DEVELOPMENT LOG



[**http://titan.dcs.bbk.ac.uk/~anasee01/p1fma/**](http://titan.dcs.bbk.ac.uk/~anasee01/p1fma/)

**Abu Bakar Naseer**

**anasee01**

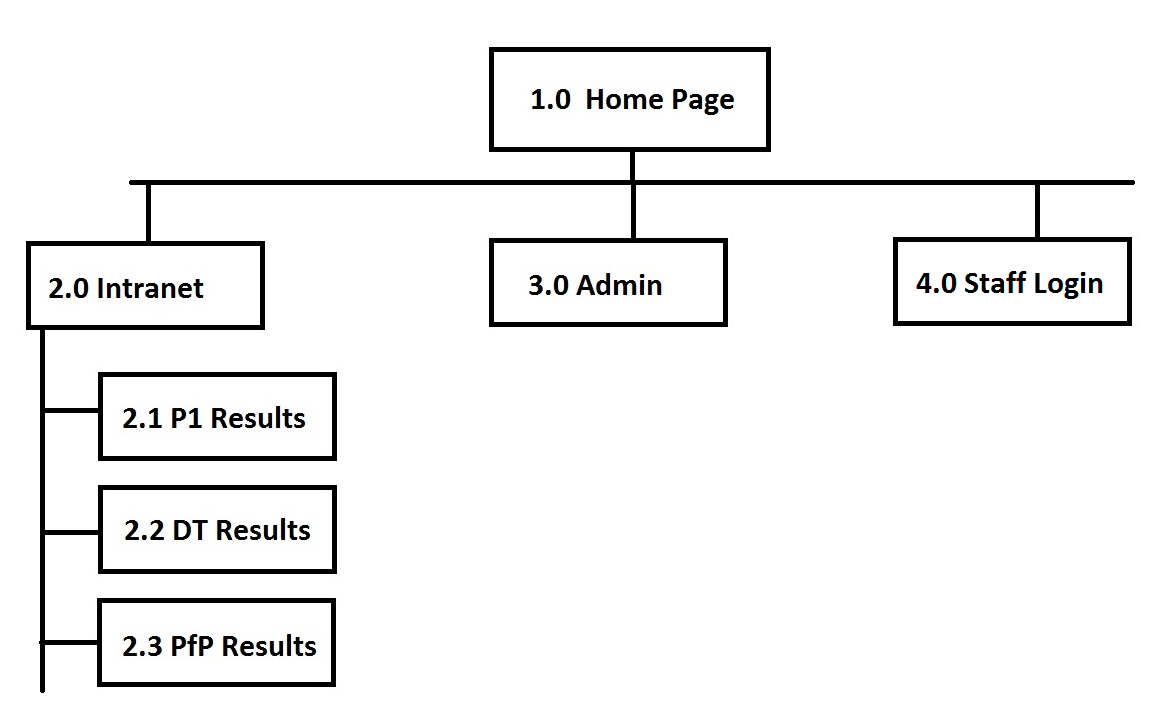
**13139991**

DCS TITAN WIPP is the result of many tens of hours of work over the course of Christmas holidays.

I hope you enjoy playing around with it as much I have had developing it.

**Summary**

Site Map:

****

Notes:

1. The website contains 4 pages

* Homepage
* Intranet
* Admin
* Staff Login

1. The home page can be accessed by any user (logged in or out)
2. The FMA spec was a bit unclear on whether the administrator was to have a username and password similar to a staff member and have the same login page or was it just going to be a master password.

I did not receive a reply on my email to the instructor requesting clarification on this, therefore I have implemented a permissions system which is what made sense to me most. Please see in the next point how this works.

1. The website works on a permission level system where each registered user has a staff-level permissions using which they can access the intranet only and are asked for a password on trying to access the admin page.

