ECS417U
Fundamentals of Web Technology
Assessment Guide

Academic Year 2020/21

Module Team:
Dr Usman Naeem
Dr Sukhpal Singh Gill

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

Queen Mary
University of London

Contents

Assessment Breakdown

There are two assessment components for this module:
001 - Weekly Lab Exercises - 20%
002 - Mini Project Assessment - 80%

The assessment period/deadlines for the weekly lab exercises and the mini project are as follows:

Date	Weeks to complete	Topic No.	Lab Exercises - Assessment	Assessment % Breakdown	Assessment Period/Location
Every Week			Weekly Questions - QMPlus	2%	Deadline: Every Friday - 10.00am QMPlus
02/02/2021	2	1	Setup GitHub and OpenShift	2%	Week 2 - QMPlus Submission only
02/02/2021	2/3	2	HTML Basics	3%	Week 2, 3, **4** - Lab Session
09/02/2021	3/4	3	CSS Basics	3%	Week 3, 4, **5** - Lab Session
16/02/2021	4/5	4	HTML Tables and Forms	4%	Week 4, 5, **6** - Lab Session
23/02/2021	5/6/7	5	Advanced CSS	4%	Week 5, 6, 7, **8** - Lab Session
02/03/2021	6	-	Content Creation - QMPlus	2%	Week 6 - QMPlus Submission only
Assessment Component 001 Total				**20%**	
23/03/2021	9	Mini Project	Front End Assessment	20%	Week 9 - Lab Session
13/04/2021	12	Mini Project	Report - Submission QMPlus	10%	Week 12 - QMPlus 13/04/2021 - 10.00am
			Client and Server Side Assessment	50%	Week 12 - Lab Session
Assessment Component 002 Total				**80%**	
Module Assessment Total				**100%**	

You must achieve an overall module mark of 40% in order to pass this module.

Weekly Lab Exercises - 20%

Weekly Lab Exercises (18%)
For this assessment component, you will need to complete a series of weekly lab exercises during the first eight weeks of this module. These tasks will be based on the practical content introduced in the lecture sessions. The marking scheme for these exercises will be included within the lab exercise sheets, which are all available on QMPlus.

Submission and Assessment Instructions

The assessment of your lab exercises will take place during your allocated lab sessions. You will have a two week window to get your weekly lab exercises assessed by the module team (except for the OpenShift lab). For example, the assessment period for topic 2 lab exercise will be until 16th February 2021 (during your allocated topic 4 lab session). You will need to make sure that you get your work assessed within the **first one and a half hour of your lab session**, as your work will not be assessed after this period. Hence, it is strongly recommended that you don't leave the assessment of your lab exercises to the last moment, as the module team will not be able to assess your work. So if you do not get assessed within this time frame, then you will get a zero for that weekly exercise. Therefore, **once you complete the weekly exercise you should get it assessed ASAP.**

In order to get your weekly lab exercises assessed, you will need to follow these steps:

1. Create a folder in your local Git repository. This should be named after the topic that will be assessed, e.g. folder name for topic 2 should be 'topic2'.
2. Deploy your folder (should include the code for your weekly exercises) to the OpenShift platform by pushing it to your Git repository.
3. Submit the URL for your exercises via the dedicated submission link on QMPlus.
4. The assessment of these exercises will be conducted on *Microsoft Teams*. So please ensure you have downloaded *Microsoft Teams* and understand how to share your screen with another user. Perhaps, you can test out screen sharing with one of your friends.
5. Complete the online contact form (link on QMPlus) to initiate contact with an assessor, who will then contact you via *Microsoft Teams*. Hence, can you please ensure that you are logged in and ready to be assessed.
6. During this assessment you will be required to turn on your camera for verification purposes. So please ensure you have your ID card to hand, so that the assessor can verify your identity.

Note that you can only have your assessment conducted during your allocated lab session. For example, if you are assigned to the 11.00-13.00 session, then you cannot have your work assessed in the 13.00-15.00 session.

