



American International University-Bangladesh (AIUB)

ADVANCED PROGRAMMING WITH .NET

SECTION – A

FALL 2020-21

Name: Ikramul Haque Chowdhury

ID: 17-34072-1

Midterm Assignment - 1

17 Nov 2020

Browser screenshots:

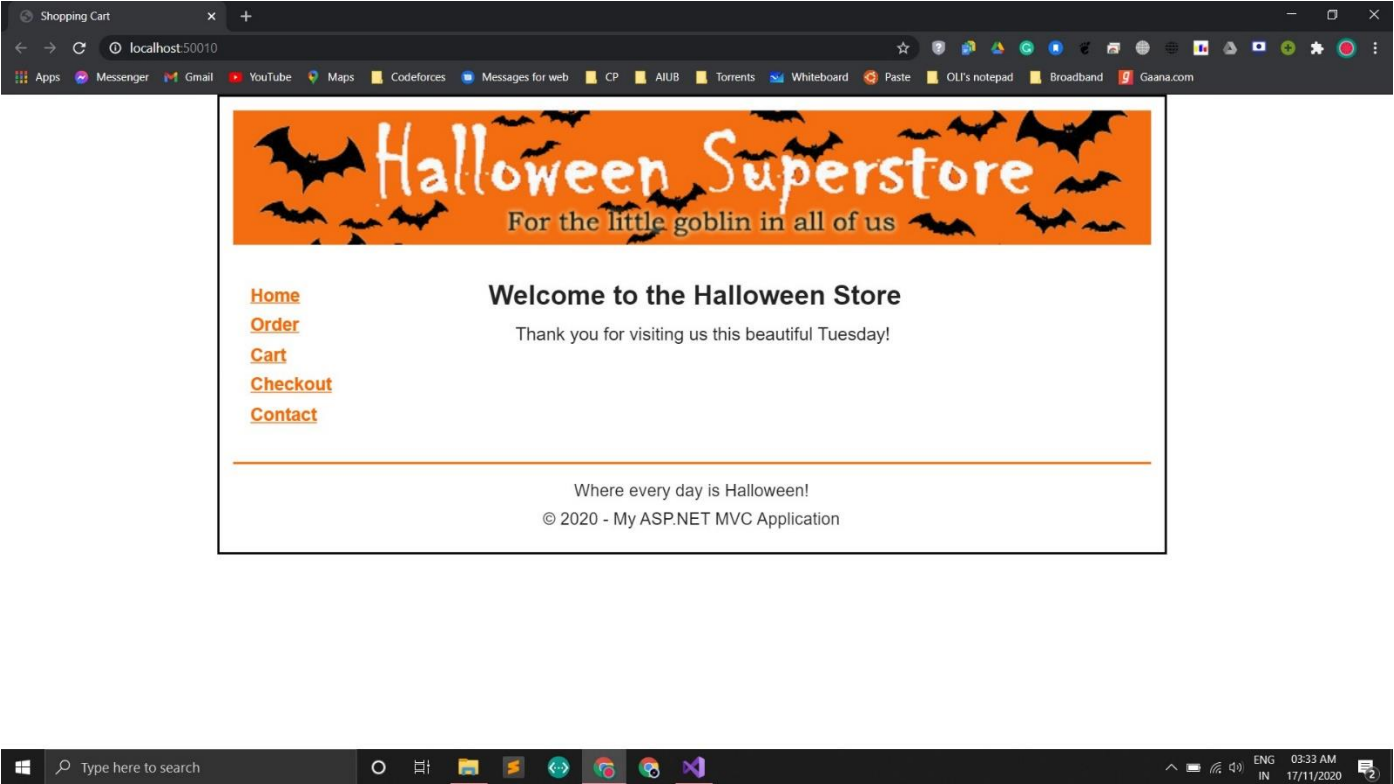


Figure: Home (/home/index)

