You are expected to use the technologies introduced in this module. For example, you must use PHP for server-side programming. You can use the official Bootstrap library (https://getbootstrap.com/docs/4.5/getting-started/download). However, implementations using Bootstrap templates will not be marked. Data can be stored in text files using CSV, JSON or XML. Implementations using databases will not be marked.

Please test your application using a local web server, if possible. If you have to use the teaching web server, test your application using different folders to make sure that links in your application are relative and that your application works regardless of which folder structure is used for testing.

## Marking scheme for the website

The weighting for completion of each section of the tasks is indicated in Table 1. Across these components, marks will be awarded according to the scheme defined in Table 2, where possible.

**Table 1.** Required website components for CSC-20021 coursework (part one). Proportions in the right-hand column indicate the relative weighting of components **in this part** of the coursework.

| | |
|---|---|
| **1. UI/UX**: User interface and experience design, navigating between the pages that make up your website, links to other sites, and branding. | 25% |
| **2. Portfolio of the following selected tasks from practicals.** Extend and/or adapt your work from the related practicals. Make sure that the resulting web pages and resources are consistent with the rest of your web application.<br>**a. CV – 15%** (Practical 2): Your HTML-based CV as a Web page. Assume that this CV will be used to apply for a *web developer* position. **Do not** include personal details such as your names or photos. The page should include your computer science skills, achievements, modules you took (module names should appear as links to the Keele website and links should be working), the areas in which you would like to do your 3rd year project, and a profile summary. Please be creative and extend this page with more details.<br>**b. Video widget - 15%** (Practical 4): A video widget on the home page that can be controlled by the user without using the standard HTML5 controls (i.e., programming the interaction yourself). You will use the example video file from the practical.<br>**c. Login system - 10%** (Practical 6): Admin users can login and logout, and the authentication details are kept using the session. | 40% |
| **3. Online repository**: Develop an online repository to record and retrieve details of University specific events to make announcements for students and staff members. Events can be about guest lectures, conferences, sports activities and so on. Assume that each event takes place at one of the university buildings. Please be creative and use your own imagination. You can use text-based information to retrieve and display information. However, if you wish, please feel free to explore utilising map-based JavaScript frameworks to capture and/or display event details.<br>**a) submit.php - [15%]**: A web page for submitting new records. The page should present a form to collect details of each record. The details can include event title, organisers, date and time, location and so on. All the fields must be validated and the page must only be accessible by admin users.<br>**b) list.php [15%]**: A web page to list all events. Accessible to all users. In order to get 15%, this task will be implemented using AJAX to asynchronously retrieve event details. If AJAX is not used, 10% will be given.<br>**c) display.php [5%]**: A web page for displaying details of a single event. Accessible to all users. | 35% |

**Table 2.** Marking scheme for part one.

| | |
|---|---|
| Design, coding and mark-up quality | 40% |
| Functionality | 30% |
| Presentation quality (consistency of presentation, not design) | 20% |
| Coding comments | 10% |