# PGCert IT: Programming with Web Technologies

Web Assignment - An Article Homepage

## Due: Sunday 15th April 2018 at 11:59pm

This assignment requires you to develop a website using HTML5, CSS, and JavaScript / jQuery. There are **Six** tasks in this assignment.

Before starting the assignment, read through and gain an understanding of the requirements.

**Notes:**

- The assignment is out of **60 marks** and is worth **10%** of your final Programming with Web Technologies grade.
- You are not provided with any starting resources for this assignment - you must create everything yourself.
- After completing the assignment, you should have a functional web app with some number of HTML, CSS, and JS files, in an appropriately configured git repository.
- To submit this assignment, submit the URL of your Bitbucket repository to Canvas / Moodle on or before the due date.
- **IMPORTANT**: Read the instructions carefully before attempting each task.

## Introduction

In this assignment, you'll gain experience developing part of the client-side of a simple article viewer web app. You'll put to use a variety of the HTML5, CSS, and JavaScript / jQuery techniques you've learned so far. You'll also make use of git / Bitbucket to keep a history of changes to your project, and you'll create branches at various points so the markers can easily checkout those branches to see your progress at those points.

Note that some of the instructions in this handout are *intentionally* vague. The purpose is to allow you to explore your creative side without getting bogged down by too many specific requirements.

## Task One - Bitbucket & New IntelliJ Project

To begin, create yourself a Bitbucket repository that will hold all your work for this assignment. **Make sure that you don't make the repository private** - the markers will need to access this!

At this stage, also create a new "static web" project using IntelliJ, naming it something sensible, and configure the project to use git as its version control system. Copy the `.gitignore` file from one of your web labs into your new project, then commit and push the starting point to your Bitbucket repository with an appropriate initial commit message.

### Use of git for this assignment

For each of the following tasks below, create a new branch with the task name (e.g. TaskTwo, TaskThree, etc). When that task is complete, then merge the branch back onto master. **Do not delete** any of the branches, and make sure they are all pushed to Bitbucket. This is so the markers can checkout each task's branch to see how your assignment looked after the completion of each task.

Upon the completion of this assignment, your Bitbucket repository should contain a master branch with your completed assignment, along with branches named TaskTwo through TaskX, each one representing the state of your project upon the completion of the corresponding task.

Along with the required commits and pushes, make sure to make regular, granular commits with appropriate comments, to document your progress through the assignment.

## Task Two - A Basic Article Viewer Homepage

In this task, we'll begin building the website itself.

In your new TaskTwo branch, create a page which allows users to view a list of news articles. Each article has a *title*, a *thumbnail image*, and some *content*. In addition the articles themselves, your page should contain a title bar displaying the name of your website, and a sidebar with a list of quick links - one for each article. Clicking on one of the quick links will cause the page to scroll so the corresponding article is visible in the viewport.

For this task, exactly how your content is laid out on the page is up to you. You may (and should) use CSS appropriately to help you with this, but for task two you **may not** use any CSS frameworks. You may use the native CSS grid if you like.

For the article content, you may find your own thumbnail images and text content, or use websites such as lorempixel.com and lipsum.com. To get an idea of how your website will look with multiple articles, create about five articles for testing purposes.

One possible example of how your solution might look is shown in Figure One below, though the details are up to you. Marks will be awarded for a visually appealing design with an intuitive layout, though you are not expected to be winning any design awards!

When you've completed this task, make sure your final TaskTwo branch is pushed to Bitbucket, then merge it onto master.

