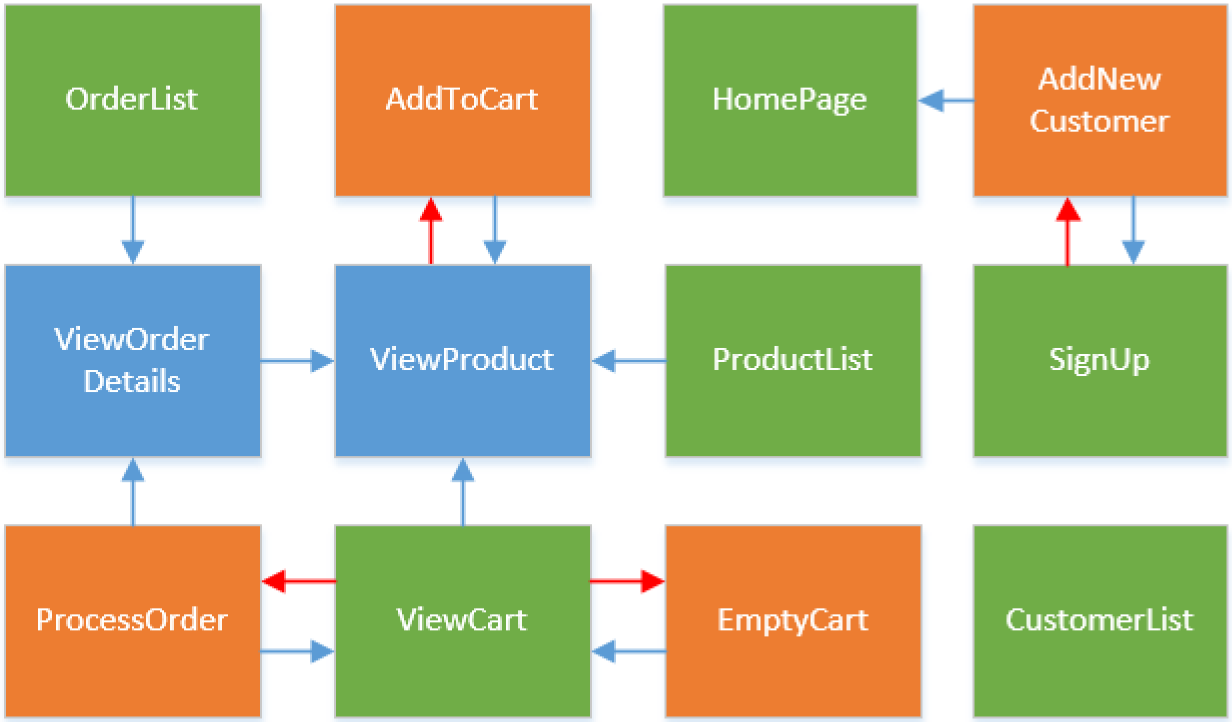
IIT Web Assignment File Details

This file describes the details of the PHP files that your online shop system will use.
You have also been provided with a diagram showing how the user will be directed around the different pages.



The red arrows represent POST requests and blue arrows represent GET requests.

The files marked in green will be visible on the navigation bar.

The files marked in orange are special technical files that the user never sees. **These files do not contain any HTML**.
Instead they perform a special task and depending in that task they redirect the user to another page.
Orange files should always be accessed by POST requests only because they modify the state of the website.

The files marked in blue will be accessed using GET requests, and also require URL parameters.

The blue and green files should have the navigation bar visible at all times.
Unless there is a system error.

I recommend completing the orange files last.

If you have not already done so, please login to phpLiteAdmin and have a look at the data in all the tables
this will help you get a better idea of what data you have to work with.

Message Cookies

Sometimes we want to display a message to the user but we don’t want to display it on the current page.

Let's consider EmptyCart.php, this file has one task, empty the users shopping cart (deleting a cookie)
When EmptyCart.php has completed its task with no errors it will redirect the user back to ViewCart.php

But we should always provide the user some feedback saying that the operation was successful.
So we should display a message like "Your cart has been emptied"

The issue here is if we just echo a message on the page and then redirect the user, the user will have no time to read the message!

What we really need to do is display the message on ViewCart.php.

Problem: How can EmptyCart.php display a message on ViewCart.php?
Solution: Cookies.

What EmptyCart.php can do is set a cookie that contains the message we want to display, then redirect the user to ViewCart.php.
ViewCart.php then looks to see if there is a message cookie and if so display it on the page then deletes the cookie.

In this situation we would consider EmptyCart.php to be the message sender and ViewCart.php to be the message receiver.

There are a number of pages that use Message Cookies, these have already been implemented for you.
Using the functions getCookieMessage() and setCookieMessage() (located in functions.php).

You must style the cookie messages so that they are noticed by the user (Error messages should be red for example)
Although not all the cookie message are errors so make sure you style them appropriately.

I would recommend using a <div> or <span> for this, and maybe have a CSS class called errorMessage or something.

NOTE: getCookieMessage() will return an empty string if there was no messages. getCookieMessage() will also perform htmlspecialchars() on the messages so you can safely echo it to the page.

ProductList.php

Allows a customer to browse the items that our website has for sale.
This page will also provide a search functionality that allows potential customers to search the database for products they are interested in.

The output is limited to 10 items per page. The user should have a hyperlink to get to the next page and a hyperlink to get to the previous page.

The products should be sorted by popularity (that is, how many orders does each product appear in).

