| BSc Computing (All Pathways)<br>CSY2028 Web programming | | | |
|---|---|---|---|
| Date of Issue: | Tuesday, 28th November 2022 | Date for Submission: | 29th January 2022 23:59:59 |
| This assignment is weighted as 50% of the Module's assessment | | | |
| Assessment Feedback is provided on the rubric via NILE | | | |

**Submission Guidelines – Please read carefully**

1. The University of Northampton's Policy on Plagiarism & Mitigating Circumstances will be strictly implemented.

2. This is not a group project, by submitting this assignment you are asserting that this submission is entirely your own individual work. You may discuss the assignment with other students but any code written should be your own. **Sharing your work with another student, or using code that was written by someone else may be deemed academic misconduct.**

3. If you have used any code that you did not write you must:
   i. Correctly reference the code in the report (use Harvard referencing)
   ii. In your report, clearly document which lines of code you used, where you used them and what they are used for.

4. You must supply **all four** items of assessment and **upload them to the correct submission points**. All work must be uploaded to turnitin
   i. Report word document (uploaded to turnitin)
   ii. Source code (zip file). The marker must be able to download and run your code. *Do not include your video in your zip file*
   iii. Source code word document (uploaded to turnitin)
   iv. Video demonstration
5. You may be asked to attend an in person viva if the marker has any questions about your work. If asked, you are required to attend the viva or you will fail the assignment.

Please make sure you double check all submissions. It is your responsibility as a student to ensure these guidelines have been met.

## Failure to follow the submission guidelines may result result in a capped grade of F-.

# CSY2028 Web Programming

## PHP & MySQL Assignment 2

Aims & Objective

The purpose of this assignment is to assess your ability to modify and improve PHP code on an existing PHP/MySQL driven website.

Brief

You are a developer for a web design agency and you have taken on a new client called *Jo's Jobs.* Jo runs a small recruitment agency and the company's website is used to list available jobs. Applicants can view the available jobs and apply for jobs they are interested in. The website was developed several years ago by Jo's cousin but is now a little outdated and contains bugs that need fixing.

The administration area is available in the /admin/ directory, the password is letmein.

Below is an excerpt from one of Jo's emails regarding the changes she wants you to make:

*1) Can you change the copyright notice? It's not 2017 any more!*

*2) I want to add a new page called "FAQs". Just add the page with some placeholder text that says "FAQs coming soon" and put a link in the menu. I'll send the content over at a later date.*

*3) I can add a new category in the administration area, but the added categories do not appear on the list on the Jobs page or the navigation bar, can you fix this so the customers can view jobs from all the categories we add?*

*4) Sometimes we get requests to re-post old jobs. At the moment, we can delete jobs but it would be useful to be able to archive them so that we can easily repost them later on without having to type in all the information again.*

*5) The job list page in the admin area currently lists all jobs. Can you:*

1. *Add the category name for each job as a column in the table*
2. *Make it so we can filter these jobs by the category they are in. For example, show all HR jobs only.*

*6) Can you make it so that customers can filter the jobs by location? E.g. see all the jobs in Northampton or all the jobs in Milton Keynes?*

*7) I'd like my colleagues to be able to use the website. At the moment we all use the same password. Instead, I'd like to be able to manage user accounts so I can give new staff access (And remove their access when they leave the company). Each member of staff should have their own username and password used to log in to the admin area.*

*8) At the moment we do a lot of the work for our clients (The company who we are advertising the job for). They send us the job details and we post it on the site. Then we manually send them details about the applicants who have applied. I'd like it so that:*

1. *I can set up client user accounts who can then access a restricted version of the admin area*
2. *Clients should then be able to add jobs and see who has applied for their jobs*
3. *It is important that clients can only see details about and edit jobs they have posted.*

*9) Each job currently has a closing date which is the date that we send the applicant details to the clients and the job disappears from the website. It would be good if we could display jobs that are about to reach the closing date on the website. On the home page can you display the 10 jobs which are going to close soonest with the job closing soonest at the top.*

*10) At the moment, the contact page just shows the telephone number and email address. Instead can you add a form that lets someone enter ether name, email address, telephone and enquiry? I'd like it so the enquiry was stored on the website instead of sent via email. In the administration area, the enquires should be listed then we respond to the customer and mark them as "Complete" when done. For our records, I'd like to be able to see a list of previous enquires and which member of staff was the one who dealt with the enquiry.*

As Jo is a new client, you will have this website as part of your portfolio for the foreseeable future. As the new maintainer of the website, you will need to take a look at the code structure and possibly change it to make future additions easier.

**You have the following tasks:**

1. Complete the changes asked for by the client.

2. Rebuild the website in a manner that makes the code easier to maintain. At the moment there is a lot of duplication and several bugs in the way the code is structured.

For a **bare pass** (D-) you must:

1. Complete at least 7 of the 10 changes requested by the client.

2. Identify then fix the major problems with the structure of the website. Amending the website after you have fixed it should never require making the same change in multiple locations.

For a **pass** (D – C+) you must:

1. Complete at least 8 of the 10 changes requested by the client

2. Fix the major problems with the current website and some of the minor ones.

For a **good pass** (B – A+) you must:

1. Complete all 10 changes requested by the client

2. Fix most of the issues on the website and re-develop the website in such a way that further amendments require smaller changes in a single location and sections of code could be reused on a different website without any modification.

