

Assignment 3 Rationale

By Dylan Forrest U3167792



May 10, 2019

Tutor: Riley Post

Tutorial time: Wednesday 12:30 – 2:30

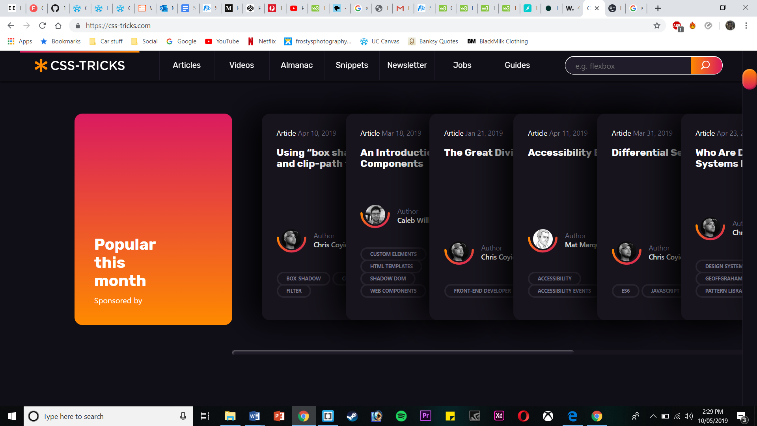
Website is in the github repository:

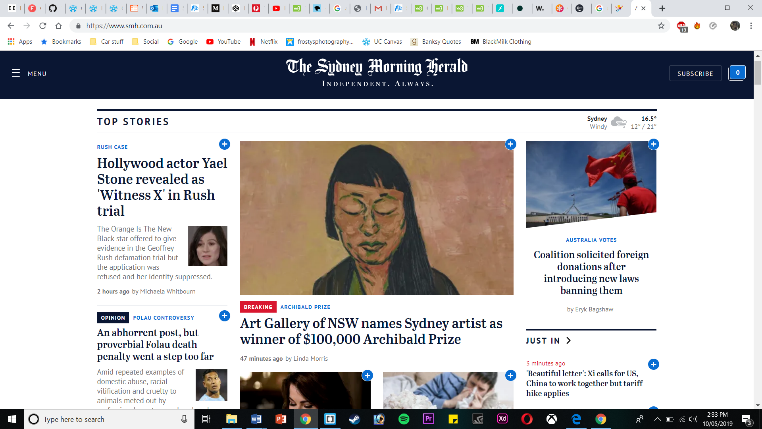
<https://github.com/Frostysnowman115/U3167792-FWD-Assignment3.git>

My website is about Norfolk Island.

