## ****Step 1: Create an ASP.NET Core MVC Project****

1. **Open Visual Studio 2022**.
2. Click **"Create a new project"**.
3. Search for **"ASP.NET Core Web App (Model-View-Controller)"** and select it.
4. Click **Next**.
5. Name the project (e.g., MVCDemoApp).
6. Choose a location and click **Next**.
7. Select **.NET 6.0 or later** and click **Create**.

## ****📌 Step 2: Create the Model****

* We'll create a **simple Employee model** with validation.

### ****🔹 Create the**** Models/Employee.cs ****file****

1. **Right-click on the project** → **Add** → **New Folder** → Name it Models.
2. Inside the Models folder, create Employee.cs and add the following code:

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace EmployeeMVCFramework.Models

{

public class Employee

{

public int Id { get; set; }

[Required(ErrorMessage = "Name is required")]

[StringLength(50, ErrorMessage = "Name cannot exceed 50 characters")]

public string Name { get; set; }

[Required(ErrorMessage = "Email is required")]

[EmailAddress(ErrorMessage = "Invalid email format")]

public string Email { get; set; }

[Required(ErrorMessage = "Department is required")]

public string Department { get; set; }

}

}

## ****Step 3: Create the Controller****

* The controller **handles requests and sends data to the view**.

### ****🔹 Create**** Controllers/EmployeeController.cs

1. **Right-click on** Controllers **folder** → **Add** → **Controller**.
2. Select **"MVC Controller - Empty"**, name it EmployeeController.cs, and click **Add**.
3. Modify the controller:

using EmployeeMVCFramework.Models;

using System.Collections.Generic;

using System.Linq;

using System.Web.Mvc;

namespace EmployeeMVCFramework.Controllers

{

public class EmployeeController : Controller

{

// Temporary storage for employees (Simulating database)

private static List<Employee> \_employees = new List<Employee>();

// GET: Employee List

public ActionResult Index()

{

ViewBag.Message = TempData["Message"]; // Retrieve success message

return View(\_employees);

}

// GET: Create Employee Form

public ActionResult Create()

{

return View();

}

// POST: Save Employee

[HttpPost]

public ActionResult Create(Employee employee)

{

if (ModelState.IsValid) // Check if the model is valid

{

// Assign a unique ID to each employee

employee.Id = \_employees.Count > 0 ? \_employees.Max(e => e.Id) + 1 : 1;

\_employees.Add(employee);

TempData["Message"] = "Employee added successfully";

return RedirectToAction("Index");

}

// If model is invalid, return the form again with validation errors

return View(employee);

}

}

}

### ****Explanation****

* Index() → Displays a list of employees.
* Create() (GET) → Shows the employee creation form.
* Create(Employee emp) (POST) → Validates and adds employees to the list.

## ****Step 4: Create Views****

* Views display data and interact with the user.

### ****🔹 Create**** Views/Employee/Index.cshtml

1. **Right-click** Views **→ Add → New Folder → Name it** Employee.
2. Inside Views/Employee, create Index.cshtml and add:

@model List<MVCDemoApp.Models.Employee>

@{

ViewData["Title"] = "Employee List";

}

<h2>Employee List</h2>

@if (TempData["Message"] != null)

{

<div class="alert alert-success">@TempData["Message"]</div>

}

<table class="table table-bordered">

<thead>

<tr>

<th>Name</th>

<th>Email</th>

<th>Department</th>

</tr>

</thead>

<tbody>

@foreach (var emp in Model)

{

<tr>

<td>@emp.Name</td>

<td>@emp.Email</td>

<td>@emp.Department</td>

</tr>

}

</tbody>

</table>

<a class="btn btn-primary" href="/Employee/Create">Add Employee</a>

### ****Create**** Views/Employee/Create.cshtml

* This form **adds a new employee**.

@model MVCDemoApp.Models.Employee

@{

ViewData["Title"] = "Add Employee";

}

<h2>Add Employee</h2>

<form asp-action="Create" method="post">

<div class="mb-3">

<label class="form-label">Name</label>

<input asp-for="Name" class="form-control" />

<span class="text-danger" asp-validation-for="Name"></span>

</div>

<div class="mb-3">

<label class="form-label">Email</label>

<input asp-for="Email" class="form-control" />

<span class="text-danger" asp-validation-for="Email"></span>

</div>

<div class="mb-3">

<label class="form-label">Department</label>

<select asp-for="Department" class="form-control">

<option value="">Select Department</option>

<option value="HR">HR</option>

<option value="IT">IT</option>

<option value="Finance">Finance</option>

</select>

<span class="text-danger" asp-validation-for="Department"></span>

</div>

<button type="submit" class="btn btn-success">Submit</button>

</form>

<a href="/Employee/Index" class="btn btn-secondary">Back to List</a>