Additional information:

- No functionality/features should be lost during your updates (e.g. every page currently has a different title, that should still be the case after you have finished making changes)
- Use the existing website layout, do not completely redesign the site.
- You do not need to amend the HTML or CSS of the website though you may change the HTML and CSS code if you wish, but marks are awarded for PHP code, not HTML and CSS improvemeents.
- You cannot pass the assignment solely by implementing the requested changes. You must also improve the code quality.
- Marks will be lost if implemented features are buggy or break functionality which currently exists.
- You should keep security and maintainability in mind when implementing the changes

**Deliverables**

1. Technical documentation (Word document uploaded to turnitin and included in zip file)
   a) A checklist of the requested changes you have implemented clearly showing which changes you have/haven't completed.
   The checklist should have the following structure. The second row is an example:

| Feature | Completed? | Relevant files | Notes |
|---|---|---|---|
| Admin accounts | Partially | admin/login.php , admin/adduser.php | There is currently no way to delete an account |

   b) Evidence of testing:
      i. Test logs providing information of all the tests carried out (including any failed tests for functionality not implemented).
      ii. Output of PHPUnit along with PHPUnit code coverage reports
   c) References (use Harvard referencing).If you have used code from a book, video, article, etc you must indicate clearly which parts of your code they are and include references.
2. Source code

a) The source code must be well documented with relevant comments. Consistent and clear indentation of the code is also important. You must submit the source code in two forms:
   i. A zip file containing the *websites directory*. Your website folder must contain the up to date `database.sql`
   ii. A word document contain all your PHP code (Copy and paste each file into a single document) for uploading to turnitin
b) Please note the following when uploading your work:
   i. You must include the database.sql with your code
   **ii. If the marker cannot run your work your grade may be capped to F-**
   iii. Do not include your video in the source code zip
   iv. You must only upload code you wish to be marked. Do not include, for example folders called "Assignment2a", "assignment 2 version 3", "as2" and "assignment backup". **It should be clear to the marker which folder you intend to have marked.**
   v. **Please also include your technical documentation word document in the zip file for ease of marking**

3. In Addition to the report you must provide a video demo of your assignment. The demo should be 5-10 minutes (No longer than 15 minutes) and demonstrate all of the new features of the website as if you were showing the client the completed changes that were asked for.

Marking Criteria

This is a programming course, not a design course. As such, marks will be awarded for code quality instead of visual design. You do not need to change the look of the website. However, thought should be placed in how easy to use any changes you made are for non-technical users. Users should never need to write PHP code or change the files to make changes to the website. You should also consider usability, users should never be expected to remember numerical ID numbers, manually amend the address bar or type in information which is already in the system.

The grade for this assignment will form 50% of the overall assignment grade for the module. The rubric below gives an indication of how the marks are split. In general the following criteria will act as a guide to what you should expect:

| | A | B | C | D | F | G |
|---|---|---|---|---|---|---|
| Implementation of requested changes (20%)<br><br>Note: These numbers are *indicative* and marks may be lost if a feature is incomplete, implemented with bugs or in | All 10 requested changes have been implemented and work without bugs and the changes are easy to use. | 8 of the 10 changes have been implemented and work without bugs and the changes are easy to use. | 7 of the 10 changes have been implemented and work without bugs and the changes are easy to use. | 7 of the 10 changes have been implemented but contain bugs, do not work correctly or are done in an unintuitive way. | Fewer than 7 of the 10 changes have been implemented. | No submission or no submission of merit |

| | | | | | | |
|---|---|---|---|---|---|---|
| a way that is difficult to use or makes future modifications more difficult. | | | | | | |
| Testing (15%) | Everything for B and/or use of PHPUnit to run automated tests for form submissions and any functions/classes that have been written. | Tests have been carried out on all completed functionality. There is enough information provided for someone else to replicate each test exactly | Tests have been carried out on all completed functionality | Tests have been carried out on most completed functionality. | Little to no testing, no clear testing strategy | No submission or no submission of merit |
| Code quality, security and efficiency (60%) | Everything for B plus:<br><br>Use of controllers with one function per "page"<br><br>Code is written so that sections can be moved to a different website without modifying a single character in the file<br><br>Use of entity classes | Everything for C plus:<br><br>Use of objects and classes.<br><br>Database abstraction (no hardcoded queries)<br><br>Classes are loaded via an autoloader<br><br>Everything goes through a single entry point<br><br>Classes are in namespaces | Everything for D plus:<br><br>No repeated bootstrap code (e.g. requires statements) at the top of each file<br><br>Separation of concerns: Use of template files with no queries or database logic in templates<br>No separate add/edit pages | Repeated code is in its own files. Making a change should not require making the same change in multiple locations. | F+: No attempt to improve the structure of the code.<br><br>F - F-: Changes implemented have introduced more repeated code than there was at the start. | No submission or no submission of merit |
| Demonstration (5%) | Covers all implemented features in sufficient detail. Any known bugs are highlighted. Validation is tested (e.g. entering invalid values) | Covers all implemented features in sufficient detail. Any known bugs are highlighted. | Covers the functionality from the basic requirements in detail. | Covers the functionality from the basic requirements but needs more detail. | Video does not demonstrate all of the basic requirements. | No submission or no submission of merit |