|  |  |  |
| --- | --- | --- |
|  | Faculty of Computing, Engineering and Science |  |

**Assessment Cover Sheet and Feedback Form** 2019-20

|  |  |  |  |
| --- | --- | --- | --- |
| Module Code:  IS3S664 | Module Title:  Advanced Internet and Mobile Computing | | Module Team:  Daniel Cunliffe, Phil Davies |
| Assessment Title and Tasks:  jQuery and Data sources | | | Assessment No.  1 |
| Date Set:  23-Sep-2019 13:00 | | Submission Date:  13-Dec-2019 23:00 | Return Date:  **31-Jan-20** |

**IT IS YOUR RESPONSIBILITY TO KEEP RECORDS OF ALL WORK SUBMITTED**

|  |
| --- |
| **Marking and Assessment** |
| This assignment will be marked out of 100%  This assignment contributes to 50% of the total module marks. |
| **Learning Outcomes to be assessed** (as specified in the validated module descriptor [https://icis.southwales.ac.uk/](https://icissafe.southwales.ac.uk/studentmodules/11463/studentmodulespecifications) ):  1) To demonstrate a critical understanding of the prevailing technologies associated with the development of web-based and mobile applications.  2) To be able to critically evaluate the various technological options available for diverse web-based and mobile development. |
| *Provisional mark only: subject to change and / or confirmation by the Assessment Board* |

|  |  |
| --- | --- |
| ***Part A: Record of Submission (to be completed by Student)*** | |
| **Extenuating Circumstances**  If there are any exceptional circumstances that may have affected your ability to undertake or submit this assignment, make sure you contact the Advice Zone on your campus prior to your submission deadline. | |
| **Fit to sit policy**:  The University operates a fit to sit policy whereby you, in submitting or presenting yourself for an assessment, are declaring that you are fit to sit the assessment. You cannot subsequently claim that your performance in this assessment was affected by extenuating factors. | |
| **Plagiarism and Unfair Practice Declaration:**  By submitting this assessment, you declare that it is your own work and that the sources of information and material you have used (including the internet) have been fully identified and properly acknowledged as required[[1]](#footnote-1). Additionally, the work presented has not been submitted for any other assessment. You also understand that the Faculty reserves the right to investigate allegations of plagiarism or unfair practice which, if proven, could result in a fail in this assessment and may affect your progress. | |
| **Intellectual Property and Retention of Student Work:**  You understand that the University will retain a copy of any assessments submitted electronically for evidence and quality assurance purposes; requests for the removal of assessments will only be considered if the work contains information that is either politically and/or commercially sensitive (as determined by the University) and where requests are made by the relevant module leader or dissertation supervisor. | |
| **Details of Submission:**  Note that all work handed in after the submission date and within 5 working days will be capped at 40%[[2]](#footnote-2). No marks will be awarded if the assessment is submitted after the late submission date unless extenuating circumstances are applied for and accepted (Advice Zone to be consulted). | |
| You are required to acknowledge that you have read the above statements by writing your student number(s) in the box: | Student Number(s): |

**Grading Criteria and Feedback**

|  |  |  |
| --- | --- | --- |
| **Marking Scheme** | **Marks Available** | **Marks Awarded** |
| Client & Server-side logic (PDO/XML processing & JSON) to display suitable HTML tables | **25** |  |
| Client & Server-side logic (PDO/XML processing & JSON) to append new element values to the original database table or XML file | **25** |  |
| Client-side logic (PDO/XML processing & JSON) to **search for specified contents** | **25** |  |
| Report | **25** |  |

**ASSESSMENT - DETAILED REQUIREMENTS**

There are two elements to this coursework, an implementation and a report.

**Implementation**

You are required to produce web based files to present a single HTML web page to a client browser demonstrating JavaScript/Ajax techniques associated with:

* PHP PDO processing;
* JQuery framework;
* XML & JSON;
* CSS.

The HTML page will access two data sources, which you will make available on the Web Server:

* MySQL database tables (One or more tables);
* XML files (One or more files) stored in a specific folder.

The HTML web page provides the user with the means to select the data source (database or XML). Having chosen the data source, the user is presented with a list identifying the available database tables or XML files. Selecting a specific database table or XML file must cause the display of the associated data as a HTML table on the same web page without refreshing the whole web page.

Drop-down list boxes are to be used to permit the user to select the data source and present the list of available database tables or XML files to the user. The latter drop-down list box contents must reflect the database tables or XML files presently available on the Web Server without further web development amendments i.e. available database table names and XML filenames must not be hardcoded into the web page. This information must be obtained at run-time after the web page has been loaded and the information must be returned as JSON.

The web page layout should be similar to the following:

Table (Only available once a database table or XML file is selected)

Data set Drop-down box (Only available once a data source is selected)

Insert functionality (Only available when table is visible)

Footer

Search functionality (Only available when table is visible)

Data source Drop-down box

Header

There is no specific single theme to the contents of the database tables and XML files. Do not design a web page which reflects a single theme such as films, sport or other single topic.