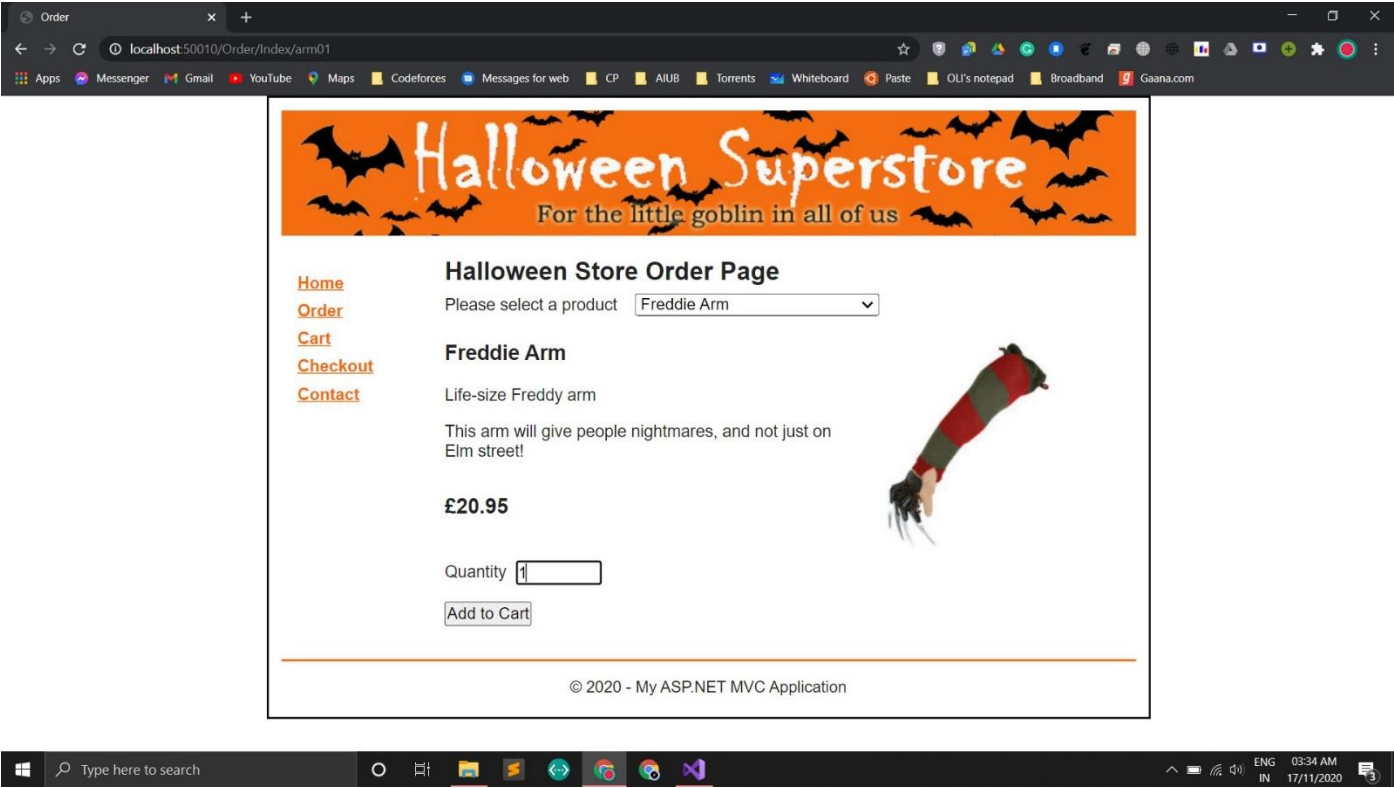


Figure: Order (/order/index)

