Health Advice Group Documentation Log

Development Notes

The task brief outlines the client requirements that must be met by the digital solution that I will be designing. This solution should allow its users to access:

- Weather forecasting
- Air quality monitoring dashboard
- Advice on how to deal with health matters affected by weather and environmental conditions.

Some of the features available on the market are:

- Personalised health advice based on location
- Accessibility features
- A personal health tracking tool.

Since the proposed solution I have come up with an idea of how the Health Advice Group website should look like on the final iteration of the prototyping. The index page of the website will be the home page with promotional material to attract more traffic to it. The page will describe the service provided by Health Advice Group, having two to three promotional photos, and paragraphs. This will lead the user to a signup section, that will prompt them to create an account, to join the service.

The personal health tracking page could include points such as: body weight, diet type and sleeping schedules. This data can be used to provide personal health advice, for example a different diet through proposed recipes and fluid intakes. A health tracking tool like this in combination with the weather forecasting and air quality data, will help to make a safer and improved quality health decisions. Health Advice Group can have many uses for its digital solution, as customers will have more options, meaning they won't have to go to a competitor's website. This will also be convenient and easy for users to personalise their health.

The accessibility popup will be available for both signed in users and visitors. This will require the use of cookies so that the modified accessibility settings can be retrieved on a return to the website. Whether the user agrees to the use of cookies, that is essential. I believe that this should be an automatic agreement when the user visits the website, since it will help them to navigate and operate it.

Weather forecast monitoring will use a visual weather API. This API provides all weather forecast data, and air quality index data, which is required for the personalised dashboard.

I have added a donation box on the footer of the website. This means that it is not an essential element; the user does not view this box straight away when they visit the website therefore they are not forced to do so. The donation box uses PayPal which Health Advice Group can set up in the future iterations. However, this form of donation is safer than using an API for the user to fill out their details, as in case of a malicious code being inserted into Health Advice Group codebase, we

would be liable for any loss for the user. The donation box instead redirects the user to the PayPal donate website, which is secured and encrypted.

Health advice will be supplied by an accredited health organisation: NHS England. The health advice from the NHS will be based on the user's sex, and health problems. I am using content from the NHS as they are certified healthcare providers therefore, users can trust the advice that is shared on the website. https://developer.api.nhs.uk/nhs-api/documentation/video In general, every user that visits the website will have a NHS news feed available, for those that have not signed up. However a signed up user can access a dashboard for their health advice which will include articles, environmental change such as climate change data and informational videos. The videos taken from the NHS website are in the form of a YouTube video, therefore they will by default include subtitles and navigation settings, to help with accessibility.

The NHS logo must be visible on all pages that use information provided by them. This is to ensure that Health Advice Group website follows standard licensing terms, and does not take this content as their own.

Certain content that I will retrieve from this API will be saved in a data store so that this website does not overwhelm the NHS digital, and so that the limit per month is not used up.

For health alerts a push notification feature should be added. In case a user has forgotten about the Health Advice Group service, a push notification can be sent out with the daily NHS news. This ensures that users attend the Health Advice Group website regularly so that we do not loose them. To make this push notification system be easier to implement, there will be no user credentials used, however, because the user's browser will most likely cache the website, this can send notifications in the background.

The environmental conditions page will include both written and API generated content. The environmental conditions must be updated, however some advice will not change, because it is a standard advice based on that environment.

Additionally, instead of making a push notification system, the user can receive these daily news through their email, since some of the web browsers do not support this feature. This will also allow Health Advice Group to add some personalisation since they can look at useful alerts to add into the message body.

These emails will be sent by using the local mercury smtp server. This is not ideal for larger projects, however for a charity like Health Advice Group, and as a prototype, this server can be used to send emails to users. The php server script should be running non-stop on the tomcat server, so that users receive emails at any time of the day, even if the staff are not out of work hours.

The mercury server is much less secure than a third party smtp server. Your credentials must be saved in the configuration files for the google smtp server to authorise sending of emails. The emails must be sent from an actual email domain, such as google. However, saving the login credentials in the configuration file is a risk for the security of your google account. Therefore, for this function I will create a google account that does not link to me. The mercury server required configuration from the php, and sendmail scripts, so that emails are sent from my email and using the google smtp server, which is free. Most importantly, the email will not appear as spam, since it has been sent from an official google email account, however some email api's will be seen as unsecure, and therefore automatically placed in the spam folder.

This functionality is important as it will be used for two factor authentication for the website, meaning that if a user signs in from a different device, they have the ability to make their account secure, making sure that only they can log in, and access their personal information.

The port used by mercury: 25, is an unsecure web port, meaning that mail traffic could possibly be seen by other users of the network. Therefore, account verification could possibly become breached and personal information misused.

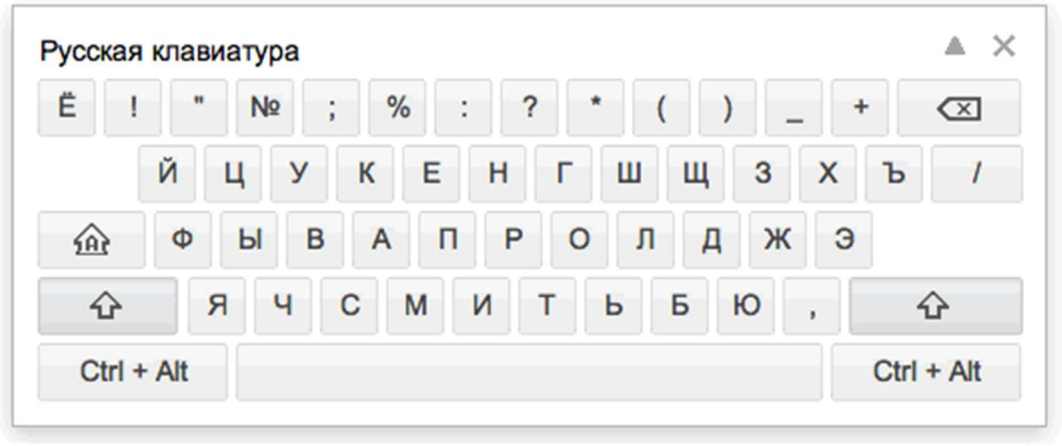
For testing of the mail activation function I am using a 10 minute email service, that provides a free mail box to receive the activation email from the Health Advice Group website. This function verifies the email provided by the user and prevents them from creating an account for someone else. The registration process assigns a token to the user record and places a 0 against their status. This 0 means that the user has not yet activated their account and will not be able to access their dashboard.

Administrators can view user records under their account logo. They will have a link to access a administration page, to modify user information if a user makes a request to do so. Part of this administration page will include web traffic and error logs, so the maintenance team can easily find what went wrong. This page will list the number of requests made to the local server over the last hour, however the administrator can also change the timestamp of the log file; average requests per minute could also be shown, and the error logs so that the maintenance team can tell if there is anything that needs fixing.

Most websites include some form of customer support service. This can be a form, social media messaging, or a live chat. For the Health Advice Group website, I have planned to implement both a form and a live chat. Because Health Advice Group may be short-staffed, the chatbot will be available for the most urgent questions that the user may have. For example, questions about a rash on their hands, which could be caused by a sunlight allergy. The answers given will be short and to the point on what could be the cause of that rash, however in the future versions of the solution, this could be replaced with a live chat with the staff. The chat bot will be powered by the AWS Lex service which is based on the Amazon Alexa, and that device tends to give reliable answers to the questions that a user may have. For additional accessibility, the chatbot could enable the user to use voice to text, and the user does not have to use an on screen keyboard if they cannot type.

With a chatbot, users do not have to fill out the forms to get a short answer and it is always there for a user to utilise. This can help boost user experience.

An onscreen keyboard is important for accessibility. This keyboard allows a user to use pointing devices such as a mouse, for those that are physically challenged while using a computer it serves an alternative option to a physical keyboard. This should be something that a user can drag out of the way from the other elements, for this reason, I will be using the Google virtual keyboard. It has been depreciated, however it should still work fine due to their policy.



A user can select the language they want to type in and it has all the necessary buttons that a 40% keyboard includes. This will be available throughout the website as a switch button on the navigation bar. The user might not realise this is available if it is hidden in the accessibility settings menu.

The local server will require an application that runs in the background for example a python script so that emails can be sent automatically based on the time and the database. The script will also make the weather warnings up to date by requesting the weather warnings for that day and replacing them in the database.

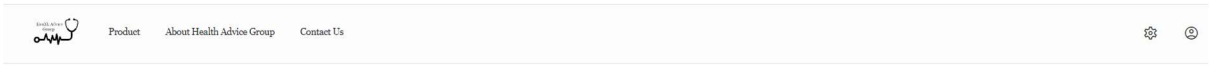
Using a gauge chart on the AQI data page will increase the ease of understanding for the end user, and using colour coding for values 0-100, 101-200, 201-300, 301-500+ is intuitive and immediately interpretable by the reader.

An account with two factor authentication can be used to send emails. For this a Health Advice Group Google account should be created with a name such as – hag-do-not-reply@gmail.com which can be used to send verification emails, and news since these not need a reply from the users, however we can keep a history of a week in case there are any important messages. hag-customer-support@gmail.com can be another email. These are easy to remember and having two separate inboxes allows Health Advice Group to track and filter different categories such as users subscribed to news and so on.

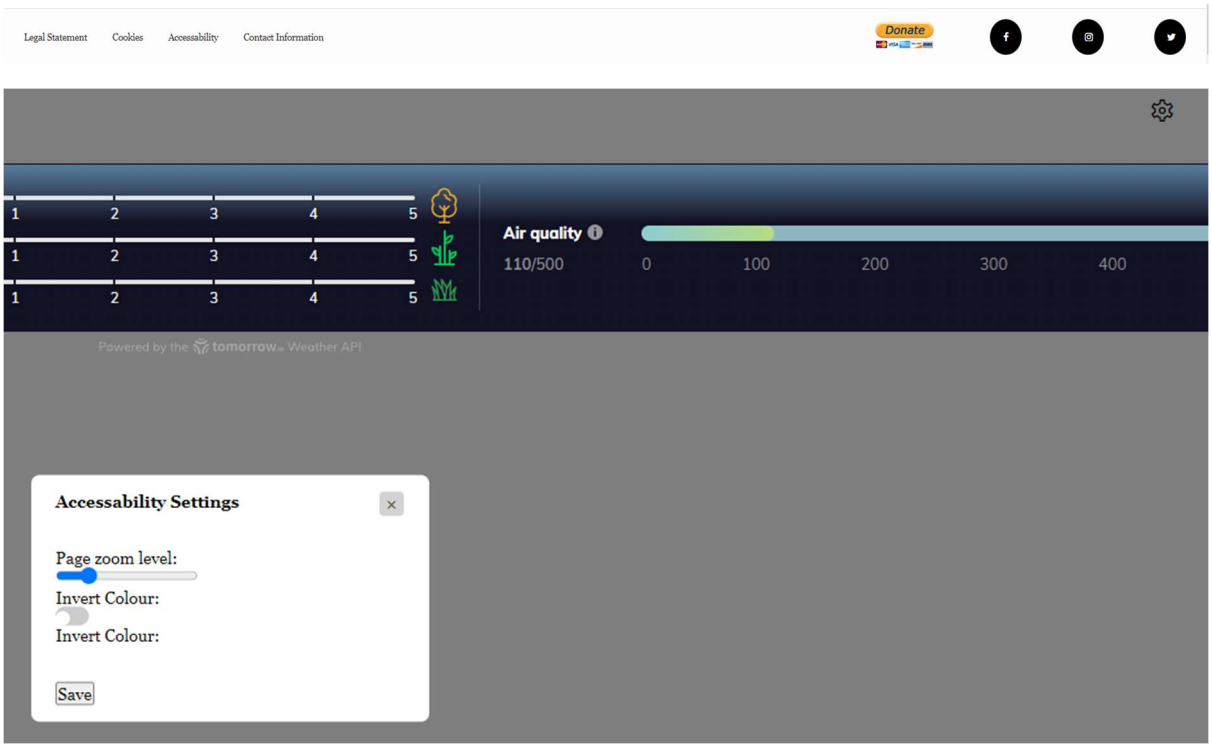
Development Log

Version 1 - 06/03/23 – 14/03/23

- Designing the website template



For the first few sessions I have started developing and researching different ways that the Health Advice Group solution could be implemented. I have started by making the layout of the pages that I have designed previously, having the navigation bar at the top and the footer.

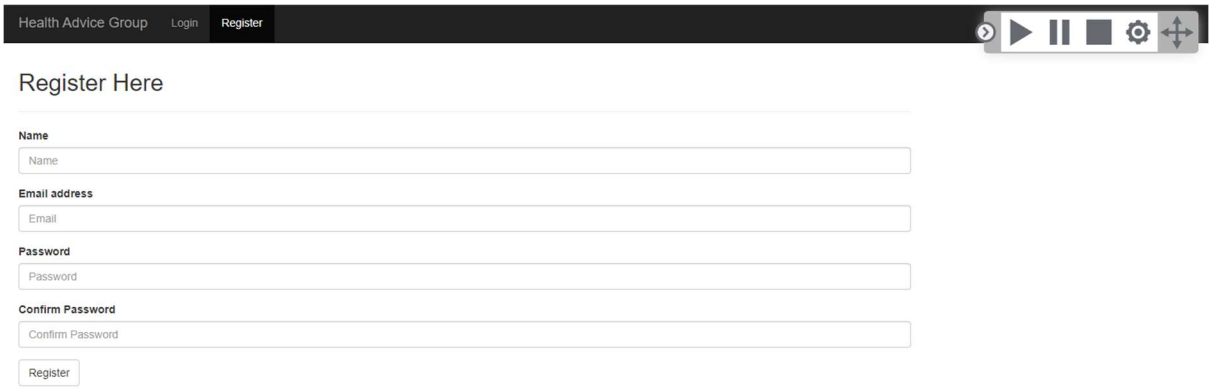


I have implemented some of the accessibility of the Health Advice Group. The page zoom that changes the zoom in level of the page for a better view of the elements.

Invert colour function that helps those with colour blindness to distinguish between the colours of the page.

Version 2 – 15/03/23 – 21/03/23

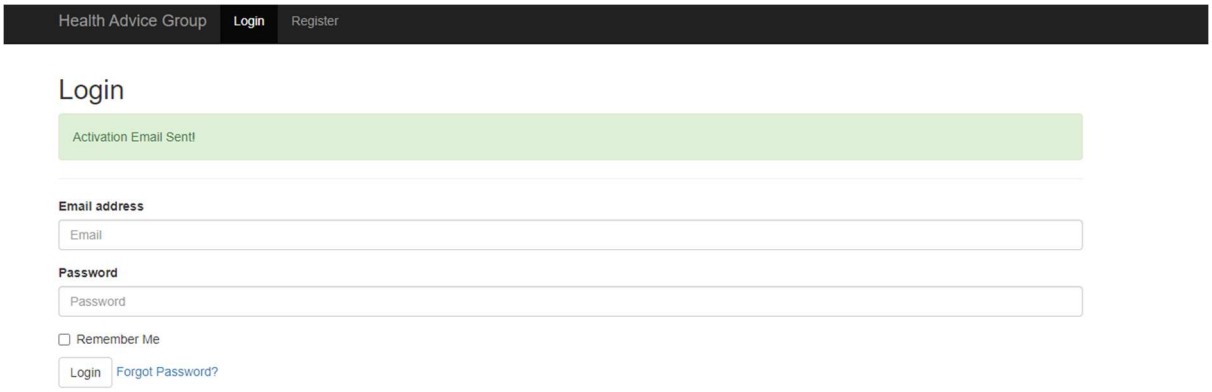
I have decided to go for a simpler layout for the testing of the login, and registration, which has limited viewing on the index page.



The pages include a third party widget that will read aloud the content of the page. This accessibility is important as not all users will be able to see the small writing on the website.

The Health Advice Website contains a floating text to speech widget which allows those with visual impairments or dyslexia to read the contents of the website without having to download any extensions to their web browser. The widget is open by default; therefore, I should put it somewhere where it will not obstruct any other elements.

The widget reads the whole page; however, a user can highlight the specific text they want to be read out and the API widget will continue from that point.



The registration form sends an email to the user's email address that they have entered and will add a token into the database. The email is a link to the account verification that gets the token from the users email and checks it against the database, as seen below.

I have tested the account activation, by visiting the URL that the user would receive in their inbox, however the mail function does not yet work, as the sending account would have to lessen their security settings in order to send automatic emails. But if the mail would have been sent, the token is verified against the database and, the status is changed to 1. The user can now log into their dashboard.

When the user logs in, a session cookie is initiated. This is not yet encrypted, however to prevent sniffers from knowing session cookie values and creating malicious code on the server, this session cookie should be encrypted to provide an additional level of protection. Since a localhost server is not secure an SSL certificate should be installed, to make the connection HTTPS.

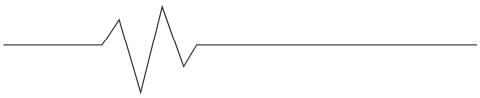
For testing purposes, I will generate my own SSL certificate so that functions that make unsecure connections to external sources will be blocked, which will force me to implement a more secure solution. This requires me to configure the Apache server and modifying the certificate file.

If this website is deployed, a free lets encrypt SSL certificate can be generated for the web server. Due to restrictions, I am not able to self-generate a SSL certificate and therefore will use a http connection, however, I will implement the cookie encryption, and make sure that any requests made are secure. https://stackoverflow.com/questions/64800565/how-to-create-valid-ssl-in-localhost-for-xampp

```php
<?php

include('includes/config.php');
include('includes/db.php');

if(isset($_GET['token'])){
    $token = $_GET['token'];
    $query = "update users set status='1' where token='$token'";
    if($db->query($query)){
        header("Location:index.php?success=Account Activated!!");
        exit();
    }

}

?>
```

```php
<?php

session_start();

session_destroy();

setcookie("user_email" , "" , time()-60*5);
header("Location:index.php?success=" . urlencode("Logged Out Successfully!"));
exit();

?>
```

Version 3 – 22/03/23 – 28/03/23



I have created a loading screen for the website in the shape of a heartbeat. This is for the user to wait while the main page content will load.

At the moment Health Advice Group website runs on the Apache server, if a user runs into an error that has not been identified, then it will display error on the server or page not found error. In that case we should put some contingency in so that the user is redirected to a 404 page or the error that the server or user is experiencing so that it is a more user-friendly website to the visitors even in the

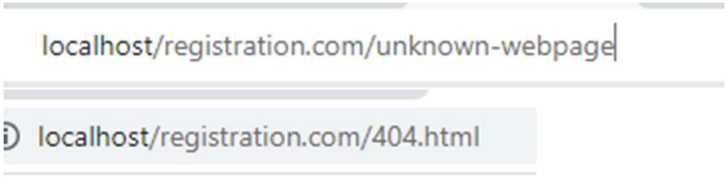
midst of an error. A custom 404 page is a good opportunity to keep the user on the Health Advice Group website and do more to redirect them to their destination.

.HTACCESS file does not work. ErrorDocument command does not redirect the user to 404.html when they get the Not Found error.

- However, because Health Advice Group is about health advice and weather, and any health based on the weather, I decided to make the error page as though it was sick, and thus the reason why it cannot load. This allows the user to visually understand that the website is not well and return to the last page.
- I have fixed the ErrorDocument redirect. The issue was that I have not included the RewriteEngine on, and the Apache server did not have permission to change the URL that the user tried to access.

```
RewriteEngine on
ErrorDocument 404 http://localhost/registration.com/404.html
```

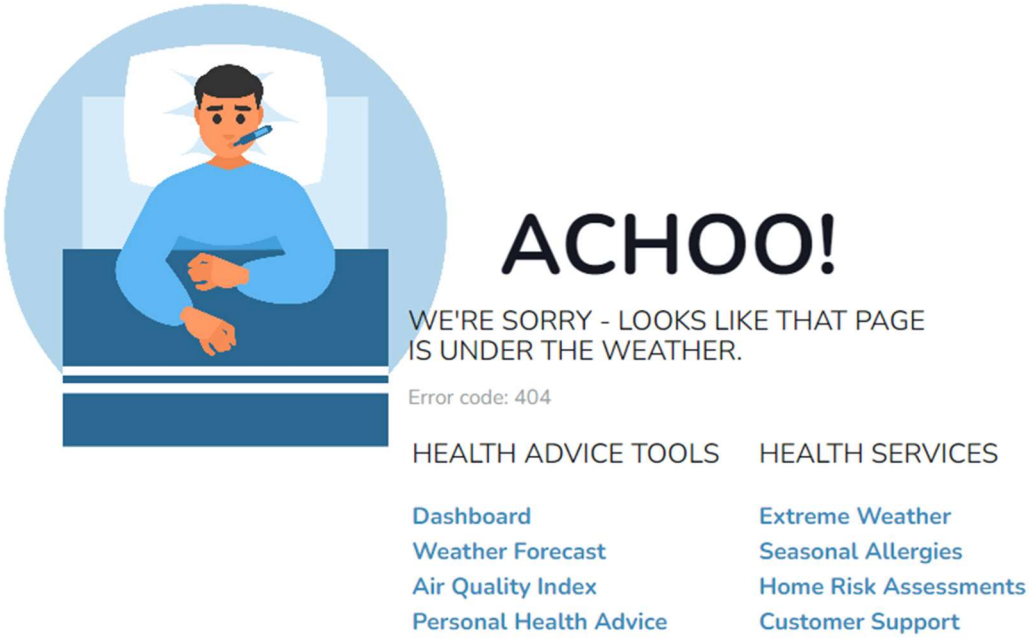
-
- Now if the user tries to access a site root that does not exist, they will be redirected to the custom 404 page.



-
- This ensures that the error is explained to the user in a straightforward manner, as the Apache server is a generic, page not found error, a user might get confused, especially with the white background and black text which does not match the colour scheme of Health Advice Group.

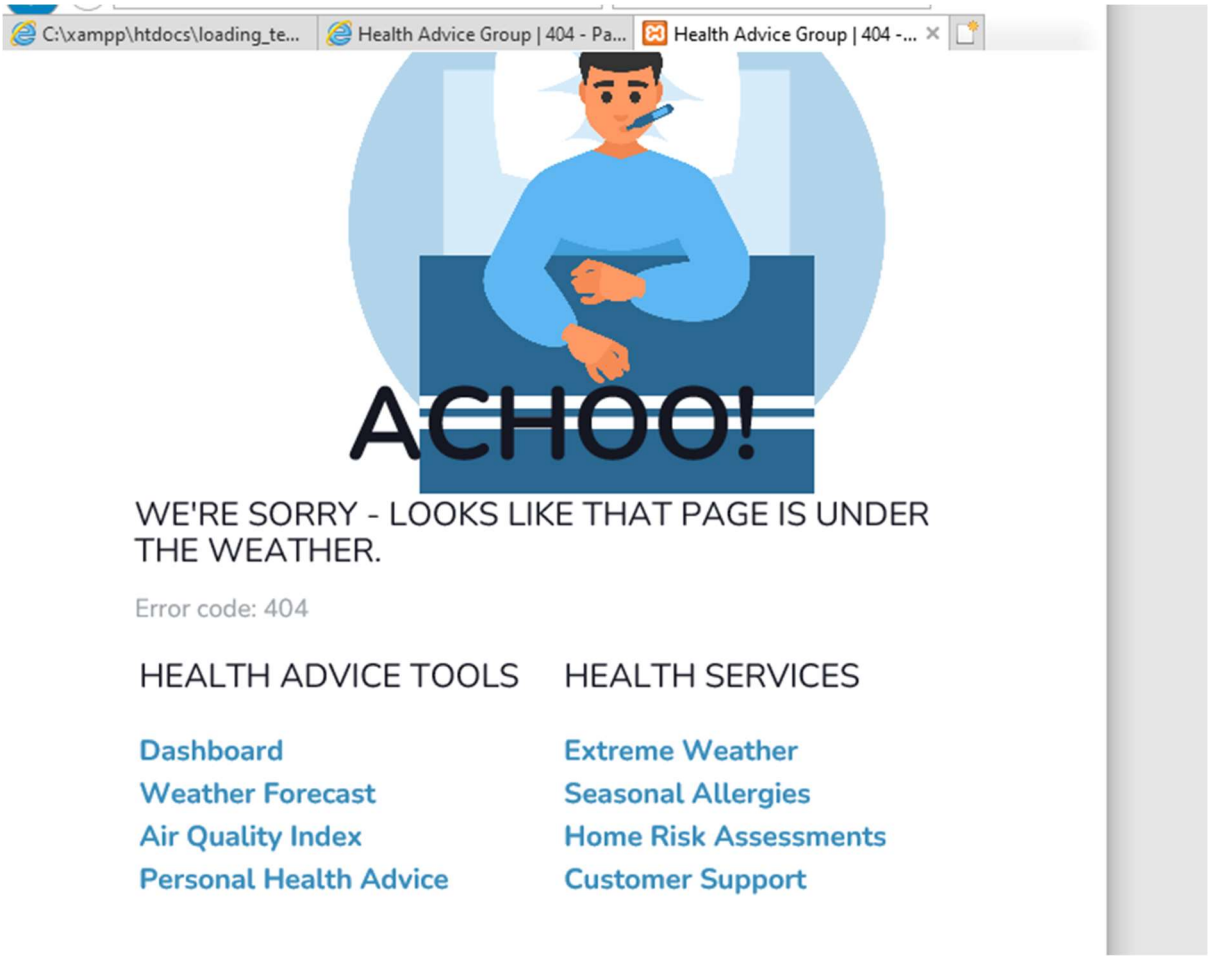
A loader is a nice addition to the Health Advice Group page, it improves user experience, and tells them that the page is working. The loader ensures that the static parts of the page are loaded before the user can see them. This loader should only last 3 seconds. For the loader I have decided a simple heartbeat loader, as it follows the health theme and does not require much data or power to display.