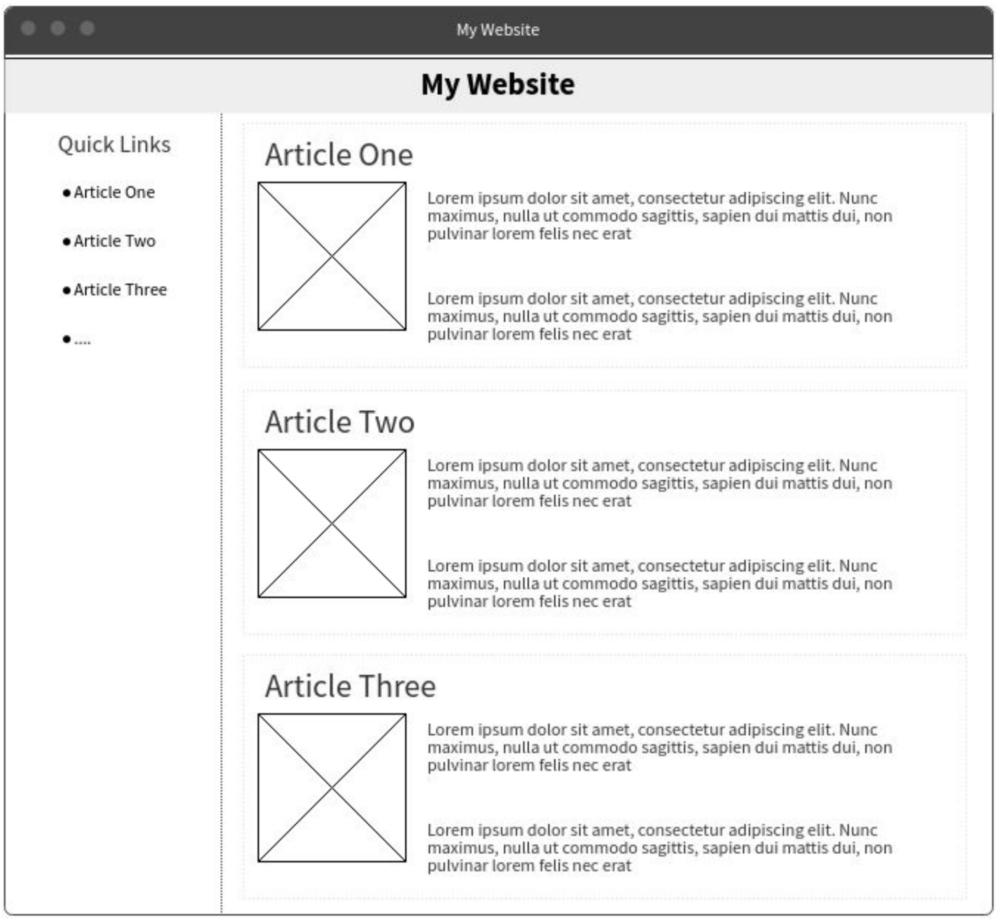

My Website
My Website
Quick Links
Article One
Article Two
Article Three
.....
Article One
Article Two
Article Three
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

*Figure One: An example wireframe diagram of how your completed solution to task one might look.*

## Task Three - Adding a Simple HTML Form

Many blogging websites have an easily accessible form allowing users to sign up to receive emails / weekly newsletters / etc. In this task, you'll add such a form to your website.

To begin, remember to create and checkout a new branch, called TaskThree. Then, add an HTML form to your page. Add the form on the page's sidebar, just below the quick links. The form should consist of an appropriate title and description, along with an `email` input named "`email`", and a `submit` button. Users should not be able to submit the form without entering an address. An example of how the form might look is given in Figure Two below - though again, the details are up to you. At this stage, do not use any CSS frameworks.

