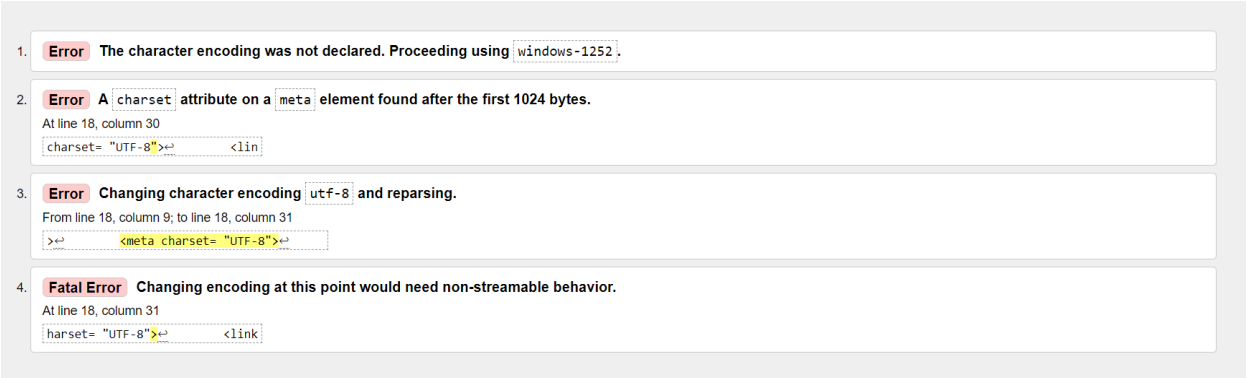
Results of running the W3C Validator.

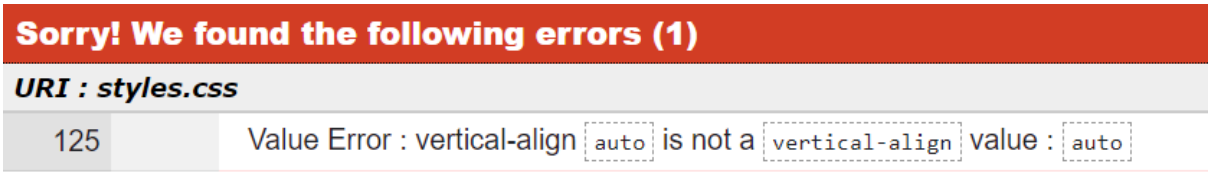
To get started with the review we will represent the results of running the W3C validator.



As you can see from the picture above, the code had 3 errors and 1 fatal error.

The first error is that the HTML file did not have a specified encoding which led to a default one being used. By this, you risk that characters in your content are incorrectly interpreted on the page. All of the following errors are related to the first error.

For CSS:



Conclusion: Always specify the encoding, to avoid bugs/ incorrect functioning.

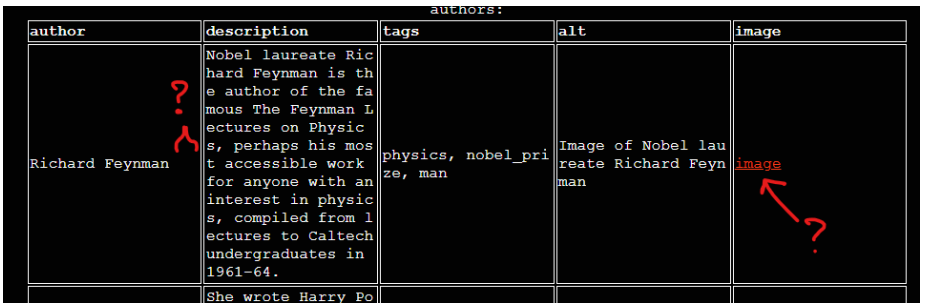
Accessibility.

Guideline 1.1:

- Controls, Input.
- Time-Based Media.
- Test.
- Sensory.
- CAPTCHA.
- Decoration, Formatting, Invisible.

