23DCS025-YATRI DUNGARANI

Design a simple web application to display an employee directory. This application will consist of a model, a controller, and a view. The goal is to demonstrate your understanding of the Model-View-Controller (MVC) pattern and the use of ASP.NET Core helpers for rendering data. Take input from user.

Tasks:

1) Create an Employee class with properties for EmployeeId, FirstName, LastName, JobTitle, and Email.
2) Implement an EmployeeController with an Index action that populates a list of at least three Employee objects and passes them to the view.
3) Create a strongly-typed Index.cshtml view that uses a foreach loop to iterate through the employee list and, using helpers, displays each employee's details with appropriate labels

SOLUTION:
Controllers/EmployeeController.cs

```
using Microsoft.AspNetCore.Mvc;
using EmployeeDirectory.Data;
using EmployeeDirectory.Models;
using System.Linq;

namespace EmployeeDirectory.Controllers
{
   public class EmployeeController : Controller
   {
      private readonly ApplicationDbContext _context;
      public EmployeeController(ApplicationDbContext context)
      {
         _context = context;
      }

      // Index: show employees (if DB empty, seed a few sample employees)
      public IActionResult Index()
      {
         var employees = _context.Employees.ToList();

         if (!employees.Any())
         {
            // sample data
            employees = new List<Employee>
            {
               new Employee { FirstName = "Yatri", LastName = "Dungarani", JobTitle =
   "Software Engineer", Email = "yatri@gmail.com" },
            new Employee { FirstName = "Swara", LastName = "Jariwala", JobTitle = "QA
Analyst", Email = "swara@gmail.com" },
            new Employee { FirstName = "Aesha", LastName = "Kalathiya", JobTitle =
"Product Manager", Email = "aesha@gmail.com" }
        };

        _context.Employees.AddRange(employees);
        _context.SaveChanges();
      }

      return View(employees);
    }

    // GET: Create form
    [HttpGet]
    public IActionResult Create()
    {
      return View();
    }

    // POST: Receive form and save to DB
    [HttpPost]
    [ValidateAntiForgeryToken]
    public IActionResult Create(Employee employee)
    {
      if (ModelState.IsValid)
      {
        _context.Employees.Add(employee);
        _context.SaveChanges();
        return RedirectToAction(nameof(Index));
      }

      return View(employee);
    }
  }
}
```

Models/Employee.cs

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace EmployeeDirectory.Models
{
  public class Employee
```

```
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int EmployeeId { get; set; }

        [Required]
        [StringLength(100)]
        public string FirstName { get; set; } = string.Empty;

        [Required]
        [StringLength(100)]
        public string LastName { get; set; } = string.Empty;

        [StringLength(100)]
        public string JobTitle { get; set; } = string.Empty;

        [Required]
        [EmailAddress]
        [StringLength(256)]
        public string Email { get; set; } = string.Empty;
    }
}
```

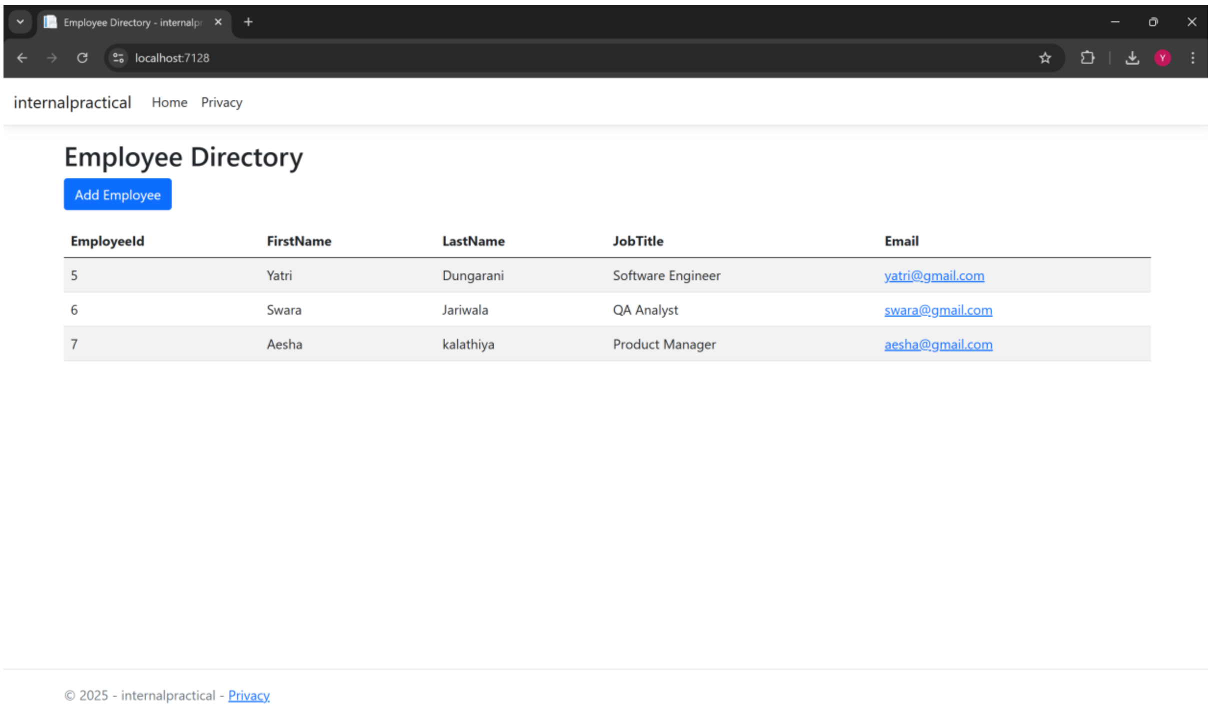
Views/Employee/Create.cshtml

