ECS417U
Fundamentals of Web Technology
Assessment Guide - COVID-19 Update

Academic Year 2019/20

Module Team:
Dr Usman Naeem
Dr Jeremy Gow

School of Electronic Engineering and Computer Science
Queen Mary University of London
Mile End Road
London
E1 4NS

Queen Mary
University of London

Contents

Revised Assessment Breakdown

The original close-book written exam will not run as the learning outcomes of this module will be fully covered by the coursework (weekly labs and mini project). The weightings of coursework (originally 50%) have been adjusted as follows:

- **Weekly labs - 14% (remains the same)**
- **Mini project assessment - 86% (originally 36%)**

The mini project assessment has been modified to cover the remaining learning outcomes in light of the removal of the exam. This mini-project will now have a weighting of 86% of the final module mark.

The assessment breakdown, status and deadlines for the weekly lab exercises and the individual mini project are as follows:

Lab Exercises Assessment	Assessment % Breakdown	Deadline
HTML Basics	3%	Assessed
CSS Basics	3%	Assessed
HTML Tables and Forms	3%	Assessed
Advanced CSS	3%	Assessed
Content Creation	2%	Assessed
Mini Project Assessment	86%	Submission - 20th May 2020 Presentations - TBC
Assessment Total	**100%**	

Weekly Lab Exercises - 14%

All of the weekly lab exercises have now been assessed.

Individual Mini Project - 86%

In this mini project, you will need to create a portfolio website that will showcase your skills to potential employers. The site will need to include a simple tool for writing and reading a weblog (blog). One user, the blogger (yourself), should be able to add text entries to the blog. The most recent entry should appear at the top of the web page, followed by the next most recent, and so on for all entries. Links elsewhere on the page should provide access for the blogger to log in and add new entries.

To complete this project, you will need to work on the mini project on a weekly basis, as the content you cover in the lab exercises will enable you to build your portfolio website progressively over the 12 weeks of this module. Your portfolio website will need to be developed using the web technologies covered in this module. Specifically, *HTML5* and *CSS* will be used mainly to create structure and layout of the website pages. *JavaScript* will be used to achieve extended validation for the forms on your site. *PHP* will be the server-side programming language for access control, basic database connectivity and generating dynamical web pages for the blog.

Additional Requirement

In addition to the portfolio site, you will need to submit a ***short report*** (via QMPlus) that addresses the following:

Semantic Markup

What semantic HTML5 elements have been used within your portfolio site? Why is semantic markup important? You will need to provide examples related to your project to illustrate your answer. (250 words max.)

Web Technologies/Frameworks

For this project you have made use of *HTML*, *CSS*, *JavaScript* and *PHP*. If you were to conduct this project again, which other web technologies/frameworks would you use to develop your portfolio site. Justify your choice. (250 words max.)

In order to write this section, you need will undertake research about other web technologies/ frameworks.

Note: Failure to adhere to the specified world limits will lead to sections of your report not being marked.

This mini project accounts for **86%** of the final mark for this course.

Submission Instructions

In order to get your mini project assessed, please ensure that you follow the steps below:

1. Deploy your portfolio website onto the OpenShift platform.

2. Share your GitHub repo with Usman Naeem and Jeremy Gow. The GitHub usernames are: **drunaeem** and **jeremygow**.

3. Submit the URL of your portfolio, GitHub repo and the short report by **4.00pm (GMT)** on **20/05/2020**. This will involve the following steps:
   a. Go to ECS417 QMPlus site.
   b. Click the 'Individual Mini Project - SUBMISSION LINK' link.
   c. Scroll the page down and click add submission.
   d. Upload and submit the report. This submission has a Turnitin requirement, hence it will be checked for plagiarism and collusion.
   e. Type the following in the 'online text' field box.
      - *OpenShift URL of your mini project*
      - *GitHub repository URL*
      - *Username for your Blog*
      - *Password for your Blog*
   f. Click 'Save changes'

4. The practical element of the mini-project will be assessed remotely via Microsoft Teams. The presentation schedule will be confirmed on QMPlus.

