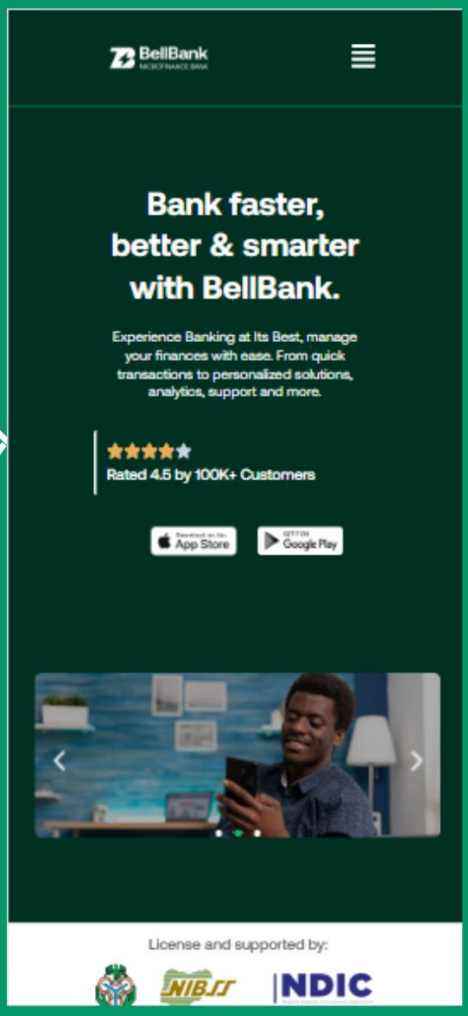
Learn Web Design With Easy



Author: Team Piccolo & Malamiromba

First Edition 2024

Table of Contents

Table of Figures 4
Chapter One: Introduction to Web Development 10
The Internet 10
Web page 10
Website 10
Web server 11
Domain Name 11
Chapter Two: HTML and CSS 12
Introduction to HTML and CSS 12
HTML Element 13
HTML Block and Inline Elements 14
Exercise 15
HTML Attributes 16
HTML Head Element 16
HTML Title Element 16
HTML Favicon 18
HTML Meta Element 18
HTML Style Element 19
HTML Link Element 20
HTML Script Element 20
Exercise 20
Styling HTML with CSS 21
Inline CSS 21
Internal CSS 21
External CSS 22
HTML Body Element 23
Basic sections of a document 23
Header Element <header> 23
Navigation Bar Element <nav> 23
Main Content Element <main> 23
Section Element <section> 23
Sidebar Element <aside> 24

Footer Element <footer> 24
HTML Image Element <img> 25
HTML id and class attributes 25
HTML List Element 27
HTML Link Element 28
CSS Introduction 31
CSS Syntax 31
CSS Background 32
CSS Height and Width 34
CSS Display 35
CSS Flexbox 35
CSS List 38
CSS Colors 39
CSS Margin 40
CSS Padding 42
CSS Border 44
CSS Font 47
CSS Google Font 49
CSS Text 51
Adding Icons with Font Awesome 54
Styling Hero section 57
Exercise 2 65
Styling Slideshow Section 66
Styling License Section 68
Styling Account Types Section 72
Styling Features Section 79
Styling Debit Card Section 81
Styling Personal Account Section 85
Styling Modular Solution Section 88
Styling Why Us Section 91
Exercise 3 (Practical) 100
Answer 100

Styling Download App Section 102
Exercise 4 (Practical) 105
Answer 106
References 107

Table of Figures

Figure 1: BellBank Website 10
Figure 2: Basic HTML structure 12
Figure 3: Basic CSS Structure 12
Figure 4: HTML Element 13
Figure 5: Empty element 13
Figure 6: Nesting HTML Elements 13
Figure 7: HTML Attribute 16
Figure 8: Head Element 16
Figure 9: Title element 17
Figure 10: Title element II 17
Figure 11: HTML Favicon 18
Figure 12: HTML Favicon II 18
Figure 13: HTML Meta 18
Figure 14: HTML Style 19
Figure 15: HTML Link 20
Figure 16: HTML Script 20
Figure 17: Inline CSS 21
Figure 18: Internal CSS 21
Figure 19: External CSS 22
Figure 20: HTML5 Semantics Standard 24
Figure 21: HTML img 25
Figure 22: HTML id attribute 25
Figure 23: CSS id selector 26
Figure 24: HTML class attribute 26
Figure 25: CSS class selector 26
Figure 26: HTML ordered list 27
Figure 27: HTML ordered list output 27
Figure 28: HTML unordered list 28
Figure 29: HTML unordered list output 28
Figure 30: HTML link element 29
Figure 31: Header section HTML Structure 30
Figure 32: CSS Syntax 31
Figure 33: CSS background-color property 32
Figure 34: CSS background-image property 32

Figure 35: CSS background-repeat property 33
Figure 36: CSS background-position property 33
Figure 37: CSS background-attachment property 33
Figure 38: CSS background shorthand property 34
Figure 39: CSS height and width property 35
Figure 40: CSS display property 35
Figure 41: Header Section 36
Figure 42: adding id attribute 36
Figure 43: CSS flex display property 37
Figure 44: Header section after add flex display property 37
Figure 45: CSS justify-content and align-items property 38
Figure 46: Header Section Display 38
Figure 47: CSS list-style-type property 38
Figure 48: List-style-type output 39
Figure 49: CSS flex-direction property 39
Figure 50: CSS flex-decoration property output 39
Figure 51: CSS color property 39
Figure 52: CSS color property output 39
Figure 53: CSS text-decoration property 40
Figure 54: CSS text-decoration property output 40
Figure 55: CSS margin property 41
Figure 56: CSS margin property output 41
Figure 57: CSS margin property II 41
Figure 58: CSS margin property III 42
Figure 59: CSS margin property IV 42
Figure 60: headerSection much better! 42
Figure 61: CSS padding property 43
Figure 62: CSS padding property II 43
Figure 63: CSS padding property III 44
Figure 64: headerSection much better! II 44
Figure 65: CSS border-style property 45
Figure 66: CSS border-width property 45
Figure 67: CSS border-color property 46
Figure 68: CSS border property 46
Figure 69: CSS border-bottom property 46
Figure 70: headerSection much better! III 47
Figure 71: CSS font-family property 47
Figure 72: CSS font-style property 48
Figure 73: CSS font-size property 48
Figure 74: CSS font-weight property 48
Figure 75: CSS font-variant property 49
Figure 76: CSS Google Font 49
Figure 77: CSS Google Font II 49

Figure 78: CSS Google Font III 50
Figure 79: CSS Google Font IV 50
Figure 80: CSS Google Font V 50
Figure 81: CSS Google Font VI 51
Figure 82: headerSection much better! IV 51
Figure 83: CSS Text property 52
Figure 84: main element 53
Figure 85: section element 53
Figure 86: working on slogan div 53
Figure 87: Slogan div output 54
Figure 88: Download Link div content 54
Figure 89: adding fontawesome icons 54
Figure 90: embedding fontawesome link 55
Figure 91: fontawesome website 55
Figure 92: searching on fontawesome 55
Figure 93: searching on fontawesome II 56
Figure 94: Adding the star icons using fontawesome 56
Figure 95: Adding the star icons using fontawesome II 57
Figure 96: Slogan and download link section 57
Figure 97: adding id attribute to the section element 57
Figure 98: CSS display property (grid) 58
Figure 99: CSS display property (grid) II 58
Figure 100: Styling Slogan div 58
Figure 101: Slogan stylesheet 59
Figure 102: Slogan div output 59
Figure 103: heroMsg div 59
Figure 104: heroMsg div CSS properties 60
Figure 105: heroMsg div output 60
Figure 106: starDiv code snippet 60
Figure 107: starDiv CSS properties 61
Figure 108: starDiv style output 61
Figure 109: adding yellow_star class to the star icons 61
Figure 110: adding the CSS properties to the yellow_star class 62
Figure 111: yellow_star class display output 62
Figure 112: adding downloadIcon class to the download icons 62
Figure 113: Adding CSS properties to the downloadIcon class 62
Figure 114: downloadIcon output 63
Figure 115: CSS align-items property 63
Figure 116: heroSection update output 63
Figure 117: adding downloadLink ID attribute 64
Figure 118: downloadLink CSS properties 64
Figure 119: Complete HeroSection design 65
Figure 120: creating slideShowSection element 66

Figure 121: creating slideShowSection element II 66
Figure 122: creating slideShowSection element III 67
Figure 123: creating slideShowSection element IV 67
Figure 124: creating slideShowSection element V 67
Figure 125: Styling License Section 68
Figure 126: Styling License Section II 68
Figure 127: Styling License Section II 69
Figure 128: Styling License Section II 69
Figure 129: Styling License Section II 69
Figure 130: Styling License Section III 70
Figure 131: Styling License Section IV 70
Figure 132: Styling License Section V 71
Figure 133: Styling License Section VI 71
Figure 134: Styling License Section VII 72
Figure 135: Styling account types section 72
Figure 136: Styling account types section II 73
Figure 137: Styling account types section III 73
Figure 138: Styling account types section IV 73
Figure 139: Styling account types section V 74
Figure 140: Styling account types section VI 74
Figure 141: Styling account types section VII 75
Figure 142: Styling account types section VIII 75
Figure 143: Styling account types section IX 76
Figure 144: Styling account types section X 76
Figure 145: Styling account types section XI 77
Figure 146: Styling account types section XII 77
Figure 147: Styling account types section XIII 78
Figure 148: Styling account types section XIV 78
Figure 149: Styling account types section XV 78
Figure 150: Styling account types section XVI 79
Figure 151: Styling features section 79
Figure 152: Styling features section II 80
Figure 153: Styling features section III 80
Figure 154: Styling features section IV 81
Figure 155: Styling features section V 81
Figure 156: Styling Debit Card Section 81
Figure 157: Styling Debit Card Section II 82
Figure 158: Styling Debit Card Section II 82
Figure 159: Styling Debit Card Section III 83
Figure 160: Styling Debit Card Section III 83
Figure 161: Styling Debit Card Section IV 84
Figure 162: Styling Debit Card Section V 84
Figure 163: Styling Debit Card Section VI 85

Figure 164: Styling Personal Account Section 85
Figure 165: Styling Personal Account Section II 86
Figure 166: Styling Personal Account Section III 86
Figure 167: Styling Personal Account Section IV 87
Figure 168: Styling Personal Account Section V 87
Figure 169: Styling Modular Solution Section 88
Figure 170: Styling Modular Solution Section II 89
Figure 171: Styling Modular Solution Section III 89
Figure 172: Styling Modular Solution Section IV 90
Figure 173: Styling Modular Solution Section V 90
Figure 174: Styling Why Us Section 91
Figure 175: Styling Why Us Section II 91
Figure 176: Styling Why Us Section III 92
Figure 177: Styling Why Us Section IV 92
Figure 178: Styling Why Us Section V 93
Figure 179: Styling Why Us Section VI 93
Figure 180: Styling Why Us Section VII 94
Figure 181: Styling Why Us Section VIII 94
Figure 182: Styling Why Us Section IX 94
Figure 183: Styling Why Us Section X 95
Figure 184: Styling Why Us Section XI 95
Figure 185: Styling Why Us Section XII 95
Figure 186: Styling Why Us Section XIII 96
Figure 187: Styling Why Us Section XIV 96
Figure 188: Styling Why Us Section XV 97
Figure 189: Styling Why Us Section XVI 97
Figure 190: Styling Why Us Section XVII 97
Figure 191: Styling Why Us Section XVIII 98
Figure 192: Styling Why Us Section XIX 98
Figure 193: Styling Why Us Section XX 99
Figure 194: Styling Why Us Section XXI 99
Figure 195: Styling Why Us Section XXII 100
Figure 196: Exercise Solution I 100
Figure 197: Exercise Solution II 101
Figure 198: Exercise Solution III 101
Figure 199: Styling Why Us Section XXIII 102
Figure 200: Styling Download App Section 102
Figure 201: Styling Download App Section II 103
Figure 202: Styling Download App Section III 103
Figure 203: Styling Download App Section IV 104
Figure 204: Styling Download App Section V 104
Figure 205: Styling Download App Section VI 105
Figure 206: Styling Download App Section VII 105

Figure 207: Exercise 1 Solution..... 106
Figure 208: Exercise 1 Solution II..... 106
Figure 209: Styling Download App Section VIII 106

Chapter One: Introduction to Web Development

The Internet

The *Internet* is the backbone of the Web, the technical infrastructure that makes the Web possible. The Internet is a large network of computers which communicate all together. The various technologies that support the Internet have evolved over time, but the way it works hasn't changed that much: Internet is a way to connect computers all together and ensure that, whatever happens, they find a way to stay connected.

Web page

A *web page* is a simple document displayable by a browser. Web Pages are what make up a website. For example, on www.bellmfb.com, you can find the home page, about us page, personal page, business page, etc. Such documents are written in the HTML language. All web pages available on the web are reachable through a unique address. To access a page, just type its address (domain name) in your browser address search bar as shown below:

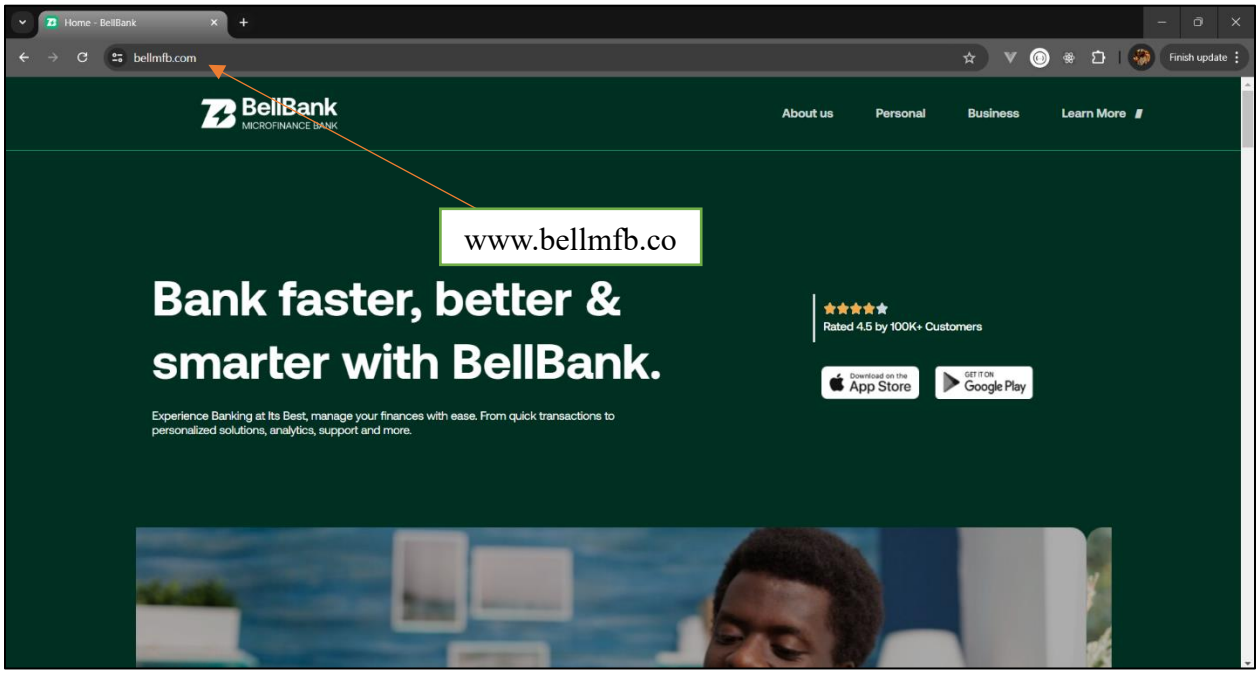


Figure 1: BellBank Website

Website

A *Website* is a collection of related web pages located under a single domain name. When you combine the home page, about us page, personal page, business page, etc. and store them under a

single domain name www.bellmfb.com. A website is a collection of linked web pages that share a unique domain name. Each web page of a given website provides links, most of the time in the form of clickable portions of text, that allow the user to move from one page of the website to another. To access a website, type its domain name in your browser search bar, and the browser will display the website's main web page, or homepage as shown in the figure 1 above.

Web server

A *web server* is a computer hosting one or more *websites*. "Hosting" means that all the *web pages* and their supporting files are available on that computer. The *web server* will send any *web page* from the *website* it is hosting to any user's browser when the user sends a request.

Domain Name

Domain names are a key part of the Internet infrastructure. They provide a human-readable address for any *web server* available on the Internet. Any Internet-connected computer can be reached through a public IP Address, either an IPv4 address (e.g. *104.21.35.168*) or an IPv6 address (*2606:4700:3037::6815:23a8*). Computers can handle such addresses easily, but people have a hard time finding out who is running the server or what service the website offers. IP addresses are hard to remember and might change over time. To solve all those problems, we use human-readable addresses called *domain names* for example www.bellmfb.com.

Footnote: *You cannot "buy a domain name", instead, you pay for the right to use a domain name for one or more years. You can renew your right, and your renewal has priority over other people's applications. But you never own the domain name.*

Chapter Two: HTML and CSS

Introduction to HTML and CSS

HTML stands for Hyper Text Markup Language. HTML is a *markup language* that defines the structure of your content. HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way.

```html
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home - Bellbank</title>
</head>
<body>
    Bell Microfinance Bank
</body>
</html>
```

Figure 2: Basic HTML structure

CSS stands for Cascading Style Sheets. CSS is used to styles web content. CSS describes how HTML elements are to be displayed on the screen, paper or in other media.

```css
body{
    background-color: green;
    font-size: 20px;
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
    margin: 0;
    padding: 0;
}

```

Figure 3: Basic CSS Structure

Footnote: *You save an HTML file with the extension of either .html or .htm but use .html as most developers use that. You save a CSS file with the extension of .css*

HTML Element

An HTML element is an individual component of an HTML document. It represents semantics, or meaning. For example, the title element represents the title of the document as shown above.

An HTML element is defined by a start tag, some content, and an end tag.

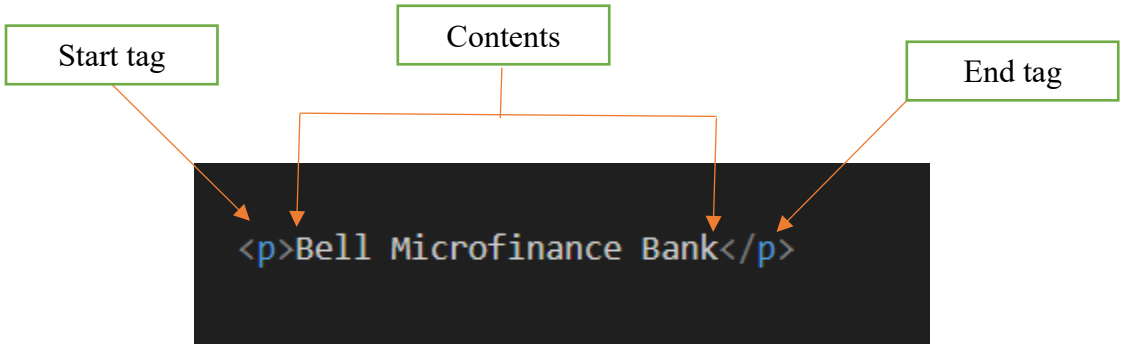


Figure 4: HTML Element

Note: Not all HTML element require the start tag and the end tag or close. They have no content and called **empty elements** example like *img, br, hr, etc.*

```
<img src="https://bellmfb.com/wp-content/uploads/2024/07/Bellbank-MFB-logo-white-300x68.png" alt="Bellbank Logo">
```

Figure 5: Empty element

HTML elements can be nested (this means that elements can contain other elements). All HTML documents consist of nested HTML elements. In fig. below, the *title* element is nested inside the *head* element and the *head* element is nested inside the *html* element.

```
1   <html lang="en">
2   <head>
3       <meta charset="UTF-8">
4       <meta name="viewport" content="width=device-width, initial-scale=1.0">
5       <title>Home - Bellbank</title>
6   </head>
7   <body>
8       Bell Microfinance Bank
9   </body>
10  </html>
```

Figure 6: Nesting HTML Elements

HTML Block and Inline Elements

Every HTML element has a default display value, depending on what type of element it is.

A **block-level** element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can). The <*div*> element is the widely used block level element. Below are examples of block level elements in HTML:

<address>	<article>	<aside>	<blockquote>	<canvas>	<dd>
	<dl>	<dt>	<fieldset>	<figcaption>	<figure>
<footer>	<form>	<h1>-<h6>	<header>	<hr>	
<main>	<nav>	<noscript>		<p>	<pre>
<section>	<table>	<tfoot>		<video	

An **inline element** does not start on a new line and only takes up as much width as necessary. The <*span*> element is the widely used inline level element. Below are examples of inline level elements in HTML:

<a>	<abbr>	<acronym>	<b>	<bdo>	<big>
 	<button>	<cite>	<code>	<dfn>	
<i>	<img>	<input>	<kbd>	<label>	<map>
<object>	<output>	<q>	<samp>	<script>	<select>
<small>			<sub>	<sup>	<textarea>
<time>	<tt>	<var>			

Exercise

1) HTML elements are used to style the website.
   a) True
   b) False
2) You don't have to nest your HTML element.
   a) Agree
   b) Neutral
   c) Disagree
3) You can nest the *title* element within the *body* element.
   a) Yes
   b) No
4) HTML elements can have attributes.
   a) True
   b) False
5) Choice the correct syntax and most appropriate syntax
   a) <img src="myphoto.png" alt="my-photo"></img>
   b) <img src="myphoto.png"></img>
   c) <img src="myphoto.png">
   d) <img src="myphoto.png" alt="my-photo"></img>

Footnote:

1) *HTML elements are not case sensitive <P> means the same as <p>. The HTML standard does not require lowercase tags, but W3C recommends lowercase in HTML.*
2) *Never skip the end tag of an HTML element. Some HTML elements will display correctly, even if you forget the end tag. However, never rely on this! Unexpected results and errors may occur.*

HTML Attributes

All HTML elements can have attributes. Attributes provide additional information about elements. Attributes are always specified in the start tag.

```
<img src="https://bellmfb.com/wp-content/uploads/2024/07/Bellbank-MFB-logo-white-300x68.png" alt="Bellbank Logo">
```

Figure 7: HTML Attribute

The <*img*> tag is used to embed an image in an HTML page. The *src* attribute specifies the path to the image to be displayed. We will learn more about images in HTML later, smile!

The *alt* attribute for the <*img*> tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the *src* attribute, or if the user uses a screen reader.

HTML Head Element

The <*head*> element is a container for metadata (data about data) and is placed between the <*html*> tag and the <*body*> tag.

```
4
5   <html lang="en">
6       <head></head>
7       <body></body>
8   </html>
```

Figure 8: Head Element

HTML metadata is data about the HTML document. Metadata is not displayed. Metadata typically define the document title, character set, styles, scripts, and other meta information.

HTML Title Element

The <*title*> element defines the title of the document. The title must be text-only, and it is shown in the browser's title bar or in the page's tab as shown in the figure below. The <*title*> element is required in HTML documents. The content of a page title is very important for search engine

optimization (SEO). The page title is used by search engine algorithms to decide the order when listing pages in search results.

```
<html lang="en">
    <head>
        <title>Home - BellBank</title>
    </head>
    <body>

    </body>
</html>
```

Figure 9: Title element

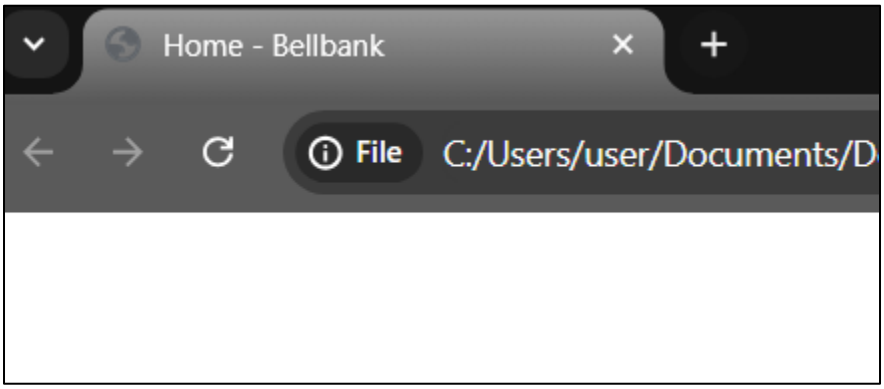


Figure 10: Title element II

HTML Favicon

A favicon is a small image displayed next to the page title in the browser tab. You can use any image you like as your favicon.

To add a favicon to your page, use the <*link*> element with HTML attributes of *rel, type* and *href* as shown below.

```
<html lang="en">
    <head>
        <title>Home - BellBank</title>
        <link rel="icon" type="image/x-icon" href="./images/favicon.png">
    </head>
    <body>

    </body>
</html>
```

Figure 11: HTML Favicon

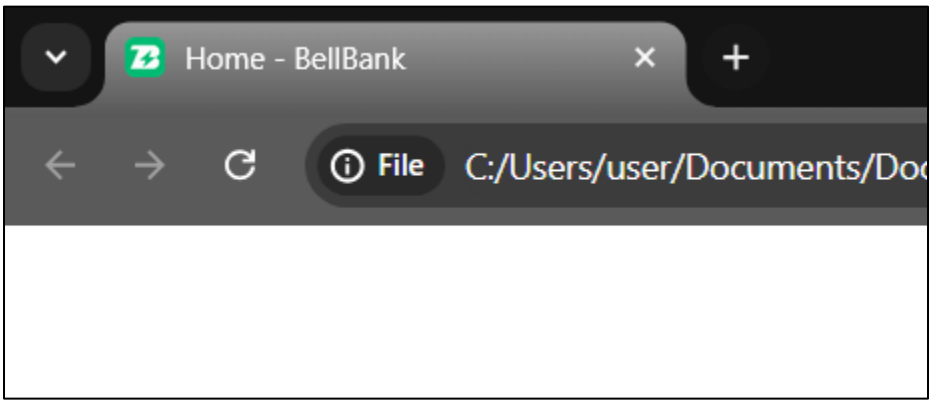


Figure 12: HTML Favicon II

HTML Meta Element

The <*meta*> element is typically used to specify the character set, viewport settings, author, keywords and description.

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="author" content="BellBank">
    <meta name="keywords" content="Banking, Savings, Microfinance">
    <meta name="description" content="Experience Banking at Its Best, manage your finances with ease.">
</head>
```

Figure 13: HTML Meta

The metadata will not be displayed on the page, but is used by browsers (how to display content or reload page), by search engines (keywords), and other web services.

HTML Style Element

The *<style>* element is used to define style information for a single HTML page or in internal css style. You can style a web page either using in-line CSS, internal CSS and external CSS. We will learn more about these methods of styling later.

```
    <style>
        body{
            background-color: #003022;
        }
    </style>
</head>
```

Figure 14: HTML Style

Footnote:

When you add the <style> element, you first write the CSS property you want to apply to the element then a colon then the CSS value then you terminate it by using a semi-colon. We will learn more about CSS later in this book.

HTML Link Element

The <*link*> element defines the relationship between the current document and an external resource. The <*link*> tag is most often used to link to external style sheets.

```
    <link rel="stylesheet" href="./css/style.css">
</head>
```

Figure 15: HTML Link

HTML Script Element

The <*script*> element is used to JavaScript to an HTML web page. We will learn deep about JavaScript later in this book.

```
    <script>
        alert('Welcome to Bellbank')
    </script>
</head>
```

Figure 16: HTML Script

Exercise

1) The <*base*> element is an HTML element.
   a) True
   b) False
2) You can add only one CSS property to an HTML element.
   a) Yes
   b) No
3) Search Engine Optimization (SEO) is important when creating a web page.
   a) Agree
   b) Disagree

Styling HTML with CSS

We have started learning about HTML and have covered some of its elements. Now, let's begin learning about CSS and how to add it to an HTML document. CSS can be added to an HTML document in three ways: inline CSS, internal CSS, and external CSS.

Inline CSS

An inline CSS is used to apply a unique style to a single HTML element. An inline CSS uses the *style* attribute of an HTML element. The following example below sets the font-size of the element to 24px, and text-color to white:

```
<h1 style="font-size: 24px; color: white;">BellBank</h1>
```

Figure 17: Inline CSS

Internal CSS

An internal CSS is used to define a style for a single HTML page. An internal CSS is defined in the <*head*> section of an HTML page, within a <*style*> element. The following example below sets the font-size of ALL the <p> elements (on that page) to 12px, and the font-family of **ALL** the <*p*> elements to Arial, if Arial is not installed on the system; it will set the font-family to Helvetica, if Helvetica not found then it will be set to sans-serif.

```
<head>
    <style>
        p{
            font-size: 12px;
            font-family: Arial, Helvetica, sans-serif;
        }
    </style>
</head>
```

Figure 18: Internal CSS

External CSS

An external style sheet is used to define the style for many HTML pages. To use an external style sheet, add a link to it in the <*head*> section of each HTML page.

```
<head>
    <link rel="stylesheet" href="./css/style.css">
</head>
```

Figure 19: External CSS

The external style sheet can be written in any text editor like Visual Studio Code (VSCode), Sublime, notepad++, etc. The file must not contain any HTML code, and must be saved with a .css extension.

Footnote:

Summary

- When you want to style a particular HTML element on a page; the Inline CSS is the best approach to use.
- When you want to style a particular Web Page on your Website; the Internal CSS is the best approach to use.
- When you want to style the entire website; the External CSS is the best approach to use.
- With an external style sheet, you can change the look of an entire web site, by changing one file.

Recommendation

- When you want to style an HTML element, a web page or the entire website; the best approach is to use the External Styling, because it makes your code much easier to maintain and it is the standard most developers used.

HTML Body Element

The <*body*> tag defines the document's body. The <*body*> element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

Basic sections of a document

Webpages can and will look pretty different from one another, but they all tend to share similar standard components, unless the page is displaying a fullscreen video or game, is part of some kind of art project, or is just badly structured:

Header Element <header>

The <*header*> element in HTML represents a container for introductory or navigational content, usually located at the top of a webpage or section. This element is often used for branding, primary navigation, and other elements like the site logo, search bar, or introductory text. This usually stays the same from one webpage to another.

Navigation Bar Element <nav>

The <*nav*> element in HTML is used to define a block of navigation links on a webpage. It helps to semantically organize the links that guide users to different sections of a site or related web pages, making it easier for both users and search engines to understand the structure and navigation options of a website. Having inconsistent navigation on your website will just lead to confused, frustrated users and poor user experience.

Main Content Element <main>

The <*main*> element in HTML is used to designate the primary content of a webpage, containing information that is directly related to the main purpose of the page. It's designed to hold the content unique to that page, excluding repetitive elements like headers, footers and navigation bars that are common across multiple pages. This is the one part of the website that definitely will vary from page to page.

Section Element <section>

The <*section*> element in HTML is used to define a distinct area or block within a webpage that groups together related content. It provides a way to structure content semantically, making it easier for both users and search engines to understand the context of each part of a webpage.

Sidebar Element <aside>

The <*aside*> element in HTML often contains supplemental information, such as sidebars, quotes, ads, or other related content, but isn't crucial to the main subject of the page. Some peripheral info, links, quotes, ads, etc.

Footer Element <footer>

The <*footer*> element in HTML is used to define the footer section of a webpage or a section within it. This section generally contains metadata or related information, like copyright notices, contact details, links to privacy policies, or author information. It's typically located at the bottom of the page or section and serves as a concluding element. The footer is also sometimes used for SEO purposes, by providing links for quick access to popular content.

```
<body>
    <header></header>
    <nav></nav>
    <main>
        <section></section>
        <aside></aside>
    </main>
    <footer></footer>
</body>
```

Figure 20: HTML5 Semantics Standard

Footnote:

There can only be one <body> element in an HTML document.

HTML Image Element <img>

High resolution images improve the design and appearance of your website. They are a lot of free websites on the internet that you can download high resolution images for free. One of my favorite websites is **pixabay.com**.

In HTML, images are defined with the <*img*> tag. The <*img*> tag is empty, it contains attributes only, and does not have a closing tag.

```
<body>
    <header>
        <img src="./images/logo.png" alt="Website Logo">
    </header>
```

Figure 21: HTML img

The *src* attribute specifies the URL (web address) of the image as in figure above. The *alt* attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader). The value of the alt attribute should describe the image.

We have learned how to add CSS to an HTML document. Now, let's start learning about CSS properties. In the meantime, let's learn about the *id* and *class* attributes.

HTML id and class attributes

The HTML *id* attribute is used to specify a unique id for an HTML element (the value must be unique within the HTML document).

```
<body>
    <header>
        <img id="logo" src="./images/logo.png" alt="Website Logo">
    </header>
```

Figure 22: HTML id attribute

The *id* attribute is used in CSS or JavaScript to perform certain tasks for the element with the specific *id* value.

In CSS, to select an element with a specific id, write a hash (#) character, followed by the id value of the element as shown below:

```
#logo{
    width: 180px;
}
```

Figure 23: CSS id selector

The HTML *class* attribute is used to define equal styles for elements with the same class name. All HTML elements with the same class attribute will get the same style.

