

Web Page Design and Production

CyberHeart

Tutor: Eiman Aeiad

Student Number: Denys Bezzubenko

Table of Contents

CyberHeart	1
Table of Contents	2
Table of Figures	3
1. Understand the basic constituent elements of a Web Page.	4
1.1. Explain at least three main basic constituent elements of a web page.	4
• Header	4
Navigation	4
• Images	5
• Footer	5
• Links	5
1.2. Describe different graphic image types on a web page.	5
• JPEG(JPG)	5
• PNG	5
• SVG	5
• WebP	5
• Overall	5
2. Understand the basic language code used in web page production	6
2.1. Describe the use of tags in current web page creation.	6
• Structural Tags	6
• Content Tags	6
• Form Tags and Attributes	6
• Universal tags	7
3. Understand how to design and produce a web page	7
3.1. Plan the layout of the web page	7
• Wireframes	7
○ Home	7
○ Pricing	8
○ Methodology	8
○ About us	9
○ Request Audit	9
• Colour choices	9
• Fonts	10
• Content decisions	10
3.2. Produce at least one web page.	10
• The main structure of the document	10
• Header, Navigation and Footer	10
• Pricing Blocks	10
• Comparing table	11
• Request Audit	11
3.3. Test and evaluate the test page using at least one web browser.	11
• Header	11

• Footer	11
• Pages	12
• Evaluation	12
4. Understand how to make a web page 'live' on the internet.	13
4.1. Explain the stages involved in making a web page available for general viewing across the internet.	13
1. Download Git	13
2. Creating a repository	13
3. Connecting to a remote repository	13
4. Activation of GitHub Pages	13
5. Functionality check	14
References	14

Table of Figures

Fig. 1.1 Web-page structure	Image	Pg. 5
Fig. 3.1 Home Page Maket	Image	Pg. 8
Fig. 3.2 Pricing Page Maket	Image	Pg. 9
Fig. 3.3 Methodology Page Maket	Image	Pg 9
Fig. 3.4 About Us Page Maket	Image	Pg 10
Fig. 3.5 Request Audit Page Maket	Image	Pg 10

1. Understand the basic constituent elements of a Web Page.

1.1. Explain at least three main basic constituent elements of a web page.

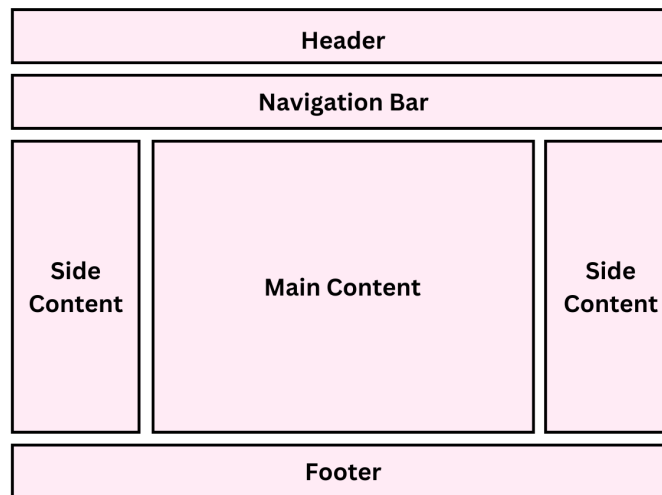


Fig. 1.1 Web-page structure

- **Header**
 - The **<header>** tag is used for implementing the upper part of a web page. Usually, **the logo, site name, and navigation** are a part of the page. Due to this, the website must not only identify the resource but also create a first impression. The header is an entity which is a visual and mental model of the organisation, containing the elements that help user orientation, **'where am I'**. According to Matt Yau (2025), headers create a map that helps all visitors understand and navigate your pages, ensuring they know where they are and can find the tickets they want.
- **Navigation**
 - Navigation is implemented through links **<a>** within the **<nav>** tag. It helps users navigate the website's pages and sections. Navigation helps organise the resource and makes the interface easier to understand. The first priority of an approach is improving UX; the similar ones remain low, not confusing, and accessible to all. According to Joshua Hay (2023), the success of a website is more dependent on the navigation design.
- **Images**
 - The **** tag allows us to insert images. This may include logos, photographs, printed text, or icons. The primary goal of these visuals is to represent information better than mere words. The inclusion of images ensures better engagement from the user. It also aids in creating a brand's

visual identity. At the same time, optimised formats need to be used for efficient loading of pages.

