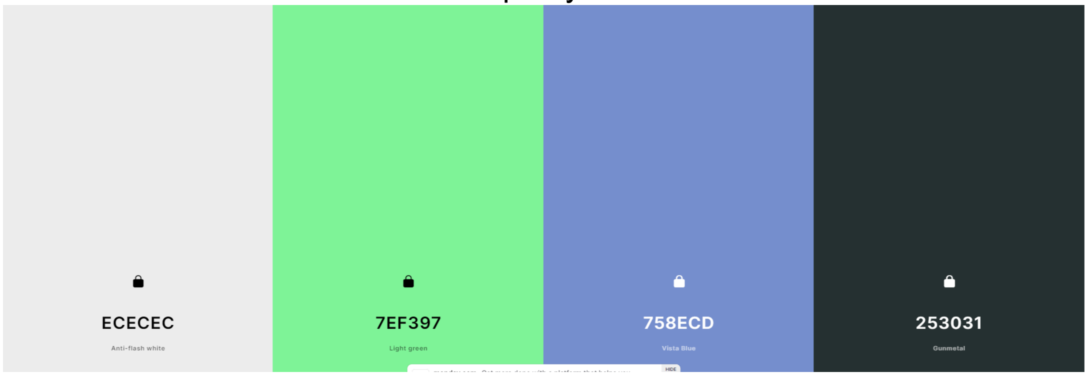
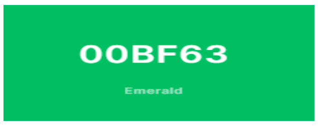
Colour scheme

The website should have a colour scheme that is friendly to look at. For example, a red-based system may not be ideal as it has connotations of emergency health where this site is more about day-to-day care. Pastel green on white is a good choice, as green has connections to life and white is a colour of purity.

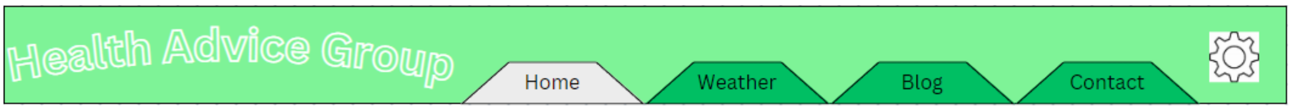


As a light-mode palette this would be an ideal colour set. The shade of white on the left is chosen to not be blindingly bright as a background, so as to not damage the users' eyes. The pastel green and medium blue colours would be great as highlights due to the green's aforementioned connotations and the blue has links to flowing water and weather. The grey on the right would be used for a dark mode background, while pure black is used for the text in light mode and pure black in dark. The darker green below may be used as a replacement for the blue; the blue and green together provide too much contrast and are unappealing to look at in certain scenarios. This darker green also has a very natural feel to it.



Site Breakdown - Navigation

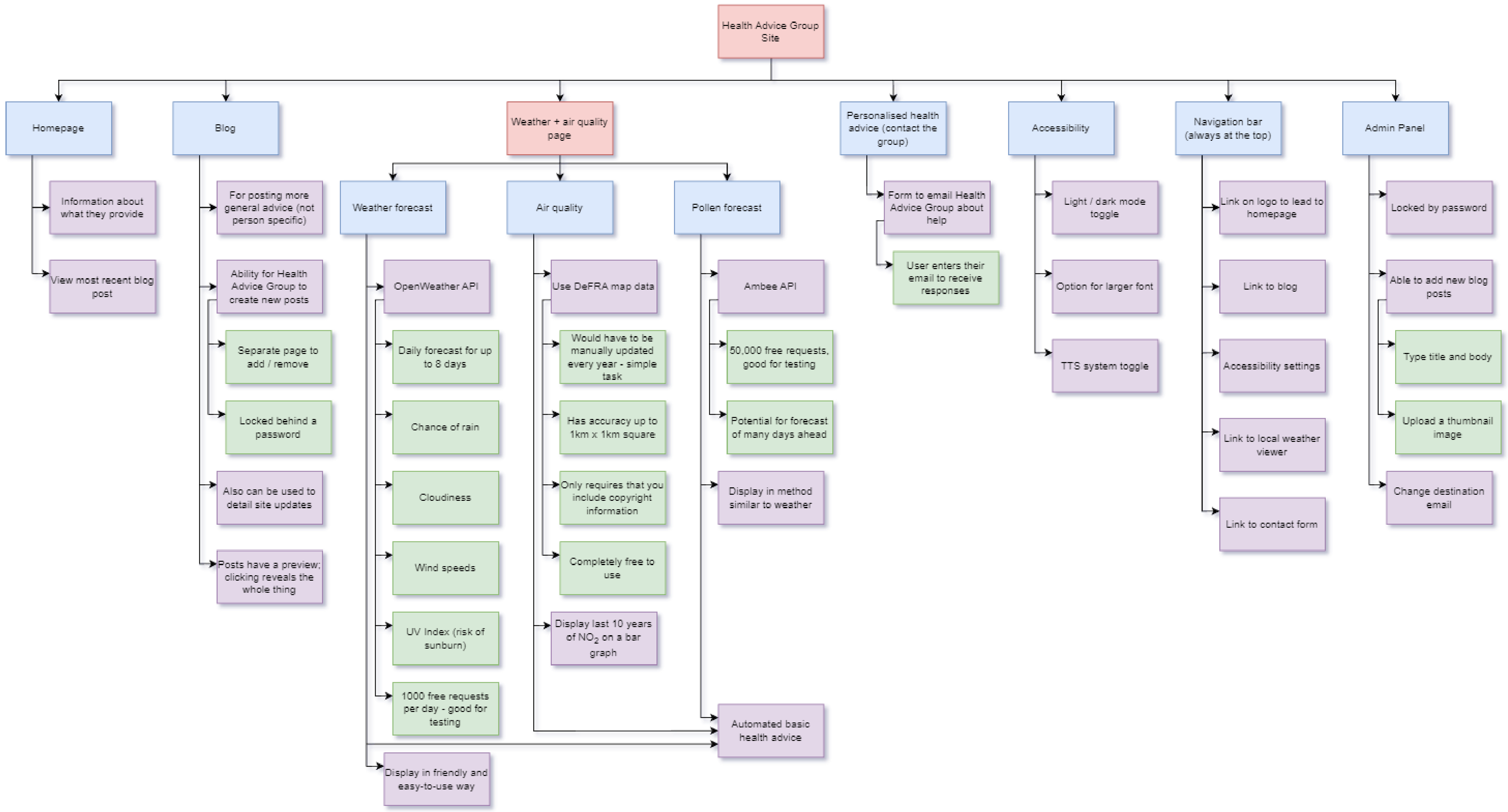
The navigation bar at the top of the site is important on any website for multiple reasons. Firstly, it is what the user first looks at. It is what they will think they can do. Secondly, the links it provides are vital for easy navigation across the site. Finally, anything that is important to keep on any page, such as settings and (though not applicable to this site) profiles. Here is the bar I have designed for this site:



Each main page of the website is shown in order of their usefulness, with the homepage first. The pages are shown like documents in a folder, with the white one being the page that is currently open. A gear is used (though of course it would have a transparent background normally) to open up settings that control accessibility features.

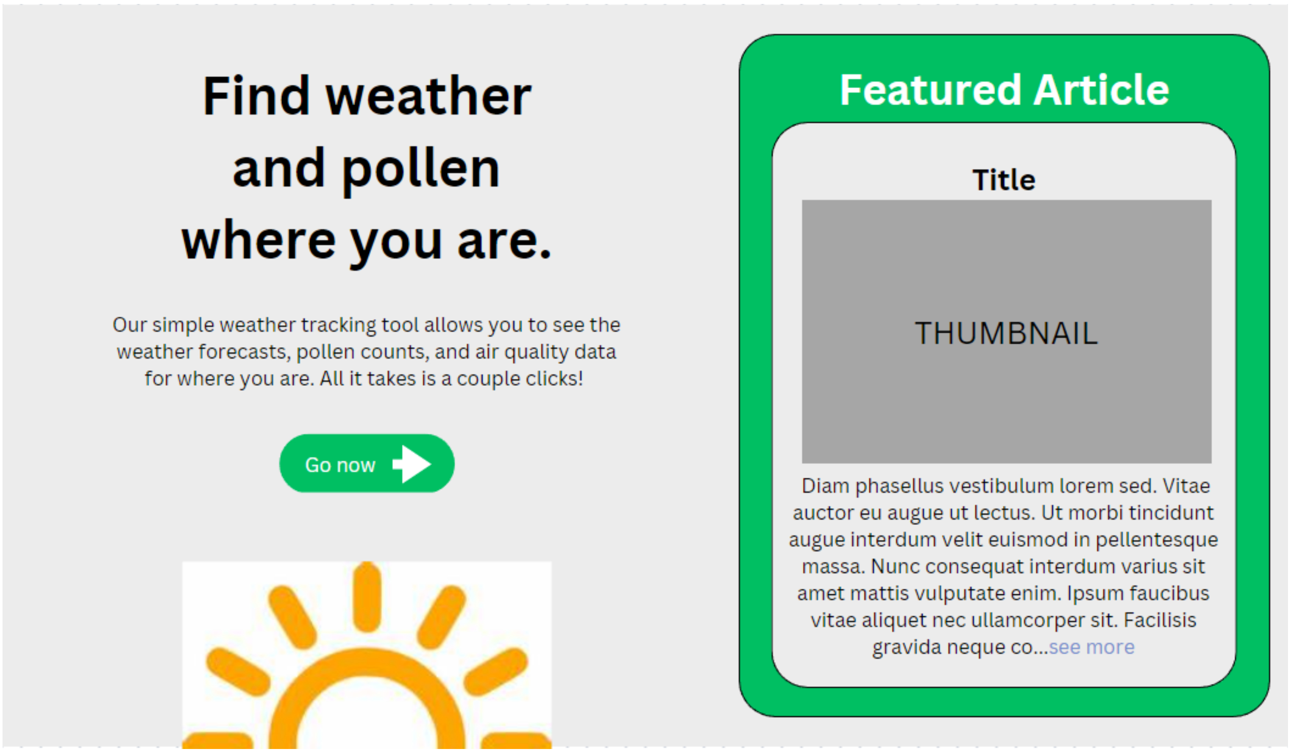
The image used here for the settings icon, as well as any images in other parts of the site, are not the images that will be used for the finished site, but are here to get an idea of what the site will look like. The actual images I will create myself to avoid any issues with copyright.

