Assignment 2 – FutureBuy or BestShop

# Purpose

* Design a Web Store using ASP.NET Web Forms

# Due Date

* This lab is due during the last lab period of week 13

# Assessment

* This Lab is worth 20% of your total course mark.

# Estimated Time

* This assignment is estimated to take 14 hours.

This is only an estimate of the time required to complete this assignment. I would encourage you to work at your own pace even if that means taking extra time to understand the concepts of the lab.

# Lab Supplies

To complete this lab you will require the following lab supplies:

* Visual Studio 2013
  + Available on MSDNAA or DreamSpark
* Windows Server 2012 R2
  + Available on MSDNAA or DreamSpark
* Microsoft SQL Server 2012
  + Available on MSDNAA or DreamSpark
* AJAX Control Toolkit
* Access to Windows Azure
* Access to Windows Server Virtual Machine

# Summary of Tasks

1. Read the Entire Assignment before Starting
2. Design
   1. ~/App\_Theme/General
   2. ~/MasterPage.Master
   3. ~/MasterPageMenu.Master
   4. ~/MasterPageAdminMenu.Master
3. ~/Default.aspx
4. ~/CategoryDetails.aspx
5. ~/ProductDetails.aspx
6. ~/Login.aspx
7. ~/Cart.aspx
8. ~/Checkout.aspx
9. ~/ThankYou.aspx
10. ~/Admin/Default.aspx
11. ~/Admin/AddCategory.aspx
12. ~/Admin/EditCategory.aspx
13. ~/Admin/AddProduct.aspx
14. ~/Admin/EditProduct.aspx
15. ~/Admin/BadWords.aspx
16. Demo Preparation

# Task 1 – Read the Entire Assignment before starting

Seriously, read the Entire Assignment before starting

# Task 2 – Design

Use the Common Look and Feel section in Blackboard (Content -> Module 2 -> Week 5) and choose a layout for your blog. Note, if you have another layout in mind please go ahead and use it; the basic layouts are just suggestions. Create an App Theme called ‘General’ that will house your design choices.

Create the master pages:

1. ~/MasterPage.Master
   * This master page will be used for
     + ~/Login.aspx
     + ~/Cart.aspx
     + ~/Checkout.aspx
     + ~/ThankYou.aspx
2. ~/MasterPageMenu.Master
   * This master page will be used for
     + ~/Default.aspx
     + ~/ProductDetails.aspx
     + ~/CategoryDetails.aspx
   * This master page must contain a menu that displays all of the categories of products in your store
     + When the user clicks on a category that user is directed to ~/CategoryDetails.aspx with the category id passed to the page using the Request string
3. ~/MasterPageAdminMenu.aspx
   * This master page will be used for
     + ~/Admin/Default.aspx
     + ~/Admin/AddCategory.aspx
     + ~/Admin/Editproduct.aspx
     + ~/Admin/AddProduct.aspx
     + ~/Admin/EditProduct.aspx
     + ~/Admin/BadWords.aspx
   * This master page must contain a menu listing the appropriate choices above

The Master Pages should contain:

* Name of your Store
* Your Name
* Student Number
* Email Address
* A link to ~/Cart.aspx
* If the user is logged out of the application a link must be provided to the Login Page
* If the user is logged in the users first and last name must be displayed

The Master Pages should also contain a breadcrumb that allows the user to easily navigate around the application. For more information about breadcrumbs read: <http://en.wikipedia.org/wiki/Breadcrumb_(navigation)>

# Task 3 – ~/Default.aspx

This page is the landing page of your web application. The page will contain a list of all the products that are on sale within your store.

This page must inherit from MasterPageMenu.Master

Use a GridView or Repeater to display each items:

* Name
* Regular Price
* Sale Price
* Image
* Short Description

If more than 5 items in your store are on sale ‘page’ the GridView or Repeater

# Task 4 – ~/CategoryDetails.aspx

This page is a landing page for your stores ‘Categories’. Each item, when added to the store, must belong to a Category. For example, the iPhone and Nexus Phone would belong to the ‘Mobile Phone’ category. Categories are defined in the administration area.

