|  |  |
| --- | --- |
|  |  |
| Due Date: | January 27, 9AM |
| Percentage of ASP.NET 2 module mark: | 40% |
| Submission Location: | ASPDotNET/Dropbox/ |
| Submission Details: | Submit your project in a ready-to-run state. |
| Late Penalty: | 20% deducted each day this assignment is late. |

Contents

[Individual Assignment 2A 2](#_Toc408215997)

[Session Management 2](#_Toc408215998)

[Collapsible Nav Bar 2](#_Toc408215999)

[Home/Index 2](#_Toc408216000)

[Home/Add 3](#_Toc408216001)

[Home/ViewCart 4](#_Toc408216002)

[Home/Thank You 5](#_Toc408216003)

[Web.config 5](#_Toc408216004)

[SQL for the ShoppingCart database 5](#_Toc408216005)

[Other Items 6](#_Toc408216006)

[Marks will be Deducted for the Following 6](#_Toc408216007)

# Individual Assignment 2A

For your project you are to create part of a shopping cart that is managed using sessions. This could be a portfolio piece. For anybody interested, you welcome to demonstrate your shopping cart at the start of my next class following the delivery time.

## Session Management

* At the end of the session in the global class file, remove the session information for *any expired sessions* in the *ProductVisit* and *Visit* tables.
* At the start of each new session:
  + remove any abandoned sessions from the database tables *Visit* and *ProductVisit* when the last update is over 1 hour old. If an update does not exist, remove any data for sessions that are over 1 hour old. Please look up how to do this and if you are really stuck you are better off to skip this requirement which isn’t worth very much than spend a lot of time on it. Or, at the very least, leave this task till the end if you are stuck figuring it out. (3 marks)
  + remove any entries from the database tables *Visit* and *ProductVisit* which have a matching session id. (3 marks)
  + Store the new session and start date and time in the Visit table. You might use string sessionID = HttpContext.Current.Session.SessionID; (1 mark)

## Collapsible Nav Bar

* Uses Twitter Bootstrap and collapses the top menu on mobile devices. (3 marks)

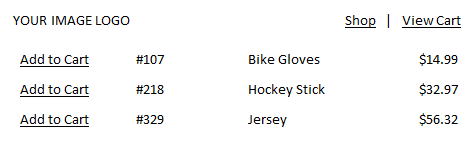


* Displays the following menu in a large device. (3 marks)



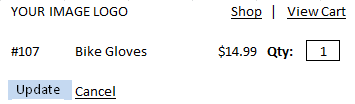
* Uses a professional looking logo image. (1 mark)
* Is referenced in your layout page which is shared by all views. (1 mark)

## Home/Index



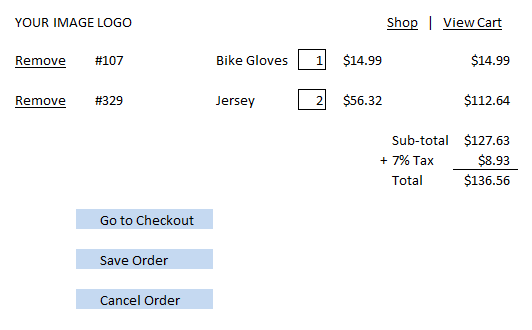
|  |  |
| --- | --- |
| Displays product information in a manner similar to the above. | **2 marks** |
| Include an image beside the product description. (Not shown in picture) Use appropriate sizing and professional presentation count. Consider this as a nice-to-have feature if you have time. It is not worth much so you might consider skipping this if you don’t have time. | **1 mark** |
| Add to Cart is a hyperlink to Home/Add for the corresponding product. | **0.5 marks** |
| Presents an attractive collapsed version of this data on mobile devices using Twitter Bootstrap. Use a responsive grid. | **2 marks** |