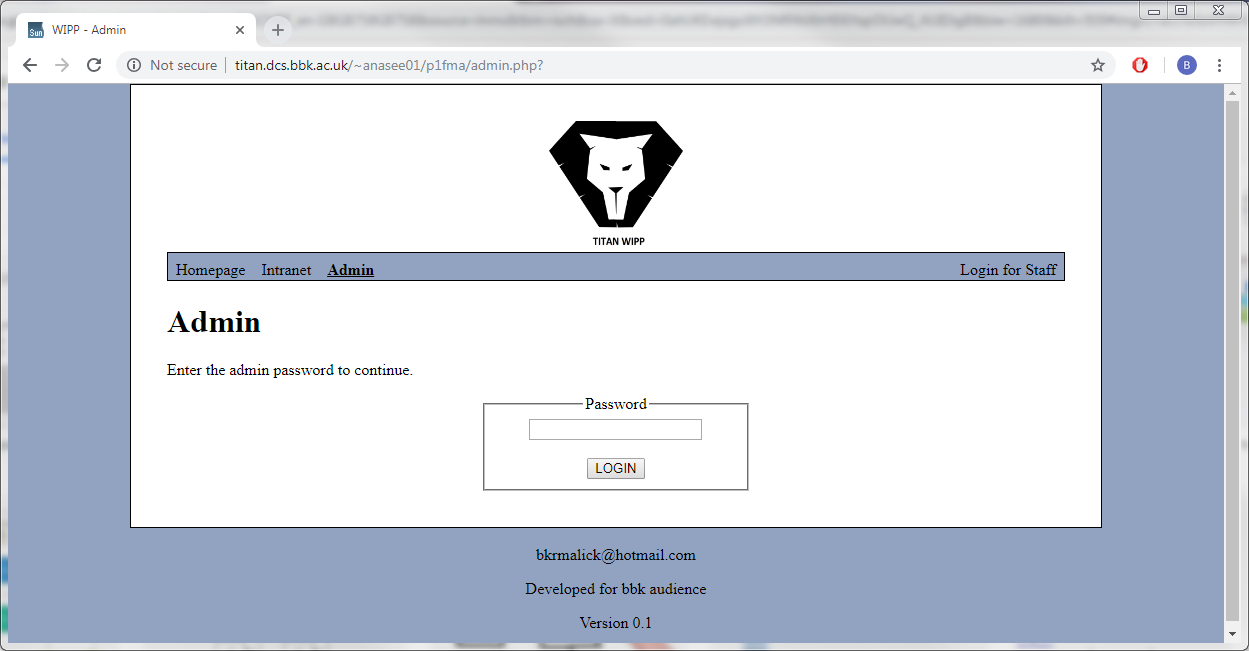
* If a logged-out user tries to access:

The Intranet Page:

They are prompted to log in. On successful login, they can view the results of the modules.

The Admin Page

They are asked to input the admin password.



If input correctly, they are logged in *temporarily* as **Staff Admin** and can proceed with adding a new user.

* If a logged-in user tries to access:

The Intranet Page:

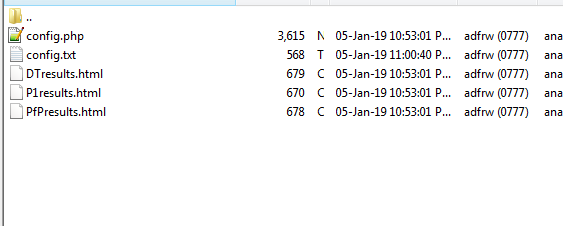
They can view the result pages

The Admin Page

They are asked to input the admin password. On successful input of this, their permissions level is *temporarily* increased to admin and they can proceed with adding a new user.

This permission level of admin will be maintained throughout the current session but once they log back in again, they will still be reverted back to staff permission levels. This is for security purposes.

1. The private directory (not web accessible) contains 5 files:



-The config.txt file contains the raw user data in the form:

**Username**,**Password**,**Title**,**First** **Name**,**Last** **Name**,**Email**

-The config.php file contains functions to open and validate credentials using this file. I thought it was best to hide even the functions in the private directory as if it the php engine crashes, the validation methods would still be safe in the private directory.

-The html files contain the tables that are included in the php pages of the main website directory

-All the files in this directory have 777 permissions and are not web accessible.

