MANCHESTER METROPOLITAN UNIVERSITY
Department of Computing and Mathematics
ASSIGNMENT COVER SHEET

Unit:	6G5Z2107: Web Design and Development
Assignment set by:	John Henry
Verified by:	CDTN Assessment Verification Panel
Assignment number:	1CWK50
Assignment title:	The PHP Frameworks Assessment
Type: (GROUP/INDIVIDUAL)	Individual
Hand-in format and mechanism:	Submission is online, via Moodle. More information is available in the attached coursework specification.
Deadline:	As indicated on Moodle.

Learning Outcomes Assessed:

LO1: Deploy client-side JavaScript libraries to add dynamic functionality within a web page

LO3: Integrate client side and server-side coding into coherent web applications

Note: 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.

All coursework has a late submission window of 5 working days, but any work submitted within the late window will be capped at 40%, unless you have an agreed extension. Work submitted after the 5-day window will be capped at zero, unless you have an agreed extension.
Please note that individual tutors are unable to grant extensions to coursework.

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). For advice relating to exceptional factors, please see the following website: https://www2.mmu.ac.uk/student-case-management/guidance-for-students/exceptional-factors/ or visit a Student Hub for more information.

Plagiarism: Plagiarism is the unacknowledged representation of another person's work, or use of their ideas, as one's own. Manchester Metropolitan University 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.

As part of a plagiarism check, you may be asked to attend a meeting with the Unit Leader, or another member of the unit delivery team, where you will be asked to explain your work (e.g. explain the code in a programming assignment). If you are called to one of these meetings, it is very important that you attend.

Assessment Criteria:	Indicated in the attached assignment specification.
Formative Feedback:	A formative submission opportunity is available on the following date: • **28th February 2020** More information about formative submission opportunities are 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 this assessment 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.

The aim of this coursework is to give you a platform to demonstrate what you have learned about making dynamic web applications that utilise a JavaScript framework and the Model-View-Controller (MVC) framework, which can incorporate real-time client-server interactions using real-time messaging.

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 and Node.js 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, you must develop a game or computer game review website. The precise details of the coursework task are detailed in Section 4 below, but the website in brief is dynamic and allows admin staff to add as many reviews as they wish. It also allows admins to chat to user though a Node.js server.

You will need to design and develop a front-end for the website, provide a way for users to comment on reviews, plan a real-time client and server interaction pattern for a chat system and then finally implement this into a full web application. This assessment does provide a blank copy of the CodeIgniter folder that can also be downloaded from the CodeIgniter website using the following link (https://www.codeigniter.com/). The completion of your weekly lab tutorials will aid you to complete this coursework, however it does not provide the solution for it. It is down to the student to utilise their skills to develop new code based on the material taught.

4. The PHP Frameworks Assessment (1CWK50)

a. Outline

To complete this assessment, you must further the development of a website using PHP and JavaScript frameworks, from skeleton code provided for you[1]. The website **must** be applied to game reviews. Provided for you in the skeleton code is a Node.js server. You should add code that allows you to chat between users and limit this interaction to chatrooms so that only members of a said chatroom can read the messages. 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.

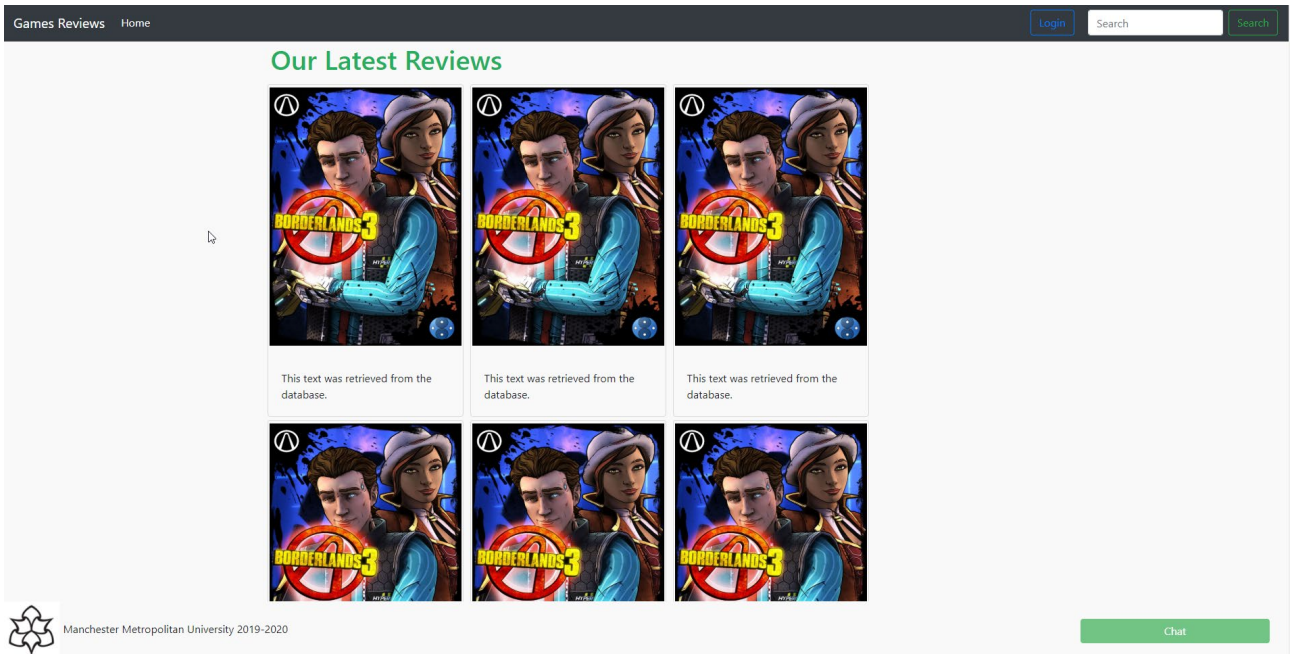

Figure 1: Example of the scalable front-end for a games review website.