5. Check the 'Mini Project Schedule' on QMPlus:
   - This will tell you the exact time you will be assessed.

6. Be ready 15 minutes prior to your allocated assessment slot.
   - This is to ensure that you are ready to be assessed on time.

7. Assessors will only have 10 minutes to assess your work, so please ensure that you have your mini project site and source files to hand. As the assessors will stop marking once the assessment time is up.

You must have a working website to show during the assessment on the OpenShift application platform (e.g. which should access your GitHub repository). You must be able to explain your code line by line. If you are unable to explain your code to the assessor then you will be given a mark of **zero**.

Portfolio

A portfolio is seen as the new 'CV', as having a portfolio is considered just as valuable. This is because it demonstrates to potential employers your skills through projects that you have worked on. This is something that is vital for students within the area of Electronic Engineering and Computer Science, as throughout your degree programme you get many opportunities to demonstrate your skills on a wide range of coursework assessments and projects. Hence, an online portfolio is a fantastic platform to provide evidence of your skills acquired throughout your degree programme.

This mini project will expect you to create a portfolio website, which should provide (as a minimum) the following information:

- About Me - short biography of yourself
- Skills and Achievements
- Education and Qualifications
- Experience
- Portfolio - links and description/evidence of projects you have worked on
- Contact - your contact details
- Blog

The web pages for your website should demonstrate the use of HTML5 semantic structure elements, such as `<header>`, `<hgroup>`, `<nav>`, `<article>`, `<section>`, `<figure>`, `<figcaption>` and `<footer>`. Figure 1 shows an example of the semantic structure that should be evident within your web pages.

