

Textual Description Of Assignment

Name : Gabriel Grimberg

Student Number : C15478448

Module : Web Development 2

List of pages in assignment:

1. **CSS** : Main-Style (Connected to all pages for the style)
2. **HTML** : Main-Page (Start-up page to direct the user)
3. **PHP** : DatabaseConnect (Included in all files to connect to database)
4. **PHP** : LoggedOut (Logging out page)
5. **PHP** : Login (Logging in page)
6. **PHP** : Register (Registration page)
7. **PHP/HTML** : Reservation (Reservation page form only)
8. **PHP** : Reserve (Actual PHP for Reservation results retrieved from Reservation page)
9. **PHP** : Search (Search page form only)
10. **PHP** : SearchResults (Actual PHP for Search results retrieved from Search page)
11. **PHP** : Unreserve (PHP to unreserved a book)

1. Main-Style (CSS)

This page is a very important page that connects to all of the pages, one line of code which is: `<link type="text/css" rel="stylesheet" href="Main-Style.css"/>` is added to every page.

The code is commented showing which code is used for which page as each page has different style to it but not too different. Simple code such as *font-family* to display the words in the type of given font. **Color** to set a colour for a font or a page. **Text-transform** to make all the letters uppercase which I think looks neat in my design. **Padding** to increase the size of a button or a div. **Text-align** to make the text go centre. **Background** to set an image for a page.

2. Main-Page (HTML)

The main page, you can call it the index page, it has the navigator at the top nicely designed with the help of the CSS page, the navigator has the following pages: **Main Page, Search, Reserve, Register and Login**. So, when clicked on the navigator on the page you want it will navigate to it. Note: **The Navigator is on every page as for the requirement states that it should be included in every page.**

At the bottom, it has a nice user friendly navigation to redirect the user to searching, registering or logging in, again when clicked it will redirect to the suitable page.

A nice footer at the bottom which every page contains, it's just for appearance and included as it is a requirement.

3. DatabaseConnect (PHP)

This page is extremely important, it's not a page it's just PHP code to connect to the database to retrieve results or add results to the tables. It connects to the database if no connection is found it will show an error message. It will then connect to where all the tables are if it can't connect then it will show an error. **This PHP code is connected to every page that requires the use of the database, it connects with a simple:**
require('DatabaseConnect.php');

4. LoggedOut (PHP)

This page is a hidden page, basically now shown on the navigator, when a user logs out it will unset the session and destroy it. That's basically it and it will redirect the user if the user clicks the hyperlink at the button called Log In which will allow the user to log in again.

Again, the navigator is included the same as the footer with a nice background image and style.

5. Login (PHP)

This page has the navigator and footer included as explained above, what this page is for is to let the user log in, as mentioned above the database PHP code is included as a require, then we check to see if a **SESSION** has been created, if not then create a new session. If the form is selected grab the input for the username and password. Search the database to see if the username and password exist. **IF** it matches then a new **SESSION** will be created and if it doesn't match then an error message will display.

If the user is already logged in and is on this page, it will display a greeting with a mini navigator to either search for books, go to the main page or log out. It also shows the books that the user has reserved by doing a search in the database at the **Book Reserve** table if a book matches with the username then it will display the book if not then it will display a message that no book has been reserved.

Then there is an unreserved button it's a small text book where you submit to unreserved the book the user has when the submit button is pressed it will interact with the ***Unreserve.php***.

6. Register (PHP)

Navigator and footer included, an HTML form at the bottom to allow the user to input values to register, this form has HTML validation and PHP validation.

Going through the PHP code as usual it includes the database connect, then it will start a session, an If statement is used to check if a user is already logged in, if logged in then it will display a message that the user is logged in and can't create a new account unless the user has logged out.

If the user is not logged in the code flow moves on and checks the values that the user has inputted. ***Mysqli_real_escape_String*** is used for all values to prevent SQL injection. Then we look for errors, first we look if the two passwords match, if they match then move on if not show an error message. Next we look to see if the password characters are bigger than 6, if they are move on if not display an error message. We then check if Telephone and Mobile are numeric if it is no problem moves on if not then display an error message. Final part of error checking is to see if the mobile is 10 characters if it is then move on, if not then display an error message.

If all correct information has been inputted, the values are then added to the database and salting the password, so the password itself won't show in the database but rather a long string would show using a ***.md5***. If the form is submitted into the Database, then it will say account created and show a hyperlink to log in if not then it will display that the username is taken.

7. Reservation (PHP/HTML)

This is basically a HTML page where the user inputs the ISBN code to reserve a book, it will pass the values onto the **Reserve.php** page to do all the error checking and reserve the book. As usual the navigator and header are included.

8. Reserve (PHP)

This is the page where it gets the form values so basically the ISBN code and does some checking and setting. Starting off we check to see if the user entered a value in if not then display an error message and a hyperlink to try again meaning to try and reserve a book again.

We check if the user is logged in, if not display an error message and a hyperlink to the login page to allow the user to log in.

Now the database file is included to connect to the database a Query value is assigned for checking if the book exists and if the book is not reserved by another user also SQL Injection prevention with ***escape_string*** so if the result brings back 0 then it will bring a message letting the user know that either the book is already reserved or the ISBN code inputted is wrong.

If a row is found, then we update the Reserved table and set the book status to Y meaning it is reserved, then show a message that the book has been reserved successfully.

A record of the reservation is made and makes a new row in the Book Reserve table to show the user who has reserved the book, the ISBN of the book and the date the user reserved it.

As usual the footer and navigator are included.

9. Search (PHP)

There are two parts to this, this displays the form to search for the book and when the submit button is pressed it will take the results and go on to the **SearchResults.php** page which will be discussed next.

Starting with this page it includes the navigator and footer as usual, going to the PHP part it connects to the database it searches the category table to display on a select option wheel, it does all the searching and if the tables and row exists then it will create select options from the row.

10. SearchResults (PHP)

This is the part two for the search page, it takes the values from the form and it does the checking. I made a function to get the data, the user is only allowed to search either the Author, Category or Title of the Book. Although this find results will be used later to check if any results have been found.

If more than 1 options are used display an error message with a hyperlink to search again and tell the user only 1 option at a time.

This function will return the results found if any and limit it to 5 per page as requested in the brief.

A function **varTransfer()** is basically passing the results to the next page if the results is more than 5.

Next part is to unset the variables to prevent any errors and complications.

If the user has not entered anything into the form and has pressed submit it will display an error message and a hyperlink to search again.

Now the **PageNumber** variable is created which will be used to see if there is a next page to display information. We need to set this variable to 0. A variable **NumberToAdd** is created which is used to move onto the next page or to go back a page. The page number I displayed at the top to let the user know which page they are on.

Database is included now and from the function above we need to see if anything is found, if nothing is found then display no books have been found with a hyperlink to search again.

If something is found, then we put it into an array and display the results doing a while loop.

A next page and previous page are added at the bottom I use a **sprintf** for this as what it does is basically displays the next remaining tables (books) on the next page

as you can see the ***varTransfer*** function is connected to it which as mentioned above it takes the results and shows them on the next page.

The navigator and footer are as usual included in this page as of every page.

11.Unreserve (PHP)

As mentioned in the login page that unreserved has another part to it, this is the other part where it does the work.

The navigator and footer included as usual.

We do a ***session_start()*** to check for the session, this is included in every page where the user needs to be checked if he or she is logged in. The database file is included as usual too.

A ***Query*** variable is made that will store the updated values, the table BookTable is updated and the reserved is set to 'N' then we go to the BookReserve table and we delete the row where the user has the book booked.

We then give a message that the book was unreserved if the code was correct, the code referring to ISBN that was entered. Two hyperlinks are then added to the bottom to either view your account or log out.