```
<div class="notes">
    The HTML id attribute is used to specify a unique id for an HTML element
    (the value must be unique within the HTML document).
</div>

<div class="notes">
    The HTML class attribute is used to define equal styles for
    elements with the same class name.
    All HTML elements with the same class attribute will get the same style.
</div>
```

Figure 24: HTML class attribute

```
.notes{
    font-size: 12px;
    font-family: 'Courier New', Courier, monospace;
}
```

Figure 25: CSS class selector

Footnote:

- The image can be of any image type example portable network graphic (PNG) etc.
- A screen reader is a software program that reads the HTML code, converts the text, and allows the user to "listen" to the content. Screen readers are useful for people who are visually impaired or learning disabled.
- An HTML element can have more than one class.

HTML List Element

In HTML, we have mainly two (2) types of lists. Ordered list and unordered list. HTML lists allow web authors to group a set of related items in lists.

An **ordered** list starts with the <*ol*> tag. Each list item starts with the <*li*> tag. The list items will be marked with numbers by default as shown below:

```
<nav>
    <ol>
        <li>About us</li>
        <li>Personal</li>
        <li>Business</li>
        <li>Learn More</li>
    </ol>
</nav>
```

Figure 26: HTML ordered list

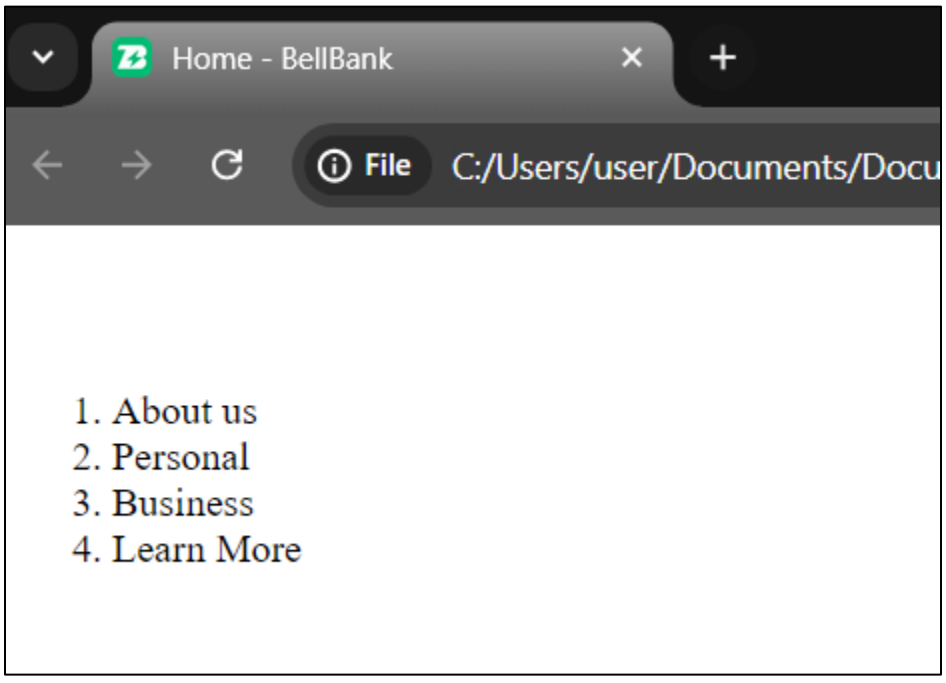


Figure 27: HTML ordered list output

An **unordered** list starts with the <*ul*> tag. Each list item starts with the <*li*> tag. The list items will be marked with bullets (small black circles) by default as shown below:

```
<nav>
    <ul>
        <li>About us</li>
        <li>Personal</li>
        <li>Business</li>
        <li>Learn More</li>
    </ul>
</nav>
```

Figure 28: HTML unordered list

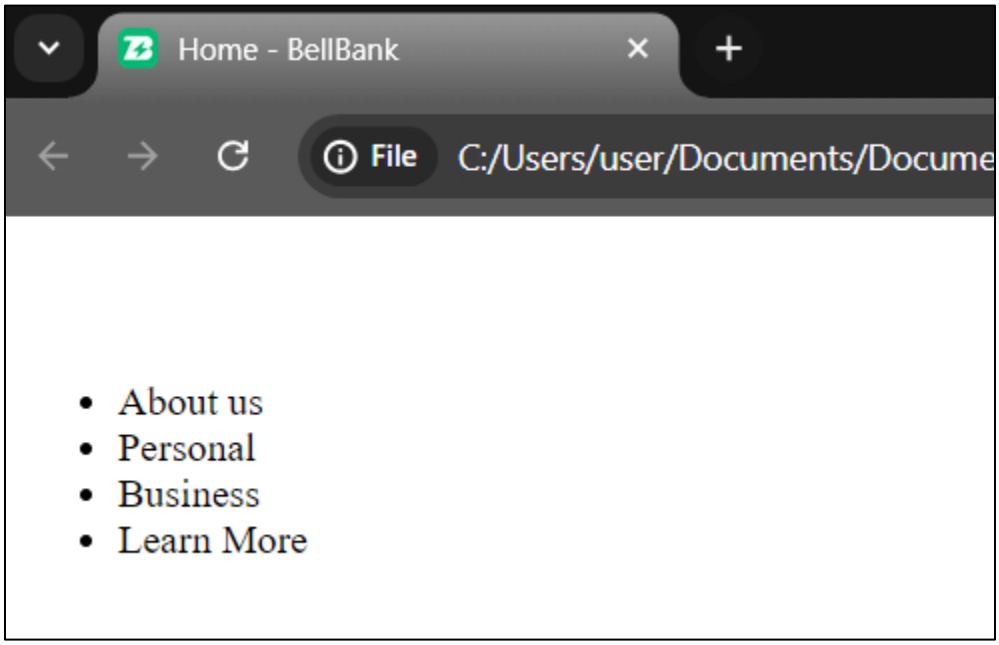


Figure 29: HTML unordered list output

HTML Link Element

Links are found in nearly all web pages. Links allow users to click their way from page to page. HTML links are hyperlinks.

You can click on a link and jump to another document or to even another website.

When you move the mouse over a link, the mouse arrow will turn into a little hand. The HTML *<a>* tag defines a hyperlink as shown below:

```
<header>
    <a href="index.html" target="_self">
        <img id="logo" src="./images/logo.png" alt="Website Logo">
    </a>
</header>
```

Figure 30: HTML link element

The *href* attribute is the most important attribute of the <a> element is the href attribute, which indicates the link's destination. The link text is the part that will be visible to the reader. Clicking on the link text, will send the reader to the specified URL address as shown above (HTML links)

The *target* attribute specifies where to open the linked document. The target attribute can have one of the following values:

- _self - Default. Opens the document in the same window/tab as it was clicked
- _blank - Opens the document in a new window or tab
- _parent - Opens the document in the parent frame
- _top - Opens the document in the full body of the window

Footnote:

- You can use a text, an Image or a Button as the link.
- You can link to a document like a Portal Document Format (PDF)

```
<!-- Header Section  -->
<div>
    <header>
        <a href="index.html" target="_self">
            <img id="logo" src="./images/logo.png" alt="Website Logo">
        </a>
    </header>
    <nav>
        <ul>
            <li>
                <a href="aboutus.html">About us</a>
            </li>
            <li>
                <a href="personal.html">Personal</a>
            </li>
            <li>
                <a href="business.html">Business</a>
            </li>
            <li>
                <a href="business.html">Learn More</a>
            </li>
        </ul>
    </nav>
</div>
<!-- End of Header Section  -->
```

Figure 31: Header section HTML Structure

CSS Introduction

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media.

CSS saves a lot of work. It can control the layout of multiple web pages all at once. CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

With CSS, you can design styles for different screen size like tablet, iPad, iPhone, Android, desktop, laptop etc. When the website is visited using a laptop, the display varies when visited on an iPhone or a tablet. It is called responsive web design and we will learn about it later in this book, Smile!

The style definitions are normally saved in external .css files.

CSS Syntax

A CSS rule-set consists of a selector and a declaration block as shown below:

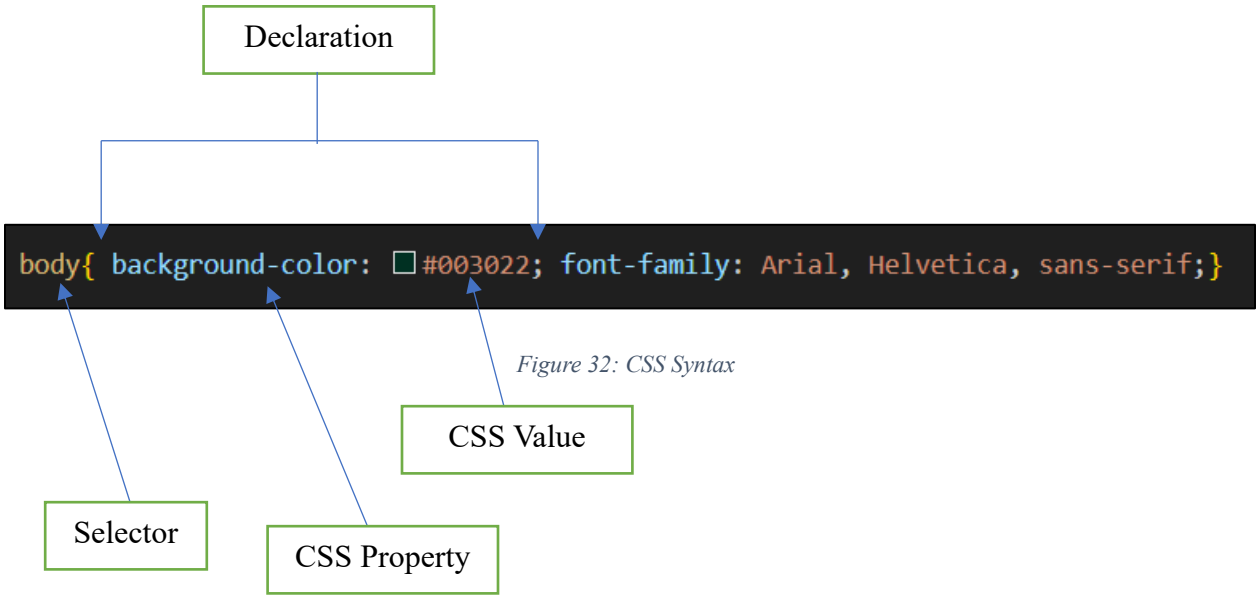


Figure 32: CSS Syntax

The selector points to the HTML element you want to style. The declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, separated by a colon. Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

CSS Background

The CSS background properties are used to add background effects for elements.

The *background-color* property specifies the background color of an element. A color is most often specified by:

- a valid color name - like “black”
- a HEX value - like “#003022”
- an RGB value - like “rgb(0,0,0)”

```
body{
    background-color: #003022;
}
```

Figure 33: CSS background-color property

Any of the three (3) methods you decide to choose will work just fine. Whether using a valid color name like “black” or using the hexadecimal value like “#003022” or using RGB value like “rgb (0, 0, 0)”

Note: RGB stands for Red, Green and Blue.

The *background-image* property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element. The background image for a page can be set as shown below:

```
body{
    background-image: url('bgcover.png');
}
```

Figure 34: CSS background-image property

By default, the *background-image* property repeats an image both horizontally and vertically. You can alter this behavior. If you want the image to repeat horizontally only, there is a property called *background-repeat* with value of repeat-x. If you want the image to repeat vertically,

change the value to repeat-y and if you don't want the image to repeat, use the value of no-repeat as shown below:

```
body{
    background-image: url('bgcover.png');
    background-repeat: no-repeat;
}
```

Figure 35: CSS background-repeat property

In some scenarios, the background image might disturb the text on the website or make them not visible. You can alter the position of the image using the *background-position* property. You can get it a value of "right top", "right bottom", "left top" or "right top" depending on the position you want to the image to appear as shown below:

```
body{
    background-image: url('bgcover.png');
    background-repeat: no-repeat;
    background-position: right top;
}
```

Figure 36: CSS background-position property

To specify that the background image should be fixed (will not scroll with the rest of the page), use the *background-attachment* property as shown below:

```
body{
    background-image: url('bgcover.png');
    background-repeat: no-repeat;
    background-position: right top;
    background-attachment: fixed;
}
```

Figure 37: CSS background-attachment property

To shorten the code, it is also possible to specify all the background properties in one single property. This is called a shorthand property. The shorthand property for background is background as shown below:

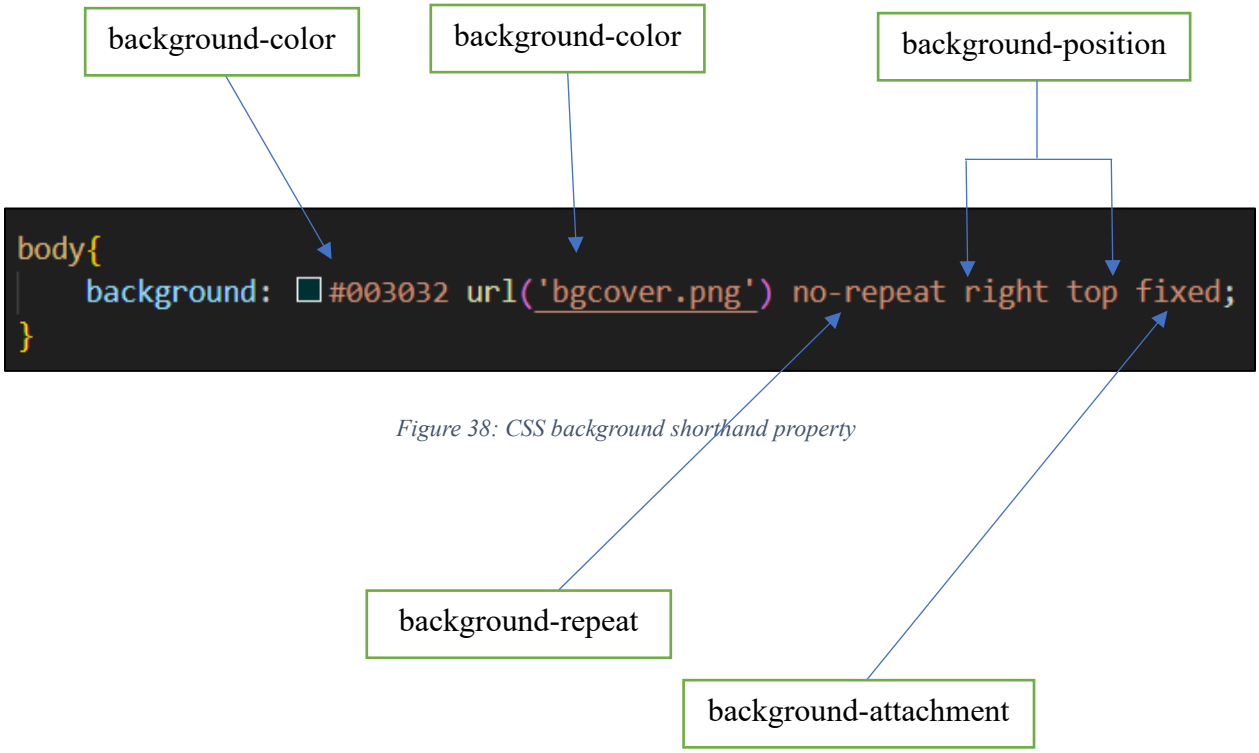


Figure 38: CSS background shorthand property

CSS Height and Width

The *height* and *width* properties are used to set the height and width of an element.

The *height* and *width* properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.

The *height* and *width* properties may have the following values:

- auto - This is default. The browser calculates the height and width
- length - Defines the height/width in px, cm, etc.
- % - Defines the height/width in percent of the containing block
- initial - Sets the height/width to its default value
- inherit - The height/width will be inherited from its parent value

Figure 39: CSS height and width property

CSS Display

The display property is the most important CSS property for controlling layout. The display property specifies if/how an element is displayed.

Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline. Hiding an element can be done by setting the display property to none. The element will be hidden, and the page will be displayed as if the element is not there:

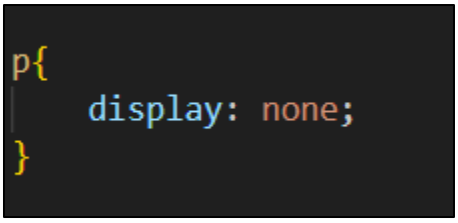


Figure 40: CSS display property

CSS Flexbox

The *flex* property sets the flexible length on flexible items. The flex container becomes flexible by setting the display property to flex.

The flex container properties are:

- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content

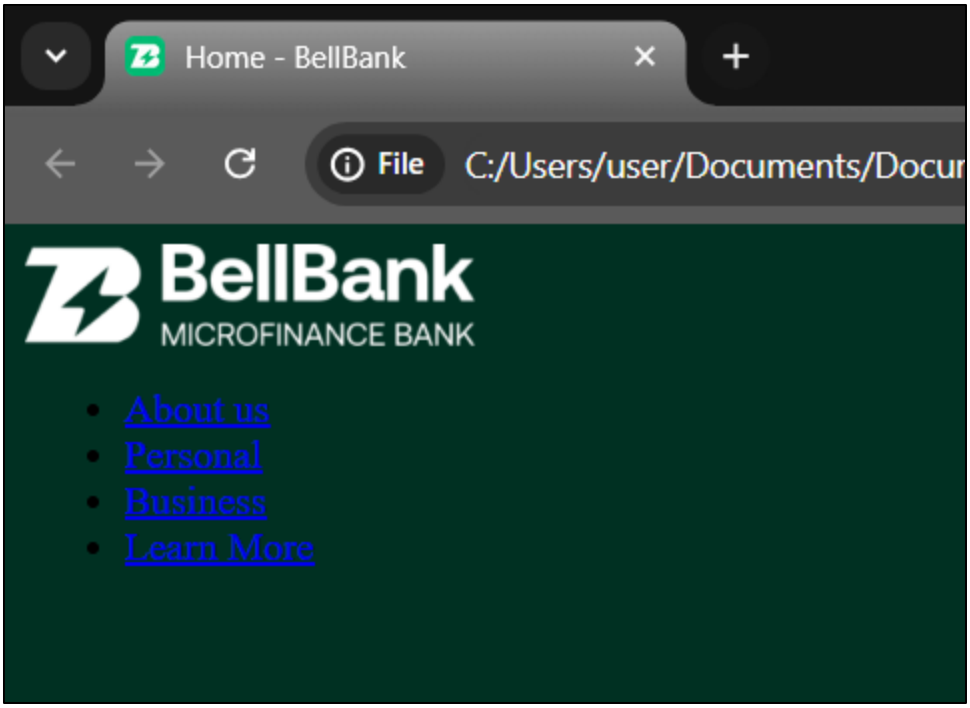


Figure 41: Header Section

Let's style the header section and use Flexbox to make the logo and navigation appear on the same row. We will start by adding an *id* attribute to the *div* element that contained the *header* and *nav* element as shown below:

```
<!-- Header Section  -->
<div id="headerSection">
    <header>
        <a href="index.html" target="_self">
            <img id="logo" src="./images/logo.png" alt="Website Logo">
        </a>
    </header>
    <nav>
        <ul>
            <li>
                <a href="aboutus.html">About us</a>
            </li>
```

Figure 42: adding id attribute

In our style.css file, we will use the *id* selector to style the *div* as shown below:

```
#headerSection{
    display: flex;
}
```

Figure 43: CSS flex display property

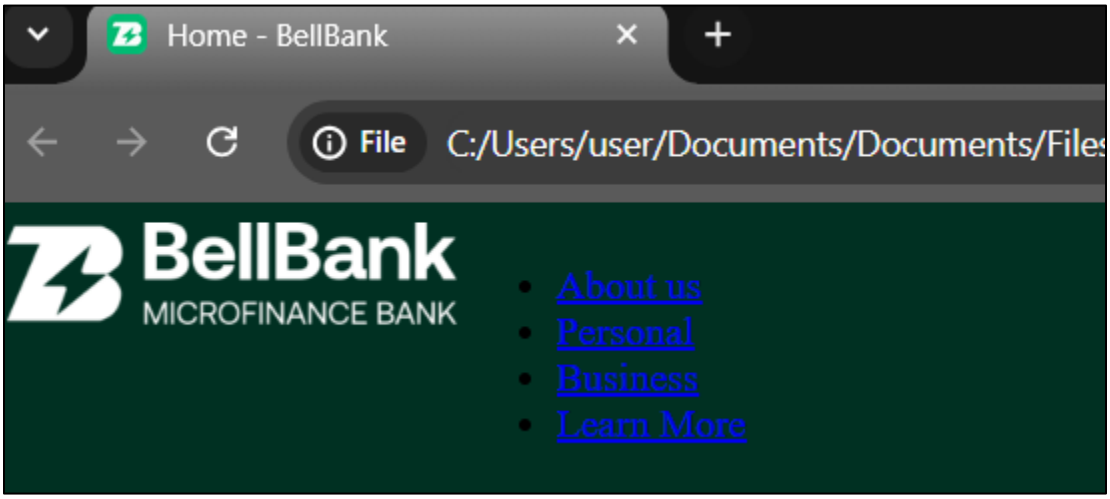


Figure 44: Header section after add flex display property

The flex container can have the following properties, depending on the design requirements:

- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content

These properties can have different values. For example, the align-items property can take values such as *center*, *flex-start*, *flex-end*, *baseline*, or *stretch*. Similarly, the justify-content property can have values like *center*, *flex-start*, *flex-end*, *space-around*, or *space-between*.

