

SortMi Report

Site content and design:

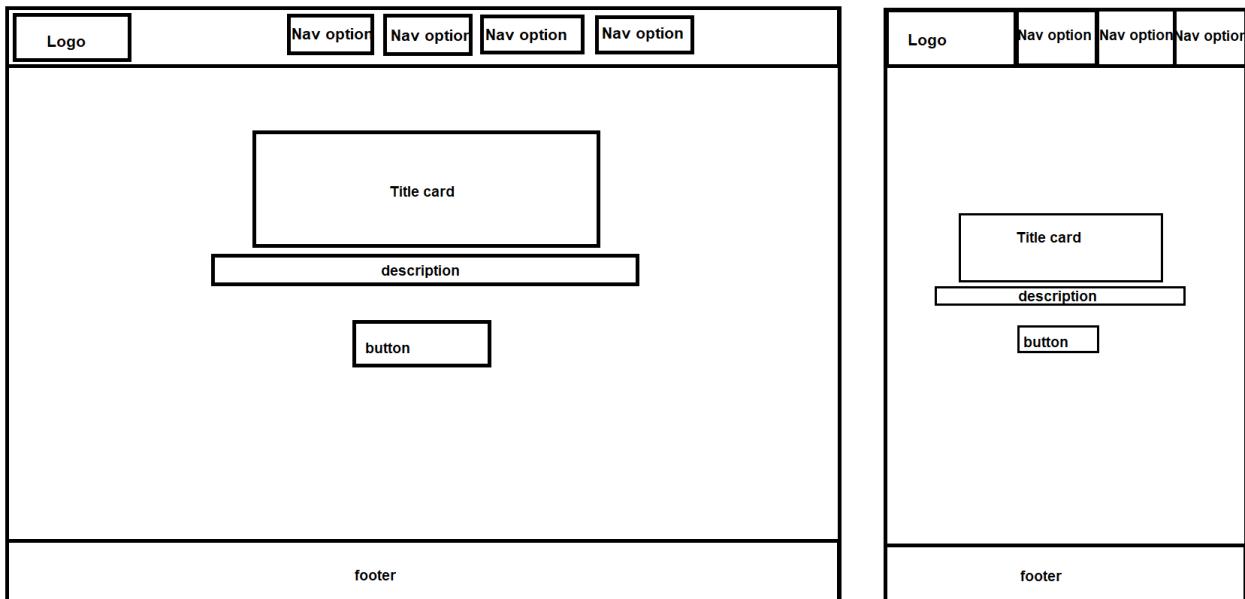
URL of site - <http://kennelling-sanka.poseidon.salford.ac.uk/webdev>

SortMi website is about learning sorting algorithm code through canvas graph visualization demonstration, providing information on different sorting algorithms, their big O time complexities and displaying their code. It features an interactive layout with a visualizer's section, a contact page, and a visually appealing colour scheme design.

