

COMP.6210

Web Services & Design Methodologies

Web Application Design Assignment

Assignment 2 – Semester 2 2019

Due Date: Thursday 31st October 2019 at 5.00 p.m.

Weighting: 40%

Marks: 40 marks

Aim:

The aim is to design and build a dynamic database driven website using a number of web development languages

Learning outcomes covered:

3. Create and utilise one or more web services.
4. Write a web application that utilises an external web service

Notes:

This is an individual assignment and must be the product of your own work. Students are reminded to read pages related to assessment rules including rules for dishonest work in the Toi Ohomai Student Guide.

You are to upload the content of your web root folder to the GitHub classroom link.

- Any libraries or frameworks should not be part of the submission, but included in an package.json file or linked from an external resource.
- You should also include a complete SQL file an export from PHPMyAdmin that includes any additional test data that you created. The structure of the database should not be different than that from the first assignment.

PART A - Connecting the database and generating the queries in PHP models. (10 Marks)

For this part you are to create the Models (MVC) required for your website. The database must connect properly to your project and must be able to query the database and return data from it using PHP.

Functions for all of the features required in the application must be created

PART B - Implementing the MVC model, create the controllers (MVC) and the Views (MVC) (18 Marks)

For this part you are to code the controllers that create the endpoints of your web service API. Each endpoint should be clearly named to indicate what its purpose is. (e.g. /pages/showAllMessages or /contact/submitForm)

For the Views you must use the HTML and CSS files from Assignment 1. All content from A1 must be deleted and JavaScript is used to pull in the data from your PHP files. You can use ajax calls or the fetch api to do so.

PART C - Testing, deploying and creating documentation for your project to deliver to your client. (10 Marks)

For this part you are to test your website application. Make sure you are able to set it up on a different machine and write the instructions for another user to set it up.

You must then find another classmate who will test it by setting up your project.

The process must be signed off.

For part C you must also keep the regular meetings with your tutor and keep track of any difficulties you may have.

| Marking Rubric Part A (Model) (Connecting the database and generating the queries in PHP models) | Achieved 2 | Some Issues 1 | Unacceptable 0 | Marks Attained /2 |
|--|---------------|------------------|-------------------|----------------------|
| Database connection code (PHP) working with no errors | | | | |
| Database connection is programmed using PDO library | | | | |
| Create all the GET models required for web service application (Functions are created with excellent detail and all of the functions return JSON data) | | | | |
| Create all the POST models required for web service application (Functions are created with excellent detail and all of the function accept JSON data and give Feedback in JSON) | | | | |
| Create all the UPDATE models required for web service application (Functions are created with excellent detail and all of the function accept JSON data and give Feedback in JSON) | | | | |
| Create all the DELETE models required for web service application (Functions are created with excellent detail and all of the function accept JSON data and give Feedback in JSON) | | | | |
| Part A Total Marks: | | | | /12 |

| Marking Rubric Part B (Controllers / Views) | Achieved 2 | Some Issues 1 | Unacceptable 0 | Marks Attained /2 |
|--|---------------|------------------|-------------------|----------------------|
| Controller: Create ALL of the required endpoints (Endpoints are created and are clearly identified as to their purpose) | | | | |
| Controller: Naming conventions (Code is written according to OOP standards. Classes, Methods and properties in Pascal case and Variables in Camel Case - Files are named the same as the class) | | | | |
| Controller: Security (Endpoints are created but and sanitise the data before the model is called) | | | | |
| View: Template files (Used the files from Part A and got the data using proper ES6 syntax) | | | | |
| View: PHP (No PHP in the Views - everything is processed and accessed using Javascript) | | | | |
| View: Contact Form (Form is displayed and is responsive and it works) | | | | |
| View: Validation on Forms (Validation is done using Javascript, using in ES6 syntax) | | | | |
| View: Data Storage (Form works, stores the data and sanitises it) | | | | |
| View: Update Forms (Update functions work, and the current data was presented to the user) | | | | |
| Part A Total Marks: | | | | /18 |

| Marking Rubric Part C (Testing, Deployment and documentation) | Achieved 2 | Some Issues 1 | Unacceptable 0 | Marks Attained /2 |
|--|---------------|------------------|-------------------|----------------------|
| Creating setup documentation (Document with screenshots with clear instructions on how to setup and deploy your MVC based web application) | | | | |
| Testing document (Student setup) (Document that shows student has successfully deployed and tested their application) | | | | |
| Testing document (User setup) (Document that shows another has been able to successfully deployed and test the web application) | | | | |
| Sign off (Successful deployment and testing has been signed off by colleague) | | | | |
| Submission (Project has been submitted to Github link with libraries packaged in json file, export SQL file included in repository) | | | | |
| Part A Total Marks: | | | | /10 |