This assignment does not count toward the final mark.

Assignment 3: CMS Web Application Evaluation

Start Assignment

Due No Due Date **Points** 100 **Submitting** a file upload **Attempts** 0 **Allowed Attempts** 2

Overview

This assessment is to develop a CMS-based web application. It includes the creation of a RESTful API that can process HTTP requests and returns JSON data for specific endpoints.

Learning Outcomes

- LO1: Design and create static and dynamic web content using contemporary web technologies and standards.
- LO2: Install and configure software used to support web environments
- LO3: Develop a web application demonstrating best practice and using current standards in

  client-side and server-side scripting and web interface design
- LO4: Select the foremost technologies for the application development using current scripting, database and information security standards.
- LO5: Utilise a content management system to allow a user to publish, edit and manage web content from a central interface.
- LO6: Analyse web application performance on a variety of platforms to test behaviour of scripts and any functional issues resolved

Conditions

- It is recommended that you spend between 30-35 hours on this assessment.
- Please refer to the Study Plan for the due date
- You can access all course materials and any other resources you wish to use as you work on this assessment.
- The work you submit must be your own work. It is an individual project.

- You can ask a tutor to clarify the instructions and/or for an advice, but they cannot assist you in completing the tasks required – you must carry out the tasks yourself!

Success Criteria

It is required to obtain a minimum of 50% of the total available marks to be successful in this assessment. Your submission will be marked according to the rubrics in the Marking Form document. Please take time to read it.

A maximum of two attempts are allowed to complete this assessment. The maximum percentage to be awarded on a second assessment attempt is 50%.

This assessment contributes to 40% of the final course grade.
Note, the student must pass each assessment task in order to pass the course.

Assignment instructions

Code Guidelines

- Write code in a simple, clear, and readable style.
- Use meaningful and self-descriptive variable names, i.e. do not use names like x, y, a, b. Take care of the alignment and format of the code instructions.
- Add code-comments where it is necessary.
- A sample code is given in the course material.

You are required to submit a supporting document, which may take the form of a blog or a ‘how to’. This document should detail the decision-making process undertaken to fulfil the requirements. Moreover, it should include the problems and their solutions that have been encountered with meeting the requirements. Highlight important aspects of the project by using screenshots and annotations. You may wish to submit a link to a short video of your work running.

ASSESSMENT REQUIREMENTS

The overall objective of this assessment is to develop a CMS-based web application by creating RESTful API that can process HTTP requests and return JSON data for endpoints. In order to achieve this objective, the project needs to fulfil the following requirements:

- A secure admin section that allows to create, edit and/or delete content.
- A secure user login/registration option. It allows to create, edit, and/or update user

  details.
- A publishing section that allows users to publish, edit, and delete contents.

- A client-side JavaScript application that will process both REST API JSON data and form input. It will sanitize and validate any data sent to the REST API.
- All forms need to be secured with CSRF mitigation measures.
- Both client-side and server-side sanitation and validation to prevent not only the

  malicious data manipulation of the API and the application as a whole.
- Application version control with an appropriate configuration.
- A test suite (including unit tests, acceptance tests) for the JavaScript application.
- Best coding practices.

SUBMISSION

Submit assessment documentation as a zip file. Include all necessary assignment files, readme files and any other required documentation. Name your zipped folder as IT7744_A2_YourStudentNumber.

Include your database export SQL file in the project root and call it database.sql. Ensure the administrator user credentials are as follows:

Username: admin

Password: admin
Remember to check that the folder does not contain any other material that you do not wish to

submit.

Submission Checklist

Before you submit your work in iQualify, make sure you have completed all the tasks in this checklist:

Task
A secure CMS admin section allowing creation, editing and deleting of API content.
A secure login, registration and user area to create, edit and update user details.
A publishing section to allow users to publish, edit and delete data.
A simple client-side JavaScript application that will process both REST API JSON data and form input. It will sanitize and validate any data sent to the REST API.
All forms secured with CSRF mitigation measures.

Use both client-side and server-side sanitation and validation to prevent malicious data manipulation of the API and the application as a whole.
Use of best practice coding conventions for php, JavaScript, HTML and CSS/SASS.
Version control for the application with appropriate configuration to optimize versioning.
The JavaScript application will have a test suite with some unit or acceptance tests included.
A secure CMS admin section allowing creation, editing and deleting of API content.
A secure login, registration and user area to create, edit and update user details.
A publishing section to allow users to publish, edit and delete data.

Submission Instructions

Once you have completed all the tasks:

- Go through the submission checklist and to ensure you have completed everything to the expected standard.
- Upload the required assessment documentation and files.
- Read the declaration and submit the assessment.

