Coursework 1 / Game Website
Website with Navigation - Marking Guidelines

Notes

I have left the submission system open to reduce the problems of slightly late submission, so could you please **check the submission date and time carefully when you mark the project**. The available marks start to halve 24 hours after the deadline, so a submission received at 17:01 on Saturday 9th November can only receive a maximum of 17.5 marks. A submission received at 17:01 on Sunday 10th November can only receive a maximum of 8.75 marks, etc. The available marks reach zero after 1 week. Mark late submissions in the usual way, then apply the mark penalty and provide an explanation in the general comments section of the marking form.

The website with navigation should be marked using the form 'CSD2550_Coursework1_WebsiteWithNavigation_MarkingForm_NAME.docx', which is available at the bottom of the course website. This form should be emailed to the student within 15 working days of the submission deadline.

The marking form will calculate the mark automatically for you as long as there are no blank marks. Select the final mark and press F9. I recommend that you experiment with this feature before relying on it to calculate the marks.

Do not deduct marks for the inclusion of copyrighted images. This is ok in their coursework, but that they should seek appropriate permission if they make their websites publicly available online.

The students are allowed to adapt templates and use WYSIWYG editors, such as Dreamweaver. However, a complete copied website without modification is not acceptable. It must be clear that the students have done some work and that they understand what they have done.

Students are allowed to use the code that I have given them as part of the course. For example, I have given them code that shows them how to use PHP to output HTML on the page. This code can be downloaded from the Week 5 section of the course website, and versions of it will appear in multiple projects.

Frameworks and third party libraries, such as Bootstrap and Angular, are allowed.

You need to look at their code carefully to check code quality. The students are expected to upload a document with screenshots. If this document is missing you might have to run the website using XAMPP.

Add a comment to justify every mark. 'Good' or 'ok' is fine if they are getting full marks. Otherwise the comment should explain why they have lost marks.

Mark the PHP from the code and the visual appearance from the screenshots (with some checks to establish that it corresponds to the code). If the student submits both PHP and HTML files and no screenshots, run both websites and mark the best looking of the two for visual appearance. Deduct marks for file organization in this case.

Detailed marking guidelines are given in the table below.

Do let me know if you have any questions.

Many thanks!

Feature	Assessment Criteria	Marking Guidelines
Website. Navigation bar and place holders for the game, registration, login and rankings functionality.	**5 marks.** Multiple pages and navigation bar.	5 marks should be given for a website that has more than one page, working navigation and some text and/or images. Single page designs are allowed as long as they include all of the required functionality. Some marks can be given for semi-functional attempts. Knock off 1-3 marks if the HTML has serious errors, for example, missing or duplicate <html> or <body> tags.
	10 marks. Attractiveness and usability. Are the pages well designed and attractive? Has thought been given to usability?	10 marks for a website that is the same quality as a commercial website. 5 marks for an amateurish looking website with some CSS and possibly images. Zero marks for a website with no CSS or images. Knock 1-3 marks off for poor usability. For example, if some pages are missing navigation links that enable the user to access the rest of the website.
PHP. PHP functions should be used to generate parts of the website dynamically. **You must use PHP *functions*. No marks are available for using *include* to add blocks of HTML to the page.**	**10 marks.** Use of PHP functions to generate parts of the website dynamically. These marks can also be awarded for other uses of PHP on the website, as long as it is not data storage.	These marks are for a basic use of PHP to output parts of the page, such as the footer or navigation bar. They are only available for the use of PHP **functions**, which will probably use 'echo' or 'print' to output HTML. No marks are available for the use of 'include' to dump HTML into the page. Other reasonable uses of PHP are permitted. 10 marks should be given for a decent amount of PHP, for example a complex function or a few functions. The full 10 marks should be awarded for a multi-page website that eliminates all duplicate HTML with PHP. The lecture gives a detailed example of a use of PHP to eliminate duplicate HTML, which would receive 10 marks. 5 marks should be given for a single simple PHP function that outputs some HTML onto the page. 1-5 marks should be awarded if the PHP pages are different from the pages that are shown in the screenshot. 5 marks should only be awarded if all duplicate code is output with PHP and if the HTML output by the PHP is similar to the website shown in the screenshots. 1-5 marks can be awarded for a non-functional attempt at this task.

		Zero marks should be awarded for submission of an unmodified copy of the PHP example code.
HTML, CSS and PHP code quality. Your code should be well commented, tidy and easy to read. Files should be sensibly organized into folders. Marks will be deducted for unused files and commented out code.	**2 marks**. HTML code quality.	1 mark for appropriate comments. For example, the student should use comments to mark key sections of the HTML, such as navigation, game, registration, footer, etc. 1 mark for tidy layout. Some editors use different formatting methods, so try to take this into account. Deduct one mark for commented out code in the submission.
	2 marks. CSS code quality.	1 marks for appropriate comments. For example, students should use comments to describe the functions of the different parts of the - navigation, registration, etc. It is not necessary to comment every CSS rule. 1 mark for tidy layout. Some editors use different formatting methods, so try to take this into account. Deduct one mark for commented out code in the submission.
	2 marks. PHP code quality.	1 mark for appropriate comments. For example, there should be a comment explaining each function and comments to explain key functionality. 1 mark for tidy layout. Some editors use different formatting methods, so try to take this into account. Deduct one mark for commented out code in the submission.
	2 marks. File organization.	Images should be in a separate folder and CSS and JavaScript files should be in separate folders. PHP files that contain the functions for outputting HTML (often called 'common.php') will ideally be placed in a separate php folder, but don't knock any marks off if this is not done. PHP pages ('game.php', 'login.php', etc.) can all be at the root of the website. The paths are more complicated if some of these files are in a subfolder and I have not shown the students how to do this. Deduct one mark for unused files in the submission. Deduct one mark if the documentation or

		screenshots are included in the folder that contains the website code. Deduct one mark if separate CSS files with near-identical contents are created for each page. For example, index.css, game.css, all with approximately the same code.
Screenshots of website pages. The screenshots can be combined into a Word or PDF document. There is no need to provide any text.	**2 marks.** Document containing screenshot(s) of all of the website's page(s). In the case of a single page website, provide screenshots of key functionality.	There should be screenshots of all of the website pages or functionality. Knock off marks proportional to the number of missing pages. For example, if there are screenshots of half the pages, give the student 1 mark. Knock off 1 mark if the screenshots cannot be viewed properly, for example if they are very low resolution.