

## Homepage

```
<?php
ini_set("session.save_path", "/home/unn_w16005124/sessionData");
session_start();
?>
<?php
require_once ("functions.php");
echo get_core_tags("Northumbria book Company | Home", "stylesheet");
?>
<body>
    <!--Gets the getRequest JS script for AJAX-->
    <script type="text/javascript" src="getRequest.js"></script>

<?php
    //Gets the login form and title header with the logo and links
    get_login_form();
    echo get_title_header(array("index.php" => "Home",
                                "orderBooksForm.php" => "Order Books",
                                "chooseBookList.php" => "Admin",
                                "credits.php" => "Credits"), "navLinks");
?>
    <div class="search">
        <label>Search for Books:
        <input type="text" id="searchBox" placeholder="Search Books"/>
    </div>

    <div class="indexWrapper">
        <main id="indexMain">
            <header>
                <h1>Homepage</h1>
                <p>Welcome to Northumbria Book Company we sell a range of different
computing books from: PHP, Databases and UNIX.</p>
            </header>

            <!--Some page information.-->
            <p>
                We have a great range of books. To search for books use the search
box on this page and the book information will be retrieved.
                If you want to buy books then click on the Buy Book link or <a
href="orderBooksForm.php"
accesskey="7">click here</a>.
            </p>

            <!--Section for the auto complete box-->
            <section>
                <h2>Searched Book</h2>
                <!--This paragraph will dissapear once a book is selected from the
menu-->
                <p id="overView">Book details will appear if you search and select
a book from the search box.</p>
                <div id="returnedBook">
                    <!--Returned content goes here.-->
                </div>
            </section>
        </main>

        <!--For the special offers AJAX-->
        <aside>
            <section>
                <h2>Special Offers</h2>
                
                <!--This paragraph will be replaced with the offer retrieved. -->
                <p id="offers">If you see this text there has been an error. Check
that javascript is enabled in your web browser.</p>
            </section>
        </aside>
    </div>
</body>
</html>
```

```

<script type="text/javascript">
    //
    'use strict';
    //Add the event listener for when page loads
    window.addEventListener('load', function () {

        //Updates the paragraph with the book offer
        function updateText(newOffer) {
            'use strict';
            document.getElementById('offers').innerHTML = newOffer;
        }

        //Gets the book offer using AJAX getRequest function
        var getOffer = function () {
            getRequest('getOffers.php', updateText);

            //Times the function and call it again after every 5
seconds
            setTimeout(getOffer, 5000);
        };

        //Cal the variable again
        getOffer();
    });
    //]]&gt;
&lt;/script&gt;
&lt;/section&gt;
&lt;/aside&gt;
&lt;/div&gt;
&lt;?php
    echo get_footer("Copyright: Northumbria Book Company 2017. Contact us by E-
mailing: BNC@outlook.com or telephone 00000 123456");
?&gt;

    &lt;!--Scripts for the JQuery autocomplete.--&gt;
    &lt;script type='text/javascript' src='//code.jquery.com/jquery-
2.1.3.js'&gt;&lt;/script&gt;
    &lt;script type="text/javascript" src="//code.jquery.com/ui/1.11.3/jquery-
ui.js"&gt;&lt;/script&gt;
    &lt;script type='text/javascript' src='JQuery/autocomplete.js'&gt;&lt;/script&gt;
&lt;/body&gt;
&lt;!--Closing html tag in function that will be imported--&gt;
&lt;/html&gt;
</pre>
</div>
<div data-bbox="114 619 253 635" data-label="Section-Header">
<h2>Order Book Form</h2>
</div>
<div data-bbox="114 644 839 886" data-label="Text">
<pre>
&lt;?php
    //create and start the session
    ini_set("session.save_path", "/home/unn_w16005124/sessionData");
    session_start();
?&gt;

&lt;?php
    //Gets core tags, login form and title header passing pages as a paramter
    require_once ("functions.php");
    echo get_core_tags("Northumbria book Company | Order Books", "stylesheet");
    echo "\t&lt;body&gt;\n";
    get_login_form();

    echo get_title_header(array("index.php" =&gt; "Home",
                                "orderBooksForm.php" =&gt; "Order Books",
                                "chooseBookList.php" =&gt; "Admin",
                                "credits.php" =&gt; "Credits"), "navLinks");

    echo "&lt;main&gt;";
?&gt;
</pre>
</div>
```

```

<?php
/* This code dynamically generates a web page containing a form designed with the
html required to display one
checkbox for each of the books currently held in the nbc_books database table has
been provided for you in the
assessment section for the module on blackboard. The user can select one or more
books that they are interested in
ordering by clicking the checkboxes.
Use the browser to look at the structure of the html generated by the php code. */

$url = "http://unn-
izgel.newnumyspace.co.uk/KF5002/assessment/orderBooksFormInc.php";
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
$result = curl_exec($curl);
curl_close($curl);
echo $result;
?>

<?php
    echo "</main>";
?>
<!-- Here you need to add Javascript or a link to a script to process the form as
required for the assignment -->
<script type="text/javascript">
    <![CDATA[
        'use strict';
        window.addEventListener('load', function start() {
            var form = document.getElementById("orderForm"); //Gets a reference to
the form
            var totalPrice = 0; //Variable for total
price
            var hasSelectedAProduct = false; //Variable for if a
product is selected for validation.

            //Check the form when submitted
            form.onsubmit = function (e) {
                return checkForm(this);
            };

            //When the submit button changes call the termsAndCondition function
            form.termsChkbox.onChange = termsAndConditionsCheck;

            //Get a referece is devlery type and add the cost if devliery to home
is selected
            var deliveryType = form.querySelectorAll('input[type=radio]');
            for(var n=0; n< deliveryType.length; n++) {
                var price; //price of te devliery for some reason making this 0
produces an error where when you click on
                // collect from store the current price isn't removed
from the total

                //Finds the radio button that is currently checked
                if(deliveryType[n].checked) {
                    price = parseFloat(deliveryType[n].dataset.price);
                    totalPrice += price;

                    form.total.value = totalPrice.toFixed(2);
                }

                //When the devliery type is changed
                deliveryType[n].onChange = function () {
                    //Removes the current price from the total
                    totalPrice -= price;
                    //Update the price variable with a new price
                    price = parseFloat(this.dataset.price);
                    //Add the new price to te total
                    totalPrice += price;

