

Models.cs

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

namespace CovidHospitalsMgmt.Models
{
    public enum City
    {
        Select,
        Chennai,
        Delhi,
        Kolkata,
        Mumbai
    }

    public class CovidHospital
    {
        [Key] public int HospitalID { get; set; }

        [Required(ErrorMessage = "Please Provide Hospital Name")]
        [MaxLength(255)]
        [Display(Name = "HospitalName")]
        public string HospitalName { get; set; }

        [Required(ErrorMessage = "Please Select a City")]
        [DisplayName("City")]
        [Range(1, int.MaxValue, ErrorMessage = "Please Select a City")]
        public City City { get; set; }

        [DisplayName("Open for Online Registrations?")]
        public bool OnlineRegistrations { get; set; }

        [Required(ErrorMessage = "Should be between 100 and 900")]
        [DisplayName("No Of Beds")]
        [Range(100, 900, ErrorMessage = "Should be between 100 and 900")]
        public int Beds { get; set; }

        [Required(ErrorMessage = "Please Provide Valid Date")]
        [DataType(DataType.Date)]
        [DisplayFormat(DataFormatString = "{0:dd-MM-yy}",
            ApplyFormatInEditMode = true)]
        public Nullable<DateTime> StartDate { get; set; }
    }
}
```

Covidhospitaldbcontext.cs

```
using CovidHospitalsMgmt.Models;
using System.Data.Entity;

namespace CovidHospitalsMgmt.Data
{
    public class CovidDbContext : DbContext
    {
        public CovidDbContext()
            : base("name=CHM_DB")
        {
        }

        public DbSet<CovidHospital> CovidHospitals { get; set; }
    }
}
```

CustomException

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Routing;

namespace CovidHospitalsMgmt.CustomFilters
{
    public class CustomExceptionHandlerAttribute : FilterAttribute,
        IExceptionHandler
    {
        public void OnException(ExceptionContext filterContext)
        {
            filterContext.HttpContext.Response.Write("oops error!");

            ConsoleLog(filterContext.Exception.Message);
            filterContext.ExceptionHandled = true;
        }

        static string scriptTag = "<script
type=\"\"language=\"\">{0}</script>";

        public static void ConsoleLog(string message)
        {
            string function = "console.log('{0}')";
            string log =
                string.Format(GenerateCodeFromFunction(function), message);
            HttpContext.Current.Response.Write(log);
        }

        static string GenerateCodeFromFunction(string function)
        {
            return string.Format(scriptTag, function);
        }
    }
}
```

```
}  
}  
}
```

CovidHospitalsController.cs

```
using CovidHospitalsMgmt.CustomFilters;  
using CovidHospitalsMgmt.Models;  
using CovidHospitalsMgmt.Data;  
using System;  
using System.Web.Mvc;  
using System.Linq;  
  
namespace CovidHospitalsMgmt.Controllers  
{  
    [CustomExceptionHandlerAttribute]  
    public class CovidHospitalsController : Controller  
    {  
        private CovidHospitalDbContext _covidDbContext = new  
CovidHospitalDbContext ();  
  
        // GET  
        public ActionResult Index()  
        {  
            var list = (from s in _covidDbContext.CovidHospitals  
                        orderby s.City  
                        select s).ToList();  
            return View(list);  
        }  
  
        [HttpGet]  
        public ActionResult AddCovidHospital()  
        {  
            CovidHospital ch = new CovidHospital();  
            ch.Beds = 0;  
            return View(ch);  
        }  
  
        [HttpPost]  
        public ActionResult AddCovidHospital(CovidHospital covidHospital)  
        {  
            var ctx = new CovidHospitalDbContext();  
            try  
            {  
                if (ModelState.IsValid)  
                {  
                    ctx.CovidHospitals.Add(covidHospital);  
                    ctx.SaveChanges();  
                    return RedirectToAction("Details", new  
{
```

```

        id = covidHospital.HospitalID
    });
    }
    else
    {
        return View(covidHospital);
    }
}
catch (Exception)
{
    return View("error");
}
}

public ActionResult Details(int id)
{
    var ctx = new CovidHospitalDBContext ();
    CovidHospital c = null;
    if (id != null)
    {
        c = ctx.CovidHospitals.FirstOrDefault(model =>
model.HospitalID == id);
        return View(c);
    }

    return View();
}
}
}