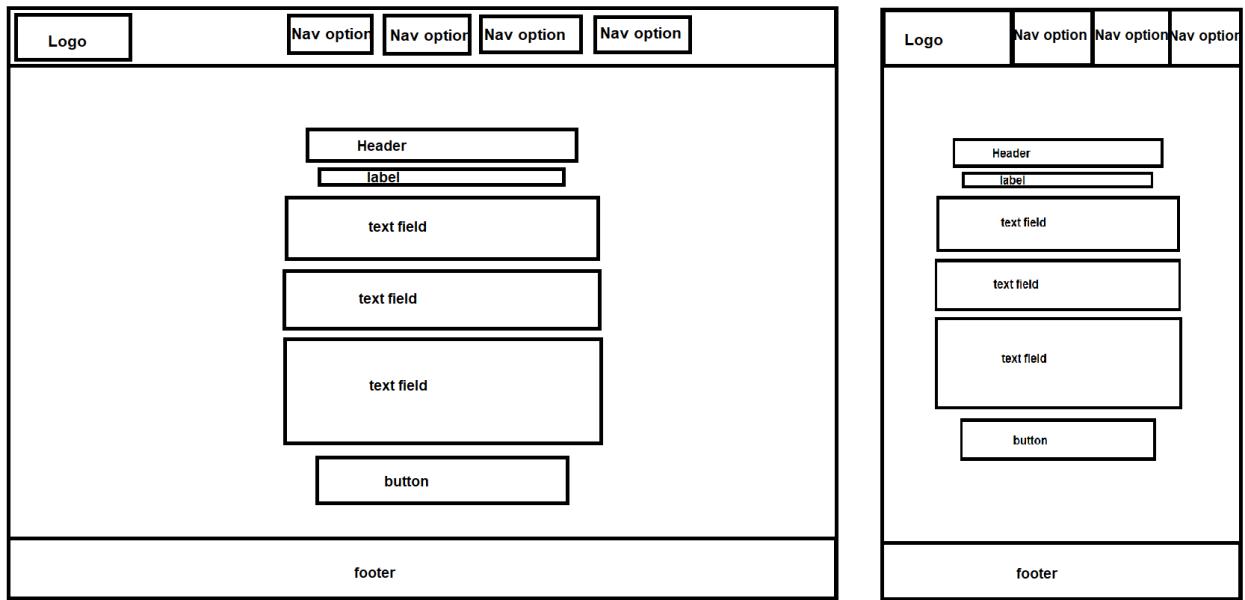
The website uses a fluid/liquid design as the layout adjusts dynamically based on the screen size. The use of grid, flexbox, and dynamic widths allows elements like the navigation bar, algorithm options, and footer to scale accordingly with different screen sizes. Additionally, I have also used media queries that are implemented to modify the layout for smaller screens such as a phone or tablet, making sure it is responsive and provides a user-friendly experience across devices. This makes the website adaptable as opposed to fixed, allowing for fluid resizing and content organization.

Website sketches:

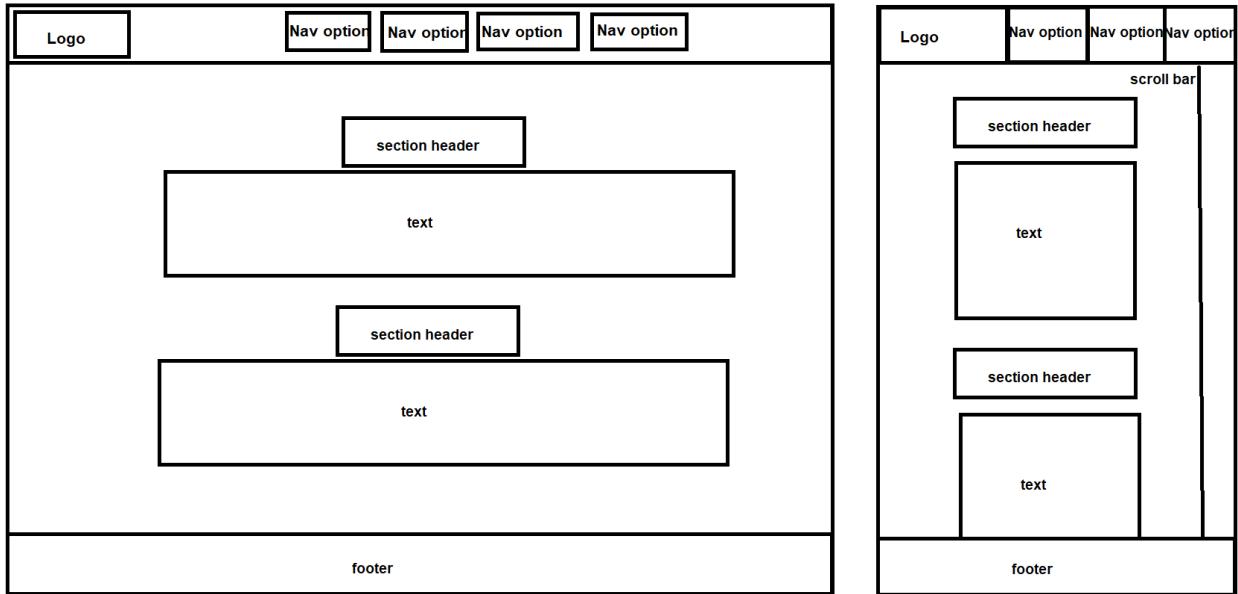
Home page [Desktop + mobile]



