S2 Multi-Device Application Assessment Brief

Key Information

Deadline	**12 – 18 January, 2026**
Marking tutor	Mr. Oliver I C. Malapit
Contribution to overall module mark	50%

The Brief

You are tasked to create a single-page web application, comprised of a responsive front-end written in HTML/CSS and a small back-end created using Node.js. It is up to you what your application will be. For example, it could be a storefront, a practical utility, or an interactive experience. But, you should discuss your idea with your tutor before you start to ensure it is within the scope of the brief.

The front-end should be responsive to at least phone and desktop devices. You should demonstrate a solid grasp of responsive design principles, focusing on both visual clarity and user experience. The back-end should serve static content (e.g. stylesheets, images, etc) and implement at least one additional feature. For example, it might serve a dynamic webpage, use data from a database or web API, or allow users to “log-in” to access content. Your implementation should efficiently use modern web development techniques.

You must iteratively design, implement, and document your project. Your documentation should explain how and why you changed the application as you iterated, and provide clear evidence of the process. Evidence could include design wireframes, test plans, peer feedback, and GitHub commits.

Deliverables

The deliverables for this assignment are as follows:

- The Application. (A GitHub repository that contains all required code and assets)
- A 1,000 word development document. (Word or PDF), including evidence of iteration.
- A short video walkthrough of your website across different screen sizes.

<u>The Application</u>

The Node.js project folder required to run your Multi-Device Application. This should be submitted to the corresponding folder in your GitHub repository. If you are unsure, ask your tutor.

You should commit the package.json file but should not commit the node_modules folder. Your GitHub repository should be configured to do this by default. If you encounter issues, please contact your tutor.

<u>The Development Document</u>

Your Multi-Device Application must be accompanied by a development document of 1,000-1,500 words. You should include any evidence of an iterative process in the development document.

It is up to you what sections your document will include. However, you must include the following elements:

- **Brief:** A short description of your Multi-Device Application, including its key features, target device types, and any requirements you might have set. You should also provide a link to your GitHub repository and to your walkthrough video. (~100 words)

- **Development Breakdown:** Outline and explain 3-5 key decisions you have made during the development of your application. This should include both design decisions (e.g. how and design differs between devices and why) and technical decisions (e.g. how and why you implemented a key feature). You should explain how iteration, testing, and/or peer feedback informed these decisions. (~500 words)

- **Critical Reflection:** Critically analyse your application to explain its successes, its limitations, and how it could be improved. You should also describe any personal reflections on how you could further improve your own web development skills. (~400 words)

- **Appendix:** You may include evidence of iteration in appendices at the end of your document. You may also upload files to your GitHub repository and include the link in your document. Appendices do not count towards the word count.

Walkthrough Video

You should also submit a brief walkthrough video to your GitHub repository and include the link at the top of your development document. The video should demonstrate how your application responds across different screen sizes and any key features of your application.

n.b. The content of this video is not a part of your documentation and will not be marked. However, it will be referenced by the marker and moderator to see how your application functions on your personal computer.

Submission

Please follow the submission instructions below. Work that is submitted incorrectly may not be accepted or could incur a points penalty.

Before submitting have you…

- Checked that your work is functioning as expected?
- Spell-checked and grammar-checked any written work that accompanies your digital work? Please make an appointment with the Academic Skills team or speak to your tutor if you are experiencing challenges in this area.
- Formatted your written work to the specification below?
- Referenced all sources of information accurately? Please refer to www.citethemrightonline.com (Harvard) for guidance.

Your development document must be submitted via Turnitin and code submitted via your GitHub repository. Please adhere to the following method:

- Check your Multi-Device Application is working as expected.

- Commit and push the final version to your Web Dev II Github repository. Ensure you push all the files required to run your application, including code, assets, and the package.json file.
- Ensure you have included links to your GitHub repository and walkthrough video at the beginning of your Development Document.
- Save your development document as a Word document or PDF file.
- On Minerva, find the Multi-Device Application submission portal in the Assessment folder.
- Submit your development document.

Only code pushed to your GitHub repository before the assessment deadline will be marked. Ensure you give yourself enough time before your final push.

Format
All written work must conform to university styling and submission guidelines. They must:

- Contain appropriate in-text citation that supplies an accurate list of references.
- Be accurate in referencing. See Bath Spa guidelines and the Harvard system described at www.citethemrightonline.com.
- Be accurate in spelling and paragraphing.

