MANCHESTER METROPOLITAN UNIVERSITY

School of Computing, Mathematics & Digital Technology

ASSIGNMENT COVER SHEET

Unit: Assignment set by:	6G5Z2107: Web Design and Development John Darby
Verified by:	Kris Welsh
Moderated by:	Alan Crispin
Assignment number:	2CWK50
Assignment title:	Coursework
Type: (GROUP/INDIVIDUAL)	Individual
Hand-in format and mechanism:	Submission is online, via Moodle. More information is available in the attached coursework specification for each portfolio element.
Deadline:	As indicated on Moodle.

Learning Outcomes Assessed:

LO1: Deploy client-side JavaScript libraries to add dynamic functionality within a web page
LO2: Use open-source tools and technologies to develop object-oriented and reusable server-side scripts that obtain, validate, process and store user input from web pages
LO3: Integrate client-side and server-side coding into coherent web applications
LO4: Manage communication sessions to authorise users and perpetuate their data across multiple page s

---

It is your responsibility to ensure that your work is complete and available for assessment by the date given on Moodle. If submitting via Moodle, you are advised to check your work after upload; and that all content is accessible. Do not alter after the deadline. You should make at least one full backup copy of your work.

Penalties for late hand-in: see Regulations for Undergraduate Programmes of Study: http://www.mmu.ac.uk/academic/casqe/regulations/assessment.php. The timeliness of submissions is strictly monitored and enforced.

Exceptional Factors affecting your performance: see Regulations for Undergraduate Programmes of Study: http://www.mmu.ac.uk/academic/casqe/regulations/assessment/docs/ug-regs.pdf

Plagiarism: Plagiarism is the unacknowledged representation of another person's work, or use of their ideas, as one's own. MMU takes care to detect plagiarism, employs plagiarism detection software, and imposes severe penalties, as outlined in the Student

Handbook (http://www.mmu.ac.uk/academic/casqe/regulations/docs/policies_regulations.pdf and Regulations for Undergraduate Programmes (http://www.mmu.ac.uk/academic/casqe/regulations/assessment.php). Bad referencing or submitting the wrong assignment may still be treated as plagiarism. If in doubt, seek advice from your tutor.

Assessment Criteria:	Indicated in the attached assignment specification.
Formative Feedback:	A formative submission opportunity is available on the following date: • **8th December 2017** More details about the formative submission opportunity is available in the attached coursework specification.
Summative Feedback format:	Written feedback in the form of a commented mark grid, plus a general comment on the whole submission. General feedback given to all students as a group.
Weighting:	This assignment is weighted at 50% of the total unit assessment.

1. Introduction

This assessment is coursework based, and has a single main component, worth 50% of the overall unit mark. The tasks you are required to complete for these assessments are detailed in this coursework specification.

2. Aim

This unit encourages you to gain practical experience of the world of web development. By the end of the unit, the idea is that you have completed the development of two large-scale web applications, each of which incorporate several technologies – mirroring the sorts of tasks you would be required to carry out in the role of a web developer. These completed web applications can form examples of completed work for future job applications, projects for you to discuss when applying for placement opportunities, or form part of a portfolio for any future creative endeavours.

In particular, the following skills will be essential for successful completion of this coursework:

- Problem solving: You will need to develop solutions for many problems along the way, as you encounter these when developing your solution. You can apply any problem solving techniques you have learnt in your first year to these problems.
- Technical skills: You will need to develop your JavaScript, PHP and MySQL skills in order to complete this assessment.
- Project planning: The assessment requires you to plan, and consider which elements of the work you will attempt in which order. You may find you cannot complete one bit without another, but finding solutions to this (through planning and careful testing) is part of the challenge.

3. Coursework Overview

To complete this assessment task, you must develop a social networking website. The precise details of the coursework are detailed in section **Error! Reference source not found.** below, but the website allows users to register and log in, manage a profile, and post to a "Twitter-like" feed. A special "admin" account allows developers to monitor *who* is using the site. And, finally, you must give your written recommendations for the best way to incorporate new future functionality into the site.

4. The PHP-focused Assessment (2CWK50)

a. Outline

To complete this assessment, you must further the development of a dynamic, server-driven website using PHP, from skeleton code provided for you[1]. Specific details of what you need to add are detailed below. **You must work from the skeleton code you have been provided, and all code you submit must be your own, unaided work.** You must not submit code you have acquired from **any other sources**, including but not limited to, online tutorials or repositories.

