University of Salford MANCHESTER

Assessment Information/Brief 2022-23

To be used for all types of assessment and provided to students at the start of the module.
Information provided should be compatible with the detail contained in the approved module specification although may contain more information for clarity.

Module title	**Client Server Systems**
CRN	50249
Level	5
Assessment title	**Assignment 2**
Weighting within module	This assessment is worth 50% of the overall module mark.
Module Leader/ Assessment set by	**Lee Griffiths,**
Submission deadline date and time	**19/4/2023 by 4pm**
How to submit	**Please read this carefully.**

1. Your solution is an extension of Assignment 1
2. A compressed .**zip** folder **must be submitted** to Blackboard in the Assessments area containing the complete folders and all files associated with your solution to the assessment task – there is a typical submission limit of 50MB – you should not need to exceed that.
3. A Word or .pdf document which contains all your main project code including the .php and .phtml and JS files **you have created** presented neatly and readable in the following way:

    a. Front page with full name, university ID and title (Client Server systems Assignment 2)
    b. Any Model files that you have added or modified since Assignment 1
    c. Any Controller files, associated View files and JS files that you have added or modified since Assignment 1
    d. Any CSS files you have created (not Bootstrap files or JS map API library files needed)
    e. Clear screen screenshot of the main database table records.

    For this assignment we would expect around 6 model files and 4-6 controller/view pairs. This is just a guide.

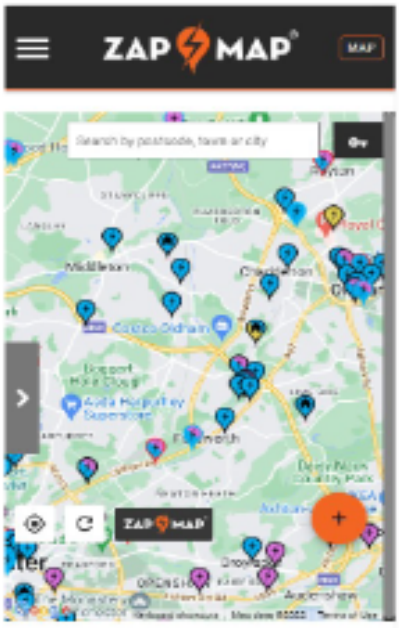
    f. a copy of the Assignment 2 Assessment Criteria and Marking Scheme grid (see later) with your self-assessment of your performance for each requirement – use the yellow

highlighter tool like this in Word to highlight what you have completed. This should be included in your Blackboard submission and on your live website with the URL on the website e.g. /clientserver/AssessmentCriteria2.docx and a link accessible from the homepage.

Assessment task details and instructions

Your task is to further develop the features of your Assignment 1 PHP MVC web application by using the Client Server technologies covered in the Trimester 2 workshop exercises (**mostly JavaScript**) – *do not use jQuery*. Your enhanced features must be written in **plain JavaScript** and for the highest marks you need to build classes in your JavaScript code, follow a design pattern and generally take an Object Oriented development approach. You need to do the following:

1) You need to develop an interactive live mapping feature which extends your Assignment 1 solution. It needs to allow **authenticated** (logged in) users to list/view all available charge points by their stored location on a map using mapping code techniques covered in the class. It should be similar to the **www.zap-map.com** shown right, but does not need to be as sophisticated. The map should initially center on the logged-in users own location using geolocation techniques .
You will need to make use of coordinate data that you store in your database for each charge point. Location data (lat/lng) can be generated using a tool such as **www.mockaroo.com**. You should use plain JavaScript AJAX techniques to implement your **live system**. Do not use web sockets or jQuery– these are beyond the scope of this work.
(60 marks for this mandatory section)



2) Then choose **one** of the following to develop your system further **(40 marks)**.

- AJAX implementation of live search feature and results for including feature information where appropriate and search filters. Consider browser memory usage. The search feature must be powerful and allow effective narrowing of results to a small number for a large dataset.
- AJAX implementation of "infinite scrolling" or "load more" on demand dynamic results of data listing (based on material in Workshops 15 & 16 and JS user interface event handlers) with appropriate number (1000s) of data items to demonstrate it well. Consider a sliding window of results to limit browser memory usage.

