.NET Internal Practical

Question:

A Hospital Management System is being developed using ASP.NET MVC. One of the key entities is the Patient model with the following properties: PatientId, Name and Disease. The system should allow adding new patients and updating existing patient details using MVC operations. Answer the following:

1) Write the ASP.NET MVC controller method for creating a new patient record.

2) Write the controller method for editing an existing patient record.

3) Design a Razor form to input patient details for the Create operation.

4) Explain how model binding works in ASP.NET MVC when submitting the Razor form.

5) Why is validation important in the Patient model, and how can it be implemented using Data Annotations?

Code:

Controllers/PatientController.cs

```
using Microsoft.AspNetCore.Mvc;

using HospitalManagement.Models;

using HospitalManagement.Data;

using System.Linq;


namespace HospitalManagement.Controllers

{

    public class PatientController : Controller

    {

        public IActionResult Index()

        {
```

```
        var list = PatientRepository.GetAll();
        return View(list);
    }

    [HttpGet]
    public IActionResult Create()
    {
        return View();
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public IActionResult Create(Patient patient)
    {
        if (!ModelState.IsValid)
        {
            return View(patient);
        }

        PatientRepository.Add(patient);
        return RedirectToAction(nameof(Index));
    }

    [HttpGet]
```

```
        public IActionResult Edit(int id)
        {
            var patient = PatientRepository.GetById(id);
            if (patient == null) return NotFound();
            return View(patient);
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public IActionResult Edit(int id, Patient patient)
        {
            if (id != patient.PatientId) return BadRequest();

            if (!ModelState.IsValid)
            {
                return View(patient);
            }

            PatientRepository.Update(patient);
            return RedirectToAction(nameof(Index));
        }
    }
}
```

Models/Patient.cs

```
using System.ComponentModel.DataAnnotations;
using System.Globalization;

namespace HospitalManagement.Models
{
    public class Patient
    {
        public int PatientId { get; set; }
        public string Name { get; set; }
        public String Disease { get; set; }
    }
}
```

Data/PatientRepository.cs

```
using System.Collections.Generic;
using System.Linq;
using HospitalManagement.Models;

namespace HospitalManagement.Data
{
    public static class PatientRepository
    {
        private static readonly List<Patient> _patients = new();
        private static int _nextId = 1;

        public static IEnumerable<Patient> GetAll() => _patients;

        public static Patient GetById(int id) => _patients.FirstOrDefault(p => p.PatientId == id);

        public static void Add(Patient p)
        {
            p.PatientId = _nextId++;
            _patients.Add(p);
        }

        public static void Update(Patient p)
        {
            var existing = GetById(p.PatientId);
            if (existing == null) return;
            existing.Name = p.Name;
            existing.Disease = p.Disease;
        }
    }
}
```

Views/Patient/Create.cshhtml

```
@model HospitalManagement.Models.Patient
@{
    ViewData["Title"] = "Create Patient";
}

<h2>Create Patient</h2>

<form asp-action="Create" method="post">
    @Html.AntiForgeryToken()
    <div asp-validation-summary="ModelOnly" class="text-danger"></div>

    <div class="mb-2">
        <label asp-for="Name" class="form-label"></label>
        <input asp-for="Name" class="form-control" />
        <span asp-validation-for="Name" class="text-danger"></span>
    </div>

    <div class="mb-2">
        <label asp-for="Disease" class="form-label"></label>
        <input asp-for="Disease" class="form-control" />
        <span asp-validation-for="Disease" class="text-danger"></span>
    </div>
    <button type="submit" class="btn btn-primary">Create</button>
</form>

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

Views/Patient/Details.cshtml

```
@model HospitalManagement.Models.Patient

<h2>Patient Details</h2>

<dl class="row">
    <dt class="col-sm-2">Id</dt>
    <dd class="col-sm-10">@Model.PatientId</dd>

    <dt class="col-sm-2">Name</dt>
    <dd class="col-sm-10">@Model.Name</dd>

    <dt class="col-sm-2">Disease</dt>
    <dd class="col-sm-10">@Model.Disease</dd>
```

```
</dl>


<a asp-action="Index" class="btn btn-link">Back to list</a>
```

View/Patient/Edit.cshtml

```
@model HospitalManagement.Models.Patient
@{
    ViewData["Title"] = "Edit Patient";
}


<h2>Edit Patient</h2>


<form asp-action="Edit" method="post">
    @Html.AntiForgeryToken()
    <input asp-for="PatientId" type="hidden" />


