SCHOOL OF ENGINEERING AND COMPUTER SCIENCE
Computer Science
ASSESSMENT DESCRIPTION 2018/19
(EXAM TESTS WORTH ≤15% AND COURSEWORK)

MODULE DETAILS:

Module Number:	500087_A18	Trimester:	1
Module Title:	Information Systems and Web Technologies		
Lecturer:	Dr Bing Wang		

COURSEWORK DETAILS:

Assessment Number:	1	of	1		
Title of Assessment:	**A Web Location Interface Design**				
Format:	Program	Demonstration	3rd format		
Method of Working:	Individual				
Workload Guidance:	Typically, you should expect to spend between	60	and	120	hours on this assessment
Length of Submission:	This assessment should be **no** more than: *(over length submissions **will be** penalised as per University policy)*	N/A - coding exercise **words** *(excluding diagrams, appendices, references, code)*			

PUBLICATION:

Date of issue:	3/10/2018

SUBMISSION:

ONE copy of this assessment should be handed in via:	Canvas	If Other (state method)	Lab Demonstration
Time and date for submission:	**2pm, 6/12/2018 (week 14)**	**N/A**	**14/12/2018 (week 15)**
If **multiple hand–ins** please provide details*:*			
Will submission be scanned via TurnitinUK?	No	If submission is via TurnitinUK, these should be one of the allowed types e.g. Word, RT, PDF, PPT, XLS etc. Specify any particular requirements in the subumission details Students MUST NOT submit ZIP or other archive formats. Students are reminded they can **ONLY** submit **ONE** file and must ensure they upload the correct file.	

The assessment must be submitted **no later** than the time and date shown above, unless an extension has been authorised on a *Request for an Extension for an Assessment* form: search 'student forms' on https://share.hull.ac.uk.
Canvas allows multiple submissions: only the **last** assessment submitted will be marked and if

submitted after the coursework deadline late penalties will be applied.

MARKING:

Marking will be by:	Student Name

ASSESSMENT:

The assessment is marked out of:	20	and is worth	50	% of the module marks
N.B If multiple hand-ins please indicate the marks and % apportioned to each stage above (i.e. Stage 1 – 50, Stage 2 – 50). It is these marks that will be presented to the exam board.				

ASSESSMENT STRATEGY AND LEARNING OUTCOMES:

The overall assessment strategy is designed to evaluate the student's achievement of the module learning outcomes, and is subdivided as follows:

LO	Learning Outcome	Method of Assessment *{e.g. report, demo}*
2	*Demonstrate an understanding of key concepts and principles of web technologies and architectures.*	Program, Demo
3	*Analyse, specify and design a network information system with a dynamic web front-end, which is appropriate for a given purpose.*	Program, Demo

Assessment Criteria	Contributes to Learning Outcome	Mark
GET/POST functionality	2	4
Functionality of web pages	2,3	6
Database integration	3	3
Application robustness	2,3	4
Website presentation and innovation	2,3	3
Failure to attend your demonstration will result in your mark being limited to 50%.		

FEEDBACK

Feedback will be given via:	Verbal (via demonstration)	Feedback will be given via:	N/A
Exemption (staff to explain why)			
Feedback will be provided no later than 4 'teaching weeks' after the submission date.			

This assessment is set in the context of the learning outcomes for the module and does not by itself constitute a definitive specification of the assessment. If you are in any doubt as to the relationship between what you have been asked to do and the module content you should take this matter up with the member of staff who set the assessment as soon as possible.

You are advised to read the **NOTES** regarding late penalties, over-length assignments, unfair means and quality assurance in your student handbook, which is available on Canvas.

In particular, please be aware that:

- Up to and including 24 hours after the deadline, a penalty of 10%
- More than 24 hours and up to and including 7 days after the deadline; either a penalty of 10% or the mark awarded is reduced to the pass mark, **whichever results in the lower mark**
- More than 7 days after the deadline, a mark of zero is awarded.

- The overlength penalty applies to your written report (which includes bullet points, and lists of text. It does not include contents page, graphs, data tables and appendices). 10-20% over the word count incurs a penalty of 10%. Your mark will be awarded zero if you exceed the word count by more than 20%.