The database tables must comprise an autoincrement numeric field and one or more string fields. No other data types may be employed and each table must comprise a different number of fields and fieldnames.

The XML files must comprise zero or more of the same child elements. Each of these elements must comprise a number of further child elements e.g**. w3school’s CD catalog XML** file.

Essentially the database table and XML file structures described above represent zero or more rows of data in which the structure of each row is the same as every other row.

Upon receiving the returned JSON, the web page should construct appropriate HTML markup for rendering. You should not use TableSorter2 in this cw or similar components.

The contents of the data source drop-down list box is fixed but the drop-down list box presenting the list of database tables or XML files and subsequent table must be populated as necessary using JQuery techniques. All Client to Server interactions must utilise getJSON and all data returned by the Server must be in a **JSON format**.

The user must be provided with the means to:

* **view** a data set (database table or XML file, including field headings), once a table or file has been selected;
* **append** new data to the selected database table or XML file. This will require form input fields and php script logic to insert the new detail into the relevant database table or XML file.
* **search** the currently selected data set for a specific field’s content so that only those rows/records satisfying the selection are displayed in the HTML table.

This involves additional input fields and logic to allow the user to select the field and field content.

A drop-down list box must be presented permitting the user to specify the relevant single field/column.

You may present the results using a second HTML table or amend the original display table.

The web page logic should be robust and cater for a variety of situations e.g.:

* *No database tables being available.*
* *No XML files being available.*
* ***Table must only be displayed when a selection has been made.***

These are just examples of robustness and do not represent a complete list for the available marks. Details of the testing you perform should be included in your report.

You must not utilize older:

* **HTML DOM selection methods such as getElementById. You must use JQuery’s selector features.**
* ***Ajax techniques to retrieve the JSON returned by the php scripts. You must use JQuery’s $getJSON feature.***

Do not merely implement the overall solution and expect it to work. You are advised to test the individual components of your solution through the use of an appropriate browser’s URLs, using parameters as necessary e.g.

* Reference the php script ( no parameters required ) to obtain the JSON detail representing the available database table names or XML files to populate the drop-down list box. The script should return expected JSON content depicting the available database tables or XML files;
* Knowing individual php functionality works allows you to progress to implementing a drop-down list box in an HTML file.
* Reference the php script (parameter specifying drop-down list selection) to obtain the JSON detail representing the contents of the specified database table or XML file. The script should return expected JSON content representing the table/file details and contents.

**Report**

Your submission should be supported by a report detailing your software design, implementation and testing, including:

* Overall software design documentation;
* Commentary on JavaScript functionality;
* Commentary on PDO processing. An explanation of how your implementation caters for prohibiting the possibility of attempts at SQL Injection together with supporting evidence showing your implementation actually prohibiting such attempts;
* Commentary on XML processing;
* Evidence of testing using different browsers.
* All your code and details of database tables and XML files should be included as appendices

You should present the material in a standard expected of level 6 studies i.e. you must present arguments and critical evaluation demonstrating an understanding of the technologies and concepts applied.

**Submission details:**

**Implementation**

Your HTML page must be available on the student webserver at the following URL:

at-web2.comp.glam.ac.uk/students/*your-enrolment-number/*AIMC/index.htm

**Report**

A Word version of your report, including copies, must be uploaded to Blackboard.

