**Web Design & Client-side Scripting**

Lecturer: Sam Cogan

**- Group Project -**

**Group Members:**

Colum Kavanagh (Student No. 21121851)

Daniel Morrissey (Student No. 21118701)

Ciara O’Malley (Student No. 20215827)

Business:

**A Music Label called ‘Bass-ic Records’**

Visit the website here:

<https://columkavanagh.github.io/MusicLabelWebsite/>

Visit the related GitHub repo here:

<https://github.com/ColumKavanagh/MusicLabelWebsite>

**Executive Summary**

We built a website for a fictional Irish music label called ‘Bass-ic Records’.

Initially, Daniel and Colum decided they would do a website for a fictional music label. They were both interested in music and the idea would allow for a couple of interesting tools to be built and used on the website (the ‘Recommendations’ and ‘Gigs’ tools our case.) When Ciara joined the team thereafter during Week One, she was happy to move forward with the ‘music label’ idea as she has experience working in the music industry.

We wanted to create an informative website that could promote the artists that were signed to our label, ‘Bass-ic Records’, but we also wanted to do this in an interactive way so that the website was engaging for people to use.

We feature an unordered list, forms with different types of inputs (typed input and dropdown menus), and also include some interactive items just for fun on our website, such as showing and hiding a slideshow and the search results on our ‘Recommendations’ and ‘Gigs’ tools (the slideshow uses JavaScript to do this while the tools use jQuery.).

We tried to push things a little further by integrating ‘iframes’ that link to relevant YouTube videos within our ‘Gigs’ tool results section. This is contrasted with our use of a HTML5 video tag on the ‘Gigs’ page use to loop a short video sting in order to make that page more visually interesting. We also integrated a pop-up (raw JavaScript) that appears overlaid on top of our homepage and prompts users to sign up to a hypothetical email subscriber list to keep up to date with news from Bass-ic Records.

**How our website was designed and how our team split the project work**

We initially discussed our overall approach to the website and used wireframes to mock-up some ideas. See the images below for our wireframe designs.

Then, after an initial skeleton of the four page website was built in html and the relevant repo set up on Colum’s GitHub (with other team members assigned as ‘contributors’), each member of the team took responsibility for a different section of the site. Ciara built the contact page; Daniel built the recommendations and gigs tools; and Colum built the homepage slideshow and helped Ciara with a pop-up to gather email addresses on our homepage. This allowed each member of the group to gain experience using HTML, CSS and JavaScript (along with Bootstrap and jQuery evidently) to contribute to the overall website.

Once the initial versions of our individually assigned sections were complete, the process became much more collaborative, whereby each member of the team helped edit and hone the initial HTML, JavaScript and CSS code. Regular team meetings, commenting on the code throughout and messaging via MS Teams enabled this process.

**Wireframes**

See below for images of wireframe designs for our ‘Home’, ‘Recommendations’, ‘Gigs’ and ‘Contact’ pages, respectively. These wireframes were built using a collaborative, online UI design tool called ‘Figma’. (<https://www.figma.com/>)

Text

Description automatically generated Graphical user interface, table

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence Graphical user interface, text, application

Description automatically generated

**Outline of schedule and deliverables**

Week 1: Skeleton of website and repo on GitHub. Design using wireframes.

Week 2: ‘Recommendations’ tool (built with jQuery). ‘Gigs’ tool (built with jQuery). Contact form. Slideshow on homepage (Bootstrap template edited with CSS and raw JavaScript).

Week 3: Pop-up on homepage. Final tweaks to all pages’ JavaScript/jQuery. Finalising CSS across the website.

Week 4**:** Deployment and speed/SEO testing and optimisation.

Week 5**:** Refining final website layout**.** Report writing.

**Responsiveness**

We used Bootstrap and media queries in our CSS file to ensure that our website was laid out correctly and looked its best no matter what device on which it is viewed. For example, the links and logo in our navbar and footer stack on top of each other when viewed on a narrow screen such as a smartphone. The ‘Developer Tools’ built into Google Chrome helped us test for this throughout the build process.

**Frameworks**

As mentioned earlier, we used Bootstrap and jQuery on our website. The use of these was also customised on different parts of the website. For example, while the slideshow on the homepage was based on ‘Carousel’ template from Bootstrap, we used raw JavaScript to edit the amount of time in milliseconds that each slide is shown for (this piece of script can be seen in our script.js file).