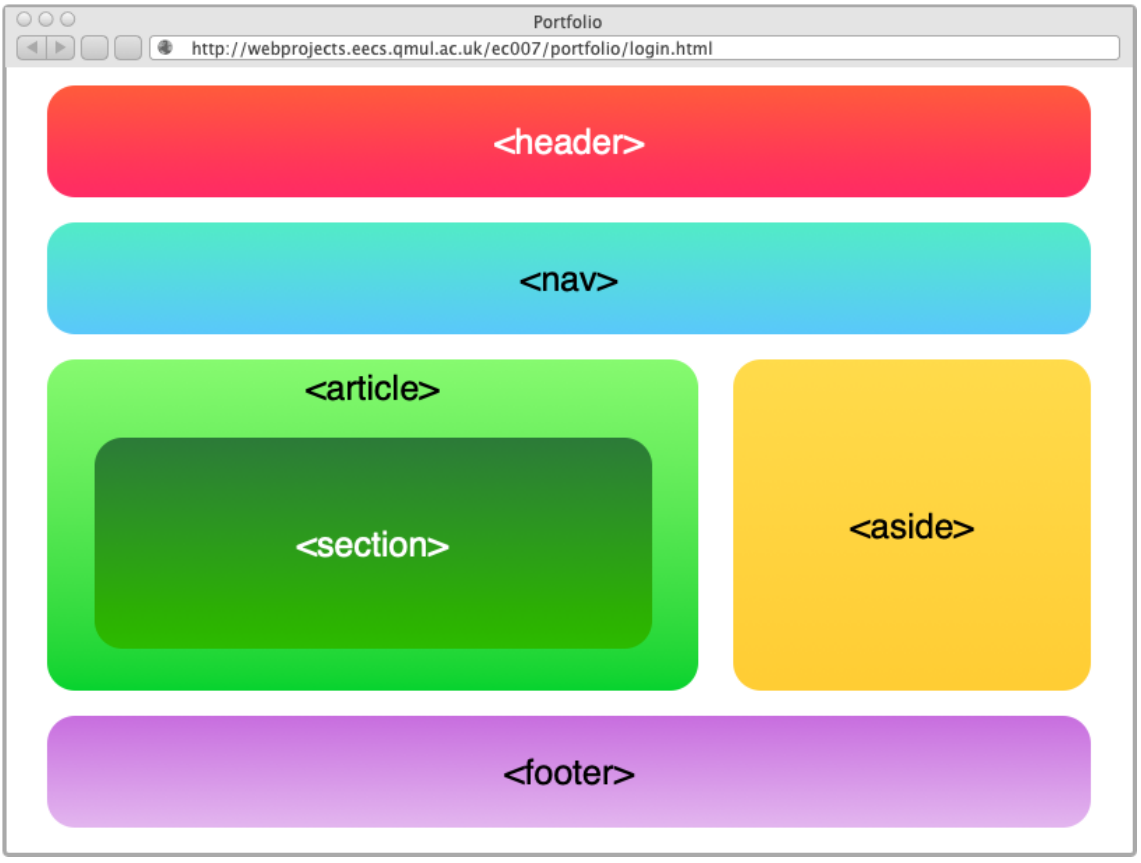

FIGURE 1

Blog

Another important aspect of this mini project is the development of a blog on your website. Like a portfolio, a blog provides you with an opportunity to stand out in a crowded job market. A blog can offer the following benefits:

- *Enhances professional profile*: A blog provides potential employers with proof regarding your expertise. Hence your blog should be current and related to your industry, as posting random blogs can have a negative impact.
- *Compliments your CV and portfolio*: A blog can add context to the information within your CV, as it provides you with a platform to show how you have applied your skills and knowledge.
- *Establishes a digital footprint*: A blog can enhance your prospects, as many recruiters tend to perform a Google search on an applicant's name. Hence, seeing a results page which has your blog and portfolio can increase the chances of getting an interview.

Blog Functionality

To post a blog on your website, you will need to create an authentication system that will require the user to log in when posting a blog. The authentication system will need to remember the valid login using sessions and also provide a link for the user to log out e.g. `logout.php`. You can include the login form within the `<aside>` element, which defines the section for additional content. Alternatively, you could also create a separate page for this, e.g. *`login.html`*. Once logged in, you should use the `<aside>` element to confirm the status of the user being logged in with a message displayed on the screen such as "*Welcome User*". Figure 2 shows an example of the login form that will need to be created for your website.

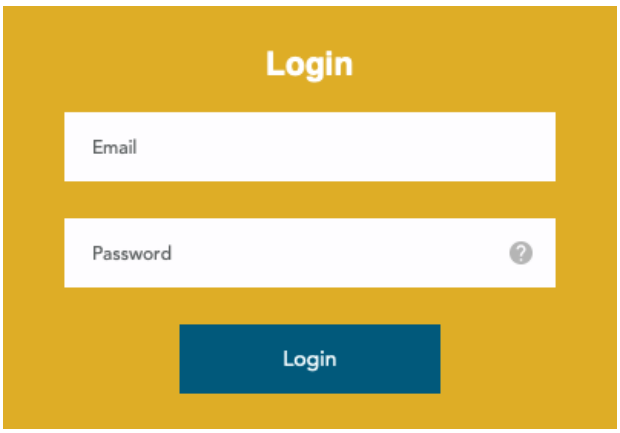

FIGURE 2

The user will input their email address and password, which will need to be validated with built-in HTML5 form elements. Once the form validation has been completed, the submitted information from the form will need to be processed by a PHP script named *`login.php`*. This PHP script will need to handle the submitted form by validating the credentials against information in a database. If the login is successful then *`addPost.html`* will be displayed, which should include a form like figure 3.