Use of online sources
If you take any code from online resources - including AI tools and online tutorials - to support the development of your application, you must fully acknowledge their use by referencing the source in both code comments and in a references list at the end of your development document.

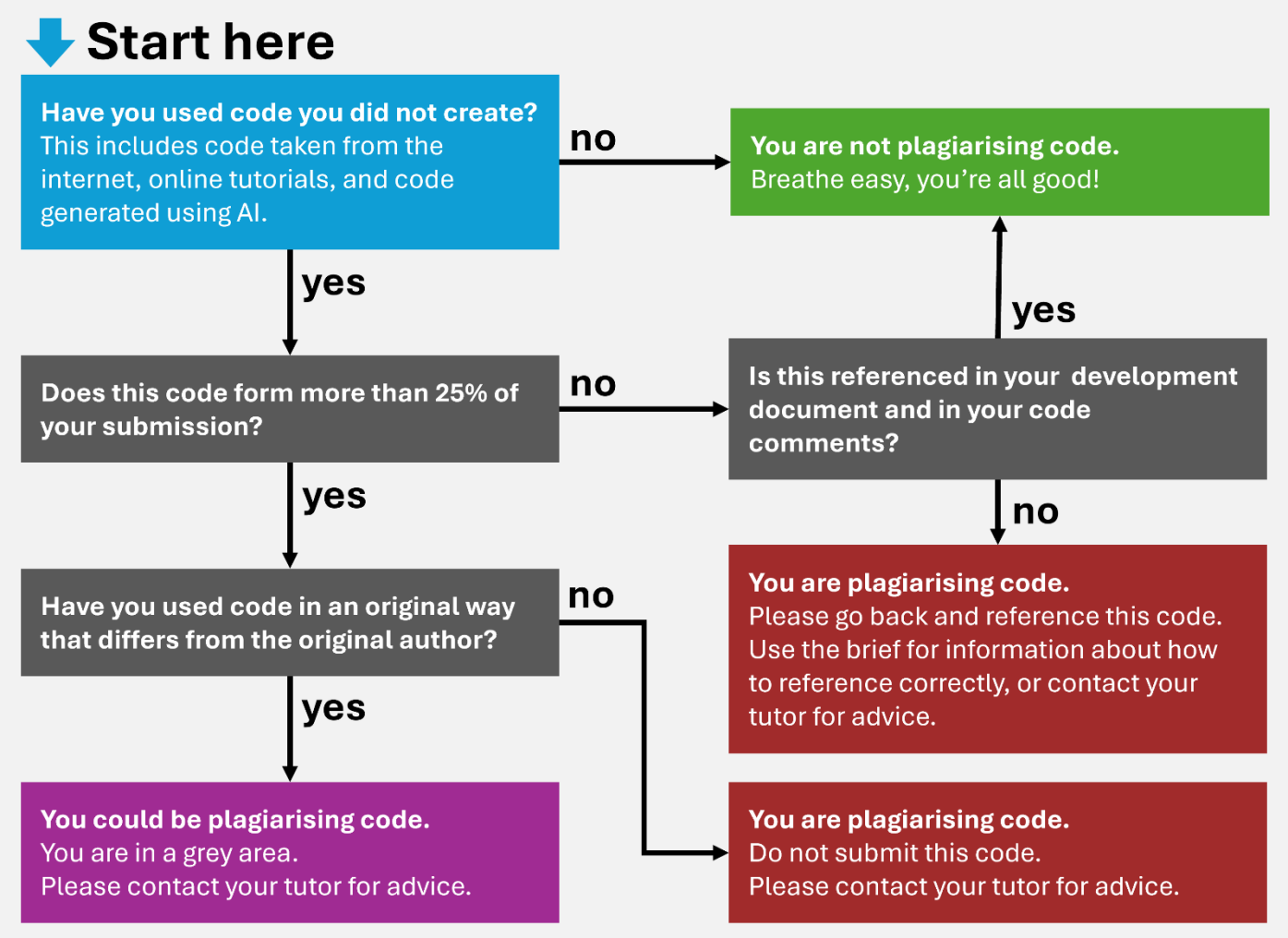
- Code Comments: A reference in the code comments should be added ahead of the section of code that has been taken or heavily developed using online resources.
- For standard online sources you should include the title of the webpage and full URL.
- *For Generative AI tools you should include the name of the tool and the prompt provided to generate the code.*

- Reference List: A full reference list should also be included at the end of your development document. These references should be in alphabetical order.
- *For standard online sources you should follow the Harvard standard for websites.*
- *For Generative AI tools you should follow the Harvard standard for generative AI.*

If you are unsure of how to appropriately reference, please consult your tutor. Failure to appropriately acknowledge use of online resources may result in an academic misconduct accusation.

