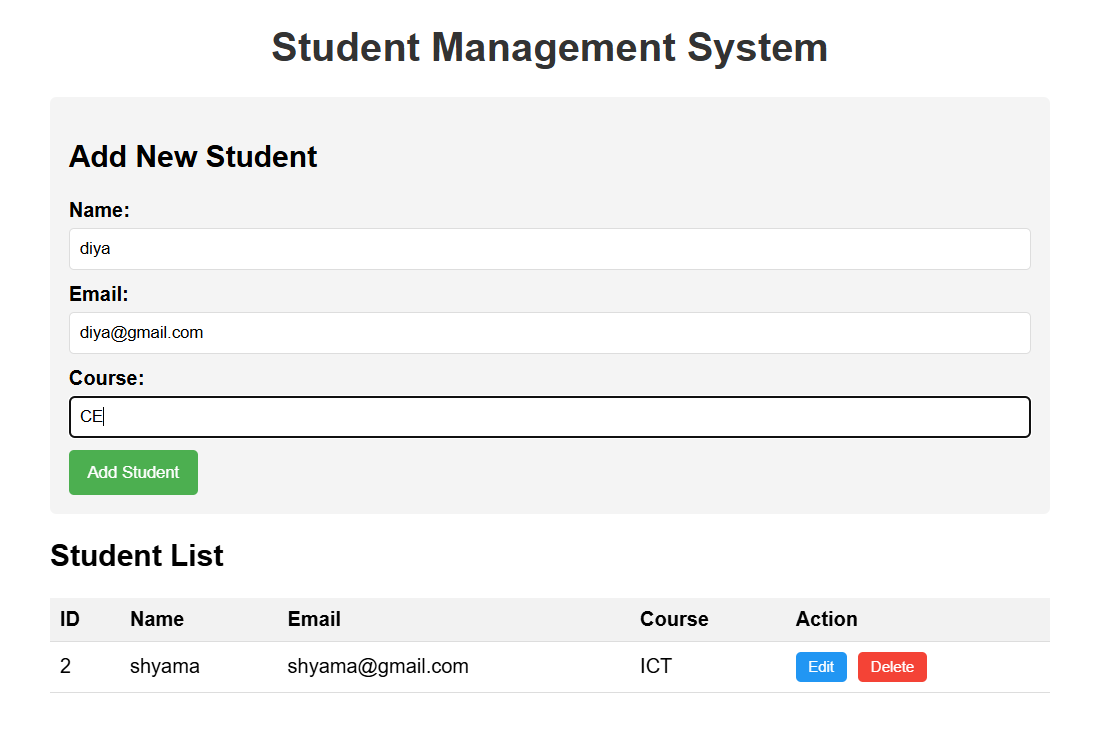
    html += '</ul>';

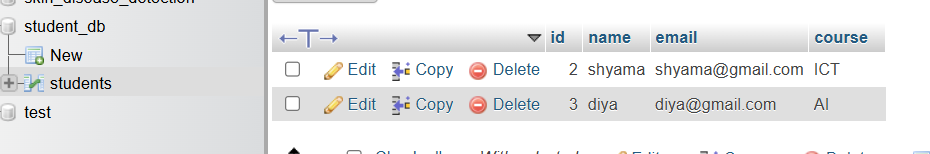
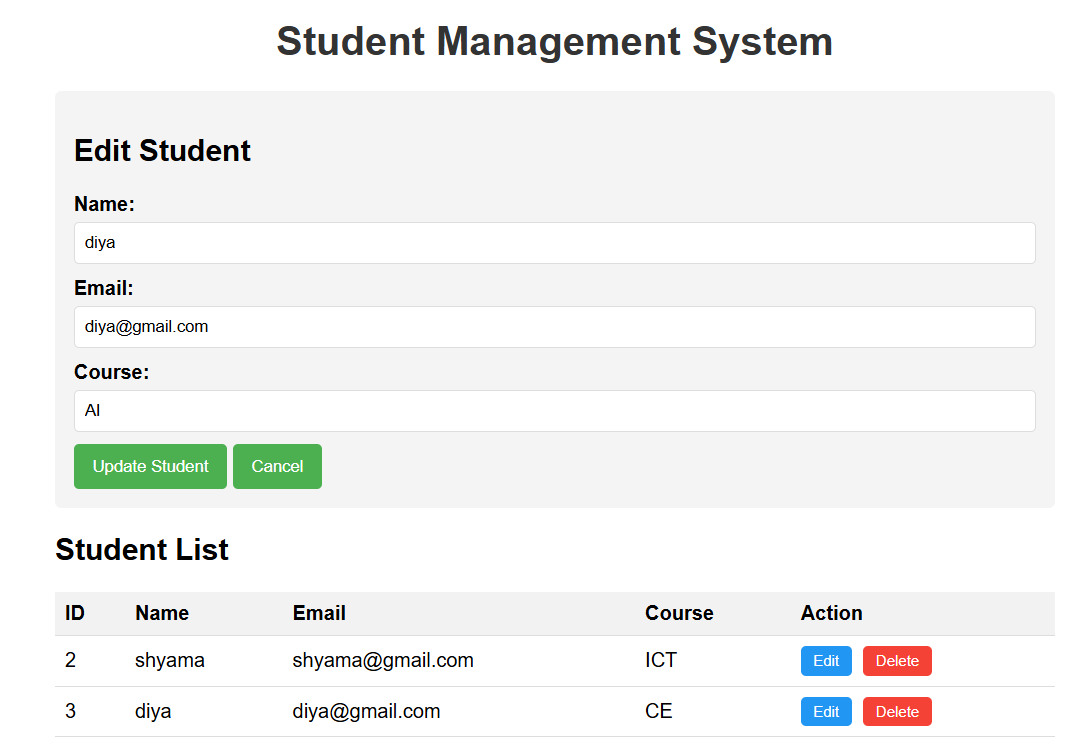
    document.getElementById('students').innerHTML = html;

  }

**OUTPUT:**

****

****

****