FINOLEX ACADEMY OF MANAGEMENT AND TECHNOLOGY, RATNAGIRI

DEPARTMENT OF MCA

PRACTICAL NO. 06

ASP.NET MVC

1. **Create an MVC application to demonstrate ViewBag Object.**

   Ans :

   - **Code :**

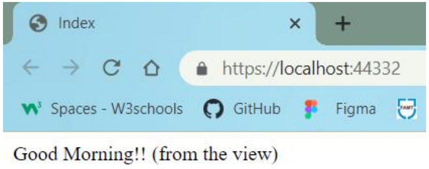
     HomeController.cs

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

namespace MyFirstMVCApp.Controllers
{
   public class HomeController : Controller
   {
      // GET: Home
      public ViewResult Index()
      {
         int hour = DateTime.Now.Hour;
         ViewBag.Greeting = hour < 12 ? "Good Morning!!" : "Good Afternoon!!";
         return View();
      }}}
```

Index.cshtml

```
@{
   Layout = null;
}

<!DOCTYPE html>

<html>
<head>
   <meta name="viewport" content="width=device-width" />
   <title>Index</title>
</head>
<body>
   <div>
     @ViewBag.Greeting (from the view)
   </div>
</body>
                  </html>
```

- **Output :**



Good Morning!! (from the view)

2. **Create an MVC application to accept Customer details and display the same using views. Use automatically implemented properties, strongly typed HTML Input helpers methods.**
   Ans :
   - **Code :**
     HomeController.cs

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

namespace CustomerApplication.Controllers
{
    public class HomeController : Controller
    {
        // GET: Home
        public ActionResult Index()
        {
            return View();
        }
        [HttpGet]
        public ViewResult CustomerInput()
        {
            return View();
        }

        [HttpPost]
        public ViewResult CustomerInput(Customer c1)
        {
            if (ModelState.IsValid)
            {
                return View("CustomerDisplay", c1);
            }
            else
            { //There is validation error
                return View();
            }

        }
    }
}
```

Indesx.cshtml

```
@{
    Layout = null;
}

<!DOCTYPE html>
```

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

        body {
            background-color: #F1F1F1;
        }
    </style>
</head>
<body>
    <div class="text-center">
        <p> Customer information system allows to store retrive   customer details </p>
        <div class="btn btn-success">
            @Html.ActionLink("Add Customer Details", "CustomerInput")
        </div>
</body>
                          </html>
```

CustomerInput.cshtml

```
@model CustomerApplication.Models.Customer

@{
    Layout = null;
}


<!DOCTYPE html>
<html>
<head>
    <link href="~/Content/bootstrap.css" rel="stylesheet" />
    <link href="~/Content/bootstrap-theme.css" rel="stylesheet" />
    <meta name="viewport" content="width=device-width" />
    <link rel="stylesheet" type="text/css" href="~/Content/StyleSheet1.css" />
    <title>CustomerInput</title>
</head>
<body>
    <div class="panel panel-success">
        <div class="panel-heading text-center">
            <h4>
                Customer Information
                System
            </h4>
        </div>
        <div class="panel-body">
            @using (Html.BeginForm())
            {
```

```
            @Html.ValidationSummary()
             <div class="form-group">
                <label>Customer ID :</label>@Html.TextBoxFor(x => x.CustID, new{ @class = "form-control" })
             </div>
                         <div class="form-group">
                            <label>Customer Name :</label>@Html.TextBoxFor(x =>x.CustName, new { @class = "form-control" })
                         </div><div class="form-group">
                            <label>Customer Address:</label>@Html.TextBoxFor(x =>x.CustAdd, new { @class = "form-control" })

                         </div>
                           <div class="text-center">
                             <input class="btn btn-success" type="submit" value="Submit Customer Data" />
                                     </div>         }
      </div>
   </div>
</body>
                    </html>
```

CustomerDisplay.cshtml