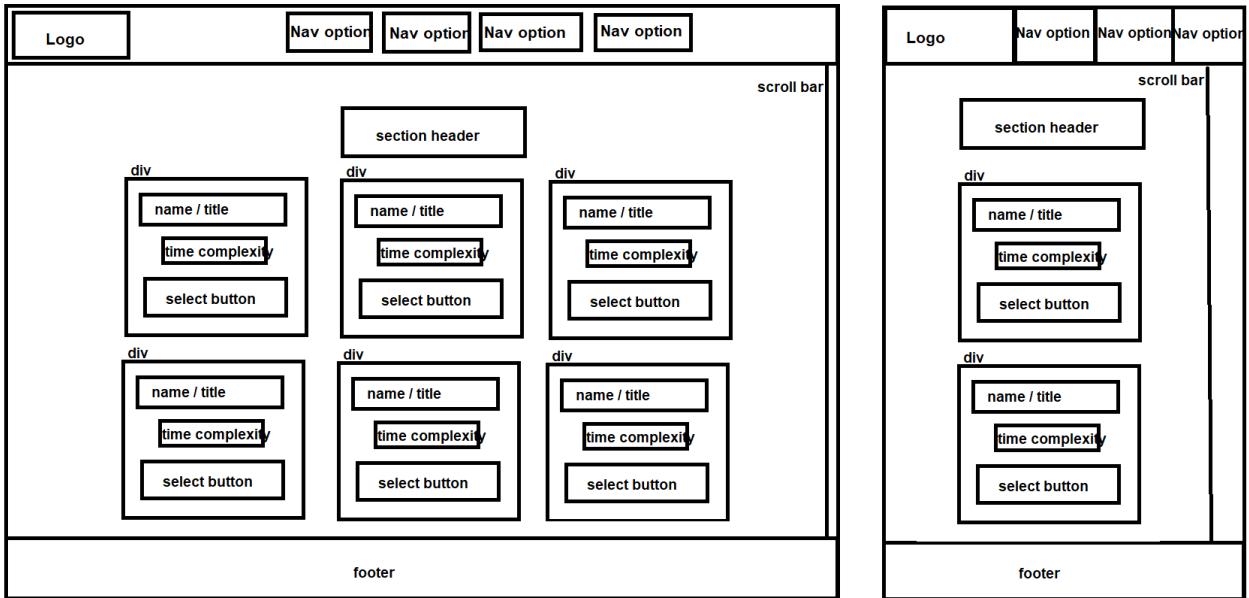
Contact page [Desktop + mobile]



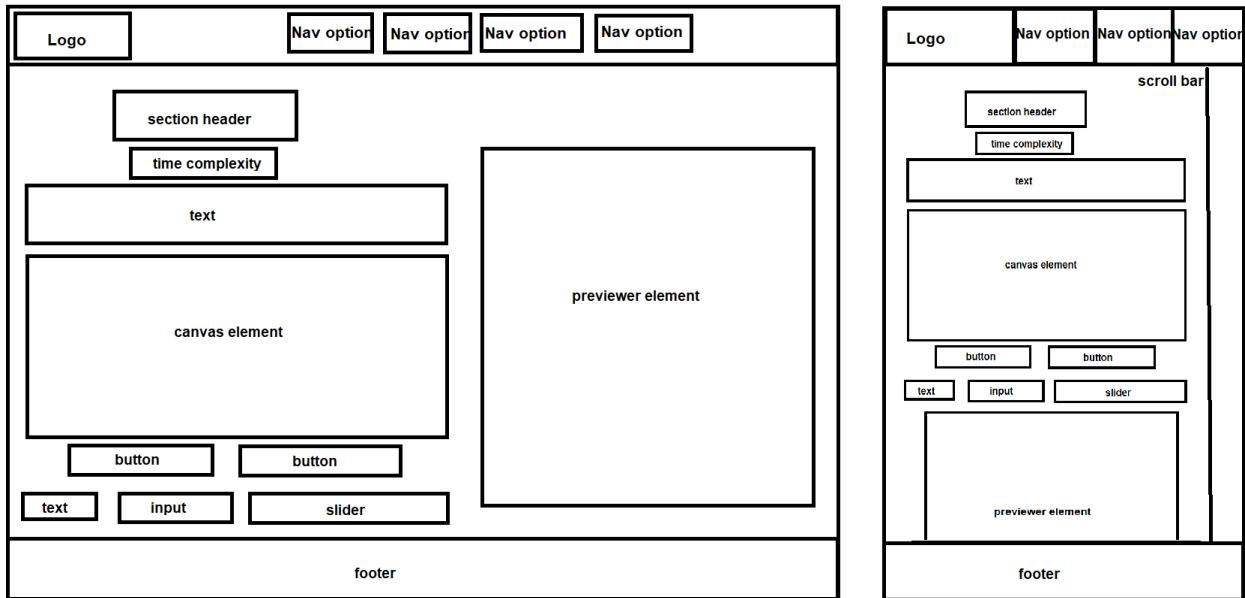
About page [Desktop + mobile]



