|  |  |  |  |
| --- | --- | --- | --- |
| **Session** |  | **Semester** | Spring |
| **Module Name** | Databases and Interfaces | **Code** | COMP1004 |
| **Module Convenor(s)**  **(CW Convenor in Bold)** | **Milena Radenkovic** | | |
|  |  |  |  |
| **Coursework Name** | **CW2: Developing web front-end to databases** | **Weight** | 25% |
| **Deliverable**  (a brief description of what is to be handed-in; e.g. ‘software’, ‘report’, ‘presentation’, etc.) | 1. A cover page file containing your name, your student ID number, username and the url of your website (please use PDF format). 2. A file named your Username-Report which includes    1. a description and list of HTML, CSS, JavaScript, PHP and SQL features that you used in your code, and 3. A folder containing all your code. | | |
| **Format**  (summary of the technical format of deliverable, e.g.  “C source code as zip file”, “pdf file, 2000 word max”, “ppt file, 10 slides max”, etc. | Zip of a folder containing all your code. | | |
|  |  | | |
| **Issue Date** | See Moodle | | |
| **Submission Date** | See Moodle | | |
| **Submission Mechanism** | Moodle | | |
| **Late Policy**  **(University of Nottingham default will apply, if blank)** |  | | |
| **Feedback Date** |  | | |
| **Feedback Mechanism** | Moodle | | |
| **Instructions** | **The coursework builds upon the Movies and Actors database you created for your Coursework 1**.  **You will have to build a web front-end to the database using HTML, CSS and JavaScript that will connect to your database and execute queries using PHP and SQL.**  The functionality should include:   * **Adding a movie** to the database * **Adding an actor** to the database * **Searching for a movie** in the database * **Searching for an actor** in the database * **Deleting an actor** from the database * **Deleting a movie** from the database   Make sure that you describe and include as many features from the assessment criteria as possible to maximise your mark but also feel free to experiment and include features of your own choice. | | |
| **Assessment Criteria** | |  |  | | --- | --- | | **Criteria** | Design and implementation of an end-to-end system for database access. The code should include both basic and advanced Databases and Interfaces features and should be produced to professional standards. | | **Fail** | The code fails to execute and/or there is no evidence of HTML, CSS, JavaScript, PHP, SQL programming within the application. The assignment has not been attempted or solved. | | **3rd** | **The code includes basic** **HTML/CSS features, only: lists, anchors, images, paragraphs, tables, classes. Browsing through** **different pages works fine**. An attempt to solve the assignment has been made but the program executes with unexpected results. | | **2:2** | **Meeting previous requirements** **AND including forms. All form sub-elements should work fine. There should be non-trivial JavaScript-provided functionality for dynamic HTML and for form validation**. The code works and produces the expected results, but functionality is limited. | | **2:1** | **Meeting previous requirements AND implementing PHP features for connecting and updating** (e.g. editing various table entries) **the database and executing SQL queries including search functionality** (e.g. find a specific artist). The code produces the expected results, but it lacks the quality of a professional solution. | | **1st** | **Meeting previous requirements AND including high-quality code that is well-documented and tested**. **Excellent structure and code reusability will result in the highest marks**. The program produces the expected results for the assignment. | | | |