```
@model EmployeeDirectory.Models.Employee

@{
    ViewData["Title"] = "Add Employee";
}

<h2>@ViewData["Title"]</h2>

<form asp-action="Create" method="post">
    <div class="form-group">
        <label asp-for="FirstName"></label>
        <input asp-for="FirstName" class="form-control" />
        <span asp-validation-for="FirstName" class="text-danger"></span>
    </div>

    <div class="form-group">
        <label asp-for="LastName"></label>
        <input asp-for="LastName" class="form-control" />
        <span asp-validation-for="LastName" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="JobTitle"></label>
        <input asp-for="JobTitle" class="form-control" />
        <span asp-validation-for="JobTitle" class="text-danger"></span>
    </div>

    <div class="form-group">
        <label asp-for="Email"></label>
        <input asp-for="Email" class="form-control" />
        <span asp-validation-for="Email" class="text-danger"></span>
    </div>

    <br />
    <button type="submit" class="btn btn-primary">Save</button>
    <a asp-action="Index" class="btn btn-secondary">Back to List</a>
</form>

@section Scripts {
    @{
        await Html.RenderPartialAsync("_ValidationScriptsPartial");
    }
}
```

Views/Employee/Index.cshtml

```
@model IEnumerable<EmployeeDirectory.Models.Employee>

@{
    ViewData["Title"] = "Employee Directory";
}

<h2>@ViewData["Title"]</h2>

<p>
    <a asp-action="Create" class="btn btn-primary">Add Employee</a>
</p>

<table class="table table-striped">
    <thead>
        <tr>
            <th>@Html.DisplayNameFor(model => model.First().EmployeeId)</th>
            <th>@Html.DisplayNameFor(model => model.First().FirstName)</th>
            <th>@Html.DisplayNameFor(model => model.First().LastName)</th>
            <th>@Html.DisplayNameFor(model => model.First().JobTitle)</th>
            <th>@Html.DisplayNameFor(model => model.First().Email)</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model)
        {
            <tr>
                <td>@Html.DisplayFor(modelItem => item.EmployeeId)</td>
                <td>@Html.DisplayFor(modelItem => item.FirstName)</td>
                <td>@Html.DisplayFor(modelItem => item.LastName)</td>
                <td>@Html.DisplayFor(modelItem => item.JobTitle)</td>
                <td>@Html.DisplayFor(modelItem => item.Email)</td>
            </tr>
        }
    </tbody>
</table>
```

SCREENSHOTS:

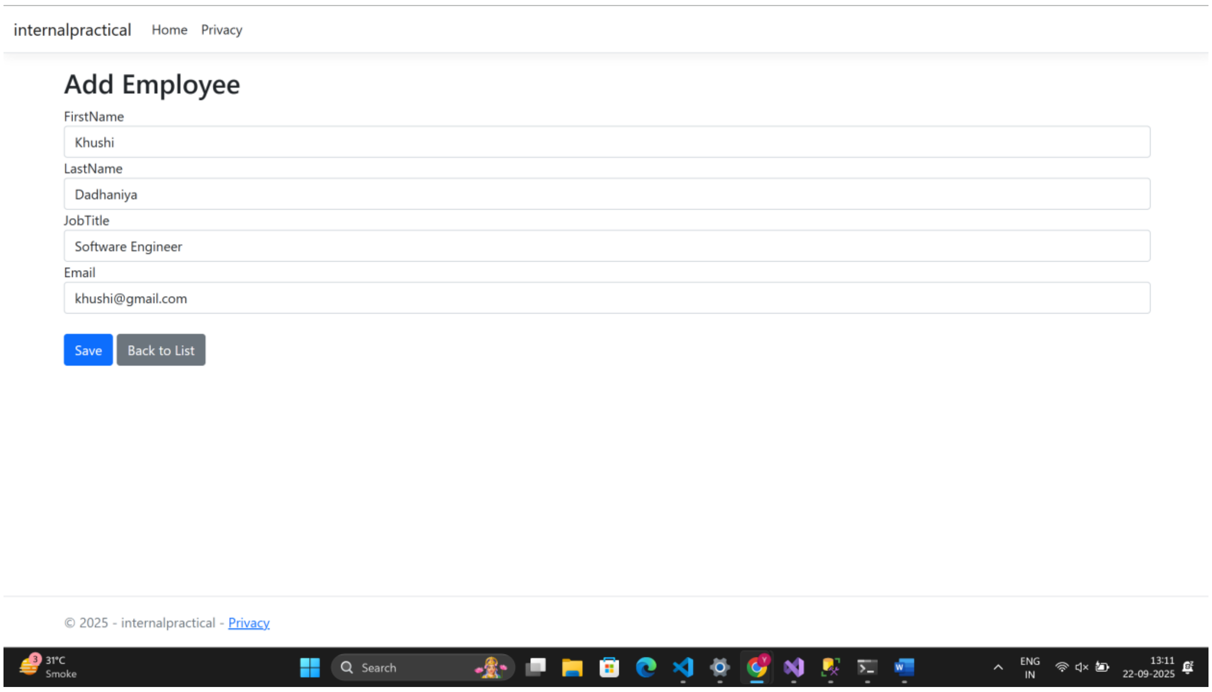


Employee Directory

EmployeeId	FirstName	LastName	JobTitle	Email
5	Yatri	Dungarani	Software Engineer	yatri@gmail.com