- **Footer**
 - This is the bottom or footer part of the page, which is marked by the tag **<footer>**. You can find its **contact details**, **link to social networks**, and **legal notices**. Another section of the page structure, which rounds off the page, is called the footer. It boosts confidence in the website and improves usability.
- **Links**
 - Links or hyperlinks **<a>** are the basis of the hypertext web. Hyperlinks facilitate the connection between separate pages, whether on the same website or to other websites. A webpage without links is like a stand-alone document. Links are the building blocks of the whole web.

1.2. Describe different graphic image types on a web page.

- **JPEG(JPG)**
 - The **JPEG** format, which might also be referred to as **JPG**, **stores photos** and complex images. It uses a compression technique where the file size is **reduced** at the expense of the quality of the file. Images and photographs that are **JPEG** are okay for the **background**.
- **PNG**
 - A **PNG** image supports transparency and **high-quality** without loss. It's employed for **logos**, **icons** and **graphical interface elements**. Nonetheless, the files are usually **larger** than a **JPEG**.
- **SVG**
 - **SVG** is a vector format. One of the primary benefits is that it can be scaled up through increased resolution and size without compromising on quality. Thus, it is perfect for **logos** and **icons**, especially in responsive design.
- **WebP**
 - A contemporary format that guarantees more efficient compression as compared to **JPEG** and **PNG**. Although it does not affect the quality, it reduces the file size. According to Google Developers (2023), **WebP** lossy images are **25-34%** smaller than comparable **JPEG** images at equivalent SSIM quality index.
- **Overall**
 - Your graphic image's format matters for delivery. According to the Web.Dev (2023), optimising images can significantly improve performance on your website. Different image formats affect page loading speed, performance quality, and ultimately users' experiences in different ways.

2. Understand the basic language code used in web page production

2.1. Describe the use of tags in current web page creation.

- Structural Tags

- Structural elements like **<header>**, **<nav>**, **<main>**, **<section>**, and **<footer>** are a great help in the construction of web pages since they make sure that the structure is defined. These elements communicate meaning to the browser as well as the search engines and define the different areas of the document.

The top section of a page, the **<header>** element, usually contains some branding or nav. The links in this area are for the **<nav>** element. The page's main tag holds the most relevant information of a web page, while the section tag helps a user to group the content in a logical block. The footer defines the bottom part of an article, page or section and typically contains contact or legal information.

The use of semantic structural tags makes the content of a website easy to read, maintain and understand. It also makes websites more accessible. Besides, it optimises the website for search engines (**SEO**) as browsers understand the purpose of each section.

- Content Tags

- Content tags including **<h1>** to **<h6>**, **<p>**, ****, **** logically organize the text information. Headings are arranged in hierarchy levels; the **<h1>** is the top heading, and the lower levels are the subsections. The hierarchy enhances the user-friendliness and understanding of search engines regarding the content.

The **<p>** paragraph tags break down text into sections which are more readable. The **** and **** list also helps you section the information neatly. Most of these elements need opening and closing tags. For example: **<p> This is a paragraph.</p>**. The first part indicates the start of the element, the latter indicates the end (with **/**).

- Form Tags and Attributes

- The **<form>**, along with **<input>**, **<textarea>**, and **<select>** tags, belong to a separate category. Web forms enable website visitors to communicate with the site and facilitate data collection.

Attributes like **required**, **type**, and **placeholder** enhance the functionality of the element. For example, **type="email"** guarantees the right input format. The

nesting principle is a semantic relationship that indicates that one element contains another one.