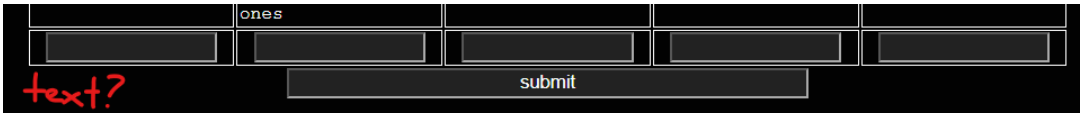
Issue number 1:

The images that were supposed to be PNGs are links to the pictures using the anchor tag. Which is wrong according to the requirements of the task that we were given. Moreover, those links do not have descriptive text, like a <figcaption>, that would comply with the WCAG guidelines.



Solution:
Change the <a> tag to the <img> and add a <figcaption>

Issue number 2:
The form in the last column of the page does not have any descriptive text. This can confuse users and lead to wrong input of data.



Solution:
The **simplest** way to fix this issue would be to add a <p> tag that describes what each input is for, however, the **correct** way to do it is to add a <caption for=" name"> tag.

Issue number 3:
Since the website that we are reviewing does not require any advanced styling/ formatting the overall performance in terms of visibility is good. However, I would have changed the font to something different and made the font size a little bit bigger

Guideline 1.3:

- Info and Relationships
- Meaningful Sequence
- Sensory Characteristics

The first requirement, Info, and Relationships, is met on the web page. The code has sections that are separated by different headings, using the tag to represent a topic and lists to present services.

Moreover, the website also meets the second requirement of the guidelines. The code is written with a logical sequence and switches the user's attention from one

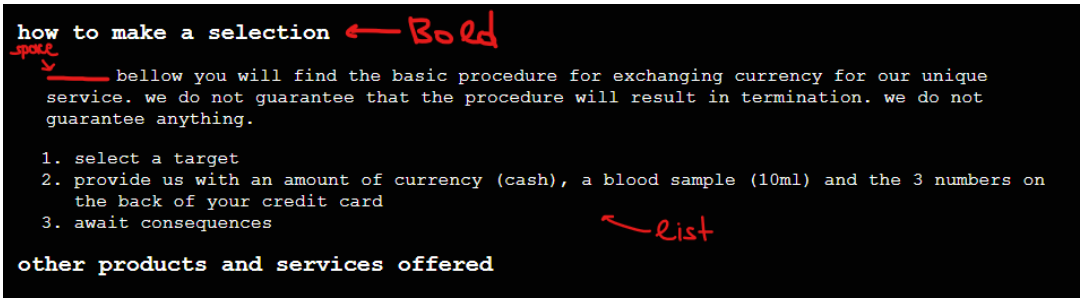
topic to another smoothly. One thing that I did not like is the fact that the tables have no margin for the text and neither is the text aligned.

However, the third requirement is not met fully. People with disabilities may not be able to scroll down to read the paragraphs written below.

Guideline 1.4:

- Use of Color
- Audio Control

The first requirement is met, as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element is not associated with color. Throughout the website, we can see that some elements are indented, therefore showing that it is a paragraph. Some are in bold showing that it is the title etc.



The second requirement has nothing to do with this particular design, therefore it is met.

Guideline 2.1:

- Keyboard
- No Keyboard Trap

The site meets the first requirement as it can be fully functional even using the keyboard only. However, there might be a slight keyboard trap, since the arrow keys do not work on the web, and using **“tab”** is the only possible way of moving around.

Guideline 2.4:

- **Bypass Blocks**
- **Page Titled**
- **Focus Order**
- **Link Purpose**

As mentioned before the website is easy to navigate through and it has certain ways of using headers, bold text, lists and such to help users understand where they are and navigate.

Responsiveness to different devices.

To test the responsiveness of the group's website to different devices, we have used our laptops and the Toggle Device Toolbar, in order to simulate a mobile device, tablet or a laptop of different sizes.