Visualizer page [Desktop + mobile]



Algorithm page [Desktop + mobile]



Use of HTML5:

The SortMi home page is structured using HTML5 semantic elements to improve readability, accessibility and maintainability. The document starts with `<!DOCTYPE html>` to define it as an HTML5 document. The `<html>` tag includes `lang="en"`, specifying English as the primary language.

In the `<head>` section, the `<meta charset="UTF-8">` ensures proper text encoding, while `<meta name="viewport" content="width=device-width, initial-scale=1.0">` enables responsiveness. The `<title>` sets the page title and `<link rel="stylesheet" href="styles.css">` links an external stylesheet for styling.

The navigation bar is contained within a `<nav>` element, which improves accessibility and SEO. Inside is a `` element structures the links correctly using `` elements combined with `<a>` elements for redirection. The SortMi logo is also inside an `<a>` tag, linking to the homepage. The navigation includes links for "Home", "Visualizers", "About" and "Contact". Some links have the class `mobile-only`, ensuring they only appear in mobile views for a cleaner layout on larger screens.

The main content includes a `<section>` element. Inside is a `<h1>` heading introducing the website, followed by a `<p>` explaining its purpose. A `<button>` is included with an `onclick` event that redirects users to the algorithm visualization page.

At the bottom, a `<footer>` element contains a `<p>` stating "University 2025 SortMi Project." Which is consistent across pages.

This structure follows best practices for web development. The use of semantic elements like `<nav>`, `<section>` and `<footer>` makes the page more readable and accessible. The viewport meta tag and mobile-specific navigation improve responsiveness. Overall, the page is well-structured, improving usability and SEO while maintaining a clean and organized layout.

In addition, more advanced HTML5 elements are used in `visualizer.html`, additional HTML elements such as `<canvas>` and `<pre>` are used to enhance the functionality and presentation of sorting algorithms. The `<canvas>` element provides a dynamic space for rendering visual representations of sorting processes. This is needed for creating interactive animations, allowing users to see how different sorting algorithms work.

The `<pre>` element is used within the `<div id="codeContainer">` to display formatted code snippets. The `<pre>` tag preserves whitespace and indentation, ensuring that

algorithm implementations are presented clearly and legibly without requiring extra styling. This is particularly useful for educational purposes, as it allows users to compare the algorithm's code with its visual execution.

Additionally, interactive elements such as `<button>` and sliders (`<input type="range">` and `<input type="number">`) are included. These allow users to generate new arrays, adjust array sizes, and initiate sorting. This makes the visualizer both informative and user-friendly.

CSS Explanation:

Layout and Positioning

The design follows a clean and modern layout with structured alignment and responsiveness. A flexbox approach is applied throughout, particularly in the navbar (nav), welcome section (#welcome-section), algorithm section (#algorithm) and footer. The website employs a column-based layout with centered content for the welcome and algorithm sections. The navigation bar is set as sticky to remain visible during scrolling. Additionally, media queries ensure adaptability across various screen sizes, optimizing the experience for desktop, tablet and mobile users.

Color Scheme

The chosen color palette consists of blue gradients and neutral tones. Backgrounds incorporate a linear-gradient (to bottom right, #cceeff, #99ccff) for a smooth transition effect. The navbar and footer feature a dark gray shade (#333) to create a strong contrast. Hover effects utilize #007bff (blue) and #41e033 (green) to highlight interactive elements. Text is primarily #333 for readability, while darker elements feature white text (#fff) to enhance contrast.

