Department of
Computing and Mathematics
ASSIGNMENT COVER SHEET

Unit title:	Web Development
Assignment set by:	Ashley Williams
Assignment ID:	1CWK100
Assignment title:	Create a website for a musician
Assessment weighting:	100%
Type: (Group/Individual)	Individual
Hand-in deadline:	10th March 2022
Hand-in format and mechanism:	Via Moodle

Learning outcomes being assessed:

LO1: Create basic web pages using HTML to mark-up content, using CSS to control its presentation in web browsers

LO2: Write efficient and readable client-side scripts that are event- and object-driven and runs on multiple browsers and platform

LO3: Apply web design usability principles in the creation of web content based on the requirements of a given scenario

Note: it is your responsibility to make sure that your work is complete and available for marking by the deadline. Make sure that you have followed the submission instructions carefully, and your work is submitted in the correct format, using the correct hand-in mechanism (e.g., Moodle upload). If submitting via Moodle, you are advised to check your work after upload, to make sure it has uploaded properly. Do not alter your work 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:	Lecture/Lab discussion and interactive with tutor onwards from when the assignment is set.
Summative Feedback Format:	You will be given individual feedback via Moodle within 20 working days of your submission deadline, as well as general feedback for all the class.

Web Development

Assignment – Create a website for a musician

1. Introduction

This assessment is coursework based, and worth 100% of the overall unit mark. The tasks that you are required to complete for this assessment are outlined in this coursework specification.

2. Aim

This unit encourages you to analyse real world situations critically. The assessment mimics industry projects by requiring you to engage with multiple disciplines. By the end of the unit, you will have completed the development of an application that uses a variety of web technologies. It is encouraged that you maintain a portfolio of projects throughout university (e.g. through GitHub) that can serve as a portfolio of your work when applying for jobs. This project could serve as one aspect of your portfolio.

The following skills will be essential for successful completion of this coursework (and including such a project in your portfolio would demonstrate these skills to potential employers):

- **Real world problem solving**: You will need to analyse a real-world situation, develop solutions for multiple problems when developing the application, and then evaluate your solutions.
- **Technical skills**: This assessment requires you to write an application using the fundamental technologies that make up the web. In addition to these technologies, you will gain a foundational understanding of how the web works.

2.2 Assessment Learning Outcomes

LO1: Create basic web pages using HTML to mark-up content, using CSS to control its presentation in web browsers

LO2: Write efficient and readable client-side scripts that are event- and object-driven and run on multiple browsers and platform

LO3: Apply web design usability principles in the creation of web content based on the requirements of a given scenario

3. Coursework Overview

To complete this assessment, you are required to develop a web application. The precise detail of the coursework tasks are detailed in section four below. However, to summarise, you will be developing a website to showcase your knowledge of web fundamentals.

4. The Assessment (1CWK100)

Create a website to showcase your favourite musician/band. The specific requirements for your website are outlined below.

4.1 HTML (30%)

Standard (30%)

You should design your site, making use of the following HTML features:

1. Multiple pages with a way of navigating between the pages using anchor tags
2. Pages that are divided into sections using the HTML tag
3. Use of headers and paragraphs for organising and displaying content about the musician
4. Use of HTML lists (ordered or unordered)
5. Use of images throughout the website with an appropriate alternative text attribute
6. All HTML pages should be well formed and validated through https://validator.w3.org/#validate_by_input

4.2 CSS (25%)

Standard (20%)

You should style your site by making use of Cascading Style Sheets. Specifically, you will be marked on the following aspects:

1. Your style rules should be in an external style file (with a .css extension) and linked to from within your HTML files
2. You must have styled your website in terms of layout/positioning, text, and content
3. You must have demonstrated appropriate use of classes and IDs to style groups and individual components

Extension exercise (5%) – medium difficulty

1. Make your website responsive. Your submission should include a text file containing a short paragraph that describes the efforts you have made to make your site responsive.

4.3 JavaScript (45%)

Standard (35%)

Your site should allow visitors to sign up to a mailing list. When the user enters their details into a HTML form and clicks a button to sign up, the details should be sent to the server provided and then provide a success/failure message to the user. Follow the steps below and the lecture notes to guide you:

1. Create a HTML form that contains two text fields (for name and email) and a submit button
2. When the submit button is clicked, use JavaScript to validate your form input. If the input fails validation, then provide feedback to the user with a reason why
3. On passing validation, use JavaScript to send a POST request to the server (mudfoot.doc.stu.mmu.ac.uk/node/api/mailinglist). The server expects raw JSON to be sent in the following format:

```
{
    "name": "Ash Williams",
    "email": "ashley.williams@mmu.ac.uk"
}
```