If you have an Extenuating Circumstance (EC) that prevents you from submitting the assessment, then you will need to complete a form via the following link:
https://www.welfare.qmul.ac.uk/guides/extenuating-circumstances/how-do-i-make-extenuating-circumstances-claim/

Weekly Review Questions (2%)
In addition to the lab exercises, you will also be expected to watch a series of weekly videos on QMPlus (before or after the lecture), which will include review questions that you will need to answer by the end of that week. The purpose of this is to assess if you are engaging and understanding the module content. This is worth **2%** of the weekly lab exercises assessment component (**20%**).

Mini Project Assessment - 80%

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 sessions 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.

The assessment for the mini project will be conducted in two phases:

Phase 1 - Front End (20%)
The module team will assess the HTML5 and CSS aspects of the mini project, which will be based on the structure and layout of your portfolio website. This assessment will be conducted on **23/03/2021**, and you will need to submit the URL of your portfolio by **23/03/2021 - 11.00AM (UK time)**.

Phase 2 - Client and Server Side Processing (50%)
The module team will assess the JavaScript and PHP aspects of the mini project, which will be based on client side validation and server side programming for access control, database connectivity and dynamic web pages. This assessment will be conducted on **13/04/2021**, and you will need to submit the URL of your portfolio by **13/04/2021 - 10.00AM (UK time)**.

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. Like the weekly lab exercises, you will be required to turn on your camera for verification purposes. So please ensure you have your ID card to hand, so that the assessor can verify your identity.

Report (10%)
In addition to the portfolio site, you will need to submit a 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.)

The submission deadline for this report is **13/04/2021 - 10.00AM (UK time)**.

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

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.