When a user clicks on a category on the MasterPageMenu.Master master page they are redirected to ~/CategoryDetails.aspx and the category id is passed to the page using the request string.

When the user lands on this page you will display all of the items that belong to this category. Paging is not required.

When a user clicks on an item’s name they are redirected to ~/ProductDetails.aspx with the product id in the Request string. Display the following product values:

* Name
* Regular Price
* Sale Price
* Image
* Short Description

This page inherits from MasterPageMenu.Master

# Task 5 – ~/ProductDetails.aspx

This page is the landing page for a single item in your store. Retrieve the item from your store using the product id passed in the Request string.

This page will have the following features:

* Name
* Price (and sale price if applicable)
* Short Description
* AJAX Tab Control with the following tabs
  + Long Description Tab
    - Where the full product description is displayed
  + Comments/Rating Tab
    - A user who is logged into the system can post comments
    - Comments must be filtered using the Bad Words list as defined in the administration area
* When the user is logged into the system an ‘Add to Cart’ button is available.
  + When clicked this item is added to the users cart
  + I suggest storing ‘product id’s in the session state, or a cookie so they can be retrieved on the ~/Cart.aspx page

This page inherits from MasterPageMenu.Master

# Task 6 – ~/Login.aspx

This page allows the user to log into the application.

Create a “SOAP Client” using the following web service address: <https://ltsservices.algonquincollege.com/AuthenticateService.asmx>

This web service allows anyone with a valid Algonquin College Network Username and Password to log into the system. Upon successful login you will be granted access to a User object that contains:

* UserName
* FirstName
* LastName
* Email
* IsStudent
* IsStaff
* IsFaculty

Store this user object in the ‘Session State’ for retrieval later (hint, Master Pages must display some form of user information if the user is logged in)

# Task 7 – ~/Cart.aspx

This page represents the users shopping cart. This page is not accessible until the user logs into the system.

When an item is ‘Added to the Cart’ from the ~/ProductDetails.aspx page it should appear on this page.

* The user should be able to remove items from the cart.
* The cart allows only a single item of any type to be purchased in a single transaction.
* This page shows the total amount of all of the items being purchased.
* This page requires a ‘Buy Now’ button. When the ‘Buy Now’ button is pressed the user is redirected to ~/Checkout.aspx.
* This page requires a ‘Continue Shopping’ button that redirects the user to ~/Default.aspx

This page inherits from MasterPage.Master

# Task 8 – ~/Checkout.aspx

This page collects FAKE credit card information from the user. The information should include:

* Name
* 9 Digit Credit Card Number (obviously fake!)
* Expiry Date

This page requires a ‘Complete Transaction’ and ‘Cancel’ button. When the ‘Complete Transaction’ button is clicked the user is redirected to ~/ThankYou.aspx. When the ‘Cancel’ button is clicked the user is redirected to ~/Default.aspx

This page inherits from MasterPage.Master

# Task 9 – ~/ThankYou.aspx

This page is a verification page that notifies the user that the transaction is complete. Provide a statement such as ‘Thank you for shopping at <store name>. Your items will be delivered shortly.’

This page requires a button to return the user to ~/Default.aspx

This page inherits from MasterPage.Master

# Task 10 - ~/Admin/Default.aspx

This area is the home page of the administration area. Only administrators are allowed to come to this page.

This page inherits from MasterPageAdminMenu.Master

On this page display all of the Categories using a GridView or Repeater. This GridView or Repeater must have a ‘Delete’ and ‘Edit’ button. If the user clicks the ‘Delete’ button the category, and all associated products are deleted from the database. If the user clicks the ‘Edit’ button the user is redirected to ~/Admin/EditCategory.aspx and the category id passed to the page using the request string.