Interesting CSS Elements

- **Navigation Bar:**
 - position: sticky ensures it remains visible while scrolling.
 - box-shadow provides a subtle elevation effect.
 - Hover effects use transform: translateY(-3px), and a ripple animation (@keyframes ripple) adds interactivity.
- **Welcome Section:**
 - A linear-gradient background enhances visual appeal.
 - Button hover effects dynamically change color (background-color: #41e033).
- **Algorithm Section:**
 - flex-wrap allows elements to adjust dynamically.
 - The sortContainer:hover effect includes transform: scale(1.05), making elements more prominent.
- **Sorting Visualizer (#sortingCanvas)**
 - Different widths are applied based on screen sizes via media queries.
 - A dark gray background (#2e2b2b) improves visibility.

- **Responsive Design Adjustments:**
 - The navbar hides non-essential links on smaller screens.
 - Sorting containers transition from 3-per-row (33.33%) on large screens to 1-per-row (100%) on mobile.

CSS Techniques

In this project, several CSS techniques were employed, many of which I had learnt from a pre-studied Udemy course titled "The Complete Full-Stack Web Development Bootcamp" (Section 7: Intermediate CSS, Lessons 43 – 46, Section 8: Advanced CSS, Lessons 48 – 52, Section 9: Flexbox, Lessons 54-58).

Key techniques include:

- **Flexbox Layout:** Used throughout the website for layout control, including the navbar, welcome section, algorithm section, and footer. Flexbox enables efficient and flexible alignment of elements. This is a technique learned from the "Flexbox" section of the course.
- **Sticky Navigation Bar:** The navigation bar was set to position: sticky to ensure visibility during scrolling, a technique learned from the "Intermediate CSS" section of the course.
- **Hover Effects:** Dynamic hover effects such as transform: translateY(-3px) and @keyframes ripple were implemented to create interactive elements. These were demonstrated in the course's "Advanced CSS" section.
- **Responsive Design with Media Queries:** Applied media queries to adjust content based on screen size. This was part of the "Advanced CSS Lesson 51. Media queries" in the course.
- **Linear Gradients:** A visually appealing linear gradient background was used for the welcome section, learnt from the Foundation Year web development coursework.

These techniques helped shape the design and functionality of the website, improving both aesthetics and user experience.

Graphics and other media

There is no copyrighted material used in this website. The theme didn't require images as it instead uses an animated canvas element with vanilla JavaScript code integration linked content comprises of the following files:

- Index.html
- Contact.html
- AboutUs.html
- Algorithm.html
- Visualizer.html
- Styles.css
- Script.js

Home page redirects:

```
<link rel="stylesheet" href="styles.css">
<a href="index.html" class="logo">SortMi</a>
<li><a href="index.html">Home</a></li>
<li class="mobile-only"><a href="algorithm.html">Visualizers</a></li>
<li class="mobile-only"><a href="aboutUs.html">About</a></li>
<li class="mobile-only"><a href="contact.html">Contact</a></li>
<button onclick="window.location.href='algorithm.html'">Start Exploring</button>
```

About us page & Contact page redirects:

```
<link rel="stylesheet" href="styles.css">
<a href="index.html" class="logo">SortMi</a>
<li><a href="index.html">Home</a></li>
<li class="mobile-only"><a href="algorithm.html">Visualizers</a></li>
<li class="mobile-only"><a href="aboutUs.html">About</a></li>
<li class="mobile-only"><a href="contact.html">Contact</a></li>
```

Visualizer page redirects:

```
<link rel="stylesheet" href="styles.css">
<a href="index.html" class="logo">SortMi</a>
<li><a href="index.html">Home</a></li>
<li class="mobile-only"><a href="algorithm.html">Visualizers</a></li>
<li class="mobile-only"><a href="aboutUs.html">About</a></li>
<li class="mobile-only"><a href="contact.html">Contact</a></li>
<button
  onclick="window.location.href='visualizer.html?algo=bubble'">Select</button>
<button
  onclick="window.location.href='visualizer.html?algo=merge'">Select</button>
<button
  onclick="window.location.href='visualizer.html?algo=quick'">Select</button>
<button
  onclick="window.location.href='visualizer.html?algo=bogo'">Select</button>
<button
  onclick="window.location.href='visualizer.html?algo=insertion'">Select</button>
```