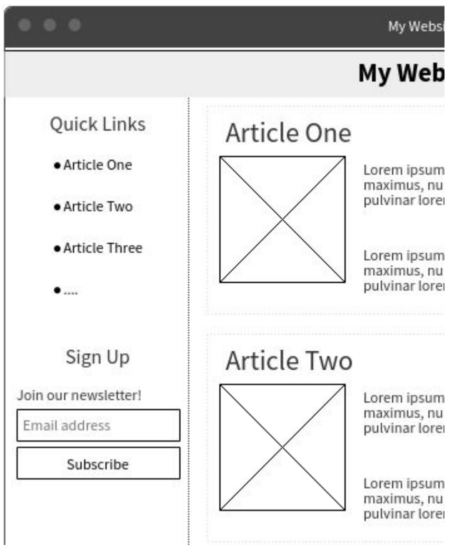
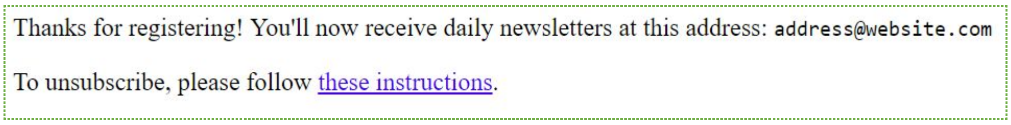


*Figure Two: Example newsletter signup form.*

When the user clicks the submit button, assuming they have entered a valid email address, the form should send a POST request to the following website:

[https://sporadic.nz/2018a_web_assignment_service/Subscribe](https://sporadic.nz/2018a_web_assignment_service/Subscribe)

If the form submission is successful, then a message such as the following will be displayed:



Thanks for registering! You'll now receive daily newsletters at this address: `address@website.com`

To unsubscribe, please follow these instructions.

If the form submission is not successful, the website will display an error message with details on what the error could be.

When you've completed this task, make sure your final TaskThree branch is pushed to Bitbucket, then merge it onto master.

## Task Four - Responsive Design

For this task, you'll make sure that your website is responsive - that is, it looks and behaves appropriately on a variety of screens such as desktop, tablet and mobile displays.

To begin, create and checkout a new branch, called TaskFour. Then, modify your website as follows.

When viewing your site on a desktop browser, it should look and behave identically to your Task Three solution. However, when the window is resized or viewed on a mobile device, the page should rearrange itself to be usable on the new screen configuration.

One possible solution to this is shown in Figure Three below, in which the content originally in the sidebar has moved to the top of the screen, in-between the header and the articles.

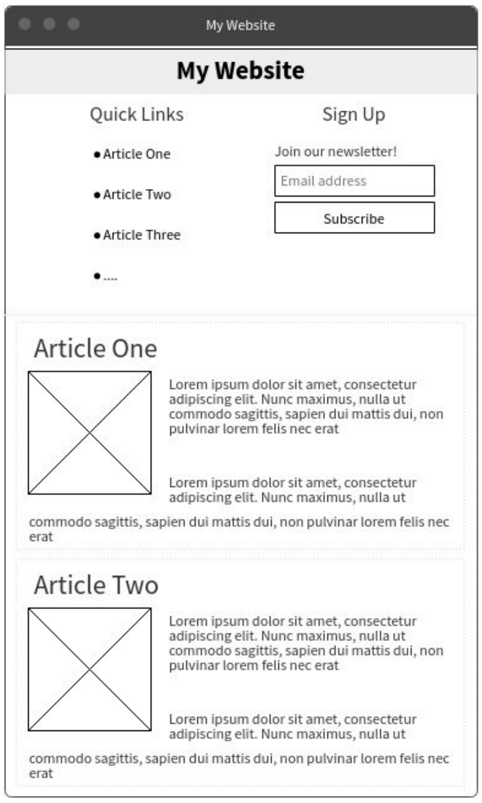

My Website
My Website
Quick Links
Article One
Article Two
Article Three
....
Sign Up
Join our newsletter!
Email address
Subscribe
Article One
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat
Article Two
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

*Figure Three: Possible mobile layout for the website.*

As with previous exercises, the look and feel of the website is up to you, except that you may not use any CSS frameworks.

When you've completed this task, make sure your final TaskFour branch is pushed to Bitbucket, then merge it onto master.

## Task Five - Bootstrap

In this task, you'll adopt the Bootstrap 4 framework, which will allow you to more easily achieve rich, complex layout and functionality for your site.