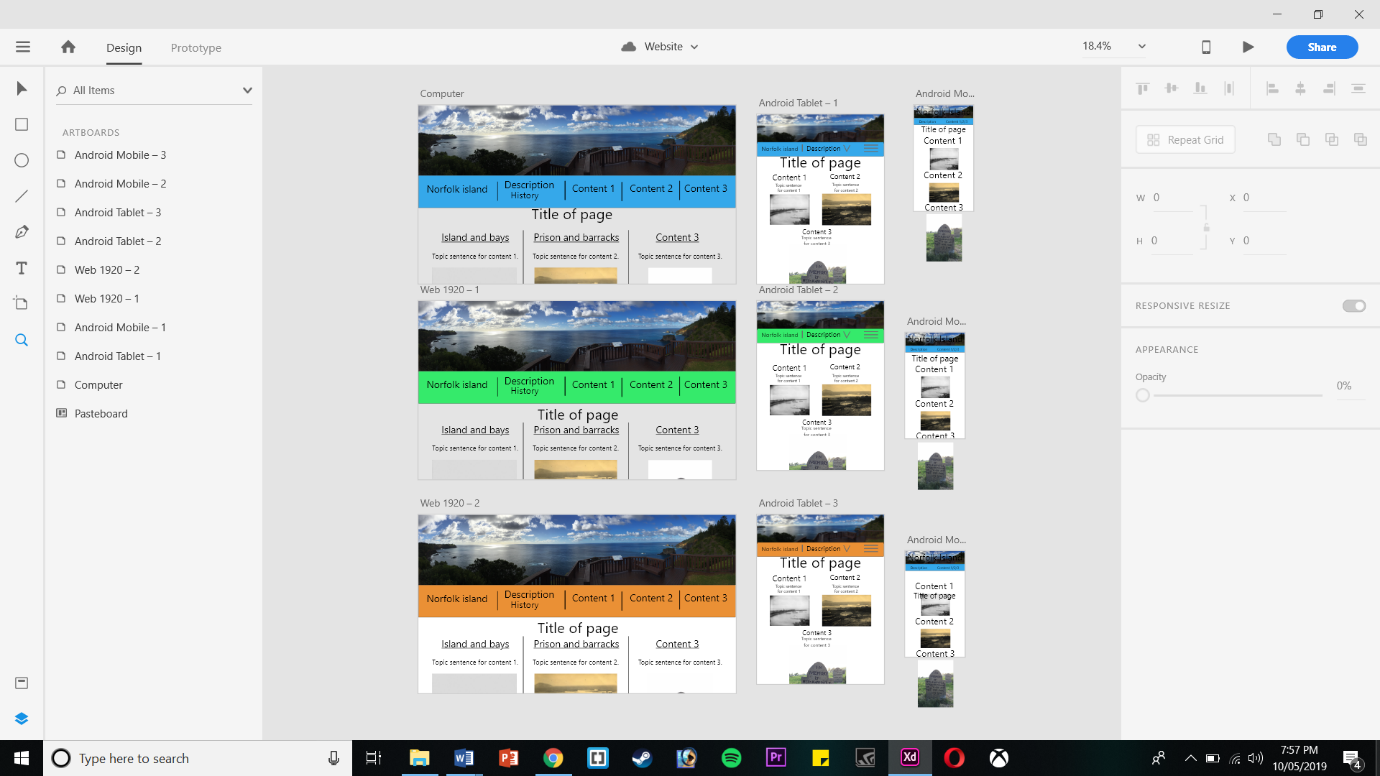
Sites that have inspired my design are land stories and CSS tricks.

Land stories uses a parallax image, I really like the aesthetic that the parallax scroll has.

CSS tricks uses a sticky navbar, I thought I could make it work nicely with a parallax scroll and I feel like I’ve achieved that.

I also liked the idea of using two columns for words, this idea came from The Sydney Morning Herald. The site has at least two categories of content across, I believe that’s too much for the viewer so I decided to do two simple columns instead.

To start making my website I used Adobe colour to help me choose the colour that I wanted to use. I began with blue as something basic. I wanted to use different colours for different pages of my site so I chose green and orange with similar brightness and colour as the blue I chose.

This is what I came up with when trying to draft the appearance of my website. I had designed three columns of pictures with basic titles and topic sentences for the viewer, this was followed by images which the viewer would be able to see and immediately tell where they wanted to go within the website. Then the site would have had the two columns of words underneath for each section. The design was created with large, medium and small sizes for viewing so the appearance was similar across all devices.

The first of many problems that I had included the <nav> element being the child to a <header> element. This caused a problem for my sticky Navbar. The Navbar would only be sticky within the header, and then as soon as the user scrolled past the header, the Navbar couldn’t be seen as it was still stuck inside the header. I fixed this problem simply by removing the Navbar code from the <header> and placing it underneath.

<header id="top">

<nav id="main-menu" class="main-menu">

<div class="wrapper">

<h3>Norfolk Island</h3>

<ul>

<li><a href="page2.html">Description</a></li>

<li><a href="page3.html" class="link">History</a></li>

<li><a href="#content1">General overview</a></li>

<li><a href="#content2">Gaol</a></li>

<li><a href="#content3">Barracks</a></li>

</ul>

<a href="#main-menu-toggle" class="backdrop" hidden></a>

</div>

</nav>

</header>

A problem that took a substantial amount of my time trying to fix was getting the images to be positioned in the correct orientation and the correct size. As soon as I had changed the orientation of the images things got very confusing for me, even when I felt like I was doing them right nothing changed. In changing the orientation of the images, this made the heights change to the widths and vice versa. Below the HTML of the images is shown.