Complete site breakdown diagram on next page


Health Advice Group Site
Homepage
Information about what they provide
View most recent blog post
Blog
For posting more general advice (not person specific)
Ability for Health Advice Group to create new posts
Separate page to add / remove
Locked behind a password
Also can be used to detail site updates
Posts have a preview; clicking reveals the whole thing
Weather + air quality page
Weather forecast
OpenWeather API
Daily forecast for up to 8 days
Chance of rain
Cloudiness
Wind speeds
UV Index (risk of sunburn)
1000 free requests per day - good for testing
Display in friendly and easy-to-use way
Air quality
Use DeFRA map data
Would have to be manually updated every year - simple task
Has accuracy up to 1km x 1km square
Only requires that you include copyright information
Completely free to use
Display last 10 years of NO_2 on a bar graph
Pollen forecast
Ambee API
50,000 free requests, good for testing
Potential for forecast of many days ahead
Display in method similar to weather
Automated basic health advice
Personalised health advice (contact the group)
Form to email Health Advice Group about help
User enters their email to receive responses
Accessibility
Light / dark mode toggle
Option for larger font
TTS system toggle
Navigation bar (always at the top)
Link on logo to lead to homepage
Link to blog
Accessibility settings
Link to local weather viewer
Link to contact form
Admin Panel
Locked by password
Able to add new blog posts
Type title and body
Upload a thumbnail image
Change destination email

<u>Site Breakdown – Homepage</u>

The homepage of a website is the first page a user will normally see. As such, it needs to get the user's attention and summarise the purpose of the site. Here is my design for Health Advice Group's homepage:



Find weather and pollen where you are.

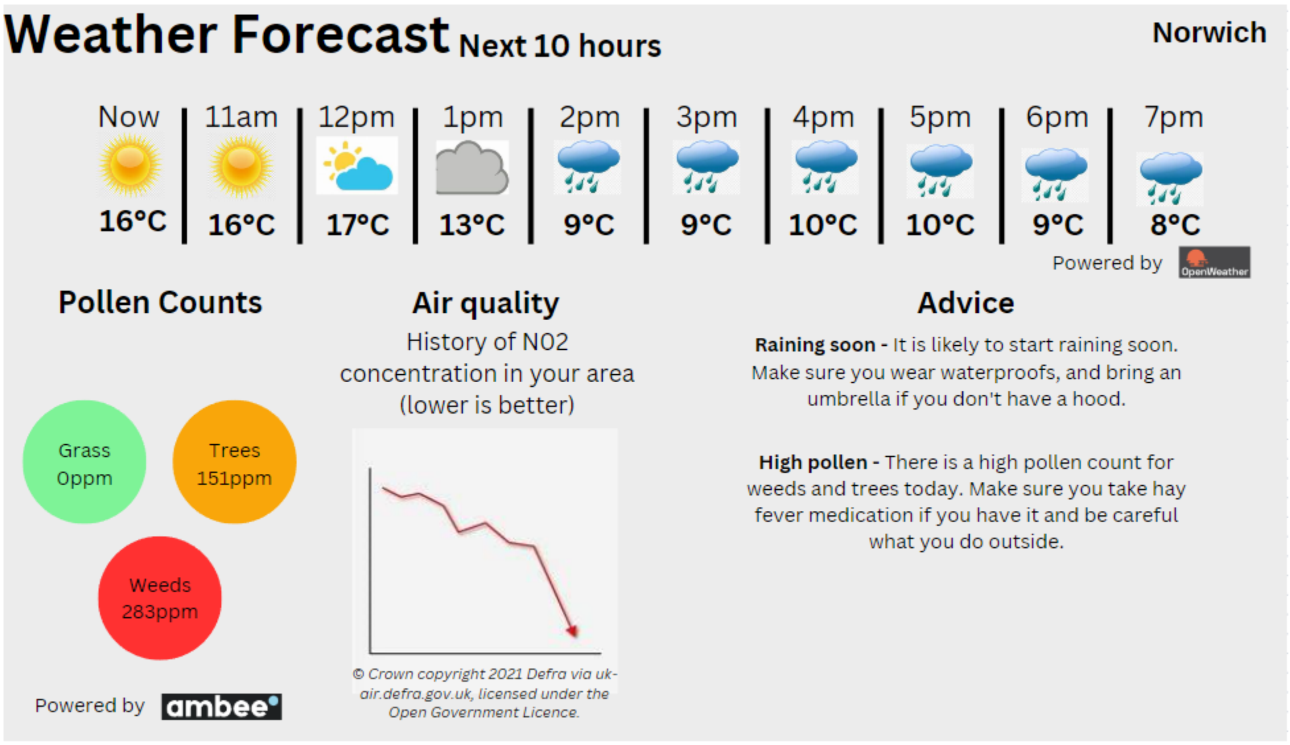
Our simple weather tracking tool allows you to see the weather forecasts, pollen counts, and air quality data for where you are. All it takes is a couple clicks!

Diam phasellus vestibulum lorem sed. Vitae auctor eu augue ut lectus. Ut morbi tincidunt augue interdum velit euismod in pellentesque massa. Nunc consequat interdum varius sit amet mattis vulputate enim. Ipsum faucibus vitae aliquet nec ullamcorper sit. Facilisis gravida neque co...see more

As you can see, on the left it clearly shows the main feature of the site, the weather tracker, as well as a button to take the user straight to it. While there is a link to the weather at the top in the navigation, this one the user is more likely to see. On the right is the featured article in the blog-like system the site will have, with a button at the bottom – the 'see more' text – that goes to the blog tab, where the featured post is at the top. To help break up the green, there is a rising sun graphic (ignore the non-transparent background) in the lower left corner. It helps to stop the user from being overwhelmed by the large amount of green and subtly points the user towards the weather service button.

<u>Site Breakdown – Weather</u>

The weather forecasting system is possibly the most important part of the site. It houses three types of data – weather forecasts that would be best displayed hourly on a table, pollen counts that are best as three wheels (or semicircles), and air quality history on a line graph. It also needs to include basic advice based on the results. My design is as follows:



Weather Forecast Next 10 hours

Norwich

Now	11am	12pm	1pm	2pm	3pm	4pm	5pm	6pm	7pm
16°C	16°C	17°C	13°C	9°C	9°C	10°C	10°C	9°C	8°C

Powered by OpenWeather

Pollen Counts

Grass 0ppm

Trees 151ppm

Weeds 283ppm

Powered by ambee

Air quality

History of N02 concentration in your area (lower is better)

© Crown copyright 2021 Defra via uk-air.defra.gov.uk, licensed under the Open Government Licence.

Advice

Raining soon - It is likely to start raining soon. Make sure you wear waterproofs, and bring an umbrella if you don't have a hood.

High pollen - There is a high pollen count for weeds and trees today. Make sure you take hay fever medication if you have it and be careful what you do outside.

The page is clearly split into four distinct sections, despite there being no lines between them. This is so that the lines are not a distraction for the user. In the top-right corner the user can see their location – this is mostly here as a confirmation to the user that their location is correct. The weather is displayed along the top, as it is the largest and arguably most important feature. Icons are used (that would normally keep a consistent style) to show what type of weather it is as the user is much more likely to understand at a quick glance. There was originally a plan to include wind speeds but it wouldn't fit without reducing the ease of use that Health Advice Group want from the site.

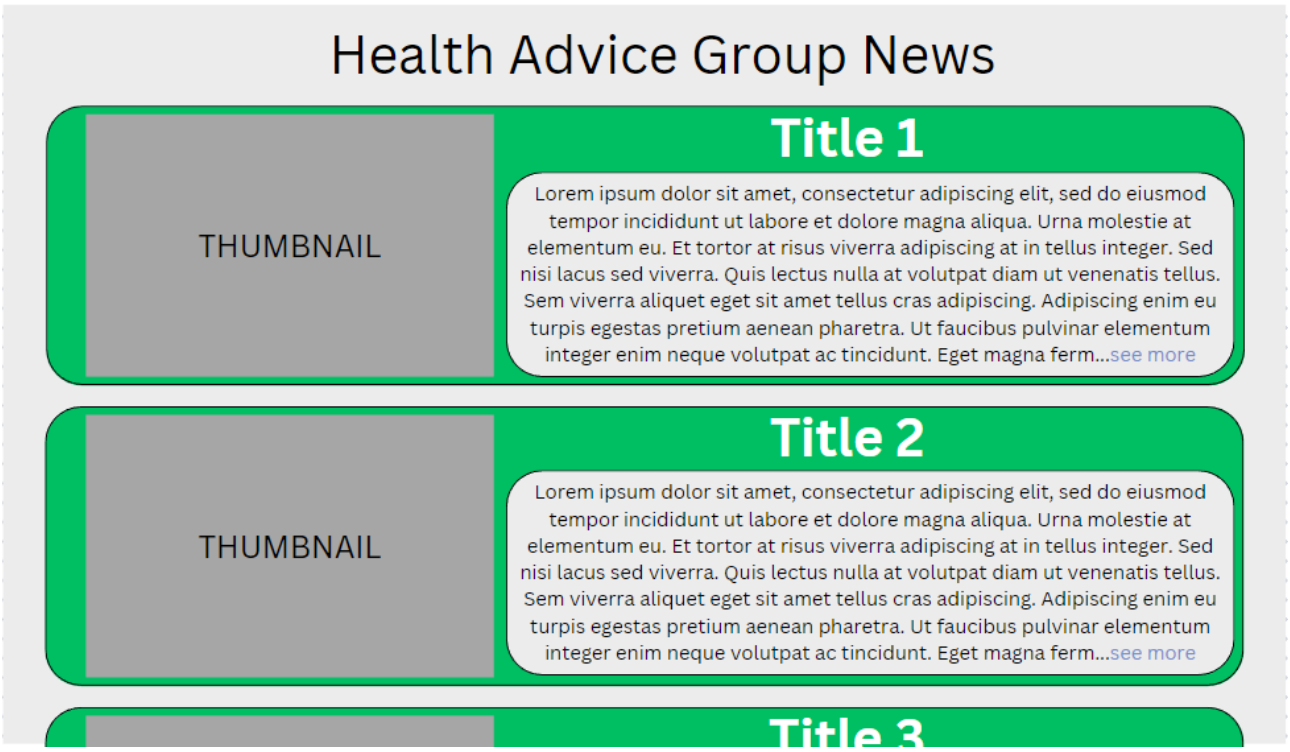
The pollen counts are displayed in the bottom left in three circles, one for each type of pollen. The colour of the circle indicates how severe the pollen is, the exact values needed for a colour change are different for each type of pollen and can be found on this website: https://www.kleenex.co.uk/pollen-count (scroll down on this site to find a table).

Air quality data is shown next to the pollen count, on a line graph. There is a title above the graph that both describes what the graph shows and reminds them that lower values are better (this may not be clear). The line graph pictured above is not what the actual graph would look like, simply a placeholder. The actual graph would have clear measurements on the y-axis and the last 5 years on the x-axis. Below is the copyright information needed in order to use this data.

Finally, the advice section in the lower left will take the data obtained for weather and pollen and generate some basic advice from it. Examples of this can be seen in the image above. It could also show advice based on data the user cannot see e.g. high winds or UV index.

Site Breakdown – Blog

The blog makes up Health Advice Group's way of spreading news and more generic advice to their users. Here is the blog page:


Health Advice Group News

Title 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Urna molestie at elementum eu. Et tortor at risus viverra adipiscing at in tellus integer. Sed nisi lacus sed viverra. Quis lectus nulla at volutpat diam ut venenatis tellus. Sem viverra aliquet eget sit amet tellus cras adipiscing. Adipiscing enim eu turpis egestas pretium aenean pharetra. Ut faucibus pulvinar elementum integer enim neque volutpat ac tincidunt. Eget magna ferm...see more

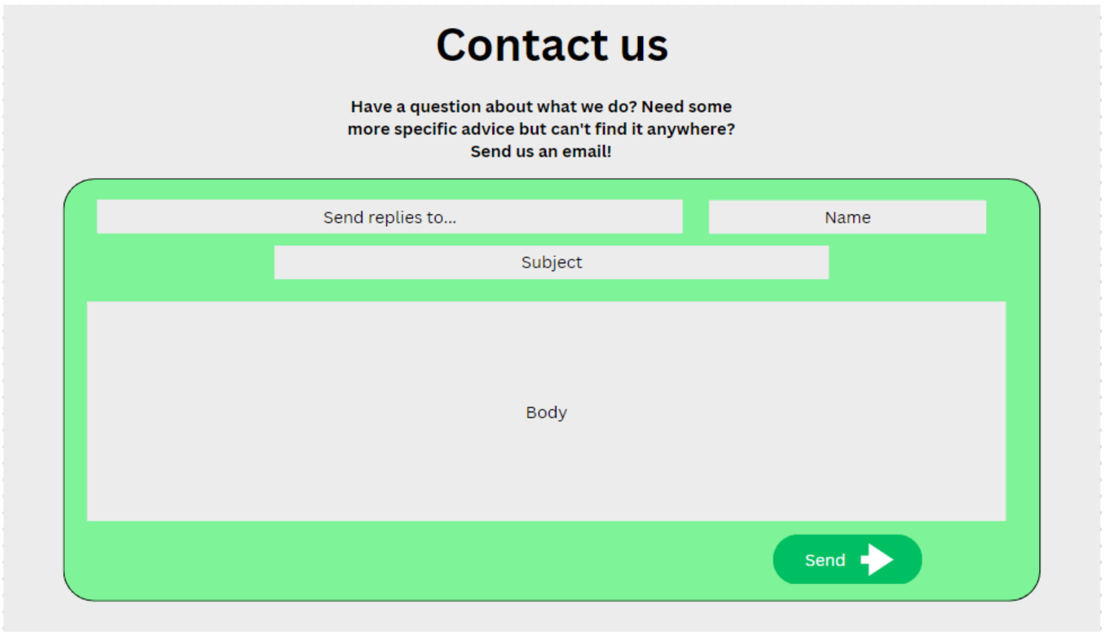
Title 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Urna molestie at elementum eu. Et tortor at risus viverra adipiscing at in tellus integer. Sed nisi lacus sed viverra. Quis lectus nulla at volutpat diam ut venenatis tellus. Sem viverra aliquet eget sit amet tellus cras adipiscing. Adipiscing enim eu turpis egestas pretium aenean pharetra. Ut faucibus pulvinar elementum integer enim neque volutpat ac tincidunt. Eget magna ferm...see more

Title 3

The posts would be organised newest to oldest, with two able to fit on screen at a time. A third article can be seen partially cut off below, so that the user knows they can scroll down on this page to see older posts. Clicking the 'see more' button at the end of each post will bring up the full post in a new page. On this page, there will be an option in the navigation settings to turn on a built-in screen reader, so that people who struggle to read the relatively small text can have the article read to them.

Site Breakdown – Contact


Contact us

Have a question about what we do? Need some more specific advice but can't find it anywhere? Send us an email!

The final user-facing page will be where users can send an email to Health Advice Group directly through the website. At the top is a simple description of what the page is for, as this may be the first time that the user is on this page. Below that is a form that allows the user to enter their name, an email to receive the reply, and a subject and body for the email itself. The send button is identical in design to the button on the homepage to keep consistency.

<u>**Algorithm design – Air quality data weighted mean**</u>

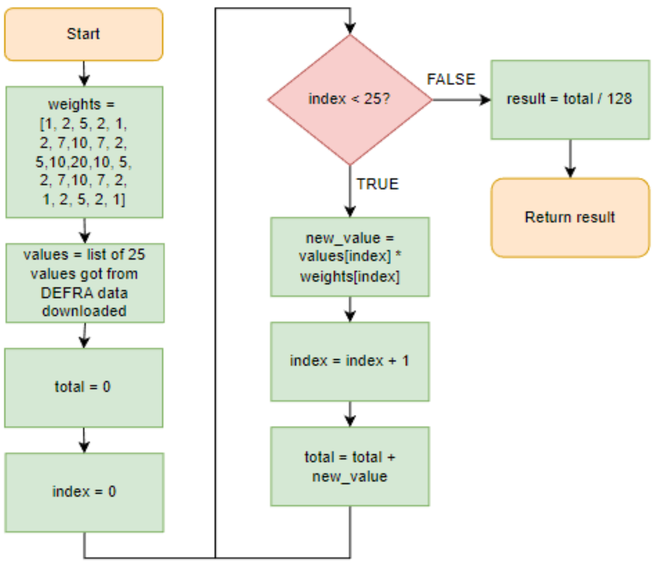
The air quality data provided by DEFRA is accurate up to a 1km by 1km square. While this allows for very precise readings, it means the values are not representative of the whole area where the user lives and spends their time. As such, it would be better to calculate a weighted mean that takes into account nearby squares in a 5km by 5km grid. Here are the weights that will be used:

1	2	5	2	1
2	7	10	7	2
5	10	20	10	5
2	7	10	7	2
1	2	5	2	1

Total: 128

This allows for a calculation that is representative of the surrounding area while still giving priority to the middle where the user actually is.

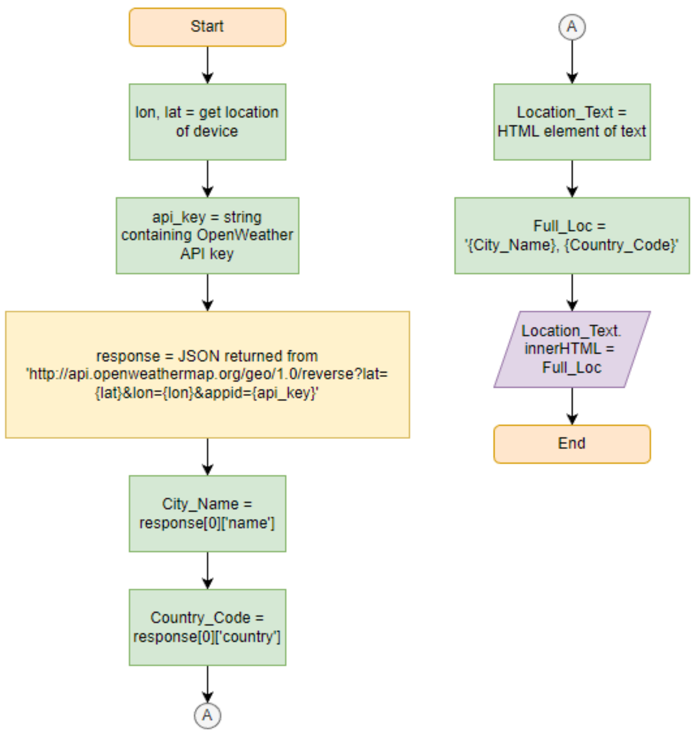
FLOWCHART


Start
weights = [1, 2, 5, 2, 1, 2, 7,10, 7, 2, 5,10,20,10, 5, 2, 7,10, 7, 2, 1, 2, 5, 2, 1]
values = list of 25 values got from DEFRA data downloaded
total = 0
index = 0
index < 25?
FALSE
TRUE
result = total / 128
Return result
new_value = values[index] * weights[index]
index = index + 1
total = total + new_value

<u>Algorithm design – OpenWeather API reverse geocoding</u>

OpenWeather's API system for weather uses exact longitude and latitude co-ordinates. However, we want to be able to display a town or city to the user so that they know the location is correct. OpenWeather provides a system for turning one into the other called Reverse Geocoding. Below is a flowchart for how a request could be made and added to the page.

FLOWCHART

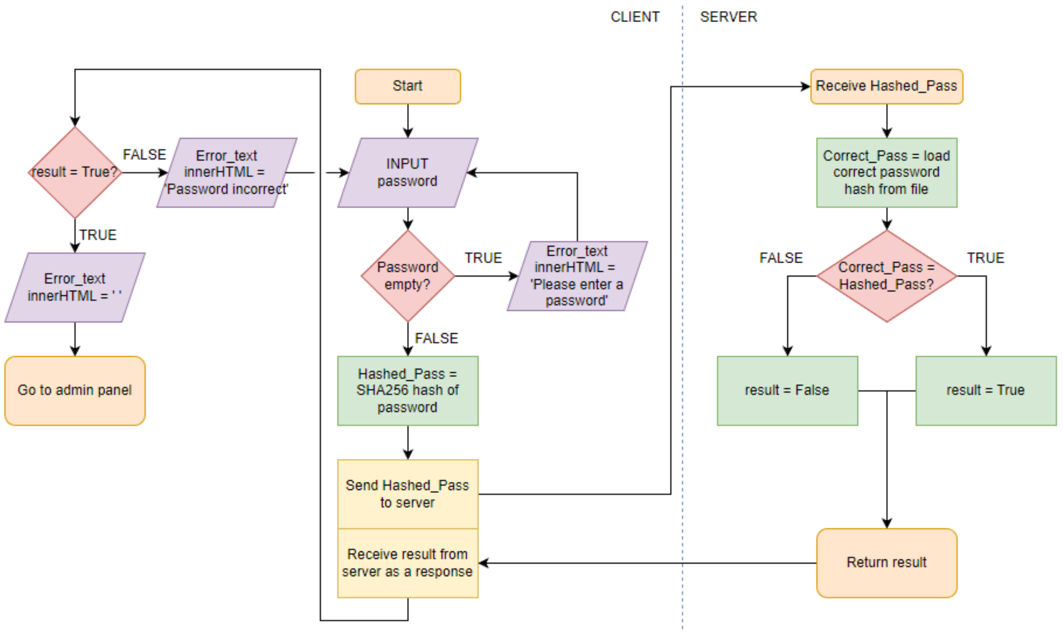


Start
lon, lat = get location of device
api_key = string containing OpenWeather API key
response = JSON returned from 'http://api.openweathermap.org/geo/1.0/reverse?lat={lat}&lon={lon}&appid={api_key}'
City_Name = response[0]['name']
Country_Code = response[0]['country']
A
A
Location_Text = HTML element of text
Full_Loc = '{City_Name}, {Country_Code}'
Location_Text.innerHTML = Full_Loc
End

