

INTERNAL PRACTICAL EXAM(.NET)

1) 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?**

//PatientController.cs

using HospitalManagementSystem.Models;

using System.Collections.Generic;

using System.Linq;

using System.Web.Mvc;

namespace HospitalManagementSystem.Controllers

```
{  
    public class PatientController : Controller  
    {  
        // Dummy data  
        private static List<Patient> patients = new List<Patient>  
        {  
            new Patient { PatientId = 1, Name = "Aesha Kalathiya", Disease = "Fever" },  
            new Patient { PatientId = 2, Name = "Yatri Dungarani", Disease = "Diabetes" },  
            new Patient { PatientId = 3, Name = "Khushi Dadhaniya", Disease = "Asthma" }  
        };  
  
        // GET: Patient  
        public ActionResult Index()  
        {  
            return View(patients);  
        }  
  
        // GET: Patient/Create  
        public ActionResult Create()  
        {  
            return View();  
        }  
  
        // POST: Patient/Create  
        [HttpPost]  
        [ValidateAntiForgeryToken]  
        public ActionResult Create(Patient patient)
```

```
{
    if (ModelState.IsValid)
    {
        patient.PatientId = patients.Max(p => p.PatientId) + 1;
        patients.Add(patient);
        return RedirectToAction("Index");
    }
    return View(patient);
}

// GET: Patient/Edit/5
public ActionResult Edit(int id)
{
    var patient = patients.FirstOrDefault(p => p.PatientId == id);
    if (patient == null)
        return HttpNotFound();
    return View(patient);
}

// POST: Patient/Edit/5
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit(Patient patient)
{
    if (ModelState.IsValid)
    {
        var existing = patients.FirstOrDefault(p => p.PatientId == patient.PatientId);
```

```
        if (existing != null)
        {
            existing.Name = patient.Name;
            existing.Disease = patient.Disease;
        }
        return RedirectToAction("Index");
    }
    return View(patient);
}
}
```

//Patient.cs

```
using System.ComponentModel.DataAnnotations;

namespace HospitalManagementSystem.Models
{
    public class Patient
    {
        [Key]
        public int PatientId { get; set; }

        [Required(ErrorMessage = "Patient name is required")]
        [StringLength(50)]
        public string Name { get; set; }

        [Required(ErrorMessage = "Disease field is required")]
```

```
[StringLength(100)]
public string Disease { get; set; }
}
}
```

//Create.cshtml

```
@model HospitalManagementSystem.Models.Patient
```

```
@{
```

```
    ViewBag.Title = "Add Patient";
```

```
}
```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet" />
```

```
<div class="container mt-5">
```

```
    <h2 class="mb-4 text-success">Add New Patient</h2>
```

```
@using (Html.BeginForm())
```

```
{
```

```
    @Html.AntiForgeryToken()
```

```
    <div class="mb-3">
```

```
        @Html.LabelFor(m => m.Name, new { @class = "form-label" })
```

```
        @Html.TextBoxFor(m => m.Name, new { @class = "form-control" })
```

```
        @Html.ValidationMessageFor(m => m.Name, "", new { @class = "text-danger" })
```

```
    </div>
```

```
    <div class="mb-3">
```

```
        @Html.LabelFor(m => m.Disease, new { @class = "form-label" })
```

```
        @Html.TextBoxFor(m => m.Disease, new { @class = "form-control" })
```

```
        @Html.ValidationMessageFor(m => m.Disease, "", new { @class = "text-danger" })
    </div>

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

    <a href="@Url.Action("Index")" class="btn btn-secondary">Cancel</a>

}
</div>
```

```
@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

//Index.cshtml

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

@{
    ViewBag.Title = "Patients";
}

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet" />
```

```
<div class="container mt-5">

    <h2 class="mb-4 text-primary">Patient List</h2>

    <a href="@Url.Action("Create")" class="btn btn-success mb-3">Add New Patient</a>

    <table class="table table-bordered table-striped table-hover">

        <thead class="table-dark">

            <tr>

                <th>ID</th>

                <th>Name</th>

                <th>Disease</th>
```

```

        <th>Actions</th>
    </tr>
</thead>
<tbody>
    @foreach (var patient in Model)
    {
        <tr>
            <td>@patient.PatientId</td>
            <td>@patient.Name</td>
            <td>@patient.Disease</td>
            <td>
                <a href="@Url.Action("Edit", new { id = patient.PatientId })" class="btn btn-
primary btn-sm">Edit</a>
            </td>
        </tr>
    }
</tbody>
</table>
</div>

```

//Edit.cshtml

```

@model HospitalManagementSystem.Models.Patient
@{
    ViewBag.Title = "Edit Patient";
}
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet" />

```

```
<div class="container mt-5">
```

```
    <h2 class="mb-4 text-primary">Edit Patient Details</h2>
```

```
    @using (Html.BeginForm())
```

```
    {
```

```
        @Html.AntiForgeryToken()
```

```
        @Html.HiddenFor(m => m.PatientId)
```

```
        <div class="mb-3">
```

```
            @Html.LabelFor(m => m.Name, new { @class = "form-label" })
```

```
            @Html.TextBoxFor(m => m.Name, new { @class = "form-control" })
```

```
            @Html.ValidationMessageFor(m => m.Name, "", new { @class = "text-danger" })
```

```
        </div>
```

```
        <div class="mb-3">
```

```
            @Html.LabelFor(m => m.Disease, new { @class = "form-label" })
```

```
            @Html.TextBoxFor(m => m.Disease, new { @class = "form-control" })
```

```
            @Html.ValidationMessageFor(m => m.Disease, "", new { @class = "text-danger" })
```

```
        </div>
```

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

```
        <a href="@Url.Action("Index")" class="btn btn-secondary">Cancel</a>
```

```
    }
```

```
</div>
```

```
@section Scripts {
```

```
    @Scripts.Render("~/bundles/jqueryval")
```

```
}
```


Output:

Hospital Management

Patient List

Add New Patient

ID	Name	Disease	Actions
1	Aesha Kalathiya	Fever	Edit
2	Yatri Dungarani	Diabetes	Edit
3	Khushi Dadhaniya	Asthma	Edit