Web Services AE1 Report: Online Article

Abstract

The website created in this project follows the structure of the provided ONS website and its narrative. A navigation feature has been added at the top, containing the contents of the page and a link to each section of the article. The article has six sections in total, where the narrative is based on the ONS webpage. Section one is about the main points copied from the ONS to give the page a feel of completeness concerning its content. Section two is the same as on the ONS page about greenhouse gas emissions and it shows the first interactive visualisation of the application. Section three is about the greenhouse gas emission intensity and contains the second interactive visualisation. Section four is about energy use and it shows the third visualisation based off the datasets from the ONS. Section five is a narrative section about the environmental goods and service sector and has been added to create an even number of sections and establish a feeling of integrality. The final section includes the live visualisation of the current mix of fuels used to generate electricity in the UK. The data is fetched from an API and the visualisation changes every five minutes after the live data got re-fetched.

Code and Requirements

(for code explanation, please check the comments in the HTML, JS and CSS file)

The application is responsive to changes in screen size. When the size decreases to under 768px, the columns from the table of contents merge into one and so do the legend items of the plots. In order to make the plots responsive, the change that has been determined to be beneficial on smaller screens, was for the lines of the plot to become less disorderly. On smaller screens it is more difficult to make out a trend of a messy plot than it would be on bigger screens. Therefore, as the screen size decreases, the original plots get replaced by a smoother plot with less data points to show the trend clearer on smaller screens. Additionally, the font sizes of the plots change, in order to stay readable on smaller screens. In this particular case, the plots' interpretability would still be maintained, as the lines are not too disorganized but in the case of more uneven data, the reduction in screen size may result in a plot hard to understand. So, for each plot, a second plot has been created where the data goes in 2-year steps in order to smooth the lines.

The index.HTML file of this project contains the skeleton of the site. The body includes the header of the page, the navigation section with the contents and is followed by the main tag and article tag. Those tags include the titles of the sections, their corresponding IDs and they envelope all sections of this page and the visualisations. The design of the page is kept simple and, in the semantics, it has been prioritized to keep a stable form and create one long article with various different sections and in those sections, multiple divisions. In total there are six sections within the article, three of which have visualisations with the data from the ONS website and one containing a live visualisation formed from the data of the requested API.

In order to create the plots from the data from the ONS page, a CSV file had been downloaded and turned into a JSON file using pandas. By using pandas to read the file, it had been noticed that a few lines, containing the information about the data at the top, had to be skipped. Then there was an unnamed column presenting the years, so it was renamed to "Years". The file was then encoded and rewritten as a JSON file and saved on the desktop. It was moved to the same folder as the project and then used to create the plots.

The four visualisations, as requested, are: The greenhouse gas emissions from the ONS dataset, the greenhouse gas emission intensity from the ONS dataset, the energy use from the ONS dataset and

the current energy mix for electricity generation in the UK from the National Grid. The visualisations have been made interactive, so, hovering (on desktop) and clicking on any point of the plot will show a line connecting the same point in time for all the lines of the plot. Additionally, an information box, including the details of the data at the specified point, appears.

All the visualisations have been added to the sections containing the narrative about their topics, which have been copied from the ONS website. Section one and section five have been kept in order to have an even number of sections and create a feel of completeness. For the final visualisation, a basic narrative was written and done in the same style as the previous sections, to make it fit in.

