INFS3202/7202 Practical 2 – Server Side Scripting using PHP

(9 marks)

You suppose to be working on this practical for two weeks (Weeks 4 and 5). You must present the results to your lab tutor during your scheduled lab sessions in Week 6 that starts 07/04/2014. The Prac could be done either in Lab, or at home. You must also submit the code you have shown to the tutor on the Blackboard by 5pm on the day of your Prac session.

Preparation

Before attempting this practical you should have a good working knowledge of HTML, PHP and server-side programming. Please ensure that you have covered the material in Lectures 1 - 3.

Your personal web-server (zone) has a dedicated directory for both static HTML and PHP code: */var/www/htdocs*

We have prepared a simple php test file that contains only one function *phpinfo()* that you can use for testing your setup. The test file is available on blackboard named test.php. You will need to upload the test file to */var/www/htdocs* in order to test your zone.

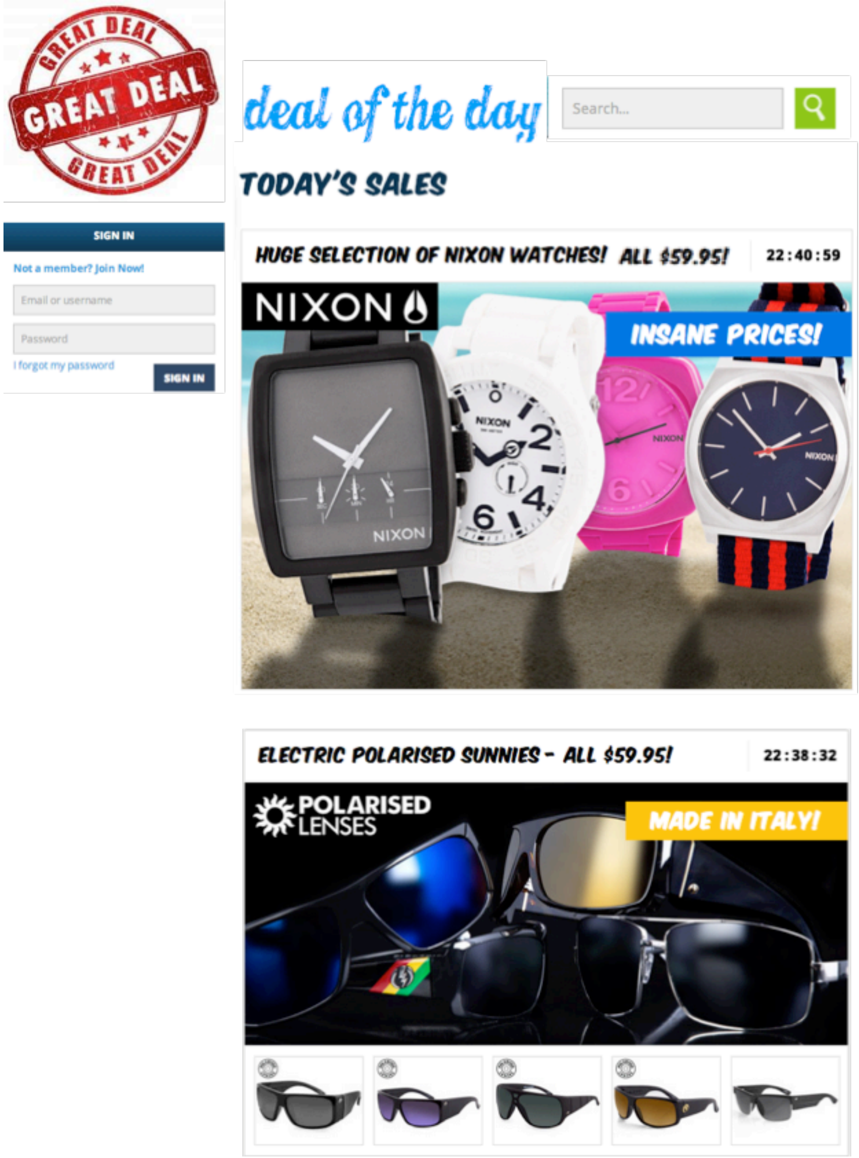
Once you uploaded the test file, you can test you zone by using the following link: *http://infs3202-xxx.uqcloud.net/test.php*

This practical exercise is divided into four tasks:

- Re-writing your Prac 1 static HTML pages in PHP (2 mark);
- Server-side login and logout by using PHP sessions (2.5 marks);
- Session timer, timer and "Stay logged in for" drop-down list (3 marks);
- File Input/Output (1.5 marks).

