Assignment -1

<!doctype html>

<html lang="en">

<head>

    <!-- Required meta tags -->

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- CSS -->

    <link rel="stylesheet" href="style.css">

    <title>Form</title>

</head>

<script>

function validateForm(event) {

  event.preventDefault(); // Prevent form submission

  // Get form inputs

    var name = document.getElementById("name").value.trim();

    var email = document.getElementById("email").value.trim();

    var phone = document.getElementById("phone").value.trim();

    var dancestyle = document.getElementById("dancestyle").value;

    var team = document.querySelector('input[name="team"]:checked');

  // Validate name (non-empty)

  if (name == "" || email == "" || phone == "" || dancestyle == "" || !team) {

        alert("Please fill in all fields.");

        return false;

  }

  // Validate email (format)

  var emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

  if (!emailRegex.test(email)) {

    alert('Please enter a valid email address');

    return false;

  }

  if (phone.length !== 10 || isNaN(phone)) {

        alert("Please enter a 10-digit phone number.");

        return false;

    }

    if (/\d/.test(name)) {

        alert("Name should not contain numbers.");

        return false;

    }

  // If all validations pass, submit the form

  alert('Form submitted successfully');

  // Reset form fields

    document.getElementById("name").value = "";

    document.getElementById("email").value = "";

    document.getElementById("phone").value = "";

    document.getElementById("dancestyle").value = "";

    document.querySelector('input[name="team"]:checked').checked = false;

    return true;

}

</script>

<body>

    <div class="modal-wrap">

        <div class="modal-header">

            <div class="title">Ready to Dance</div>

        </div>

        <div class="modal-bodies">

            <div class="modal-body modal-body-step-1 is-showing">

                <div class="description">Registration Form</div>

                <form id="myForm" onsubmit="validateForm(event)">

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

                    <input type="text" id="name" name="name" required>

                    <label for="email">Email:</label>

                    <input type="email" id="email" name="email" required>

                    <label for="phone">Phone:</label>

                    <input type="text" id="phone" name="phone" required>

                    <label for="dancestyle">Preferred Dance Style:</label>

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

                        <option value="">Select a style</option>

                        <option value="Hip Hop">Hip Hop</option>

                        <option value="Ballet">Ballet</option>

                        <option value="Salsa">Salsa</option>

                        <option value="Contemporary">Contemporary</option>

                        <option value="Tap">Tap</option>

                        <option value="Other">Other</option>

                    </select><br>

                    <label for="team">Team:</label>

                    <div class="radio-group">

                        <label><input type="radio" name="team" value="Group"> Group</label>

                        <label><input type="radio" name="team" value="Solo"> Solo</label>

                    </div><br>

                    <button type="submit" class="button">Submit</button>

                  </form>

            </div>

        </div>

    </div>

</body>

</html>

html {

    background-image: url("https://i.pinimg.com/736x/bc/29/d7/bc29d709bc3e3e1c3d05e40b50f6cf90.jpg");

    min-height: 100%;

    font-family: 'Times New Roman', Times, serif;

  }

  .title {

    text-transform: uppercase;

    text-align: center;

    margin: auto;

    color:  maroon;

    font-size: 25px;

  }

  .description {

    text-align: center;

    color: black;

    font-size: 25px;

  }

  input[type="text"],

  input[type="email"] {

    border-radius: 3px;

    display: block;

    width: 100%;

    margin-bottom: 20px;

    box-sizing: border-box;

  }

  input[type="text"]:focus,

  input[type="email"]:focus {

    border-color: #ff6f00;

  }

  label {

    display: block;

    font-size: 20px;

    color: black;

    cursor: pointer;

  }

  .button{

    padding: 10px ;

    font-size: 20px;

    border-radius: 5px;

    background: maroon;

    color: white;

    margin-left: 150px;

    cursor: pointer;

  }

  .button:hover {

    background: orangered;

  }

  .text-center {

    text-align: center;

  }

  .modal-wrap {

    max-width: 600px;

    margin: 50px auto;

  }

  .modal-header {

    height: 45px;

    background: white;

    border-bottom: 1px solid #ccc;

    display: flex;

    justify-content: center;

    align-items: center;

  }

  .modal-bodies {

    position: relative;

  }

  .modal-body {

    background: white;

    padding: 40px 100px;

    box-shadow: 0px 50px 30px -30px rgba(0, 0, 0, 0.3);

    margin-bottom: 50px;

    position: absolute;

    display: none;

    box-sizing: border-box;

    width: 100%;

    border-radius: 10px;

  }

  .modal-body.is-showing {

    display: block;

  }

        input[type="text"],

        input[type="email"],

        select {

            width: 100%;

            padding: 8px;

            margin-bottom: 10px;

            border: 1px solid #ccc;

            border-radius: 4px;

        }

  .radio-group {

    display: flex;

  }

  .radio-group label {

    margin-right: 10px;

}

Assignment -2

Index.ejs

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Employee List</title>

    <!-- Add CSS styles -->

    <style>

        /\* General styles \*/

        body {

            font-family: 'Times New Roman', Times, serif;

            margin: 0;

            padding: 0;

            background-color: purple,palevioletred;

            color: #333;