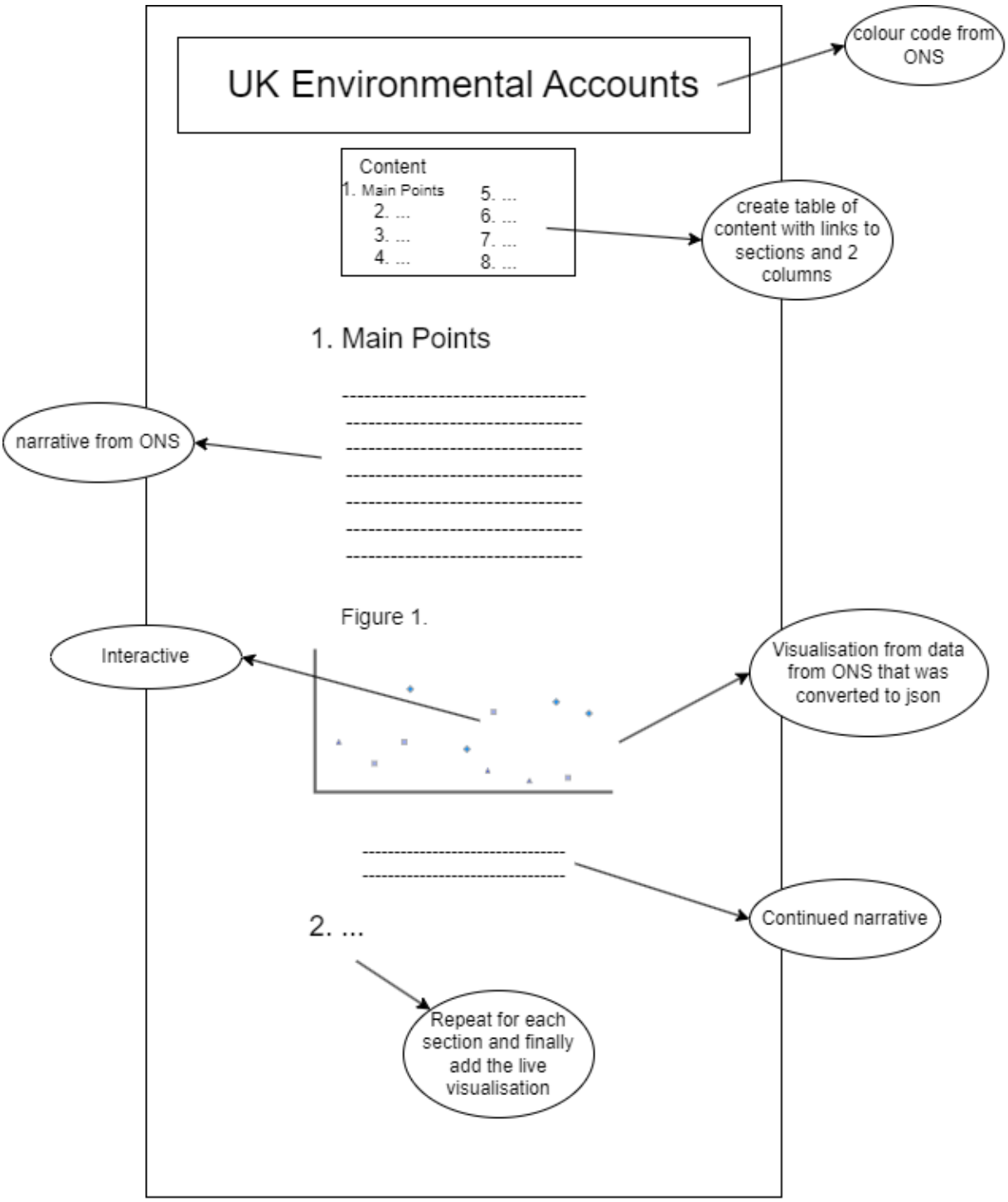
The web accessibility evaluation tool returned 0 errors, 0 alerts and it states that the website complies to WCAG AA and WCAG AAA. The narrative text colour has been experimented with and a slightly softer colour was chosen, but still dark enough to not cause accessibility issues.

For tools, the idea was to use Bulma for styling the page, but to ensure understanding of each step taken in creating the design, the choice was to not use Bulma and do everything in the CSS file manually. Flexbox was used for the navigation section to space the items in the table. Grid was used for the manually created legend container and flex was incorporated for the items in that container.[1]

[1] Further tools that have been used will be discussed in the architecture section

Design Decisions

The following is the mock up design of the web page, which has mostly been followed, with a few additions and changes:



The design of the page is heavily based on the ONS website. The article is centred, so when the screen size decreases, the change will feel more natural. For the table of contents, similar to the ONS page, two columns were created that merge into one when the screen resizes. The same was done with the items of the legend container. The narrative and visualisations have been positioned the same way as on the ONS page, with the addition of the live visualisation at the end. For usability, the

plots have been made interactive with the hovering features of the pointers from the Observable library and the tip function. The items in the table of contents are also linked to the corresponding section of the article, so the user can click on those and be redirected. For accessibility, the font sizes are chosen to be big enough, and especially after the resizing of the screen, while the narrative stayed readable, the font sizes of the plot labels were readjusted to be responsive and become bigger when the size shrinks in order to stay readable. The colours used, as mentioned above, were chosen, whilst trying to make sure that they have a strong enough contrast, but are also not piercing to the eye.

Environmental impact

Unfortunately, the design of this website is heavily coded due to the choice of doing everything manually. Especially concerning the plot and its responsiveness, as a result of not using Svelte, new plots had been created to replace the original when the screen size shrinks. This means a lot of extra JavaScript has been added to the project. Due to the formatting of the JSON file and how the plots have been created, the function of adding a legend was not possible directly in the plot. So, the legends for the plots have been created manually with multiple code snippets in the HTML file and the CSS file compared to one line, which is normally all that is necessary with Observable. This also added to the amount of code, leading to energy inefficiency.

A consideration made in favour of the environmental impact, was the setting of the time for the live visualisation to refresh and re-fetch the live data. Re-fetching the data every second had been tried and the observations were, that the plot was glitching with the constant changes and the device was starting to overwork itself. Five minutes were chosen as the time to set the setTimeout function to, in order to assure a consistent feel of the plot re-adjusting to the live data and to prevent unnecessary energy usage.

Architecture

The architecture is straightforward with six sections and four visualisations with their narrative that assist the data.

