Practical 6

Q1. Create an MVC application to demonstrate ViewBag Object.

RouteConfig.cs:

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

namespace MVCFirstApp
{
   public class RouteConfig
   {
      public static void RegisterRoutes(RouteCollection routes)
      {
         routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

         routes.MapRoute(
            name: "Default",
            url: "{controller}/{action}/{id}",
            defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
         );
      }
   }
}
```

HomeController.cs:

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

namespace MVCFirstApp
{
   public class RouteConfig
   {
      public static void RegisterRoutes(RouteCollection routes)
      {
         routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

         routes.MapRoute(
            name: "Default",
            url: "{controller}/{action}/{id}",
            defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
         );
      }
   }
}
```

Index.cshtml:

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
</head>
<body>
    <div>
        <h3>@ViewBag.Greeting  Welcome to the World of MVC!!! (From View Page )</h3>
    </div>
</body>
</html>
```

Output:

Q2. Create an MVC application to accept Customer details and display the same using views. Use automatically implemented properties, strongly typed HTML Input helpers methods.

RouteConfig.cs:

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

namespace CustomerMVC
{
   public class RouteConfig
   {
      public static void RegisterRoutes(RouteCollection routes)
      {
         routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

         routes.MapRoute(
            name: "Default",
            url: "{controller}/{action}/{id}",
            defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
         );
      }
   }
}
```

Customer.cs:

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

namespace CustomerMVC.Models
{
   public class Customer
   {
      [Required(ErrorMessage = "It's Madantory! ID")]
      public int CustId { get; set; }
      [Required(ErrorMessage = "It's Madantory! Name")]
      public string CustName { get; set; }
      [Required(ErrorMessage = "It's Madantory! Address")]
      public string CustAdd { get; set; }
   }
}
```

HomeController:

```
using CustomerMVC.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
```

```
namespace CustomerMVC.Controllers
{
    public class HomeController : Controller
    {
        // GET: Home
        public ActionResult Index()
        {
            return View();
        }

        [HttpGet]
        public ActionResult CustomerInput()
        {
            return View();
        }

        [HttpPost]
        public ViewResult CustomerInput(Customer c1)
        {
            if(ModelState.IsValid)
            {
                return View("CustomerDisplay",c1);
            }
            else
            {
                return View();
            }
```

```
        }
    }
}
```

Index.cshtml:

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
    <link href="~/Content/bootstrap.css" rel="stylesheet" />
    <link href="~/Content/bootstrap-theme.css" rel="stylesheet" />
    <style>
        .btn a {
            color: white;
            text-decoration: none
        }

        body {
            background-color: #F1F1F1;
        }
    </style>
```

```
</head>
<body>
    <div>
        <div class="text-center">
            <p><h1>Customer Information System </h1><br /> </p>
            <div class="btn btn-success">
                @Html.ActionLink("Add Customer Details", "CustomerInput")
            </div>
        </div>
    </div>
</body>
</html>
```

CutomerInput.cshtml:

```
@model CustomerMVC.Models.Customer

@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>CustomerInput</title>
    <style>
        .field-validation-error {
```

```
            color: #f00;
        }

        .field-validation-valid {
            display: none;
        }

        .input-validation-error {
            border: 1px solid #f00;
            background-color: #fee;
        }

        .validation-summary-errors {
            font-weight: bold;
            color: #f00;
        }

        .validation-summary-valid {
            display: none;
        }

    </style>
    <link href="~/Content/bootstrap.css" rel="stylesheet" />
    <link href="~/Content/bootstrap-theme.css" rel="stylesheet" />
</head>
<body>
    <div class="panel panel-success">
        <div class="panel-heading text-center">
```

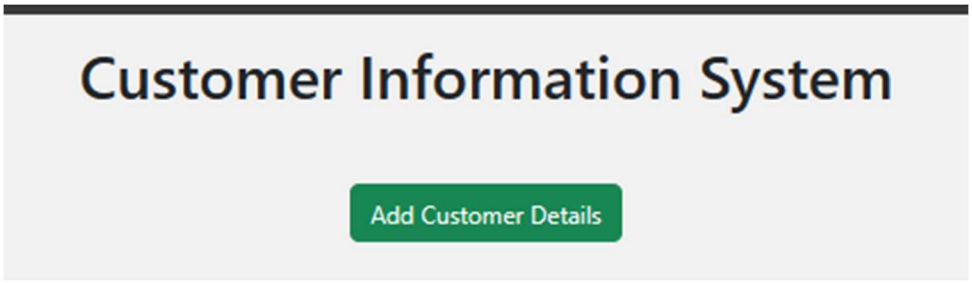
```
            <h4>
                Customer Information
                System
            </h4>
        </div>

        @using (Html.BeginForm())
        {
            @Html.ValidationSummary()
            <p>Customer ID:      @Html.TextBoxFor(x => x.CustId) </p> <br />
            <p>Customer Name:    @Html.TextAreaFor(x => x.CustName)</p><br />
            <p>Customer Address: @Html.TextAreaFor(x => x.CustAdd)</p><br />
            <div class="btn btn-sucess">
                <input type="submit" value="SubmitData" />
            </div>
        }
    </div>
</body>
</html>
```

