# Project 1: Begin to build basic E-Commerce Site

*For starters…*

Goal: You have been asked to create a simple E-Commerce site for a new E-tailer.

Example: <http://www.ist.rit.edu/~bdf/341/project1/>

Please pay attention to directory and file names. Also, make sure you include a link to your project along with any username and passwords in the dropbox.

# Requirements

* This site will consist of at least 4 PHP files: index.php, cart.php, admin.php and LIB\_project1.php
* Each page will share common graphic elements, and will have a global navigation system that links to every other page. These common elements MUST be achieved through the use of functions or a template system. Points will be deducted if common code (html or php) is copied on more than one page.
* These pages will be located in the student's: ~abc1234/Sites/341/project1/ folder on kelvin.
* The site should have an attractive design.
* When planning your code and design, you need to make sure the site can be “re-arranged” without changing the content-generating script coding. You can achieve this in one of two ways:
* Through the extensive use of <div id/class> tags and changing the order of creating the sections of the pages and CSS files OR
* Through the use of a template system, where you read in the template file replacing placeholders with content (a different template would provide a different look for the same content) in conjunction with CSS files.
* CSS (in external style sheets) should be used extensively.
* The site will pass HTML5 validation.
* index.php will have (in addition to the common elements described above):
* A “sales” section that will have any products that are on sale highlighted in this section. For each product on sale, it will include an image, brief description, original price, sale price and quantity left.
* You will have a minimum of 3 items on sale at one time and at most 5 items on sale at one time.
* A “catalog” section that will list all of the products available. For each product it will include an image, brief description, price and quantity left. This information will be displayed 5 items at a time with a paging scheme to view the next/previous page of items. If an item is on sale, **do not include** it in this section of the page.
* Each item, whether in the sale or catalog section will have a “buy/add to cart” button that when clicked will add that item to a “cart” (a table in your MySQL database) and will decrease the quantity on hand (including updating the products table and the web page).
* You will have a minimum of 15 products for sale.
* The products are loaded dynamically from a MySQL database (products table). The format for each item should be:

Product Name | Description | Price | Quantity | Image Name | Sale Price

The sale price will be 0 if the item is NOT on sale.

* cart.php will have:
* A listing of all items in the cart with a total amount of sale included.
* You should only include the product name, the description, quantity ordered (in this case only 1) and cost for the item. Do not include the image.
* There should be a button that will clear (empty) the cart (i.e remove items from the table and display an “Empty Cart” message). (Don’t worry about resetting the item count for this project).
* The information here will come from a table cart in the db.
* LIB\_project1.php will contain most of your application code:
* The code in this file will be structured as reusable functions that will be called by the other pages.
* Copious comments will describe the inputs, outputs, and purpose of each function.
* You should create functions that go beyond what we did in class.
* This file will be included (or required) by the other pages.
* admin.php will:
* Pick an item to edit and save the changes. Make sure that the number of items on sale constraint doesn’t get violated.
* Add an item for sale. The picture can be on the server already (or above and beyond allow uploading of a picture). Again, if it is on sale, make sure to follow the constraint.
* No editing or adding will be done without a correct password. (Above and beyond would be to add a login with access control and sessions instead of a password every time).
* Just in case: create backups of your catalog table when you have completed the site.
* You may elect to have additional pages.
* **ALL DATABASE QUERIES MUST BE PARAMETERIZED QUERIES USING PREPARED STATEMENTS.**
* **ALL PAGING MUST BE DONE ON THE SERVER SIDE**
* **DUE DATE: Week 7 @ 11:59pm (drop box in mycourses)**
* **Zip up entire project (file structure intact!) in a folder with your last name**

**Grading**

* Meeting the above requirements will result in a ‘B’. See the rubric for options required to get an ‘A’