- Failure to attend your demonstration will result in your mark being limited to 50%.

Please be reminded that you are responsible for reading the University Code of Practice on Academic Misconduct through the Assessment section of the Quality Handbook (via the SharePoint site). This govern all forms of illegitimate academic conduct which may be described as cheating, including plagiarism. The term ‘academic misconduct’ is used in the regulations to indicate that a very wide range of behaviour is punishable.

In case of any subsequent dispute, query, or appeal regarding your coursework, you are reminded that it is your responsibility to produce the assignment in question.

Description of assessment task.

A Web Location Interface Design

You must build a web application with server side scripting on a full scale webserver. You must design a live website for recording locations and finding people.

Requirements

Write a web application for the student location service. In addition to storing the current location of each student, it must also keep track of each student’s previous locations. Your web application must implement the following tasks:

1) A database storing locations and details of students.

a. For each student, a first name and surname must be uniquely stored in the SQL database.

b. For each location, the database must store the new location and the date and time at which it was recorded. Old locations and timestamps should not be removed from the database.

2) In the root folder of your RDE website, there must be a page called ‘location.php’ OR ‘location.aspx’ which must accept GET and POST requests. GET requests to this web page must return the most recent location of the student in question. It must not return HTML as a result of a GET or POST request. POST requests to this page URL must allow a student’s location to be updated. On receiving a valid POST, your website must update the database record for that student. If the student does not exist in the database, they must be added. Note that PUT requests need not be supported.

3) In the **root folder** of your RDE website, there must be a single index page called **‘index.php’ or ‘default.aspx’** from which the following human-readable pages (which you must also create) can be accessed:

A page which

a. shows the current locations of all student
b. allows a user to change the location of a student
c. allows a user to edit the l details of a student
d. allows a user to choose a student and list their locations for the last 24 hours
e. allows the user to select from a list of all locations used by people and show a list of the people currently in the selected location

4) All data entry in your web pages must be validated to prevent the user entering invalid values.

Request formats

Your web application must respond to GET and POST requests sent in the standard HTTP request formats. Sample GET and POST requests are shown below (note that these are only examples to show the format – the data in them will need to be different for your system).

Typical POST request:

```
POST /123456/location.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: webdev.net.dcs.hull.ac.uk
Content-Length: 22

jsmith14=in+a+meeting
```

Typical GET request:

```
    GET /123456/location.php?student=jsmith15 HTTP/1.1
    Host: webdev.net.dcs.hull.ac.uk
```

Development tools

You may use either PHP or ASP.NET to implement your website. You may use javascript in your web pages. You may use the jQuery library if you wish, but you may not use any other third-party libraries (including Bootstrap) or controls other than those provided in the standard University installation of Visual Studio and RDE. You may develop your web site in any environment you choose, **but it must be deployed in your RDE web space and must use your RDE database.** All assessment will be based only on this installation and its execution on a laboratory machine. You must make sure that your web pages can be correctly displayed on different browsers (e.g Internet Explorer or Google Chrome).

Note that the presentation of your web pages does not carry very many marks. While you are free to style your pages how you wish, you are advised to use your limited time on ensuring your web site's functionality and robustness.

Database files

Your web application must store data in a database about the staff whose locations it is tracking. You must use your RDE SQL Server instance to store the database. At the very least, this database must store the user name, first name, surname, locations and times of update for each student. You may find it worthwhile to store other information as well.

Deliverables

By the submission deadline, you must submit via Canvas a single .zip file containing all files required to create your web site, with the correct folder structure. You must create the .zip file using the 7-Zip program as installed on the university machine. It must be possible to simply unzip this file into an empty web space folder to recreate your web site. Note that Visual Studio solution files are not required. Do not submit files in any other archive format than .zip.

You must also install on the RDE server a functioning version of the same files you have submitted via Canvas. The demonstration will be based on your files in the RDE server.