Algorithm page redirects:

```
<link rel="stylesheet" href="styles.css">
<a href="index.html" class="logo">SortMi</a>
<li><a href="index.html">Home</a></li>
<li class="mobile-only"><a href="algorithm.html">Visualizers</a></li>
<li class="mobile-only"><a href="aboutUs.html">About</a></li>
<li class="mobile-only"><a href="contact.html">Contact</a></li>
<script src="script.js"></script>
```

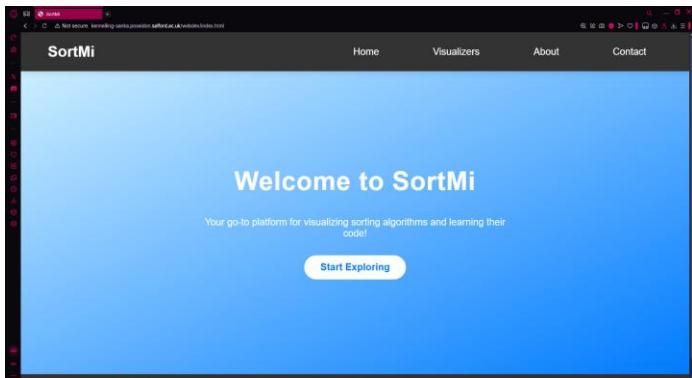
Testing

OperaGX:

100% zoom



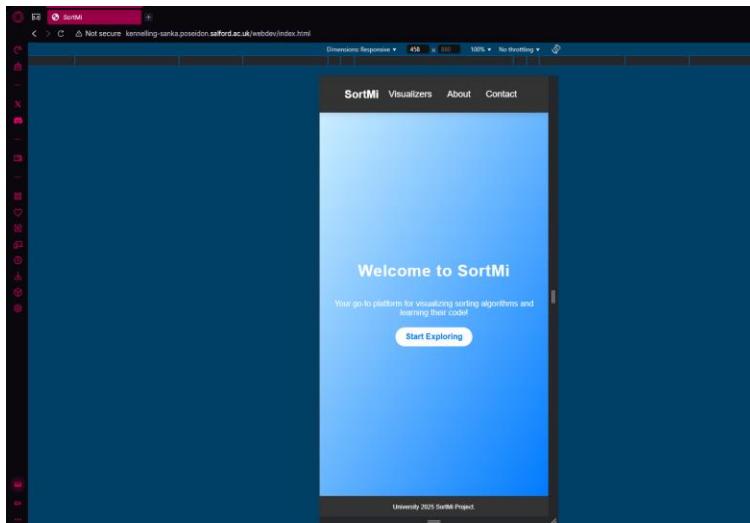
150% zoom



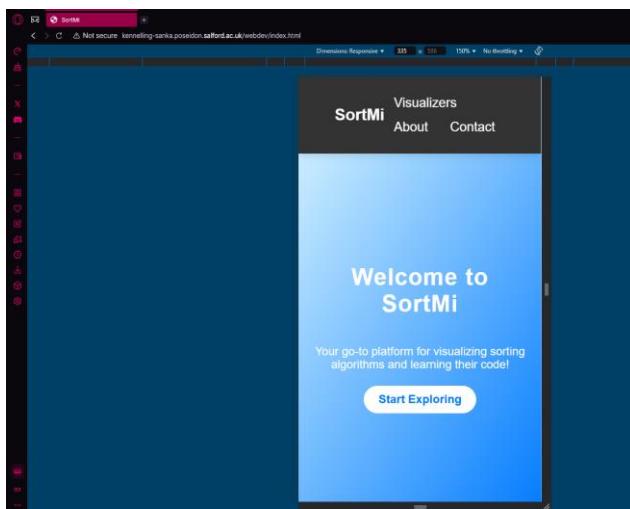
Comments: all elements scale accordingly to zoom increase without fault.

OperaGX – mobile viewer:

100% zoom



150% zoom



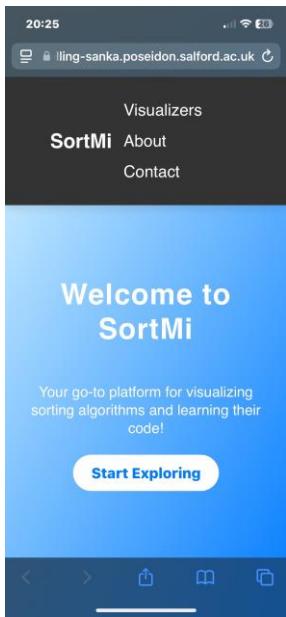
Comments: zooming by +50% causes the nav bar to stack inline-block style to allow content to fit.