- Universal tags

- The **<div>** tag is a generic container that assists in grouping content and creating layouts. Unlike a semantic tag, the **<div>** tag does not convey any meaning. According to Scott O'Hara (2022), the div element has no special meaning at all. It is merely a structuring tag. This is used mostly with **CSS** for layout. An essential key to valid and working **HTML** is proper nesting of the tags.

3. Understand how to design and produce a web page

3.1. Plan the layout of the web page

- Wireframes

Each page was wireframed with a schematic layout showing where items go without styling yet. Layouts were developed for the **Home**, **About us**, **Methodology**, **Pricing**, and **Request Audit** pages, as well as the **Header** and **Footer**. All layouts provided below were created on the Figma platform.

- Home

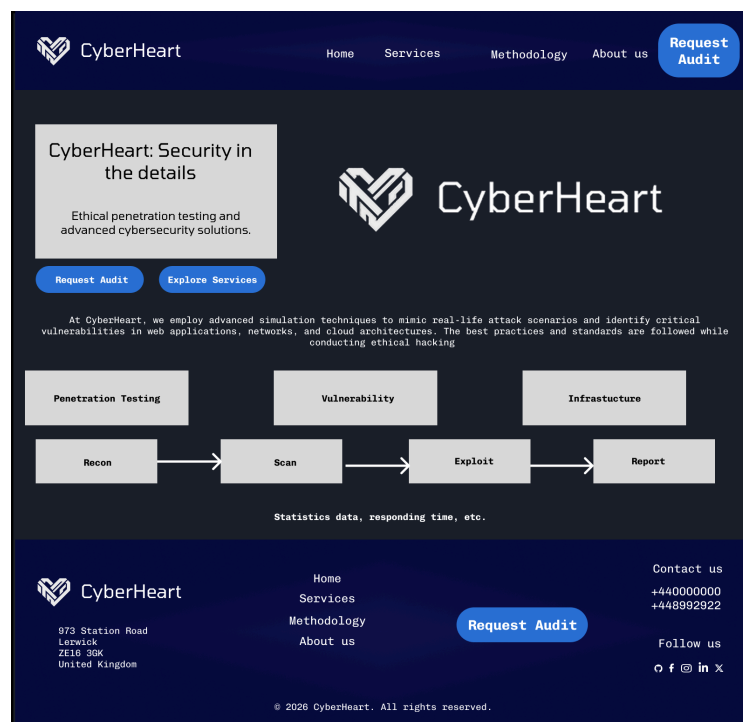


Fig. 3.1 Home Page Maket

- **Pricing**

- At first, the page was envisaged as “**Services**”, but during the development of the concept, it was decided to change it into “**Pricing**”. As a result, a more logical structuring of the content, a comparison table and the expansion of the page’s functionality were permitted.