```css
#headerSection{
    display: flex;
    justify-content: space-between;
    align-items: center;
}
```

Figure 45: CSS justify-content and align-items property

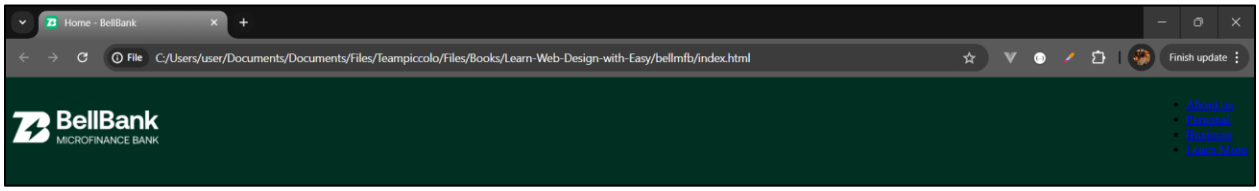


Figure 46: Header Section Display

CSS List

We have already learned in HTML that there are two main types of lists: unordered lists (<*ul*>) - the list items are marked with bullets and ordered lists (<*ol*>) - the list items are marked with numbers or letters.

The CSS list properties allow you to:

- Set different list item markers for ordered lists
- Set different list item markers for unordered lists
- Set an image as the list item marker
- Add background colors to lists and list items

The list-style-type property specifies the type of list item marker. This property has some values like: *circle*, *decimal*, *none*, *square*, *upper-roman*, *lower-alpha*, or *none* etc.

```css
nav li{
    list-style-type: none;
}
```

Figure 47: CSS list-style-type property

The list-style-type property with the value *none* will remove the default style of HTML list items.

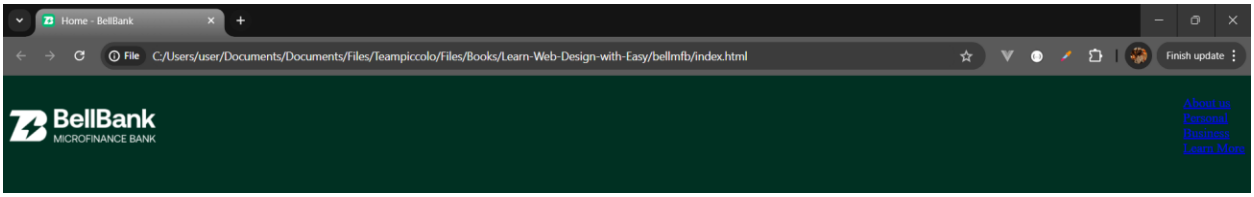


Figure 48: List-style-type output

Let's work on the navigation menu using what we have learned from HTML and CSS.

```
nav ul{
    display: flex;
    flex-direction: row;
}
```

Figure 49: CSS flex-direction property

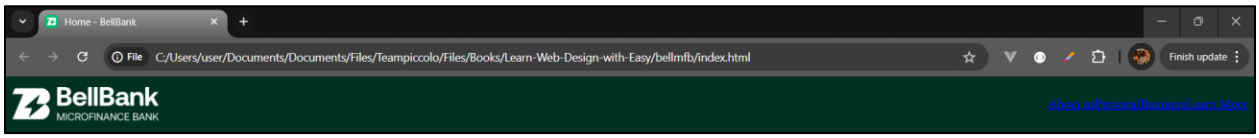


Figure 50: CSS flex-decoration property output

CSS Colors

Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

Let's change the font color of the navigation menu using the CSS color property as shown below:

```
nav a{
    color: white;
}
```

Figure 51: CSS color property

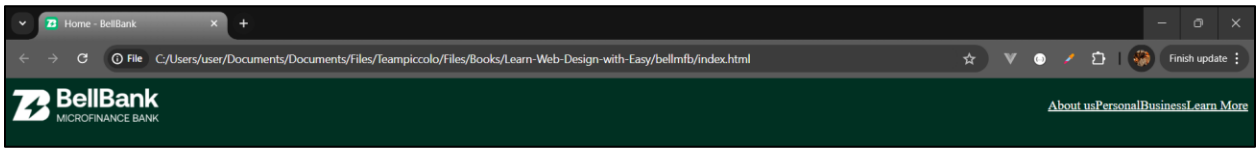


Figure 52: CSS color property output

To remove the underline in the navigation menu, the *text-decoration* property is used. The text-decoration property is mostly used to remove underlines from links.

```
nav a{
    color: white;
    text-decoration: none;
}
```

Figure 53: CSS text-decoration property

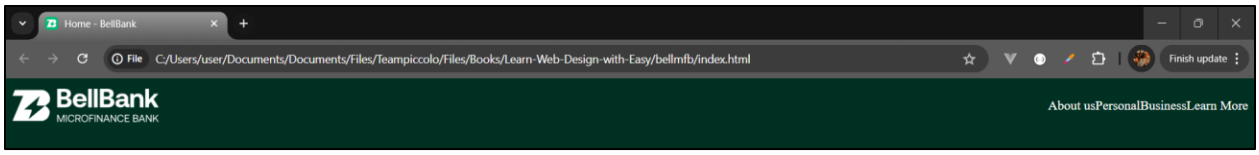


Figure 54: CSS text-decoration property output

CSS Margin

The CSS margin properties are used to create space around elements, outside of any defined borders.

With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

CSS has properties for specifying the margin for each side of an element: margin-top, margin-right, margin-bottom, margin-left.

All the margin properties can have the following values:

- auto - the browser calculates the margin
- length - specifies a margin in px, pt, cm, etc.
- % - specifies a margin in % of the width of the containing element
- inherit - specifies that the margin should be inherited from the parent element

Note: Negative values are allowed.

Let's add some margin to our navigation menu using the CSS margin property, as shown below:

```css
nav li{
    list-style-type: none;
    margin-top: 0px;
    margin-bottom: 0px;
    margin-left: 15px;
    margin-right: 15px;
}
```

Figure 55: CSS margin property

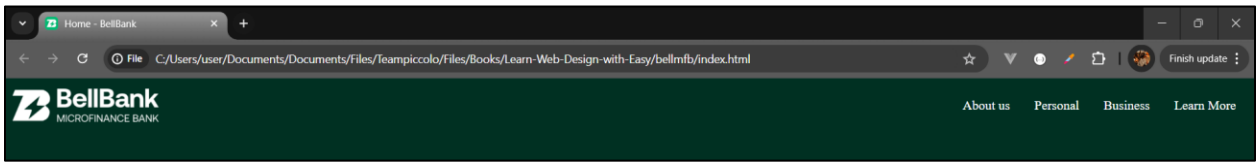
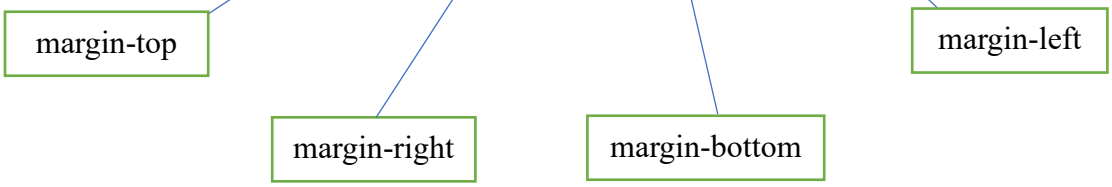


Figure 56: CSS margin property output

You can shorten the margin code above; it is possible to specify all the margin properties in one property using the margin property as shown below:

```css
nav li{
    list-style-type: none;
    margin: 0px 15px 0px 15px;
}
```

Figure 57: CSS margin property II



If the margin property has three values: margin: 0px 15px 0px; top-margin is 0px, right and left margins are 15px, bottom-margin is 0px.

If the margin property has two values: margin: 0px 10px; top and bottom margins are 0px, right and left margins are 15px.

If the margin property has one value: margin: 10px; all four margins are 10px.

You can set the margin property to auto to horizontally center the element within its container. The element will then take up the specified width, and the remaining space will be split equally between the left and right margins.

```
nav li{
    list-style-type: none;
    margin: 0px 15px;
}
```

Figure 58: CSS margin property III

Let's add some margin to the *headerSection* to make the header look better!

```
#headerSection{
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin: 0px 10%;
}
```

Figure 59: CSS margin property IV

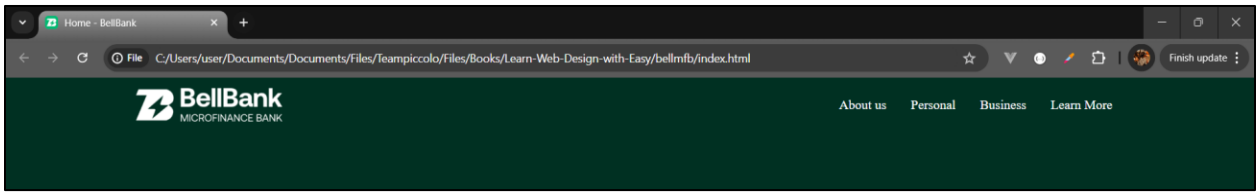


Figure 60: headerSection much better!

CSS Padding

The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

CSS has properties for specifying the padding for each side of an element: padding-top, padding-right, padding-bottom, padding-left.

All the padding properties can have the following values:

- length - specifies a padding in px, pt, cm, etc.
- % - specifies a padding in % of the width of the containing element
- inherit - specifies that the padding should be inherited from the parent element

Note: Negative values are not allowed.

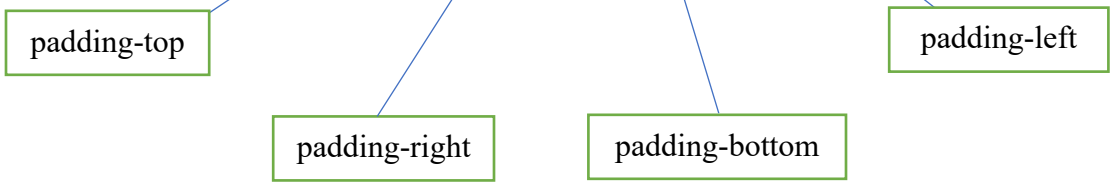
```
#headerSection{
    padding-top: 1%;
    padding-right: 2%;
    padding-bottom: 1%;
    padding-left: 2%;
}
```

Figure 61: CSS padding property

You can shorten the code above; it is possible to specify all the padding properties in one property using the padding property as shown below:

```
#headerSection{
    padding: 1% 2% 1% 2%;
}
```

Figure 62: CSS padding property II



If the padding property has four values: padding: 1% 2% 1% 2%; top-padding is 1%, right-padding is 2%, bottom-padding is 1%, left-padding is 2%.

If the padding property has three values: padding: 1% 2% 1%; top-padding is 1%, right and left paddings are 2%, bottom padding is 1%.

If the padding property has two values: padding: 1% 2%; top and bottom paddings are 1%, right and left paddings are 2%.

If the padding property has one value: padding: 2%; all four paddings are 2%.

Let's update the *headerSection* CSS code and add the padding property to the *headerSection*.

```
#headerSection{
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin: 0px 10%;
    padding: 1% 2%;
}
```

Figure 63: CSS padding property III

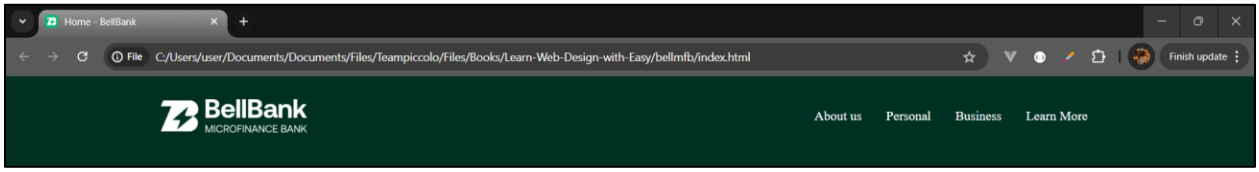


Figure 64: headerSection much better! II

CSS Border

The CSS border properties allow you to specify the style, width, and color of an element's border.

The *border-style* property specifies what kind of border to display. The following values are allowed:

- dotted - Defines a dotted border.
- dashed - Defines a dashed border.
- solid - Defines a solid border.
- double - Defines a double border
- groove - Defines a 3D grooved border. The effect depends on the border-color value.
- ridge - Defines a 3D ridged border. The effect depends on the border-color value.
- inset - Defines a 3D inset border. The effect depends on the border-color value.

- outset - Defines a 3D outset border. The effect depends on the border-color value.
- none - Defines no border.
- hidden - Defines a hidden border.

The *border-style* property can have from one to four values (for the top border, right border, bottom border, and the left border) example border-top, border-bottom, border-left, border-right.

```
p{
    border-style: solid;
}
```

Figure 65: CSS border-style property

NOTE: None of the other CSS border properties we will learn below will have any effect unless the border-style property is set. Make sure to set it first before using any other border-related properties.

The *border-width* property specifies the width of the four borders. The width can be set as a specific size (in px, pt, cm, em, etc.) or by using one of the three pre-defined values: thin, medium, or thick. The border-width property can have from one to four values (for the top border, right border, bottom border, and the left border).

```
p{
    border-style: solid;
    border-width: 2px;
}
```

Figure 66: CSS border-width property

The border-color property is used to set the color of the four borders. The color can be set by:

- name - specify a color name, like “black”
- Hex - specify a hex value, like “#000000”
- RGB - specify a RGB value, like “rgb(0,0,0)”

- transparent - the border-color property can have from one to four values (for the top border, right border, bottom border, and the left border). If border-color is not set, it inherits the color of the element.

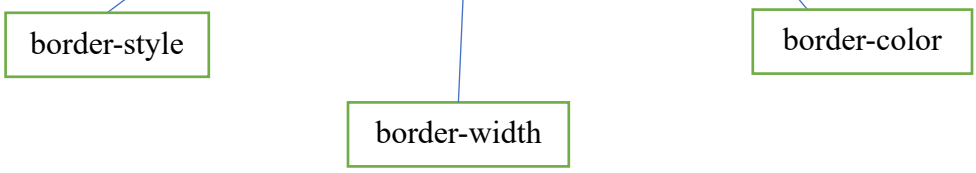
```
p{
    border-style: solid;
    border-width: 2px;
    border-color: black;
}
```

Figure 67: CSS border-color property

You can shorten the code above on a single line using the border property as shown below:

```
p{
    border: solid 2px black;
}
```

Figure 68: CSS border property



Let's add a border-bottom property to the *headerSection*. We will use the CSS border property to achieve this, as shown below:

```
#headerSection{
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin: 0px 10%;
    padding: 1% 2%;
    border-bottom: 2px solid #005c3c;
}
```

Figure 69: CSS border-bottom property

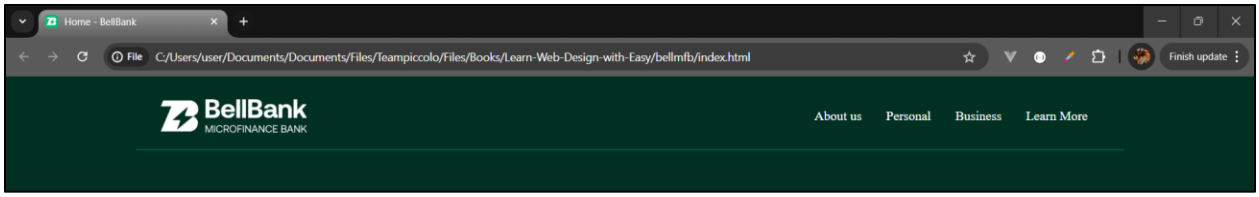


Figure 70: headerSection much better! III

CSS Font

Choosing the right font has a huge impact on how the readers experience a website.

The right font can create a strong identity for your brand.

Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.

The CSS font properties define the font family, boldness, size, and the style of a text. With CSS, you can alter the font family of your web page for make your website much attractive and user friendly.

The font family of a text is set with the *font-family* property. The font-family property should hold several font names as a “fallback” system. If the browser does not support the first font, it tries the next font, and so on.

Note: If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman”.

