

# BBC Coding task evaluation

---

## *The Task*

---

I broke the task down into a list of requirements before I began so I could plan the different sections of the code:

- Read articles from JSON data using GET requests
- Load in each article one at a time using adaptive CSS and semantic HTML
- Rate each article and POST the results

This outline pretty much shaped the rest of the project for me. Next, I considered some of the bonus tasks:

- Accessibility
- Preloading articles
- Testing

I made time to implement these tasks but still made sure the time spent was not to the detriment of the core functionality.

---

## *The Opportunities/Hurdles*

---

Right off the bat it was clear certain areas of the requirements specification were going to be a challenge to me. I treated the new concepts as opportunities to learn rather than hurdles to get over – in my opinion the right mindset is half the battle.

- JSON Data  
I have never used JSON before in my life, sure I had heard of it but never had the time or need to learn about it properly.  
With my course at university reaching an apex, JavaScript has been included a lot more over the last few months and I was recently given a task that included grabbing data from an online XML file using a GET request. Making a link between these two requirements made it easy to parse the data from the JSON file and after a quick research session I was spitting out the data faster than you can say 'asynchronous'.
- POST Request  
I have used POST requests plenty of times before whilst programming other webpages in PHP so this section of the requirements spec actually snuck up on me.  
The plan seemed simple enough before implementation but in actual fact when it came around I was a bit lost because I did not originally plan on using any server side scripting.

This obviously turned out to be a huge disadvantage, but, again after a bit of research and trial and error, the ratings were being uploaded to the XML file no problem.

---

*“I love it when a plan comes together”*

---

My original plan <sup>(1)</sup> was extremely simple and quite naïve in hindsight. I allocated a single day for each task from the requirement spec although some tasks could take 10 minutes, and some could take a few hours. Due to this, after falling behind slightly I updated the plan <sup>(2)</sup>.

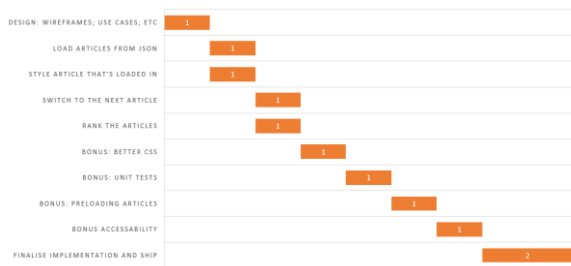


Figure 1

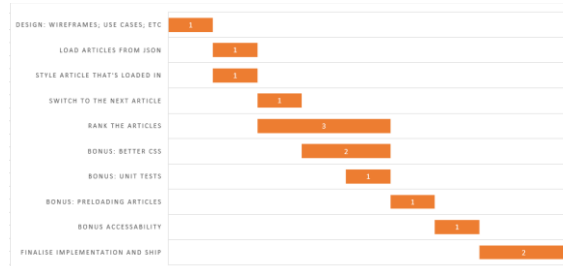


Figure 2

The project was slowed down a bit during the week due to an illness I had, hence the updated chart. In reality however, the final distribution of time was different yet again. I did not write any documentation for testing each unit of the software and spent a much larger amount of time on ranking the articles as this involved some PHP techniques that were new to me.

In terms of design, the webpage <sup>(1)</sup> looked fairly similar but a lot of the elements I had included in the diagrams were either unnecessary or just too impractical to implement. I ended up not including the “Related articles” <sup>(1.1)</sup> section, since the articles contained no way of identifying the common themes, this would be more hassle than it was worth as I would have to translate the articles which were made up of just placeholder text by the looks of it anyway.

Rating articles turn out quite different as well <sup>(2)</sup>. Instead of rating the articles at the same time, they are rated one by one <sup>(2.1)</sup> and the data is added to the XML each time you move onto the next article. They are then displayed on a third page <sup>(3)</sup> which was not included in the wireframe diagrams where they are listed with their rating next to each one. An option to toggle a high contrast mode was also added to each page.





1



2



3

---

## *The Good*

---

All in all, I am quite happy with the final product from over the last two weeks. Some parts of the project that I am proud of:

- The CSS  
The CSS works great overall and adapts well to whatever screen size, including mobile devices and tablets by using media queries to check the size of the screen. On mobiles the design has a more centralised look. The high contrast mode works a treat on any device too and is easy to toggle on and off at any time.
- The XML  
I have only worked with XML once before, so the state of the rating system is something I am quite happy with. Before ratings are submitted, the XML file is emptied of previous ratings and every time a rating is submitted it is added to the XML file using AJAX.
- Loading articles  
Loading articles is a breeze. Once the user jumps into the list of articles they are all loaded into local storage, so the load time is drastically reduced for each subsequent article. This also means that if at any point after that the user loses network connection, they can keep reading the rest of the articles until the list runs out.

---

## *The Bad*

---

Although the project was successful in many ways there are also a few things I did not have time to implement or just missed out during development:

- Testing documentation  
Even though I have made sure each part of the website works as intended, I did not write any documentation or plans to check this. This is simply due to a lack of time but would have been something I included given an extra day or two.
- Bootstrap  
I am relatively familiar with the bootstrap CSS library however, I did not make use of it in this project. This is because I thought there were already a few challenges for me to take on in terms of the JSON data and XML and I did not want to overload the projects with new things to learn and get the hang of.

---

## *The Ugly*

---

The last of the three sections evaluating my code, is 'The Ugly'. This stuff works but is a little bit hacky in some places and could do with a bit of streamlining and TLC.

- **Article stubs**  
In some places I have made use of what I call "article stubs". These are just small previews of an article showing the title and image if there is one. While they look nice, the code that grabs and displays them is repeated a few times due to the stubs being shown individually in one place and as a list in another. I should have planned this in advance and made a function that just returns the article stub information, this would get rid of the repeated code and clean up the project a bit.
- **Displaying the ratings**  
Unfortunately, I did not have time to squash every bug. When displaying the ratings, sometimes they do not show at all or the show in the wrong order. This is because of the asynchronous nature of the way the data is passed around I think as the JavaScript seems to sometimes read an empty file. Refreshing the page a few times at most will resolve this problem so it is not a huge error, but something would like to have fixed if given the time.

---

## *Conclusion*

---

With some minor bugs to fix and streamlining issues the project is not flawless, but I think the work I have produced is worthy of merit in a lot of other places. After the project it is clear there are things in the planning stage of the design that I need to think about more, rather than rushing in to start coding as soon as possible. The front end of the website is an good achievement of mine and the opportunities to learn new things that the project has presented has been invaluable. Thanks to the BBC giving me a good challenge and a great opportunity to learn.