## Home/Add



|  |  |
| --- | --- |
| Presents an attractive collapsed version of this data on mobile devices using Twitter Bootstrap responsive grid. | **2 marks** |
| Only one item can appear on this page at a time depending on which item was selected by the user from Home/Index. | **2 marks** |
| The ‘Cancel’ link redirects the user back to Home/Index. | **1 mark** |
| A default quantity of 1 is entered in the box on arrival to this view. However, if a quantity was already submitted during the session then show the previous quantity that was selected. | **3 marks** |
| Update allows the user to submit a quantity and store it with the session data. If the item was already added to the cart then the new quantity replaces the original value that was submitted. | **5 marks** |
| Clicking update takes the user to Home/ViewCart | **1 mark** |
| The quantity submitted is properly validated to ensure it is an integer greater than 0. Otherwise a helpful error message is displayed if an incorrect value is submitted. Be sure to validate this data on the client and server. | **3 marks** |

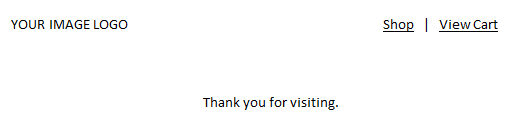
## Home/ViewCart



Note: See the assignment hint for information on how to save the order.

|  |  |
| --- | --- |
| Shows all items that have been selected during the session and their quantities. | **5 marks** |
| Shows all items in in a similar manner as above for large devices and in an attractive collapsed state for small devices using Twitter Bootstrap. | **2 marks** |
| Validates updated quantities to ensure integers greater than zero are entered. | **2 marks** |
| Updates totals as shown below when **Save Order** is clicked. | **2 marks** |
| **Go to Checkout** takes the user to Home/Thankyou | **1 mark** |
| **Cancel Order**: |  |
| * + ends the session and deletes all items in the cart. | **5 marks** |
| * + directs the user to Home/Thankyou |  |

## Home/Thank You



|  |  |
| --- | --- |
| Looks similar to the above view. | **1 mark** |

## Web.config

|  |  |
| --- | --- |
| Set the sliding timeout to 60 minutes. | **0.5 marks** |
| Use InProc mode and cookieless must not be turned on. | **0.5 marks** |

## SQL for the ShoppingCart database

Here is some SQL to help you get started. Please do not change the table structure and do not add more tables. Please keep the name “ShoppingCart” for your database.

|  |
| --- |
| DROP TABLE ProductVisit;  DROP TABLE Product;  DROP TABLE Visit;    CREATE TABLE Visit  (  sessionID VARCHAR(44) PRIMARY KEY,  started DATETiME  );  CREATE TABLE Product  (  productID INTEGER NOT NULL PRIMARY KEY,  productName VARCHAR(30),  price MONEY  );  CREATE TABLE ProductVisit  (  sessionID VARCHAR(44) NOT NULL,  productID INTEGER NOT NULL,  qtyOrdered INTEGER,  updated DATETiME,  PRIMARY KEY (sessionID, productID),  FOREIGN KEY (sessionID) REFERENCES Visit(sessionID),  FOREIGN KEY (productID) REFERENCES Product(productID)  );  GO  INSERT INTO Product VALUES(107, 'Bike Gloves', 14.99);  INSERT INTO Product VALUES(218, 'Hockey Stick', 23.97);  INSERT INTO Product VALUES(329, 'Jersey', 56.32);  INSERT INTO Visit VALUES('asdf234dfa23lkjkj232', '20140101 10:34:09 AM')  INSERT INTO ProductVisit VALUES('asdf234dfa23lkjkj232', 107, 2, '20140101 10:34:09 AM')  SELECT \* FROM ProductVisit  GO |

### Other Items

* Implements attractive and professional looking formatting with Twitter Bootstrap. Focus on alignment and intuitiveness.
* Uses a layout.
* Uses the repository / view model patterns.

# Marks will be Deducted for the Following

* Inefficient data types.
* Duplicated and inefficient code.
* Improper case usage.

camelCase – variables, private members

PascalCase – classes, namespaces, public members, enums, interfaces, readonly

UPPER\_CASE – for constants

* Improper code spacing.
* Poor and non-uniform indentation.
* Non-minimally scoped variables.
* Variable names that are not descriptive of the data they store
* Method names which are not named according to the task they fill.
* Methods which can be broken down into more succinct tasks.
* Improper split between presentation, business logic, and data access layers.
* Non-uniform formatting throughout the site.