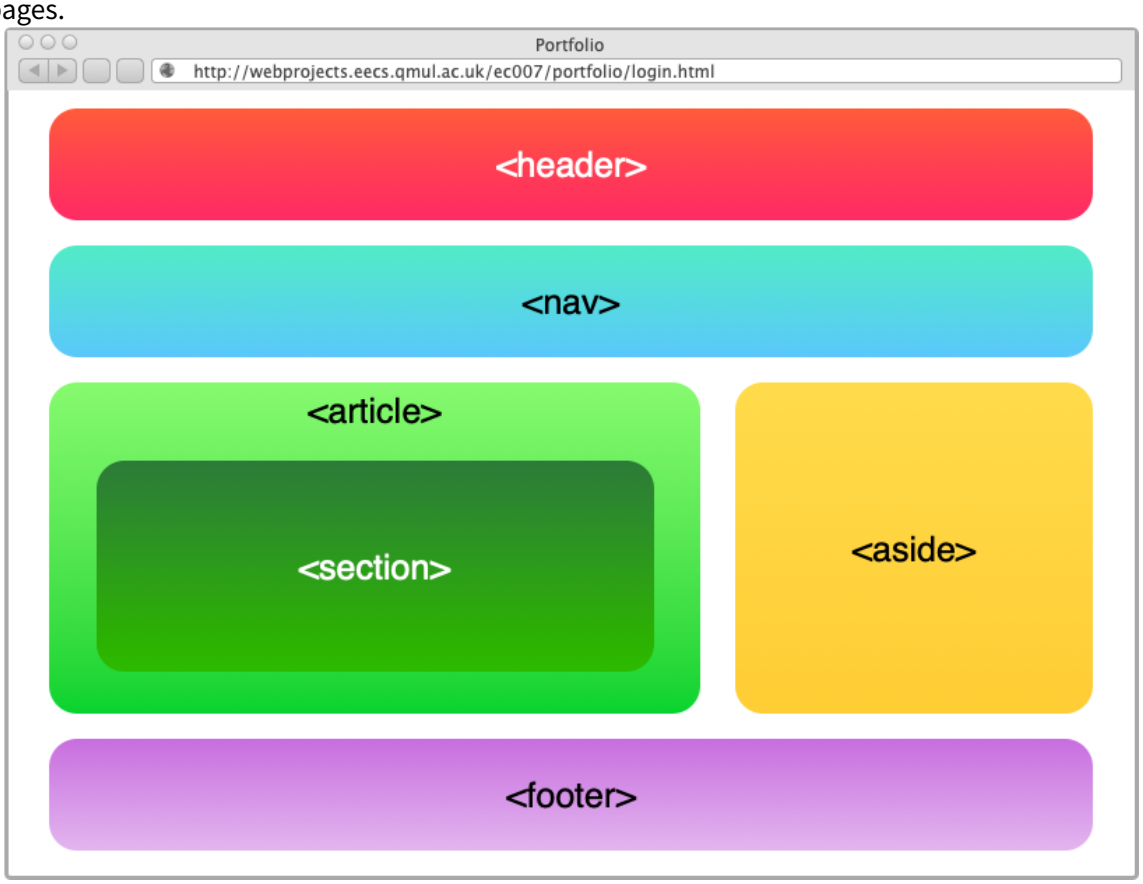

FIGURE 1

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.

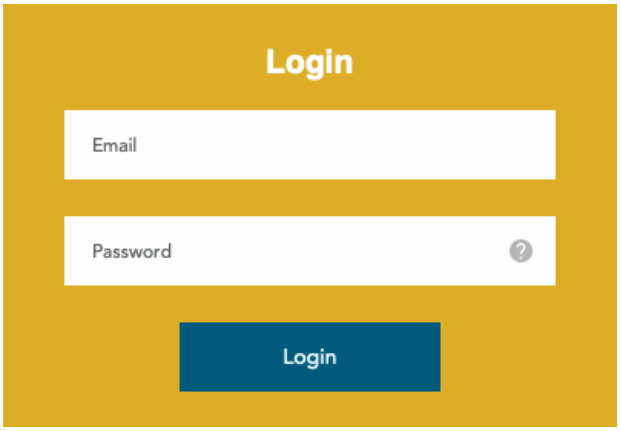

FIGURE 2

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 *`addEntry.php`* will be displayed, which should include a form like figure 3.

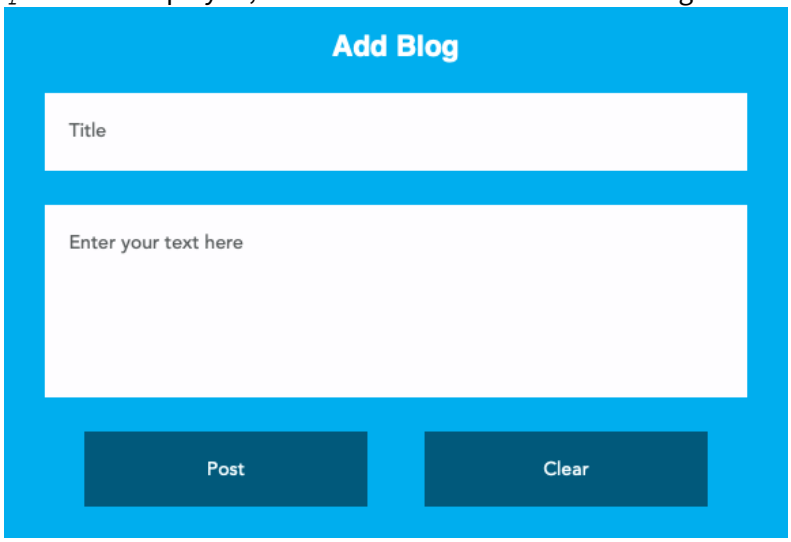

FIGURE 3

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.

