Roll Number	@00677611
ID	CHC119
Name	George Wilkinson

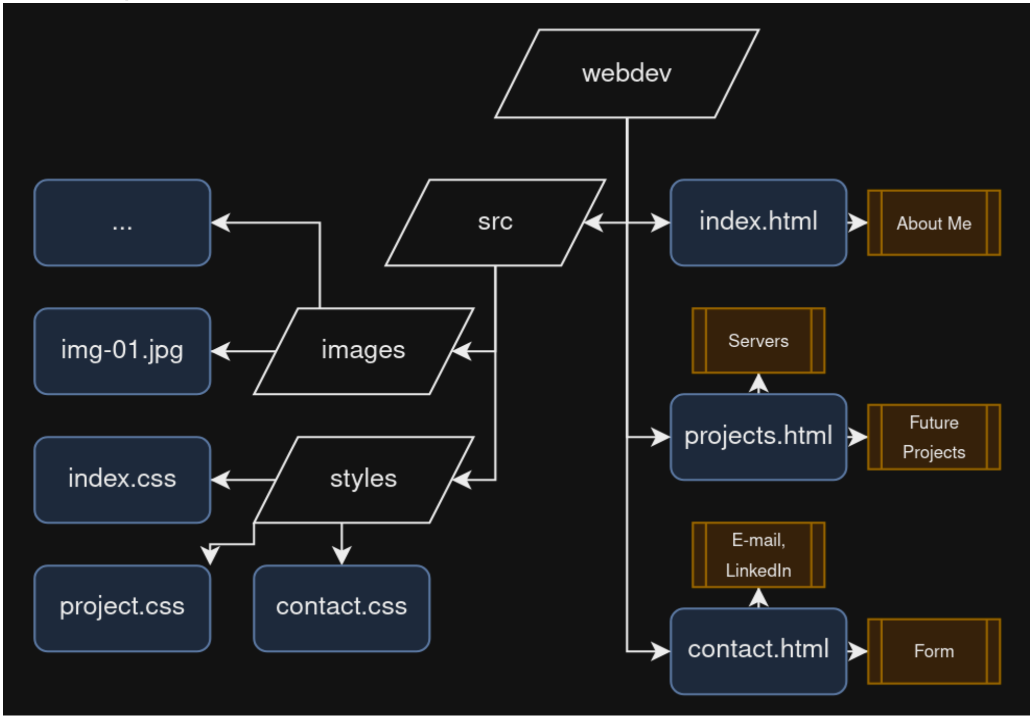
Site Content & Design

URL: http://chc119.poseidon.salford.ac.uk/webdev/index.html

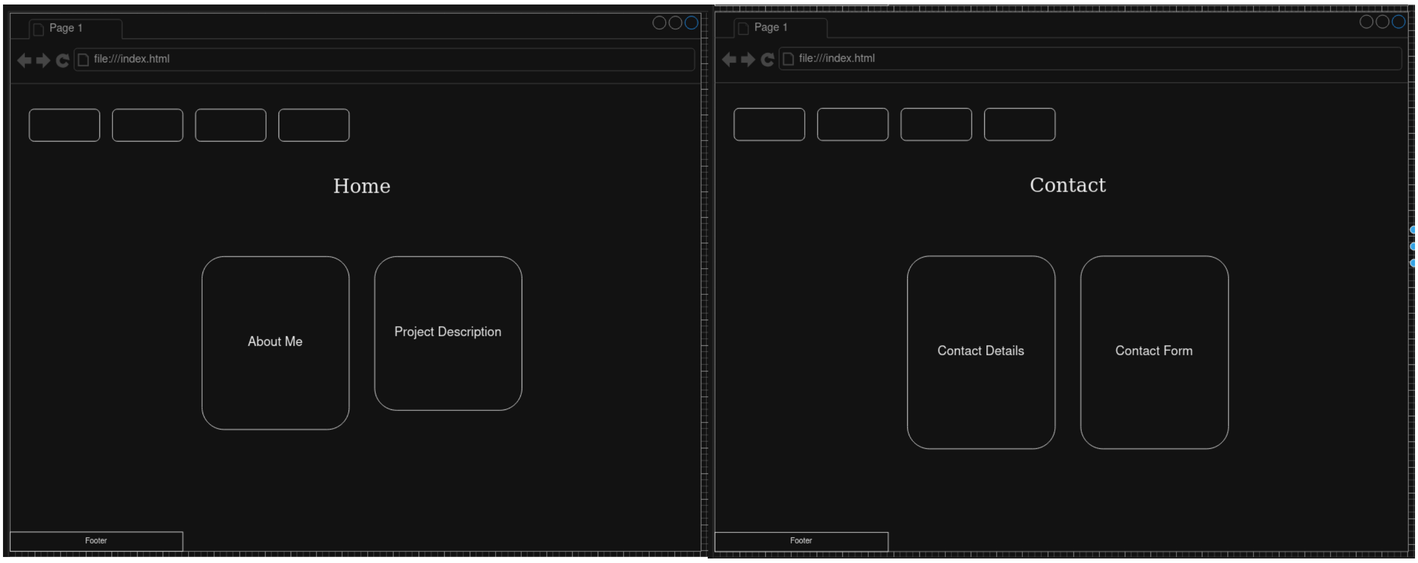
This website is designed to be a portfolio / personal blog style website, listing my web hosting, networking and server building hobby.

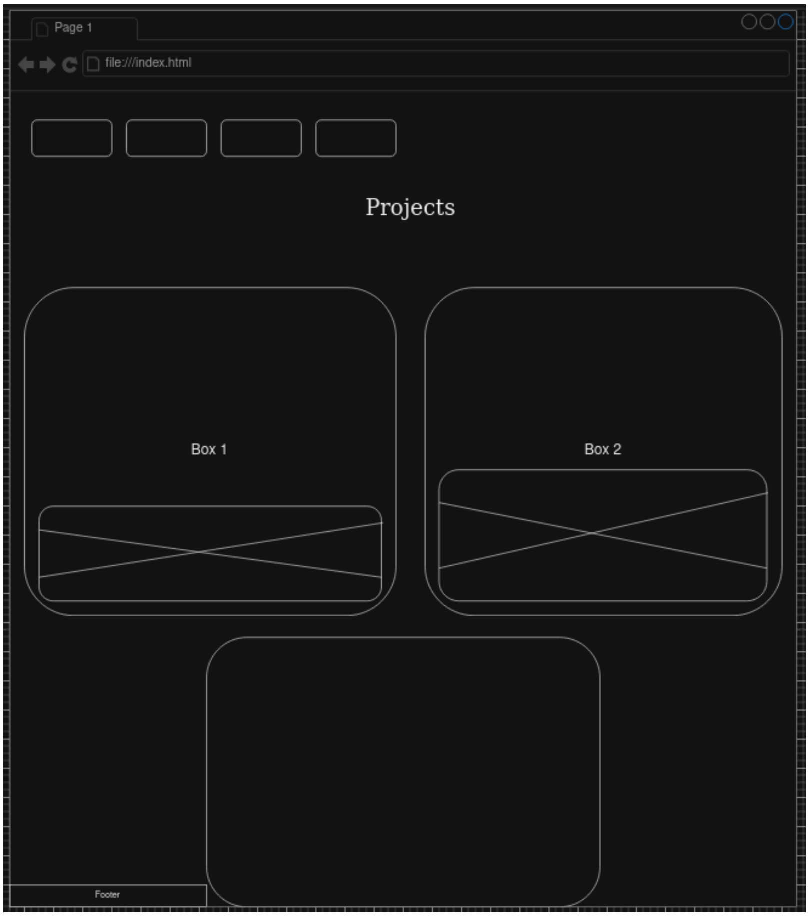
The site is designed to be liquid / elastic. A fully liquid layout would be too complex for the design, so taking some compromises with display compatibility will be likely.

Site Map:



Wireframes:





Theme Ideas:

There are plenty of themes made by people under the MIT License, allowing people to use in any commercial environment.

So far I have found 3 that I like:

https://github.com/nordtheme/nord/ (Cold, Dark)

https://github.com/morhetz/gruvbox (Warm, Retro)

https://github.com/dracula/dracula-theme (Warm, Sleek)

Use of HTML5

In my home page, I have separated the content into 2 parts: the top-bar and the content itself. This is wrapped in a division element (id “main”), so I can extend the body to the full length and still scroll reliably and properly.

Inside the “main” division I have a division (id “top-bar”), which is another wrapper, for my top-bar. Here, I decided that despite my wireframes, I wanted a responsive menu bar. This contains a division (id “toggle-navbar”), for the hamburger menu, and a section element “title-header”, due to it being for the title of the page so it can be identified from both technical and development points of view.

Inside the “toggle-navbar” division, I have 3 divisions with a class “hamburger-division” repeated adjacently, to form the structure of a hamburger when I come to the CSS. Following, I have an unordered list containing anchor elements inside list elements for each page.

In the section “tltle-header” I have simply used a header tag, containing the heading tags used for the page. The header tag is not required for the home page, since there is only one heading, but is left for consistency with other pages.

Coming to the next part, a division (id “content”), which is used to hold the informational elements of the page.

Contained within this division is several other divisions with a class named “card”. These “cards” will come to form a dynamic system that serves to allow for dynamic addition of content.

Within these “cards” I have included 3 parts to segregate the semantic function. A section with a class “card-header”, containing a header element with heading tags; a division with a class “card-content”, holding relevant tags such as paragraphs, anchors, or lists; a footer tag with a class “card-footer”, to contain less informative, but relevant information to the card’s content.

Lastly, I have a footer at the end of the page, before the body closing tag to show my name and the date of last modification.

CSS Explanation

I have fragmented my CSS files into 3 parts: an all-encompassing “index.css”, to style every page’s common elements / potential common elements; “projects.css”, to style exclusively projects.html; “contact.css”, to style exclusively contact.html. My overall aim with the website is to make it as dynamic as I can to the best of my ability without using JavaScript. This however means the CSS used can get a bit creative and convoluted, with a lot of time invested into trial & error.

index.css

I start out every website I make with a media query for light and dark mode colour schemes, to change automatically given the setting of the browser. Contained within this media query will be every element with a colour, with the respective colour for the colour scheme. I used an Open-Source colourway called Nord for the dark theme, and inverted some colours for the light theme.

For the top-bar, I use a block display to order all elements on top of each other on new lines (navbar & title). This means they will not intersect even if they viewport changes size. I use a height of 25% for a similar reason, to keep the size of the 2 parts of my page at a constant ratio to each other.

For the navbar, I have used a combination of transitions and transformations to move an unordered list from out of view on the activation of a checkbox with combinators. The hamburger menu now has a background, and spaced evenly, with a 0 opacity checkbox infront of them. By default, the unordered list holding my navigation buttons in a row direction flex-box (for dynamic additions) is out of view, transformed by 400% upwards, when the checkbox is checked, the transformation is set to none, and it returns to its intended position, in view. There is then a cubic bezier transition over 0.5 seconds – honestly I just messed with the x y values until it looked good, it could’ve been set to ease. Each of the list items in the nav-bar have a decorative division, that when the item is hovered, a combinator is used to get the child div and set its width with another cubic bezier transition.

The second part – the content, is contained in a flex-box, allowing for dynamic addition of “cards”. This row directed flex-box is allowed to wrap, and height is set to 75%, to keep the 1:3 ratio of top bar to content.

Inside this flex-box, are the “cards”, which are displayed as block elements to ensure proper placement on the page. Inside the cards is nothing complex until the projects page.

project.css

Here, I have used a collapsing list to add a lot of information in a confined space. This was done by using combinators and a checkbox, the same as my navbar. By default the height is 0 with overflow hidden; by checking the box, the height of the content becomes auto, and overflow is visible.

I also used rounded corners on images to make them less jarring in my “cards”.

contact.css

In the contact page, I have used CSS to style all of the input elements inside a flexbox to space evenly, using rounded corners and removing default styling.

CSS Techniques

Most of my CSS that I didn’t already just know came from W3Schools, only 1 was from another source. All of the techniques I had to look up came from these sources:

Using media query to adjust element attributes: https://www.geeksforgeeks.org/css3-media-query-for-all-devices/

CSS Media Query: https://www.w3schools.com/css/css3_mediaqueries.asp

CSS Combinators: https://www.w3schools.com/css/css_combinators.asp

CSS Drop-downs: https://www.w3schools.com/css/css_dropdowns.asp

CSS Navigation Bar: https://www.w3schools.com/css/css_navbar.asp

Any other techniques were learned at Sixth Form in A-Level Computer Science through tutors and assignments, or at Salford in lectures / workshops.

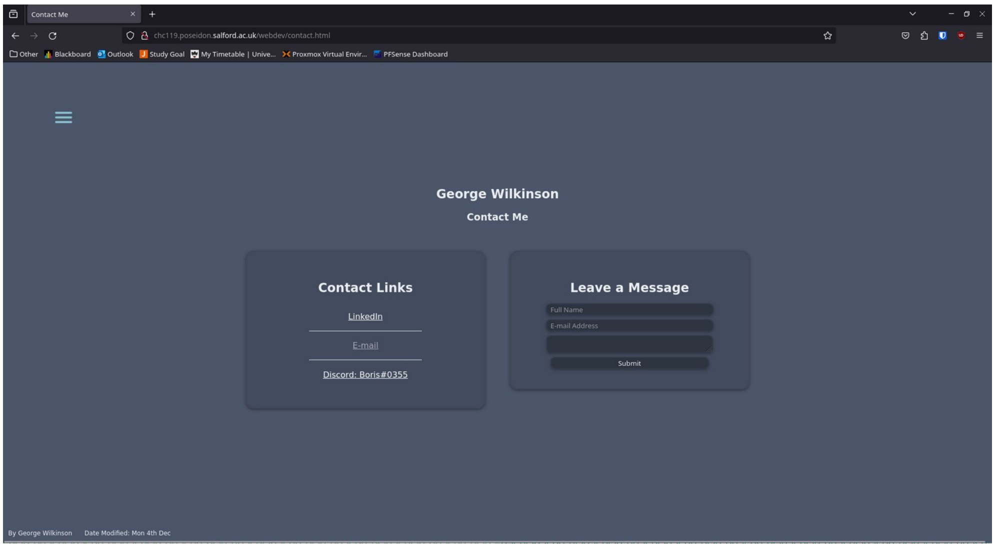
Graphics and Other Media

All media used in my website is my own work, taken from the camera on my phone, or taken from the GUI of Proxmox (https://www.proxmox.com/en/), which is licensed under the GPL-3 License (Free & Open Source). No other images, icons, buttons or other media are used. Visual elements such as the hamburger menu were made from division elements, rather than using a potentially copyrighted image.

Testing

I only have access to a Linux and Android device as of writing, so I have tested to the best of my ability cross-platform, cross-browser and using both themes.

contact.html open in Firefox:



George Wilkinson

Contact Me

Contact Links

LinkedIn

E-mail

Discord: Boris#0355

Leave a Message

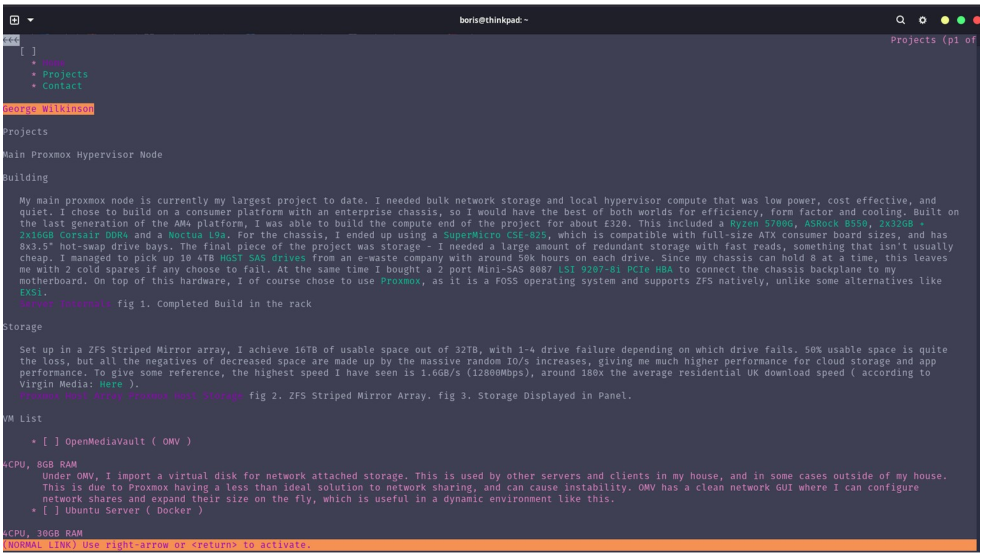
Full Name

E-mail Address

Submit

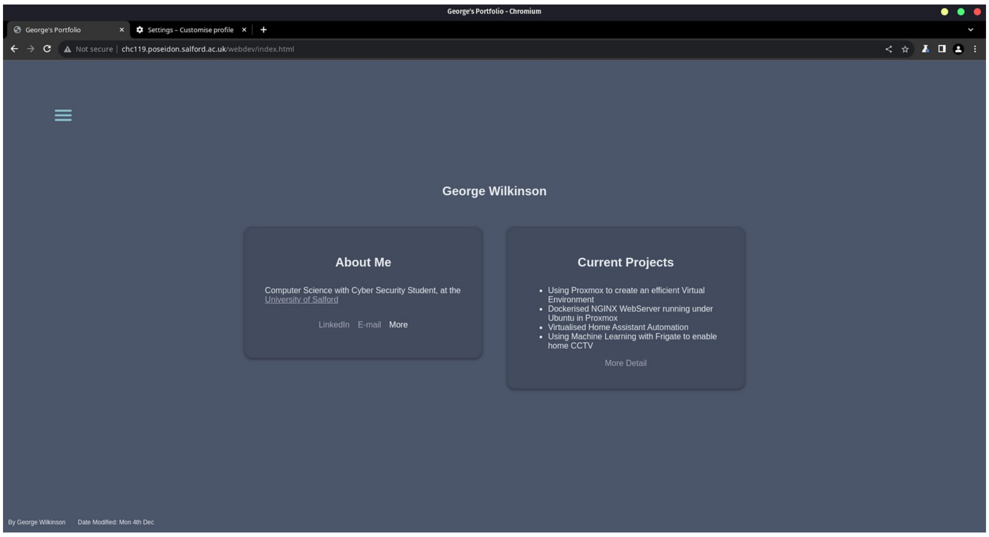
By George Wilkinson Date Modified: Mon 4th Dec

projects.html open using Lynx:



```
<<<                                                                                                Projects (p1 of
   [ ]
     * Home
     * Projects
     * Contact

George Wilkinson

Projects

Main Proxmox Hypervisor Node

Building

   My main proxmox node is currently my largest project to date. I needed bulk network storage and local hypervisor compute that was low power, cost effective, and
   quiet. I chose to build on a consumer platform with an enterprise chassis, so I would have the best of both worlds for efficiency, form factor and cooling. Built on
   the last generation of the AM4 platform, I was able to build the compute end of the project for about £320. This included a Ryzen 5700G, ASRock B550, 2x32GB +
   2x16GB Corsair DDR4 and a Noctua L9a. For the chassis, I ended up using a SuperMicro CSE-825, which is compatible with full-size ATX consumer board sizes, and has
   8x3.5" hot-swap drive bays. The final piece of the project was storage - I needed a large amount of redundant storage with fast reads, something that isn't usually
   cheap. I managed to pick up 10 4TB HGST SAS drives from an e-waste company with around 50k hours on each drive. Since my chassis can hold 8 at a time, this leaves
   me with 2 cold spares if any choose to fail. At the same time I bought a 2 port Mini-SAS 8087 LSI 9207-8i PCIe HBA to connect the chassis backplane to my
   motherboard. On top of this hardware, I of course chose to use Proxmox, as it is a FOSS operating system and supports ZFS natively, unlike some alternatives like
   EXSi.
   Server Internals fig 1. Completed Build in the rack

Storage

   Set up in a ZFS Striped Mirror array, I achieve 16TB of usable space out of 32TB, with 1-4 drive failure depending on which drive fails. 50% usable space is quite
   the loss, but all the negatives of decreased space are made up by the massive random IO/s increases, giving me much higher performance for cloud storage and app
   performance. To give some reference, the highest speed I have seen is 1.6GB/s (12800Mbps), around 180x the average residential UK download speed ( according to
   Virgin Media: Here ).
   Proxmox Host Array Proxmox Host Storage fig 2. ZFS Striped Mirror Array. fig 3. Storage Displayed in Panel.

VM List

     * [ ] OpenMediaVault ( OMV )

4CPU, 8GB RAM
       Under OMV, I import a virtual disk for network attached storage. This is used by other servers and clients in my house, and in some cases outside of my house.
       This is due to Proxmox having a less than ideal solution to network sharing, and can cause instability. OMV has a clean network GUI where I can configure
       network shares and expand their size on the fly, which is useful in a dynamic environment like this.
     * [ ] Ubuntu Server ( Docker )

4CPU, 30GB RAM
(NORMAL LINK) Use right-arrow or <return> to activate.
```

index.html open using Chromium


George's Portfolio - Chromium
George's Portfolio
Settings – Customise profile
Not secure
chc119.poseidon.salford.ac.uk/webdev/index.html

George Wilkinson

About Me

Computer Science with Cyber Security Student, at the University of Salford

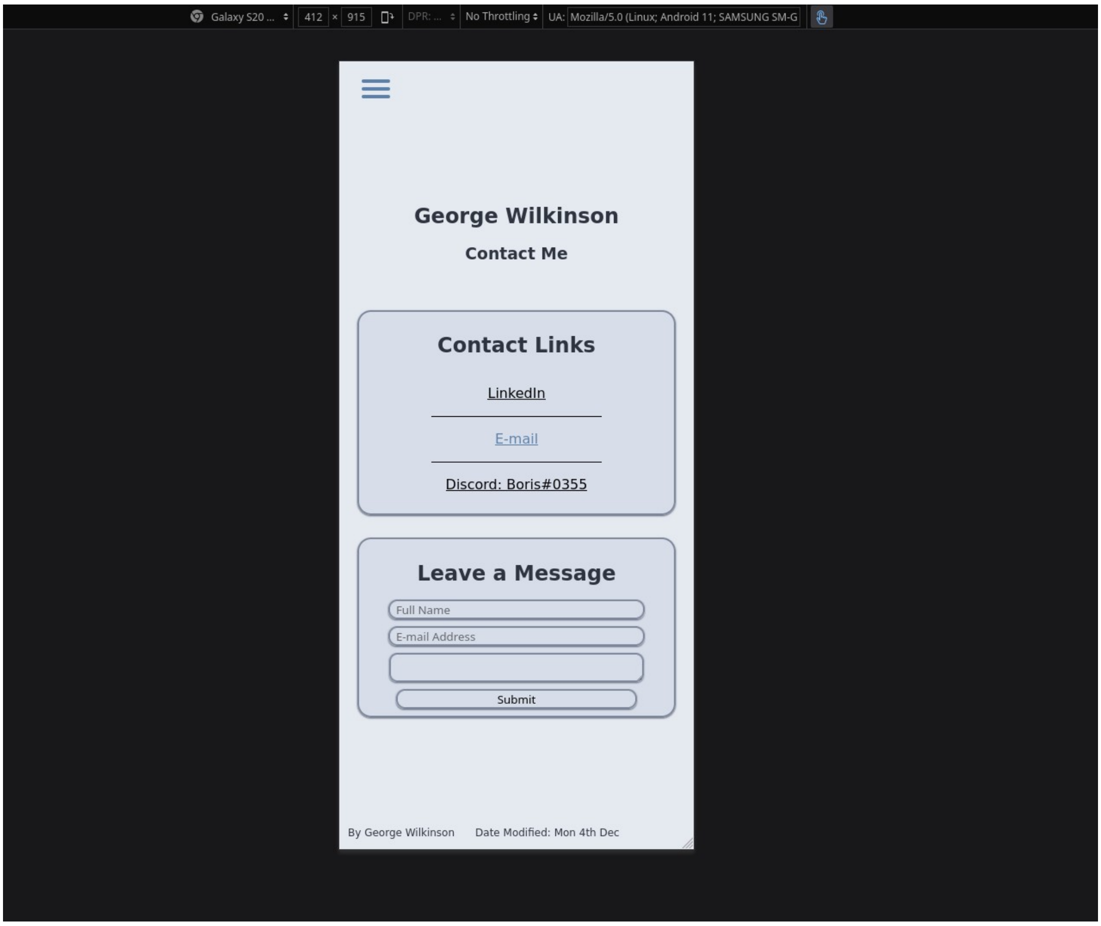
LinkedIn E-mail More

Current Projects

- Using Proxmox to create an efficient Virtual Environment
- Dockerised NGINX WebServer running under Ubuntu in Proxmox
- Virtualised Home Assistant Automation
- Using Machine Learning with Frigate to enable home CCTV

More Detail

By George Wilkinson Date Modified: Mon 4th Dec contact.html shown in vertical Galaxy S20 Ultra emulation (light mode), in Firefox:



George Wilkinson

Contact Me

Contact Links

LinkedIn

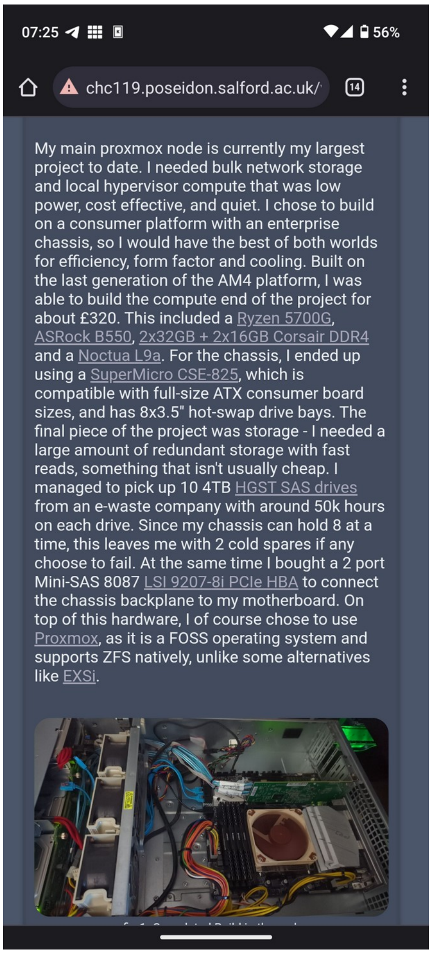
E-mail

Discord: Boris#0355

Leave a Message

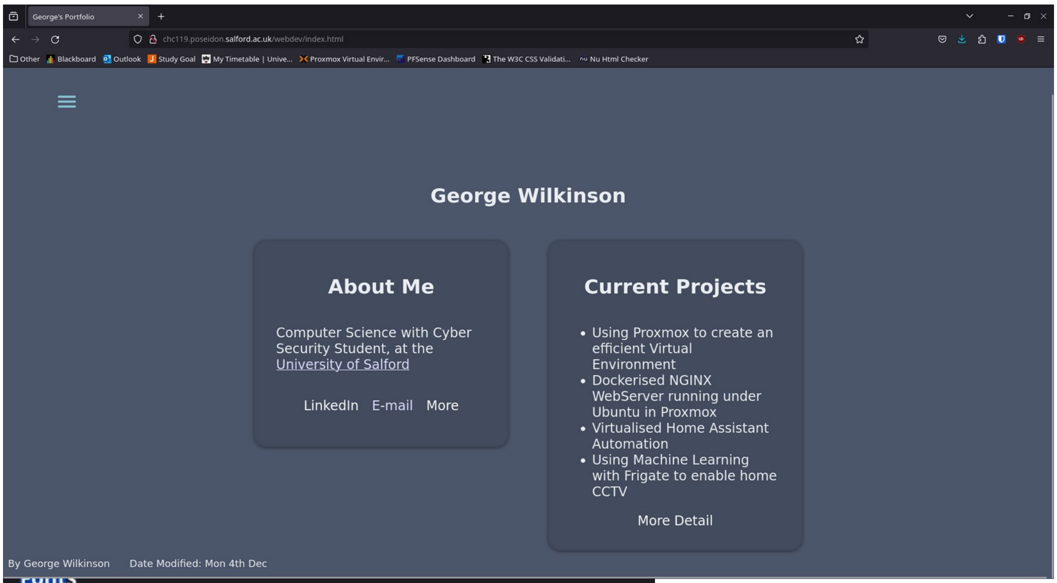
By George Wilkinson Date Modified: Mon 4th Dec

projects.html shown on my Pixel 7 in Vanadium (Chromium fork):



My main proxmox node is currently my largest project to date. I needed bulk network storage and local hypervisor compute that was low power, cost effective, and quiet. I chose to build on a consumer platform with an enterprise chassis, so I would have the best of both worlds for efficiency, form factor and cooling. Built on the last generation of the AM4 platform, I was able to build the compute end of the project for about £320. This included a Ryzen 5700G, ASRock B550, 2x32GB + 2x16GB Corsair DDR4 and a Noctua L9a. For the chassis, I ended up using a SuperMicro CSE-825, which is compatible with full-size ATX consumer board sizes, and has 8x3.5" hot-swap drive bays. The final piece of the project was storage - I needed a large amount of redundant storage with fast reads, something that isn't usually cheap. I managed to pick up 10 4TB HGST SAS drives from an e-waste company with around 50k hours on each drive. Since my chassis can hold 8 at a time, this leaves me with 2 cold spares if any choose to fail. At the same time I bought a 2 port Mini-SAS 8087 LSI 9207-8i PCIe HBA to connect the chassis backplane to my motherboard. On top of this hardware, I of course chose to use Proxmox, as it is a FOSS operating system and supports ZFS natively, unlike some alternatives like EXSi.

index.html on Firefox with font size preferences set to 24:



George Wilkinson

About Me

Computer Science with Cyber Security Student, at the University of Salford

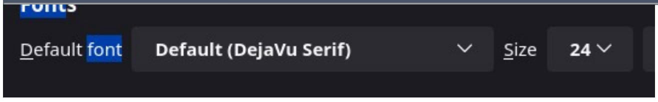
LinkedIn E-mail More

Current Projects

- Using Proxmox to create an efficient Virtual Environment
- Dockerised NGINX WebServer running under Ubuntu in Proxmox
- Virtualised Home Assistant Automation
- Using Machine Learning with Frigate to enable home CCTV

More Detail

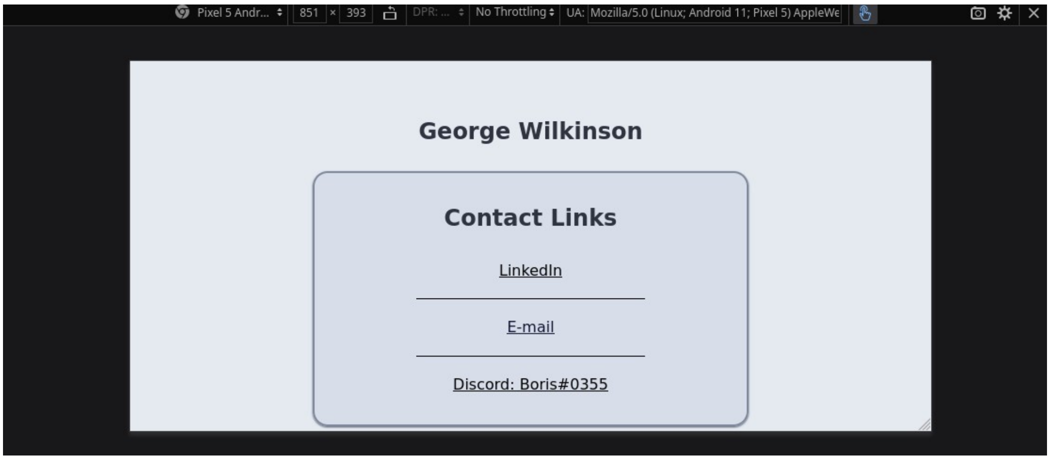
By George Wilkinson Date Modified: Mon 4th Dec



Fonts

Default font: Default (DejaVu Serif) Size: 24

contact.html on Firefox, emulating a Portrait Pixel 5 in Light theme:



Pixel 5 Andr... 851 × 393 DPR: ... No Throttling UA: Mozilla/5.0 (Linux; Android 11; Pixel 5) AppleWe

George Wilkinson

Contact Links

LinkedIn

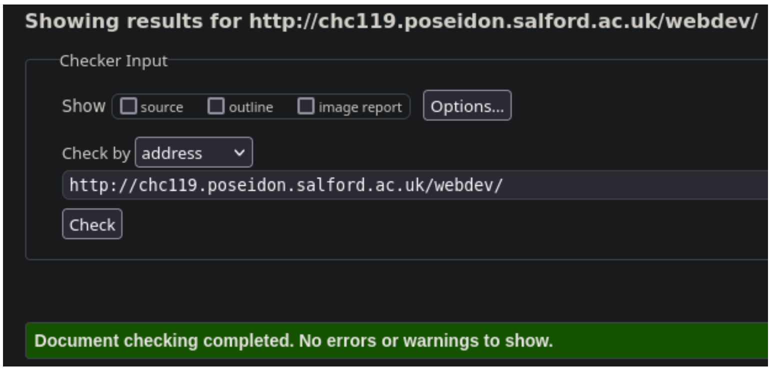
E-mail

Discord: Boris#0355

Validation

index.html

HTML:



Showing results for http://chc119.poseidon.salford.ac.uk/webdev/

Checker Input

Show □ source □ outline □ image report Options...

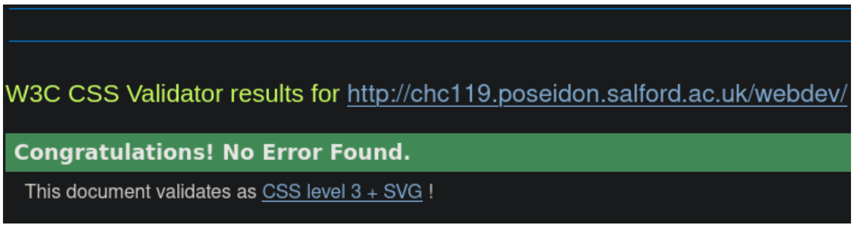
Check by address

http://chc119.poseidon.salford.ac.uk/webdev/

Check

Document checking completed. No errors or warnings to show.

CSS:

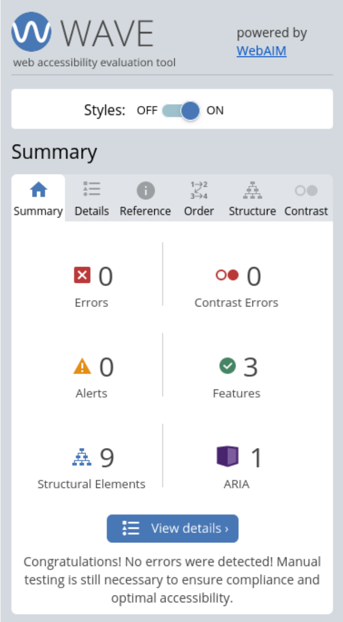


W3C CSS Validator results for http://chc119.poseidon.salford.ac.uk/webdev/

Congratulations! No Error Found.

This document validates as CSS level 3 + SVG !

Accessibility (WAVE WebAim):



Styles: OFF ON

Summary

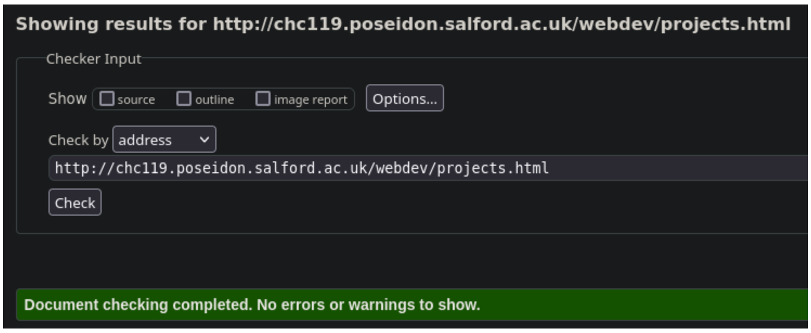
0 Errors	0 Contrast Errors
0 Alerts	3 Features
9 Structural Elements	1 ARIA

Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.

(Fully Accessible)

projects.html

HTML:



Showing results for http://chc119.poseidon.salford.ac.uk/webdev/projects.html

Checker Input

Show ☐ source ☐ outline ☐ image report Options...

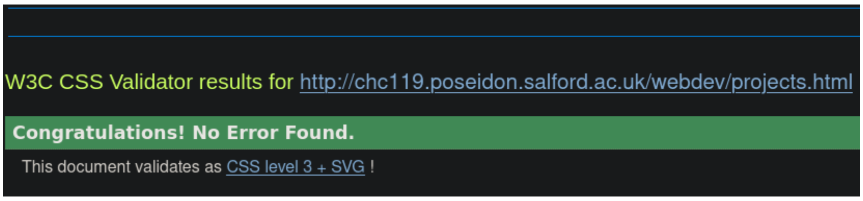
Check by address

http://chc119.poseidon.salford.ac.uk/webdev/projects.html

Check

Document checking completed. No errors or warnings to show.

CSS:

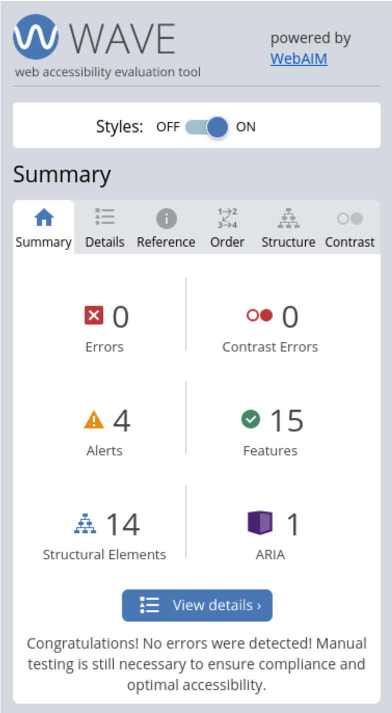


W3C CSS Validator results for http://chc119.poseidon.salford.ac.uk/webdev/projects.html

Congratulations! No Error Found.

This document validates as CSS level 3 + SVG !

Accessibility (WAVE WebAim):



0 Errors

0 Contrast Errors

4 Alerts

15 Features

14 Structural Elements

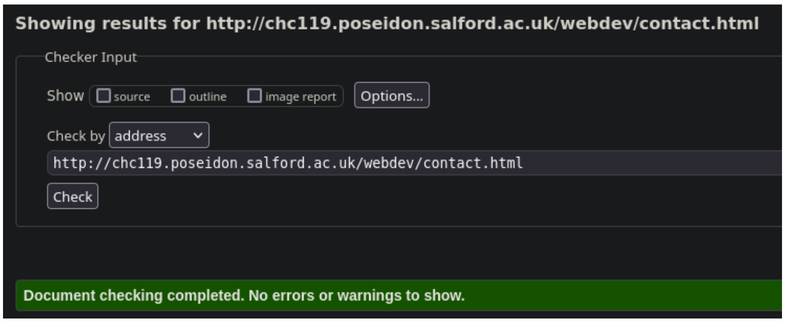
1 ARIA

Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.

(Fully Accessible)

contact.html

HTML:



Showing results for http://chc119.poseidon.salford.ac.uk/webdev/contact.html

Checker Input

Show ☐ source ☐ outline ☐ image report Options...

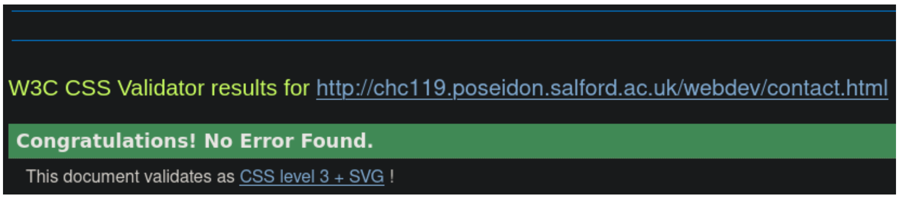
Check by address

http://chc119.poseidon.salford.ac.uk/webdev/contact.html

Check

Document checking completed. No errors or warnings to show.

CSS:

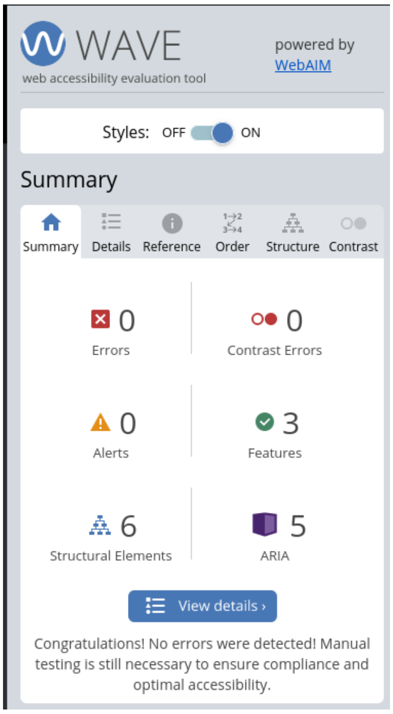


W3C CSS Validator results for http://chc119.poseidon.salford.ac.uk/webdev/contact.html

Congratulations! No Error Found.

This document validates as CSS level 3 + SVG !

Accessibility (WAVE WebAim):



0 Errors

0 Contrast Errors

0 Alerts

3 Features

6 Structural Elements

5 ARIA

Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.

(Fully Accessible)

Copyright Permission

All content produced here is work of my own. Any and all media used is free from copyright from third parties, and pictures taken are my sole intellectual property. Drawio was used the in design phase (https://app.diagrams.net/) (https://github.com/jgraph/drawio), where screenshots were taken, but is free for commercial use without any permission or watermarking (Apache v2). Screenshots of the Proxmox Graphical Interface are also free to use, due to the GPL-3 License (https://github.com/proxmox).