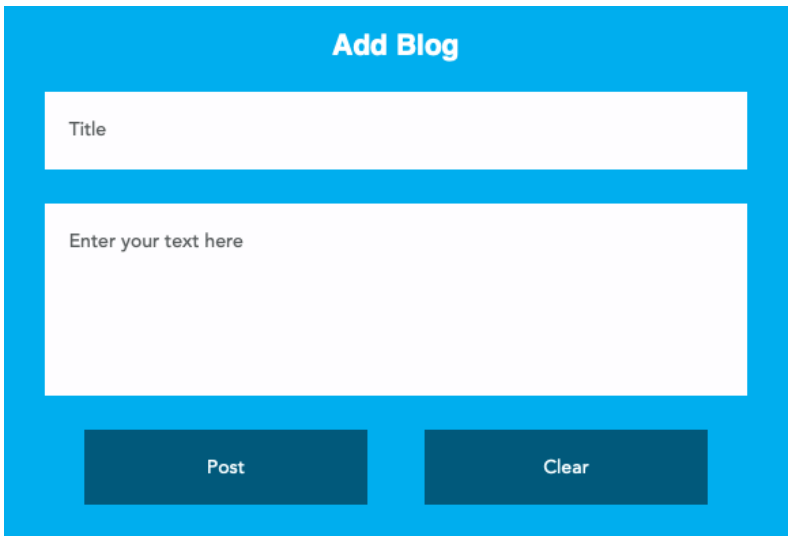

FIGURE 3

There are two buttons: “Post” and “Clear”. If the “Clear” button is clicked, a message window should pop up (alert box) and ask the user to choose between “OK” or “Cancel”. If “OK” is chosen, the inputs in the text box and text area will be cleared. If “Cancel” is clicked, then the content won’t be cleared. This is to prevent the user from clicking the “Clear” button by mistake (intending to click “Post”) and lose all the input.

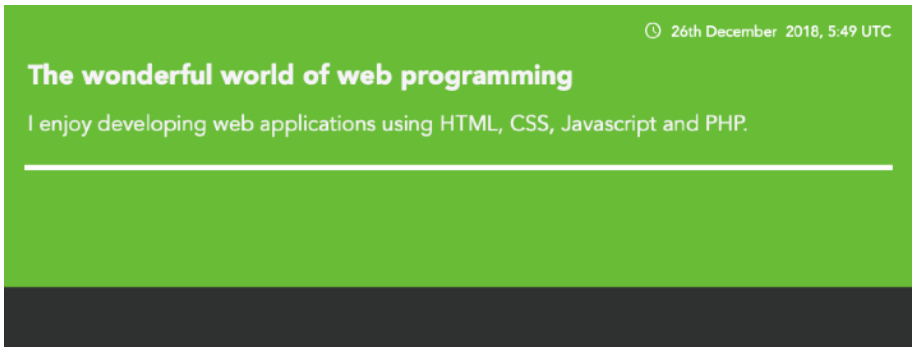

FIGURE 4

After inputting a title and blog content in the text box and text area, the user will click the “Post” button. The input data will then need to be processed by a PHP script named `addPost.php`. This script should save the post into a simple one table database and then redirect the page to another PHP document named `viewBlog.php`. The output of this page should like figure 4.

The main functionality of `viewBlog.php` is to display all the blog posts stored in the MySQL database on the server. This page will be accessed from your homepage via the Blog link within the `<nav>` element.

Each blog entry must include three components:

- The date and time when the post was added. It must follow the same format as shown in figure 4.
- A title
- The post

Different entries are separated using horizontal rules. There should be a CSS style rule dedicated to each of the three components making them distinguishable (e.g., different font sizes/colours).

The `viewBlog.php` should allow the user to add a new post by clicking the “Add Post” link. This will take the user to the login page (if not logged in) and the procedure to add a new post is identical to that of adding the first post to the blog. Figure 5 shows an example of a blog with multiple posts:

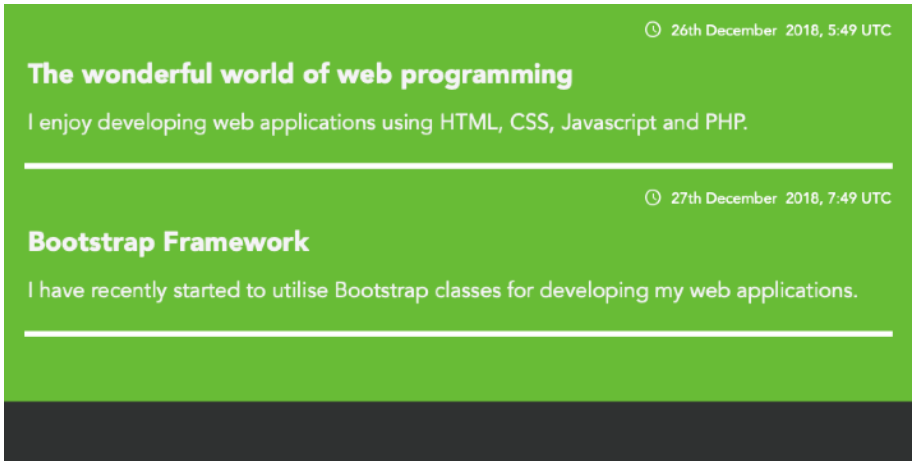
26th December 2018, 5:49 UTC
The wonderful world of web programming
I enjoy developing web applications using HTML, CSS, Javascript and PHP.
27th December 2018, 7:49 UTC
Bootstrap Framework
I have recently started to utilise Bootstrap classes for developing my web applications.

