**Term Project Part I: Make a Basic PHP Site (150pts)**

For our term project, we will be creating a dynamic PHP site that will combine all of the topics we cover throughout the semester. To get started, it is important that you build a base that will be able to access your site’s data (i.e. users, customers, and inventory) in the second half of the semester—so that is part I of the term project: to create the PHP site that will drive your project.

When you are done, ensure that all of your files (images, php files, style sheets, include files, etc) are pushed to gitand that it is in a directory like ***cis-222/p1*** on the CIS Linux server. **It is essential that I can see a working version of your page by entering** [***http://cis.hfcc.edu/~USERNAME/cis-222/p1/***](http://cis.hfcc.edu/~USERNAME/cis-222/p1/)**in a browser**.

This is due at the start of class on October 8th, 2019.

1. Make *index.php* Drive Your Entire Site (40 points)

Use PHP to make your *index.php* file the only URL that will ever need to be entered by users. Pages should be displayed by your index file using PHP logic.

At a minimum, each page in your site should have a header, a navigation menu, and a footer with some form of copyright and/or contact information and correctly implemented content.

Your site should have, at a minimum, 3 pages in your menu: the home (or introductory) page, a contact page, and a catalog page (or relevant page of your choosing).

The home page of the site should contain an introduction to your project.

1. Create Database Tables (40 points)

Store all of your information in separate MySQL tables so that this version of your site can function for testing. You must have at least a few rows of data in each table for testing.

Customers/user records should be able to store a customer ID number, name, an address, a phone number, some notes, and anything else related about the customer.

Product records should store an item number, an item name, an item description, an item price, and an item quantity.

1. Your Contact Page (20 points)

The contact page should contain a form that allows users to enter information about themselves and a brief message. When submitted, your site must store that contact request in a database table and display a confirmation that indicates a successful transmission **of the information the user entered**. You do not need to send out an email or text, however I can show you how if you are interested. Just note that this functionality may be disabled on the school’s server.

1. Your Catalog Page (40 points)

Your catalog page must provide a list of the products in your products table. Table rows should alternate colors for readability. The table should contain labels. The columns that should appear in the list should include the number, the name, and the price. The item number will eventually appear as a link that, when clicked, takes the user to a detail view for that item which will contain all information about it. That is where the item would be added to the cart.

1. Appearance (10 points)

You must use external CSS for all formatting and appearance. It is important to be in the habit of separating content and layout, so make sure that you do not use tags or attributes that modify appearance in the markup.

Your source (PHP, markup, and CSS) must be neatly organized and formatted with appropriate tabbing, comments, and adherence to the standards we discuss in class.

1. Extra Credit (20 points)

For security, you should be using prepared queries. For easier migration, you should be using the PDO library. This is a real security concern in live environments. If every query in your project is using a PDO prepared statement, you will be eligible for the extra credit. If a single query is interacting with your database in an unprepared or unsecure way, you will not be eligible for these points.

It is a very good idea to get in the habit of security from the start. Rather than waiting until later, or after a breach to begin making changes.