

## Web Programming using PHP (P1)

### ***Final Marked Assignment (FMA)***

#### **Introduction**

The FMA, which carries 75% of the total marks for this module, requires you to use your knowledge of PHP, form validation and file manipulation to create a prototype web site, which models user registration and log-in system for access to a simple website and password protected intranet pages for the department of Computer Science at Birkbeck College.

The completed FMA deliverables should be submitted electronically in the Assignment Dropbox in Moodle BEFORE the appropriate submission deadline for your class.

#### **Completing the FMA**

You should work on your FMA both during and after class. Begin your work early, as the FMA is a substantial task that requires much planning and effort to complete satisfactorily.

#### **Getting support**

You may email your tutor up to 2 weeks before the FMA deadline with any queries you may have on your FMA. You can request feedback on your work by email. You only have one opportunity for feedback.

#### **FMA Specifications**

##### ***Your Task***

The Department of Computer Science (DCS) at Birkbeck College, have requested that you develop a prototype main page and link this page to prototype secure intranet pages. Your system should include the following aspects:

1. **Main index page**, which should provide some dummy content, as an introduction to DCS, plus the following links:
  - i. *Intranet*: links to a secure page which provides access to module results via a login
  - ii. *Administrator*: links to a registration page/form via admin password, which if successful, allows an administrator to set up new members of staff.
2. **Log-in**: A member of staff should be able to log-in to the intranet by entering a valid username and password in a form.
3. **Log-out**: A logged in member of staff will be able to log-out of the system from any page, and a message will be displayed to this effect returning the user to the index page.
4. **Admin page**: A page for an administrator to register a new member of staff to allow them to login to the intranet. The data required for a member of staff to be registered as a user is:
  - a. Title
  - b. First name
  - c. Surname
  - d. Email
  - e. Username
  - f. Password (The administrator password should be "DCSadmin01").
5. **Secure intranet content pages**: An intranet index page which links to 3 module results pages (data provided in the FMA resources folder in Moodle). These three pages should show module results for the following modules:
  - a. Web Programming using PHP - P1 Results
  - b. Introduction to Database Technology – DT Results

### c. Problem Solving for Programming – PfP Results

6. The user should be able to browse between pages while maintaining their logged-in or out state, regardless of their browser settings. N.B. the module results pages should only be accessible if a member of staff has successfully logged in (including the administrator user), otherwise only the index page should be available.
7. All pages should display a link or form to log-in or out depending on the current user status. If the user is logged in, the page should display the username of the logged in user and if a user logs out they should be redirected to the index page.

A user navigating to any of the public content pages should be able to view the content directly whether they are logged-in or logged-out. The pages should clearly display their user status (apart from the URL links to <https://www.dcs.bbk.ac.uk/>... pages). If a logged out user tries to access the intranet pages, they should be politely requested to log-in.

There is **no specific requirement to apply any CSS or other formatting to your html output**, however, you may do so, if you feel that it adds to the presentation of your solution.

However, the output should use htmlentities and validate correctly under the DOCTYPE you have specified.

### **Working towards a Solution - Some Hints!**

Your solution should be as robust as possible while providing a usable experience for visitors to the site - remember the motto of form submission "Never trust user input". Any security flaws in your final design should be clearly outlined in a learning/development log, which you will need to submit as one of your deliverables. It may also be helpful to carefully consider the following before you begin any coding:

- Consider what data you will need to store, and where you will store it. You are not permitted to use a MySQL (or other) database for this assignment, so you will probably want to store user data in an external file. Consider the security implications of using such a file and appropriate permissions settings.
- Remember the quickest way to write this application is to avoid repeating any PHP or HTML code. Thus, it may help to plan out some useful functions and/or include files that you could write once and then reuse throughout your project.
- You will need to maintain a user's state as they browse between pages. It would, therefore be useful to review the class slides on PHP security and session and cookie management.

### **Deliverables for submission**

You must submit the following deliverables in the Assignment Dropbox in Moodle by the stated FMA submission deadline for your class (replacing *username* with your actual ITS username):

1. A zip file containing all your PHP source files, saved as **username\_p1fma.zip**. Note: the examiners will not be able to mark your work without these PHP source files!
2. A design and learning/development log, saved as **username\_p1fma.doc**. This should contain a site map and pseudocode design and a brief log of any problems you encountered during your work and how you solved them. 3-5 pages of A4.

You are also required to make a copy of your files available on your student web area on the School server. The URL of your page should be:

**<http://titan.dcs.bbk.ac.uk/~username/p1fma/index.php>**

You should also include the full address of this page in your learning/development log.

**Note:** If a required file is not submitted, the examiners will not search for missing files and 0% will be awarded for any missing files.

### **Getting feedback**

Feedback on the marked FMA can be downloaded from Moodle and will normally be returned to you within 6-8 weeks of submission.

### **Backing up files**

Always keep a back-up copy of all work submitted for assessment in case of unforeseen submission problems.

### **Plagiarism**

Plagiarism, which is claiming the work of others as your own, is a serious offence and can result in your exclusion from all colleges of the University of London. You should be aware that we use a range of automated tools to spot potential plagiarism in work submitted for assessment. Providing you clearly reference work done by others that you have included in your FMA you will not be penalised.

### **Criteria for assessment**

You will gain maximum credit for a solution that uses solely your own original code. If you have used any code that is not your own in your solution, or have used other's work as a reference to develop your own code, the source should be fully referenced in code comments and in your learning log (see note about plagiarism above).

The criteria below show the proportion of the marks (out of 100%) that will be awarded for each component of the assignment:

1. **Design and Learning Log (15%):** Documentation of how you designed your solution using a site map and pseudocode. A reflective log of how you developed your solution; the stages you went through, problems you encountered and how you solved them and any reference sources you used.
2. **Functionality (55%):**
  - Log-in Form/Page:** confirms a staff user is legitimate and creates appropriate associations between them and the stored data.
  - Log-out :** logs a staff user out of the system and removes all associated data.
  - Admin Page:** logs an admin user in and gathers and stores new staff user data.
  - Secure intranet content pages/navigation:** access to restricted pages depends on whether a staff user is logged in or not, regardless of browser settings, suitable navigation model.
3. **PHP Coding Style (20%):** clearly commented, use of functions or classes to prevent code repetition, consistent indentation style, "release-quality" code without unnecessary debugging additions.
4. **Publishing/HTML Presentation (10%):** Published on correct URL. Valid HTML to specified doctype, with no CSS or inline formatting.

#### **Note: No marks will be awarded for the following:**

- The use of non-PHP technologies/deprecated PHP functions to achieve the required functionality
- The use of *Regular expressions* in your solution
- Code which suppresses PHP error messages or attempts to alter PHP's configuration in order to suppress error messages.