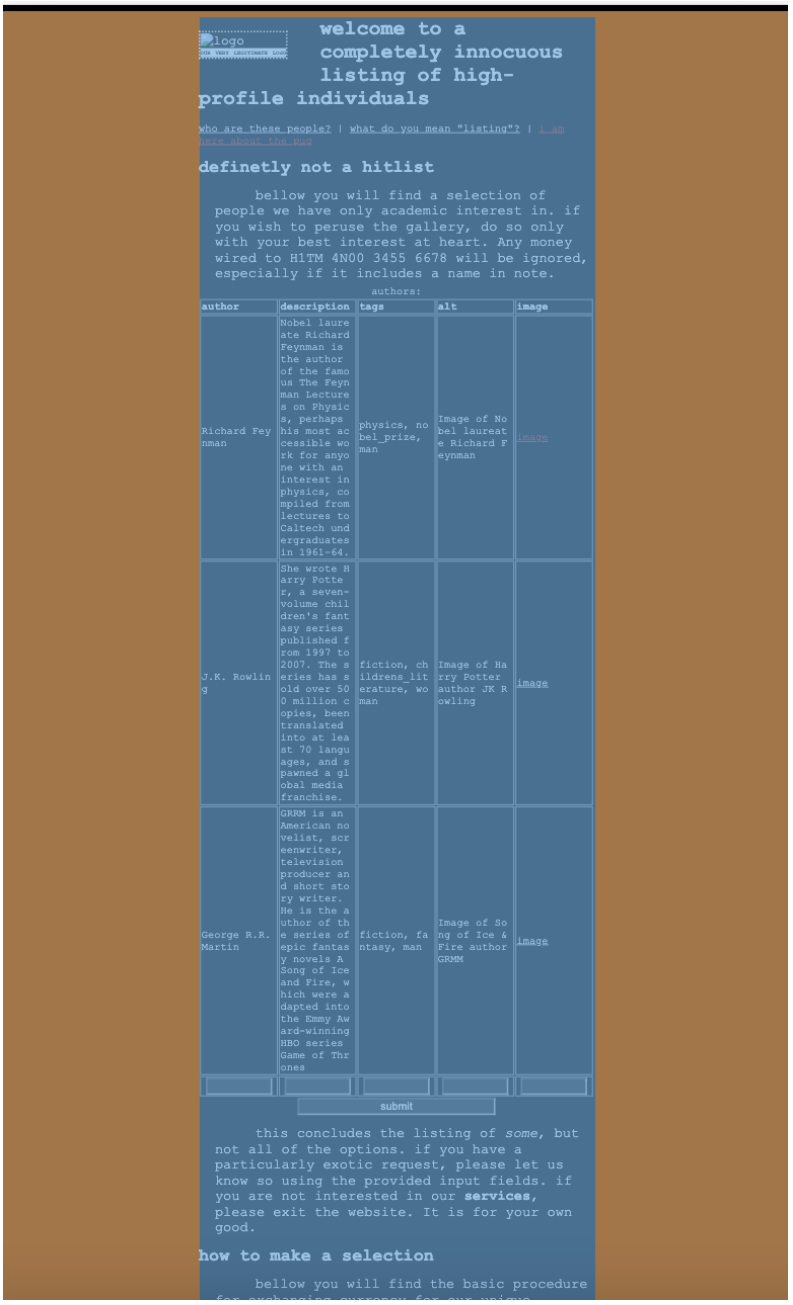
There was a bar on top, with preset media query breakpoints, such as: Mobile S, Mobile M, Mobile L, Tablet, Laptop etc. It allowed us to easily check if the group's website was responsive to each device.

That is an example of how the website would look like on an Iphone 12 Pro :

As we can see, it looks pretty good and the responsiveness is on point.

However, as we can see from the picture below, the margins and width are kind of annoying and would be more eye-pleasing if they were changed.

