

Restaurant.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;
//using System;
//using System.Collections.Generic;
//using System.Linq;
//using System.Web;

namespace RestaurantManagement.Models
{
    public class Restaurant
    {
        [Key]
        public int Id { get; set; }

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

        [Required(ErrorMessage = "Please Select Cuisine Type")]
        [Display(Name = "Type of food")]

        public CuisineType? Cuisine { get; set; }

        [Required(ErrorMessage = "Please Provide Online Orders")]
        [Display(Name = "Open for Online Orders?")]
        public bool OnlineOrders { get; set; }

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

        //public enum CuisineType
        //{
        //    Italian,
        //    Indian,
        //    French,
        //    Mexican,
        //    Chinese,
        //    Japanese,
        //    Iranian,
        //    Spanish,
        //    Greece,
        //    American,
        //    Arabian
        //}
    }
}
```

CuisineType.cs

(nisar)



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace RestaurantManagement.Models
{
    public enum CuisineType
    {
        Italian,
        Indian,
        French,
        Mexican,
        Chinese,
        Japanese,
        Iranian,
        Spanish,
        Greece,
        American,
        Arabian
    }
}

```

CustomExceptionFilterAttribute.cs

```

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

namespace RestaurantManagement.Models
{
    public class CustomExceptionFilterAttribute : FilterAttribute, IExceptionFilter

    {
        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);
        }
    }
}

```



```

        public void OnException(ExceptionContext filterContext)
        {
            ConsoleLog(filterContext.Exception.Message);
        }
    }
}

```

RestaurantManagemntDbContext.cs

```

//using RestaurantManagement.Models;
//using System;
//using System.ComponentModel.DataAnnotations.Schema;
//using System.Data.Entity;
//using System.Linq;
using RestaurantManagement.Models;
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace RestaurantManagement
{
    public class RestaurantManagementDbContext : DbContext
    {
        public RestaurantManagementDbContext() : base("RMS_DB")
        {
        }
        public DbSet<Restaurant> Restaurants { get; set; }
        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
        }
    }
}

```

RestaurantController

```

//using RestaurantManagement.Models;
//using System;
//using System.Collections.Generic;
//using System.Linq;
//using System.Web;
//using System.Web.Mvc;
using System;
using System.Linq;
using System.Web.Mvc;
using RestaurantManagement.Models;

namespace RestaurantManagement.Controllers
{
    [CustomExceptionFilterAttribute]
    public class RestaurantController : Controller
    {
        RestaurantManagementDbContext context = new
RestaurantManagementDbContext();
        [HttpGet]

```



@*<h2>AddRestaurant</h2>*@

```
@using (Html.BeginForm("AddRestaurant", "Restaurant", FormMethod.Post))
{
    @Html.AntiForgeryToken()
    <h2>Create</h2>
    <div class="form-horizontal">
        <h4>Restaurant</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Name, new { htmlAttributes = new
                { @class = "form-control", @id = "Name" } })
                @Html.ValidationMessageFor(model => model.Name, "", new { @class =
                "text-danger" })
            </div>
        </div>

        @*<div class="form-group">
            @Html.LabelFor(model => model.Cuisine, htmlAttributes: new { @class =
            "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EnumDropDownListFor(model => model.Cuisine, "None", new
                { htmlAttributes = new { @class = "form-control", @id = "Type of Food" } })
                @Html.ValidationMessageFor(model => model.Cuisine, "", new { @class =
                "text-danger" })
            </div>
        </div>*@
        <div class="form-group">
            @Html.LabelFor(model => model.Cuisine, htmlAttributes: new { @class =
            "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EnumDropDownListFor(model => model.Cuisine, "None",
                htmlAttributes: new { @class = "form-control", @id = "Type of Food" })
                @Html.ValidationMessageFor(model => model.Cuisine, "", new { @class =
                "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.LaunchDate, htmlAttributes: new { @class =
            "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.LaunchDate, "dd/mm/yyyy", new
                { htmlAttributes = new { @class = "form-control", @id = "LaunchDate" } })
                @Html.ValidationMessageFor(model => model.LaunchDate, "", new
                { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.OnlineOrders, htmlAttributes: new { @class =
```



```

"control-label col-md-2" })
    <div class="col-md-10">
        <div class="checkbox">
            @Html.EditorFor(model => model.OnlineOrders)
            @Html.ValidationMessageFor(model => model.OnlineOrders, "", 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 Restaurant" id="Submit" class="btn btn-
default" />
        </div>
    </div>
</div>
}

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

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

```

Details.cshtml

@model RestaurantManagement.Models.Restaurant

```

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

```

```

<h4>Restaurant Added Successfully!</h4>
<hr />
<div>
    @*<h4>Restaurant</h4>
    <hr />*@
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Name)
        </dt>

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

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

        <dd>
            @Html.DisplayFor(model => model.Cuisine)

```



```

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

        <dd>
            @Html.DisplayFor(model => model.LaunchDate)
        </dd>
        <dt>
            @Html.DisplayNameFor(model => model.OnlineOrders)
        </dt>

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

    </dl>
</div>
<p>
    @*@Html.ActionLink("Edit", "Edit", new { id = Model.Id }) |*@
    @Html.ActionLink("Back to List", "Index")
</p>

```

Index.cshtml

```

@model IEnumerable<RestaurantManagement.Models.Restaurant>

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

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "AddRestaurant")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Name)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Cuisine)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.LaunchDate)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.OnlineOrders)
        </th>
        <th></th>
    </tr>

    @foreach (var item in Model) {
    <tr>
        <td>

```



```

        @Html.DisplayFor(modelItem => item.Name)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Cuisine)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.LaunchDate)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.OnlineOrders)
    </td>
    @*<td>
        @Html.ActionLink("Edit", "Edit", new { id=item.Id }) |
        @Html.ActionLink("Details", "Details", new { id=item.Id }) |
        @Html.ActionLink("Delete", "Delete", new { id=item.Id })
    </td>*@
</tr>
}
</table>

```

_Layout file

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - Restaurant Management System</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("Restaurant Management System", "Index", "Restaurant",
new { area = "" }, new { @class = "navbar-brand" })
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">
                    <li>@Html.ActionLink("RestaurantList", "Index", "Restaurant")</li>
                    @*<li>@Html.ActionLink("About", "About", "Home")</li>
                    <li>@Html.ActionLink("Contact", "Contact", "Home")</li>*@
                </ul>
            </div>
        </div>
    </div>
    <div class="container body-content">
        @RenderBody()
    </div>

```




```
<footer>
  <p>&copy; @DateTime.Now.Year - Restaurant Management Application</p>
</footer>
</div>

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

_ViewStart.cshtml

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