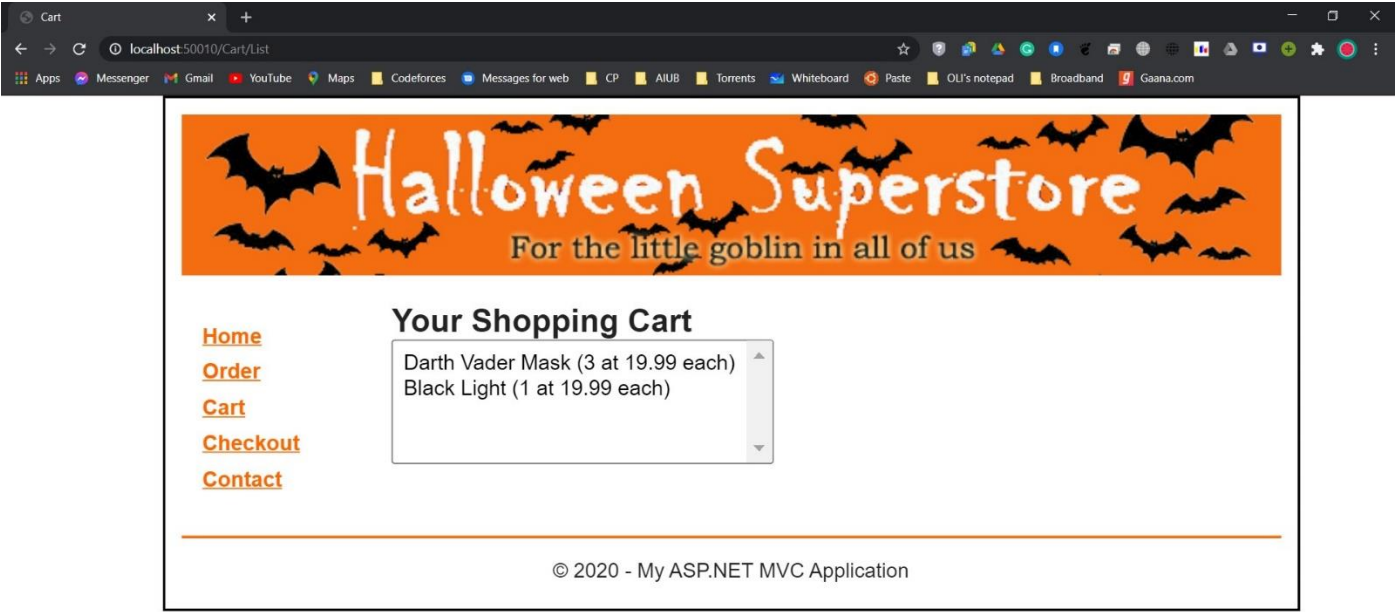


Figure: Cart (/cart/index)

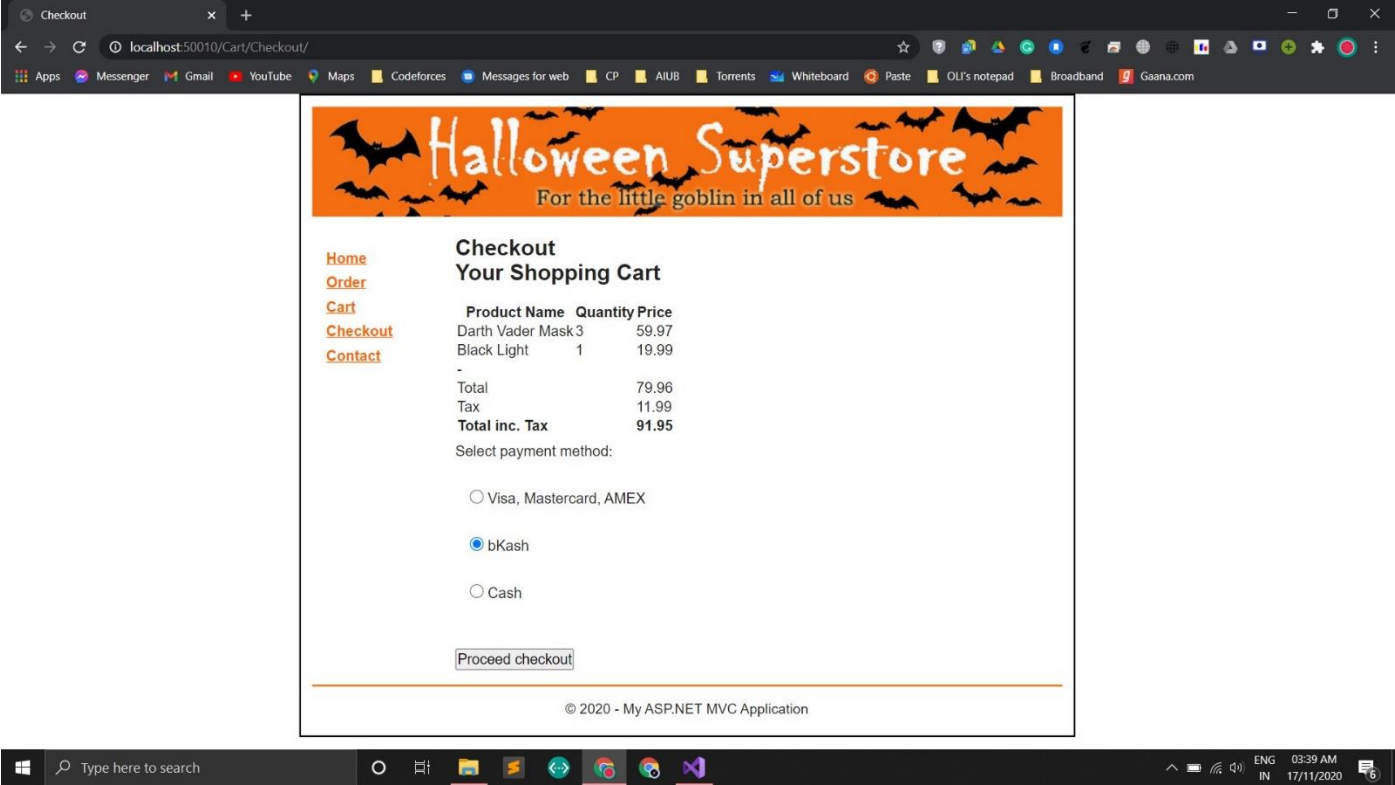


Figure: Checkout (/cart/checkout)