Also be aware that use of online resources to develop your code should be kept to a minimum and excessive use may be deemed academic misconduct. The flowchart below is offered as guidance to what is and is not acceptable. Again, if you are unsure, please consult your tutor.

Am I Plagiarising Code?



Marking Criteria

The Multi-Device Application will be marked against the following criteria:

- **Responsive Design and User Experience Design (30%)** - Quality of visual design and user experience design across device types.
- **Technical Implementation (40%)** - Use of front-end and back-end techniques. Use of code conventions (e.g. comments, indentation, kebab-case, etc) and GitHub (e.g. file structure, regular commits, etc).
- **Documentation (20%)** – Clarity and depth of information and explanations provided. Depth of critical reflection.
- **Evidence of Iteration (10%)** - Evidence of iterative design, implementation, testing, and peer feedback to inform the final outcome.

Criteria	Description	Mark Range
Responsive Design and User Experience Design (30%) Quality of visual design and user experience design across device types.	A highly limited design that pays little to no attention to user experience across devices. The site is not responsive.	0 – 19 (Low Fail)

	A poor design that demonstrates an attempt to create a user experience, but lacks coherence and clarity. The design responds poorly and/or features other major errors.	20 – 39 (Fail)
	A basic design that pays some attention to user experience across devices, but with modest results. Visual design or interaction design may be outdated or lack coherence. Visual hierarchy and layout may lack clarity. The design is mostly responsive.	40 – 49 (Third)
	A fair design that demonstrates a sound approach to responsive web design. Visual hierarchy and layout for most or all devices is appropriate, but other design aspects may require refinement. The design responds well across all target devices, potentially with minor errors.	50 – 59 (2:2)
	A good design that demonstrates a sound approach to user experience design. There may be errors in visual hierarchy or interaction design, but these do not significantly disrupt the user experience. Attention to detail has been paid to responsive design across all target devices.	60 – 69 (2:1)
	A very good design across all target devices, likely with only minor errors. Visual hierarchy and interaction design	70 – 79 (First)

	demonstrate a strong attention to detail. Visual elements (e.g. image, video, font, colour) are highly cohesive. The design likely responds well to a range of devices.	
	An excellent design that pays close attention to user experience across all target devices. Visual hierarchy and interaction design have been implemented to a high standard, and this has been expanded upon further using techniques not taught in class. Visual elements are highly effective. The design responds to a range of devices.	80 – 89 (High First)
	Beyond expectations for this level of study.	90 – 100 (Outstanding)
Technical Implementation (40%) Use of front-end and back-end techniques. Use of code conventions (e.g. comments, indentation, kebab-case, etc) and GitHub (e.g. file structure, regular commits, etc).	A very limited implementation that does not deploy the modern web development techniques required.	0 – 19 (Low Fail)
	A poor implementation that may deploy essential web development techniques, but with major errors. Front-end use of HTML/CSS is likely substandard or ill-considered. Back-end use of Node.js/JavaScript may demonstrate limited evidence of programming skill.	20 - 39 (Fail)
	A basic implementation that successfully deploys Node.js, but otherwise only uses critical web development techniques.	40 – 49 (Third)