```
@model CustomerApplication.Models.Customer

@{
   Layout = null;
}

<!DOCTYPE html>

<html>
<head>
   <link href="~/Content/bootstrap.css" rel="stylesheet" />
   <link href="~/Content/bootstrap-theme.css" rel="stylesheet" />
   <meta name="viewport" content="width=device-width" />
   <title>CustomerDisplay</title>
   <style>
      body {
         background-color: #F1F1F1;
      }
   </style>
</head>
<body>
   <div class="text-center">
      <h1>Customer Information System</h1>
      <p>Customer ID : @Model.CustID</p>
      <p>Customer Name : @Model.CustName</p>
      <p>Customer Address : @Model.CustAdd</p>
   </div>
</body>
</html>
```

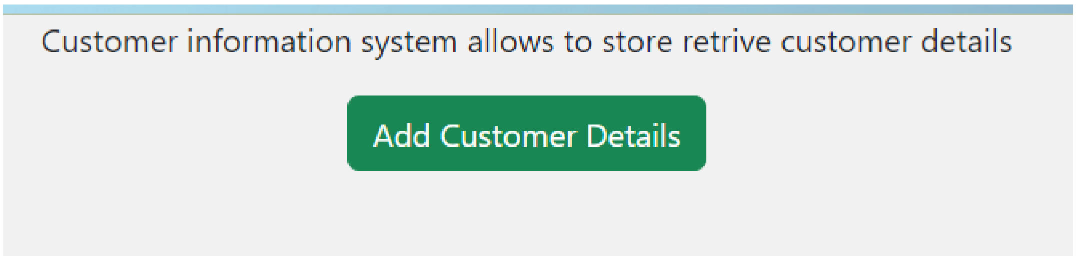
Customer.cs

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

namespace CustomerApplication.Models
{
    public class Customer
    {
        [Required(ErrorMessage = "Please enter Customer ID!")]
        public int CustID { get; set; }
        [Required(ErrorMessage = "Please enter Customer Name!")]
        public string CustName { get; set; }
        [Required(ErrorMessage = "Please enter Customer Address!")]
        public string CustAdd { get; set; }
    }
                    }
```

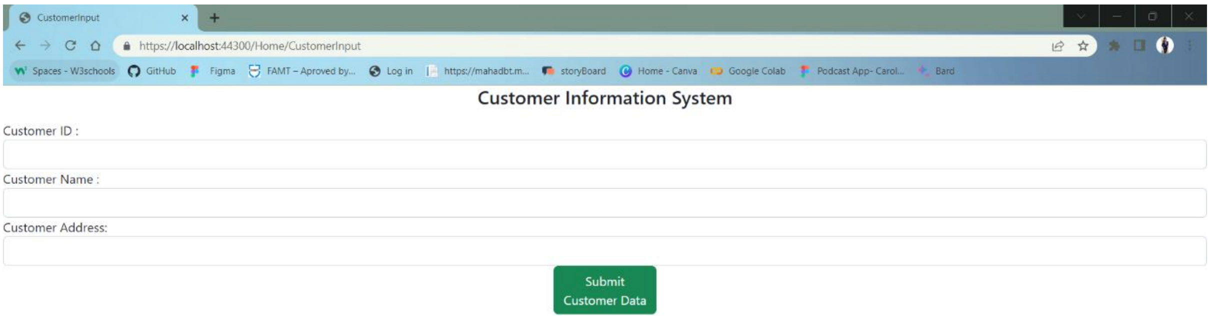
StyleSheet1.css

```
body {
    .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;
}
```

