

## INFS3202/7202 Practical 3

### Advanced Client-Side scripting

#### Objectives

In this practical we continue client-side programming in Javascript. In particular, we explore how to use cookies, timers and how to check user input. This practical counts 5% towards your assessment. You must present this practical to your lab tutor during your scheduled lab sessions in week 6 that starts 2/04/2012. The prac could be done either in lab, or at home.

This practical exercise is divided into following tasks:

- Client-Side Login (2 marks);
- Using Timers (2 marks);
- Annotation page (1 mark).

#### Preparation

Before attempting this Practical you should have a good working knowledge of HTML & CSS. You should also be familiar with the concept of Web client-side technology, and how JavaScript relates to HTML/XHTML and CSS.

Please Ensure:

- You have covered the material in Lectures 1 - 4.
- You have tried the example JavaScript code given in lectures in Week 4 and 5.
- You have reviewed the practical 2.
- You have a basic understanding of the Javascript timing functions and cookies.

#### Useful links:

JavaScript Cookies	<a href="http://www.w3schools.com/js/js_cookies.asp">http://www.w3schools.com/js/js_cookies.asp</a>
JavaScript timing functions	<a href="http://www.w3schools.com/js/js_timing.asp">http://www.w3schools.com/js/js_timing.asp</a>

#### Task 1: Client-Side Login (2 marks)

Create a login HTML page that has any appearance you like. The login algorithm has to be implemented as a client-side JavaScript. You must use cookies to record that the user is logged in/out.

If the user inputs incorrect credentials, the user should not become logged in, and the phrase "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 gallery page.



Fig. 1. An example of the login page.

On the main gallery page place a button “Login” that calls the login page. When the user is logged in, the button text should change to “Logout”. When the “Logout” button is clicked, the user should be logged out by modifying the cookie. The page is not changed when logging out (HINT: implement a JavaScript function that logs out, and make the logout button call it when clicked rather than changing page).

## Task 2: Using Timers (2 marks)

The goal of this task is to implement an automatic time out for the login created in task 1. A user should be automatically logged out 30 seconds after logging in, if the current page has not been changed. Please notice that such the short timeout is impractical on a normal web site and is used for marking purposes only. The log out should be exactly the same as when a user press the logout button, and should not change the page the user is currently viewing. The timer should reset when the user changes page.

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 number of seconds remaining. When the time is up, the title should be changed to “Time out”. The title is required to update automatically to show the countdown, as on the <https://my.uq.edu.au/> web site.

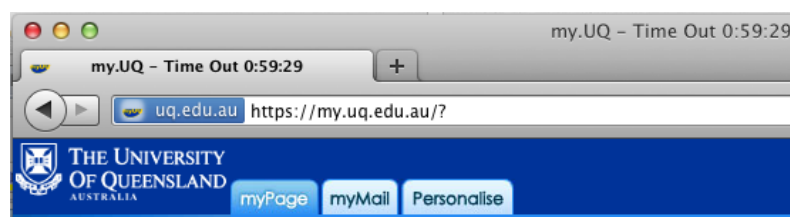


Fig 2

## Task 3: An image annotation page (1 mark)

As required in Prac 1 (Task 1.5), a separate page is shown when a user clicks on an image. This page will be used as an annotation page (Fig 3). In this task students are required to add a form under the image to input image tags. The form must have one text input field and two buttons: “reset” and “submit”. If you have created separate pages for each image in Prac 1, then now you should modify your code in order to use one page to display every image.



New tags:

reset

submit

Fig 3. An example of the annotation page.

Hint: The annotation page must be called from the main gallery page with an argument that specifies the image path and/or image ID. This will be useful later, when you will be required to add code that stores tags in a database. You may use any suitable mechanism, such as GET, POST, cookies, etc.