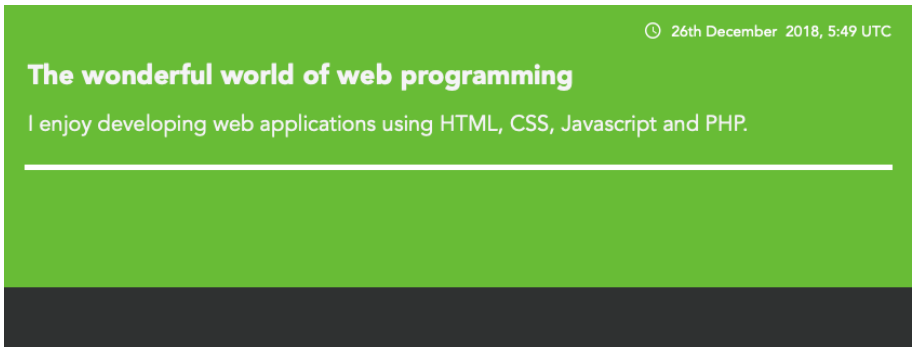

FIGURE 4

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 table within a database (which you created in topic 9) 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:

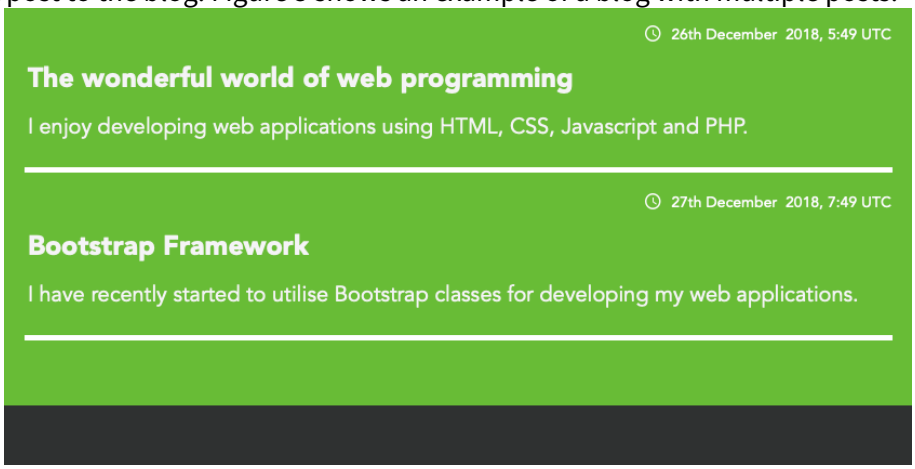

FIGURE 5

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

To summarise, the following are key documents (suggested filenames) for creating the blog:

- **`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.
- **`addEntry.php`** - 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:

Phase 1 Assessment

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.

Phase 2 Assessment

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 are limited to:

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.

Mini Project Submission Instructions

Phase 1 Submission

The submission deadline for phase 1 of the mini-project is Tuesday 23rd March 2021 at 11.00AM (UK time).

In order to get assessed, you will need to follow the submission steps below:

1. Deploy your portfolio website onto the OpenShift platform by pushing it to your Git repository.
2. Share your GitHub repo with Usman Naeem and Sukhpal Singh Gill. The GitHub usernames are: **drunaeem** and **iamssgill**.
3. Submit the URL of your portfolio and GitHub repository by 11.00am (UK time) on 23/03/2021. This will involve the following steps:
   (a) Go to ECS417U QMPlus site.
   (b) Click the '*Phase 1 - Front End Assessment - 20%*' link.
   (c) Scroll the page down and click add submission.
   (d) Type the following in the 'online text' field box.
   - *OpenShift URL of your mini project*
   - *GitHub repository URL*

   (e) Click 'Save changes'

Phase 1 Assessment

The practical aspect (demonstration) of your phase 1 submission will take place on Tuesday 23rd March 2021. An assessment schedule will be available on QMPlus.

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

1. Check the GitHub username of your assessor in the 'Phase 1 Practical Assessment Schedule' spreadsheet and share your repo with them.
2. The practical element of the mini-project will be assessed remotely via *Microsoft Teams*.
3. Your assessor will contact you via Teams, so please ensure you are logged in.
4. Be ready **15 minutes** prior to your allocated assessment slot.
   - This is to ensure that you are ready to be assessed on time.
5. 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.
6. During this assessment you will be required to turn on your camera for verification purposes. So please ensure you have your ID card to hand, so that the assessor can verify your identity.

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**.

Phase 2 Submission

The submission deadline for phase 2 of the mini-project is Tuesday 13th April 2021 at 11.00AM (UK time).