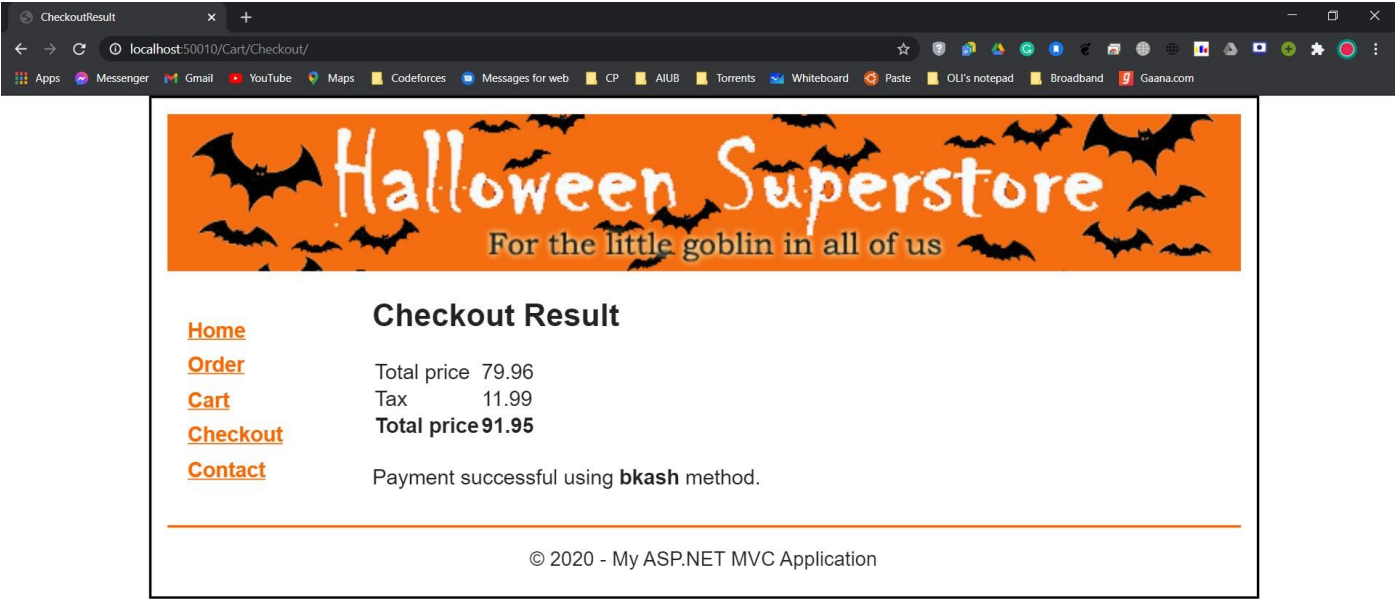


Figure: Checkout result (/cart/checkout)