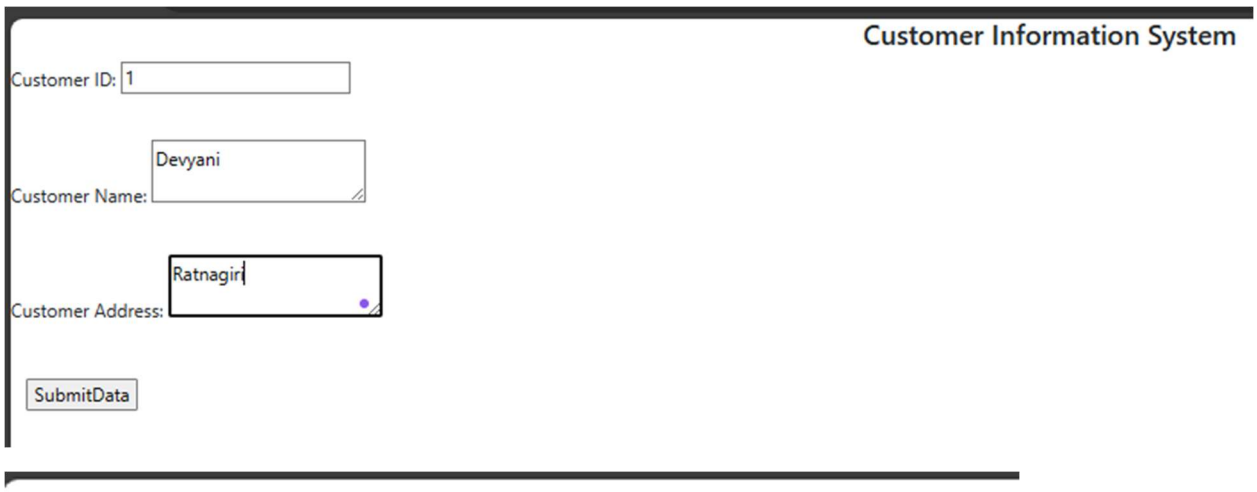
CustomerDisplay.cshtml:

```
@model CustomerMVC.Models.Customer

@{
    Layout = null;
}

<!DOCTYPE html>
```

```
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>CustomerDisplay</title>
</head>
<body>
    <div>
        <h3>Customer Information System</h3>
        <p>Customer Id:      @Model.CustId</p> <br />
        <p>Customer Name:    @Model.CustName</p> <br />
        <p>Customer Address: @Model.CustAdd</p> <br />

    </div>
</body>
</html>
```

Output:





Customer Information System

Customer Id: 1

Customer Name: Devyani

Customer Address: Ratnagiri

Q3. Create an MVC application to demonstrate extension methods and use layout.

RoteConfig.cs:

```
using System.Web.Mvc;
using System.Web.Routing;

namespace MVCApp3
{
   public class RouteConfig
   {
      public static void RegisterRoutes(RouteCollection routes)
      {
         routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

         routes.MapRoute(
            name: "Default",
            url: "{controller}/{action}/{id}",
            defaults: new { controller = "Home", action = "UseExtension", id =
UrlParameter.Optional }
         );
      }
   }
}
```

Product.cs:

```
namespace MVCApp3.Models
{
   public class Product
```

```
    {
        public int ProductID { get; set; }
        public string Name { get; set; }
        public decimal Price { get; set; }
    }
}
```

ShoppingCart.cs:

```
using System.Collections.Generic;

namespace MVCApp3.Models
{
    public class ShoppingCart
    {
        public List<Product> Products { get; set; }
    }
}
```

MyExtensionMethod.cs:

```
namespace MVCApp3.Models
{
    public static class MyExtensionMethods
    {
        public static decimal TotalPrices(this ShoppingCart cartParam)
        {
            decimal total = 0;
            foreach (Product prod in cartParam.Products)
            {
```

```
            total += prod.Price;
        }
        return total;
    }
  }
}
```

HomeController.cs:

```
using MVCApp3.Models;
using System;
using System.Collections.Generic;
using System.Web.Mvc;
namespace MVCApp3.Controllers
{
  public class HomeController : Controller
  {
    // GET: Home
    public ActionResult Index()
    {
      return View();
    }
    public ViewResult UseExtension()
    {
      ShoppingCart cart = new ShoppingCart
      {
        Products = new List<Product>
        {
          new Product {ProductID=1, Name="Kayak", Price=275M},
```

```
                new Product {ProductID=2, Name="LifeJacket", Price=48.95M},
                new Product {ProductID=3, Name="Soccer Ball", Price=19.50M},
                new Product {ProductID=4, Name="Corner Flag", Price=34.95M},
            }
        };
        decimal cartTotal = cart.TotalPrices();
        return View("Result", (object)String.Format("Total: {0:c}", cartTotal));
    }
  }
}
```

_LayoutPage1.cshtml:

```
<!DOCTYPE html>

<html>
<head>
   <meta name="viewport" content="width=device-width" />
   <title>@ViewBag.Title</title>
</head>
<body>

      <h1>Product Information</h1>
      <div style="padding: 20px; border: solid medium black; font-size: 20pt">
         @RenderBody()
      </div>
      <h2>Visit <a href="http://flipkart.com">flipkart</a></h2>

</body>
```

</html>

Output: