Version 1

Main goal – Display multiple webpages and can go between them

Result:

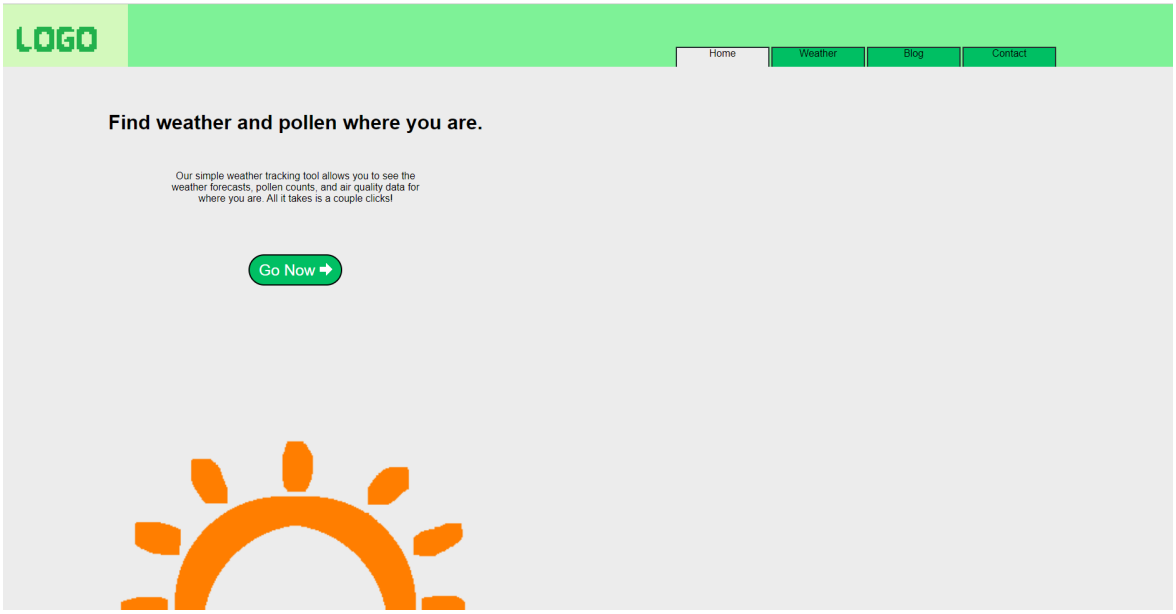
I have achieved exactly the above goal, with there being two pages that connect to each other. Neither of them look the part at the moment, but this version is much more a test of Flask than focussing on user experience. I also managed to include an image – this will be very important later down the line. As I was not given a logo for Health Advice Group, I have made a stock one to act as a placeholder.

The folder structure has been changed slightly to better match the Flask recommended structure.

The next task is to flesh out the homepage so that it matches the design I created (except for showing a blog post)

Version 2

Main goal – Homepage matches design created in Task 1
Result:



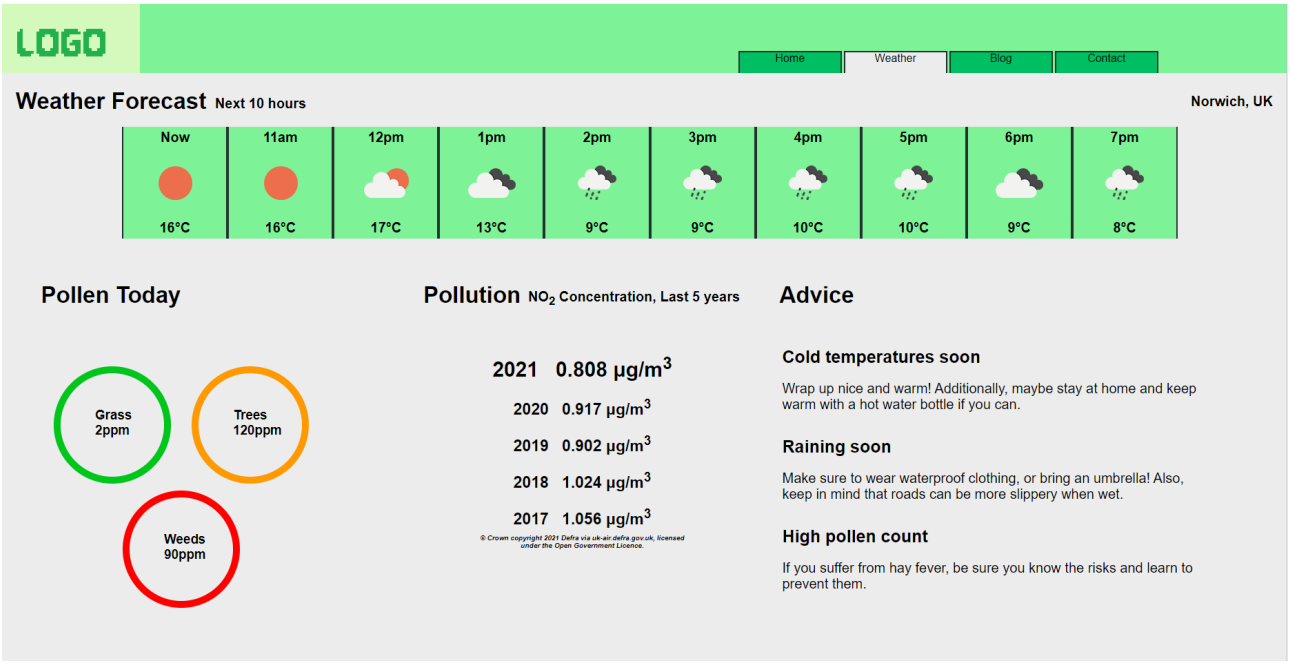
After further adjustments to the folder structure to allow for CSS files to be separate from their HTML so that it is easier to understand the code, the homepage is now much closer to the initially proposed design. This version uses the correct colour pallete, and I made my own version of the sun image so I could use it without copyright issues.

There are some key things missing from the design, however. Notably, the entire featured blog section on the right, as the blog system has not been implemented yet. Additionally, the setting for a dark mode toggle is also missing, as I will save accessibility features until later, after the core of the site works. Finally, the design shows each of the links at the top as if they are dividers in a folder; however, I could not find a reliable way to transform the links to be more trapezial.

The next goal, and the hopeful result of Version 3, is a working weather system. This will take some time.

Version 3

Main goal – Weather UI and connect to backend for data generation
Result:



The design for the weather page is complete, as seen here. Some changes were made from the initial design, however. Firstly, the weather forecast has a green background, which was chosen so that the weather images (thanks to OpenWeather for their free weather icons) can be clearly seen, There was initially the same grey background, but it meant the clouds weren't visible at all. Secondly, the circles that show each type of pollen have been changed to rings so that the text in the center is clearer. Thirdly, the line graph for air quality has been changed to just showing the numbers, with the most recent data larger and at the top, and the title has been changed from 'Air Quality' to 'Pollution', to make it clear that lower is better without making the user feel stupid. Finally, the layout of the 'advice' section has been rearranged slightly to be clearer.

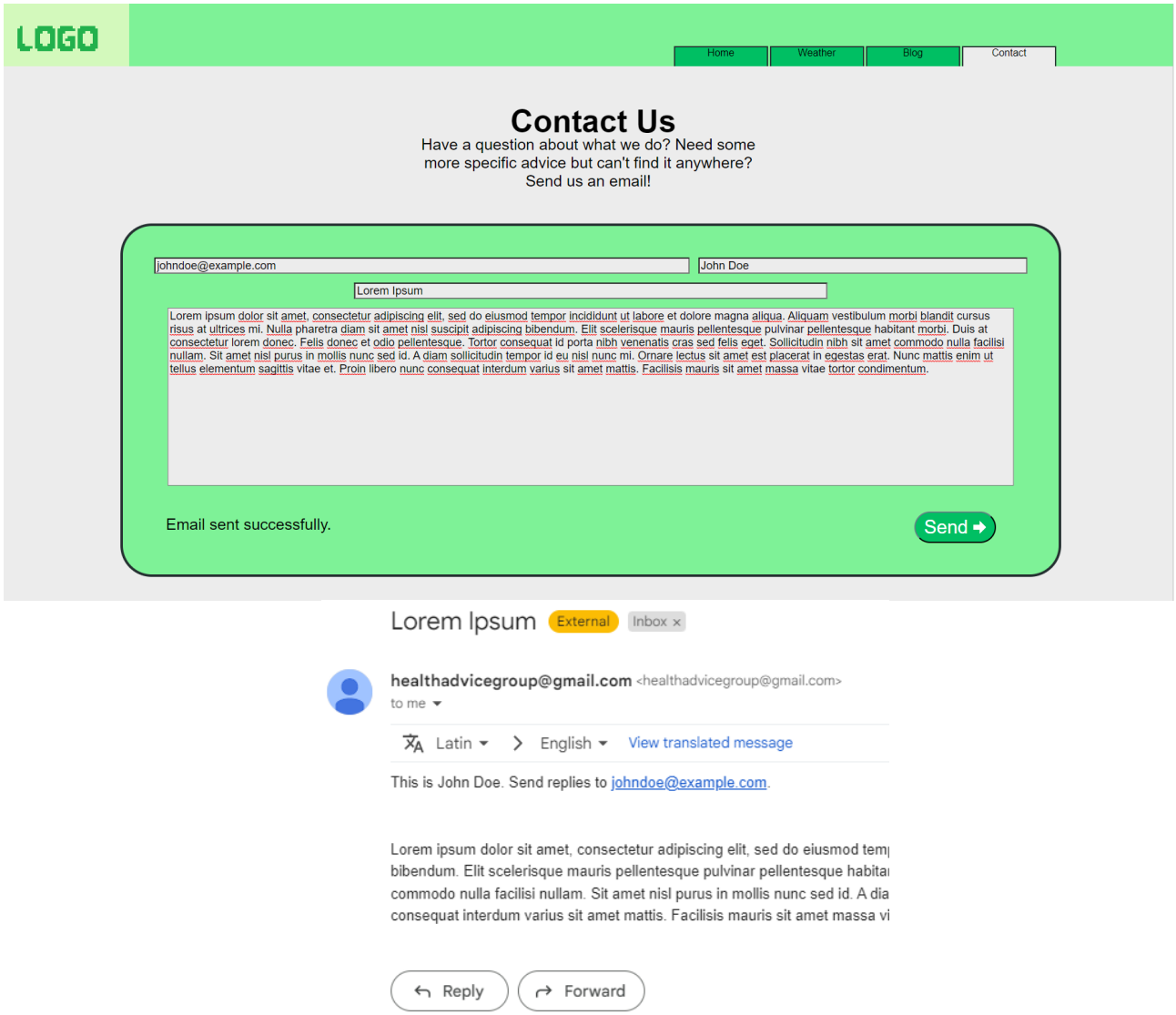
In the backend, there has also been some changes. The site can now respond to a POST (not GET anymore, see test log) request to /weather/content, which causes it to generate

the weather data. For now, it is just using sample data instead of actually connecting to APIs and map data, so that the frontend could easily be tested. Also, the 'go now' button on the homepage actually works now.

In the next version, I will move away from the weather systems and start working on the contact page.

Version 4

Main goal – Completely functional contact page that can send emails
Result:



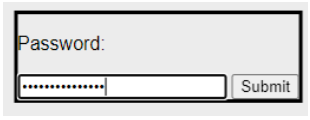
The contact page is now fully up and running, with a UI that almost perfectly matches the originally planned design from Task 1 and a working mail system that can send emails from healthadvicegroup@gmail.com, as you can see in the example above. For this to work, the Python Yagmail third-party module was used to make sending emaiils easier, though I still had some struggles allowing it to interact with the Gmail account.

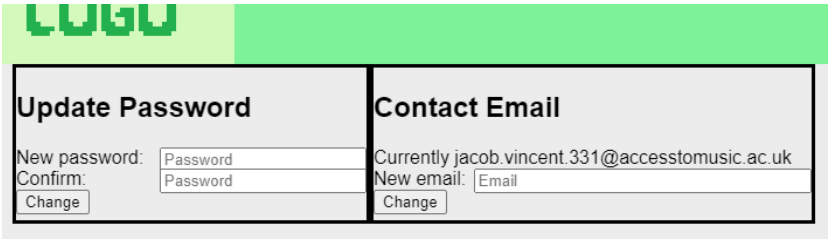
The only thing that was changed between the originally planned design and the result is the addition of a small message that says 'Please wait...' once the send button is clicked and 'Email sent successfully' once the process is finished. The goal of the next version is to create the admin panel and allow it to change the destination email.