– Made a functional 404 page that provides links to all Health Advice Group services.



Humour is an important aspect of life. It also improves mental wellbeing and increases tolerance to pain. As a user they might be confused seeing this page, however the humour incorporated within the error – page not found – page, helps the user understand what the problem is. Also this page includes navigational links around the website, so that the user knows what they can access, if they have typed the wrong URL into the search bar, the custom 404 page will direct them onto the right direction.

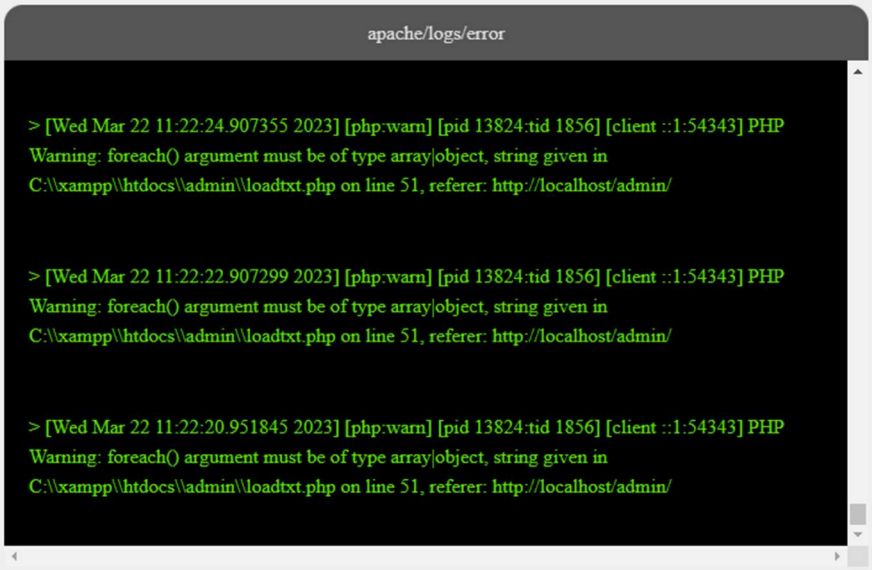
The image is responsive on most resolutions, however on some smartphones it will cut slightly and out of the page.



Internet Explorer has lost its support in 2022. Additionally, internet explorer comprises only about 29 million users. This number is still on a decline to date. It is true that people still use Internet explorer for web applications such as those for their cameras which require less secure features so that camera image can be browsed, however, internet explorer does not support the majority of modern web technologies. Because of this, Health Advice Group should not support Internet Explorer as a means of accessing the service.

Internet Explorer will automatically redirect to Edge, however if the user does not have edge on their machine, then I will attempt to make a support page that explains to the user that their browser does not support Health Advice Group's website.

For maintenance staff should be able to easily find the problem with the website, therefore an admin page containing Apache error and access logs will be a great way for them to maintain the site.



A staff member can view the error logs for Apache so that they can see on which pages there is an internal error that needs to be resolved.

In later versions, Health Advice Group should assess their ticketing strategy, as using a mailing service is inefficient and may cause problems when staff are trying to access previous feedback tickets to resolve an issue that has occurred in the past. Health Advice Group can use a system such as Zendesk to store tickets, and their state on the cloud. Reference numbers are also attached which can help with identifying and dealing with issues for individual customers, that can be implemented across the board.