	Use of front-end and back-end techniques are appropriate, but notable refinement may be needed. Repository presentation and/or use of coding conventions may be substandard.	
	A fair implementation that successfully deploys responsive front-end and dynamic back-end techniques, but with scope for improvement. Front-end HTML/CSS are deployed appropriately, though there may be minor errors in semantic markup or responsive techniques. Back-end JavaScript implements some dynamic features, although there is room for improvement. Repository presentation and use of coding conventions is fair.	50 – 59 (2:2)
	A good implementation that successfully deploys a range of web development techniques, but may have minor errors. Front-end techniques are used effectively. Use of back-end techniques demonstrates good understanding of how to implement dynamic features, though use of these techniques could be extended further. Repository presentation and use of coding conventions feature only minor errors.	60 – 69 (2:1)
	A very good implementation that skilfully deploys a range of web development	70 – 79 (First)

	techniques, likely including techniques beyond the class materials. Code is efficient and avoids duplication. The range of techniques used demonstrates very good exploration into of Node.js and HTML/CSS features. Repository presentation and use of coding conventions are to a high standard.	
	An excellent implementation that deploys a range of techniques, including techniques beyond the class materials, demonstrating an understanding of the industry standard. The design and features of the application have been implemented with precision. Sophisticated use of a range of techniques and evidence of non-trivial problem solving demonstrates strong programming skill. Repository presentation and use of coding conventions are to a very high standard.	80 – 89 (High First)
	Beyond expectations for this level of study.	90 – 100 (Outstanding)
Documentation (20%) Clarity and depth of information and explanations provided. Depth of critical reflection.	Very limited documentation that demonstrates little or no understanding of the development process. Required elements are missing or structure is unacceptable.	0 – 19 (Low Fail)
	Poor documentation that provides little insight into the development process. Content is unclear or poorly structured. Required elements may be	20 – 39 (Fail)

	missing, or the critical reflection may be inadequate.	
	Basic documentation that includes information about the design and technical decisions made, but lacks depth or clarity in some places. There may be too much description, and insufficient explanation of key decisions. Key concepts may be misunderstood. The critical reflection may be vague or generic.	40 – 49 (Third)
	Fair documentation that provides key insights into the development process and is communicated clearly. Some sections may lack clarity or depth, but meaningful insight is provided throughout. The critical reflection contains appropriate insights.	50 – 59 (2:2)
	Good documentation that provides sound insights into the development process. Decisions are contextualised and explained well. The critical reflection contains sound analysis. Content is well structured and communicated clearly with only minor flaws	60 – 69 (2:1)
	Very good documentation that clearly and concisely documents the development process. Both design and technical decisions are explained in detail. The critical reflection contains thoughtful takeaways. Content is well presented and is communicated clearly.	70 – 79 (First)

	Excellent documentation that provides astute insights into the development process. A wide range of decisions are explained in detail. The critical reflection is highly critical and insightful. Communication is highly effective. Structure is without error.	80 – 89 (High First)
	Beyond expectations for this level of study.	90 – 100 (Outstanding)
Evidence of Iteration (10%) Evidence of iterative design, implementation, testing, and peer feedback to inform the final outcome.	Very limited or no evidence. There may be no indication that the work was developed with sufficient time to iterate.	0 – 19 (Low Fail)
	Poor evidence of an iterative process that provides only minimal information. Evidence may be unclear or vague.	20 – 39 (Fail)
	Basic evidence of an iterative process is provided. The amount of iteration is limited or has had minimal impact on the final outcome.	40 – 49 (Third)
	Fair evidence of an iterative process. There is clear evidence of repeated iteration, which informs at least some key aspects of final outcome.	50 – 59 (2:2)
	Good evidence of an iterative process. A range of evidence informs most or all key aspects of the final outcome.	60 – 69 (2:1)
	Very good evidence of an iterative process. Iteration is frequent and informs all key decisions, including both design and technical. Notable aspects of the work have been	70 – 79 (First)

	significantly improved through an iterative process.	
	Excellent evidence of an iterative process, which comprehensively documents the development process. Iteration has been used to refine macro and micro aspects of the final outcome.	80 – 89 (High First)
	Beyond expectations for this level of study.	90 – 100 (Outstanding)

Intended Learning Outcomes (ILOs)

ILO	Assessed
The application of HTML5, CSS and JavaScript to deliver interactive and responsive web content suitable for key device types.	✓
An ability to implement web applications that conform to contemporary web design conventions (semantic markup, accessibility, SEO).	✓
Successful deployment of a range of strategies for testing, troubleshooting and debugging responsive and interactive web projects.	✓
An ability to document the successes/limitations of an original build and identify personal learning opportunities for improving web development skills.	✓

Mark penalties may be applied to late submissions without prior approval of an extension. Please ensure that you prepare and submit your work in good time to allow for any issues that may arise.