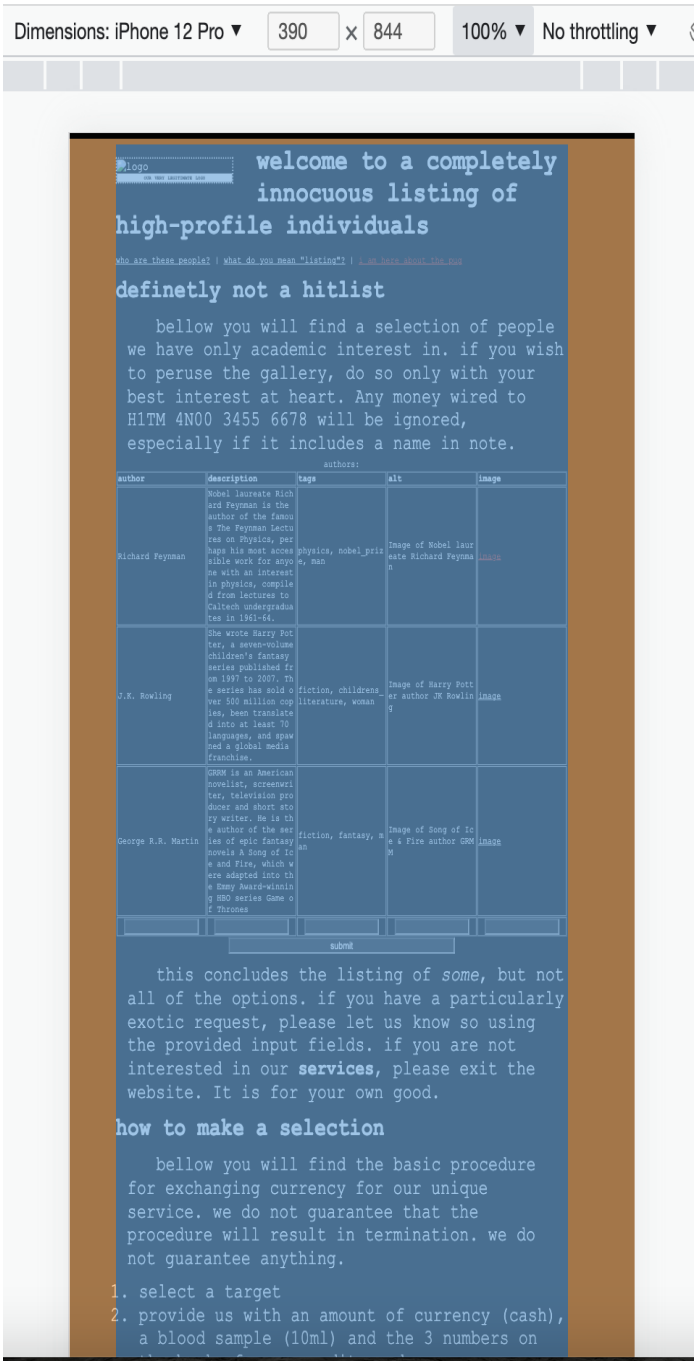
The orange part is the margins :



They could be changed to this:

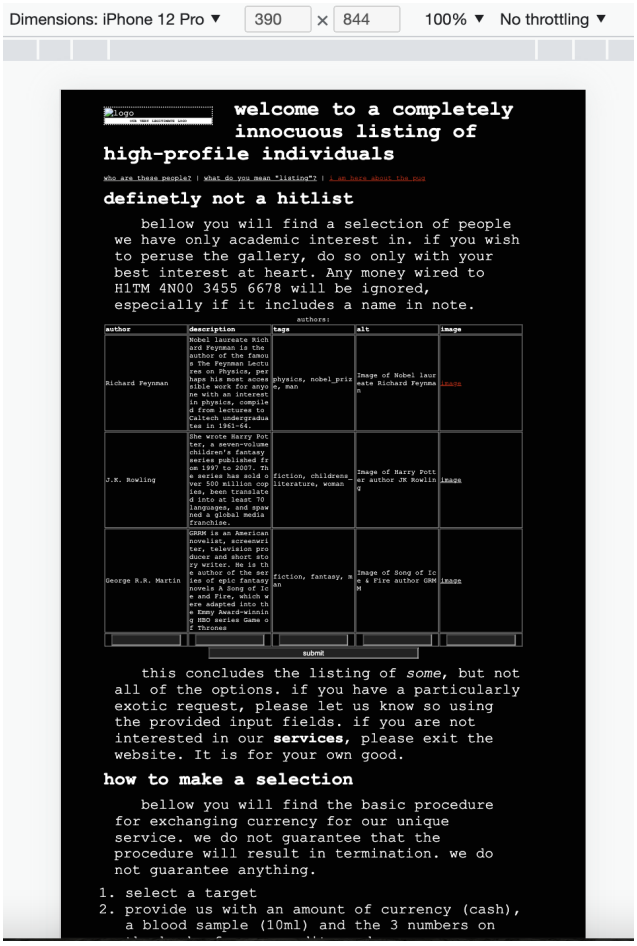
```
    margin-left: 80px;
    margin-right: 80px;
    width: 80%;
```