On this page you will also display a list of all of the Products in your store using a GridView or Repeater. This GridView or Repeater must have a ‘Delete’ and ‘Edit’ button and display the products Name, Short Description and Price. If the user clicks the ‘Delete’ button the product, is deleted from the database. If the user clicks the ‘Edit’ button the user is redirected to ~/Admin/EditProduct.aspx and the product id passed to the page using the request string.

# Task 11 - ~/Admin/AddCategory.aspx

This area allows the administrator to create new product Categories. For example, iPhone and Nexus phones might belong to a ‘Mobile Phone’ category.

In your database store the following information about Categories

* CategoryId, int, identity, primary key
* Name, varchar(255)

Design a form to collect the appropriate information and save it in the database.

This page inherits from MasterPageAdminMenu.Master

# Task 12 - ~/Admin/EditCategory.aspx

This area allows the administrator to edit an existing product Category. This page should be able to edit the ‘Name’ column of your category table.

This page inherits from MasterPageAdminMenu.Master

# Task 13 - ~/Admin/AddProduct.aspx

This area allows the administrator to add a product to your database.

In your database store the following information about Products

* ProductId, int, identity, primary key
* CategoryId, int, not null
* Name, varchar(255), not null
* ShortDescription, varchar(512) , not null
* LongDescription, varchar(MAX) , not null
* Price, decimal, not null
* SalePrice, decimal, not null
* IsOnSale, bit, not null
* ImageName, varchar(255), not null

Design a form to collect the appropriate information and save it in the database.

* The LongDescription must be collected using the AJAX HTHMEditor control
* The CategoryId must be collected using a drop down list of all available Categories in the system
* Everything else is up to you

Use the lessons learned in Lab 7 task 2 to allow the administrator to upload and save a product image into Azure Storage and retrieve the image using a Generic Handler.

This page inherits from MasterPageAdminMenu.Master

# Task 14 - ~/Admin/EditProduct.aspx

This area allows the administrator to edit an existing Product. This page should be able to edit and change all of the information stored in both the database and the image stored in the cloud.

This page inherits from MasterPageAdminMenu.Master

# Task 15 - ~/Admin/BadWords.aspx

This are allows an administrator to add and remove a ‘Bad Word’ to your database.

In your database store the following information about BadWords.

* BadWordId, int, identity, primary key
* Word, varchar(50), not null

Any word listed here is filtered out of the comments on the ~/ProductDetails.aspx page.

This page inherits from MasterPageAdminMenu.Master

# Task 16 – Demo Prep

* Load your database with at least 3 categories
* Load each category with a least 6 products
* Make at least 7 of the products ‘On Sale’ using the IsOnSale flag.

# NOTES:

* The words ‘Item’ and ‘Product’ are used interchangeably throughout this document. They both mean a product found in your store.
* Each page must have a title
* Master Pages must contain breadcrumbs
* All pages that collect data must have appropriate validation
* Email Addresses and Passwords must be encrypted before they are stored in the database.
* Do not overthink the requirements
* …
* More to come as questions arise.

# Submission

Once you have confirmed that your webpage is available online, you are ready to hand in your assignment.  
  
To hand in your lab go to Blackboard and navigate to Content  Assignments and click on ‘Assignment 2’ link.

Under “Assignment Materials”, in the Submission text box write out the following Information:

* Student Number
* First Name
* Last Name
* Network Username
* The URL, or hyperlink, prepared in the lab

If you want to leave your professor a quick note fill out the Comments section.

Finally, once the Submission and Comments section are complete, click the ‘Submit’ button to send the lab to your professor.

**IMPORTANT NOTE**:

If the URL, or hyperlink, does not direct the professor to the lab you will receive a ZERO for the lab assignment.

**IMPORTANT NOTE:**

You may only submit an Assignment ONE TIME. Be sure the assignment is complete before clicking on the ‘Submit’ button.