# Capstone Deliverable #1 Securing your CMS/Adding missing features

In term 1, you learned how to use PHP to manage data gathered through web forms and to retrieve that data for display on the web. You have been provided with a base CMS to manage items and associate them with specific categories.

**Injection-proof your queries**

o   Ensure you do not have **$\_POST** or **$\_GET** appearing in your queries. You must save them into variables  
  
o   Escape all variables used in queries with **mysqli\_real\_escape\_string** function (use **strip\_slashes** function to undo)

**Add “missing” functionality**

o   Add the picture upload field in the edit item form. If the field is blank (i.e. no new picture to replace), do nothing to the existing picture. If the field isn’t blank, delete the old picture, move the new picture into the images directory, and update the picture name in the database.  
  
o   Add delete functionality to the manage categories page but only show the delete button when there are no items currently in that category. Also on the **delete\_category.php** page, ensure that there are no items for the category to be deleted before allowing the DELETE query to execute.  
  
o   Delete the item image when deleting an item  
  
o   Add WYSIWYG editor to the item description on the add/edit item pages  
  
o   Ensure your admin pages are in their own directory and not mixed with public pages

**Add user authentication**

o   Create a table called **admin\_info** which consists of the following fields: **admin\_id**, **username**, **password**, **active**. For now, you can store the password in plain text.  
  
o   Create a login page. The form must include:        user name and password  
  
o   Your form will go to **checkLogin.php** where you must verify that the username and password fields match an entry in the database. The entry must also be active (i.e. **stored value of 1**). If no match, redirect back to the login form. If a match is found, retrieve the **user\_id**, generate a **session\_id** and set these values as session variables. Send the **user\_id** and **session\_id** to a table called **current\_sessions** along with the current time. Redirect to the admin main page.  
  
o   Wrap every page in your CMS (minus the login and check login pages), with the following procedure: Retrieve the **user\_id** and the **session\_id** from the browser session, check for a match in **current\_sessions** table. If there is a match, show the page. If there is no match, redirect to the login page.  
  
o   Change Password functionality (remember to prompt for old password as well as new password twice). If password is successfully changed, redirect to confirmation page.

# Capstone Deliverable #2 Web site design / Connecting to CMS / Image resizing

**Web Site Design**

o   Your product list page must have a minimum of 3 categories (links) in a vertical table at the left of the page (which will be pulled from the database). Your product listing (table on right) will display all products in the database.  You must mock-up at least 2 rows of products. For each product, you will display a thumbnail picture, the title of the item, the price and a “buy now” button. When you click on either the thumbnail or title, you will be brought to the details page where you will display a larger picture plus all other fields that are required to be stored (**title**, **product\_id**, **description**, **price**, **quantity**, **sku**).

**Connect inventory system to design prototype**

o    Modify your mockups to pull categories and products from the database. When the category link is followed, only products in that category are to be displayed

**Add image resizing into add item/check add item and edit item/check edit item**

o    After the image is moved into the images directory, resize the picture so that the thumbnail matches the width in your mockup (prefix filename with **tn\_**). Do a second resize so this one matches the width in your product display mockup (prefix filename with **lrg\_**).   
o    Use **SimpleImage** or **phpthumb** (search for it online)

# Capstone Deliverable #3 Building the Shopping Cart

-    To keep track of the shopper, you must use session variables to identify them. You will create a **session\_id** and determine their **IP address** and set them as a session variables in your browser. If you cannot retrieve a session id and IP from the session variables, you must set them up. This code should appear on every public page.  
  
-    When you press the “**add to cart”** button, you will insert the **item\_id**, the **session\_id**, **IP address**, and **quantity of 1** into a table called **shopping\_cart**. You will then redirect to **shopping\_cart.php**  
-    On the shopping cart page, you will retrieve all items for that user and display the title, quantity and price in a table. The only field that should be editable is the quantity field. To the right of the listing is an update and remove buttons. When update is pressed, you go to **update\_cart.php**, the quantity is updated (do not exceed item quantity in the database), and you are redirected back to the shopping cart. When removed is pressed, you go to **remove\_item.php**, delete the item from the shopping cart table and redirect back to the shopping cart  
  
-    At the bottom of the items, you should display the subtotal.

# Capstone Deliverable #4 Create the Order Processing System

-    Add customer information fields to the bottom of the shopping cart page (first name, last name, phone, and email). When they click the submit order button at the bottom of the form, they will go to **check\_order.php** where you will ensure that none of the fields are empty and the email is properly formatted. If any field is empty (or email is not proper), redisplay the form with the values the user entered. Display an error beside any field that is in error.  If the fields are ok, insert the user details (along with the session values) in a table called **order\_info** . Retrieve the **order\_id**. Loop through the cart and move each item into a table called **items\_sold** which contains the **item\_id**, **order\_id**, **item\_price**, and **quantity**. Redirect to **order\_thankyou.php** where you will display a receipt for the order (list of items ordered, cost of order, customer details) by retrieving the **session\_id** and **IP** from the browser. At the end of script, unset the **session\_id**. Add code to the top of the page that redirects you to the product page if the **session\_id** is not set.  
  
-    Add a view orders page to the admin area. Display a list of completed orders. When the link is pressed (or button beside it is pressed), display the customer receipt.