Your submission will be marked in the following areas:

- User Login
- User Interaction
- Developer Tools
- Video Sharing
- Code Quality

Please ensure you have examined the mark scheme closely, to ensure that work you are completing is part of the mark scheme.

b. Additional Guidance

User Login

Users should be able to sign up to the site using a unique username (one not already in use), and a password. Users should then be able to log in using sessions (and cookies to store session IDs), and enter profile data which is stored in a database table. This data should be inputted using a simple form and should *at least* include the following details:

- A first name
- A surname
- An email address
- A date of birth
- Number of pets

The form should validate all user input on the client-side, using the rules below, and these validation steps should be repeated on the server-side to prevent any potential abuse:

- Check a first name has been supplied AND that it's not too long for the database table
- Check a surname has been supplied AND that it's not too long for the database table
- Check an email address has been supplied AND that it's a valid email address AND that it's not too long for the database table
- Check a date of birth has been supplied AND that it's a valid date
- Check a number of pets has been supplied AND that it's a valid number of pets AND that it's not too large for the database table

[1] Available on the *Web Design and Development* Moodle area.

As a starting point, refer to:

- Persisting Data IV - Client and Server Validation (see the lab sheet for examples of validating profile data on both the client-side and the server-side)

User Interaction

Users should be able to access a list of all other users, and upon clicking on a username that user's associated profile information should be displayed. (Note: your create_data.php script should create fake profile data for a sensible number of users to allow you to test this function ality[2]) Users should be able to post short messages to a global "feed" which can be seen by all users. Users should be able to "like" posts on the global feed and new posts and new likes should appear dynamically, without the need for a page refresh.

As a starting point, refer to:

- [Optional: Case Study - Robin's Nest (see the code for an example of how to browse user profiles)]
- Persisting Data V – APIs and Data Structures (see the lab sheet for more on building a global feed)

Developer Tools

The site should provide a special "admin" account for developers, logged into using the username: "admin" and the password: "secret". This account should provide developers with the ability to mute abusive feed users. It should also allow developers to analyse *who* is using the site, by providing them with useful summaries of profile data for all players simultaneously (e.g., the ages of all users, the number of pets owned by all users). This information should be provided to developers in graphical form, and you are strongly advised to investigate the use of JavaScript plugins like Google Charts. Finally, developers should be able to filter users, based on their profile data, to identify related subgroups (e.g., those users born after a certain year, those users with a certain number of pets), and target promotional materials to these subgroups in some convenient way (e.g., via mass emails, via users' profile pages, or via "pinned" posts at the top of the global feed).

As a starting point, refer to:

- Persisting Data VI – Images and Dashboards (see the lecture slides and lab sheet for more on graphical summaries and interactive filtering)

Video Sharing

You are not required to do any programming for this component. Instead, you must make a written recommendation for how best to solve a programming challenge.

In the future, the site's owners want to allow users to share videos via the global feed.

The site's owners will take responsibility for the implementation and its specific details. What they require from you is a recommendation about which JavaScript video library would be best for them to use. They want to know which libraries you think are most relevant, the specific criteria you would

[2] Note, therefore, that it is not necessary for you to complete the profile creation functionality from the "User Login" component before working on this shared profile functionality.

use to evaluate and compare them, and your judgements in relation to each of these criteria for each of the relevant libraries. They are as interested in the *way* you compare the available libraries as they are in your final recommendation.

You should include your recommendations as part of the site itself, and a new page for you to extend (called libraries.php) has been included in the skeleton code. You should make use of simple HTML formatting to improve the clarity of your analysis wherever appropriate. E.g., using tables, bullet point lists, images, hyperlinking to relevant materials.

As a starting point, refer to:

- Persisting Data VI – Images and Dashboards (see the lecture slides for more on how to compare JavaScript libraries, including comparison criteria)

Code Quality

Code quality is very important and it's vital that you consider this component right from the start. You should be commenting *all* the new code that you write. It's fine if your files originate from the lab exercises or skeleton code, but the comments from those original files don't count towards your marks here. It's the *new* comments you've written to describe the *new* lines of code you've added to the file that are important.

You should ensure that you are using good practices when writing your code, particularly with reference to indention, the use of curly braces, and semi-colons, etc. As a starting point, you may consider following a style guide[3] to help you format your code in a readable, professional manner. Ensuring that you are structuring your code *consistently* is an essential component to earn good marks here.

You will be rewarded appropriately for use of sensible features of PHP to solve your problems. Appropriate use of variables, arrays, objects and functions will all earn you marks – and misusing these features may be penalised.

You should include a README.txt file as part of your submission. This should include details on how to set up your submission (e.g., "run create_data.php", "navigate to login.php", a username/password to log in with, etc.) and documentation for how your site works. The documentation doesn't need to be very long, but it does need to clearly explain what your site can do, covering all its individual pages.

A useful tip is to break the documentation up into sections for each of the features in the mark scheme (User Login, User Interaction, Developer Tools) and briefly describe how each feature works. Once you have completed the documentation for each feature you have addressed, you should naturally find you have described all of the pages in your site (it is not necessary to document your libraries.php page).

c. Submission

Submission of this coursework will be online, through the university's Virtual Learning Environment (Moodle). You must include all of your code, your README file (see previous section), and a

[3] For example, the Framework Interoperability Group's PHP style guide: https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-2-coding-style-guide.md

create_data.php file that sets up your database, database tables, and testing data (a starting example for you to extend is provided in the skeleton code). When compressing these things into a single file (i.e. a zip file), it is important that the directory structure is preserved.

d. Mark Scheme

Your work in this coursework will be graded in a number of areas, attracting a number of marks for each. Guidance on the assessment criteria for each area is included below. For each area you will be given a mark and these marks will then be combined to give an overall mark for 2CWK50, which is worth 50% of the final unit mark.

User Login (LO4,LO3)		**User Interaction (LO2,LO3)**		**Developer Tools (LO1,LO3)**		**Video Sharing (LO1,LO3)**		**Code Quality**	
No/non-functioning attempt at new account creation.	0-4 marks	No/little attempt at supporting user interaction.	0-4 marks	No/non-functioning attempt at an administrator account.	0-6 marks	No/erroneous analysis of relevant JavaScript libraries.	0-6 marks	Code is **poorly** presented, with **no** documentation or comments.	0-3 marks
New accounts can be created **and** logged into.	5-9 marks	Users can access individual user profiles for **all** other users.	5-9 marks	The administrator can mute **any** abusive feed users.	7-12 marks	Adequate analysis covering **at least one** relevant JavaScript video library.	7-12 marks	Code is **moderately** presented, with **some** documentation, and **occasional** comments.	4-6 marks
Users can additionally enter profile information with **client-side** validation.	10-14 marks	Users can additionally post text (140 char max) to a global feed, accessible by **all** other users.	10-14 marks	The administrator can additionally view **graphical** summaries of profile information for all users simultaneously.	13-18 marks	Precise/rigorous analysis covering **at least two** relevant JavaScript video libraries.	13-18 marks	Code is **well** presented, with **good** documentation and **sensible** comments.	7-10 marks
Profile information is additionally validated on the **server-side**.	15-20 marks	Users can additionally "like" posts, and see a **dynamic** view of the feed (new likes and new posts appear automatically).	15-20 marks	The administrator can additionally filter users based on profile information and send **targeted** information to subgroups (e.g., by email, or via the global feed).	19-25 marks	Meticulous/insightful analysis covering **at least three** relevant JavaScript video libraries.	19-25 marks		

For example, a student earning 20 marks in *User Login*, 20 marks in *User Interaction*, 10 marks each for their *Developer Tools* and *Video Sharing* work, and 5 marks for *Code Quality* would achieve an overall mark of 65% (20 + 20 + 10 + 10 + 5).

5. Support for the Assessment

a. Help! I don't know where to begin or what to do!

Don't panic! This assignment has many starting points, and many routes through the piece of work, which can be scary if you do not know where to start. We introduced important concepts for the assignment in the laboratory exercises, so perhaps you could think about which laboratory exercises you are most comfortable with, and start building your assignment from these.

If in doubt, examine the mark scheme, and try to plan where you think you can earn the marks to get at least a passing grade. If you can pick a starting point (e.g. working on your user logins), and decide you are aiming for 10 marks in this area, you might find you can exceed your original expectations once you have made a start.

You should make good use of the materials provided for you on Moodle. There are numerous resources which you may find useful – particularly those in the *Looking to take it further?* section of each week. Lynda.com videos are an excellent source of information, and a great way to refamiliarise yourself with some of the core concepts of the unit.

It may also be advisable to acquire one or more of the books on the unit reading list, which you can check on Moodle.

b. Opportunities for Formative Feedback

There is an opportunity for formative feedback on your coursework progress at **8th December 2017**. You **must** submit your "work in progress" at this date to receive feedback on your progress so far. An online Moodle submission will be available for you to submit your work – how ever far you have progressed – which also offers a way to practice preparing your work for your final submission.

c. Your Final Feedback

You will receive written feedback on your assignment, in the form of a commented assessment grid identical to that included above, with a short comment on each column, and a general comment covering your piece of work.

d. When, where and how can I get support from the unit tutors?

Assessment support is available by arrangement, and if you find that you are struggling with the assessment, you should contact your tutor **at the earliest possible opportunity** to arrange support.

You can contact your tutor with the following details:

Dr John Darby
John Dalton, E152
j.darby@mmu.ac.uk
0161 247 1542
For my most up-to-date office hours, please check the Moodle area.