        }

        .container {

            max-width: 800px;

            margin: 20px auto;

            padding: 0 20px;

        }

        h1 {

            text-align: center;

            margin-bottom: 20px;

        }

        table {

            width: 100%;

            border-collapse: collapse;

            margin-bottom: 20px;

            background-color:lightgreen;

            border-radius: 5px;

            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

        }

        th, td {

            border: 1px solid #ddd;

            padding: 8px;

            text-align: left;

        }

        th {

            background-color: #f2f2f2;

        }

        tr:nth-child(even) {

            background-color: #f9f9f9;

        }

        tr:hover {

            background-color: #f0f0f0;

        }

        a {

            display: block;

            text-align: center;

            padding: 10px 0;

            background-color: lightgreen;

            color: black;

            text-decoration: none;

            border-radius: 5px;

            transition: background-color 0.3s;

        }

        a:hover {

            background-color: #f0f0f0;

        }

    </style>

</head>

<body>

    <div class="container">

        <h1>Employee List</h1>

        <table>

            <thead>

                <tr>

                    <th>Name</th>

                    <th>Age</th>

                    <th>Position</th>

                    <th>Salary</th>

                </tr>

            </thead>

            <tbody>

                <% employees.forEach(employee => { %>

                    <tr>

                        <td><%= employee.name %></td>

                        <td><%= employee.age %></td>

                        <td><%= employee.position %></td>

                        <td>$<%= employee.salary %></td>

                    </tr>

                <% }) %>

            </tbody>

        </table>

        <a href="/add">Add Employee</a> <!-- Link to the add employee form -->

    </div>

</body>

</html>

Form.ejs

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Add Employee</title>

    <!-- Add any CSS styles or links here -->

    <style>

        /\* General styles \*/

        body {

            font-family: Arial, sans-serif;

            background-color: #f4f4f4;

            color: #333;

        }

        .container {

            max-width: 400px;

            margin: 20px auto;

            padding: 20px;

            background-color: #fff;

            border-radius: 5px;

            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

        }

        h1 {

            text-align: center;

            margin-bottom: 20px;

        }

        form {

            margin-bottom: 20px;

        }

        label {

            display: block;

            margin-bottom: 5px;

            font-weight: bold;

        }

        input[type="text"],

        input[type="number"] {

            width: 100%;

            padding: 8px;

            margin-bottom: 10px;

            border: 1px solid #ccc;

            border-radius: 5px;

            box-sizing: border-box;

        }

        input[type="submit"] {

            background-color: #007bff;

            color: #fff;

            padding: 10px 20px;

            border: none;

            border-radius: 5px;

            cursor: pointer;

            width: 100%;

            display: block;

        }

        input[type="submit"]:hover {

            background-color: #0056b3;

        }

        a {

            display: block;

            text-align: center;

            padding: 10px 0;

            background-color: #ccc;

            color: #333;

            text-decoration: none;

            border-radius: 5px;

            transition: background-color 0.3s;

        }

        a:hover {

            background-color: #999;

        }

    </style>

</head>

<body>

    <div class="container">

        <h1>Add Employee</h1>

        <form action="/add" method="post">

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

            <input type="text" id="name" name="name" required>

            <label for="age">Age:</label>

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

            <label for="position">Position:</label>

            <input type="text" id="position" name="position" required>

            <label for="salary">Salary:</label>

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

            <input type="submit" value="Add Employee">

        </form>

        <a href="/">Back to Employee List</a> <!-- Link back to the employee list -->

    </div>

</body>

</html>

Server.js

const express = require('express');

const mongoose = require('mongoose');

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

const Employee = require('./models/employee');

const app = express();

const PORT = process.env.PORT || 3000;

// Connect to MongoDB database

mongoose.connect('mongodb://localhost:27017/employeesDB', {

    useNewUrlParser: true,

    useUnifiedTopology: true

}).then(() => console.log('Connected to MongoDB'))

    .catch(err => console.error('Error connecting to MongoDB:', err));

// Set up EJS as view engine

app.set('view engine', 'ejs');

// Middleware for parsing request bodies

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

// Route to render the employee list

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

    try {

        const employees = await Employee.find();

        res.render('index', { employees });

    } catch (err) {

        console.error(err);

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

    }

});

// Route to render the form for adding a new employee

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

    res.render('form');

});

// Route to handle adding a new employee

app.post('/add', async (req, res) => {

    const { name, age, position, salary } = req.body;

    const employee = new Employee({ name, age, position, salary });

    try {

        await employee.save();

        res.redirect('/');

    } catch (err) {

        console.error(err);

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

    }

});

// Start the server

app.listen(PORT, () => {

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

});

Employee.js

const mongoose = require('mongoose');

const employeeSchema = new mongoose.Schema({

    name: { type: String, required: true },

    age: { type: Number, required: true },

    position: { type: String, required: true },

    salary:{ type:Number,required: true}

});

const Employee = mongoose.model('Employee', employeeSchema);

module.exports = Employee;