<div class="container">

<div>

<h2>General overview</h2>

<img src="assets/img/norfolk-island-16.JPG" alt="Looking towards hospital from Gaol grounds.">

</div>

<div>

<h2>Gaol</h2>

<img src="assets/img/norfolk-island-23.JPG" alt="Gaol entrance" class="rotateimage">

</div>

<div>

<h2>Barracks</h2>

<img src="assets/img/norfolk-island-10.jpg" alt="Looking towards soldier barracks and Commissariat Store">

</div>

</div>

The next section of code shows the CSS of the images, and the grid that I used to separate them into three sections.

.container {

display: grid;

grid-template-columns: 1fr 1fr 1fr;

width: 90%;

}

.container img {

width: 100%;

height: auto;

}

I decided not to use the images or the grid in this way as I couldn’t get the images to work as I wanted and I felt that the grid wasn’t much use without them. This involved me removing these elements from my code and so my current website does not contain images in this way.

I wanted to use a hamburger menu as my responsive menu, however this didn’t work as I had planned either because I found that it was too much of a complicated system to learn in the short period of time that I left for myself. I resolved the problem of my responsive Navbar by using an inline block for the un-ordered list with a big display and a block list for a small display.

nav ul {

width: 100%;

display: inline-block;

text-align: center;

}

Above shows the list as an inline-block. The next section of code shows the un-ordered list as separate blocks.

@media screen and (max-device-width: 650px) {

li{

display: block;

width: 100%;

}

The following table shows an accessibility audit. An accessibility audit is important because it reviews the problems that some people with disadvantages might have. Some examples of these might be; limited sight or hearing, limited network speed, different browser use, or being a keyboard only user. It is important that an accessibility audit is done on the website as it allows people with all disadvantages to access the site and still have the opportunity to get the information that they want or require.

|  |  |  |
| --- | --- | --- |
| Description of accessability: | Description of available access: | Pass or Fail: |
| 1.1.1 - Non-text Content - Provide text alternatives for non-text content. | Non-text content not provided for all images. | Failed. |
| 1.3.1 – Info and Relationships – Logical Structure. | Site has a logical structure. | Passed. |
| 1.3.2 – Meaningful sequence – present in a meaningful order. | Presented in a meaningful order. | Passed. |
| 1.4.1 – Use of colour – Don’t use presentation that relies solely on colour. | Presentation of site doesn’t rely on colour. | Passed. |
| 1.4.2 - Audio control – Don’t play audio automatically. | No audio present. | Passed. |
| 2.1.1 -Keyboard – Accessible by keyboard only. | Navigation using a keyboard is easy, all options are available for keyboard users. | Passed. |
| 2.1.2 – No keyboard trap – Don’t trap keyboard users. | Keyboard users can’t get trapped. | Passed. |
| 2.3.1 – Three flashes or below – no content flashes more than three times per second. | No flashes present. | Passed. |
| 2.4.2 – Page titled – use helpful and clear title pages. | Pages are titled clearly and helpfully. | Passed. |
| 2.4.3 – Focus order – logical order. | Pages are displayed in a logical way. | Passed. |
| 2.4.4 – Link purpose – every link’s purpose is clear from it’s context. | All links have a description of where they lead to. | Passed. |
| 3.1.1 – Language of page – Page has a language assigned. | Page has a language assigned. | Passed. |
| 4.1.1 – Parsing – No major code errors. | No major errors in code. | Passed. |

**Links of References:**

Sites that were used for inspiration:

<https://www.landstories.co.uk/> - Land stories

<https://css-tricks.com/> - CSS Tricks

<https://www.smh.com.au/> - Sydney Morning Herald