Here’s how it would look like after the margin-left and margin-right have been changed from auto to 80% and the width from 50% to 80% :



It looks better doesn’t it?

Here’s the same picture after the changes, but without the highlighting :

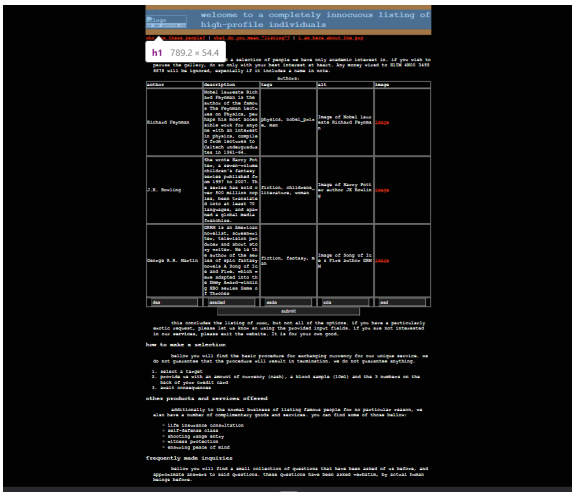


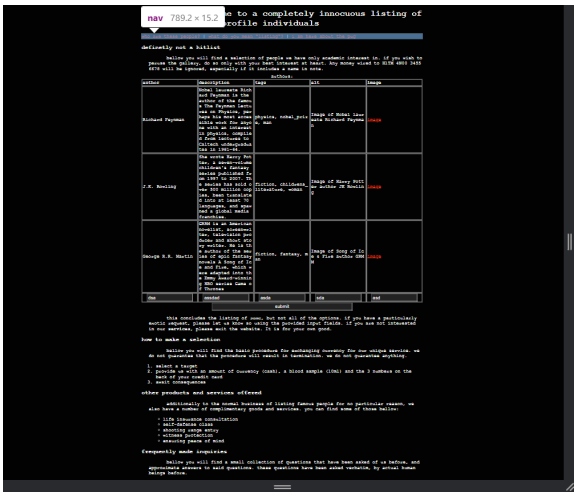
To conclude, the group’s approach to web design resulted in a website that rendered effectively on a variety of devices and screen sizes from the smallest to largest display size, ensuring usability and satisfaction, but again , a simple change in margins and width would make it look even better :)

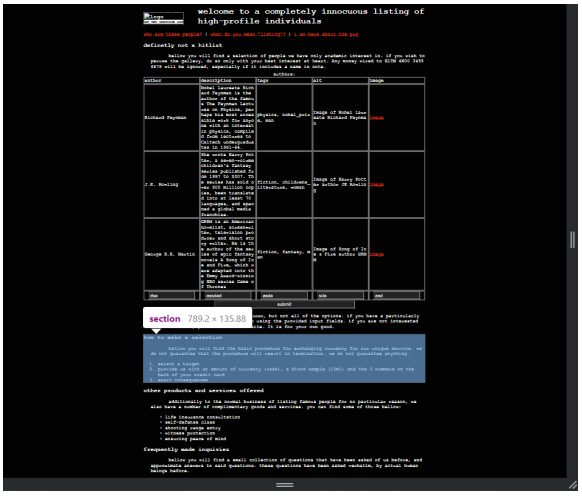
Semantic Markup.

```
<body>
  <figure>…</figure>
  <h1>welcome to a completely innocuous listing of high-profile individuals</h1>
  <nav>…</nav>
  <section>…</section>
  <section>…</section>
  <section>…</section>
  <section>…</section>
  <section>…</section>
  <footer>…</footer>
```