4. Make sure that your JavaScript can handle errors coming back from the server (e.g., if the wrong data is sent). This is another layer of validation.

```
"email" must be a valid email
```

5. On success (200 OK from the server), parse the response and provide a success message to the user.

```
{
    "message": "Thank you for signing up",
    "data": {
        "name": "Ash Williams",
        "email": "ashley.williams@mmu.ac.uk"
    }
}
```

Extension exercise (10%) – hard difficulty

The server also accepts GET requests to mudfoot.doc.stu.mmu.ac.uk/node/api/halloffame. Sending a GET request to this URL provides the 'Rock and Roll Hall of Fame' data for the year 2021. You can change the year by adding a query parameter to the URL (e.g., mudfoot.doc.stu.mmu.ac.uk/node/api/halloffame?year=1999). Add a page to your site that allows the user to query hall of fame data for any given year. Your page should parse the response from the server to display the data in an appealing way, include images using the server data, and handle any problems/errors gracefully.

For full marks, your page should load up the data for 2021 initially, and then allow the user to change the year using the form.

5. Submission

Submission of this coursework will be online, through the university's Virtual Learning Environment (Moodle). You must upload **a single zip file**, which includes the following:

1. All your source code along with any additional files that are required to run the application

6. Assessment Marking Criteria

	Fail (0 to 29%)	**Marginal Fail (30 to 39%)**	**3rd Class (40 to 49%)**	**2nd Class: 2 (50 to 59%)**	**2nd Class: 1 (60 to 69%)**	**1st Class (70 to 85%)**	**Exceptional 1st (86 to 100%)**
HTML **30%**	Little or no HTML has been submitted.	You have submitted some HTML, but it is not well formed and littered with errors.	You have submitted at least one HTML document that parses correctly on the screen. However, it is lacking in functionality and/or contains some errors.	Your HTML is mostly well formed and covers most elements required. However, other elements are missing and/or you have consistently used inappropriate elements.	You HTML is well-formed and works. However, there are some examples of errors and/or inappropriate element use scattered throughout.	You have submitted multiple well-formed HTML pages that link together. You have covered all elements required but there are parts of your site where other elements would have been appropriate.	You have submitted multiple well-formed HTML pages that link together and contain all elements required in a way that is appropriate.
CSS **20%**	Little or no CSS has been submitted.	You have submitted some CSS, but it is not well formed and littered with errors.	You have an external CSS sheet, which is correctly pulled into at least one HTML file. However, it is lacking in functionality and/or contains some errors.	Your CSS is stored in an external sheet and linked to by all HTML files. Your styles cover a large portion of the site, but you haven't fully demonstrated what the specification asks.	Your external CSS demonstrates the points asked for in the specification. However, there are some examples of errors and/or areas for improvement.	Your external CSS sheet contains styling for the entire site. You have covered all points in the specification, but there are parts where you could have made your CSS more efficient.	Your external CSS sheet contains styling for the entire site to a professional standard.
Extension 1: Responsiveness **5%**	Little or no attempt has been made.	You have submitted a text file explaining some steps taken, but there are many errors/broken parts.	You have made progress towards making your site responsive, but there are a lot of bugs on most displays.	Your site displays correctly on at least two different formats (e.g., browser and mobile). You have outlined appropriate changes in your text file.	Your site is mostly responsive, however there are some elements that don't appear to format correctly when changing the display size.	You have demonstrated use of a flexible framework (e.g. flex-box). However, there are some minor bugs/caveats on certain displays.	Your site is fully responsive and will display correctly on any device.
JavaScript **35%**	Little or no attempt has been made.	You have a HTML form that calls JS validation functions on submission. However, they don't work/are littered with errors.	You have appropriate validation functions that prevent the form from submitting. You have attempted the API request but there are a lot of errors.	Your validation and API call works. However, you are incorrectly handling the call/response and/or you aren't following the specification. There are also some bugs throughout.	The mailing list works and mostly fits with the requirements outlined in the specification. There are some bugs and/or areas that could be made more efficient.	Your mailing list adheres to the specification, but there are some minor bugs.	Your mailing list follows the specification and is working to a professional standard.
Extension 2: Hall of Fame **10%**	Little or no attempt has been made.	You have a HTML form and some validation. However, no API call and/or the code is littered with errors.	The validation prevents the form from submitted and you can make a call to the API. However, you are not parsing the data correctly or at all.	The validation and API call works with data displayed on screen. However, there are bugs throughout.	Your hall of fame page mostly adheres to the specification. There are some bugs and/or areas that could be made more efficient.	Your hall of fame page works an adheres to the specification, but with some minor bugs.	Your hall of fame page works to a professional standard and adheres fully to the specification.