Your submission will be marked in the following areas:

- Front – End Design
- Structure and Scalability
- Interactivity
- Client-server Networking
- Code Structure and Quality

Please ensure you have examined the mark scheme closely, to ensure that work you are completing is part of the mark scheme. Marks are not, for example, allocated based on the quality of the graphics you have chosen, and time spent on this could be more sensibly used to improve the networking functionality or the scalability of your web application.

b. Additional Guidance

Front End Design

The front-end design of the webpage requires you to focus on user interaction and a simplistic layout in order to achieve in this criterion. The skeleton code does not provide any design elements to you; therefore, it is your responsibility to implement it. You are encouraged to utilise the Bootstrap framework and their online documentation to complete this element of the coursework. Good presentation and good usability will provide higher scoring than advanced graphics and colour schemes with poor usability. As a rule of thumb, a user of the website must access all areas with ease and not be hindered by the design.

Structure and Scalability

You will need to expand on the skeleton code provided to you for the web application and add new pages and dynamic content, following the workflow of Model-View-Controller (MVC). To complete this objective, you must create appropriate models, views and controllers for your pages, promoting scalability.

Pay close attention to the data that flows through your web application. This will help you decide on what classes in MVC you will create. For example, not all views (webpages) will require their own Model. It is possible to link a model to multiple views.

For a starting point, refer to the following weeks of material in Moodle:

Week 10 - Model-View-Controller: An Introduction

To help you decide on the amount of classes (model, views and controllers) you should have, refer to the following weeks:

Week 11 - CodeIgniter: Models and Views

Week 12 - CodeIgniter: URI Routing and Controllers

Interactivity

Multiple elements form the interactivity in your web application. To meet this objective, consider the interaction between web pages and user navigation within your web application. You must also consider how a user can input information into your web application, and have this displayed immediately in a page, as this too forms part of interaction.

In relation to navigation and links, utilise the following:

Week 12 - CodeIgniter: URI Routing and Controllers

For displaying input immediately on a web page, refer to:

Week 19 - Vue JS: Integrating with CodeIgniter

Client-Server Networking

You will need to further the development of the skeleton code for your server and add accompanying code to the skeleton code for the client. You will then need to develop a Node.js server, **using Socket.IO**, which both receives and sends messages. Marks will be awarded in this area for achieving the following tasks, which should be attempted in this order:

1. When a user messages in a server, they are presented with a basic introductory message.
2. A basic user's message displays differently to the message of an admin.
3. A message is sent to a specific chatroom when a user connects to a server
4. The messaging facility is disabled if the server is offline.

For a starting point, refer to:
Week 20 - Working with Node.js

For handling messages, refer to:
Week 22 - socket.io and Messaging

It is your responsibility to expand on the skeleton code to include chatrooms. You are required to complete **individual research** into chatrooms within socket.io. Please utilise your tutors to ask for assistance if required.

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 code originates 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 JavaScript to solve your problems. Appropriate use of variables, arrays, objects and functions will all earn you marks – and misusing these features may be penalised.

c. Mark Scheme

Your work for this coursework will be graded in several areas, attracting a mark 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 1CWK50, which is worth 50% of the final unit mark.

Front-End Design (LO1)		**Interactivity (LO1)**		**Structure and Scalability (LO3)**		**Client-Server Networking (LO3)**		**Code Quality**	
No user created content displayed on page.	0 marks	Single webpage produced	0 marks	Single page produced with no framework or dynamic content	0 marks	No/non-functioning attempt at chat functionality	0-4 marks	Code is **poorly** presented, with **no** documentation or comments.	0-3 marks
All content displayed on page, with minimal effort on presentation (basic CSS used)	1-9 marks	Web page produced with dynamic content	1-9 marks	Basic use of the Model-View-Controller (MVC) framework, but little to no scalability in code. (Single classes)	1-12 marks	A chat server has been established that can be connected to from a web app	5-9 marks	Code is **moderately** presented, with **some** documentation, and **occasional** comments.	4-6 marks
						The above, **plus** an interface for chatting to all users on a server.	10-12 marks		
The above, plus web app is clear and well aligned and allows users to drill into related pages.	10-14 marks	The above plus additional pages interlinking with support for user inputted dynamic content	10-14 marks	The above, plus good use of the MVC framework with appropriate classes, encouraging scalability.	13-18 marks	The above, **plus** the ability to distinguish when the message originates from a basic user or an admin and disabling the chat feature if the server is offline.	13-18 marks	Code is **well** presented, with **good** documentation and **sensible** comments.	7-10 marks
The above, plus design is scalable using the Bootstrap framework	15-20 marks	The above, plus ability for login, account management, and chat	15-20 marks	The above plus a complete integration with a JavaScript framework for dynamic content	19-25 marks	The above, **plus** the integrations of chatrooms to prevent all users on a server from receiving a message	19-25 marks		

For example, a student earning 14 marks in *Graphics and Animation*, 14 marks in *Interactivity*, 12 marks each for their *Multiplayer Design Plan* and *Client-Server Networking,* and 6 marks for *Code Quality* would achieve an overall mark of **58%** (14 + 14 + 12 + 12 + 6).

5. Support for the Assessment

a. Help! I do not know where to begin or what to do!

Do not 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 the front-end), 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 – 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 familiarise 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 units' 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 **28th February 2018**. 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 – however 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 Henry
John Dalton, E137
j.henry@mmu.ac.uk
0161 247 1450