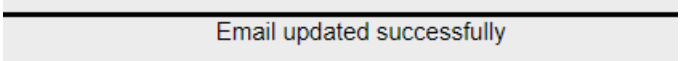
Version 5

Main goal – Admin panel creation, with a password lock, change password, and change destination email for contact

Result:







The admin panel is now starting to take shape. By going to it at /admin (there is no link to it on the site so it is less likely to be accidentally come across) you are asked to enter your password as seen in the left image. If the password is entered incorrectly, nothing will happen. This acts as a basic extra security measure, as if someone tried to guess the password and got it wrong they would think the system is currently offline and wouldn't keep guessing.

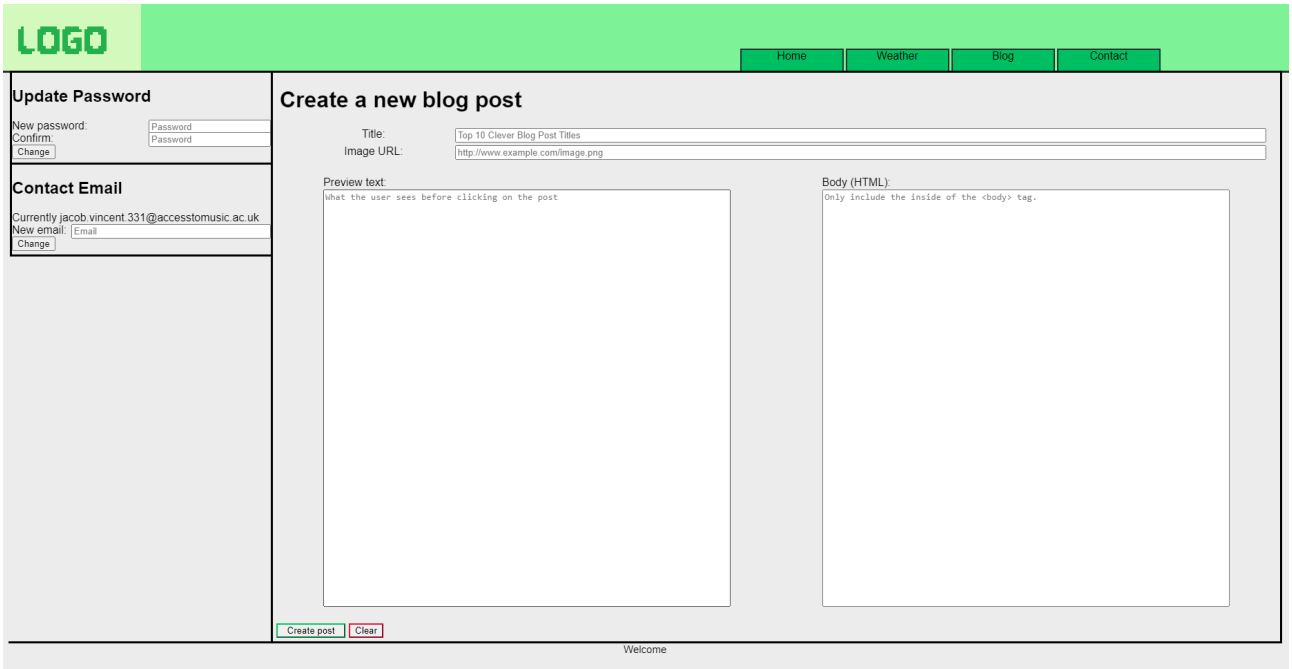
While the plan noted in Task 1 was that the password would have to be entered after every single interaction, it was decided that the user would only have to enter the password once so that it doesn't get irritating for them; besides, the page can be refreshed to make the password popup appear if needed.

The admin page itself currently looks a bit empty, so there is space to add a blog creation system in the future. There are three features on the page at the moment: a space to change the password, a space to change the contact email, and a message at the bottom that updates to show the status of changes made.

For Version 6, the goal is to add to this panel a system for creating blogs. This will fill the right-hand side of the screen and allow the user to create posts with custom HTML.