```

```

        //Update the price box using toFixed() for rounding to 2 DP
        form.total.value = totalPrice.toFixed(2);
    }
}

//Get a reference to the product
var product = form.querySelectorAll('input[type=checkbox]');

//Taking off 1 from the total length will stop the checkbox for terms
and conditions being selected
var totalProductLength = product.length - 1;
for(var i=0; i< totalProductLength; i++) {
    product[i].onchange = calculateCost;

    form.total.value = totalPrice.toFixed(2);
}

//Function calculates gets the cost of the products and updates the price
function calculateCost() {
    'use strict';

    //Ternary if. If product selected add to the total price and make
    semected product = true. If unchekced
    //Removed the price and make selected a product = false.
    (this.checked) ? totalPrice += parseFloat(this.dataset.price) :
totalPrice -= parseFloat(this.dataset.price);
    (this.checked) ? hasSelectedAProduct = true : hasSelectedAProduct =
false;

    //Sends the total to the total price box
    form.total.value = totalPrice.toFixed(2);
}

//Hides or displays the company name box depeneding on what is selected
from the select list
form.customerType.onchange = function () {
    var customerTradeBox = document.getElementById('tradeCustDetails');
    if (form.customerType.value === "trd") {
        customerTradeBox.style.visibility = "visible";
    } else {
        customerTradeBox.style.visibility = "hidden";
    }
};

//Function checks that the form is complete
function checkForm(form) {
    'use strict';
    //trim() === "" checks if the input boxes are empty
    if(form.forename.value.trim() === '') {
        alert("Forename is empty");
        theForm.forename.focus();
        return false;
    }
    if (form.surname.value.trim() === '') {
        alert("Surname is empty");
        theForm.surname.focus();
        return false;
    }
    if (form.customerType.value === "trd") {
        if(theForm.companyName.value.trim() === '') {
            alert("Company name is empty");
            theForm.companyName.focus();
            return false;
        }
    }
}

```

```

    }
    if(form.customerType.value === "") {
        alert("You need to select a customer type");
        theForm.customerType.focus();
        return false;
    }
    //Hasn't selected at least 1 book
    if(!hasSelectedAProduct) {
        alert("You need to select at least book");
        return false;
    }
    return true;
}

function termsAndConditionsCheck() {
    'use strict';

    var submitButton = this.form.querySelector('input[type=submit]');

    //If button disabled property will be opposite if the t&c box is
    checked. For example if the t&c box is
    //checked then submitButton.disabled = false.
    submitButton.disabled = !this.checked;

    //Use a ternary if statement to change the colour and style of the
    parent node which is a label when the checkbox
    //is ticked or un-ticked
    this.parentNode.style.color = (this.checked) ? 'black' : 'red';
    this.parentNode.style.fontWeight = (this.checked) ? 'inherit' :
    'bold';
}
}); //End addActionListener
//]]>
</script>
</body>
</html>

```

## Select Book to Edit Page

```

<?php
ini_set("session.save_path", "/home/unn_w16005124/sessionData");
session_start();
?>
<?php
    //gets the function file
    require_once ("functions.php");
    //gets the development environment file
    require_once ("setEnv.php");
    //Import the main core tags
    echo get_core_tags("Northumbria book Company |Choose Book List", "stylesheet");
    echo "\n<body>\n";
    //Gets the login form
    get_login_form();
    // gets the title header
    echo get_title_header(array("index.php" => "Home",
                                "orderBooksForm.php" => "Order Books",
                                "chooseBookList.php" => "Admin",
                                "credits.php" => "Credits"), "navLinks");
?>

    <!--HTML and opening body tag are inside get_title_header() function-->
    <main>
        <header>
            <h1>Select Book to Edit</h1>

            <?php
                //Checks to see if the user is logged on with the user name using the
                check_login function
            </?php>
        </header>
    </main>

```

```

        if (check_login('logged-in') && check_login('username'))
        {
            //User is logged on so they have access to restroicted content
            echo "\t<p>Click on a book title in the list to select one for
editing.</p>\n";
            echo "\t\t</header>\n";
            //Adds title headers to each coloumn
            echo "\t\t<div class='item'>\n
                <span class='bookTitleHeader'>Book Title</span>
                <span class='bookYearHeader'>Book Year</span>
                <span class='catDescHeader'>Category</span>
                <span class='bookPriceHeader'>Price</span>
            </div>\n";

            try {
                //Include the getConnection function for the database connection
                $db = get_connection();

                $sql = "SELECT bookISBN, bookTitle, bookYear, bookPrice, catDesc
                        FROM nbc_books
                        INNER JOIN nbc_category
                        ON nbc_books.catID = nbc_category.catID
                        ORDER BY bookTitle";
                $query = $db->query($sql);

                //Iterate over the results of the query
                while ($rowObj = $query->fetchObject())
                {
                    $bookISBN = $rowObj->bookISBN;
                    $bookTitle = $rowObj->bookTitle;
                    $bookYear = $rowObj->bookYear;
                    $catDesc = $rowObj->catDesc;
                    $bookPrice = $rowObj->bookPrice;

                    //Remove any tags that might of been part of what has been
                    selected.
                    $bookTitle = filter_var($bookTitle, FILTER_SANITIZE_STRING,
                    FILTER_FLAG_NO_ENCODE_QUOTES);
                    $bookYear = filter_var($bookYear, FILTER_SANITIZE_STRING,
                    FILTER_FLAG_NO_ENCODE_QUOTES);
                    $catDesc = filter_var($catDesc, FILTER_SANITIZE_STRING,
                    FILTER_FLAG_NO_ENCODE_QUOTES);
                    $bookPrice = filter_var($bookPrice, FILTER_SANITIZE_STRING,
                    FILTER_FLAG_NO_ENCODE_QUOTES);

                    //Display the books selected
                    echo "\t\t<div class='item'>
                        <span class='bookTitle'>
                            <a href='editBookForm.php?bookISBN=$bookISBN'
                            title='Click on edit this book.'
                            class='bookTitleLinks'>$bookTitle</a>
                        </span>
                        <span class='bookYear'>$bookYear</span>
                        <span class='catDesc'>$catDesc</span>
                        <span class='bookPrice'>$bookPrice</span>
                    </div>\n";
                }
            }
            catch (Exception $e){
                //If enviroment is set to development
                if (defined("DEVELOPMENT") && DEVELOPMENT == true) {
                    echo "<p>Query failed: ".$e->getMessage()."</p>\n";
                }
                else {
                    //If false create an log error
                    echo "<p>Sorry something went wrong. Please try again
later</p>";
                    log_error($e);
                }
            }
        }
    }
}

```

```

    }
}
else {
    //Not logged i so they dont get access to restricted content
    echo "\t<p>You do not have permission to view this page!</p>\n";

    //closing header tag needed since the original closing one will be
    displayed if not logged in
    echo "\t\t</header>\n";
}
?>
</main>
<?php
    echo get_footer("Copyright: Northumbria Book Company 2017. ");
?>
</body>
</html>

```

## Credits

```

<?php
ini_set("session.save_path", "/home/unn_w16005124/sessionData");
session_start();
?>
<?php
    require_once ("functions.php");
    echo get_core_tags("Northumbria book Company |Credits", "stylesheet");

    echo "\n\t<body>\n";

    get_login_form();
    echo get_title_header(array("index.php" => "Home",
                                "orderBooksForm.php" => "Order Books",
                                "chooseBookList.php" => "Admin",
                                "credits.php" => "Credits"), "navLinks");
?>
<main>
    <h1>Credits</h1>
    <p>Student ID: w16005124</p>
    <p>Student Name: Chris Aston</p>
    <p>E-mail: chris.aston@northumbria.ac.uk</p>
    <p>The title logo text was my own graphics however the The book used in the
title logo was obtained by a web search http://www.carechartsuk.co.uk 2014, Book
Reviews - Care Charts UK[online]
    available at http://www.carechartsuk.co.uk/book-reviews [ Accessed 18
October 2017]
    <p>
    <p>The special offers image on the index page obtained by a web search
http://www.chalkevalleystores.co.uk 2015, Promotions - Chalke Valley Stores
[online]
    available at: http://www.chalkevalleystores.co.uk/promotions [accessed
20 October 2017]
    </p>
    <p>
    Autocomplete CSS was used by a web search http://www.jqueryui.com.com
2013, All jQuery UI Downloads | jQuery UI [online]
    available at https://jqueryui.com/download/all/ [accessed 26 November
2017]
    </p>
</main>
<?php
    echo get_footer("Copyright: Northumbria Book Company 2017. All web pages were
produced by Chris Aston w16005124.")
?>
</body>
</html><!--Start HTML tag in function-->

```

## Edit Book Form

```
<?php
ini_set("session.save_path", "/home/unn_w16005124/sessionData");
session_start();
?>
<?php
//Import the files and get the get core tags, login form and title header
require_once ("functions.php");
require_once ("setEnv.php");
echo get_core_tags("Northumbria book Company |Edit Book Form", "stylesheet");

echo "\n<t<body>\n";

get_login_form();
echo get_title_header(array("index.php" => "Home",
                           "orderBooksForm.php" => "Order Books",
                           "chooseBookList.php" => "Admin",
                           "credits.php" => "Credits"), "navLinks");
?>
<main>
<h1>Edit Book Form</h1>
<?php
//Checks if the user is logged in. If so display the rest of the page, if
not it shows an error saying user doesn't have
//permission to view this page.
if (check_login('logged-in') && check_login('username'))
{
    echo "<!--Overview about the form.-->
    <p>Fill out the form completely and click on submit to edit a
current record. There are accesskeys enabled. Starting with
        ALT + 7 for Book title, ALT + 8 for book year etc with category
being ALT +0 and price being ALT + -.</p>
    <p>If there are any errors you will be redirected here and the
error(s) will be displayed
        and the original pre-edited value will be shown instead.</p>";

    //Checks to see if there are any errors found from the editBookScript
    $errors = isset($_SESSION['bookFormErrors']) ?
$_SESSION['bookFormErrors'] : null;
    //Checks to see if there are any errors and if $error is an array
    if (!empty($errors) && is_array($errors))
    {
        foreach($errors as $error) {
            //echo each error to the web browser
            echo "<p id='errorMessages'>$error</p>";
        }

        //Empty the array so errors don't appear when clicking on a new
title
        unset($_SESSION['bookFormErrors']) ;
    }
    try {
        //Gets the bookISBN
        $bookISBN = filter_has_var(INPUT_GET, 'bookISBN')? $_GET['bookISBN']
: null;

        $db = get_connection();

        //SQL to select the rest of the book properties.
        $sql = "SELECT bookISBN, bookTitle, bookYear, bookPrice, catID,
pubID
                FROM nbc_books
                WHERE bookISBN = :bookISBN";

        //Prepare the statement and execute it
        $stmt = $db->prepare($sql);
        $stmt->execute(array(':bookISBN' => $bookISBN));
```



```

$rowObj = $stmt->fetchObject();

//get the data
$bookISBN = $rowObj->bookISBN;
$bookTitle = $rowObj->bookTitle;
$bookYear = $rowObj->bookYear;
$bookPrice = $rowObj->bookPrice;
$catID = $rowObj->catID;
$pubID = $rowObj->pubID;

//Remove any tags that might of been part of what has been selected
by the sql.
$bookTitle = filter_var($bookTitle, FILTER_SANITIZE_STRING,
FILTER_FLAG_NO_ENCODE_QUOTES);
$bookYear = filter_var($bookYear, FILTER_SANITIZE_STRING,
FILTER_FLAG_NO_ENCODE_QUOTES);
$catID = filter_var($catID, FILTER_SANITIZE_STRING,
FILTER_FLAG_NO_ENCODE_QUOTES);
$pubID = filter_var($pubID, FILTER_SANITIZE_STRING,
FILTER_FLAG_NO_ENCODE_QUOTES);
$bookPrice = filter_var($bookPrice, FILTER_SANITIZE_STRING,
FILTER_FLAG_NO_ENCODE_QUOTES);

echo "<form method='get' action='editBookScript.php'>
<fieldset>
<legend>Book Details</legend>
<!--Hidden since we don't want tot display the ISBN
number to the user-->
<input type='hidden' value='$bookISBN' name='bookISBN' />

<label>Book Title:
<input type='text' value='$bookTitle'
name='bookTitle' accesskey='7' />
</label>

<label>Book Year:
<input type='number' value='$bookYear'
name='bookYear' accesskey='8' />
</label>

<label>Book Publisher:
<select name='bookPublisher' accesskey='9'>\n";
    try{
        //SQL to make a list from the DB.
        $pubSQL = "SELECT pubID, pubName
FROM nbc_publisher
WHERE pubID = :pubID";
        $stmt = $db->prepare($pubSQL);
        $stmt->execute(array(':pubID'=>$pubID));
        $rowObj = $stmt->fetchObject();
        $pubID = $rowObj->pubID;
        $pubName = $rowObj->pubName;

        $pubID = filter_var($pubID,
FILTER_SANITIZE_STRING, FILTER_FLAG_NO_ENCODE_QUOTES);
        $pubName = filter_var($pubName,
FILTER_SANITIZE_STRING, FILTER_FLAG_NO_ENCODE_QUOTES);

        echo "<option
value='$pubID'>$pubName</option>\n";

        //Gets rest of the publishers names from
the DB
        $pubSQL2 = "SELECT DISTINCT pubID, pubName
FROM nbc_publisher
ORDER BY pubName";
        $pubQuery2 = $db->query($pubSQL2);

```

```

        while ($rowObj = $pubQuery2->fetchObject())
        {
            $pubID2 = $rowObj->pubID;
            $pubName2 = $rowObj->pubName;

            $pubID2 = filter_var($pubID2,
FILTER_SANITIZE_STRING, FILTER_FLAG_NO_ENCODE_QUOTES);
            $pubName2 = filter_var($pubName2,
FILTER_SANITIZE_STRING, FILTER_FLAG_NO_ENCODE_QUOTES);

            if ($pubID != $pubID2)
            {
                echo "<option
value='$pubID2'>$pubName2</option>\n";
            }
        }
    }
    catch (Exception $e) {
        if (defined("DEVELOPMENT") && DEVELOPMENT ==
true) {
            echo "<p>Query failed: ".$e->
getMessage(). "</p>\n";
        }
        else {
            echo "<p>There was an error please try
again later.</p>";
        }
    }
    //Creates an error log
    log_error($e);
}
echo"</select>
</label>

<label>Book Category:
<select name='bookCategory' accesskey='0'>\n";
    try
    {
        //Gets selected category.
        $catSQL="SELECT catID, catDesc
FROM nbc_category
WHERE catID = :catID";

        $stmt = $db->prepare($catSQL);
        $stmt->execute(array(':catID'=>$catID));
        $rowObj = $stmt->fetchObject();
        $catID = $rowObj->catID;
        $catDesc = $rowObj->catDesc;

        //Filter out any tags that might of been
selected
        $catID = filter_var($catID,
FILTER_SANITIZE_STRING, FILTER_FLAG_NO_ENCODE_QUOTES);
        $catDesc = filter_var($catDesc,
FILTER_SANITIZE_STRING, FILTER_FLAG_NO_ENCODE_QUOTES);

        echo "<option
value='$catID'>$catDesc</option>\n";

        //Gets the rest of the data
        $catSQL2 = "SELECT DISTINCT catID, catDesc
FROM nbc_category
ORDER BY catDesc";
        $catQuery2 = $db->query($catSQL2);

        while ($rowObj = $catQuery2->fetchObject())
        {
            $catID2 = $rowObj->catID;

```

```

$catDesc2 = $rowObj->catDesc;

$catID2 = filter_var($catID2,
FILTER_SANITIZE_STRING, FILTER_FLAG_NO_ENCODE_QUOTES);
$catDesc2 = filter_var($catDesc2,
FILTER_SANITIZE_STRING, FILTER_FLAG_NO_ENCODE_QUOTES);

//if the ID doesn't match to the one
being edited then add it to the option select list
if ($catID != $catID2)
{
    echo "<option
value='$catID2'$>$catDesc2</option>\n";
}
}
}
catch (Exception $e) {
    if (defined("DEVELOPMENT") && DEVELOPMENT ==
true) {
        echo "<p>Query failed: ".$e->
getMessage()."</p>\n";
    }
    else {
        echo "<p>There was an error please try
again later.</p>";

        //Creates an error log
        log_error($e);
    }
}

echo"</select>
</label>

<label>Book Price:
<input type='number' step='0.01' name='bookPrice'
value='$bookPrice' accesskey='-'/>
</label>

</fieldset>
<input type='submit' value='Submit' id='submitButton'
accesskey='+'/>
</form>\n";
}
catch (Exception $e){
    if (defined("DEVELOPMENT") && DEVELOPMENT == true) {
        echo "<p>Query failed: ".$e->getMessage()."</p>\n";
    }
    else {
        echo "<p>There was an error please try again later.</p>";

        //Creates an error log
        log_error($e);
    }
}
}
else
{
    //user not logged on and shouldnt see this page
    echo "<p>You do not have permission to view this page!</p>";
}
?>
</main>
<?php
    echo get_footer("Copyright: Northumbria Book Company 2017");
?>
</body>
</html>

```

## Edit Book Script

```
<?php
//Start the session
ini_set("session.save_path", "/home/unn_w16005124/sessionData");
session_start();
?>
<?php
//Gets the files and create the nav links
require_once ("functions.php");
require_once ("setEnv.php");
echo get_core_tags("Northumbria book Company |Edit Book Submission",
"stylesheet")
?>
<body>
<?php
    get_login_form();

    //gets the title header with the nav links and logo from the functions file
    echo get_title_header(array("index.php" => "Home",
                                "orderBooksForm.php" => "Order Books",
                                "chooseBookList.php" => "Admin",
                                "credits.php" => "Credits"), "navLinks");
?>
<main>
    <h1>Edit Book Submission</h1>
    <?php
        //Checks if logged in.
        if (check_login('logged-in') && check_login('username'))
        {
            //make a list that would e populated by the result of validate_for
            list($input, $errors) = validate_form();
            //If $errors is not null there is errors
            if ($errors) {
                show_errors($errors, $input);
            } else {
                //No errors so process the form
                process_form($input);
            }
        }
        else
        {
            //user not logged in
            echo "<p>You do not have permission to view this page!</p>\n";
        }

        //Function checks the form for blank space and checks all the input is
        legal
        function validate_form()
        {
            //filter_has_var() gets the variable from the arrays
            //trim() removes any blank space
            $input = array();
            $errors = array();

            $input['bookISBN'] = filter_has_var(INPUT_GET, 'bookISBN') ?
$_GET['bookISBN'] : null;

            $input['bookTitle'] = filter_has_var(INPUT_GET, 'bookTitle') ?
$_GET['bookTitle'] : null;
            $input['bookTitle'] = trim($input['bookTitle']);
            $input['bookTitle'] = filter_var($input['bookTitle'],
FILTER_SANITIZE_STRING, FILTER_FLAG_NO_ENCODE_QUOTES);

            $input['bookYear'] = filter_has_var(INPUT_GET, 'bookYear') ?
$_GET['bookYear'] : null;
            $input['bookYear'] = trim($input['bookYear']);
```

```

        $input['bookPublisher'] = filter_has_var(INPUT_GET, 'bookPublisher') ?
$_GET['bookPublisher'] : null;
        $input['bookPublisher'] = trim($input['bookPublisher']);

        $input['bookCategory'] = filter_has_var(INPUT_GET, 'bookCategory') ?
$_GET['bookCategory'] : null;
        $input['bookCategory'] = trim($input['bookCategory']);

        $input['bookPrice'] = filter_has_var(INPUT_GET, 'bookPrice') ?
$_GET['bookPrice'] : null;
        $input['bookPrice'] = trim($input['bookPrice']);

        //create a variable and add to the array from whats in the input array
        $categories = array($input['bookCategory']);
        $publishers = array($input['bookPublisher']);

        //Checks to make sure the data not empty. If empty add to to the errors
array
        if (empty($input['bookISBN']))
        {
            $errors[] = "A book has not been selected.";
        }
        if (empty($input['bookTitle']))
        {
            $errors[] = "The Book Title is empty";
        }
        if (empty($input['bookYear']))
        {
            $errors[] = "The year is empty";
        }
        if (empty($input['bookPublisher']))
        {
            $errors[] = "The book publisher is empty";
        }
        if (empty($input['bookCategory']))
        {
            $errors[] = "The book category is empty";
        }
        if (empty($input['bookPrice']))
        {
            $errors[] = "The price must have .00 on the end or the price isn't
a float";
        }

        //Compares the bookCategory in the input array to the cateogroes array
t make sure they match with each
        if (!in_array($input['bookCategory'], $categories))
        {
            $errors[] = "The category selected doesn't exist";
        }
        if (!in_array($input['bookPublisher'], $publishers))
        {
            $errors[] = "The publisher doesn't exist";
        }

        //Checks the price is a floating point number
        if (!filter_var($input['bookPrice'], FILTER_VALIDATE_FLOAT))
        {
            $errors[] = "The price is not a float";
        }
        if(!filter_var($input['bookYear'], FILTER_VALIDATE_INT))
        {
            $errors[] = "The year should be an int";
        }

        //returns the array
        return array($input, $errors);
    }

    function show_errors($errors, $input)

```

```

    {
        //Put the errors into an array and go back to the page
        $_SESSION['bookFormErrors'] = $errors;
        $bookISBN = $input['bookISBN'];
        header("location:http://unn-
w16005124.newnumyspace.co.uk/_secondYear/assigment/editBookForm.php?bookISBN=$bookI
SBN");
    }

```

```

function process_form($input)
{
    try {
        $bookISBN = $input['bookISBN'];
        $bookTitle = $input['bookTitle'];
        $bookYear = $input['bookYear'];
        $bookPublisher = $input['bookPublisher'];
        $bookCategory = $input['bookCategory'];
        $bookPrice = $input['bookPrice'];
    }

```

```

        //Get the DB connection from the functions file
        $db = get_connection();

```

```

        //SQL to update the book
        $updateSQL = "UPDATE nbc_books SET
            bookTitle = :bookTitle,
            bookYear = :bookYear,
            catID = :bookCategory,
            pubID = :bookPublisher,
            bookPrice = :bookPrice
            WHERE bookISBN = :bookISBN
        ";

```

```

        //prepare and execute the SQL
        $stmt = $db->prepare($updateSQL);
        $stmt->execute(array(':bookTitle' => $bookTitle,
            ':bookYear' => $bookYear,
            ':bookCategory' => $bookCategory,
            ':bookPublisher' => $bookPublisher,
            ':bookPrice' => $bookPrice,
            ':bookISBN' => $bookISBN)
        );

```

```

        $catID = $input['bookCategory'];
        $pubID = $input['bookPublisher'];

```

```

        //SQL to get the catDesc since we don't want to display PK/FK
        $catSQL = "SELECT catDesc FROM nbc_category WHERE catID = :catID";
        $stmt = $db->prepare($catSQL);
        $stmt->execute(array(':catID' => $catID));
        $rowObj = $stmt->fetchObject();

```

```

        $catDesc = $rowObj->catDesc;
        $catDesc = filter_var($catDesc, FILTER_SANITIZE_STRING,
        FILTER_FLAG_NO_ENCODE_QUOTES);

```

```

        //SQL to get the pubName or the invoice will display the ID instead
        $pubSQL = "SELECT pubName FROM nbc_publisher WHERE pubID = :pubID";

```

```

        //Prepare the statement and execute
        $stmt = $db->prepare($pubSQL);
        $stmt->execute(array(':pubID' => $pubID));

```

```

        //fetch the objects and make the data safe
        $rowObj = $stmt->fetchObject();
        $publisher = $rowObj->pubName;
        $publisher = filter_var($publisher, FILTER_SANITIZE_STRING,
        FILTER_FLAG_NO_ENCODE_QUOTES);

```

```

        //Display the invoice to the user
        echo "<p>Update successful. The book you edited now reads: </p>
            <p>Book title: $bookTitle</p>
            <p>The book year is: $bookYear</p>
            <p>The publisher is: $publisher</p>
            <p>The category is: $catDesc</p>
            <p>The price is: $bookPrice</p>
            <p>To go back to the book list to edit click <a
href='chooseBookList.php'>here</a>.</p>
            ";
    }
    catch (Exception $e){
        if (defined("DEVELOPMENT") && DEVELOPMENT == true) {
            echo "<p>Query failed: ".$e->getMessage()."</p>\n";
        }
        else {
            echo "<p>An error occurred, please try again later.</p>";
            log_error($e);
        }
    }
}
?>
</main>
<?php
    echo get_footer("copyright: Northumbria Book Company 2017");
?>
</body>
</html>

```

## Login process

```

<?php
ini_set("session.save_path", "/home/unn_w16005124/sessionData");
session_start();
?>
<?php
    //include the files
    require_once ("functions.php");
    require_once ("setEnv.php");
    echo get_core_tags("Northumbria book Company |Login", "stylesheet");
?>
<body>
<?php
    echo get_title_header(array("index.php" => "Home",
                                "orderBooksForm.php" => "Order Books",
                                "chooseBookList.php" => "Admin",
                                "credits.php" => "Credits"), "navLinks");
?>
    <main>
        <header>
            <h1>Log On</h1>
            <p>Please wait while we try to log you on.</p>
        </header>

        <?php
            //gets the result of the login function. If errors dieplay the errors. I
            //If good then the code in the else will execute
            list($input, $errors) = validate_logon();
            if ($errors) {
                show_errors($errors);
            }
            else {
                //Set the session so the user is logged-in then re-dirceted to the
                choose list form
                set_session('username', $input['username']);
                set_session('logged-in', 'true');
            }
        </?php>
    </main>

```

```

        echo "<!--Go back to previuos page whre the login attempt was.-->
<script type='text/javascript'>
    'use strict';
    window.history.back();
</script>";
    }

    //Checks the form to make sure entered details are valid.
    function validate_logon() {
        $input = array();
        $errors = array();

        $input['username'] = filter_has_var(INPUT_POST, 'username') ?
$_POST['username']:null;
        $input['password'] = filter_has_var(INPUT_POST, 'password') ?
$_POST['password'] : null;

        $input['username'] = trim($input['username']);
        $input['password'] = trim($input['password']);

        if (empty($input['username']))
        {
            $errors[] = "the username is empty";
        }
        if (empty($input['password']))
        {
            $errors[] = "The password is empty";
        }

        try {
            //Get connection
            $db = get_connection();

            //Select the user and their password hash
            $sql = "SELECT username, passwordHash FROM nbc_users WHERE username
= :username";
            $stmt = $db->prepare($sql);
            $stmt->execute(array(':username'=>$input['username']));
            $rowObj = $stmt->fetchObject();
            if ($rowObj)
            {
                $passwordHash = $rowObj->passwordHash;

                //If the password hash doesn't match then add to the error
array
                if (!password_verify($input['password'], $passwordHash))
                {
                    $errors[] = "Incorrect password";
                }
            }
            else
            {
                //If the user clicked on log in with entering anything an
incorrect login details wont be displayed
                //with the username and password empty error message. This kind
of felt pointless to have another
                //message shown at the same time
                if (! empty($input['username']) && ! empty($input['password']))
            {
                $errors[] = "Incorrect Login details";
            }
        }
        catch (Exception $e){
            if (defined("DEVELOPMENT")&& DEVELOPMENT == true) {
                echo "<p>Query failed: ".$e->getMessage()."</p>\n";
            }
        }
    }

```



```

        else {
            echo "<p>Sorry there been an error, try again later.</p>";
            log_error($e);
        }
    }

    //Returns the array
    return array($input, $errors);
}

//Go back to the previous page
function show_errors($errors)
{
    $_SESSION['logInErrors'] = $errors;
    echo "<script type='text/javascript'>
        'use strict';
        window.history.back();
    </script>";
}
?>
</main>

<?php
    echo get_footer("Copyright: Northumbria Book Company 2017");
?>
</body>
</html>

```

## Log out process

```

<?php
    ini_set("session.save_path", "/home/unn_w16005124/sessionData");
    session_start();
?>
<?php
    require_once ("functions.php");
    get_core_tags("Northumbria book Company | logout", "stylesheet")
?>
<body>
<?php
    echo get_title_header(array("index.php" => "Home",
                                "orderBooksForm.php" => "Order Books",
                                "chooseBookList.php" => "Admin",
                                "credits.php" => "Credits"), "navLinks");
?>

    <!--Displays a message to the user-->
    <main>
        <header>
            <h1>Log Out</h1>
            <p>Please wait while we log you out.</p>
        </header>
    </main>

<?php

    //unset the global variable
    $_SESSION = array();

    //Destroy the session
    session_destroy();

    //Go back to the previous page were the log button was clicked on
    echo "<script type='text/javascript'>
        'use strict';
        window.history.back();
    </script>";

```

```

    echo get_footer("Copyright: Northumbria Book 2017.")

?>
</body>
</html>

```

## Set Development Environment

```

<?php
//Defines for a constant that the webpages are in development.
//If true the full exception will be generated on the webpage
//if false a user friendly error message will be displayed and a error_log
generated
define("DEVELOPMENT", true);

```

## Get Request JavaScript

```

function getRequest( url, callbackFunction ) {
    'use strict';
    var httpRequest;

    if (window.XMLHttpRequest) { // For firefox, chrome
        httpRequest = new XMLHttpRequest();
    }
    else if (window.ActiveXObject) { // IE
        try {
            httpRequest = new ActiveXObject("Msxml2.XMLHTTP"); //for IE
        }
        catch (e) {
            try {
                httpRequest = new ActiveXObject("Microsoft.XMLHTTP");
            }
            catch (e) {}
        }
    }

    if (!httpRequest) {
        alert('Giving up :( Cannot create an XMLHTTP instance');
        return false;
    }

    //Called when the state is changed
    httpRequest.onreadystatechange = function() {
        var completed = 4, successful = 200;
        if (httpRequest.readyState == completed) {
            if (httpRequest.status == successful) {
                callbackFunction(httpRequest.responseText);
            } else {
                alert('There was a problem with the request.');
```

## Autocomplete jQuery

```
$(document).ready(function () {
    'use strict';

    //Gets the search box
    $('#searchBox').autocomplete({
        source: 'server/getBook.php', //Source of the books
        minLength: 3, //Min length for the term

        select: function (event, ui) { //Called when a book selected from the list
            console.log(ui);
            $.ajax({
                method: 'get',
                url: 'server/getFullBook.php?bookISBN=' + ui.item.value //Go to the
                php script to get full details
            })
                .done(function (data, status, jqxhr) {
                    console.log(data); //For debugging
                    $('#overview').hide(); //Hide the overview text
                    $('#returnedBook').html(data); //Write the returned text to
                    the index page
                })
            };
    });
});
```

## Get Book for autocomplete

```
<?php
require_once("../functions.php");
require_once("../setEnv.php");

//Gets the value from the autocomplete book since it uses a term
$term = isset($_REQUEST['term']) ? $_REQUEST['term'] : null;

//the search box is empty
if(empty($term)) {
    echo "";
    return;
}

try {
    $dbConn = get_connection();

    //SQL for books to be selected. Selected book title twice so the book would be
    displayed
    //in the box not the bookISBN or anything else
    $sql = "SELECT bookISBN AS 'value', bookTitle AS 'label'
            FROM nbc_books
            WHERE bookTitle LIKE :term";

    $stmt = $dbConn->prepare($sql);

    // term is a wildcard
    $stmt->execute(array(':term' => "%{$term}%"));

    //fetch all returns an array fetch_assoc return with array index by column
    name
    $array = $stmt->fetchAll(PDO::FETCH_ASSOC);

    //Change the header
    header('Content-Type: application/json');

    //Display the data as JSON
```

```

        echo json_encode($array);
    }
    catch (Exception $e) {
        if (defined("DEVELOPMENT") && DEVELOPMENT == true) {
            echo "<p>Query failed: ".$e->getMessage()."</p>\n";
        }
        else {
            echo "<p>Sorry there been an error, try again later.</p>";
            log_error($e);
        }
    }
}

```

## Get Full Book Autocomplete

```

<?php
require_once ("../functions.php"); //Gets the functions
require_once ("../setEnv.php"); //Get the development enviroment

//Gets the ISBN from the URL being called
$bookISBN = filter_has_var(INPUT_GET, 'bookISBN')? $_GET['bookISBN'] : null;

try {
    $dbConn = get_connection();

    //SQL for full book details
    $sql = "SELECT bookISBN, bookTitle, bookYear, bookPrice, catDesc, pubName
            FROM nbc_books
            INNER JOIN nbc_category
            ON nbc_books.catID = nbc_category.catID
            INNER JOIN nbc_publisher
            ON nbc_books.pubID = nbc_publisher.pubID
            WHERE bookISBN = :bookISBN";

    //Prepare the statement and execute it
    $stmt = $dbConn->prepare($sql);
    $stmt->execute(array(':bookISBN' => $bookISBN));

    $rowObj = $stmt->fetchObject();

    //gets the values from the request stream
    $bookISBN = $rowObj->bookISBN;
    $bookTitle = $rowObj->bookTitle;
    $bookYear = $rowObj->bookYear;
    $catDesc = $rowObj->catDesc;
    $publisher = $rowObj->pubName;
    $bookPrice = $rowObj->bookPrice;

    //Remove any tags that might of been part of what has been selected.
    $bookTitle = filter_var($bookTitle, FILTER_SANITIZE_STRING,
        FILTER_FLAG_NO_ENCODE_QUOTES);
    $bookYear = filter_var($bookYear, FILTER_SANITIZE_STRING,
        FILTER_FLAG_NO_ENCODE_QUOTES);
    $catDesc = filter_var($catDesc, FILTER_SANITIZE_STRING,
        FILTER_FLAG_NO_ENCODE_QUOTES);
    $publisher = filter_var($publisher, FILTER_SANITIZE_STRING,
        FILTER_FLAG_NO_ENCODE_QUOTES);
    $bookPrice = filter_var($bookPrice, FILTER_SANITIZE_STRING,
        FILTER_FLAG_NO_ENCODE_QUOTES);

    //Div will be shown to the user with the data on the index page
    echo "<div>
        <p>Book ISBN: $bookISBN</p>
        <p>Book Title: $bookTitle</p>
        <p>Book Year: $bookYear</p>
        <p>Book Price: $bookPrice</p>
        <p>Category: $catDesc</p>
        <p>Publisher: $publisher</p>
    </div>";
}

```

```

        </div>";
    }
    catch (Exception $e) {
        //If envirement is set to development
        if (defined("DEVELOPMENT") && DEVELOPMENT == true) {
            echo "<p>Query failed: ".$e->getMessage(). "</p>\n";
        }
        else {
            echo "<p>Sorry something went wrong. Please try again later</p>";
            error_log($e);
        }
    }
}

```

## Stylesheet

```

/*Index only*/
.indexWrapper {
    /*Flexes the coloumns so they are the same size in the index*/
    display: flex;
    flex-direction: row;
}

/*For the search autocomplete box*/
.search {
    position: absolute;
    width: 35%;
    top: 123px;
    left: 420px;
}

main {
    width: 100%;
    border: thin solid black;
}

#indexMain {
    width: 70%;
}

aside {
    margin-left: 4px;
    width: 30%;
    float: right;
    border: thin solid black;
}

/*For all pages*/
body {
    position: relative;
    font-weight: inherit;
}

html, body {
    margin: 1px;
    padding: 1px;
    border: 1px;
}

/*For the login form placing it top right*/
#loginForm {
    width: 30%;
    float: right;
    margin-right: 4px;
}

/*Make the error messages for the login red*/

```

```

#errorMessageLogin {
    color: red;
    font-size: 14px;
    float: right;
}

/*Position the logout link in the same place as as the login form*/
#logoutLink {
    position: relative;
    float: right;
    font-size: 1.3em;
    top: 103px;
}

#userInput, #passwordInput {
    width: 50%;
}

p {
    /*This changes all text in a paragraph tag.*/
    font-family: arial, sans-serif, vendona;
    /* text size chosen at 16px to make it easy to read for a wider range of
people*/
    font-size: 16px;
    color: #000000;
    /*Padding to cushion the paragraphs from the border lines. */
    padding-left: 4px;
}

h1, h2 {
    font-family: arial, sans-serif, vendona;
    padding-left: 4px;
}

.banner {
    /*This is the bar which contains the navigation links */
    width: 100%;
    top: 30px;
    background-color: #3366ff;
    height: 175px;
    margin: 0 0 4px;
}

ul {
    position: relative;
    top: 20px;
    /* Moves the links onto the bar*/
    font-size: 1.3em;
    list-style-type: none;
    margin: 0;
    padding: 0;
}

li {
    float: left;
    display: block;
}

a.navLinks, a#logoutLink {
    font-family: arial, sans-serif, vendona;
    text-decoration: none;
    padding: 14px 16px;
    color: #000000;
}

```

```

a.navLinks:hover, a#logOutLink:hover {
    color: #ffffff;
}

#titleBarLogo {
    width: 35%;
    position: relative;
    height: 100px;
}

footer {
    border: thin solid black;
    margin-top: 4px;
    position: relative;
    float: left;
    /*This keeps the footer at the bottom of the page*/
    background-color:#3366ff;
    width: 100%;
}

/*Select book list*/
.item {
    display: flex;
    flex-direction: row;
    border-bottom: thin solid black;
}

.bookTitle, .bookTitleHeader {
    font-family: arial, sans-serif, vendona;
    width: 40%;
}

.bookYear, .bookYearHeader {
    font-family: arial, sans-serif, vendona;
    width: 15%;
}

.catDesc, .catDescHeader {
    font-family: arial, sans-serif, vendona;
    width: 30%;
}

.bookPrice, .bookPriceHeader {
    font-family: arial, sans-serif, vendona;
    width: 15%;
}

.bookTitleLinks {
    /*This changes all text in a paragraph tag.*/
    font-family: arial, sans-serif, vendona;
    /* text size chosen at 16px to make it easy to read for a wider range of
people*/
    font-size:16px;
    padding-left:4px;
    /*Padding to cushion the paragraphs from the border lines. */
}

/*Make the coloumn headers bold*/
.bookTitleHeader, .bookYearHeader, .catDescHeader, .bookPriceHeader {
    font-weight: bold;
}

span {
    margin: 4px 2px 4px 4px;
    /*Creates vertical spacing between the books */
    /* Creates horizontal spacing between the coloumns */
}

```

```

@media screen and (max-width: 960px) {
    .bookTitle, .bookYear, .catDesc, .bookPrice, .item {
        display: block;
        width: 96%; /*Not 100% because it causes a vertical scroll bar */
        padding-left: 4px;
        padding-right: 4px;
        text-align: center;
    }
    .bookTitle {
        font-size: 18px;
        font-weight: bold;
    }

    .bookTitleHeader, .bookYearHeader, .catDescHeader, .bookPriceHeader {
        display: none;
    }
}

/*Edit Book form and login form */

label {
    font-family: arial, sans-serif, vendona;
    display: block;
    margin-bottom: 1em;
    text-align: right;
    position: relative;
    padding: 0.25em 80% 0.25em 0;
}

label > input, label > select {
    font-family: arial, sans-serif, vendona;
    position: absolute;
    left: 25%;
    top: 0.25em;
    /*A width 60% was chosen so the book with the longest title could be entirely
    seen. In this case the book
    with the longest title is:
    The Definitive Guide to Project Management: The fast track to getting the job
    done on time and on budget, 2nd edition 2008 */
    width: 60%;
}

/*The select boxes looked a bit off so i had some extra css to make them match the
input boxes it overrides current width*/
label > select {
    width: 60.4%;
    height: 22px; /*Makes the selects have same height as the input boxes */
}

#submitButton {
    font-family: arial, sans-serif, vendona;
    margin: 4px 0 4px 4px; /*Move the submit button away from the main border*/
}

legend {
    font-family: arial, sans-serif, vendona;
}

/*Error message for the edit book form*/
#errorMessages {
    color: red;
}

```



## Functions page

```
<?php
//This function connects to a database using PDO
function get_connection() {
    try {
        $connection = new PDO("mysql:host=localhost;dbname=unn_w16005124",
            "unn_w16005124", "Laptop3299");
        $connection->setAttribute(PDO::ATTR_ERRMODE,
            PDO::ERRMODE_EXCEPTION);
        return $connection;
    } catch (Exception $e) {
        //Create a log file
        log_error($e);
        throw new Exception("Connection error ". $e->getMessage(), 0, $e);
    }
}

//Sets the session for when a user logs in
//key is the string input
//value is input to be inserted into the session array
function set_session($key, $value) {
    // Set key element = value
    $_SESSION[$key] = $value;
    return true;
}

//Returns if the is logged in
function get_session($key) {
    $return = null;
    if (isset($_SESSION[$key])) {
        $return = $_SESSION[$key];
    } //if
    return $return;
}

//Checks if the user is logged in
function check_login($key) {
    if (get_session($key)) {
        return true;
    }
    else {
        return false;
    }
}

//This function creates the "core" title and head tags of a webpage
//PageTitle is the text between the title tags
//Stylesheet is the css sheet, can be null
function get_core_tags($pageTitle, $stylesheet = null) {
    $coreTags = <<<CORETAGS
    <!DOCTYPE html>
    <html lang="en">
    <head>
        <meta charset="UTF-8"/>
        <meta name="viewport" content="width=device-width, initial-scale=1"/>
        <title>$pageTitle</title>
        <link rel="stylesheet" type="text/css" href="$stylesheet.css"/> <!--My
    stylesheet-->
        <link rel="stylesheet" type="text/css" href="jQuery/jquery-ui-1.12.1/jquery-
    ui.css"/> <!--Stylesheet from JQuery website for automcomplete-->
    </head>\n
    CORETAGS;
    $coreTags.="\\n";
    return $coreTags;
}
```

```

//Gets the title bar including the images and web links
function get_title_header( array $links, $class) {
    $webLink = null; //Create a web link variable

    //For each element in the link array iterate through them assign each link a
    value from what called
    foreach ($links as $link => $value) {
        $webLink.="<li><a href='$link' class='$class'>$value</a></li>\n";
    }

    $titleHeader = <<<TITLEHEARE
    <div class="banner">
        <a href='index.php'><img src='titleBarLogo.png' id='titleBarLogo'
alt='Title Logo'/></a>
        <nav>
            <ul>
                <!--Add the web links from the foreach loop.-->
                $webLink
            </ul>
        </nav>
    </div>\n
TITLEHEARE;
    $titleHeader.="\\n";
    return $titleHeader;
}

//Gets the login form
function get_login_form() {
    //checks to make sure the user isn't already logged in
    echo "\t<div class='loginForm'>\n";
    if (!check_login('logged-in') && !check_login('username')) {
        echo "\t\t<form action='loginProcess.php' method='post' id='loginForm'>
        <fieldset>
            <legend>Log In</legend>
            <label>Username:
                <input type='text' name='username' placeholder='Enter Username'
accesskey='5' id='userInput' />
            </label>

            <label>Password:
                <input type='password' name='password' placeholder='Enter
Password' accesskey='6' id='passwordInput' />
            </label>
            <input type='submit' value='log In' />";
        //Gets any log in errors and displays them in the form
        get_login_errors();
        echo "</fieldset>
        </form>\n";
    }
    else {
        //user already logged in so they get the log out link instead of the form.
        echo "\t<a href='logOutProcess.php' id='logOutLink' accesskey='5'>Log
Out</a>\n";
    }
    echo "\t</div>\n";
}

//gets any log in errors and displays then
function get_login_errors() {
    $loginErrors = isset($_SESSION['loginErrors']) ? $_SESSION['loginErrors'] : null;
    if ($loginErrors) {
        foreach ($loginErrors as $error) {
            echo "<p id='errorMessagesLogin'>*$error</p>";
        }
    }
}

```

```

//Makes and returns a footer for a webpage
function get_footer($footerText) {
    $footer = <<<FOOTER
    <footer>
        <p>$footerText</p>
    </footer>
FOOTER;
    $footer .= "\n";
    return $footer;
}

/* A global exception handler that will fire if no catch block is found
 * @param $e
 */
function exceptionHandler ($e) {
    echo "<p><strong>Problem </strong>" . $e->getMessage() .
        " in file " . $e->getFile() .
        ", on line " . $e->getLine() . "</p>";
}

set_exception_handler('exceptionHandler');

//Error handle
function errorHandler ($errno, $errstr, $errfile, $errline) {
    if(!(error_reporting() & $errno)) {
        return;
    }
    throw new ErrorException($errstr, $errno, 0, $errfile, $errline);
}
set_error_handler('errorHandler');

//Creates an error log. param $e is the exception triggered
function log_error($e) {
    $fileHandle = fopen("error_log_file.log", "ab");
    $date = date('D M j G:i:s T Y');
    fwrite($fileHandle, "$date");
    fwrite($fileHandle, $e->getMessage() . PHP_EOL);
    fclose($fileHandle);
}

```