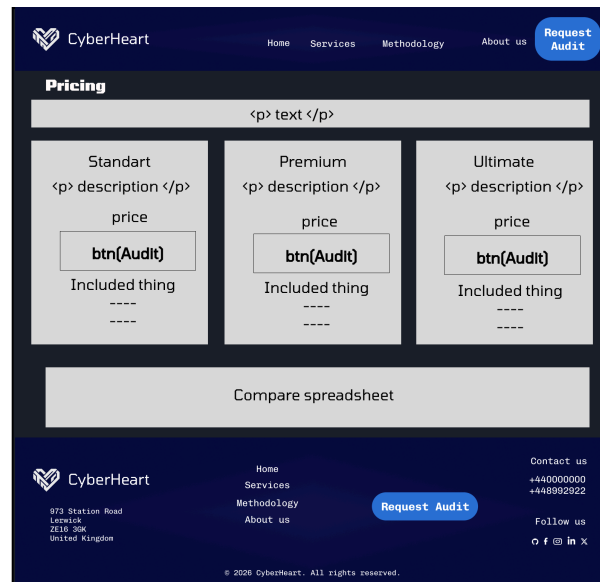


Fig. 3.2 Pricing Page Maket

- **Methodology**



Fig. 3.3 Methodology Page Maket

- **About us**

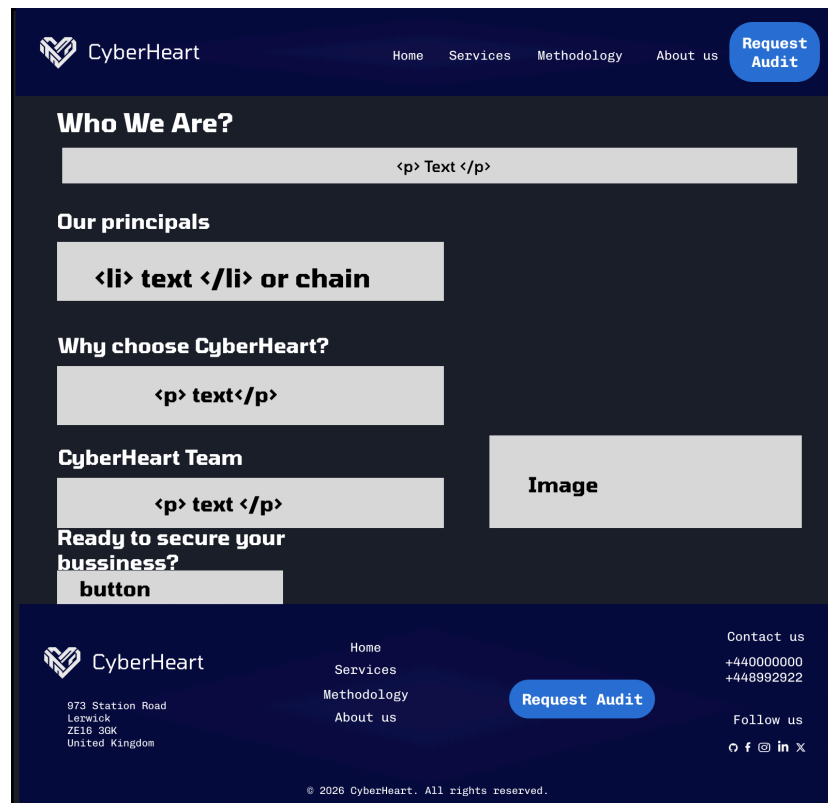


Fig. 3.4 About Us Page Maket

- **Request Audit**

Fig. 3.5 Request Audit Page Maket

- **Colour choices**

- A dark colour palette with a gradient background was selected (**#0F1115** → **#111827**). Dark colours, which symbolise technological excellence and professionalism, were the deciding factor for cybersecurity themes.

The colour `#1A1F2B` was used for the cards so that the background would look different. While the buttons were assigned the colour `#0059FF`, a similar but lighter colour `#66B2FF` was used for relevant elements like headings to make the text appear clearer. This also creates a visual hierarchy which captures the user's attention.

- **Fonts**
 - Utilising a single font family, with multiple sizes, gives the website a consistent and readable design. Additionally, “**Tomorrow**” is a modern and very technological typeface that fits the website theme and has great suitability.
- **Content decisions**
 - The content was arranged in blocks to enhance perception. Lists, short paragraphs, and cards were used. The **home page** introduces the company, whereas the **pricing page** compares packages and the **methodology page** describes how the process works. Lastly, the **Request Audit** page collects the required data.

The design of the page was sketched before the actual coding began, which means that functionality and aesthetic look are properly synchronised before the launch of a website.