FIGURE 5

Summary
The layout of the website should be created using CSS.

To summarise, the following are key documents for creating the blog (please make sure to use the suggested file names):

- **`index.php`** - loads the home page for your website.
- **`viewBlog.php`** - displays blog post stored in your table within the MySQL database on your server, and redirects the user to `login.html` if there is no entry.
- **`login.html`** - asks the user to input an email address and password (Alternatively, this can also be included with index.php within the `<aside>` element).
- **`login.php`** - checks the username and password by validating the credentials against information in your database. If login is successful then the script redirects to `addPost`; otherwise, display an error message.
- **`logout.php`** - Resets all the $_SESSION variables and redirects to the homepage.
- **`addPost.html`** - asks the user to post a blog entry.
- **`addPost.php`** - adds a new post to a simple table within a MySQL database and redirects to `viewBlog.php`.

Requirements

The minimum requirements for the portfolio site are as follows:

HTML5	
1	Create the pages of website using the HTML5 semantic structure elements , such as `<header>`, `<hgroup>`, `<nav>` , `<article>`, `<section>` , `<figure>`, `<figcaption>` and `<footer>`.
2	For login and `addpost`, create the required forms with HTML5 form elements and align elements.
3	Use HTML5 form elements that allow validation of the fields within the form for login.

CSS	
4	Create an external style sheet called `reset.css` that removes all the browser formatting from the main HTML elements and reference.
5	Create an external style sheet that specifies the styling rules for the HTML elements. You strongly encouraged to be creative with the styles for your website. Be sure to group your style rules together in appropriate commented sections and to make your sizes scalable.

JavaScript	
6	Event processing for clicking the “Clear” button in `addPost.html`.
7	Ensure that the code prevents submission of the form in `addPost.html` (`preventDefault()`) if the fields title and post are left blank. The missing fields will need to be highlighted using CCS styles.

PHP	
8	For the login form, your script will need to handle the submitted form by validating the credentials against the information in your database.
9	Upon successful login, you script will need to start a session.
10	Create script for ending the session, e.g. `logout.php`.
11	For posting, your script will receive the input from the form in `addpost.html`, which then inserts this data into a MySQL database. This information will include date, time, title and body text of the post.

Extra Features

This mini-project is meant to be an open-ended project. After finishing the above-mentioned basic functionalities, you could add an extra feature using either techniques covered in the module or those you've taught yourself.

Extra features include:

1. Organising your blog entries into different months. Provide a drop-down menu for the blog viewers to view the entries of different months stored in an entry archive.

2. Allowing blog viewers to log in and add comments to the entries. You, as the administrator of the blog should be able to delete entries or comments.

3. Adding a "preview" button in `addpost`. When this button is clicked, the new entry is previewed and you can then decide whether to upload the entry or go back to edit it.

Useful websites

For redirection with PHP:
http://php.net/manual/en/function.header.php
http://php.about.com/od/learnphp/ht/phpredirection.htm

If you can't finish all the functionalities, don't panic. Do as much as you can and you will get marks for your efforts.

Please note that plagiarism cases will be dealt with seriously.