    <div asp-validation-summary="ModelOnly" class="text-danger"></div>


    <div class="mb-2">
        <label asp-for="Name" class="form-label"></label>
        <input asp-for="Name" class="form-control" />
        <span asp-validation-for="Name" class="text-danger"></span>
    </div>
```

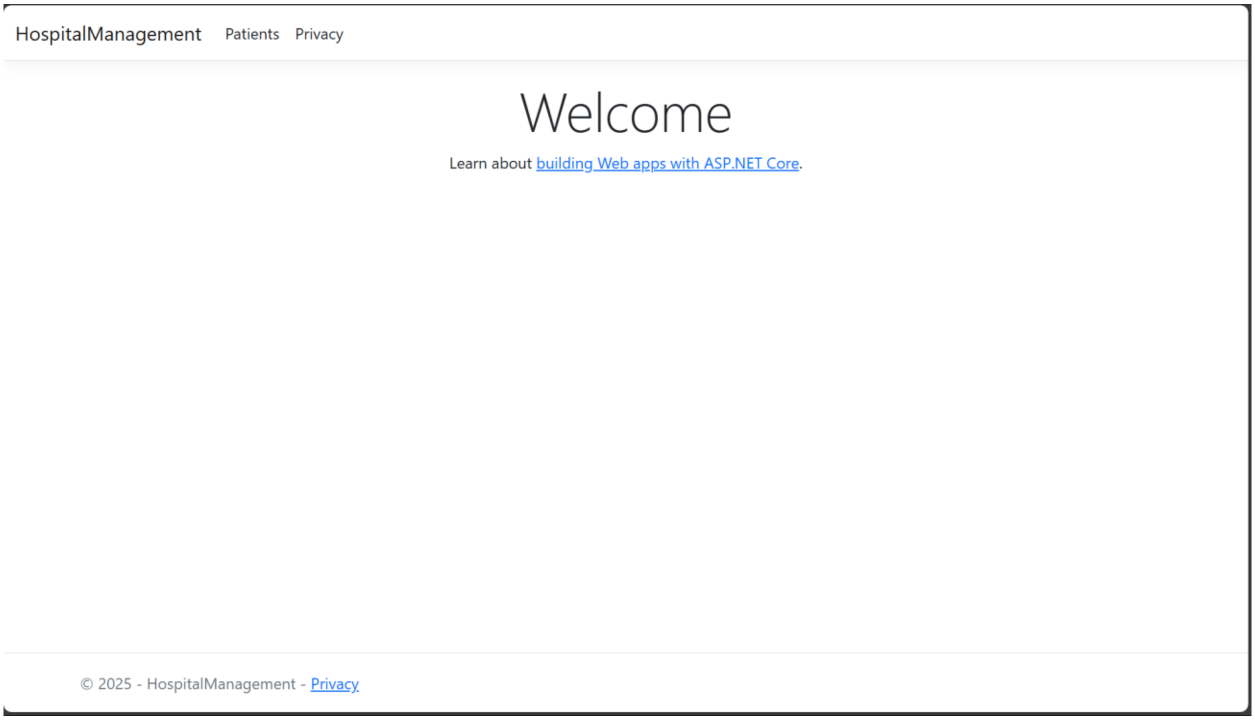
```
  <div class="mb-2">
    <label asp-for="Disease" class="form-label"></label>
    <input asp-for="Disease" class="form-control" />
    <span asp-validation-for="Disease" class="text-danger"></span>
  </div>

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

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

Views/Patient/Inder.cshtml

```
@model IEnumerable<HospitalManagement.Models.Patient>

@{
  ViewData["Title"] = "Patients";
}

<h2>Patients</h2>
```

```
<p>
    <a asp-action="Create" class="btn btn-primary">Create New Patient</a>
</p>

<table class="table table-striped">
    <thead>
        <tr>
            <th>Id</th>
            <th>Patient Name</th>
            <th>Disease</th>
            <th></th>
        </tr>
    </thead>
    <tbody>
        @foreach (var p in Model)
        {
            <tr>
                <td>@p.PatientId</td>
                <td>@p.Name</td>
                <td>@p.Disease</td>
                <td>
                    <a asp-action="Edit" asp-route-id="@p.PatientId">Edit</a> |
                    <a asp-action="Details" asp-route-id="@p.PatientId">Details</a>
                </td>
            </tr>
        }
    </tbody>
</table>
```

OUTPUT:

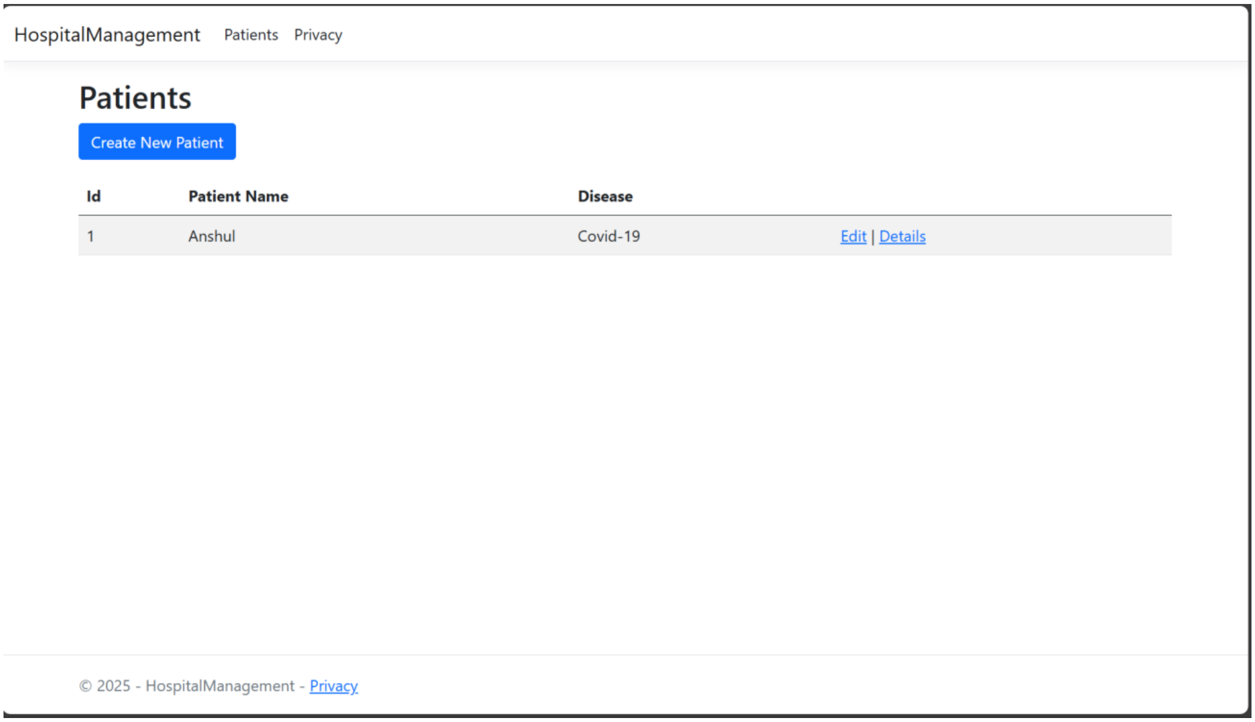


HospitalManagement Patients Privacy

Welcome

Learn about building Web apps with ASP.NET Core.

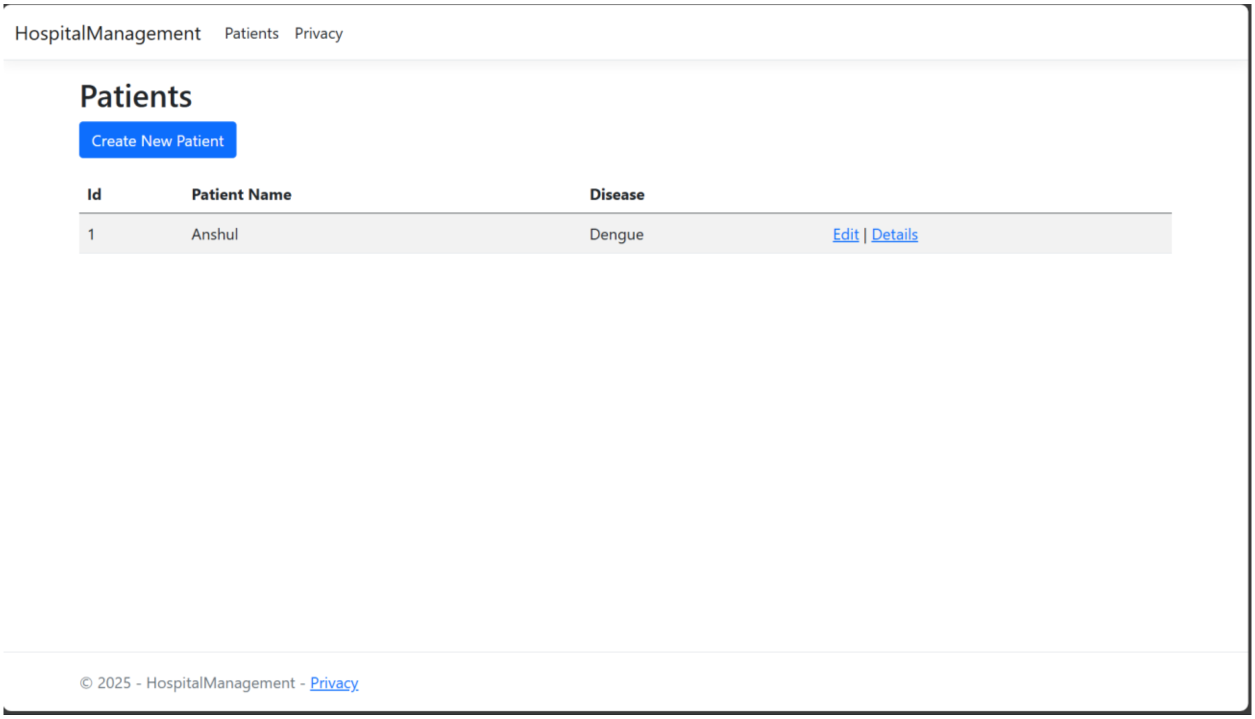
© 2025 - HospitalManagement - Privacy



Patients

Id	Patient Name	Disease	
1	Anshul	Covid-19	Edit | Details

© 2025 - HospitalManagement - Privacy



Patients

Id	Patient Name	Disease	
1	Anshul	Dengue	Edit | Details

© 2025 - HospitalManagement - Privacy