3.2. Produce at least one web page.

- **The main structure of the document**
 - Every page starts with a semantic container `<main>`, which holds a document's main content. The `<main>` element highlights the dominant section in a document, which improves accessibility through assistive technologies accessing the foremost area of a webpage.
 - The content is organised by means of `<section>`, which allows logical divisions of pages in thematic blocks. It allows better reading of the code and project maintenance. Wherever possible, semantic elements are used instead of unnecessary `<div>` elements.
- **Header, Navigation and Footer**
 - The logo and navigation are located in the `<header>` and `<footer>` area made of `<nav>` and `<a>` links. The logo is both an image and a link, and a style is applied that dynamically changes the colour of the image **filter: brightness saturate invert sepia saturate hue-rotate brightness contrast**. The navigation is built using relative paths so that they move correctly between.
 - Grid was chosen as the main instrument for organising elements. It provided more accurate control of columns and their distribution and aided in creating more complex grids.
- **Pricing Blocks**
 - The Pricing page has tariff cards. Every card contains an `<h2>` heading, a `<p>` description, a price, and an `` list and is a well-structured container.

- In CSS, the display: **grid** with the **grid-template-rows** property is used to control the **vertical placement** of items. Using **1fr** is particularly useful for the alignment of components like price and button. If a card has the maximum text content, then **1fr** will stretch the rest of the cards to have equal height.
- **Comparing table**
 - The HTML **<table>** markup was never used; rather, a grid has been implemented. This approach aimed to gain better control over styles and make it flexible.
 - **grid-template-columns** adjusts the **width of the columns**, whereas the identical structure of the rows ensures visual symmetry.
- **Request Audit**
 - This input is optional in this form due to the default validation working under the hood. The form structure utilises tags such as **<form>**, **<input>**, **<select>**, and others.
 - Using a two-column CSS grid, the form fields are structured clearly and neatly. This solution will give a professional impression and bring a connection to the elements. Using a grid enables you to control the distance between the blocks, their positioning and how the space is distributed. The form is compacted to fit on standard desktop screens by using a two-column structure.

3.3. Test and evaluate the test page using at least one web browser.

- **Header**

<i>Header Test Table</i>				
Test ID	Test Case Name	Expected Result	Actual Result	Pass/Fail
CH001	Logo link	Opens Home page	Opens Home page	Pass
CH002	Home link	Opens Home page	Opens Home page	Pass
CH003	Pricink link	Opens Pricing page	Opens Pricing page	Pass
CH004	Methodology link	Opens Methodology page	Opens Methodology page	Pass
CH005	About Us link	Opens About Us page	Opens About Us page	Pass
CH006	Request Audit link	Open Request Audit page	Open Request Audit page	Pass

- **Footer**

<i>Footer Test Table</i>				
Test ID	Test Case Name	Expected Result	Actual Result	Pass/Fail
CH007	Home link	Opens Home page	Opens Home page	Pass
CH008	Pricink link	Opens Pricing page	Opens Pricing page	Pass

CH009	Methodology link	Opens Methodology page	Opens Methodology page	Pass
CH010	About Us link	Opens About Us page	Opens About Us page	Pass
CH011	Request Audit link	Open Request Audit page	Open Request Audit page	Pass
CH012	GitHub link	Open GitHub page	Open GitHub page	Pass
CH013	Facebook Link	Open Facebook page	Open Facebook page	Pass
CH014	Instagram Link	Open Instagram page	Open Instagram page	Pass
CH015	LinkedIn Link	Open LinkedIn page	Open LinkedIn page	Pass
CH016	X Link	Open X page	Open X page	Pass

- Pages

<i>Pages Test Table</i>				
Test ID	Test Case Name	Expected Result	Actual Result	Pass/Fail
CH017	Home "Request Audit" link	Open Request Audit page	Open Request Audit page	Pass
CH018	Home "Explore Pricing" link	Opens Pricing page	Opens Pricing page	Pass
CH019	Pricing Standard "Request Audit"	Opens Pricing page	Opens Pricing page	Pass
CH020	Pricing Premium "Request Audit"	Opens Pricing page	Opens Pricing page	Pass
CH021	Pricing Ultimate "Request Audit"	Opens Pricing page	Opens Pricing page	Pass
CH022	About Us "Request Audit" link	Open Request Audit page	Open Request Audit page	Pass
CH023	Request Audit "Submit Request" button	Clear Form	Clear Form	Pass

- Evaluation

- The primary goals of structural organisation, semantic markup and application of modern CSS methods were satisfied during the website development. The project primarily has logical construction of page architecture, consistent website navigation and a single visual style. Using CSS Grid, it was possible to create blocks of structure, especially on the Pricing page, where the tariff cards as well as the comparison table were systematically and predictably aligned. The usage of semantic tags made the code readable and followed

modern development practices.