- **Output :**



Customer information system allows to store retrive customer details

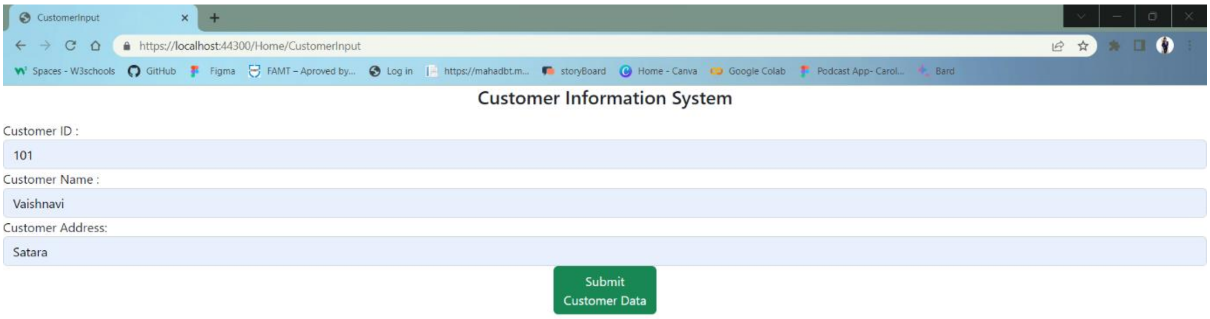


Customer Information System

Customer ID :

Customer Name :

Customer Address:

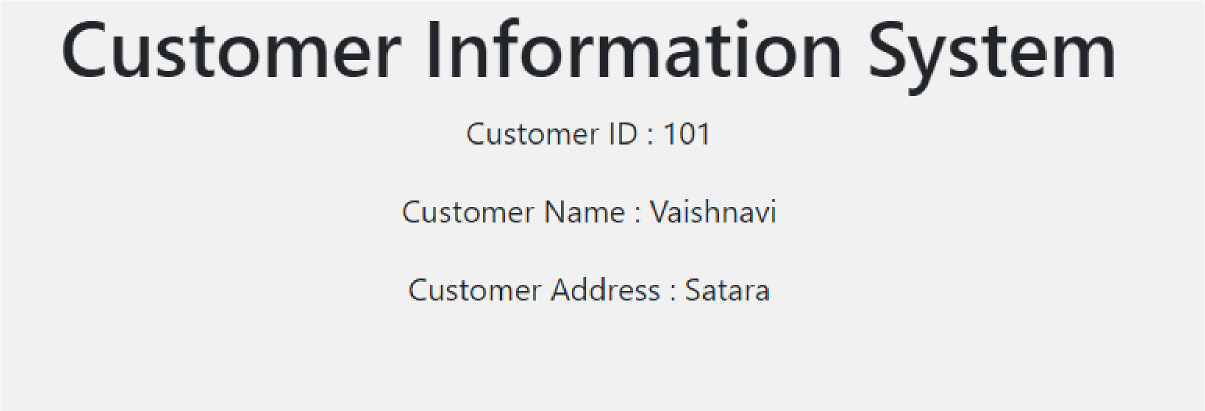


Customer Information System

Customer ID : 101

Customer Name : Vaishnavi

Customer Address: Satara



Customer Information System

Customer ID : 101

Customer Name : Vaishnavi

Customer Address : Satara

3. **Create an MVC application to demonstrate extension methods and use layout.**
   Ans :
   - **Code :**

     HomeController.cs

```
using ProductMVCApplication.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Web;
using System.Web.Mvc;

namespace ProductMVCApplication.Controllers
{
  public class HomeController : Controller
  {
    // GET: Home

    public ViewResult AutoProperty()
    {
      Product p1 = new Product();
      p1.Name = "Kayak";
      string pName = p1.Name;
      return View("AutoProperty", (object)String.Format("Product name: {0}", pName));
    }
    public ViewResult CreateProduct()
    {
      Product p1 = new Product();
      p1.ProductID = 100;
      p1.Name = "Kayak";
      p1.Price = 1500;
      return View("AutoProperty", (object)String.Format("Price: {0}", p1.Price));
    }
    public ViewResult CreateCollection()
    {
      string[] stringArray = { "apple", "orange", "plum" };
      List<int> intList = new List<int> { 10, 20, 30, 40 };
      Dictionary<string, int> myDict = new Dictionary<string, int>
{ { "apple", 10 }, { "orange", 20 }, { "plum", 30 } };
      return View("Result", (object)stringArray[1]);
    }

    public ViewResult UseExtension()
    {
      ShoppingCart cart = new ShoppingCart
      {
        Products = new List<Product>
          {
            new Product {Name = "Kayak", Price = 275M},
            new Product {Name = "Lifejacket", Price = 48.95M},
            new Product {Name = "Soccer ball", Price = 19.50M},
            new Product {Name = "Corner flag", Price = 34.95M}
```

```
                }
            };
            decimal cartTotal = cart.TotalPrices();
            return View("Result", (object)String.Format("Total: {0:c}", cartTotal));
        }
        public ViewResult CreateAnonArray()
        {
            var oddsAndEnds = new[]
            {
                new { Name = "MVC", Category = "Pattern" },
                new { Name = "Hat", Category = "Clothing" },
                new { Name = "Apple", Category = "Fruit" }
            };
            StringBuilder result = new StringBuilder();
            foreach (var item in oddsAndEnds)
            {
                result.Append(item.Name).Append(" ");
            }
            return View("Result", (object)result.ToString());
        }
        Product myProduct = new Product
        {
            ProductID = 1,
            Name = "Kayak",
            Description = "A boat for one person",
            Category = "Watersports",
            Price = 275M
        };
        public ActionResult Index()

        {
            return View(myProduct);
        }
    }
}
```