The tools used for this project include grid and flexbox for the navigation section and the legend container. Flexbox was used for the ordered list of the contents, to have an equal number of items in two columns next to each other. The observable library has a build in legend feature, but due to how the visualisations had been plotted by re-defining all lines, the legend feature rendered unusable, as mentioned above. To create a legend, the HTML and CSS file had been used and programmed manually with grid and flexbox.

The Observable library has been used for plotting the visualisations. The format of the JSON file resulted in creating the plot by fabricating various lines. For each lineY created, a stroke colour had to be specified in order to differentiate the lines and their meaning. A ruleX feature with a pointerX, plus a dot feature, have been attached to make the plot interactive. The tip function has also been included and the channels to display have been determined, to create an information box of the data at the point where the user either hovers over the plot or clicks on a point. The specified channels are the different lines of the plot. For the first plot, the margin on the left is appointed to be 60, because the Y values are too big and without the specification they would be cut off. The frame-anchor containing the information for the first two plots is at the top, right corner, but for the third plot it is at the left-hand side, because the line of the plot cuts through the box if it were on the right side.

Multiple media queries were used in the CSS file to ensure responsiveness of the page and adjust to changes in screen size. For the table of contents and the legend container, a media query has been generated to merge the two columns containing the items into a single one as the screen size decreases.

In order for the last plot to be live and therefore constantly changing, the SetTimeout function was utilized to follow a 5-minute timer, with which the data gets re-fetched from the live data in the API. The five minutes were chosen, to prevent a constant glitching of the plot that had been witnessed after setting the function to one second.

For responsiveness of the plots, event listeners have been used to check for window size changed and then determine when to switch the plots.

Reflection

As declared previously, due to the methods used to convert the CSV files and how the plots were created, the energy efficiency is not optimized. Using Svelte could have could have aided in optimizing the efficiency of the responsiveness of the plots and also made the process smoother, as now the plots are being abruptly replaced by different plots.

The choice of not using Bulma, led to each section being created manually, ensuring that there must be an understanding of the different design steps, but it also means that the application is kept at a fairly basic level and it does not particularly stand out.

The responsiveness that has been created manually helps with oversight of the trend on smaller screens, but it does remove some data points incorporated in the plot. Had the data been messier, then this replacement would have made more sense, but in this case, it is done more for exemplary purposes, to show how responsiveness can be built in.

The task of making the plots interactive has not been fully accomplished, as the integration of using a filter on the plot, has failed. It had been attempted to achieve by using the checkbox function together with viewof, but it did not work out. The whole creation of the plot should have been optimizable, but at the current stage, the way to do so has not been recognized.

References (used in code):

HubSpot Developers, “Centering your website using max-width and auto margins”: https://designers.hubspot.com/docs/snippets/design/centering-your-website-using-max-width-and-auto-margins (Last accessed: 02.03.2024)

MDN Web Docs, “Element: innerHTML property”: https://developer.mozilla.org/en-US/docs/Web/API/Element/innerHTML (Last accessed: 02.03.2024)

MDN Web Docs, “flex”: https://developer.mozilla.org/en-US/docs/Web/CSS/flex?retiredLocale=de (Last accessed: 02.03.2024)

MDN Web Docs, “flex-direction”: https://developer.mozilla.org/en-US/docs/Web/CSS/flex-direction?retiredLocale=de (Last accessed: 02.03.2024)

MDN Web Docs, “flex-wrap”: https://developer.mozilla.org/en-US/docs/Web/CSS/flex-wrap?retiredLocale=de (Last accessed: 02.03.2024)

MDN Web Docs, “grid-template-columns”: https://developer.mozilla.org/en-US/docs/Web/CSS/grid-template-columns?retiredLocale=de (Last accessed: 02.03.2024)

MDN Web Docs, “Node: appendChild() method”: https://developer.mozilla.org/en-US/docs/Web/API/Node/appendChild (Last accessed: 02.03.2024)

MDN Web Docs, “Window: resize event”: https://developer.mozilla.org/en-US/docs/Web/API/Window/resize_event (Last accessed: 02.03.2024)

Mike Bostock (2021), Observable, “Input: Color”: https://observablehq.com/@observablehq/input-color (Last accessed: 02.03.2024)

Mike Bostock (2024), Observable, “Input: Checkbox”: https://observablehq.com/@observablehq/input-checkbox (Last accessed: 02.03.2024)

Observable Plot, “Bar mark”: https://observablehq.com/plot/marks/bar (Last accessed: 02.03.2024)

Observable Plot, “The JavaScript library for exploratory data visualization”: https://observablehq.com/plot/ (Last accessed: 02.03.2024)

Observable Plot, “Pointer transform”: https://observablehq.com/plot/interactions/pointer (Last accessed: 02.03.2024)

Office for National Statistics (ONS), released 5 June 2023, ONS website, statistical bulletin: https://www.ons.gov.uk/economy/environmentalaccounts/bulletins/ukenvironmentalaccounts/2023 (Last accessed: 02.03.2024)