Version 6
Main goal: Blog system (all of it)
Result:

There is a lot here to cover, so images will be shown as they become relevant. First, is the new section on the admin panel:



There are four input boxes on this page. These are for entering:

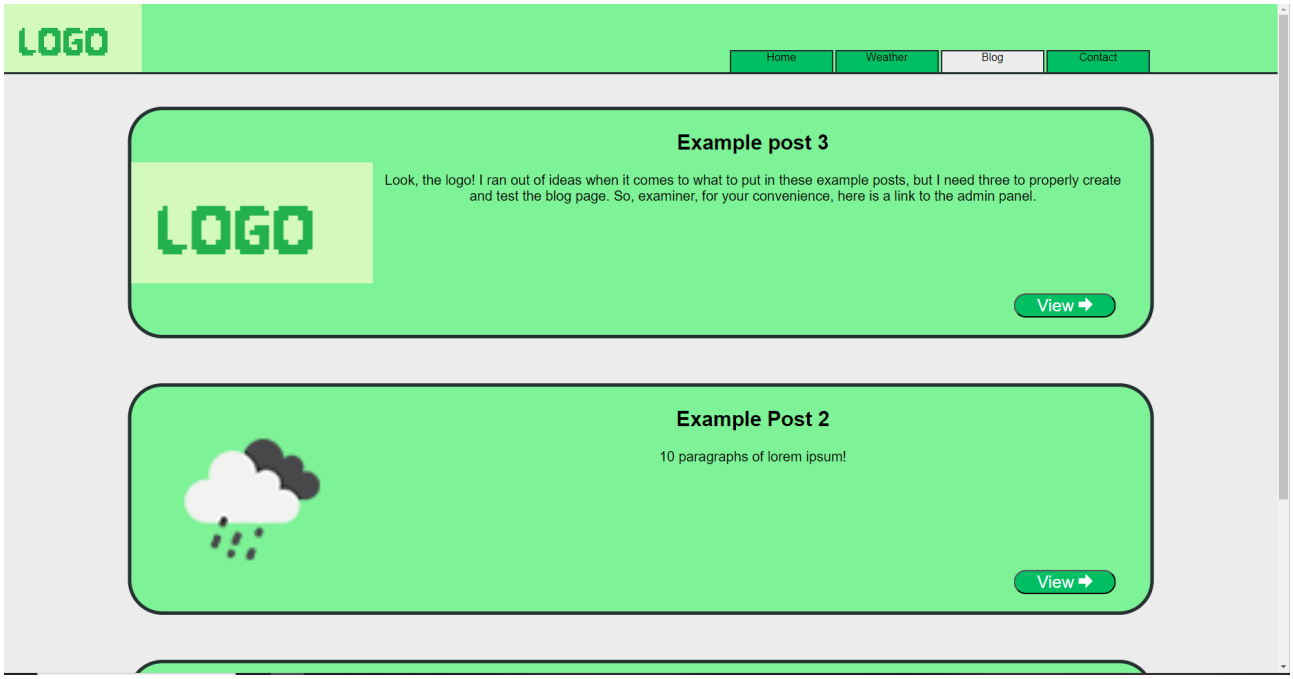
- The title of a blog post – self explanatory
- The URL to an image for the thumbnail – this can be stored locally in the backend or be a link to any other image on the Internet.
- The preview text – This is what appears to the user on the blog page and the homepage before they click to view it.
- Body (HTML) – This is where the contents of the page itself are created. It is as raw HTML so the creator has free reign over what to do with it.

The buttons at the bottom are coloured according to what they do, so there is less risk of accidentally pressing the wrong one. When you press 'Create post', there is some error checking to make sure none of the fields are empty. Using ths, I have created 3 example posts so that you can see some examples.

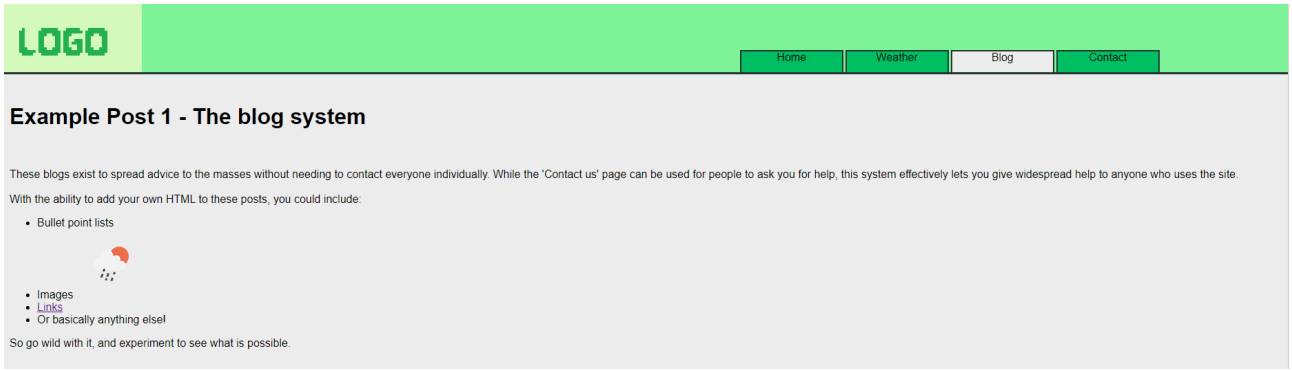
There is still currently a gap in the lower left; this is where the space for updating blog posts will be in the next version if I have the time.