We made a point to mix jQuery and raw JavaScript across the website and this can also be seen in our script.js file e.g. the slideshow on the homepage is shown and hidden using raw JavaScript and an ‘onclick’ tag in the relevant html while the results on our ‘Recommendations’ tool are shown and hidden using some very visually appealing animations from jQuery.

**Form Validation**

Also, in terms of mixing approaches in order to demonstrate our learning, for the form validation across the website we used HTML5 tags to help with this alongside writing our own ‘validateForm();’ functions in JavaScript. The form on our ‘Contact’ page, for example, uses HTML5 specific tags like ‘input type="email"’ to prompt users to input a correctly formatted email address and we also use JavaScript to highlight the incorrect HTML field if a user tries to submit the contact form with unsuitable data in it; we do this using the focus() method.

**Deployment**Our website is deployed using GitHub’s Pages feature. This allows the website to be viewable by the public. As this is a college project, the URL for the public website is:

<https://columkavanagh.github.io/MusicLabelWebsite/> as opposed to say ‘www.bass-icrecords.ie’. Deploying the website on GitHub Pages also allowed us to remove the ‘index.html’ suffix from the ‘hrefs ‘ which we used to link from page to page on our website. This is because the servers that GitHub pages uses are likely Linux based rather than Windows based (which would require the ‘index.html’ suffix to follow each ‘href’ address).

**Testing**

**HTML & CSS Validation:**

- We used the HTML (<https://validator.w3.org/>) and CSS validators (<https://jigsaw.w3.org/css-validator/>) from W3C to proof our HTML and CSS files. Any errors shown by this validator were corrected so that now, no errors show for any of the HTML or SCC used on our website. Please see the validation documents in the ‘Supplementary’ folder of our website files proving that all pages were put through this process. Each member of the team validated their own HTML files for the pages they were initially responsible for and Ciara validated our CSS file. This ensured that we all gained experience using these tools and correcting errors.

**Testing the speed of our website:**

- We used Google’s Page Speed tool (<https://pagespeed.web.dev/>) to test the speed of our website pages. Initially our scores on Google’s Page Speed tool were in the mid-80s, however after some adjustments (suggested by the Page Speed tool itself), we managed to get our mobile score up to 87 and or desktop score to 99 (see screenshots below).

Graphical user interface, text, application, email, website

Description automatically generatedGraphical user interface, application

Description automatically generated

**Optimisation**

**Speed**

Some of the actions we undertook to optimise the speed of our website included:

Adjusting images:

- We resized any images used on the website so that they are as small as possible while still remaining of good quality. After using Google’s Page Speed tool in testing, we decided to change the file format of the images used on our site to a more modern web format called ‘.webp’ (a format developed by Google for use on the web). This improved the speed of our website without sacrificing image quality to any significant degree.

Single JS and CSS files:

- We organised and collated all our JavaScript into one script.js file that all four page of the website then drew from. As each member of the team contributed some JavaScript and/or jQuery, we used comments to help explain to each other what our JavaScript was doing and then structured the overall script.js file to make it more readable and easier to follow. We also put all of our CSS into the one style.css file for the website and further optimised this by minifying this CSS file (using the online CSS minifier at [www.toptal.com/developers/cssminifier/](http://www.toptal.com/developers/cssminifier/)).

Use of CDNs:

- For both of the libraries that we used (Bootstrap to improve the responsiveness and layout of our website) and jQuery (to improve user’s experience of using the interactive aspects of our website) we used Content Delivery Networks (CDN) to import these into our website. This kept the overall size of the files need to run our website smaller than if we had downloaded the jQuery library ourselves and loaded that alongside our other website files.

**Search Engine Optimisation:**

Some of the actions we undertook to increase the Search Engine Optimisation (SEO) of our website included:

- Using different meta descriptions and meta keywords in ‘head’ section of each of our four separate webpages.

- Using relevant music-industry related keywords in the paragraph content of our different pages (i.e. the content that went in between the <p>“”</p> tags on our pages).

- Using h1 and h2 tags on our webpages where relevant to help Google understand the structure of the pages as it ‘crawls’ and indexes them.

- While we will not actually be submitting it to Google, we did create a sitemap for our website in XML format. (This can be found in the ‘ Supplementary’ folder on our GitHub repo for the website.) We created our basic sitemap using an online automatic sitemap generator (<https://www.xml-sitemaps.com>). The XML sitemap can be submitted to Google’s ‘Search Console’ to make it easier for Google to find, crawl and index your website, and this would also improve SEO/search-result-ranking in a real-life scenario, especially for a brand new website.