The project has some limitations as well. The mobile design was not completely responsive as the focus was on demonstrating the structure, logic & layout first. Regulating vertical space length was at times tough while developing the text and block alignment under the Grid. Several solutions were put into place to solve the problem, but further optimisation would enhance layout flexibility and display versatility with more devices.

○

4. Understand how to make a web page 'live' on the internet.

4.1. Explain the stages involved in making a web page available for general viewing across the internet.

The website was published on the internet using the GitHub Pages platform. Through this method, you can host a static website on the internet for free and make it public through a browser.

1. Download Git

2. Creating a repository

- a. Initially, a new repository was made on **GitHub**. An example of a repository in everyday use is a Gmail account. Post repository creation, the local project was linked with a remote repository using the **VS Code terminal**.
- b. Git initialisation command in the project folder:
 - i. `git init`
- c. Adding all project files:
 - i. `git add`
- d. Creating the first commit:
 - i. `git commit -m "Initial website commit"`

3. Connecting to a remote repository

- a. Next, the remote GitHub repository address was added:
 - i. `git remote add origin https://github.com/22dvv04/cybersecurity.git`
- b. Then, the files were sent to the main branch:
 - i. `git branch -M main`
`git push -u origin main`
- c. After executing these commands, the project files were uploaded to **GitHub**.

4. Activation of GitHub Pages

- a. Within the settings of the repository, specifically in the Pages section, the main branch was chosen as the source of publication. GitHub generated a public link like this after the settings were saved.

- i. <https://22dvv04.github.io/CyberHeart/>

5. Functionality check

- a. After publication, the site was tested in a browser to check:
 - i. the correctness of links
 - ii. image loading
 - iii. CSS display
 - iv. the absence of errors in the console

The stages involved in publishing the website were local project preparation, uploading via git and, activating GitHub Pages, and final testing. Using this method, one can very quickly push code to GitHub from a Visual Studio Code (VS Code) environment. This shows an understanding of how to host, control versions and publish via the Internet.

References

Comeau, J. W. (2023). *An Interactive Guide to CSS Grid* • Josh W. Comeau.

Joshwcomeau.com. <https://www.joshwcomeau.com/css/interactive-guide-to-grid/>

Google Developers. (2019). *A new image format for the Web | WebP | Google*

Developers. Google Developers. <https://developers.google.com/speed/webp>

Hay, J. (2023, January 11). *Understanding The Value Of Effective Website Navigation*.

[Www.hellodigital.marketing](http://www.hellodigital.marketing).

<https://www.hellodigital.marketing/learn/value-effective-website-navigation/>

Laurinavicius, T. (2017, March 9). *How To Create a Web Design Style Guide*. Designmodo.

<https://designmodo.com/create-style-guides/>

MDN. (2025, November 7). *filter - CSS | MDN*. Mozilla.org.

<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference/Properties/filter>

O'Hara, S. (2022, January 20). *Div divisiveness | scotthara.me*. Scotthara.me.

<https://www.scotthara.me/blog/2022/01/20/divisive.html>

Web.Dev. (2023, November 1). *Image performance*. Web.dev.

<https://web.dev/learn/performance/image-performance>

Yau, M. (2025). *Clear Digital Pathways: Mastering Headings, Buttons & Navigation* –

CultureSuite Article. Culturesuite.co.

<https://www.culturesuite.co/resources/articles/clear-digital-pathways-mastering-headings-buttons-navigation>