IT7744 Assignment 3 CMS Web Application Evaluation

Criteria	Ratings				Pts
1. Create a secure CMS application that allows administrators only access to create, editing and delete site content. [10 marks]	**10 to >8.0 pts** **October CMS application has backend enabling CRUD operations on users and images**	**8 to >5.0 pts** **October CMS application has backend enabling adding users and images**	**5 to >2.0 pts** **October CMS application has backend enabling adding users or images**	**2 to >0 pts** **October CMS application does not allow adding users or images**	10 pts
2. Secure login and registration forms that allow new user registration and login. [10 marks]	**10 to >8.0 pts** **Users can either register or login without issue Both forms consistently offer effective feedback on success or failure**	**8 to >5.0 pts** **Users can either register or login Both forms offer effective feedback on success or failure**	**5 to >2.0 pts** **Users can login Forms offer some feedback on success or failure**	**2 to >0 pts** **Users are unable to either register or login Neither forms offer feedback on success or failure**	10 pts
3. A publishing section to allow users to publish and delete data through asynchronous server-side requests. [15 marks]	**15 to >12.0 pts** **Images are consistently able to be uploaded and assigned to the current user which are displayed in the user's images on completion The form allows for the definition of a description, title, filters and file upload, and consistently offers effective feedback on success or failure The form uses Vue JS and Axios to send an XHR request to an API with an authorisation header generated for this user**	**12 to >8.0 pts** **Users can either register or login Both forms offer effective feedback on success or failure**	**8 to >4.0 pts** **Images are able to be uploaded and assigned to the current user The form allows for the definition of a description, title, filters and file upload, and offers some feedback on success or failure The form uses an XHR request to an API**	**4 to >0 pts** **Images are not able to be uploaded and assigned to the current user The form fails to allow for the definition of a description, title, filters and file upload, and offers little or no feedback on success or failure The form does not use an XHR request to an API**	15 pts
4. A javascript framework that processes both REST API JSON data and form input. [15 marks]	**15 to >12.0 pts** **The Vue JS Theme is able to do all of the following: 1. Consistently process the following requests: • fetch user images • fetch images from other users • fetch**	**12 to >8.0 pts** **The Vue JS Theme is able to do most or all of the following: 1. Process the following requests: • fetch user images • fetch images from**	**8 to >4.0 pts** **The Vue JS Theme is able to do some of the following: 1. Process the following requests: • fetch user images • fetch images**	**4 to >0 pts** **The Vue JS Theme is unable to do all or most of the following: 1. Process the following requests: • fetch user**	15 pts

Criteria	Ratings					Pts
	the latest images 2. Consistently correctly stores state on whether a user is logged in or not 3. Store a JWT token in local storage upon login for 7 days and expire the token upon logout	other users • fetch the latest images 2. Correctly stores state on whether a user is logged in or not 3. Store a JWT token in local storage upon login and expire the token upon logout	from other users • fetch the latest images 2. Correctly stores state on whether a user is logged in or not. 3. Store a JWT token in local storage upon login and expire the token upon logout.	images • fetch images from other users • fetch the latest images 2. Correctly stores state on whether a user is logged in or not 3. Store a JWT token in local storage upon login and expire the token upon logout		
5. Has client-side and server-side sanitation and validation to prevent malicious data submission. [10 marks]	**10 to >8.0 pts** **All forms consistently sanitize and validate data to prevent duplicate or erroneous submissions**	**8 to >5.0 pts** **All forms sanitize and validate data to prevent duplicate or erroneous submissions**	**5 to >2.0 pts** **Forms santize or validate some data to prevent duplicate or erroneous submissions**	**2 to >0 pts** **No forms sanitize and validate data to prevent duplicate or erroneous submissions**		10 pts
6. Use of best practice coding conventions for php, javascript, HTML and CSS/SASS. [25 marks]	**25 to >20.0 pts** **SASS nested correctly, but not overly so, to reduce code Styles are top-down and general-specific Few or no HTML validity errors HTML indented well and uses sematic tags where suitable JS is well-commented, efficient, tidy and error and warning free. Files are organised consistently and logically. PHP is well-commented, efficient, tidy**	**20 to >15.0 pts** **SASS nested correctly to somewhat reduce code Styles are mostly top-down and general-specific Some HTML validity errors HTML indented well and uses sematic tags JS is commented, tidy and error free. Files are organised logically. PHP is commented, tidy and error free. Files are organised logically.**	**15 to >10.0 pts** **SASS mostly nested correctly but could be more efficient Some styles are mostly top-down and general-specific Moderate HTML validity errors HTML indented and uses sematic tags JS is tidy and mostly error free. Files are organised. PHP is tidy and mostly error free. Files are organised.**	**10 to >5.0 pts** **Some SASS is nested correctly but is inefficient Few styles are top-down and general-specific Many HTML validity errors HTML indented JS could be tidier and has several errors. Files could be better organised. PHP could be tidier and has several errors. Files could be better organised.**	**5 to >0 pts** **SASS is not nested correctly or not evident Styles are not organised in order Excessive HTML validity errors HTML not indented JS is untidy and has many errors. Files lack organisation.**	25 pts

Criteria	Ratings				Pts
7. Version control for the application with appropriate configuration to optimize versioning. [5 marks]	and error and warning free. Files are organised consistently and logically. **5 to >3.0 pts** **A .git repo exists in both the plugin and theme folders, each with a separate remote master. A .gitignore file that ignores vendor and CMS application files**	**3 to >1.0 pts** **A .git repo exists in both the plugin and theme or in the actual application.**	**1 to >0 pts** **A .git repo does not exist**		5 pts
8. A set of javascript unit test scripts that accurately test aspects of the application. [10marks]	**10 to >8.0 pts** **Mocha unit tests offer full coverage of each view**	**8 to >5.0 pts** **Mocha unit tests offer coverage of most views**	**5 to >2.0 pts** **Mocha unit tests offer coverage of at least one view**	**2 to >0 pts** **Mocha tests do not exist or fail to work**	10 pts
					Total Points: 100