Individual Mini Project Marking Criteria

Assessment Criteria 1
Website pages created using HTML5 semantic structure elements

Marks breakdown

0 mark	No evidence of elements
2.5 marks	One to four elements
5 marks	More than five elements

Assessment Criteria 2
Forms created with HTML5 form elements

Marks breakdown

0 mark	No evidence of elements
2 marks	One form created
4 marks	Both forms created

Assessment Criteria 3
HTML5 form elements used for validation of the fields in the login form

Marks breakdown

0 mark	No evidence of validation using HTML5 form elements
2 marks	Validation of the fields carried out using HTML5 form elements

Assessment Criteria 4
Create an external style sheet called reset.css that removes all the browser formatting from the main HTML elements and reference

Marks breakdown

0 mark	No evidence of resetting elements
1 mark	Reset one to five elements
2 mark	Reset six to ten elements
2.5 marks	Reset eleven to fifteen elements
3 marks	Reset sixteen to nineteen elements

Assessment Criteria 5
Create an external style sheet that specifies the styling rules for the HTML elements

Marks breakdown

0 mark	No style rules
1 mark	One to six styles rules
2 marks	Seven to twelve styles rules
3 marks	Thirteen to eighteen styles rules
4 marks	Nineteen to twenty-four styles rules

Assessment Criteria 6
Event processing for clicking the “Clear” button in addPost.html

Marks breakdown

0 mark	Not working
1 mark	Function created in JavaScript but not working
2.5 marks	Partially functional
4 marks	Fully functional

Assessment Criteria 7
Ensure that the code prevents submission of the form in addPost.html (preventDefault()) if the fields title and post are left blank. The missing fields will need to be highlighted using CCS styles

Marks breakdown

0 mark	Not working
1 mark	Function created in Javascript but not working
5 marks	preventDefault function working, but missing fields were not highlighted using CSS
10 marks	Fully functional

Assessment Criteria 8

For the login form, your script will need to handle the submitted form by validating the credentials against the information in your database. If correct then redirect to the addPost page using PHP

Marks breakdown

0 mark	Not working
4 marks	Partially functional
8 marks	Fully functional

Assessment Criteria 9

Upon successful login, you script will need to start a session

Marks breakdown

0 mark	Not working
2.5 marks	Partially functional
5 marks	Fully functional

Assessment Criteria 10

Create script for ending the session, e.g. logout.php

Marks breakdown

0 mark	Not working
2.5 marks	Partially functional
5 marks	Fully functional

Assessment Criteria 11

For posting a blog entry, your script will receive the input from the form in addPost.html, which then needs to be inserted into a MySQL database. This information will include date, time, title and body text of the post. This script should then redirect to viewBlog.php

Marks breakdown

0 mark	Not working
4 marks	Partially functional
10 marks	Fully functional

Assessment Criteria 12
Each time a new entry is added, the most recent post appears on top, followed by the next most recent post using PHP

0 mark	Not working
4 marks	Partially functional
10 marks	Fully functional

Assessment Criteria 13
Implement extra features specified in the coursework specification

Marks breakdown

0 mark	Not working
10 marks	Implemented one fully functional feature
20 marks	Implemented two fully functional features

Assessment Criteria 14
Is the website content relevant for a portfolio website?

Marks breakdown

0 mark	Not relevant
5 marks	Partially relevant
10 marks	Relevant

Assessment Criteria 15 - Short Report

Identification of semantic HTML5 elements

Marks breakdown

0 mark	Non existent
2.5 marks	Partially identified
5 marks	Student has clearly identified the semantic HTML5 elements within the portfolio site

Assessment Criteria 16 - Short Report

Importance of semantic markup

Marks breakdown

0 mark	Non existent
2 marks	Partially highlighted the importance
4 marks	Student has identified the importance of semantic markup
5 marks	Student has identified the importance of semantic markup with examples

Assessment Criteria 17 - Short Report

Web technologies/framework

Marks breakdown

0 mark	No mention of potential web technologies/framework
2 marks	Partial mention of potential web technologies/framework
4 marks	Student has identified potential web technologies/framework
5 marks	Student has identified potential web technologies/framework with clear and appropriate justification

Assessment Criteria 18 - Short Report

Quality of report

Marks breakdown

0 mark	The report is unclear and not concise (above the word-limit)
2.5 marks	Adequate write-up, lacking clarity in places, or containing irrelevant material
5 marks	Logical structure, good flow and concise style