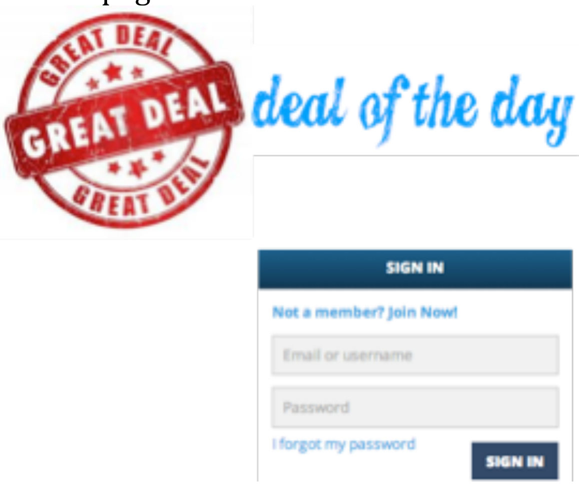
Task 1. Re-writing in PHP (2 marks)

In Prac 1 you have designed few pages of your "MyPlaces" site using HTML. Now you need to re-write your code in PHP. Modify you site and make it looks like an online auction place similar to site following "Deal of the day".

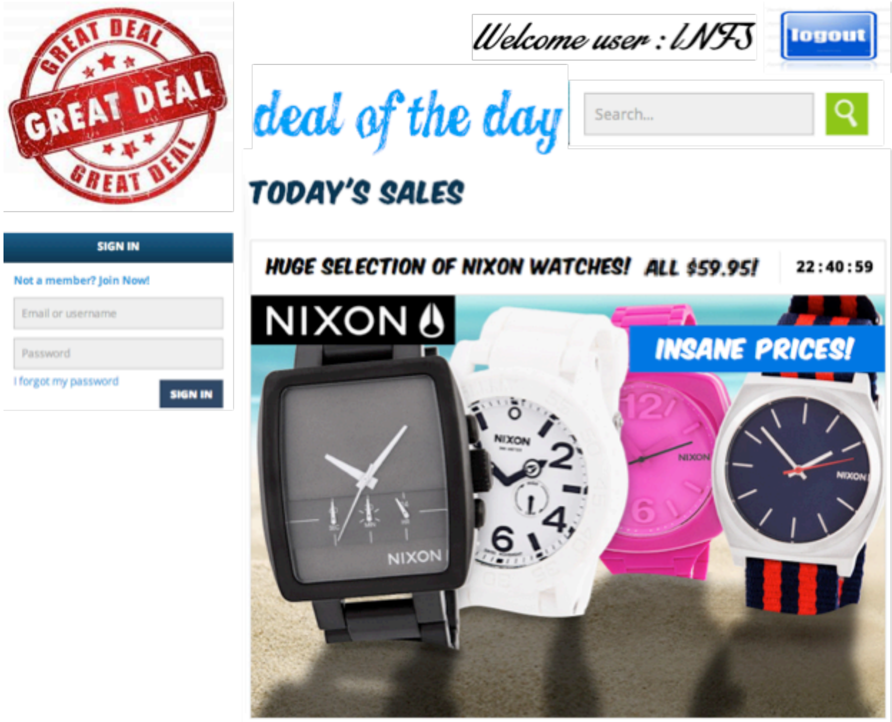


Task 2. Server-side login and logout by using PHP sessions (2.5 marks)

You could create a login PHP page that has any appearance you like, but an example has provided below as referencing. The login algorithm has to be implemented as a server-side with PHP sessions. The main page of your project must have a "SIGN IN" button to call the login form page.



The login form is only required to accept one set of credentials: the username "INFS" or "infs" and password "3202". If a user inputs incorrect credentials, he/she should not become logged in, and the warning "Incorrect username/password" should be displayed on the login page, without using a message box or alert. A successful login should redirect the user to the main "Deal of the day" page.



You do not need to be able to log in with any other credentials, nor do you need to be able to register new credentials. The correct credentials must not be stored in any client-side code or cookies. The browser should always use a POST request to login in this particular exercise.

Successful login should redirect a user to the main page. Once logged in, the "login" button or link placed on the main page should get changed to "logout". When the "Logout" button/link is clicked, the user should be logged out by modifying PHP-session variables and return to the login page. An example is shown below.

If a user accesses the Login page, and he/she is already logged in, the browser should be automatically redirected to the main gallery page.

Task 2 marks:
1.0 mark – Login page
0.5 mark – "login" button on the main page get changed to "logout"
0.5 mark – "Incorrect username/password" warning on the login page
0.5 mark – logout

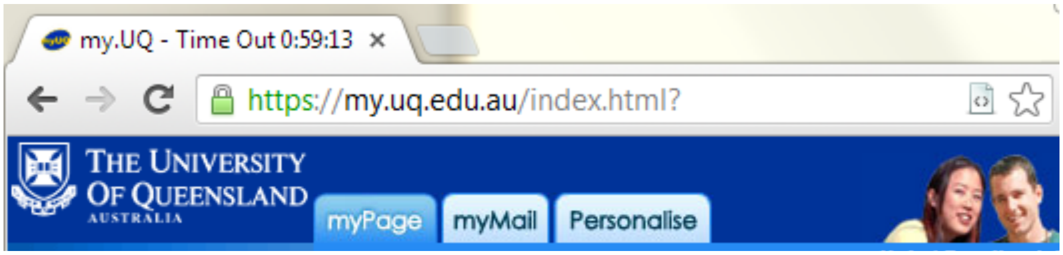
Task 3. Session timer and “Stay logged in” drop-down list

Task 3.1 (1 mark)

In this task, you will implement session time-out and a “Stay logged in” drop-down list, which keeps a user continuously logged-in for a particular period of time, and then automatically closes the session.

