IIT Web Assignment - Appendix

Relative URLs

A relative URL is a URL that is determined based on the path of the current page.

Examples of a relative URL:

```
<a href = 'ViewCart.php'>View Cart</a>
<img src = '../images/image.jpg'/>
```

Examples of an absolute URL **This type of URL is NOT allowed on this assignment.**

```
<a href = 'http://webprog.cs.latrobe.edu.au/12345678/IIT/Lab08/ViewCart.php'>View Cart</a>
<a href = 'U:\public_webprog\IIT\Lab08\ViewCart.php'>View Cart</a>
```

That last one is not even a valid URL

A URL that starts with a slash is called a root relative URL. **This type of URL is NOT allowed on this assignment**.

```
<a href = '/12345678/IIT/Lab08/ViewCart.php'>View Cart</a>
```

NOTE: you are allowed to use an absolute URL if you are linking to the brands website.

The reason this is so important is because we will not be marking your assignment from your home directory, instead we run the assignments from a special UNIX account.

Remember that Windows is not cases sensitive, but UNIX is. Please ensure your URLs are the correct case.
This may cause problems if you develop your website on Windows and then copy it to UNIX.

Old HTML tags

There are a number of HTML tags and attributes that have been deprecated, which means your website must not use them because they are outdated and have been replaced by CSS.

You can always check if the tag or attribute you want to use is outdated by checking the w3schools website.

List of deprecated tags and attributes:
http://www.tutorialspoint.com/html/html_deprecated_tags.htm

Here is a list of new tags in HTML5
http://www.w3schools.com/htmL/html5_new_elements.asp

CSS Tricks

Animations.

You can do animations entirely using CSS3. http://www.w3schools.com/css/css3_animations.asp

Take a look at this: http://jsfiddle.net/nys5qzkd/5/

Imagine if we had something like this on the homepage, but instead of photos we had products that the user could click on. It would look very nice on your webprog.

Note: because CSS animations are so new, some browsers need to have special browser specific CSS.
Most browsers will accept `animation` and `@keyframes`
But Chrome, Safari and Opera need to have `-webkit-animation` and `@-webkit-keyframes`

You should also investigate CSS Transitions.

Box sizing

When you specify a width in CSS, it does not include the margins and borders.
So a div with a width of 200px and a margin of 10px ends up taking 220px of space.

This can cause all kinds of issues when using percentages.

To solve this issue you can use the box-sizing CSS property.

Other Notes

If you're concerned that other students may try to copy your assignment (remember that your site is open to the public internet) what you can do is place an empty file named index.html in your public_webprog/IIT/ directory

Then rename you Lab08 directory to something that only you know.

So now the only way to access your assignment is if you know the URL.

Detecting if the user already has a product in their cart. [NEW!]

Some of you may have noticed then when the user clicks on the 'Add To Cart' button nothing happens.
But something is happening, it's just too fast to see!

When you click the 'Add To Cart' button the user gets redirected to AddToCart.php which adds the product to the ShoppingCart cookie. AddToCart.php then redirects the user back to ViewProduct.php instantly.

This happens so fast that it appears that nothing happened, so we need to provide the user some feedback!

What we can do is make it so that the 'Add To Cart' button disappears if the product is already in the users cart.
This also prevents the same product being added multiple times by mistake (people might double click the button)

There is a few ways to do this but this might be a good opportunity to use a user defined function.

Let's create a function called `DoesCartContainProduct`() which takes one parameter: the ProductID to look for.
Our function will return true if the cart contains the specified product and false if it does not.

Step 1

Open **functions.php** and add a new function definition to the bottom of the file like so:

```
function DoesCartContainProduct($ProductID)
{
    // TODO code here
}
```

Notice that our function takes one variable as the input, this is known as the parameter (or argument)

Step 2

Inside out function we need to check if the user even has a shopping cart.
If the user does not have a shopping cart, return false.

Inside our new function place this code:

```
if(isset($_COOKIE['ShoppingCart']))
{
    // user has a shopping cart
    // TODO more code here
}
else
{
    //user has no shopping cart.
    return false;
}
```

Step 3

Ok so now we need to see if the users shopping cart contains the ProductID.
I have provided a handy function called stringContains(), which takes two parameters: a $haystack and a $needle. stringContains() will return true if it finds the needle in the haystack and returns false is it cannot be found.

We can use this function because the shopping cart cookie is a string!

Inside the IF statement from step 2 (where the TODO is) place this code:

```php
if(stringContains($_COOKIE['ShoppingCart'], $ProductID))
{
    return true;
}
else
{
    return false;
}
```

Step 4

Congratulations you have implemented you first PHP function!

Now we need to put it to good use. Open your **ViewProduct.php** and place this code inside the file.

```php
// if the user has this item in their shopping cart
if(DoesCartContainProduct($ProductID))
{
    // don't show the 'Add To Cart' button.
    // and instead echo a message like "you have this item in your cart".
}
else
{
    // echo the 'add to cart' button (and its <form> tag)
    echo "<form action = 'AddToCart.php?ProductID=$ProductID' method = 'POST'>";
    //TODO echo the submit button here.
    echo "</form>";
}
```