6) Titan does not support URL propagated sessions, but I have implemented session ids to all my internal links to ensure the website works regardless of user’s browser settings (which does not work on titan, of course)

7) The username and password for the staff login must be alphanumeric and both are also case sensitive.

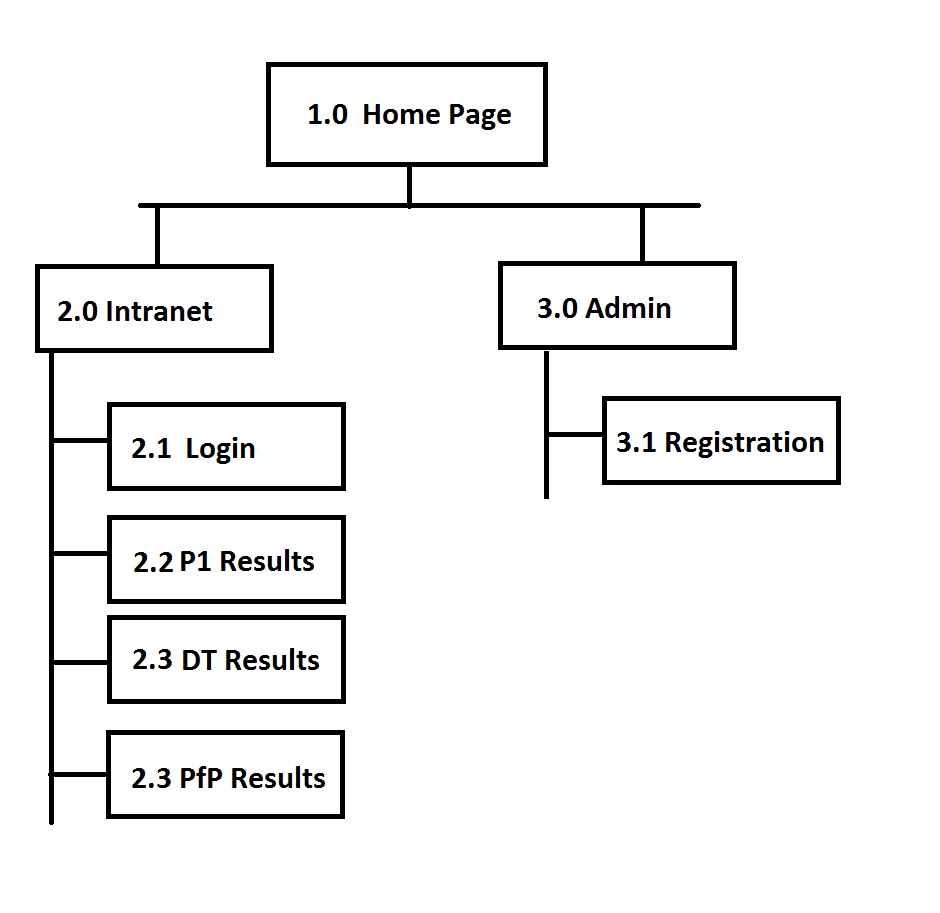
8) I have included comments in the code as much as I could hoping to clarify what each block of code is aims to do.

**Please see from the next page onwards, the complete development log detailing my thought processes and any issues I had while developing this website.**



**25/12/2018**

On the first read through of the spec, following is the first draft of the site map that I’ve come up with:



-The intranet page will have links to rest three pages (if user logged in otherwise redirects to login page)

-The intranet and admin pages will be protected by logins, but my concern is on how to store the passwords. The slides make mention of storing them in a separate private directory that’s not web accessible, but will they still be safe in a text file? What other options are available here? Perhaps in a php file?

-Need to make sure that if a user that’s not logged in inputs the direct link to P1 results web page, they are redirected to the login page.

This can probably be achieved through an if condition that checks if there are any cookies stored on the users pc.

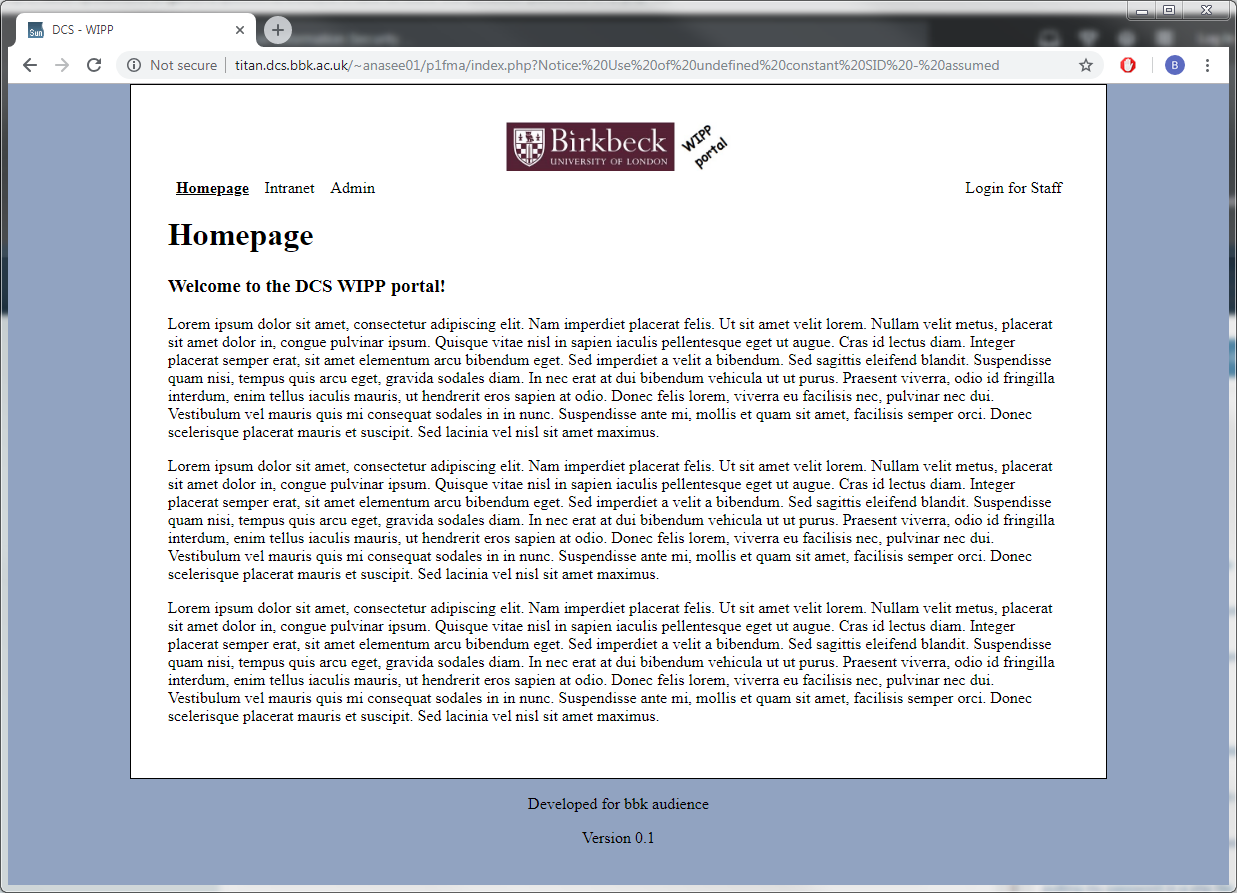
Something perhaps like: **if(count($\_COOKIES))==0) { //Re direct to login page}**

Index.php

-Will need to have a footer and header, same to all other pages. Will need to make separate standalone files in includes folder.

-Will need to have a dynamic navigation bar.

**25/12/2018**



-The CSS and HTML structure seems to be set out after many hours of going back and forward

-Had to go against the idea of maintaining the navigation bar in the included header file as a dynamic navigation bar was needed which requires each page to have a different *active* class attribute and hence navigation bar code cannot be the same for all pages.

-Someone in an online form has mentioned that passwords can be stored in an array in a php file. Which is what I’m going to be doing.

-Currently storing the result pages in the same directory as the index file but this may not be secure. Will need to shift this to private directory at a later point.



-Changed the logo, added the form on the login page

-Passwords and usernames required to be alphanumeric type

-This will be one of two login pages on website. (one for admin, one for staff user)

-Need to restructure the login form now such that:

If(form\_submitted)

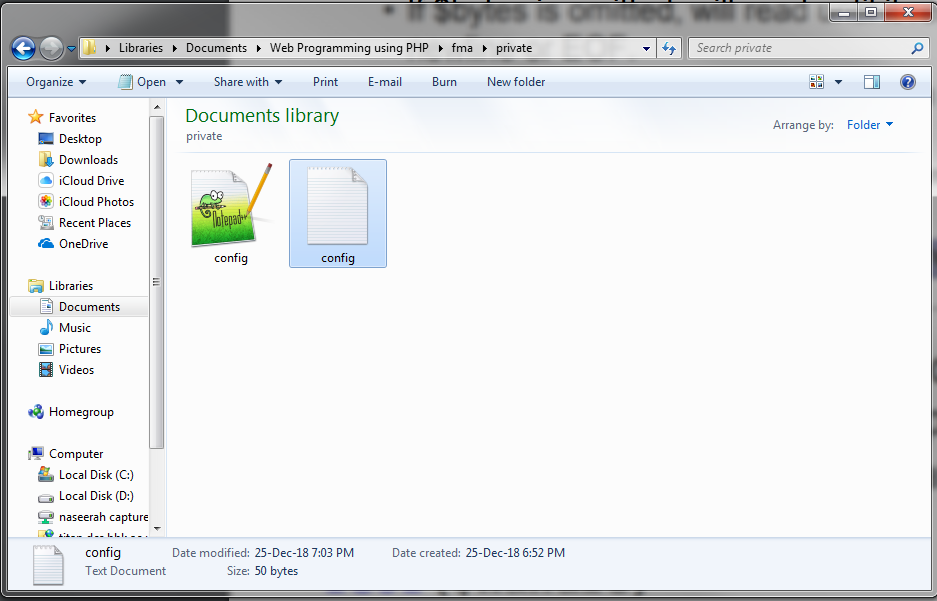
{ //validate }

if(form\_submitted and !errors\_found)

{ //process }

Else { //redisplay form with errors if any}

-The validation function in the first if condition will check if the username and password are valid and will make call to config.php



-I’ve decided to have a private (not web-accessible) php config file and a text file. The text config file stores the credentials and the php config file loads and processes them. I thought it was best to make the php file private too as any the validation checks carried on file, or the reading of the file will reveal the format of the password text file and should not be public as this would be a security concern (susceptible to reverse engineering).

-The config.php contains the following functions:

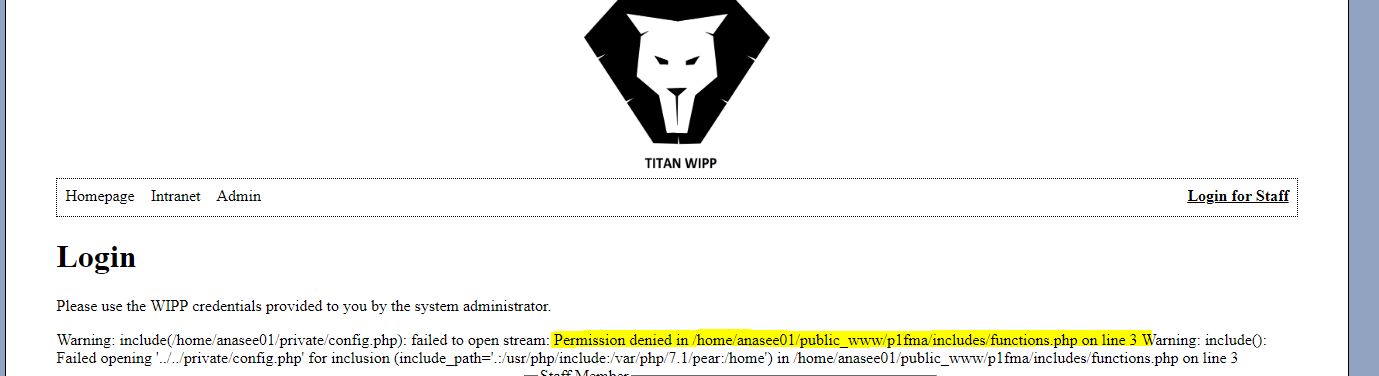
1. checkUser($username,$password) -> Checks credentials of user by loading and reading config.txt
2. addUser($username, $password) -> Adds new user to config.txt file

-The overall structure is this:

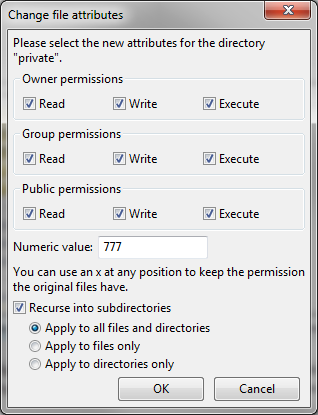
* login.php includes and uses the function validateStaffForm() which calls checkUser() in config.php to validate credentials.
* config.php opens and validates credentials from config.txt

-I thought it was best to use a text file for storing passwords and not in an array in the config file because the credentials are going to be stored long term for which files are more fit for purpose.

(HDD vs RAM)



-The config file is included in functions.php file but it was having trouble opening the file due to a permissions error



-Changed the file permissions on the private folder, and that seems to have fixed it. Not sure if these permissions are secure enough yet.

**26/12/2018**

-I am including the config.php(which is stored in non web accessible private folder) code using paths such as:

Include ‘../../private/config.php’;

-This seems to be working fine and doesn’t seem to be directly accessible if I type in the URL

<http://titan.dcs.bbk.ac.uk/~anasee01/../private/config.txt>

ADMIN

-The CW spec is a bit unclear on if the admin is to have its own username and password or just a master password that can be used to access the admin page.

-I’ve decided to set it up such that:

1. Users can login as staff members at which their permissions level will be maintained as staff
2. When a logged in staff user tries to access the admin page, they are asked for a master admin password.
3. If they successfully enter this, their permissions level is increased to admin. Where they can add new members of staff.
4. If a logged-out user tries to access the admin page, they are asked for the admin password.
5. If they successfully enter this, they are logged in as an “admin” user with admin permissions

-The permissions level will be maintained in session data.

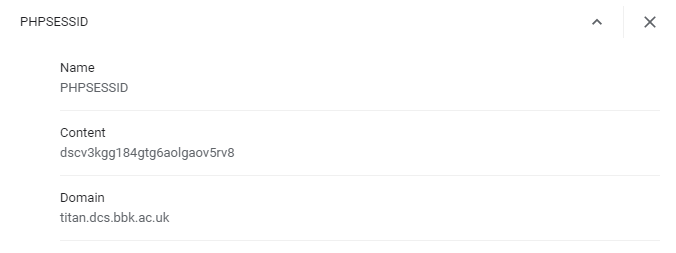
$\_SESSION[‘plevel’] = 1

1. staff
2. admin

-I’ve decided to incorporate a log out page. Which on being opened clears the session data and clears any associated cookies

-There seems to be a few issues on reprinting data of log in page which should be taken from the session data, but it is not. This is probably because of page caching. I’ve implemented the refresh header code at the end of each page and it seems to be a bit better but not fully working.