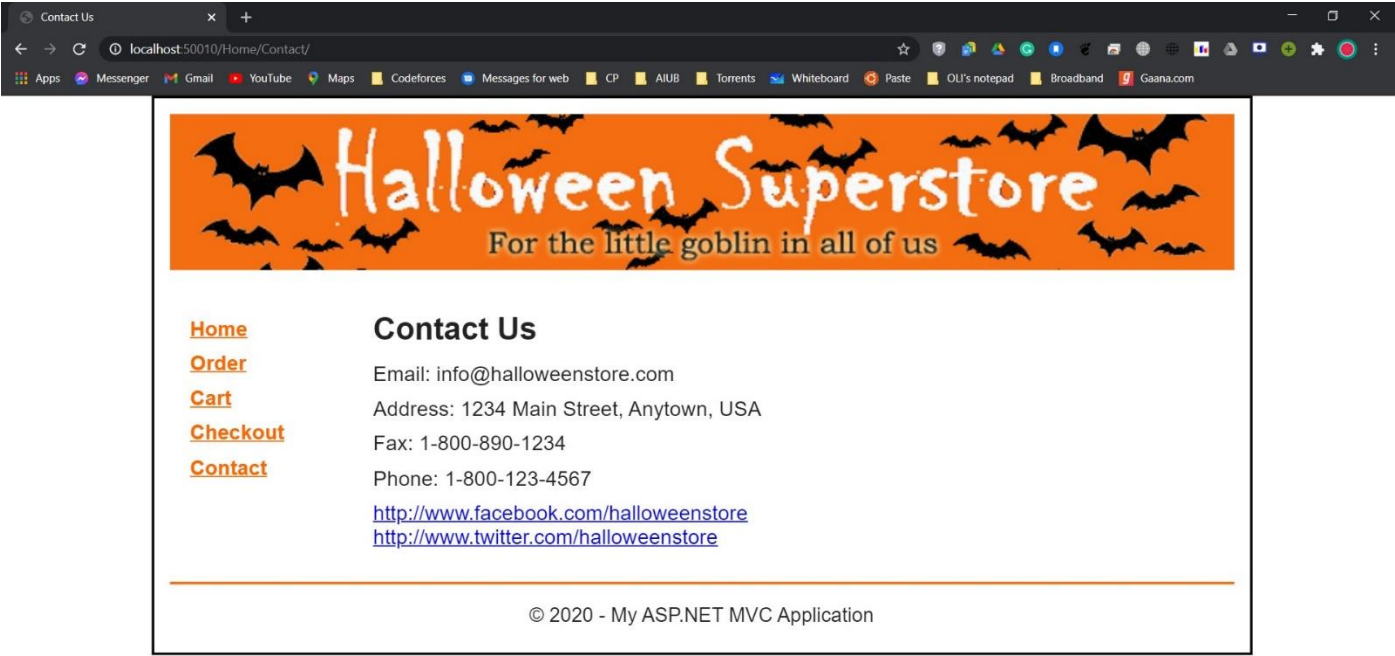


Figure: Contact (/home/contact)

Source code screenshots:

(i) Models:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Ch24ShoppingCartMVC.Models;

namespace Ch24ShoppingCartMVC.Models {
    5 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
    public class OrderViewModel
    {
        1 reference | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public SelectList ProductsList { get; set; }
        4 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public ProductViewModel SelectedProduct { get; set; }
    }
}
```

Figure: OrderViewModel

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using Ch24ShoppingCartMVC.Models;

namespace Ch24ShoppingCartMVC.Models {
    10 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
    public class CartViewModel
    {
        3 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public List<ProductViewModel> Cart { get; set; }
        6 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public ProductViewModel AddedProduct { get; set; }
    }
}
```

Figure: CartViewModel

```

namespace Ch24ShoppingCartMVC.Models {
    18 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
    public class ProductViewModel
    {
        6 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public string ProductID { get; set; }
        2 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public string Name { get; set; }
        1 reference | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public string ShortDescription { get; set; }
        1 reference | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public string LongDescription { get; set; }
        2 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public decimal UnitPrice { get; set; }
        1 reference | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public string ImageFile { get; set; }
        6 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public int Quantity { get; set; }

        //display item's property values
        0 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public string Display()
        {
            string display = string.Format("{0} ({1} at {2} each)",
                Name, Quantity.ToString(), UnitPrice.ToString("c"));
            return display;
        }
    }
}

```

Figure: ProductViewModel

```

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

namespace Ch24ShoppingCartMVC.Models {
    2 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
    public class ContactViewModel
    {
        0 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public string Email { get { return "info@halloweenstore.com"; } }
        0 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public string Phone { get { return "1-800-123-4567"; } }
        0 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public string Fax { get { return "1-800-890-1234"; } }
        0 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public string Address { get { return "1234 Main Street, Anytown, USA"; } }
        0 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public List<string> SocialMediaUrls {
            get {
                List<string> urls = new List<string>();
                urls.Add("http://www.facebook.com/halloweenstore");
                urls.Add("http://www.twitter.com/halloweenstore");
                return urls;
            }
        }
    }
}

```

Figure: ContactViewModel


```

namespace Ch24ShoppingCartMVC.Models{
    2 references | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
    public class CartModel
    {
        //Data Access methods
        1 reference | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
        private List<ProductViewModel> GetCartFromDataStore()
        {
            List<ProductViewModel> cart;
            object objCart = HttpContext.Current.Session["cart"];
            //Convert objCart to List<ProductViewModel>
            cart = (List<ProductViewModel>) objCart;
            if (cart == null)
            {
                //Create the object cart
                cart = new List<ProductViewModel>();
                //Assign cart to the Session object cart
                HttpContext.Current.Session["cart"] = cart;
            }
            return cart;
        }
        1 reference | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
        private ProductViewModel GetSelectedProduct(string id)
        {
            //Create an OrderModel object called order
            OrderModel order = new OrderModel();
            //Call the method GetSelectedProduct of the class OrderModel. Return the object returned by th
            return order.GetSelectedProduct(id);
        }
        3 references | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
        public CartViewModel GetCart(string id = "")
        {
            CartViewModel model = new CartViewModel();
            //Call the method GetCartFromDataStore
            model.Cart = GetCartFromDataStore();
            if (!string.IsNullOrEmpty(id))
                //Called the method GetSelectedProduct with parameter id and assign the return object to t
                model.AddedProduct = GetSelectedProduct(id);
            return model;
        }
        1 reference | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
        private void AddItemToDataStore(CartViewModel model)
        {
            //Add the AddedProduct to the cart
            model.Cart.Add(model.AddedProduct);
        }
        1 reference | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
        public void AddToCart(CartViewModel model)
        {
            if (model.AddedProduct.ProductID != null)
            {
                //Get the product id of the added product
                string productId = model.AddedProduct.ProductID;
                //Find the product in the car that matches the id using lambda expression.
                ProductViewModel inCart = model.Cart.Find(p => p.ProductID == productId);
                if (inCart == null)
                    //Call the method AddItemToDataStore
                    AddItemToDataStore(model);
                else
                    //Increase the Quantity by the quantity of the added product
                    inCart.Quantity += model.AddedProduct.Quantity;
            }
        }
    }
}