Product.cs

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

namespace ProductMVCApplication.Models
{
    public class Product
    {
        public int ProductID { get; set; }
        public string Name { get; set; }
        public string Description { get; set; }
        public decimal Price { get; set; }
        public string Category { set; get; }
    }
                    }
```

ShoppingCart.cs

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

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

MyExtensionMethods.cs

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

namespace ProductMVCApplication.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;
      }
   }
                   }
```

AutoProperty.cshtml

```
@model String
@{
   Layout = null;
}

<!DOCTYPE html>

<html>
<head>
   <meta name="viewport" content="width=device-width" />
   <title>AutoProperty</title>
</head>
<body>
   <div>
      @Model
   </div>
</body>
</html>
```

Index.cshtml

```
@model ProductMVCApplication.Models.Product

@{
    ViewBag.Title = "Product Name";
    Layout = "~/Views/_BasicLayout.cshtml";
}


<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
</head>
<body>
    <div>
        Product Name = @Model.Name<br />
        Product Price=@Model.Price
    </div>
</body>
</html>
```

Result.cshtml

```
@model String
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Result</title>
</head>
<body>
    <div>
        @Model
    </div>
</body>
</html>
```

_BasicLayout.cshtml

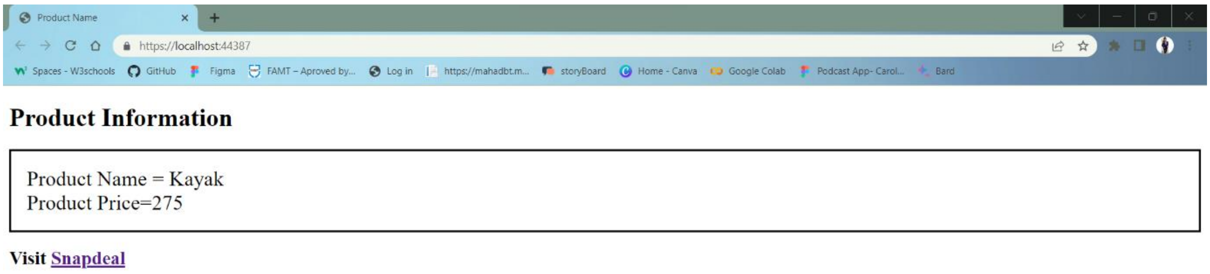
```
<!DOCTYPE html>

<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="https://www.snapdeal.com/">Snapdeal</a></h2>
</body>
</html>
```