Important notes

1. HTTP server: You will be using a commercial web server (Microsoft Internet Information Server). You only need to write scripts and web pages which control its behaviour, as you have done in the laboratory tutorials. You may write these in ASP.NET or PHP. Both are supported by the server.

2. Precision: Some of the assessment will be assisted by automated tools. This simulates the situation in the real world where your software forms a component of a larger system, and the ‘user’ is in fact another software component. This means that tests will fail (and you will lose marks) if you do not follow the defined specification. Where the specification tells you to give a file a particular name, do as it says. Where a particular output or input format is defined, make sure that you use it. In the real world, failure to do so will mean that the software relying on yours will not be able to work. In the context of this ACW, it will mean that you will lose marks (even if your software “works”).
3. During assessment, you will be expected to be able to explain any part of the code to the assessor’s satisfaction.

4. It is not permitted to demonstrate you software using your laptop or home computer. The software must be demonstrated on a laboratory machine. Data must be stored in your RDE database, and your web pages must be located on your RDE web server.

Method

The following schedule of development is suggested and supported by the laboratory exercises:

Step1. Ensure you have created your RDE site and add a suitable index.html page and view it using the URL http://webdev.net.dcs.hull.ac.uk/yourID/. You should create some subsidiary pages and link them from the main page to check you can create and use web links that you will need later. Make a simple web enquiry form called location.html that does a GET request. This will be useful when you later test your PHP or ASP implementation. Note, that later you will need to remove or rename the index.html file when you create your index.php or Default.aspx files.

Step 2. Create an SQL database to store usersnames and the users locations on the RDE server, following the lab instructions. You may try adding to your basic HTML client to provide POST requests and better functionality, like selecting servers. This may require some Javascript.

Step 3. Create an location.php file to contain code to lookup a user in your SQL database using an HTML form to provide the data for a GET request, following the example from the lab instructions. Add the ability to update the information using a POST request, also with data from an HTML form.

Step 4. Create a location.aspx file to contain code to lookup a user in your SQL database using an HTML form to provide the data for the GET and POST requests, following the examples from the lab instructions. Don’t forget to install the necessary files from Visual Studio to your RDE folder location to ensure they operate properly.

Step 5. Decide if you want to complete your assessment using PHP or ASP now you have completed exercises in each. Remove or rename your initial index.html file and create an index.php or Default.aspx file, as appropriate for your chosen method. Expand your SQL database to store any extra information, such as update date and time and so on. Implement the subsidiary pages which permit the 5 specified enquiries to the database.

Step 6. Ensure your pages are installed properly, test the various functions of the different pages. You may find the location.html page that you made earlier useful in testing the GET and POST functionality. Bundle your web site into a ZIP and submit the assessment.

Assessment

The assignment will be assessed by demonstration in the lab. You will be notified of your demonstration time. You will be expected to show all the required functionality of your web site using only a web browser (without Visual Studio or any other IDE). Your web site will be assessed based only on the installation in your RDE web space using your RDE database. You may not demonstrate your software on anything other than a laboratory machine or using any other server.

Feedback

You will receive individual verbal feedback on your work when it is demonstrated.

Marking Sheet (An example)

(login 433598) Student no 200540200 John Smith

Administration

Attended demonstration		Y/N	□

Demonstration

POST functionality	Server accepts POST requests from command line client or via assessment client	2	□
GET functionality	Server returns standard response to GET request from command line client or web request.	2	□
Database integration	Server successfully updates RDE db with new location.	2	□
	New record is created after POST for unknown users.	1	□
Web pages	Index.php or default.aspx with links to other pages	1	□
	Page showing live location of all staff	1	□
	Page allowing editing of user data for a member of staff	1	□
	Page providing last 24 hours locations for a given student	1	□
	Page allowing selection of location from a valid list and showing all users at that location	2	□
Robustness	Application copes with (or prevents) bad data input e.g. duplicate IDs, long data, SQL injection)	2	□
	Application does not fail (lose 1 for each failure)	2	□
Innovation and presentation	Application has innovative features and is a delight to use. Pages can be shown correctly on different browers.	3	□
Demonstration total		20	□

Student signature...

Assessor signature...