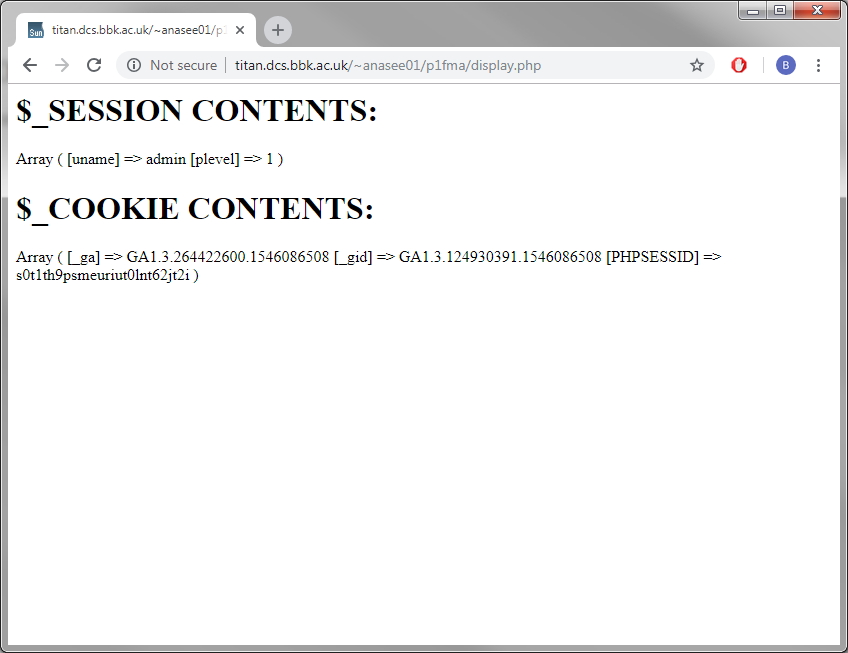
-I’ve narrowed down the problem above to be because my logout.php is not working correctly. It should be deleting all cookies but it is not. There is still a cookie called PHPSESSID that’s being left.



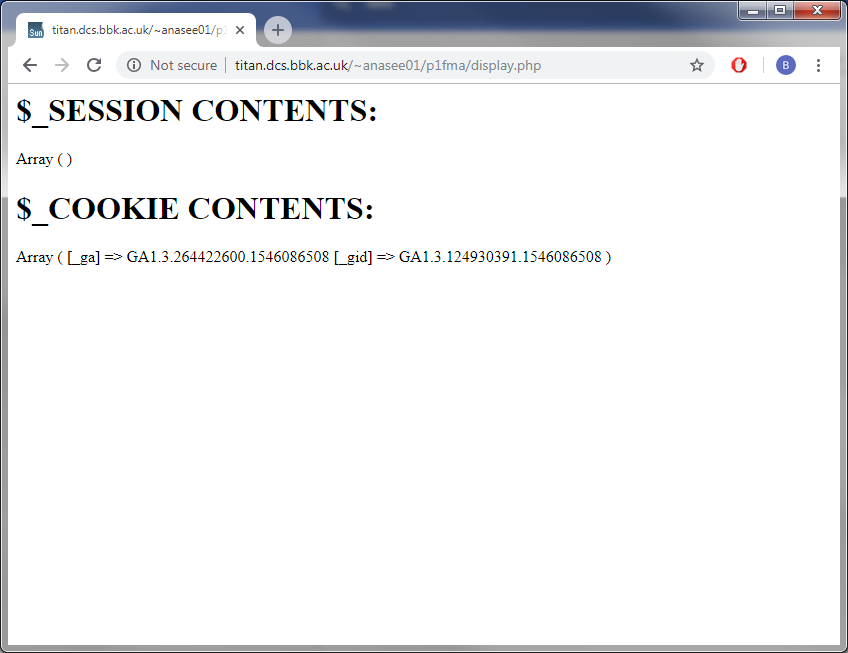
**29/12/2018**

-After hours of experimentation, I’ve discovered that I was using header() and exit() functions in my logout.php script which were causing the code not to execute properly until the page was refreshed twice. I’ve now amended the script to not use these and have added a **display.php** page which shows the contents of the arrays and can now confirm that all cookies and session data is being deleted after logging out:

While logged in:



After logging out:



**01/01/2019**

Optional ideas to include for later:

-random password generator

-w3 validate all pages

-messages when permissions level escalated

**05/01/2019**

-W3 validated all pages

-Included comments in codes

-Shows message now when permissions escalated