```
p{
    font-family: 'Times New Roman', Times, serif;
}
```

Figure 71: CSS font-family property

The *font-style* property is mostly used to specify italic text. This property has three values:

- normal - The text is shown normally,
- italic - The text is shown in italics,
- oblique - The text is “leaning” (oblique is very similar to italic, but less supported).

```css
p{
    font-style: italic;
}
```

Figure 72: CSS font-style property

The *font-size* property sets the size of the text. Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

Note: Always use the proper HTML tags, like <*h1*> - <*h6*> for headings and <*p*> for paragraphs. You can use cm, em, px or % as the unit of measurement.

```css
p{
    font-size: 24px;
}
```

Figure 73: CSS font-size property

Note: Many developers use em instead of pixels. The em size unit is recommended by the W3C.

The *font-weight* property specifies the weight of a font: This property has some values:

- bold,
- bolder and
- normal.

```css
p{
    font-weight: bold;
}
```

Figure 74: CSS font-weight property

The *font-variant* property specifies whether or not a text should be displayed in a small-caps font. In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appear in a smaller font size than the original uppercase letters in the text.

```
p{
    font-variant: small-caps;
}
```

Figure 75: CSS font-variant property

CSS Google Font

If you do not want to use any of the standard fonts in HTML, you can use Google Fonts. Google Fonts are free to use, and have more than 1000 fonts to choose from.

Let's try to use "DM Sans" font type on our website.

1. Go to https://fonts.google.com/
2. Use the search box and type DM Sans as shown below:

Figure 76: CSS Google Font

3. Click on "Get Font" button as shown below:

Figure 77: CSS Google Font II

4. Click on "Get embed Code" button as shown below:

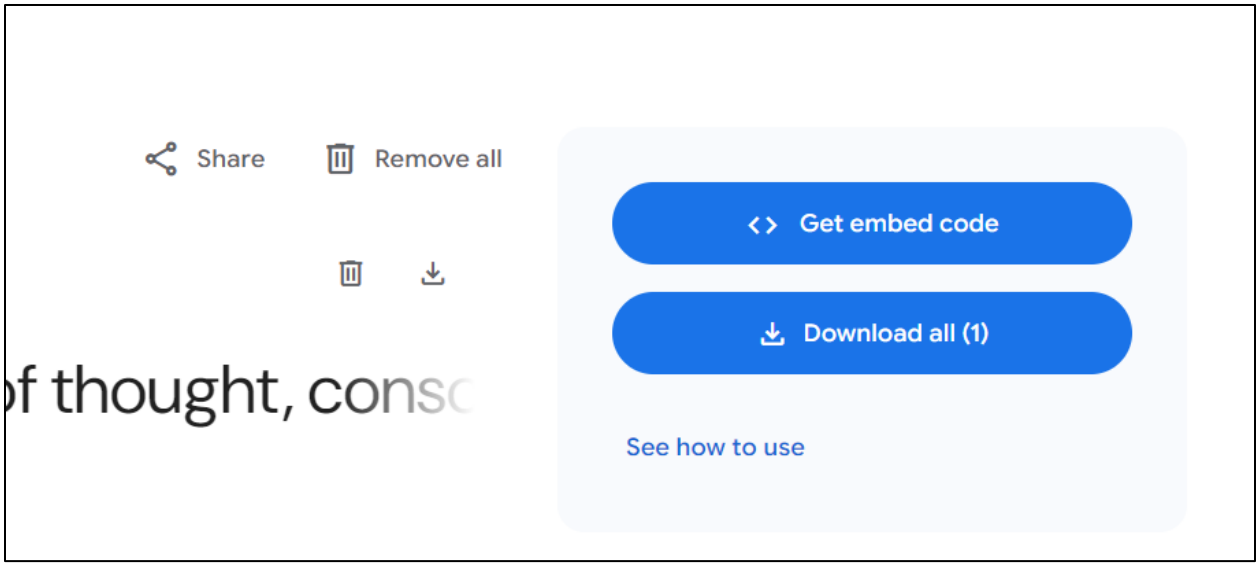


Figure 78: CSS Google Font III

5. Click on “Copy Code” button as shown below:

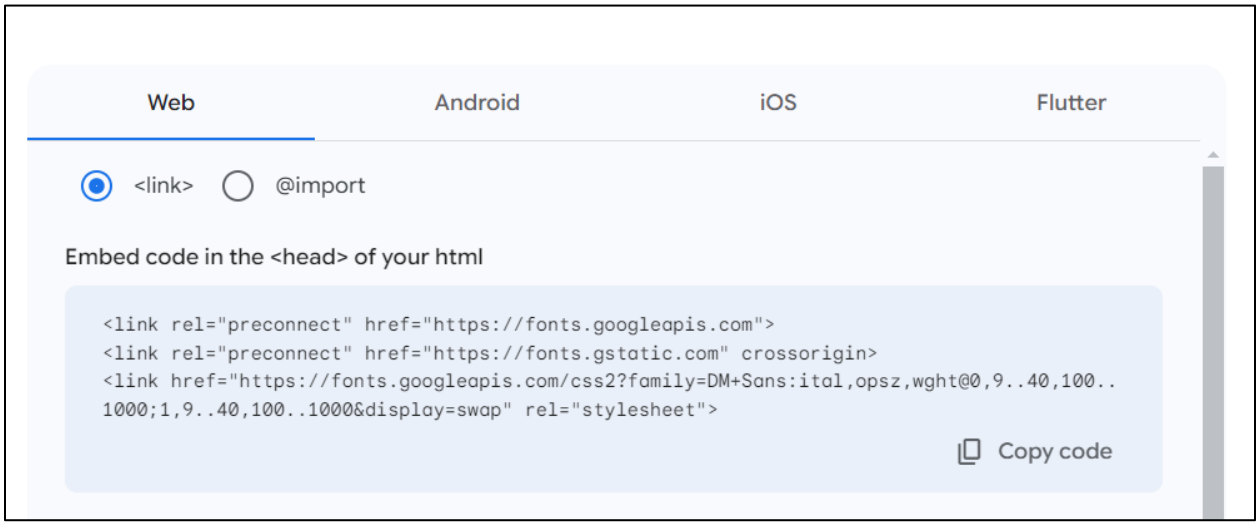


Figure 79: CSS Google Font IV

6. Paste the code inside the *head* element on the index.html file as shown below:

```
    <!-- Google Font  -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=DM+Sans:ital,opsz,wght@0,9..40,100..1000;1,9..40,100..1000&display=swap" rel="stylesheet">
</head>
```

Figure 80: CSS Google Font V

7. Add the font-family property to the body element in the style.css file, as shown below:

```
body{
    background-color: #003022;
    font-family: "DM Sans", sans-serif;
}
```

Figure 81: CSS Google Font VI

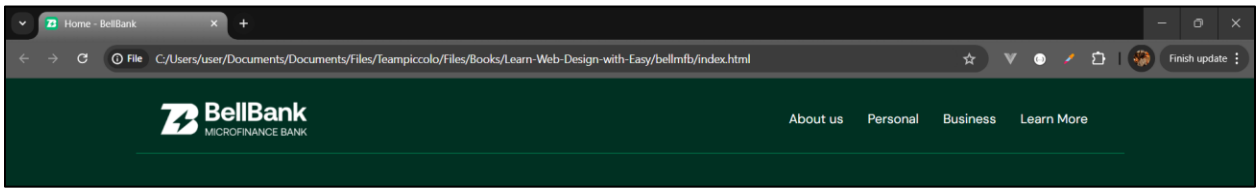


Figure 82: headerSection much better! IV

Note: sans-serif is a fallback font-family. If DM Sans is not available, the system will default to sans-serif.

CSS Text

In this section, we will learn about some CSS properties used for text formatting. CSS properties like *color*, *text-align*, *text-decoration*, *text-transform, text-indent*, *letter-spacing*, *line-height*, *direction* and *word-spacing*.

The *color* property is used to set the color of the text. With CSS, a color is most often specified by: a color name - like “red”, a HEX value - like “#ff0000”, an RGB value - like “rgb(255,0,0)”.

Note:

- The default text color for a page is defined in the body selector.
- For W3C compliant CSS: If you define the color property, you must also define the background-color property.

The *text-align* property is used to set the horizontal alignment of a text. A text can be left or right aligned, centered, or justified. The value of text-align property includes: *left*, *right*, *center* and *justify*. If you have used a word processing software like Microsoft word, the text-align property works exactly as in Microsoft word.

The *text-decoration* property is used to set or remove decorations from text. The value text-decoration: none; is often used to remove underlines from links. The value of text-decoration property includes: *underline*, *line-through*, *over-line*.

Note: It is not recommended to underline text that is not a link, as this often confuses the reader.

The *text-transform* property is used to specify uppercase and lowercase letters in a text. It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word. The value of text-transform property includes: *capitalize*, *lowercase*, *none*, *uppercase*, *initial*, *inherit*, *unset*.

The *text-indent* property is used to specify the indentation of the first line of a text. You use the unit of measurement like px, cm, em, etc. when specifying the value of the property: text-indent: 30px;

The *letter-spacing* property is used to specify the space between the characters in a text. You use the unit of measurement like px, cm, em, etc. when specifying the value of the property: letter-spacing: 5px;

The *line-height* property is used to specify the space between lines. Using the line-height, you don't specify the unit of measurement like cm, px or em: line-height: 1.5;

The *direction* property is used to change the text direction of an element. Some of the values you can use on the direction property includes: rtl and ltr.

The *word-spacing* property is used to specify the space between the words in a text. You use the unit of measurement like px, cm, em, etc. when specifying the value of the property: word-spacing: 3px;

```
p{
    color: black;
    text-align: justify;
    text-decoration: underline;
    text-transform: capitalize;
    text-indent: 30px;
    letter-spacing: 5px;
    line-height: 1.5;
    direction: ltr;
    word-spacing: 3px;
}
```

Figure 83: CSS Text property

Let's continue with the website design. Having completed the header section, we will now start working on the main content of the website.

```
        </div>
        <!-- End of Header Section  -->

        <main>
            <section></section>
            <aside></aside>
        </main>
        <footer></footer>
    </body>
```

Figure 84: main element

Create two divs element inside the *section* element within the *main* element as shown below:

```
<main>
    <section>
        <div></div>
        <div></div>
    </section>
    <aside></aside>
</main>
```

Figure 85: section element

Inside the first, create two divs and write the content as shown below:

```
<section>
    <!-- Slogan  -->
    <div>
        <div>
            Bank faster, better & smarter with BellBank.
        </div>
        <div>
            Experience Banking at Its Best, manage your finances with ease. From quick transactions to personalized solutions, analytics, support and more.
        </div>
    </div>
    <!-- Download Link  -->
    <div></div>
</section>
```

Figure 86: working on slogan div

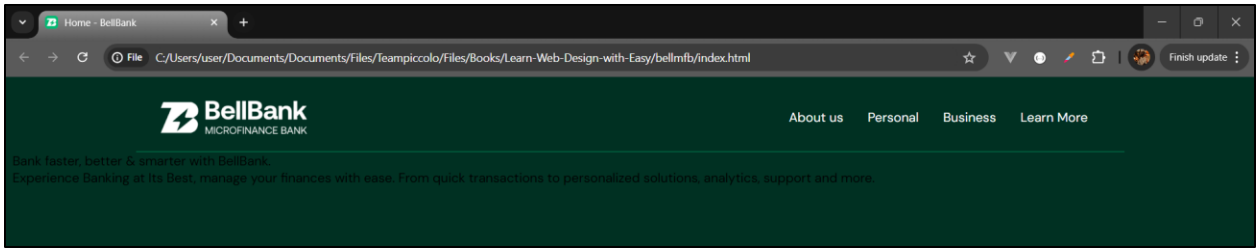


Figure 87: Slogan div output

Let's add the download-link div content. Below is the code snippet:

```
<!-- Download Link  -->
<div>
    <div>
        <div></div>
        <div>Rated 4.5 by 100K+ Customers</div>
    </div>
    <div>
        <span> <img src="./images/apple_white_icon.png" alt="Apple Icon"></span>
        <span> <img src="./images/playstore_white_icon.png" alt="Android Icon"></span>
    </div>
</div>
```

Figure 88: Download Link div content

Adding Icons with Font Awesome

You can add icons to your website using various method like using bootstrap icons, flaticons, fontawesome, etc.

To embed fontawesome, follow the steps below:

1. Go to google.com and search "cdn font awesome" as shown below:

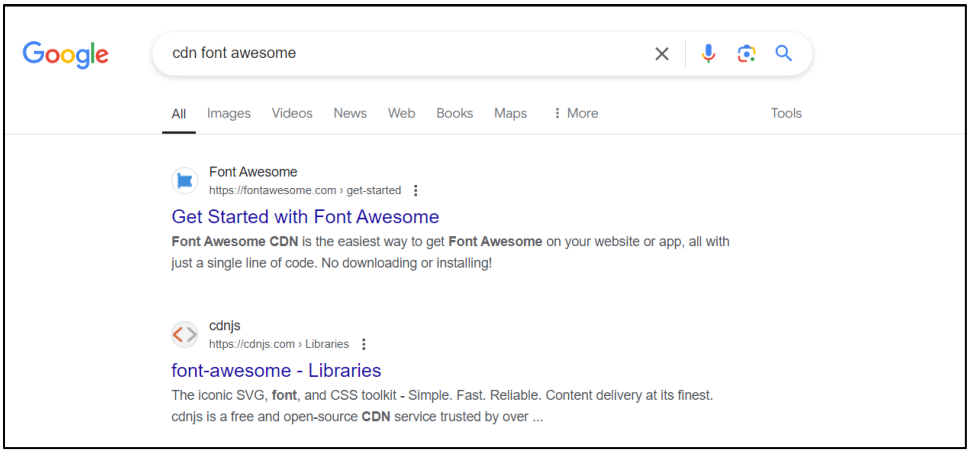
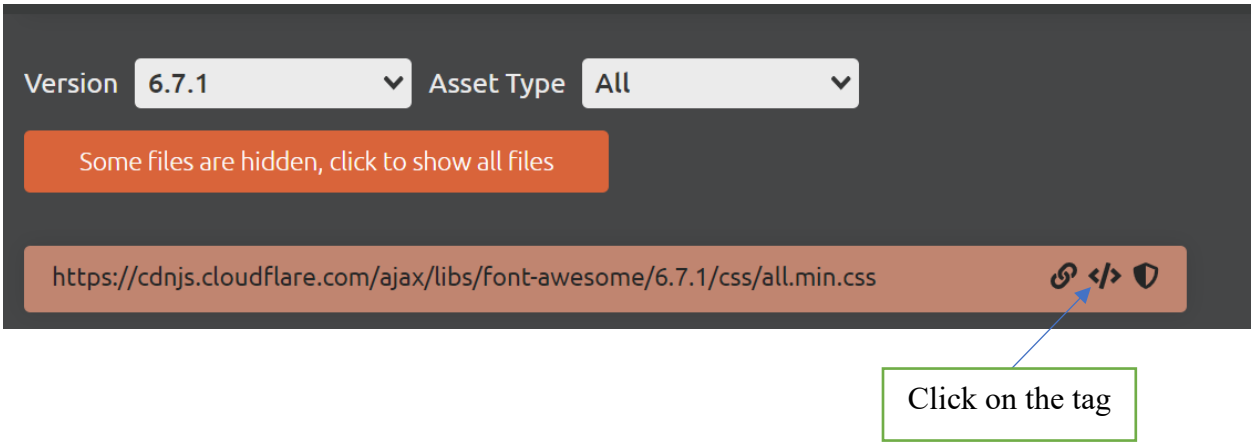


Figure 89: adding fontawesome icons

2. Click on the "**cdnjs**" as shown below:



3. In the index.html file, within the <*head*> element, paste the code shown below:

