**ASP.NET Core MVC Registration Form Setup**

**Step 1: Create a New MVC Project**

Open the terminal and run the following command:

dotnet new mvc

**Step 2: Create the Registration Model and View**

**2.1 Create Model**

Create a model named RegisterEntityModel.cs under the Models folder with the following properties:

using System.ComponentModel.DataAnnotations;

namespace mvc\_with\_reg\_login.Models

{

public class RegisterEntityModel

{

public int Id { get; set; }

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

public string FirstName { get; set; }

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

public string LastName { get; set; }

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

[EmailAddress(ErrorMessage = "Invalid email address.")]

public string Email { get; set; }

[Required(ErrorMessage = "Username is required.")]

public string UserName { get; set; }

[Required(ErrorMessage = "Password is required.")]

public string Password { get; set; }

[Required(ErrorMessage = "Confirm Password is required.")]

[Compare("Password", ErrorMessage = "Passwords do not match.")]

public string ConfirmPassword { get; set; }

}

}

**2.2 Create the Registration View**

Under the Views/Home folder, create a new file named Register.cshtml and add the following form:

@model mvc\_with\_reg\_login.Models.RegisterEntityModel

@{

ViewData["Title"] = "Register";

}

<h2>Register</h2>

<form asp-action="Register" 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="Email"></label>

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

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

</div>

<div class="form-group">

<label asp-for="UserName"></label>

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

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

</div>

<div class="form-group">

<label asp-for="Password"></label>

<input asp-for="Password" class="form-control" type="password" />

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

</div>

<div class="form-group">

<label asp-for="ConfirmPassword"></label>

<input asp-for="ConfirmPassword" class="form-control" type="password" />

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

</div>

<button type="submit" class="btn btn-primary">Register</button>

</form>

@section Scripts {

@{await Html.RenderPartialAsync("\_ValidationScriptsPartial");}

}

**Step 3: Add Navigation Menu and Controller Actions**

**3.1 Add NavBar Menu Item**

Open \_Layout.cshtml under Views/Shared and add the following inside the <ul class="navbar-nav flex-grow-1 ms-auto">:

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Register">Register</a>

</li>

**3.2 Add Controller Actions**

Open HomeController.cs and add the following methods:

using mvc\_with\_reg\_login.Models;

public class HomeController : Controller

{

[HttpGet]

public IActionResult Register()

{

return View();

}

[HttpPost]

public IActionResult Register(RegisterEntityModel model)

{

if (ModelState.IsValid)

{

// Save the user to the database or perform any other action

return RedirectToAction("Index");

}

return View(model);

}

}

**Step 4: Run the Application**

In the terminal, run either of the following commands to see the registration form live:

dotnet watch run

or

dotnet run

**Step 5: Install Required Packages**

Install the following NuGet packages:

* Microsoft.EntityFrameworkCore
* Microsoft.EntityFrameworkCore.SqlServer

You can install them via the Package Manager Console:

Install-Package Microsoft.EntityFrameworkCore

Install-Package Microsoft.EntityFrameworkCore.SqlServer

Or from the NuGet Package Manager UI.

**Step 6: Add Connection Strings**

Open appsettings.json and add the following connection strings:

{

"ConnectionStrings": {

"DefaultConnection": "data source=.\\SQLEXPRESS; database=MyDatabase; User ID=test01; Password=Test@123; TrustServerCertificate=True;",

"DefaultConnection1": "data source=.\\SQLEXPRESS; database=MyDatabase; Integrated Security=True; TrustServerCertificate=True;"

}

}

* **DefaultConnection**: for SQL Server authentication with username and password.
* **DefaultConnection1**: for Windows Authentication (works if SQL Server is installed on the same system).

**Step 7: Create the Data Folder and AppDbContext**

Create a folder named Data under the project root (same level as Controllers).

Inside Data, create a new file named AppDbContext.cs:

using Microsoft.EntityFrameworkCore;

using mvc\_with\_reg\_login.Models;

namespace mvc\_with\_reg\_login.Data

{

public class AppDbContext : DbContext

{

public AppDbContext(DbContextOptions<AppDbContext> options) : base(options)

{

}

public DbSet<RegisterEntityModel> RegisterEntityModels { get; set; } = null!;

}

}

**Step 8: Register AppDbContext in Program.cs**

Open Program.cs and add the following code **above** var app = builder.Build();:

builder.Services.AddDbContext<AppDbContext>(options =>

options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));

This line registers AppDbContext with the connection string DefaultConnection.

**Step 9: Update Register method under HomeController.cs**

Open HomeController.cs and update the following methods:

Add package name on top after model

using mvc\_with\_reg\_login.Data;

**After that implement these inside the controller class**

private readonly ILogger<HomeController> \_logger;

private readonly AppDbContext \_context;

public HomeController(ILogger<HomeController> logger, AppDbContext context)

{

\_logger = logger;

\_context = context;

}

**Example : the below is implemented logic of database connectivity for registering table information to the table named “RegisterEntityModels”, to achieve that need to modify post method of Register**

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

private readonly AppDbContext \_context;

public HomeController(ILogger<HomeController> logger, AppDbContext context)

{

\_logger = logger;

\_context = context;

}

[HttpGet]

public IActionResult Register()

{

return View();

}

[HttpPost]

public IActionResult Register(RegisterEntityModel model)

{

if (ModelState.IsValid)

{

// Save the user to the database or perform any other action

\_context.RegisterEntityModels.Add(model);

// Save the RegisterViewModel (without ConfirmPassword)

\_context.SaveChanges();

\_logger.LogInformation("Registeration Completed successfully....");

return RedirectToAction("Index");

}

return View(model);

}

}

**Final Note**

You have now created a basic registration page in ASP.NET Core MVC with form validations, database connectivity setup, and context configuration!