Appendix 1: Coursework Grading Grid

Grade	A	B	C	D	E	F	NS
Definition	EXCELLENT Outstanding Performance	COMMENDABLE Meritorious Performance	GOOD Highly Competent Performance	SATISFACTORY Competent Performance	BORERLINE FAIL Open to Compensation	FAIL Unsatisfactory	No Submission
Code (30%)	Code is suitably complex but navigable: it is exceptionally easy to tie up elements on the website and follow their functionality. Code is documented through appropriately named variables and suitable comments detailing design or functionality decisions. Code is extendable and maintainable. It is perfectly obvious how the code could be extended in the future. Future code maintenance (including package/library updates) is carefully considered.	Code is navigable: most elements can be tied to their code. Functionality can be followed with a little effort. Code is documented although there may be some minor issues with obscure variable names or over- or under- commenting Code could be extended with a little rearrangement of some functionality. Code maintenance is considered.	Code is unnecessarily complicated but can be navigated with some effort. Functionality can be found with effort. Some code is documented through comments, but some variable names may be meaningless. Code could be extended with some considerable rearrangement of functionality. Code maintenance and future updates may be acknowledged.	Code is difficult to navigate, but functionality can be tied to lines of code for some elements. Code is difficult to understand, but a few parts can be followed lucidly. Some sections of code would require re-writing to extend functionality or to maintain the code in future.	Code is difficult to navigate or so minimal that code navigation is mostly unnecessary. Code is difficult to understand. Code extension would be difficult and most of the code would require re-writing for maintenance.	Minimal code. Code cannot be followed. Code would be scrapped and re-written rather than extended or maintained.	
Functionality (70%)	The A/B options are complex and multiple user metrics are successfully gathered. The front end is fully operational with no obvious issues in any browser. It stylish and easy to use.	The A/B options are suitably complex. User metrics are successfully gathered. The front end is fully operational with minor display issues not affecting functionality. It is	The A/B options are simple in nature. User metrics are gathered but there may be some minor issues. The front end is largely operational although there may be major display issues or issues which impact functionality.	There are A/B options and user metrics are gathered but there may be major issues. Major issues impact functionality of the front end to the extent that the web system is difficult to use.	There are A/B options but no metrics (working or otherwise). Front end is largely non-functional.	There is some attempt at A/B options. Front end is present but minimal functionality.	

		consistently styled throughout.					
	There is a complex, working dashboard accurately presenting the current status of multiple A/B metrics and allowing manual resets by authorised users.	There is a sufficiently detailed, working dashboard accurately presenting the current status of A/B metrics only visible to authorised users.	There is a working dashboard presenting A/B metrics.	There is some attempt at a dashboard to present the user metrics.	Dashboard page exists but may be empty.	No dashboard.	
	There is a clear separation between front and back end and functionality is obviously assigned appropriately.	There is separation between front and back end and functionality is assigned appropriately.	Front and back are separable but functionality may be assigned inefficiently to each.	Front and back end coexist but functionality is assigned arbitrarily to each.	There is an attempt at a back end but functionality is minimal.	No back end.	
	The fully operational and technically advanced web system is available online via a supplied URL.	There is an attempt to supply a valid URL but the technically advanced server clearly works well locally.	No URL but the sufficiently complex server works locally with minor issues.	No URL but the server works locally with some issues.	No URL and the server is minimal.	Little evidence of a working server side.	
	The web system makes advanced use of API programming for data handling.	The web system makes full use of APIs to access data.	The web system uses APIs to access some data.	Some relevant API use is attempted.	Minimal API usage.	No API usage.	
	Data for the web system is gathered/handled appropriately and stored securely.	Data for the web system is gathered appropriately and stored securely.	Data for the web system is gathered and safely stored.	Data is gathered.	Data is considered	No data handling.	

Grades will be averaged across all components of the grading grid to give the total grade for the coursework.

Appendix 2: Module Combination Grid

	Coursework Web System						
Coursework Report	**Grade**	**A**	**B**	**C**	**D**	**E**	**F**	**NS**
	A	A	A	B	B	C	E	
	B	A	B	B	C	C	E	
	C	B	B	C	C	D	E	
	D	B	C	C	D	D	E	
	E	C	C	D	D	E	E	
	F	E	E	E	E	E	F	
	NS							

Appendix 3: CM4025: Web System Sample Submission Template

Name and student number:

Submissions

Working URL	(insert link)
GitHub Repository	(insert link)
Video	(link or "submitted to Moodle")

Requirements: For each requirement, please enter in the table details about how your web system meets that particular requirement. This is your opportunity to clarify your web system.

Context	(e.g. "My website is a job recruitment site")
A/B tests	(e.g. "My website has different display settings to display the job listings. One setting has a green button to access the link, one setting has a white button.")
Test metrics	(e.g. "The number of user clicks are recorded.")
Experimenter Dashboard	(e.g. "The dashboard shows the total number of clicks for the green and white buttons for all time and for the last 24 hours.")
External Data/Database usage	(e.g. "I take the job adverts from a 3rd party API. I use my own databases to store usernames/passwords and to record clicks")
User accounts	(e.g. "I have user accounts for people to view my A/B test content and A/B metric dashboard)

Stack description (including frameworks and packages used)

Any special instructions for using your web system (including test user name and password if appropriate)

Any other comments or notes