```

Figure: CartModel

```

namespace Ch24ShoppingCartMVC.Models {
    4 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
    public class OrderModel
    {
        private List<ProductViewModel> products;
        //Implement GetAllProductsFromDataStore
        1 reference | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public List<Product> GetAllProductsFromDataStore()
        {
            using (HalloweenEntities1 data = new HalloweenEntities1())
            { //get all the products from the Collection Products order by name using HalloweenEntities
                return (from product in data.Products
                        orderby product.Name ascending
                        select product).ToList();
            }
        }
        //Implement the method ConvertToViewModel
        2 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        private ProductViewModel ConvertToViewModel(Product product)
        {
            ProductViewModel model = new ProductViewModel();
            model.ProductID = product.ProductID;
            model.Name = product.Name;
            //implement other required properties
            model.ImageFile = product.ImageFile;
            model.LongDescription = product.LongDescription;
            model.ShortDescription = product.ShortDescription;
            model.UnitPrice = product.UnitPrice;

            return model;
        }
        //Implement the method GetProductList
        1 reference | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public List<ProductViewModel> GetProductsList() {
            if (this.products == null)
                //Call the method GetAllProducts
                this.products = GetAllProducts();
            //Return the products
            return this.products;
        }
        1 reference | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public List<ProductViewModel> GetAllProducts()
        {
            List<ProductViewModel> productmodels = new List<ProductViewModel>();
            // Call the GetAllProductsFromDataStore()
            List<Product> products = GetAllProductsFromDataStore();
            foreach (Product p in products)
            { //Call the method ConvertToViewModel to convert p and pass the method ConvertToViewModel to
              productmodels.Add(ConvertToViewModel(p));
            }
            return productmodels;
        }

        1 reference | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public Product GetProductByIdFromDataStore(string id)
        {
            using (HalloweenEntities1 data = new HalloweenEntities1())
            { //Get a product from Products of data where ProductID is matched with id parameter
                return (from product in data.Products
                        where product.ProductID == id
                        select product).FirstOrDefault();
            }
        }
        1 reference | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public OrderViewModel GetOrderInfo(string id)
        {
            OrderViewModel order = new OrderViewModel();
            //Call the method GetSelectedProduct and assign the return value to SelectedProduct property
            order.SelectedProduct = GetSelectedProduct(id);
            return order;
        }
        2 references | Ikramul Haque Chowdhury, 2 hours ago | 1 author, 1 change
        public ProductViewModel GetSelectedProduct(string id)
        {
            if (this.products == null)
                //call the method ConvertToViewModel and pass the method GetProductByIdFromDataStore(id)
                return ConvertToViewModel(GetProductByIdFromDataStore(id));
            else
                //Get the product from the products where ProductID is matched with id (Using Lambda expression)
                return products.Find(p => p.ProductID == id);
        }
    }
}

```

Figure: OrderModel

(ii) Views:

```
1  @{ //INITIALIZE web page title to Shopping Cart
2      ViewBag.Title = "Shopping Cart";
3      string day = DateTime.Now.DayOfWeek.ToString();
4  }
5  @section css {
6      <link href="~/Content/Home.css" rel="stylesheet" />
7  }
8  @*Display
9  Welcome to the Halloween Store
10 Thank you for visiting us this beautiful CURRENT DAY!*@
11 <h1>@ViewBag.HeaderText</h1>
12 <div>Thank you for visiting us this beautiful @day!</div>
13
14 @section footer {
15     @*Display
16     Where every day is Halloween!*@
17     <p>@ViewData["FooterText"]</p>
18 }
```

Figure: Home/Index.cshtml

```
1  @model Ch24ShoppingCartMVC.Models.ContactViewModel
2  @{
3      ViewBag.Title = "Contact Us";
4  }
5  @* Display
6      Contact Us
7      Email
8      Phone
9      Fax
10     Social Media URLs*@
11 <h1>Contact Us</h1>
12 <p>Email: @Model.Email</p>
13 <p>Address: @Model.Address</p>
14 <p>Fax: @Model.Fax</p>
15 <p>Phone: @Model.Phone</p>
16
17 <p>@foreach (string url in Model.SocialMediaUrls) {
18     <a href="@url" target="_blank">@url</a> <br />
19 }
20 </p>
```