<u>Algorithm design – Password Verification</u>

Doing anything with the administrator panel is locked behind a password to prevent unauthorised access to the system for creating blogs. In order for this password to remain secure, it should be hashed on the client-side, sent across, and then compared to a hash stored server-side.

FLOWCHART



CLIENT
SERVER
Start
INPUT password
Password empty?
TRUE
Error_text innerHTML = 'Please enter a password'
FALSE
Hashed_Pass = SHA256 hash of password
Send Hashed_Pass to server
Receive result from server as a response
result = True?
FALSE
Error_text innerHTML = 'Password incorrect'
TRUE
Error_text innerHTML = ' '
Go to admin panel
Receive Hashed_Pass
Correct_Pass = load correct password hash from file
Correct_Pass = Hashed_Pass?
FALSE
TRUE
result = False
result = True
Return result

Algorithm design – API structure

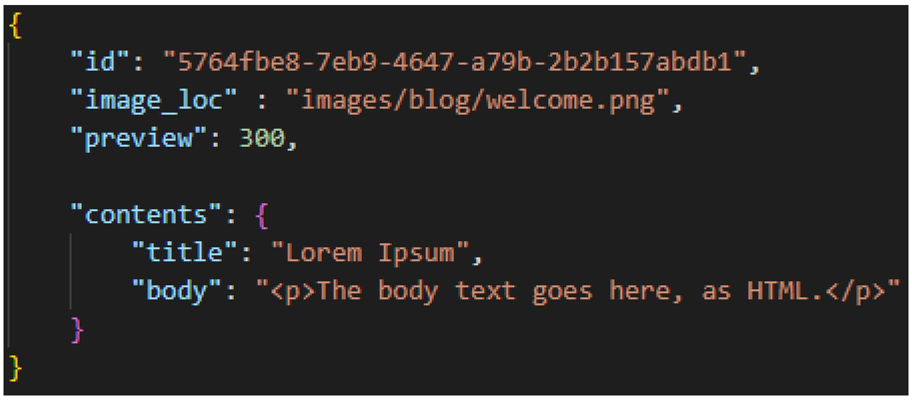
The website itself is an API built in Flask that returns webpages and JSON content. This needs to be structured so that all needed data can be accessed by anything that needs it. Here is the structure for this API:

Path	HTTP method	Body	Response contents
/ (none)	GET	N/A	Return the homepage
/lastblog	GET	N/A	Return the last blog post (for the homepage)
/blog	GET	N/A	Return the blog page
/blog/all	GET	N/A	Return all blog posts in reverse order (for the blog page)
/blog/view/<id>	GET	N/A	Return the page to view the post with id <id> on its own – this is user-facing so not in the body
/weather	GET	N/A	Return the weather page
/weather/content	GET	Encrypted Longitude, Encrypted Latitude, Local Time	Return all values for all parts of the weather forecast, pollen, air quality, and advice systems
/encryptkey	GET	N/A	Return the key for encrypting the longitude / latitude
/contact	GET	N/A	Return the contact page
/contact/send	POST	Email contents	Send an email to the address saved on the server with specified contents
/admin	GET	N/A	Return the admin panel (not accessible unless the link is manually typed in)
/admin/create	POST	Blog contents, password hash	Add a blog onto the server to be viewed on other pages. Throw error 401 unauthorised if password is wrong. ID is generated server-side.
/admin/password	POST	New password hash, current password hash	Update the stored password hash. Throw error 401 unauthorised if password is wrong.
/admin/email	POST	New email destination, password hash	Update the destination email for the contact form. Throw error 401 unauthorised if password is wrong.
/imagefor	GET	Blog ID	Get the thumbnail associated with a certain blog ID.
/images	GET	N/A	Get all images in /images/blog, so the admin user can choose a thumbnail.
/saveimage	POST	Image, name, password hash	Add an image to /images/blog to be used as a thumbnail. Throw error 401 unauthorised if password is wrong.
/title-id	GET	N/A	Get just the title and id of every blog so the admin can choose to save over them.

Accessibility features like dark mode and screen reader are done entirely client-side, so do not need any part of the above table.

Data requirements – Blog structure

As mentioned above, the blog posts will need to be sent to the server, stored server-side, and sent back to the client. This means it will need to be in a JSON format, so it is standardised. Here is the structure of a blog:



```
{
    "id": "5764fbe8-7eb9-4647-a79b-2b2b157abdb1",
    "image_loc" : "images/blog/welcome.png",
    "preview": 300,

    "contents": {
        "title": "Lorem Ipsum",
        "body": "<p>The body text goes here, as HTML.</p>"
    }
}
```

The first field is the blog id, which will be generated using the UUID4 algorithm. This is also the name of the file, and is used to identify it in most systems.

