Intern Program: Week 4 Assignment

**Purpose:** This document outlines the assignment that will be handed out following the first review for the intern program. This project is to be worked on while you complete the assigned readings for Week 4, and must be completed for your Week 4 review.

# Week 4 Requirements:

This week will require users to build the following elements:

* **Customer Layout:** A shared layout will be used so that all customer focused pages share the same design.
* **Search View:** This page will provide the customer with the ability to search the application. They will identify products they are interested in and add them to their shopping cart.
* **ShoppingCart View:** This will allow customers to manage and maintain their shopping cart, by removing or updating quantities and then eventually placing orders.
* **Register:** A page that allows users to register with the application.
* **Login:** A page that allows users to login to the application.
* **GitHub:** Make sure to commit your changes to the repository when you are done.

## Customer Layout:

The following are the business requirements for the page:

All customer pages should use the same Shared layout, and it will display the following:

* The Store Title in the Upper left hand corner, and the navigation bar underneath that, with the following options:
  + Home
  + Search
  + Profile
* A link in the upper right hand corner that says “View Cart”, with a count next to it of the number of items in your shopping cart.

## Search:

The following represent the business requirements for the search page.

The user should have the ability to search through all the published products in the system. And their returned results should show the following:

* The name of the product
* An image for that product
* The price of the product

In addition to that, each row in the grid should have an “Add to Cart” link available, which allows the user to immediately add the items they want to the cart.

## ShoppingCart:

This screen should provide a listing of all the items that are currently in a user’s shopping cart. This should show a screen that displays a grid. That grid should display:

* Product Name
* Image
* Quantity
* Price

Additionally the quantity should be an editable field that displays as a textbox. Allowing the user to change the quantity ordered.

There should also be a button on each row called “Remove” that allows users to remove a product from the shopping cart.

Finally there is a button at the lower right hand corner of the grid that says “Place Order”.

# Technical Requirements:

The following are the technical requirements for each page in this assignment: Below is a listing of the approaches that are to be used to satisfy the business requirements above. All database transactions in this week should be completed using Entity Framework with a Database First Approach.

## Customer Layout:

The following requirements must be met for this week for this page:

* This MVC layout should reside in the “Views\Shared” folder, and be titled \_CustomerLayout.cshtml.
* You must create a separate class file under “Models” that holds the CustomerBaseViewModel. This class will have the following defined:
  + A property called “UserName”, which for the time being will return your name in the first initial, last name. Example “kmack” for “Kevin Mack”. This property should access the Session to retrieve the username.
  + A property called “UserID”, which for the time being will return the ID of your record in the user table. If you have not done so, please add a record to the user table for yourself.
* This page must have three specific criteria defined:
  + ViewBag.Title: So that the title of the page can be unique with each page.
  + Body: The main body section of the page, for normal page content, should be rendered using RenderBody().
  + Script: Located after all CSS and Javascript calls at the end of the footer.
  + (See the pre-generated “Views\Shared\\_Layout.cshtml” for an example.)
* This page should use the CustomerBaseViewModel as it’s “@model” and reference the “UserName” and “UserID” properties created above to display the user’s information.

## Search:

The following requirements must be met for this week for this page:

* These pages will be created using ASP.NET MVC using a controller and multiple views.
* Create a new controller called “SearchController”
* This controller should have the following Actions:
  + The “Index” action will be a blank search page with a text box and search button.
  + The “Search” action will happen when the user clicks the search button and take a string parameter that will be used to search against the products in the database and display the results. This page may use the same view as “Index” or a new one.
    - There must be a special message when a search returns no results.
  + The “AddToCart” action will happen when the user clicks the Add To Cart button on a product result line and will use AJAX to add the product to the customer’s cart without refreshing the page.
* This page should implement the Customer Layout.
* This page will utilize a method called “SearchProducts”, which will accept a single string parameter. This method needs to return an object that fills the columns defined above.
* This grid should display the image alongside the product name and price.
* The “Add to Cart” action should call the “AddShoppingCartItem” method, this method should work the same as the “spAddShoppingCartItem” stored procedure.
* This grid should have paging set to 50 rows.
* A new view model called “SearchViewModel” should be created that inherits from “CustomerBaseViewModel”.
  + This model should be utilized on the Search Views.
* This page will implement a single textbox and button with a label called “Search”, and a button that says “Search Products”. These controls must have unqiue and meaningful names.

## ShoppingCart:

The following requirements must be met for this week for this page:

* These pages will be created using ASP.NET MVC using a controller and multiple views.
* This page should implement the Customer Layout.
* A new view model called “ShoppingCartViewModel” should be created that inherits from “CustomerBaseViewModel”.
  + This model should be utilized on the Shopping Cart views.
  + The model should have a property called “ShoppingCartItems” that is loaded by Entity Framework. This list of items should be displayed as the main content of the page.
* It will display the same columns as “Search”.
* Additionally it will provide the ability to display the image inline in the grid.
* There should additionally be a button that allows for removing items from the cart, this button should use AJAX to make the data operation not require a refresh. The row should disappear from the screen only if it is successfully removed from the database.
* The quantity should also be a textbox, displaying the number of items. There should be a button on the page called “Update Cart”, which will save these changes to the database.

## Register:

The following are the requirements for this page:

* This page may be implemented using either MVC or Web Forms.
* This page should provide the user with fields to enter the following:
  + Username
  + Password
  + Confirm Password
  + EmailAddress
* This page should also implement required field validators on all fields.
* This page should implement a compare validator between “Password” and “ConfirmPassword”.
* This page should implement a regular expression validator to check if the email address is a valid email.

## Login:

The following are the requirements for this page:

* This page may be implemented using either MVC or Web Forms.
* This page should have the following fields:
  + Username
  + Password
* This page should implement required field validators on both.­

## GitHub

Ensure that all your changes are committed to your GitHub repository for review.