Safari – iphone

Zoom 100%



Zoom 150%



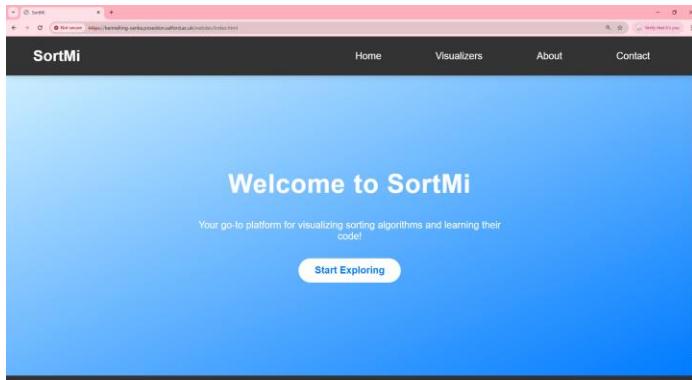
Comments: zooming by +50% causes the nav bar to stack inline-block style to allow content to fit.

Google Chrome

Zoom 100%



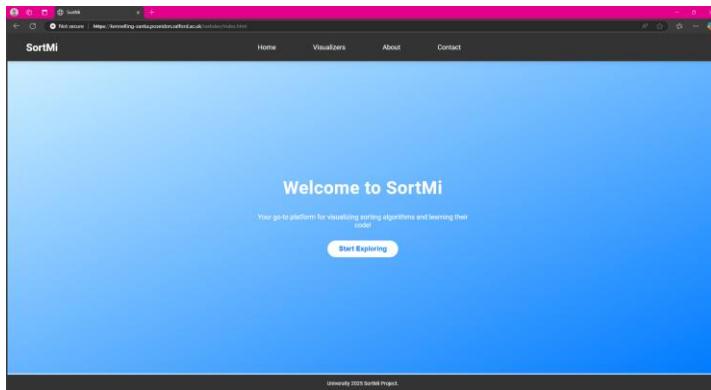
Zoom 150%



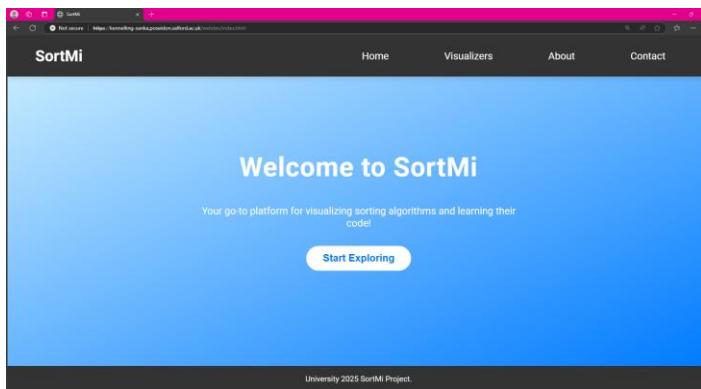
Comments: all scaling works as intended except for the footer which requires scrolling.

Edge

Zoom 100%



Zoom 150%



Comments: all scaling works as intended

Validation of HTML5 & CSS:

Index.html

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for uploaded file index.html

Checker Input

Show source outline image report Options...

Check by No file chosen
Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.
Used the HTML parser.
Total execution time 2 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 25.3.6

Contact.html

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for contact.html

Checker Input

Show source outline image report Options...

Check by No file chosen
Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.
Used the HTML parser.
Total execution time 3 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 25.3.6

AboutUs.html

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for aboutUs.html

Checker Input

Show source outline image report Options...

Check by No file chosen
Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.
Used the HTML parser.
Total execution time 3 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 25.3.6

Algorithm.html

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for algorithm.html

Checker Input

Show source outline image report

Check by No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 3 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 25.3.6

Visualizer.html

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for visualizer.html

Checker Input

Show source outline image report

Check by No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 4 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 25.3.25

Styles.css

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for styles.css

Checker Input

Show source outline image report

Check by No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Total execution time 12 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 25.3.6