Figure: Home/Contact.cshtml

```

@model Ch24ShoppingCartMVC.Models.OrderViewModel
@{
    ViewBag.Title = "Order";
    var product = Model.SelectedProduct;
}
@section css {
    <link href="~/Content/Order.css" rel="stylesheet" />
}
<form id="frmDropDown" action="/Order/Index/" method="post">
    <h2>Halloween Store Order Page</h2>

    <label>Please select a product<\/label>
    <select id="ddlProducts" name="ddlProducts" class="autoPostBack">
        @foreach (var item in Model.ProductsList) {
            <option value="@item.Value" selected="@item.Selected">@item.Text<\/option>
        }
    <\/select>
<\/form>
@using (Html.BeginForm("List", "Cart", FormMethod.Post, new { id = "frmCart" })) {
    <div id="productData">
        <label id="lblName">@product.Name<\/label>
        @Html.Label("", @product.ShortDescription, new { id = "lblShortDescription" })
        <label id="lblLongDescription">@product.LongDescription<\/label>
        <label id="lblUnitPrice">@product.UnitPrice.ToString("c2")<\/label>
        <label id="lblQuantity">Quantity<\/label>
        @Html.TextBoxFor(m => m.SelectedProduct.Quantity, new { id = "txtQuantity" })
        @Html.HiddenFor(m => m.SelectedProduct.ProductID)
        <br />
        <input type="submit" id="btnAdd" value="Add to Cart" />
    <\/div>
    
}

```

Figure: Order/Index.cshtml

```

@model Ch24ShoppingCartMVC.Models.CartViewModel
@{
    ViewBag.Title = "Cart";
}
@section css {
    <link href="~/Content/Cart.css" rel="stylesheet" />
}
<h2>Your Shopping Cart<\/h2>
@if (Model.Cart.Count == 0)
{
    <p>Nothing in your cart yet.<\/p>
}
else
{
    <select multiple="multiple">
        @foreach (var item in Model.Cart)
        {
            <option>@item.Name (@item.Quantity at @Math.Round(item.UnitPrice, 2) each) <\/option>
        }
    <\/select>
}

```

Figure: Cart/List.cshtml

```

@{
    ViewBag.Title = "Checkout";
}

<h2>Checkout</h2>
@model Ch24ShoppingCartMVC.Models.CartViewModel
@section css {
    <link href="~/Content/Cart.css" rel="stylesheet" />
}
<h2>Your Shopping Cart</h2>
<br />
@if (Model.Cart.Count == 0)
{
    <p>Nothing in your cart yet.</p>
}
else
{
    decimal totalCost = 0;
    <form method="post">
        <table>
            <tr><th>Product Name</th><th>Quantity</th><th>Price</th></tr>
            @foreach (var item in Model.Cart)
            {
                <tr> <td>@item.Name</td> <td>@item.Quantity</td> <td>@Math.Round(item.Quantity * item.UnitPrice, 2)</td> </tr>
                totalCost += (item.Quantity * item.UnitPrice);
            }
            <tr> <td>-</td> <td></td> <td></td> </tr>
            <tr> <td>Total</td> <td></td> <td>@Math.Round(totalCost, 2)</td> </tr>
            @if (decimal tax = totalCost * (decimal)ViewBag.Tax; )
            <tr> <td>Tax</td> <td></td> <td>@Math.Round(tax, 2)</td> </tr>
            <tr> <td><b>Total inc. Tax<br /></td> <td></td> <td><b>@Math.Round(totalCost + tax, 2)</b></td></tr>
        </table>
        <input type="hidden" name="total" value="@Math.Round(totalCost, 2)" />
        <input type="hidden" name="tax" value="@Math.Round(tax, 2)" />
        <input type="hidden" name="totalinctax" value="@Math.Round(totalCost + tax, 2)" />
        <p>Select payment method: </p>
        <ul><input type="radio" name="payment" value="card" /> <span>Visa, Mastercard, AMEX</span></ul>
        <ul><input type="radio" name="payment" value="bkash" /> <span>bKash</span></ul>
        <ul><input type="radio" name="payment" value="cash" /> <span>Cash</span></ul>
        <br /><input type="submit" value="Proceed checkout" name="proceed" />
    </form>
}

```

Figure: Cart/Checkout.cshtml

```

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

<h2>Checkout Result</h2>
<br />
<table>
    <tr>
        <td>Total price</td>
        <td>@Model["total"]</td>
    </tr>
    <tr>
        <td>Tax</td>
        <td>@Model["tax"]</td>
    </tr>
    <tr>
        <td><b>Total price</b></td>
        <td><b>@Model["totalinctax"]</b></td>
    </tr>
</table>
<br />
Payment successful using <b>@Model["payment"]</b> method.