In order to get assessed, you will need to follow the submission steps below:

1. Push your changes (phase 2 updates) for the portfolio website to the OpenShift platform.
2. Submit the URL of your portfolio and GitHub repo by **11.00AM (UK time)** on **13/04/2021**. This will involve the following steps:
   (a) Go to ECS417U QMPlus site.
   (b) Click the '*Phase 2 - Client and Server Side Processing Assessment - 50%*' link.
   (c) Scroll the page down and click add submission.
   (d) 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*

   (e) Click 'Save changes'

Phase 2 Assessment

The practical aspect (demonstration) of your phase 2 submission will take place on Tuesday 13th April 2021. An assessment schedule will be available on QMPlus.

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

1. Check the GitHub username of your assessor in the 'Phase 2 Practical Assessment Schedule' spreadsheet and share your repo with them.
2. The practical element of the mini-project will be assessed remotely via *Microsoft Teams*.
3. Your assessor will contact you via Teams, so please ensure you are logged in.
4. Be ready **15 minutes** prior to your allocated assessment slot.
   - This is to ensure that you are ready to be assessed on time.
5. 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.
6. During this assessment you will be required to turn on your camera for verification purposes. So please ensure you have your ID card to hand, so that the assessor can verify your identity.

You must have a working website to show during the assessment on the OpenShift application platform (e.g. which should access your GitHub repository). If you have not deployed your project onto OpenShift then we will only be able to assess the JavaScript aspect of your project.

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**.

Mini Project Report Submission

In order to get assessed you will ned to submit your report by **11.00AM (UK time)** on **13/04/2021**. This will involve the following steps:

(a) Go to ECS417U QMPlus site.

(b) Click the '*Mini Project Report - 10%*' 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) Click 'Save changes'

The mini project must be submitted in a .doc, .docx or pdf formats. Documents submitted in other formats will not be accepted. Corrupt or otherwise unreadable files will not be accepted.

Late submissions will receive late penalties in line with the late penalty policy, see EECS handbook and QMUL assessment handbook.

Mini Project Phase 1 - Marking Criteria

Assessment Weighting: 20%

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
2.5 marks	One form created
5 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
3 marks	Reset eleven to fifteen elements
4 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

Mini Project Phase 2 - Marking Criteria

Assessment Weighting: 50%

Assessment Criteria 1

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 marks	Partially functional
4 marks	Fully functional

Assessment Criteria 2

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
3 marks	preventDefault function working, but missing fields were not highlighted using CSS
4 marks	Fully functional

Assessment Criteria 3

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
2 marks	Partially functional
4 marks	Fully functional

Assessment Criteria 4

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

Marks breakdown

0 mark	Not working
1.5 marks	Partially functional
3 marks	Fully functional

Assessment Criteria 5

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

Marks breakdown

0 mark	Not working
1.5 marks	Partially functional
3 marks	Fully functional

Assessment Criteria 6

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
2 marks	Partially functional
4 marks	Fully functional

Assessment Criteria 7

Each time a new entry is added, the most recent post appears on top, followed by the next most recent post using a PHP driven technique (writing a sorting algorithm) as opposed to a SQL Query

Marks breakdown

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

Assessment Criteria 8

Implement extra features specified in the coursework specification

Marks breakdown

0 mark	Not working
8 marks	Implemented one fully functional feature
16 marks	Implemented two fully functional features

Assessment Criteria 9

Is the website content relevant for a portfolio website?

Marks breakdown

0 mark	Not relevant
3 marks	Partially relevant
6 marks	Relevant

Mini Project Report - Marking Criteria

Assessment Weighting: 10%

Assessment Criteria 1 - Short Report
Identification of semantic HTML5 elements

Marks breakdown

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

Assessment Criteria 2 - Short Report
Importance of semantic markup

Marks breakdown

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

Assessment Criteria 3 - Short Report
Web technologies/framework

Marks breakdown

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

Assessment Criteria 4 - Short Report
Quality of report

Marks breakdown

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