6	Swara	Jariwala	QA Analyst	swara@gmail.com
7	Aesha	kalathiya	Product Manager	aesha@gmail.com

© 2025 - internalpractical - Privacy

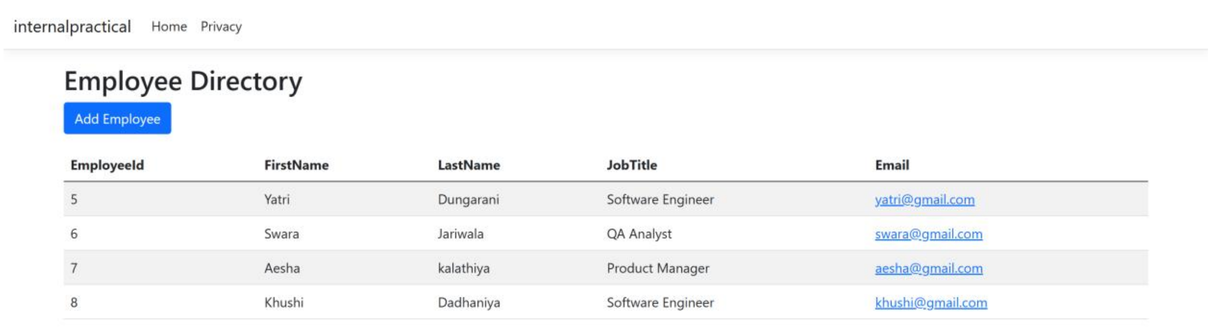
23DCS025-YATRI DUNGARANI


internalpractical Home Privacy

Add Employee

FirstName	Khushi
LastName	Dadhaniya
JobTitle	Software Engineer
Email	khushi@gmail.com

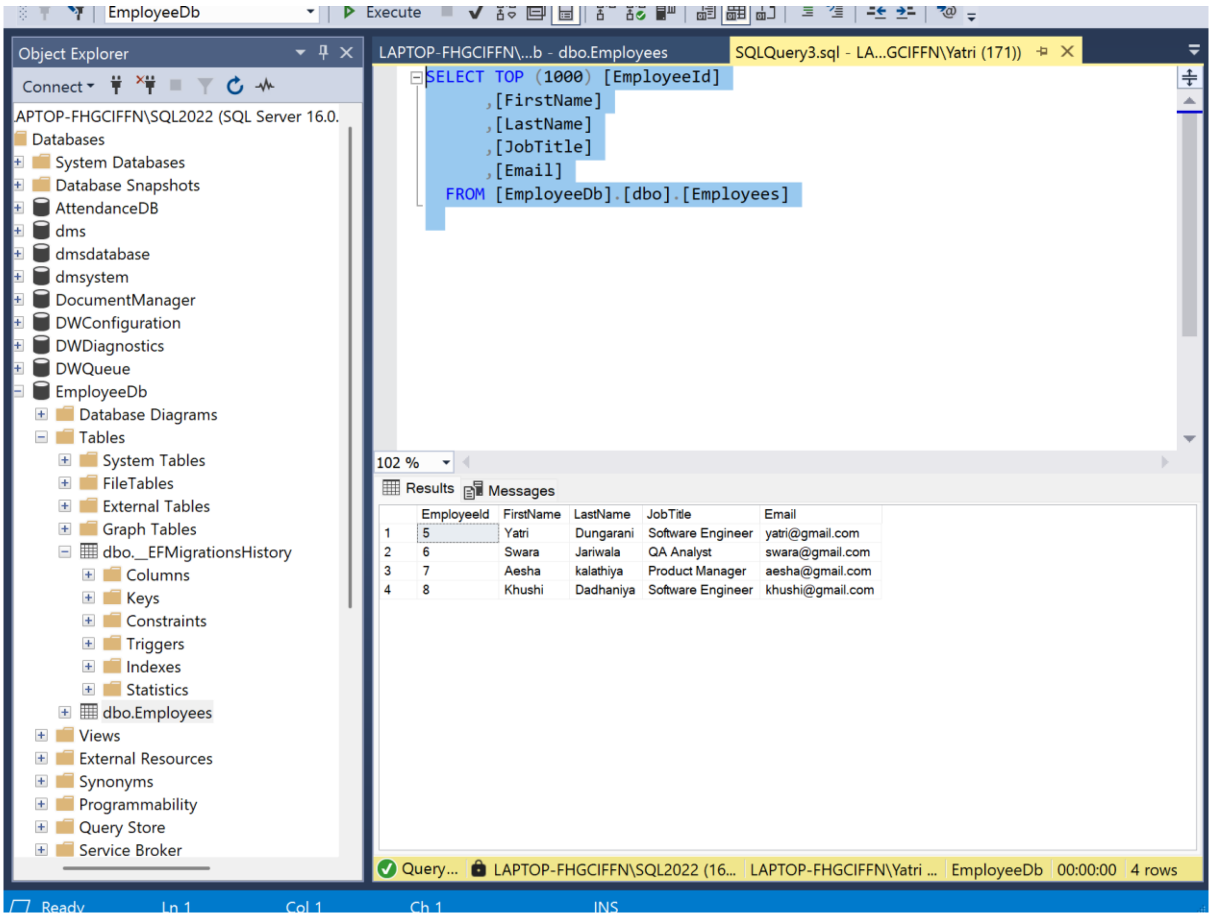
© 2025 - internalpractical - Privacy


internalpractical Home Privacy

Employee Directory

EmployeeId	FirstName	LastName	JobTitle	Email
5	Yatri	Dungarani	Software Engineer	yatri@gmail.com
6	Swara	Jariwala	QA Analyst	swara@gmail.com
7	Aesha	kalathiya	Product Manager	aesha@gmail.com
8	Khushi	Dadhaniya	Software Engineer	khushi@gmail.com

23DCS025-YATRI DUNGARANI



```sql
SELECT TOP (1000) [EmployeeId]
      ,[FirstName]
      ,[LastName]
      ,[JobTitle]
      ,[Email]
  FROM [EmployeeDb].[dbo].[Employees]
```

	EmployeeId	FirstName	LastName	JobTitle	Email
1	5	Yatri	Dungarani	Software Engineer	yatri@gmail.com
2	6	Swara	Jariwala	QA Analyst	swara@gmail.com
3	7	Aesha	kalathiya	Product Manager	aesha@gmail.com
4	8	Khushi	Dadhaniya	Software Engineer	khushi@gmail.com

```
Query... | LAPTOP-FHGCIFFN\SQL2022 (16... | LAPTOP-FHGCIFFN\Yatri ... | EmployeeDb | 00:00:00 | 4 rows
```