Displaying blog posts

The most recent blog appears on the homepage, as was initially decided. The design for this is almost identical to that of the design initially proposed. Also near-perfect is the main blog page:



This page came out exactly as was initially planned, with the ability to see two posts at once and a third partially visible to let the user know they can scroll. Each preview shows all the information it needs to, with its own 'view' button taking you to (a variation of) this page:



This is the view page, where the contents of a blog post are clear to see. The page starts completely blank apart from the navigation bar, but is filled in with the necessary HTML from that blog post.

New blog JSON structure

The biggest change made from the proposed design is the internal structure of a blog post.

Before:	After:

{
id : generated with UUID4 algorithm
image_loc: URL to image
preview: number of characters in main body to show as a preview
contents: {
title: blog title
body: HTML body
}
	{
	html: HTML body
	image: URL to image
	preview: preview text
	timestamp: time blog was created, in milliseconds since the start of 1970
	title: blog title
	}

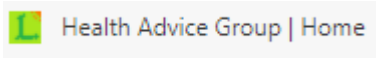
}	

The main changes are:

- 'id' was replaced with 'timestamp'. It's unlikely that two blogs will be created in the same millisecond, so can still be used to uniquely identify a post, but unlike the UUID4 algorithm this can also be used to find the creation order.
- The things in 'contents' have been moved outside of the structure. There was really no need for 'contents' to exist to begin with.
- 'preview' has gone from the number of characters in the body to use as a preview to just the preview itself. This prevents issues regarding the HTML tags and allows for the preview to be different than the start of the blog.

Other features

Dark grey bar underneath the navigation bar – this is because the blog page looked odd when scrolling without it.

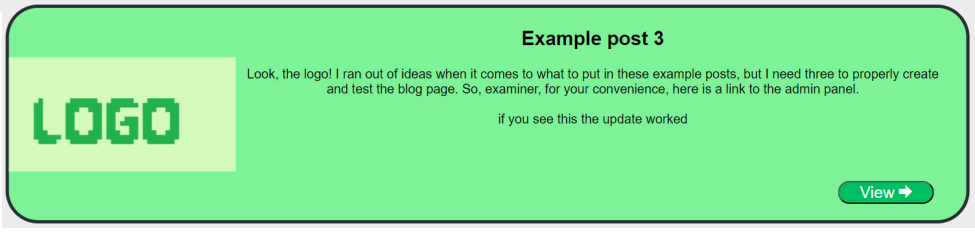


There is an icon now. The design here is an L for 'Logo here' with a small sun in the corner. Like the logo image used elsewhere, this is only meant to be temporary and is used because I have not been provided with a logo for Health Advice Group.

If there is time for Version 7 (I have 3 hours left), the only goal is to be able to edit posts on the admin panel.

Version 7

Goal: Add the ability to update blog posts
Result:



The space on the left is now filled with a list of existing blog posts. They are shown with only the title and image, as the creator should really be able to recognise their work. Clicking on the orange 'Update' button (colour chosen to not get mixed up with nearby green and red ones) will turn the blog creator into the editor you can see in the above image. The current data is loaded into the existing fields, which makes it easier to just change what is already there and is another reason why it isn't needed on the left. The clear button also serves an extra purpose in this mode – it allows for the user to go back to creating new blog posts. As you can see from the lower of the two above images, this system is fully functional. While creating this edit to test it, I realised that the preview text can also have any HTML tags in it, which makes it more powerful than before and allows for tings such as the newline in that same image.

The plan for Version 8 was to start working on API integration, so that the weather page isn't just dummy data, but that would take a week minimum and I only have 1 hour left. However, the sample data that is provided should give you a clear example of what the system *would* look like if given the extra time to incorporate it.