First, you should implement a session timer. The timer must count the time of the session at the server site. When the time is up, the session must be automatically closed, and the user gets logged out.

In order for the user to be aware of the time left before automatic logging out, the title of the main gallery page should be modified to include the time remaining. The title is required to update automatically to show the countdown, as on the https://my.uq.edu.au/ web site.



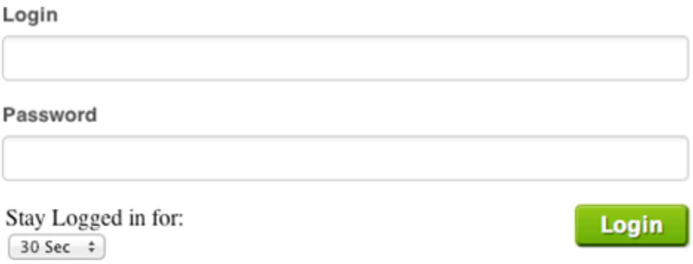
my.UQ - Time Out 0:59:13

https://my.uq.edu.au/index.html?

Task 3.2(1 mark)

To set up the timer delay you should add a “Stay logged in for …” drop-down list to the login page. The drop-down list should have two options: 30 seconds (for assessment) and 1 day.

The login page may look like this:



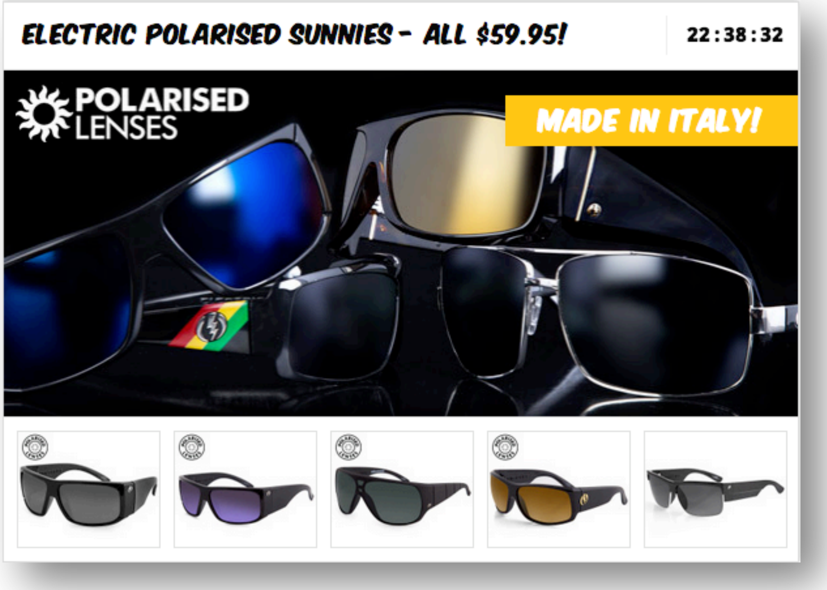
Login

Password

Stay Logged in for: 30 Sec

Task 3.3(1 mark)

Each deal has a separately countdown. Each deal countdown should base on its settle due time similar mechanism on eBay. Therefore each deal`s countdown should appear differently when user login to the main page.



Task 4. File Input/Output (1.5 Marks)

In this task you will be implementing a user activity log-file. Concretely, every time a user logins or logouts (including automatic logouts by timer) a record of the events is added to the file. Each line of the file must contain three portion of information: date/time, user name and user activity (i.e. “login” or “logout”). You may also add some extra info, such as the reason of logging out. The format of data/time representation could be any. For example:

2013-03-10 12:15 INFS Login
2013-03-10 13:43 INFS Logout by timer
2013-03-10 15:21 INFS Login
2013-03-10 15:26 INFS Logout by user

The personal web-server you are using is built on an advanced operating system (Sun OS) with a well-designed set of file access permissions. In particular, for security reason the web-server is not allowed to write anywhere except the temporary directory */tmp*. Thus, in this task you may place your log file in this directory. However, keep in mind that all files from the */tmp* directory are deleted when your virtual server or host server is rebooted.

Additional information.

The directory must be created under your web-server root folder (in our Pracs, it is */var/www/htdocs*). You must change the assess permissions to allow the web-server (and PHP interpreter) writing files into it. From a security point of view, we do not recommend giving “rwx” permissions to everyone. Indeed, the better way is to change the owner of the directory to the web-server’s user.

Hint 1: use ssh access and Unix shell commands to create a directory and set permissions.
Hint 2: the web-server runs under user “www”.
Hint 3: the following console commands might be used in the sub-task:

```
ls
pwd
stat
mkdir
chown
chmod
su
```