To begin, create and checkout a new branch, called TaskFive. Then, convert your website to use Bootstrap. Here are some points to consider:

- Remember to import the required CSS and JavaScript links.
- Use the Bootstrap column classes to lay out your website, rather than CSS grid or other methods.
- Consider what bootstrap controls and components would be useful for your design (for example, you could use Bootstrap cards for each of your articles).

As a first step, modify your website to have identical functionality as it does currently, but using Bootstrap instead of your own CSS. Then, continue to develop the page to add additional functionality.

Upon completion of Task Five, your page should meet the following requirements:

- The page should contain a fixed nav bar at the top of the page. The navbar should contain your page's title / logo, quick links, and signup form. The title and quick links should appear on the left, while the signup form should appear on the right, similar to the search box in the linked example.
- When viewing the site on a mobile, instead the navbar should appear *collapsed* instead, with only the title displayed on the left, and a button to expand the navbar appearing on the right. When the user clicks the button, the quick links and signup form should drop down from the navbar. Play around with the linked example above at different screen sizes for a demonstration of this behaviour.
- Articles should be displayed using an appropriate Bootstrap component such as a Card. Articles should be displayed in two columns when viewing the site on larger screens, and in a single column when viewing the site on smaller windows.
- Aside from the above features, your site should show off *at least two other* Bootstrap 4 components of your choice.
- You should include some of your own CSS, or locate and use a Bootstrap theme, so that your website doesn't simply look like a default Bootstrap application. Give your site a distinctive look.

When you've completed this task, make sure your final TaskFive branch is pushed to Bitbucket, then merge it onto master.

## Task Six - AJAX

In this task, you'll finalize your assignment solution by using article data from a "real" web service, rather than your own static data.

To begin, create and checkout a new branch called TaskSix. Then, remove all of your articles from your page (or, comment them out in case you need to re-use anything later!), and replace them with appropriate component showing a "Loading Content…" message. This message is to be displayed until your page's AJAX call is complete.

Next, create a new file in your project called load_articles.js, and link to it appropriately from your main html page. Inside your new JS file, **using jQuery**, create an event handler that will run when the document has finished loading.

In that event handler, again using jQuery, create an AJAX call which will issue a GET request to the following URL:

https://sporadic.nz/2018a_web_assignment_service/Articles

This link will return JSON data containing an array of articles to display. Each article in the array has an `id`, a `title`, some `content`, and an `imageUrl` pointing to the location of the article's thumbnail image. You may open the link in a web browser to see exactly what data is being returned.

Further modify the AJAX call so that it, when successful, will show the article on the page, using appropriate HTML elements (as similar as possible to your Task Five solution).

When you've completed this task, make sure your final TaskFive branch is pushed to Bitbucket, then merge it onto master and submit your Bitbucket URL to Canvas / Moodle.

## Marking Guide

This assignment will be marked as follows:

| | |
|---|---|
| **Task One** | |
| Bitbucket repository created | *2 marks* |
| Branches for all completed tasks are present, with correct content | *3 marks* |
| Version control is appropriately used throughout the assignment (fine-grained commits, useful comments, etc) | *3 marks* |
| **Task Two** | |
| Required information is shown on page | *7 marks* |
| Site look & feel | *3 marks* |
| **Task Three** | |
| Form is functional | *6 marks* |
| Form’s look & feel is consistent with the rest of the page | *2 marks* |
| **Task Four** | |
| Site behaves as described (i.e. is responsive) | *10 marks* |

| | |
|---|---|
| **Task Five** | |
| Navbar as described on desktop displays | *3 marks* |
| Navbar as described on mobile displays | *3 marks* |
| Articles display appropriately | *2 marks* |
| Extra Bootstrap components used | *4 marks* |
| Non-default look & feel | *2 marks* |
| **Task Six** | |
| AJAX call successfully made using jQuery | *5 marks* |
| Resulting articles displayed correctly | *5 marks* |