

E-Commerce Development

Final Project

This final project will serve as your tour-de-force of programming skill, your magnum opus of e-commerce understanding. Develop an E-Commerce system with the following (minimum) characteristics:

Final Project Front End – present three links:

1. Consumer E-commerce application
 2. Administrative Interface
 - a. The link to the administrative interface must be password protected.
 - b. Check the validity of the password on the server side, not in the client-side Javascript (unless you're clever with one-way hashes; then it might work on the client).
 3. Link to the Web Service Interface
 - o Link to the web service test page on your site.
-
1. Consumer Application
 - a. Display the list of products – yours and other people's – available for sale.
 - i. Give many options for filters and sort orders, like on the homework, but more elaborate.
 - b. Allow the consumer to select **one** product to buy.
 - c. Call the “in stock” web service of the vendor of the selected product
 - i. Look up the selected product in your database.
 - ii. Note the vendor of the product.
 - iii. Call the appropriate web service by the URL you read for it.
 - d. If the item is in stock, present a fully validated and bulletproof credit card payment screen. If the item is out of stock, return to the product page with an appropriate message.
 - i. Use the common BONDS credit card engine for back end processing.
 - e. When all the credit card validation is complete, generate a receipt page for the customer.
 2. Administrative Interface
 - a. Publish your data to XML document.
 - i. Use the XML document format we developed in class web site.
 - ii. Use your *sites/directoryName* plus “.XML” as a file name.
 - b. Read another vendor's XML document and add their products to your database.
 - i. Prompt for a filename so you know what file to read
 - ii. Be sure to first delete any products from that vendor already in your database.
 3. Web Service
 - a. Accept SKU as a parameter.
 - b. Look up that product in your database to see if you have any in stock
 - c. Return the number in stock as an integer.

Extra credit will be given for additional functionality and exceptional implementations, like ...

- AJAX and or scripting tools for especially user-friendly front ends
- implementing middleware in objects in and applicataion DLL or in App_Code
- ... and more. Be creative. Be professional. Be excellent.

E-Commerce Development

Grading Criteria

- **Final Project Front End – presents three links:** **5%**
 - Consumer E-commerce application - catalog
 - Administrative Interface
 - The link to the administrative interface must be password protected.
 - Securely check the validity of the password.
 - Link to the Web Service Interface
 - Link to the web service test page on your site.
- **B2C Application** **30%**
 - Tab order and other aesthetic properties
 - Displays the list of products – yours and other people’s – available for sale.
 - Gives many options for filters and sort orders.
 - Allows the consumer to select one product to buy.
 - Calls the “in stock” web service of the vendor of the selected product
 - Looks up the selected product in the database.
 - Notes the vendor of the product.
 - Calls the appropriate web service by the URL you read for it.
 - If the item is in stock, present a fully validated and bulletproof credit card payment screen.
 - If the item is out of stock, return to the product page with an appropriate message.
 - When all the credit card validation is complete, calls the common BONDS CC processor
 - Generates a receipt page for the customer. (Or a retry option on fail.)
- **B2B Administrative Application** **30%**
 - Publishes data to XML document.
 - Uses the posted XML document format.
 - Read another vendor’s XML document and add their products to the database.
 - Prompts for a filename or presents a list so we know what file to read.
 - First deletes any products from that vendor already in the database
 - Adds those products to the database
- **Web Service** **15%**
 - Accept SKU as a parameter.
 - Look up that product in your database to see if you have any in stock
 - Return an integer of the number in stock.
- **Database** **5%**
 - Table and fields (names, data types, etc.)
 - Test data
- **Code Style** **15%**
 - Meaningful and plentiful comments
 - Readability
 - Identifier names
 - Consistent style
 - Overall use of good development principles