SQL Statements Required:

1. One SQL SELECT statement to select the relevant products.

ViewProduct.php

Shows the details of a given product including a picture, description and price.
The ProductID will be provided by a URL parameter.

Also shows the details of the brand, including a brand logo and a hyperlink to the brands website.

This page will also provide a button that says 'Add to Cart'. When the user clicks this button they will be taken to AddToCart.php which will then add the product to the users shopping cart. AddToCart.php will then send them back to this page.

Check to see if the user has this product in their shopping cart, if so don't display the 'Add to Cart' button.

SQL Statements Required:

1. One SQL SELECT statement to select the given product.

CustomerList.php

Allows the business owners to view all the customers in the database.

For every customer in the database list ALL their details including CustomerID, name, address etc.

This data should be displayed using a <table>

This file was not provided to you, you must make it yourself.

SQL Statements Required:

1. A SELECT statement to select all the Customer details in the database.

OrderList.php

Allows the business owners to view all the orders in the database and the customer who placed the order.

This must include the details of the order (OrderID and Time placed) and ALL the details of the customer who placed the order.

The OrderID should function as a hyperlink to ViewOrderDetails.php

This file was not provided to you, you must make it yourself.

SQL Statements Required:

1. A SELECT statement to select all the Order and Customer details for all Orders in the database.

ViewOrderDetails.php

This file is like a receipt, it shows the details of a given order and details about the customer who placed it.
Users will be sent to this page when they complete an order.

The OrderID will be provided by a URL parameter.

This page must include the details of the order (OrderID and Time placed) and the details of the customer who placed the order.

Also must display the details of each product that was in the order (description, brand, price, quantity ordered etc.). There can be multiple products in one order.

The product picture should function as a hyperlink to ViewProduct.php

Don’t forget to have an image of the product and brand for each product in the order.

This page should also show the total price of the order (make sure you factor in Quantity)

SQL Statements Required:

1. A SELECT statement to select all the information about a particular Order including information about the customer who placed the order.
2. A SELECT statement to select the details of the products in the given order.

ViewCart.php

Allows the user to view the items in the shopping cart (a cookie).

Must also display the product and brand details for each product in the cart including pictures of both the brand and product.

This page must also calculate the total cost of the items in the cart.

Please note that this page will have two separate <form> tags each with its own submit button,
one to confirm the order and one to empty the shopping cart.
NOTE: that both of these <form>s should only appear if the user has items in the shopping cart.

If the user clicks the "Confirm Order" button the user will be sent to ProcessOrder.php with some POST data.

If the user clicks the "Empty Cart" button the user will be sent to EmptyCart.php with some POST data.

SQL Statements Required:

1. A SELECT statement to select the details of the products in the users shopping cart

ProcessOrder.php

This file will process the customer's order (add it into the database) and will never be seen by a customer.

If the order is a success we will redirect the user to ViewOrder.php and say that the order was a success using a cookie message.

If the order fails (bad username provided) we will redirect them back to ViewCart.php and display an error message using a cookie message.

You can assume that the user will only be buying one of each item (Quantity is always 1)

If there were no items in the cart, echo a system error.

SQL Statements Required:

1. A SELECT statement to select the CustomerID of a given Username.
2. An INSERT INTO statement to add a new Order to the Order table.
3. An INSERT INTO statement to add the products in the cart to the OrderProducts table.

EmptyCart.php

This file will empty a user's shopping cart (deleting the shopping cart cookie)
Then redirect the user back to ViewCart.php with a message saying that the cart has been emptied.

This page will be accessed by a POST request from ViewCart.php.
This POST request will contain the name of the button that the user clicked on "EmptyCartButton"
the shopping cart should only be deleted if this variable has been set.

SQL Statements Required:

[None]

AddToCart.php

When the user wants to add an item to the shopping cart they will be sent to this file from ViewProduct.php

This file will add the item to the users shopping cart and redirect them back to ViewProduct.php.
This file all also give the user a cookie that contains the shopping cart info.

This file will not be visible to the user, they will be instantly redirected back to ViewProduct.php.

SQL Statements Required:

[None]

SignUp.php

This page will allow new users to register to the website so that they can purchase products.

This file will need to have input text fields so that a user can sign up to the shop.
Including: UserName, FirstName, Last Name, Address, and City.
All of which are required.

When the user clicks submit they will be taken to AddNewCustomer.php

AddNewCustomer.php may send the user back to this page if the username was not available.

This page must display any message cookies.

SQL Statements Required:

[None]

AddNewCusomer.php

This file will not be visible to the user.

When the user clicks the submit button on SignUp.php they will be sent to this page with some POST data.

This page needs to process the data and store it into the database, then if successful send the user to the homepage with a welcome message cookie.

Usernames should be unique so we need to check to see if the chosen username is available.

If the customer's chosen username was taken by someone else redirect the user back to SignUp.php with a message cookie saying that the user name was not available.

If the username IS available insert the customer into the database and redirect them to the homepage with a message that says that the registration was a success.

You should also use the trim() function on the input data to remove any whitespace from the start/end of the string.

SQL Statements Required:

- SQL SELECT statement to check if the username is available.
- SQL INSERT INTO statement to add the new customer to the customers table.

Hompage.php

This file is the main starting point for users who visit your website.

I will leave it up to you on what things should go on this page. But you must not leave it blank.
Suggestion: an animation might be a good idea.

The only requirement is that it should have a text box near the top so that users can search.
When the user clicks submit they should be taken to ProductList.php with the appropriate search string.

This page needs to display any message cookies (form AddNewCustomer.php)

You may rename this file to index.php if you choose.