RouteConfig.cs

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

namespace ProductMVCApplication
{
   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 }
         );
      }
   }
}
```

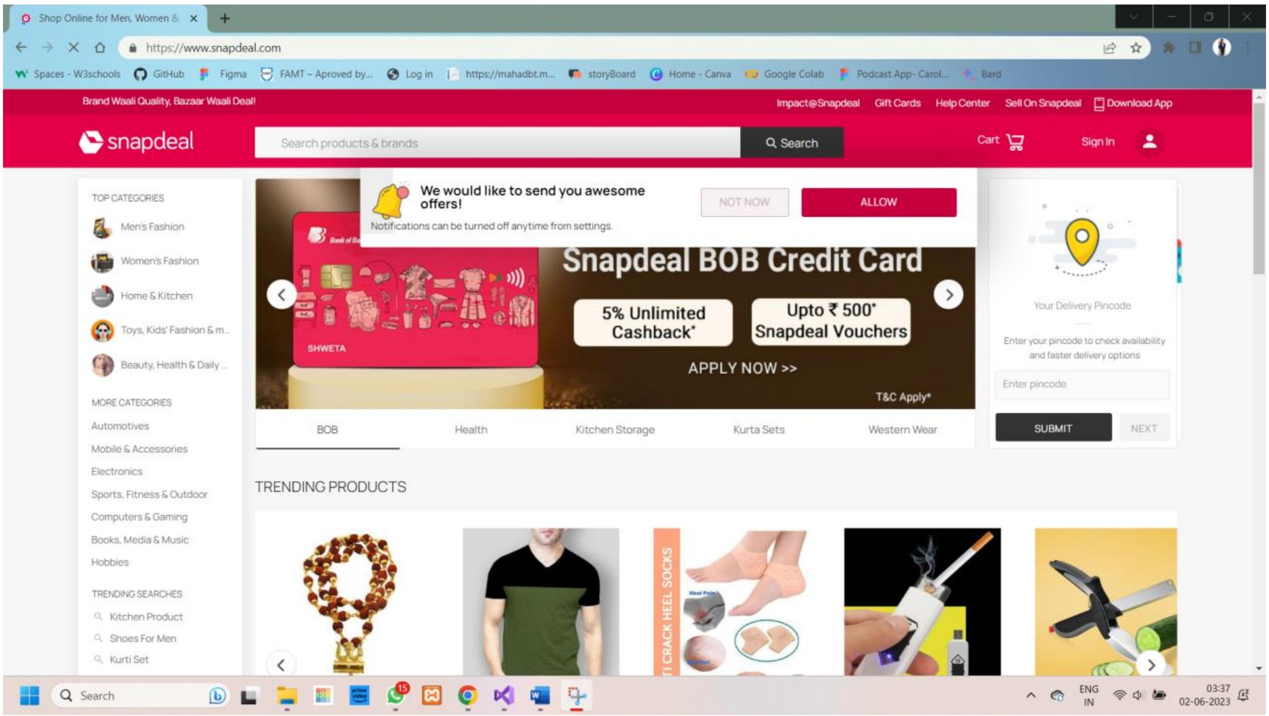
- **Output :**



Product Information

Product Name = Kayak
Product Price=275

Visit Snapdeal


Shop Online for Men, Women &
https://www.snapdeal.com
Brand Waali Quality, Bazaar Waali Deal!
Impact@Snapdeal
Gift Cards
Help Center
Sell On Snapdeal
Download App
snapdeal
Search products & brands
Search
Cart
Sign In
We would like to send you awesome offers!
Notifications can be turned off anytime from settings.
NOT NOW
ALLOW
TOP CATEGORIES
Men's Fashion
Women's Fashion
Home & Kitchen
Toys, Kids' Fashion & m...
Beauty, Health & Daily ...
MORE CATEGORIES
Automotives
Mobile & Accessories
Electronics
Sports, Fitness & Outdoor
Computers & Gaming
Books, Media & Music
Hobbies
TRENDING SEARCHES
Kitchen Product
Shoes For Men
Kurti Set
Snapdeal BOB Credit Card
5% Unlimited Cashback*
Upto ₹ 500* Snapdeal Vouchers
APPLY NOW >>
T&C Apply*
BOB
Health
Kitchen Storage
Kurta Sets
Western Wear
Your Delivery Pincode
Enter your pincode to check availability and faster delivery options
Enter pincode
SUBMIT
NEXT
TRENDING PRODUCTS
Search
ENG IN
03:37
02-06-2023