```

Figure: Cart/CheckoutResult.cshtml

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width" />
  @*Create a ViewBag property called Title*@
  <title>@ViewBag.Title</title>
  @Styles.Render("~/Content/css")
  @Scripts.Render("~/bundles/modernizr")
  <link href="~/Content/Main.css" rel="stylesheet" />
  @RenderSection("css", required: false)
</head>
<body>
  <header></header>
  <aside>
    <nav>
      <ul id="menu">
        <li>@Html.ActionLink("Home", "Index/", "Home")</li>
        <li>@Html.ActionLink("Order", "Index/", "Order")</li>
        <li>@Html.ActionLink("Cart", "Index/", "Cart")</li>
        <li>@Html.ActionLink("Checkout", "Checkout/", "Cart")</li>
        <li>@Html.ActionLink("Contact", "Contact/", "Home")</li>
      </ul>
    </nav>
  </aside>
  <section>
    @RenderBody()
  </section>
  <footer>
    @RenderSection("footer", required: false)
    <p id="copyright">&copy; @DateTime.Now.Year - My ASP.NET MVC Application</p>
  </footer>
  @Scripts.Render("~/bundles/jquery")
  <script src="~/Scripts/halloween.js" type="text/javascript"></script>
  @RenderSection("scripts", required: false)
</body>
</html>

```

Figure: Shared/_Layout.cshtml

(iii) Controllers:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Ch24ShoppingCartMVC.Models;

namespace Ch24ShoppingCartMVC.Controllers {
    0 references | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
    public class HomeController : Controller
    {
        [HttpGet]
        0 references | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
        public ActionResult Index() {
            ViewBag.HeaderText = "Welcome to the Halloween Store";
            ViewData["FooterText"] = "Where every day is Halloween!";
            return View();
        }
        [HttpGet]
        0 references | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
        public ActionResult Contact() {
            //CREATE A ContactViewModel OBJECT call model model
            ContactViewModel model = new ContactViewModel();
            //Pass model to View
            return View(model);
        }
    }
}
```

Figure: HomeController.cs

```
namespace Ch24ShoppingCartMVC.Controllers {
    0 references | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
    public class OrderController : Controller
    {
        private OrderModel order = new OrderModel();

        [HttpGet]
        0 references | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
        public ActionResult Index(string id) {
            //get list for drop down from temp data called products
            SelectList products = (SelectList)TempData["products"];
            if (products == null) {
                //CALL THE METHOD GetProductList
                var list = order.GetProductsList();
                //CREATE THE SelectList products
                products = new SelectList(list, "ProductID", "Name", id);
            }
            //if no URL parameter, get first product from list and refresh
            if (string.IsNullOrEmpty(id)) {
                id = products.ElementAt(0).Value;
                //ASSIGN products to temp data called products
                TempData["products"] = products;
                //Redirect to the action method Index of the Order controller with id parameter.
                return RedirectToAction("Index", new { Id = id });
            }
            else { //get selected product and return in view method
                //Call the method GetOrderInfo to get an OrderViewModel object called model
                OrderViewModel model = order.GetOrderInfo(id);
                //Assign products to ProductsList property of model
                model.ProductsList = products;
                //Assign the quantity of the SelectProduct of the model to 1
                model.SelectedProduct.Quantity = 1;
                //Send the model object to the view.
                return View(model);
            }
        }

        [HttpPost] //post back - get selected ddl value and refresh
        0 references | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
        public RedirectToRouteResult Index(FormCollection collection)
        {
            string pID = collection["ddlProducts"];
            //Redirect to the action method index of the Order controller with parameter the id assigned to pID
            return RedirectToAction("Index", new { Id = pID });
        }
    }
}
```

Figure: OrderController.cs

```

public class CartController : Controller
{
    private CartModel cart = new CartModel();

    [HttpGet]
    0 references | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
    public RedirectToRouteResult Index()
    {
        return RedirectToAction("List/");
    }

    [HttpGet]
    0 references | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
    public ViewResult List()
    {
        CartViewModel model = (CartViewModel)TempData["cart"];
        //if the model is null, then call the method GetCart
        model = cart.GetCart();
        //Passing model to View
        return View(model);
    }

    [HttpPost]
    0 references | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
    public RedirectToRouteResult List(OrderViewModel order)
    {
        CartViewModel model = cart.GetCart(order.SelectedProduct.ProductID);
        //Assign the quantity of the selected product to the quantity of the added product
        model.AddedProduct.Quantity = order.SelectedProduct.Quantity;
        //Call the method AddtoCart
        cart.AddtoCart(model);
        //Assign model to the TempData
        TempData["cart"] = model;
        return RedirectToAction("List", "Cart");
    }

    [HttpGet]
    0 references | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
    public ViewResult Checkout()
    {
        CartViewModel model = (CartViewModel)TempData["cart"];
        //if the model is null, then call the method GetCart
        model = cart.GetCart();
        //Passing model to View
        ViewBag.Tax = 0.15;
        return View(model);
    }

    [HttpPost]
    0 references | Ikramul Haque Chowdhury, 3 hours ago | 1 author, 1 change
    public ViewResult Checkout(FormCollection data)
    {
        return View("CheckoutResult", data);
    }
}

```

Figure: CartController.cs