You are not required to submit a paper based copy of your report

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Fail (0 – 29)** | **Narrow Fail (30 – 39)** | **3rd Class / Pass** | **Lower 2nd Class / Pass** | **Upper 2nd Class / Merit** | **1st Class / Distinction** |
| **Client & Server-side logic to display suitable HTML tables (25)** |  |  |  |  |  |  |
| **JSON PDO SQL** | Code not functional | Code partially functional  Code is not robust  Code does not use jQ, JS, JSON, PDO as required | Code functions under limited conditions  Code is robust under some conditions  JS used where jQ would have been the preferred approach  Interface could be improved | Code functions as expected  Code is robust under most conditions  Interface is reasonable | Code functions as expected  Code is robust  Interface is well designed | Code functions as expected  Code provides additional functionality  Code is robust  Interface is very well designed |
| **JSON XML** | Code not functional | Code partially functional  Code is not robust  Code does not use jQ, JS, JSON, PDO as required | Code functions under limited conditions  Code is robust under some conditions  JS used where jQ would have been the preferred approach  Interface could be improved | Code functions as expected  Code is robust under most conditions  Interface is reasonable | Code functions as expected  Code is robust  Interface is well designed | Code functions as expected  Code provides additional functionality  Code is robust  Interface is very well designed |
| **Client & Server-side logic to append new element (25)** |  |  |  |  |  |  |
| **JSON PDO SQL** | Code not functional | Code partially functional  Code is not robust  Code does not use jQ, JS, JSON, PDO as required | Code functions under limited conditions  Code is robust under some conditions  JS used where jQ would have been the preferred approach  Interface could be improved | Code functions as expected  Code is robust under most conditions  Interface is reasonable | Code functions as expected  Code is robust  Interface is well designed | Code functions as expected  Code provides additional functionality  Code is robust  Interface is very well designed |
| **JSON XML** | Code not functional | Code partially functional  Code is not robust  Code does not use jQ, JS, JSON, PDO as required | Code functions under limited conditions  Code is robust under some conditions  JS used where jQ would have been the preferred approach  Interface could be improved | Code functions as expected  Code is robust under most conditions  Interface is reasonable | Code functions as expected  Code is robust  Interface is well designed | Code functions as expected  Code provides additional functionality  Code is robust  Interface is very well designed |
| **Client-side logic to search for specified contents (25)** |  |  |  |  |  |  |
| **JS jQ** | Code not functional | Code partially functional  Code is not robust  Code does not use jQ, JS, JSON, PDO as required | Code functions under limited conditions  Code is robust under some conditions  JS used where jQ would have been the preferred approach  Interface could be improved | Code functions as expected  Code is robust under most conditions  Interface is reasonable | Code functions as expected  Code is robust  Interface is well designed | Code functions as expected  Code provides additional functionality  Code is robust  Interface is very well designed |
| **Code presentation** | Code layout is poor  No code comments | Code layout is poor  Code comments are poor | Code layout could be improved  Code comments could be improved | Code layout is generally good  Code comments are generally good | Code layout is good  Code comments are good | Code layout is excellent  Code comments are excellent |
| **Report (25)** |  |  |  |  |  |  |
| **Software design documentation** | No design documentation | Design documentation poorly executed  Design documentation partial  Design documentation not of sufficient standard | Design documentation contains reasonable and appropriate information  Design documentation is rather restricted in coverage  Design documentation uses a limited range of notations | Design documentation contains good, appropriate information  Design documentation has a good coverage  Design documentation uses a reasonable range of notations | Design documentation is well conceived and well executed  Design documentation is fairly comprehensive  Design documentation uses a good range of notations | Design documentation is thoughtful and shows both insight and attention to detail  Design documentation is comprehensive  Design documentation uses a wide range of notations |
| **Commentary on JS and jQ** | No commentary | Commentary poorly executed  Partial commentary  Commentary not of sufficient standard | Commentary could be written more effectively  Commentary would benefit from further detail  Commentary demonstrates a reasonable understanding of the code | Commentary is reasonably well written  Commentary has an appropriate level of detail  Commentary demonstrates a solid understanding of the code | Commentary is well written  Commentary has a good level of detail  Commentary demonstrates a good understanding of the code | Commentary is very well written  Commentary is very detailed  Commentary demonstrates an excellent understanding of the code |
| **Commentary on PDO, including SQL injection and demonstration** | No commentary  SQL injection not addressed | Commentary poorly executed  Partial commentary  Commentary not of sufficient standard  SQL injection poorly addressed | Commentary could be written more effectively  Commentary would benefit from further detail  Commentary demonstrates a reasonable understanding of the code  Consideration of SQL injection rather limited | Commentary is reasonably well written  Commentary has an appropriate level of detail  Commentary demonstrates a solid understanding of the code  Reasonable consideration of SQL injection | Commentary is well written  Commentary has a good level of detail  Commentary demonstrates a good understanding of the code  Good consideration of SQL injection | Commentary is very well written  Commentary is very detailed  Commentary demonstrates an excellent understanding of the code  Excellent consideration of SQL injection |
| **Commentary on XML processing** | No commentary | Commentary poorly executed  Partial commentary  Commentary not of sufficient standard | Commentary could be written more effectively  Commentary would benefit from further detail  Commentary demonstrates a reasonable understanding of the code | Commentary is reasonably well written  Commentary has an appropriate level of detail  Commentary demonstrates a solid understanding of the code | Commentary is well written  Commentary has a good level of detail  Commentary demonstrates a good understanding of the code | Commentary is very well written  Commentary is very detailed  Commentary demonstrates an excellent understanding of the code |
| **Testing, including browser testing** | No software testing  No browser testing | Software testing not of a sufficient standard  Browser testing not of a sufficient standard | Software testing could have been more comprehensive  Software testing could have been reported more effectively  Browser testing could have been more extensive  Browser testing could have been reported more effectively | Software testing reasonably thorough  Software testing reasonably well reported  Browser testing reasonably thorough  Browser testing reasonably well reported | Software testing is thorough  Software testing is well reported  Browser testing is thorough  Browser testing is well reported | Software testing is thorough and well-conceived  Software testing is very effectively reported  Browser testing is thorough and well-conceived  Browser testing is very effectively reported |
| **Appendices** | No appendices | Incomplete appendices | Appendices largely complete  Appendices could be presented more effectively | Appendices complete  Appendices reasonably well presented | Appendices complete  Appendices well presented | Appendices complete  Appendices presented very effectively |
|  | | | | | | |

1. University Academic Misconduct Regulations [↑](#footnote-ref-1)
2. Information on exclusions to this rule is available from the Advice Centre at each Campus [↑](#footnote-ref-2)