```

_Layout.cshtml

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - Covid Hospitals Management</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            @Html.ActionLink("Covid Hospitals Management", "Index",
"CovidHospitals", new {area = ""}, new {@class = "navbar-brand"})
        </div>
    </div>
</div>

```

```

        <div class="navbar-collapse collapse">
            <ul class="nav navbar-nav">
                <li>@Html.ActionLink("Hospitals List", "Index",
"CovidHospitals")</li>
                <li>@Html.ActionLink("Create new", "AddCovidHospitals",
"CovidHospitals")</li>
            </ul>
        </div>
    </div>
</div>
<div class="container body-content">
    @RenderBody()
    <hr/>
    <footer>
        <p>&copy; @DateTime.Now.2022 - Covid Hospitals Management</p>
    </footer>
</div>

@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/bootstrap")
@RenderSection("scripts", required: false)
</body>
</html>

```

Index.cshtml

```

@model List<CovidHospitalsMgmt.Models.CovidHospital>

@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "AddCovidHospitals", "CovidHospitals")
</p>
<h1>Covid Hospitals List</h1>
<table>
    <tr>
        <th>Hospital Name</th>
        <th>City</th>
        <th>Start Date</th>
        <th>Open for Online Registrations?</th>
        <th>No of Beds</th>
    </tr>
    @foreach (var item in Model)
    {
        <tr>
            <td>@Html.DisplayFor(modelItem => item.HospitalName)</td>
            <td>@Html.DisplayFor(modelitem => item.City)</td>
            <td>@Html.DisplayFor(modelitem => item.StartDate)</td>
            <td>@Html.DisplayFor(modelitem => item.OnlineRegistrations)</td>
            <td>@Html.DisplayFor(modelitem => item.Beds)</td>
        </tr>
    }

```

```
        </tr>
    }
</table>
```

Details.cshtml

```
@model CovidHospitalsMgmt.Models.CovidHospital

@{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2 id="message">Covid Hospital added successfully!</h2>

<div>

    <hr/>
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.HospitalName)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.HospitalName)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.City)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.City)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Beds)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Beds)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.StartDate)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.StartDate)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.OnlineRegistrations)
        </dt>
```

```

        </dt>

        <dd>
            @Html.DisplayFor(model => model.OnlineRegistrations)
        </dd>

    </dl>
</div>
<p>
    @Html.ActionLink("Back to List", "Index", "CovidHospitals")
</p>

```

AddCovidHospitals.cshtml

```

@model CovidHospitalsMgmt.Models.CovidHospital

@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Create</h2>

@using (Html.BeginForm())
{
    <div class="form-horizontal">
        <h4>Covid Hospital</h4>
        <hr/>
        @Html.ValidationSummary(true, "", new
        {
            @class = "text-danger"
        })
        <div class="form-group">
            @Html.LabelFor(model => model.HospitalName,
                htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model =>
                    model.HospitalName, new
                    {
                        htmlAttributes = new { @class = "form-control" }
                    })
                @Html.ValidationMessageFor(model =>
                    model.HospitalName, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.City,
                htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EnumDropDownListFor(model =>
                    model.City, "None",
                    new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model =>

```

```

        model.City, "", new {@class = "text-danger"})
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Beds,
        htmlAttributes: new {@class = "control-label col-md-2"})
    <div class="col-md-10">
        @Html.EditorFor(model => model.Beds, new
        {
            htmlAttributes = new {@class = "form-control"}
        })
        @Html.ValidationMessageFor(model =>
            model.Beds, "", new {@class = "text-danger"})
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(model => model.StartDate,
        htmlAttributes: new {@class = "control-label col-md-2"})
    <div class="col-md-10">
        @Html.EditorFor(model => model.StartDate,
            new {htmlAttributes = new {@class = "form-control"}})
        @Html.ValidationMessageFor(model =>
            model.StartDate, "",
            new {@class = "text-danger"})
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(model =>
        model.OnlineRegistrations, htmlAttributes:
            new {@class = "control-label col-md-2"})
    <div class="col-md-10">
        <div class="checkbox">
            @Html.CheckBoxFor(model => model.OnlineRegistrations)
            @Html.ValidationMessageFor(model =>
                model.OnlineRegistrations, "", new {@class = "text-danger"})
        </div>
    </div>
</div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Add Covid Hospital" id="Submit"
class="btn btn-default"/>
        </div>
    </div>
</div>
}

```