Admin-Side Class List

The get/set methods should get or set their respective variables.

addProductCategtory() shouldtake a category name as an argument, check the database to make sure that there is not already a category that exists with that name, if not the new category is added to the table.

addNewProduct() will be the method that pulls all of the product attributes from a form and adds them as a new product record in the database. It should take as arguments all of the class variables with the exception of maybe $isSpecial if handling product specials is going to be done on a separate page. $picture is going to be a tricky one but we can address that at a later time...

getProductInfo() should take an argument of a product ID, check if it exists, and pull all fields related to that product from the database and store them in the proper class variables to be spit out into form fields.

updateProductInfo() should take an argument of a product ID, check if it exists, and pull all of the product attributes from the form and update the record in the database for that product ID.

uploadPicture() will be a method that will bring up a dialog allowing a user to browse for a picture file to upload to the server, and then transfer the file. This one might end up becoming a class of its own if the process turns out to be more complicated than that...

The get/set methods should get or set their respective variables.

notifyOrderPlaced() should take an argument of an order number, pull the information from the database related to that order, and send a confirmation email to both the customer and the admin.

compileOrderList() should search the database for all records that are not marked isShipped and display them in a list. The info displayed for each record is something that we can discuss, but minimally I think order number, order date, and customer name would be a good start. The admin can always pull up a report with all the information on any given record after reviewing the list.

I'm open to ideas for other methods to add to this class, but I think we talked about just pulling information from the database to generate different reports on a page by page basis. I think the DBConnect class might need to have the functionality to generate a query for the report information, but I think all that is something we will need to discuss in class since I don't think either group was really assigned to create that class.

get/set methods... You know the drill.

authenticateLogon() should take a username and password as arguments, and check them against the database. Then a session variable should be created indicating the admin is logged in, all admin pages should check this variable before displaying.

logout() should reset the session variable and/or kill the session.