You can see the semantic markup in the image above. The body of the code has several sections, a nav bar, and a footer. The semantic markup of this code is good, however, it could be better. The fact that the <section> tag was used several times instead of adding another tag makes it a bit confusing which section stands for what part of the code. Other tags like header, aside, article, figure, etc. could have been used instead to make the semantic markup of the code better.







This is what the website looks like if you were to examine it in the **“Chrome Developer Tools”.** It looks clear and makes the user inspecting familiar with the code.

Mobile-first vs desktop-first.

The mobile-first approach is generally considered to be the better option when writing code in HTML.

As a person who has also completed this task, I can say that adjusting a website to work as intended on a mobile device is harder than on a large-screen device.

Moreover, since the amount of users that are joining the internet on their mobile phones is rising, working on a mobile version first makes it more likely that they will have a positive experience when accessing the website on their mobile devices.

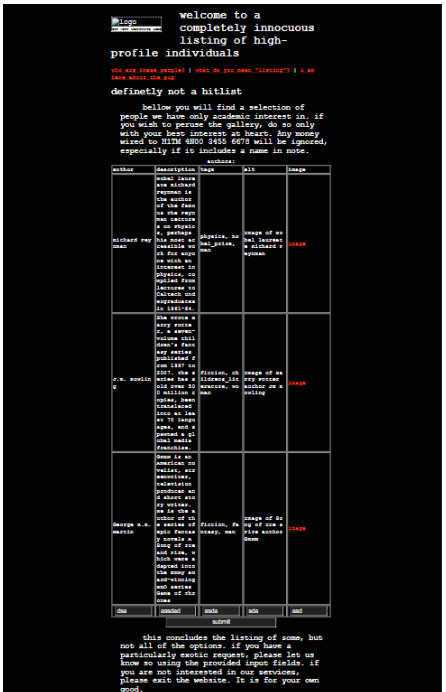
Additionally, because phones perform worse than laptops/ PCs, the mobile-first approach ensures that the website will be optimised for both phones and desktops in terms of bandwidth.

Furthermore, developing a website with a mobile-first approach ensures that users with disabilities will also be able to use the website equally the same as the others.

When it comes to the team's choice. It can be evaluated that they had a desktop-first approach. To come to this conclusion, we decided to compare how the website performed on a laptop and a phone. We did this using the **Google Developer Tools**, more specifically the **device toolbar**.

This is what the website looks like on an iPhone 12 Pro.
As you can see the text is too small to see without zooming in. This is due to the margins on the left and right of the website, which shrinks the size of the table making the columns thin.

iPhone 12 Pro:



Desktop:

Actionable feedback.

First of all, the title could be shorter and start with a capital letter, thus making it more eye-catching.

Also all the following headers, paragraphs etc. could also start with capital letters, because if they start with lower-case letters it makes everything look monotone, however the divisions are still clearly seen due to the headers being more vivid than the paragraphs and we guess lower-case letters is the “style” the group was going for.

Second of all, the images should be PNGs instead of links to the Wikipedia pictures. It would be much better if there were already images in place, instead of having to click and be sent to another page with the image, thus saving time and the nerves of the user.

Finally, the logo is not shown properly.

Conclusion.

In conclusion, we have learnt more about the semantic markup idea. For instance, a semantic tag like "header" or "footer" can be used to explicitly communicate the purpose of a particular section to both users and web browsers, as opposed to using a generic tag like "div" to construct a section on a page. We learnt that this can help with search engine optimization, accessibility, and making it simpler to apply CSS styling to the page.

We also learnt that having too many nested elements or too many CSS classes can make the HTML difficult to read and difficult to comprehend the page's structure.