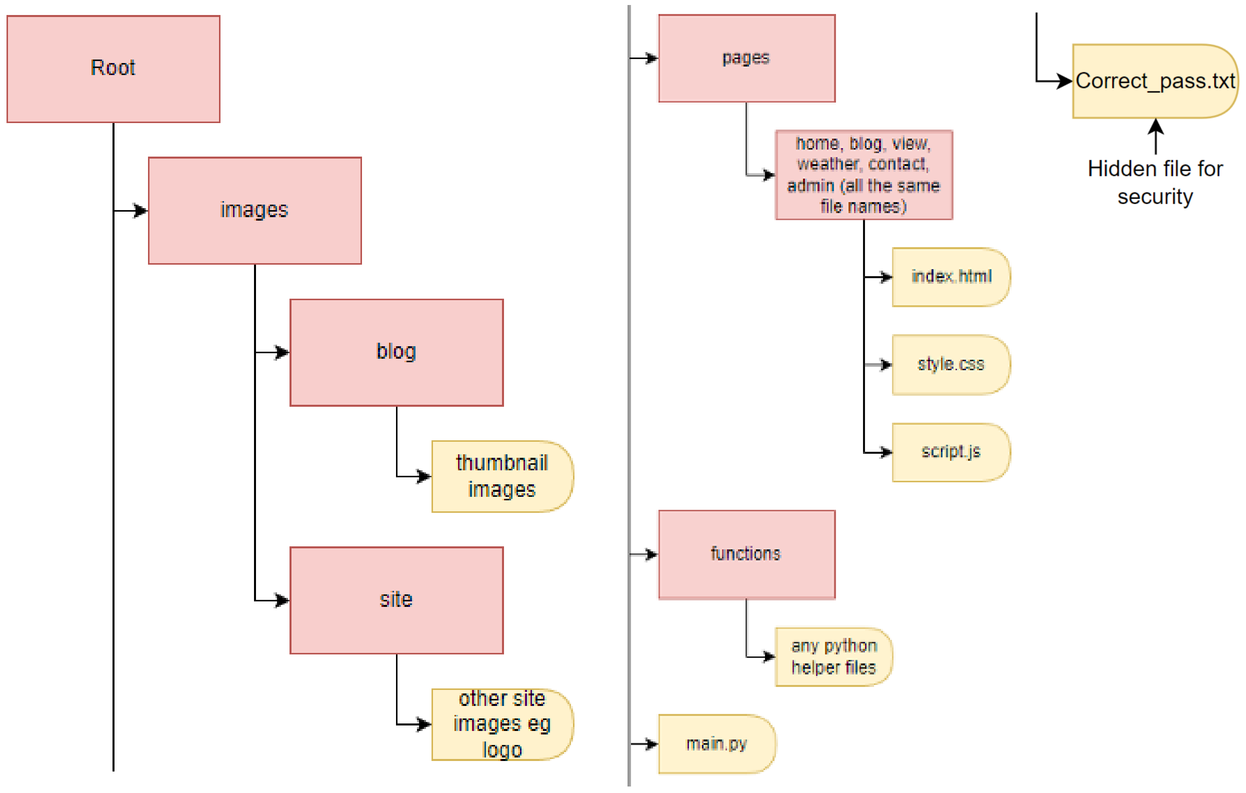
The next key, 'image_loc', is the file path on the server to the thumbnail image.

Preview represents the number of characters before the '...see more' link appears on the blog page and homepage.

The last key, 'contents', contains a dictionary that has its own key-value pairs. 'Title' is the title of the blog, and 'body' is the main body text of the blog. It was chosen to have this part as raw HTML so that blogs can be created with bullet points, other images, and even custom CSS and JavaScript.

Data requirements – Folder structure

This is how the file structure will look on the server:



Test strategy

May not include all parts of the solution. Dates may be inaccurate.

Date of test	Component to be tested	Type of test to be carried out	Prerequisites and dependencies
09/03/23	Website structure	Test that webpages are returned correctly when links are followed	Requires a basic Flask frontend
15/03/23	Weather - Forecast	Test that all parts of the weather forecasting system work as a group (integration testing)	Requires weather page to be prepared for the forecast Requires OpenWeather API subscription
16/03/23	Weather - Pollen	Test as above for the pollen count system, and using forced data test the colours	Requires Ambee API subscription Requires weather page to be ready
16/03/23	Weather – Air quality	As above, for air quality line graph. Use a standard table if the line graph proves impossible.	Requires the weather page to be ready for the data Requires DEFRA data to be downloaded
16/03/23	Weather - Advice	Advice should be created as the other parts are made, test that it displays properly and only with the correct conditions	Requires the advice section of the weather page Requires advice generation
22/03/23	Blog – Create on admin	Check that the admin panel can create blogs	Admin panel Password lock
23/03/23	Blog – view on blog page	Test the blog page to see if blogs are displayed correctly and in the correct order	Ability to create blogs Blog page
23/03/23	Blog – view entire post	Test to show that clicking on a blog will let you view the whole thing	Blog page Individual blog page
29/03/23	Contact – send emails	Check the system that automatically sends emails based on the contents of the form	Contact page Email-sending system Email to send from Admin can change destination
29/03/23	Homepage – Featured post	Make sure that the most recent blog post is displayed on the homepage	Working homepage Working blog system
30/03/23	Complete system test	Final check through all systems	All of the above