Total 100 marks

Assessment See "**Marking Scheme**" on the next page (page 4) as well as "**Grade Descriptors**" at

Criteria	the end of this document (pages 7-9). You should look at the assessment criteria to find out what we are specifically looking at during the assessment.

Assessment Criteria for Semester 2.

You should attend all lectures and workshops to fully understand what is required from work. All AJAX work should use native JS calls such as HTTPRequest or fetch, **not jQuery**.

You should look at the assessment criteria to gauge your solution and progress. **Note that this is only a guide to marking – credit will be given where appropriate.**

Mark range %	100-80% Outstanding to Excellent *(60-48 marks)*	79-60% Very Good to Good *(48-36 marks)*	59-40% Fair to Adequate *(36-24 marks)*	39-20% Unsatisfactory to Poor *(24-12 marks)*	19-0% Very poor to Extremely poor *(12-0 marks)*
Mapping **required feature** 50 marks	**Industry ready** application with **excellent performance** and **efficient data usage**. • Excellent OO code structure including reusable classes, design pattern(s) in both JavaScript and PHP as appropriate. *(up to 10 marks)* • Extensive use of **HTTPRequest** on **three or more** features of the solution UI to improve efficiency and performance using AJAX. *(up to 10 marks)* • Excellent and **secure input** validation, and demonstration of security **protection** such as URL tokens. *(up to 10 marks)* • **JSON or XML** data formats used for AJAX. PHP DB classes modified/extended to produce JSON/XML as necessary. *(up to 10 marks)* • **Sophisticated** map system with AJAX driven **real-time notifications** and **map updates**. Current user geo located. *(up to 10 marks)* • Excellent and consistently commented code. *(up to 10 marks)*	• Reusable JavaScript functions or classes added to perform robust input validation and displaying data to the users **dynamically**. • At least two AJAX type data transactions to acquire data from the PHP backend. • **JSON or XML** data formats used for AJAX. • **Useful** map system functioning using AJAX. Current user geo located. • Good comments evident throughout.	• Some JavaScript added to perform input validation and/or displaying data to the users dynamically. • At least one AJAX type data transactions to acquire data from the PHP backend. • Plain text data format used for AJAX. • **Basic** map/list system functioning using AJAX. • Some useful code comments evident.	• Some JavaScript added to perform basic input validation and/or displaying location data to the users from PHP backend but non- functioning or incomplete/has issues and unsatisfactory system. • **Significant amounts of code taken from internet sources.** • Minimal code comments.	• Little or no JavaScript or extra features added to your semester one work. • **Significant amounts of code taken from internet sources or other students.** • No code comments.
Chosen feature 30 marks	**Industry ready** feature with **excellent performance** and **efficient data usage**. • **Excellent** OO code structure including reusable classes, design pattern, in both JavaScript and PHP as appropriate. *(up to 8 marks)* • Extensive use of **HTTPRequest** on different features of the feature to improve efficiency and performance using AJAX. *(up to 8 marks)* • Excellent and **secure input** validation, and demonstration of security **protection** such as URL tokens. *(up to 8 marks)* • JSON or XML data formats used for AJAX. PHP DB classes modified/extended to produce JSON/XML as necessary *(up to 8 marks)* • Excellent and consistently commented code *(up to 8 marks)*	• Reusable JavaScript functions or classes added to perform robust input validation and displaying data to the users **dynamically**. • At least two AJAX type data transactions for your chosen feature. • JSON or XML data formats used for AJAX. Good comments evident throughout.	• Some JavaScript added to perform robust input validation and/or displaying data to the users dynamically. • At least one AJAX type data transaction for the chosen extra feature. • Plain text data format used for AJAX. Some useful code comments evident.	• Some JavaScript added to perform basic input validation or displaying data to the users but non- functioning, incomplete and unsatisfactory extra feature. • **Significant amounts of code taken from internet sources.** Minimal code comments.	• Little or no JavaScript or extra features added to your semester one work. • **Significant amounts of code taken from internet sources or other students.** No code comments.
Change a form validation to Java script 5 marks	• Explain in which field and wat type of validation used				
Smooth the user	• Explain each functionality				

Assessment Information/Brief

experience with 3 JavaScript functionalities (3 case each case 5 marks) 15	