```
<!-- Fontawesome  -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.7.1/css/all.min.css"
    integrity="sha512-5Hs3dF2AEPkpNAR7UiOHba+lRSJNeM2ECkwxUIxC1Q/FLycGTbNapWXB4tP889k5T5Ju8fs4b1P5z/iB4nMfSQ=="
    crossorigin="anonymous"
    referrerpolicy="no-referrer" />
```

Figure 90: embedding fontawesome link

4. Go to www.fontawesome.com

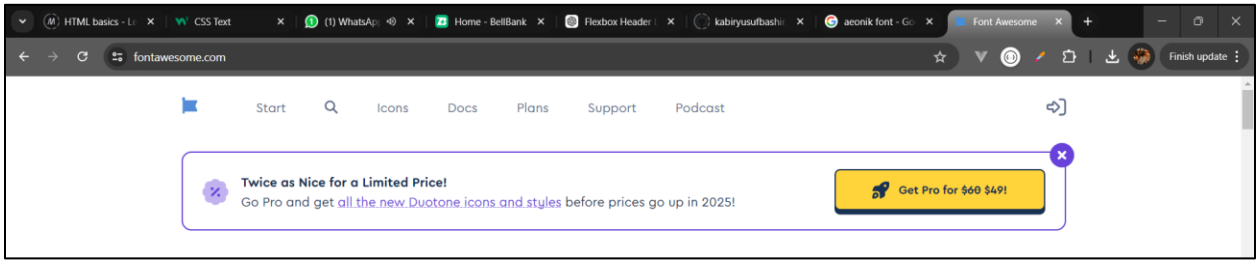


Figure 91: fontawesome website

5. Click on the search button and search for "**star**" as shown below:

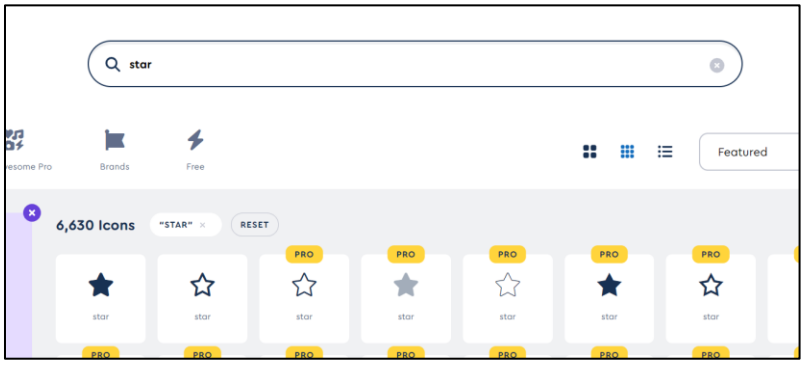


Figure 92: searching on fontawesome

6. Click on the star icon. We will use the first item from the search results.

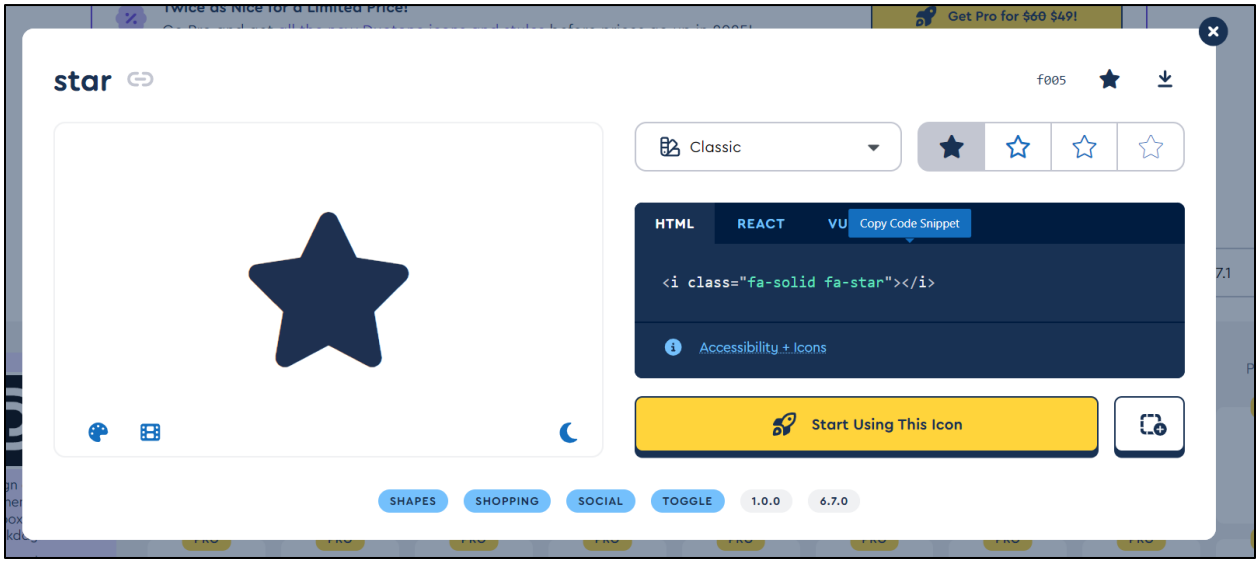


Figure 93: searching on fontawesome II

7. Copy the code snippet and paste it into the index.html file, within the download link section, as shown below:

```
<!-- Download Link  -->
<div>
    <div>
        <!-- Star -->
        <div>
            <i class="fa-solid fa-star"></i>
        </div>
        <div>Rated 4.5 by 100K+ Customers</div>
    </div>
    <div>
        <span> <img src="./images/apple_white_icon.png" alt="Apple Icon"></span>
        <span> <img src="./images/playstore_white_icon.png" alt="Android Icon"></span>
    </div>
</div>
```

Figure 94: Adding the star icons using fontawesome

8. We will be needing five (5) stars, duplicate the star icon as shown below:

```
<!-- Download Link  -->
<div>
    <div>
        <!-- Star -->
        <div>
            <i class="fa-solid fa-star"></i>
            <i class="fa-solid fa-star"></i>
            <i class="fa-solid fa-star"></i>
            <i class="fa-solid fa-star"></i>
            <i class="fa-solid fa-star"></i>
        </div>
        <div>Rated 4.5 by 100K+ Customers</div>
    </div>
    <div>
```

Figure 95: Adding the star icons using fontawesome II

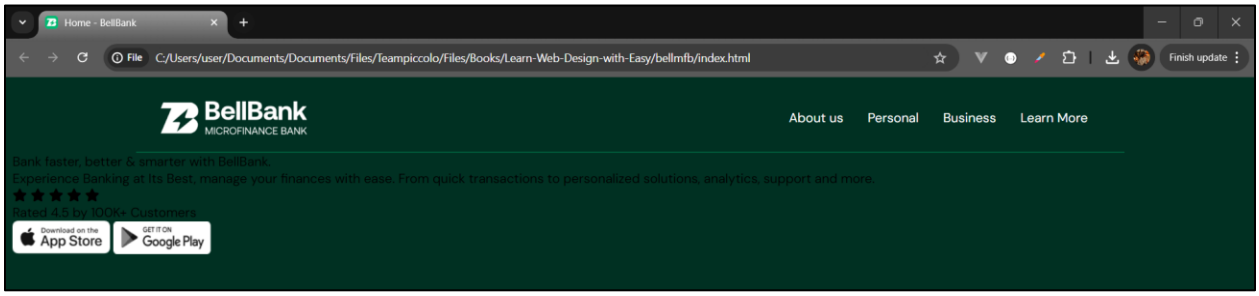


Figure 96: Slogan and download link section

Styling Hero section

The section structure is complete, but the styling needs improvement. Let's apply what we've learned in CSS to enhance the website's user interface and user experience (UI/UX).

1. Add an id attribute to the <*section*> element which contains the slogan and download link divs as shown below:

```
<main>
    <section id="heroSection">
        <!-- Slogan  -->
        <div>···
        </div>
        <!-- Download Link  -->
        <div>···
        </div>
    </section>
```

Figure 97: adding id attribute to the section element

2. In the style.css file, use the ID selector to style the hero section, as shown below:

```css
#heroSection {
    margin: 5% 10%;
    display: grid; /* Activate grid layout */
    grid-template-columns: 1fr 1fr; /* Two equal columns */
    grid-template-rows: auto; /* Rows adjust to content height */
    gap: 20px; /* Space between grid items */
}
```

Figure 98: CSS display property (grid)

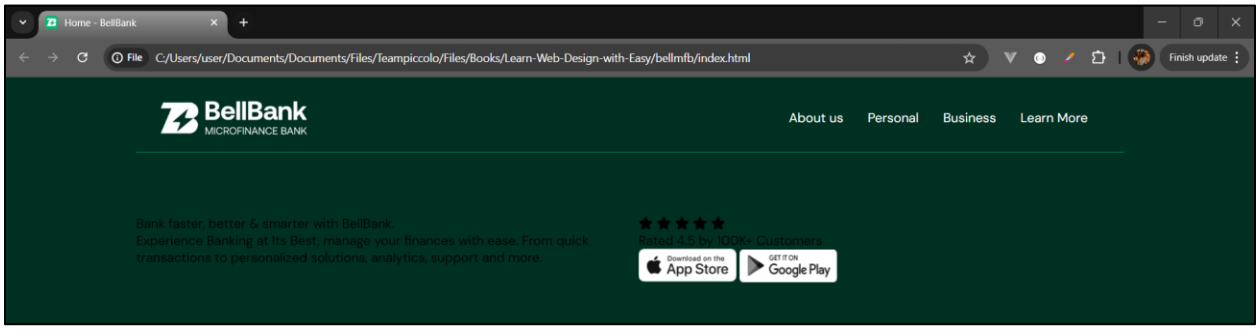


Figure 99: CSS display property (grid) II

Read: you can learn more about CSS grid on this link www.w3schools.com/css/css_grid.asp

3. Add an ID attribute to the slogan div as shown below:

```html
<!-- Slogan  -->
<div>
    <div id="slogan">
        Bank faster, better & smarter with BellBank.
    </div>
    <div>
        Experience Banking at Its Best, manage your fina
    </div>
</div>
```

Figure 100: Styling Slogan div

4. Add the following CSS properties to the div as shown below:

```
#slogan{
    color: white;
    font-size: 3.5em;
    font-weight: bold;
    line-height: 1.2;
}
```

Figure 101: Slogan stylesheet

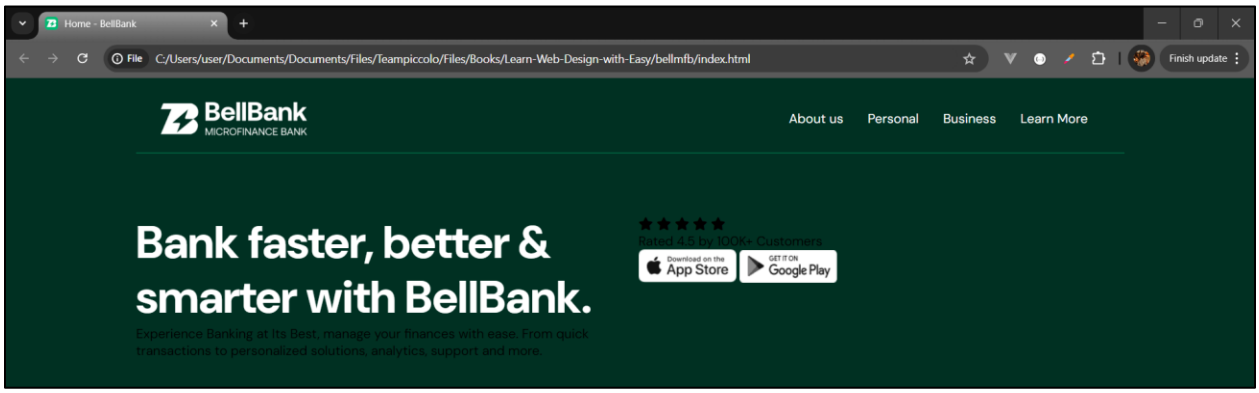


Figure 102: Slogan div output

5. Add an ID attribute to the heroMsg div as shown below:

```
<!-- Slogan  -->
<div>
    <div id="slogan">
        Bank faster, better & smart
    </div>
    <div id="heroMsg">
        Experience Banking at Its B
    </div>
</div>
```

Figure 103: heroMsg div

6. Add the following CSS properties to the heroMsg div as shown below:

```css
#heroMsg{
    color: white;
    font-size: 0.9em;
    margin: 4% 0px;
}
```

Figure 104: heroMsg div CSS properties

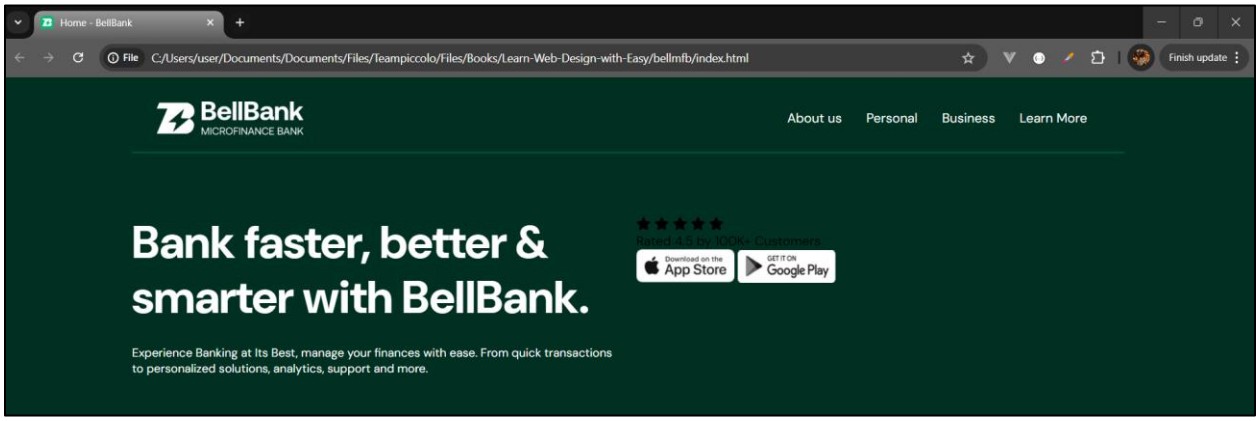


Figure 105: heroMsg div output

7. The slogan and heroMsg divs are complete; let's now work on the download-link div. Update your download-link div code as shown below:

```html
<!-- Download Link  -->
<div>
    <div id="starDiv">
        <!-- Star -->
        <div>
            <i class="fa-solid fa-star"></i>
            <i class="fa-solid fa-star"></i>
            <i class="fa-solid fa-star"></i>
            <i class="fa-solid fa-star"></i>
            <i class="fa-solid fa-star"></i>
        </div>
        <div>Rated 4.5 by 100K+ Customers</div>
    </div>
    <div>
        <span> <img src="./images/apple_white_icon.png" alt="Appl
        <span> <img src="./images/playstore_white_icon.png" alt="
    </div>
</div>
```

Figure 106: starDiv code snippet

8. Add the following CSS properties to the starDiv as shown below:

```css
#starDiv{
    border-left: 3px solid white;
    padding: 1% 2%;
    margin-bottom: 4%;
    color: white;
    font-size: 0.8em;
}
```

Figure 107: starDiv CSS properties

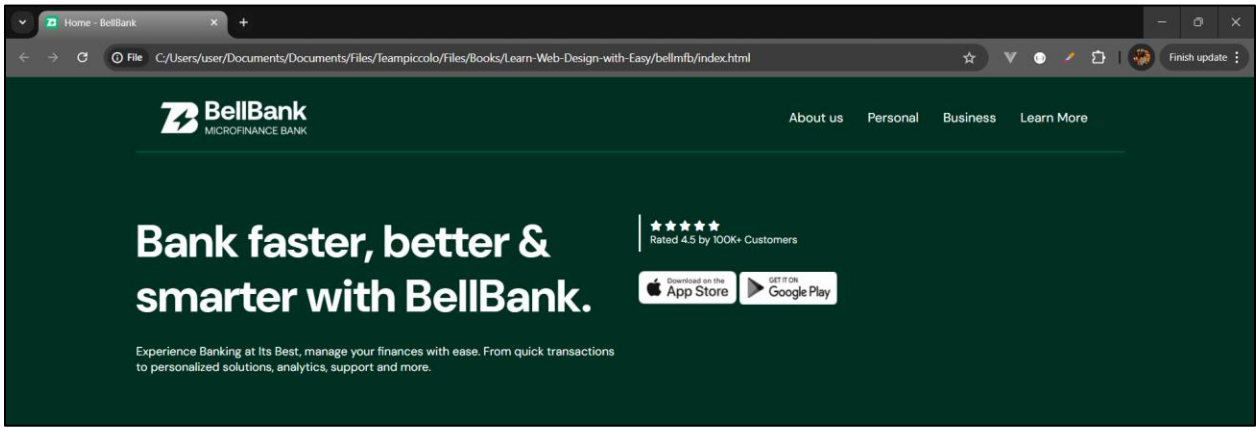


Figure 108: starDiv style output

9. Now, let's change the color of the stars using the color property. First, we will add a class to each star that we want to style and then change its color to yellow. Let's start by adding the class to the stars, as shown below:

```html
<div id="starDiv">
    <!-- Star -->
    <div>
        <i class="fa-solid fa-star yellow_star"></i>
        <i class="fa-solid fa-star yellow_star"></i>
        <i class="fa-solid fa-star yellow_star"></i>
        <i class="fa-solid fa-star yellow_star"></i>
        <i class="fa-solid fa-star"></i>
    </div>
    <div>Rated 4.5 by 100K+ Customers</div>
</div>
```

Figure 109: adding yellow_star class to the star icons

10. Now, let's add the CSS properties to the class (yellow_star) in our style.css file as shown below:

```css
.yellow_star{
    color: #f0ad4e;
}
```

Figure 110: adding the CSS properties to the yellow_star class

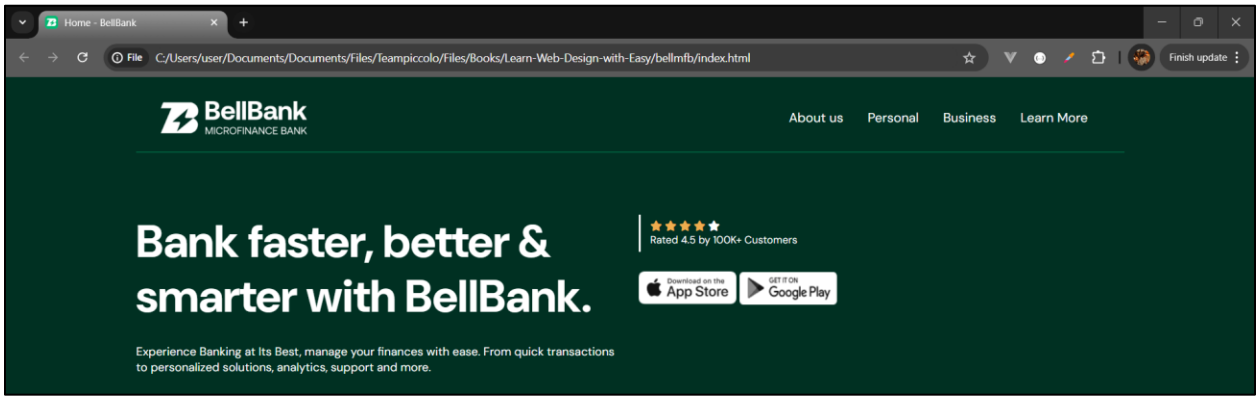


Figure 111: yellow_star class display output

11. Next, let's work on the download icons and improve the styling. Add downloadIcon class to the icons as shown below:

```html
<!-- Download Icon  -->
<div>
    <span class="downloadIcon"><img src="./images/apple_white_icon.png" alt="Apple Icon"></span>
    <span class="downloadIcon"><img src="./images/playstore_white_icon.png" alt="Android Icon"></span>
</div>
```

Figure 112: adding downloadIcon class to the download icons

12. Add the following CSS properties to the downloadIcon class as shown below:

```css
.downloadIcon{
    padding: 1% 2%;
}
```

Figure 113: Adding CSS properties to the downloadIcon class

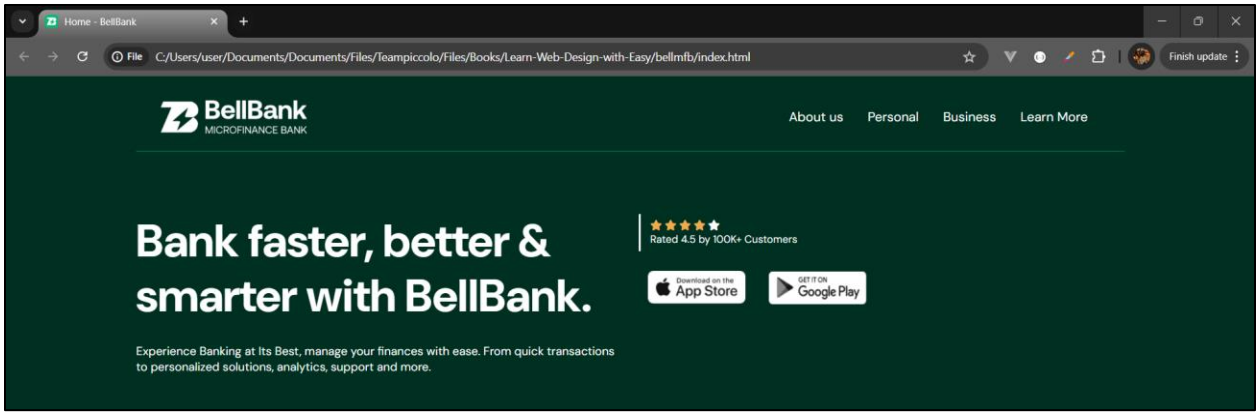


Figure 114: downloadIcon output

13. Now, let's center-align the slogan and download-link divs using the align-items CSS property. Remember, both divs are nested within the heroSection div. In the style.css file, we will update the code and add the align-items property to the heroSection div, as shown below:

```
#heroSection {
    margin: 5% 10%;
    display: grid; /* Activate grid layout */
    grid-template-columns: 1fr 1fr; /* Two equal columns */
    grid-template-rows: auto; /* Rows adjust to content height */
    gap: 20px; /* Space between grid items */
    align-items: center;
}
```

Figure 115: CSS align-items property

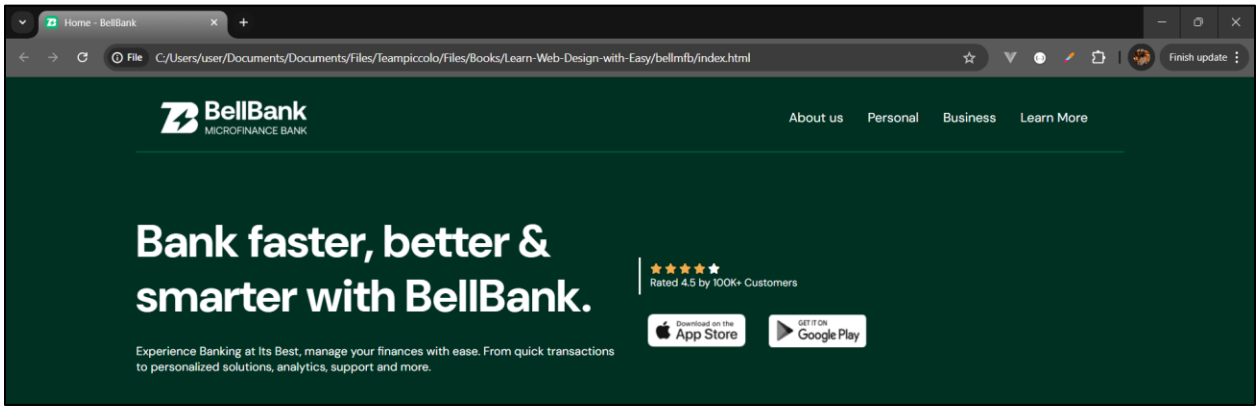


Figure 116: heroSection update output

14. Next, let's add some margin to the download-link div. Start by adding an ID attribute to the div that nests the stars div and the download-icons div, as shown below:

```
<!-- Download Link  -->
<div id="downloadLink">
    <div id="starDiv">
        <!-- Star -->
        <div>
            <i class="fa-solid fa-star yellow
            <i class="fa-solid fa-star yellow
            <i class="fa-solid fa-star yellow
            <i class="fa-solid fa-star yellow
            <i class="fa-solid fa-star"></i>
        </div>
        <div>Rated 4.5 by 100K+ Customers</di
    </div>
    <!-- Download Icon  -->
    <div>
```

Figure 117: adding downloadLink ID attribute

15. Adding the following CSS properties to the downloadLink div in the style.css file as shown below:

```
#downloadLink{
    width: 90%;
    margin: 0 35%;
}
```

Figure 118: downloadLink CSS properties

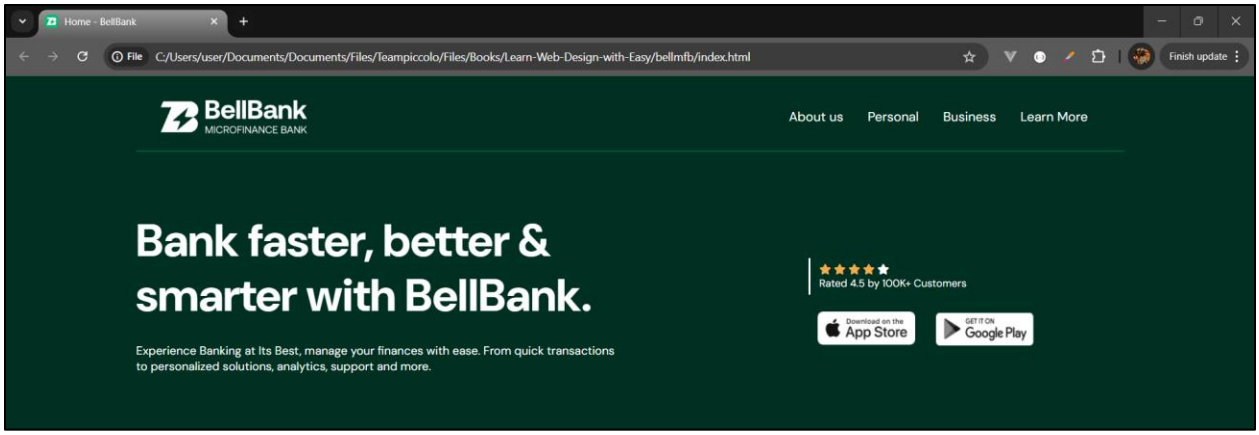


Figure 119: Complete HeroSection design

Exercise 2

1) An HTML element can have more than one ID attribute?
   a) True
   b) False
2) An HTML element can have more than one class attribute?
   a) True
   b) False
3) The CSS display property has the following values except?
   a) Flex
   b) Margin
   c) Grid
   d) None
   e) Block
   f) Inline
4) Which of the following CSS property is used to change the font-type?
   a) Font-color
   b) Color
   c) Font-families
   d) Font-family

Styling Slideshow Section

Creating a slideshow can be achieved using HTML, CSS, and JavaScript. Since we haven't learned JavaScript yet, let's start by creating the basic HTML and CSS structure. Once we've learned JavaScript, we'll update the code later. 😊

1. Now, let's start by create a <*section*> element after the heroSection element. We will give it a value of slideShowSection as shown below:

```
<main>
    <!-- heroSection  -->
    <section id="heroSection">···
    </section>
    <!-- slideShowSection -->
    <section id="slideShowSection">

    </section>
    <aside></aside>
</main>
```

Figure 120: creating slideShowSection element

2. Next, create a div element and add an img element to it as shown below:

```
<!-- slideShowSection -->
<section id="slideShowSection">
    <!-- Slider Images  -->
    <div>
        <img src="./images/slider_one.png" alt="Slider One">
    </div>
</section>
```

Figure 121: creating slideShowSection element II

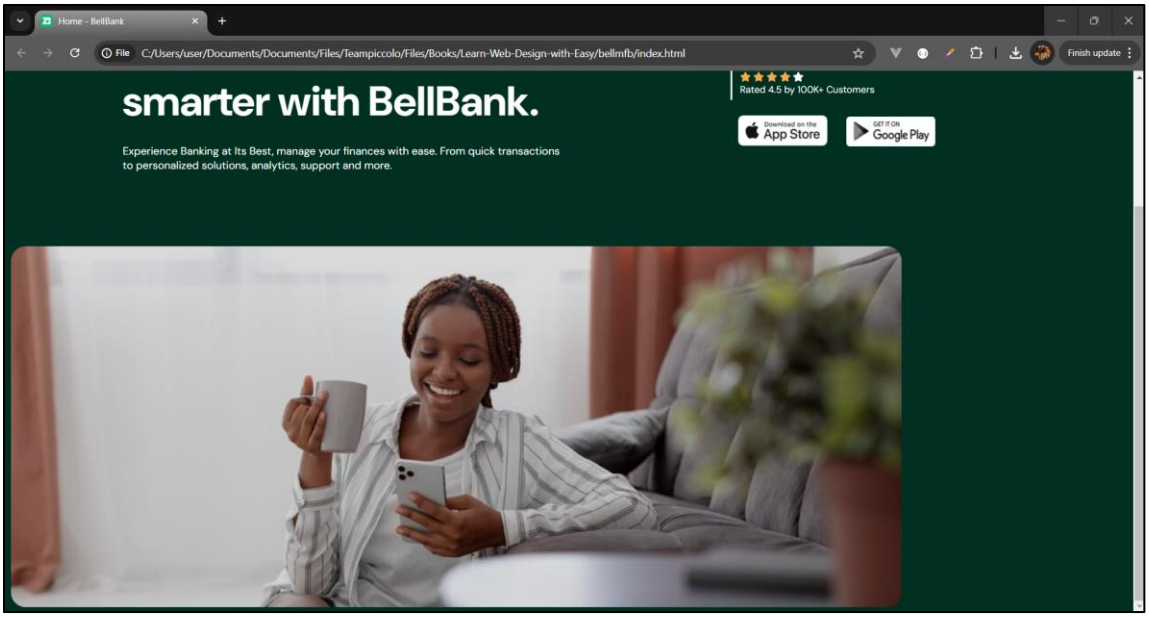


Figure 122: creating slideShowSection element III

3. Add the following CSS properties to slideShowSection element as shown below:

```
#slideShowSection{
    display: flex;
    justify-content: center;
    margin: 5% 0px;
}
```

Figure 123: creating slideShowSection element IV

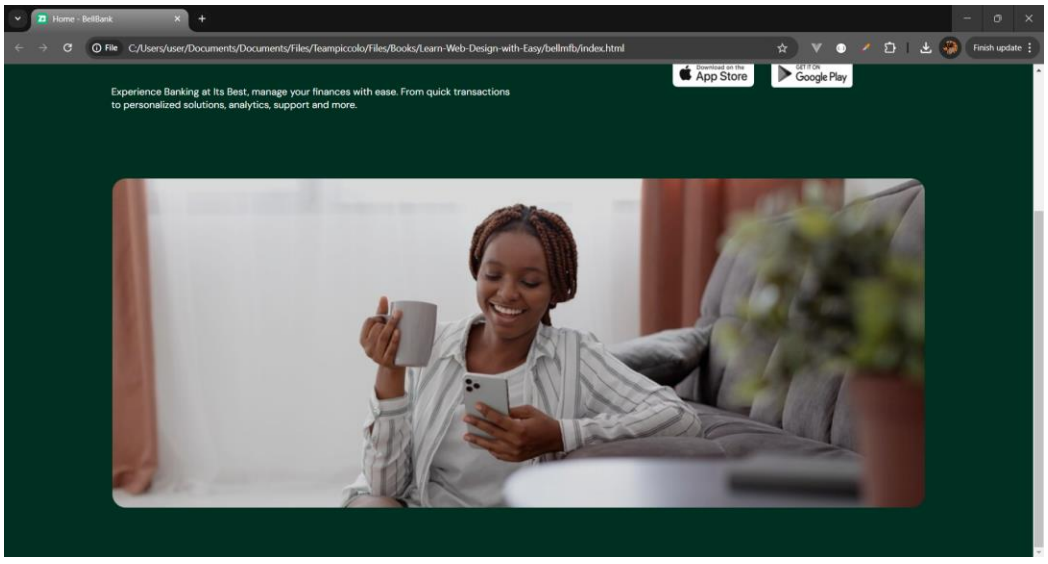


Figure 124: creating slideShowSection element V

Much better now, let will update the code and create the slideshow after learning about JavaScript. 😊

Styling License Section

1. We will create a section after the slideshow section and add an ID attribute to it with a value of "pageSection", as shown below:

```
<!-- slideShowSection -->
<section id="slideShowSection">···
</section>
<!-- pageSection  -->
<section id="pageSection">


</section>
```

Figure 125: Styling License Section

2. Create a div and add an ID attribute to it with a value of "licenseDiv". Within the licenseDiv, create two <*span*> elements and add the contents as shown below:

```
<!-- pageSection  -->
<section id="pageSection">
    <!-- License and Regulators  -->
    <div id="licenseDiv">
        <span>License and supported by:</span>
        <span>
            <img src="./images/cbn_icon.png" alt="CBN Icon">
            <img src="./images/nibss_icon.png" alt="NIBSS Icon">
            <img src="./images/ndic_icon.png" alt="NDIC Icon">
        </span>
    </div>
</section>
```

Figure 126: Styling License Section II

3. Add the following CSS properties to the "pageSection" in the style.css file, as shown below:

```
#pageSection{
    background-color: white;
}
```

Figure 127: Styling License Section II

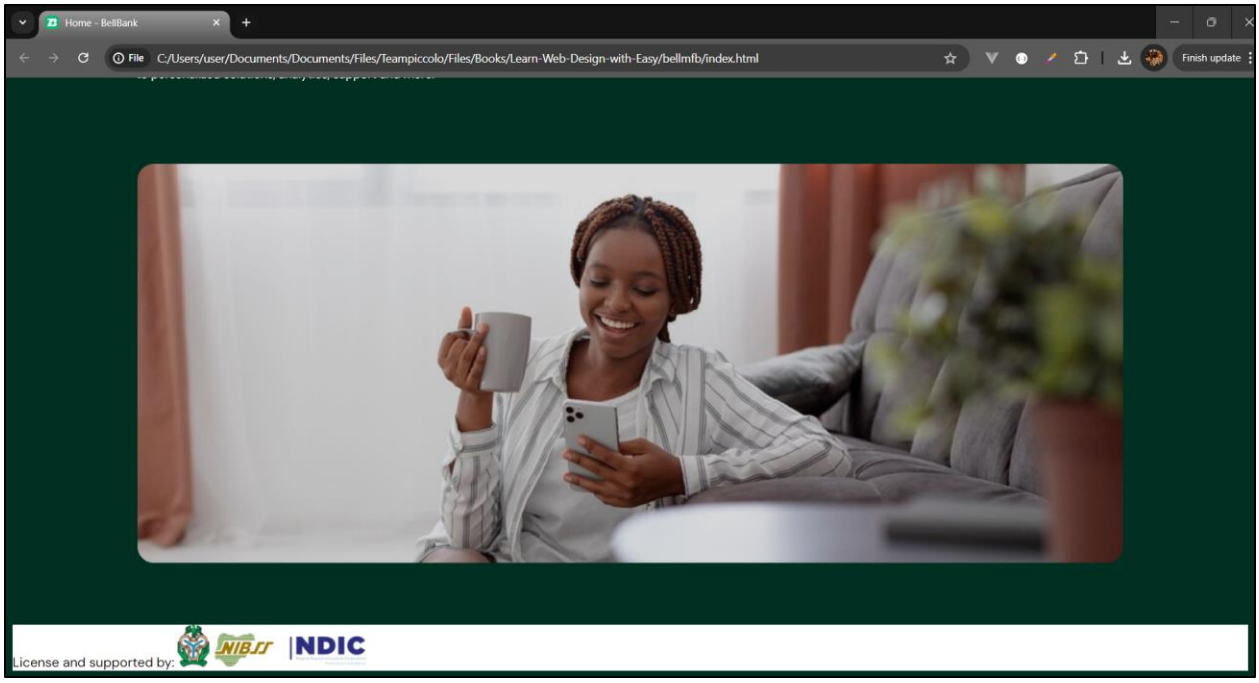


Figure 128: Styling License Section II

4. If you notice, the “pageSection” has some default margin and padding, which we don’t want. As we’ve learned, some HTML elements come with default margin and padding. To remove this behavior, we need to update the CSS for the body element, as shown below:

```
body{
    background-color: #003022;
    font-family: "DM Sans", sans-serif;
    padding: 0px;
    margin: 0px;
}
```

Figure 129: Styling License Section II

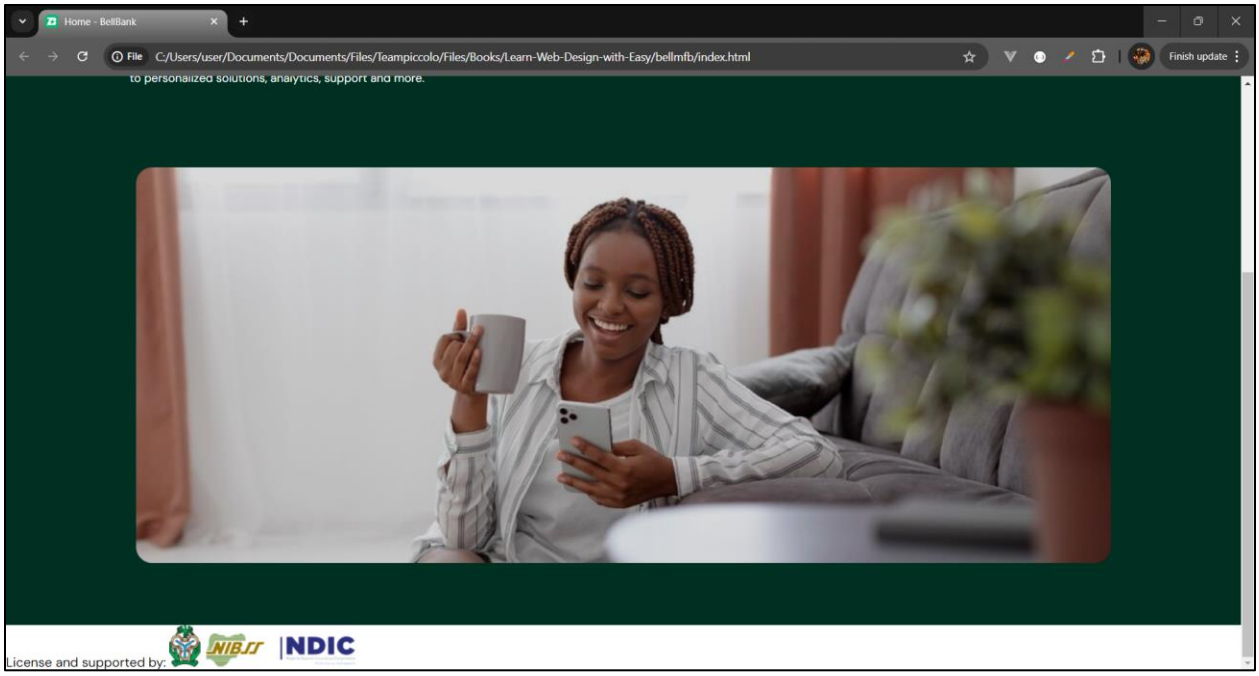


Figure 130: Styling License Section III

5. Add the following CSS properties to the “licenseDiv” in the style.css file, as shown below:

```
#licenseDiv{
    padding: 2% 0px;
    display: flex;
    justify-content: center;
    align-items: center;
    color: #858178;
    font-weight: bold;
}
```

Figure 131: Styling License Section IV

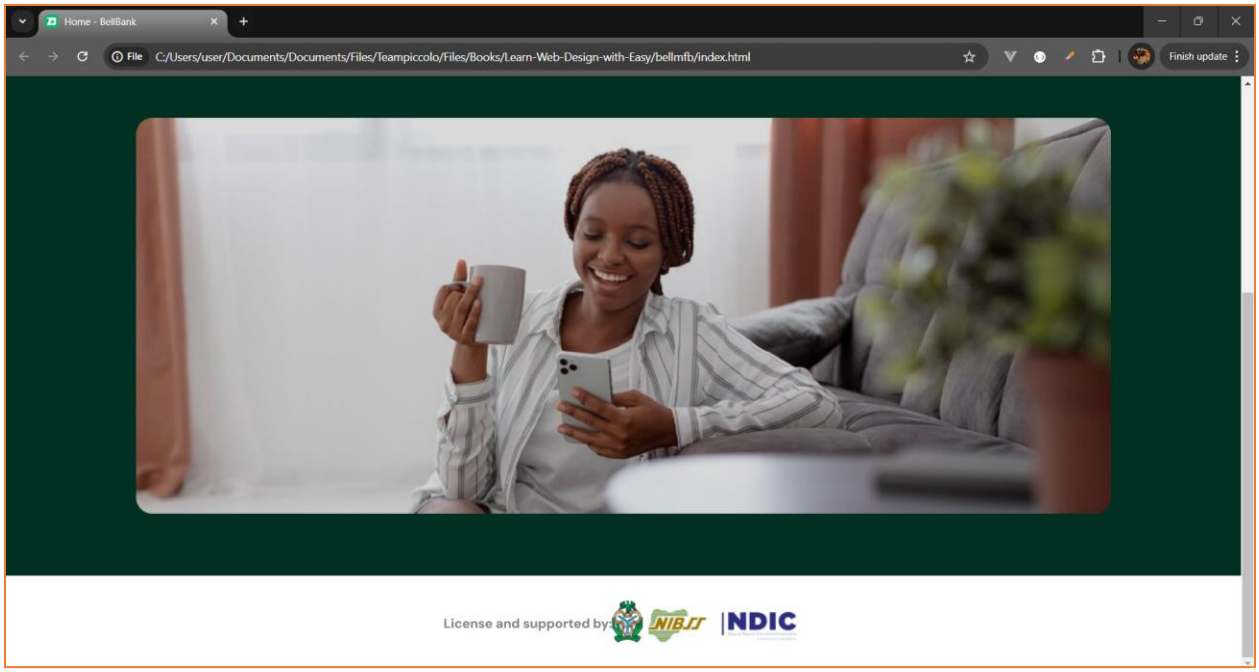


Figure 132: Styling License Section V

6. Let's create some spacing between the text and icons. We will use the HTML special character "** **", which is used to add spacing in HTML. Update the licenseDiv HTML code snippet as shown below:

```
<!-- License and Regulators  -->
<div id="licenseDiv">
    <span>License and supported by:</span>
         
    <span>
        <img src="./images/cbn_icon.png" alt="CBN Icon">
        <img src="./images/nibss_icon.png" alt="NIBSS Icon">
        <img src="./images/ndic_icon.png" alt="NDIC Icon">
    </span>
</div>
```

Figure 133: Styling License Section VI

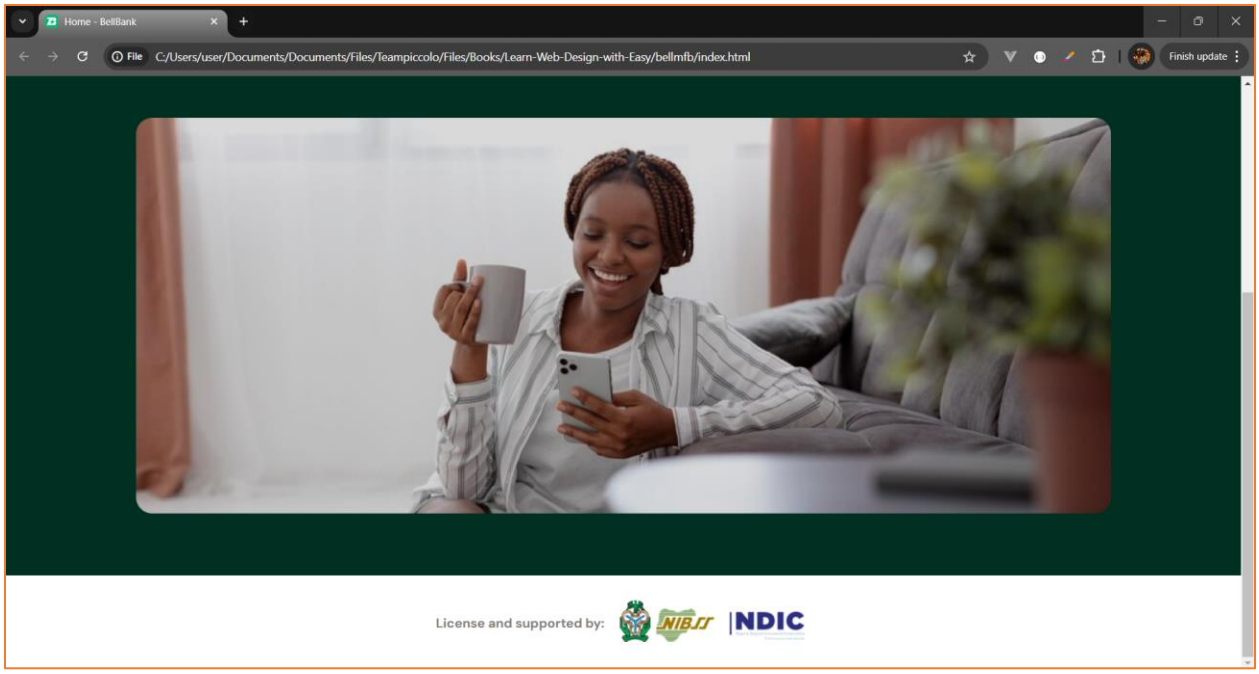


Figure 134: Styling License Section VII

Much better right? HTML and CSS is fun! 😊

Styling Account Types Section

1. Let's create a div inside the "pageSection" element and add an ID attribute to it with a value of "accountTypeSection" as shown below:

```
    </div>
    <!-- accountTypeSection  -->
    <div id="accountTypeSection">

    </div>
</section>
```

Figure 135: Styling account types section

2. Add the following code snippet to the div, as shown below:

```
    <!-- accountTypeSection  -->
    <div id="accountTypeSection">
        <div>BUILT FOR GROWTH</div>
        <div>Banking designed for personal and business growth</div>
        <div>Providing financial solutions that empowering both business and individuals
        <div>
            <img src="./images/personal_account_icon.png" alt="Personal Account">
            <img src="./images/business_account_icon.png" alt="Business Account">
            <img src="./images/loan_icon.png" alt="Loan Icon">
        </div>
    </div>
</section>
```

Figure 136: Styling account types section II

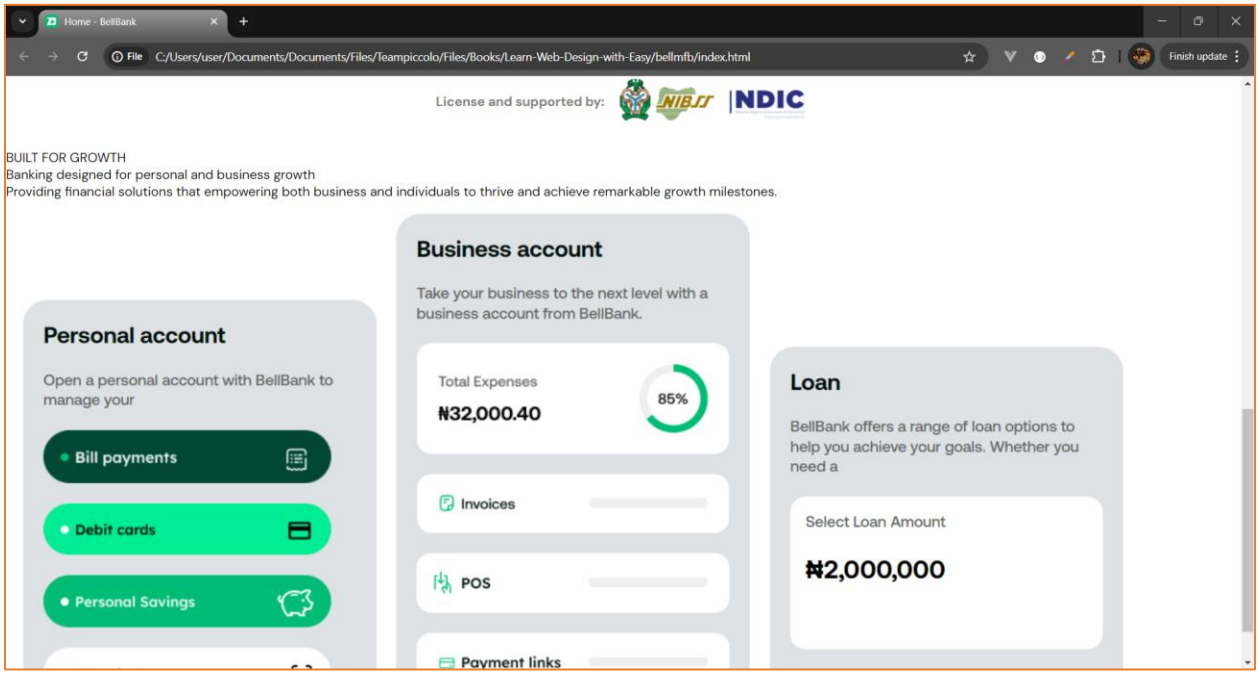


Figure 137: Styling account types section III

3. Add the following CSS properties the “accountTypeSection” div, as shown below:

```
#accountTypeSection{
    text-align: center;
    padding: 2%;
}
```

Figure 138: Styling account types section IV

4. Next, let’s learn how to create custom CSS classes. We will create the following classes in the style.css file, as shown below:

```css
.green-text{
    color: #93ce8f;
}

.grey-text{
    color: #646967;
}

.black-text{
    color: #00100b;
}
```

Figure 139: Styling account types section V

5. Now, let's update the HTML code and add the classes to the divs inside the "accountTypeSection", as shown below:

```html
<!-- accountTypeSection  -->
<div id="accountTypeSection">
    <div class="green-text">
        BUILT FOR GROWTH
    </div>
    <div class="black-text">
        Banking designed for personal <br>
        and business growth
    </div>
    <div class="grey-text">
        Providing financial solutions that empowering both business and individuals to <br>
        thrive and achieve remarkable growth milestones.
    </div>
    <div>
        <img src="./images/personal_account_icon.png" alt="Personal Account">
        <img src="./images/business_account_icon.png" alt="Business Account">
        <img src="./images/loan_icon.png" alt="Loan Icon">
    </div>
</div>
```

Figure 140: Styling account types section VI

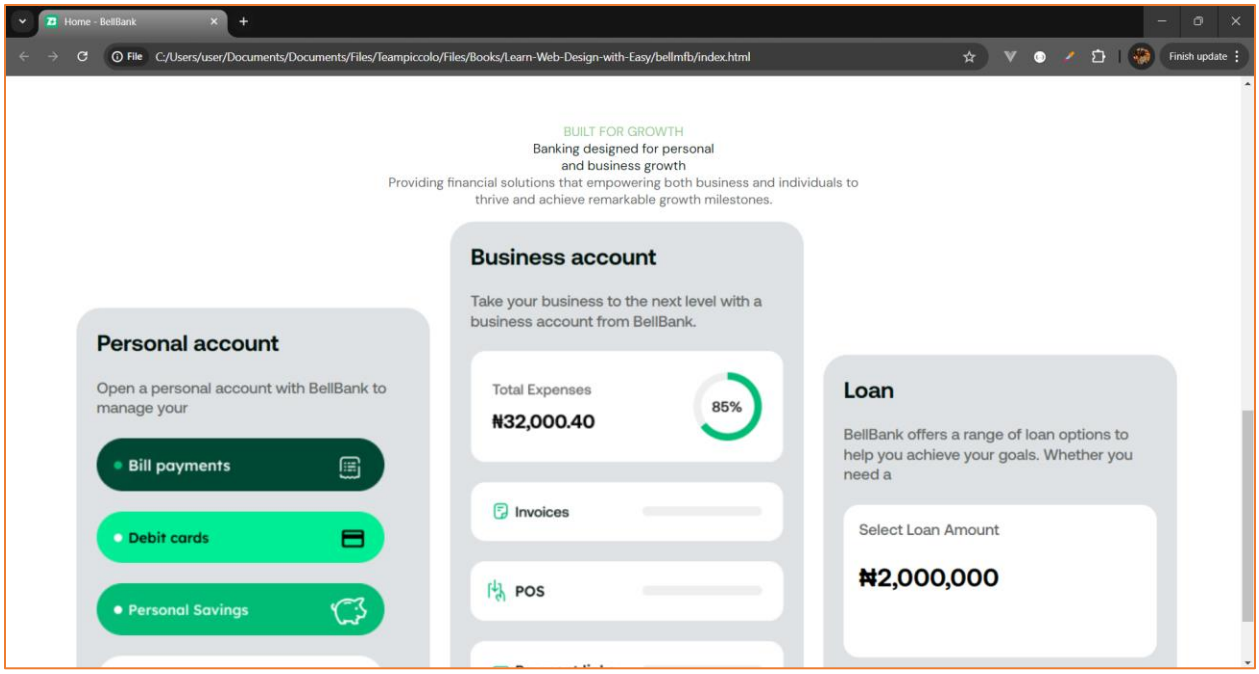


Figure 141: Styling account types section VII

6. Next, let's create some custom CSS that we will use to add padding within elements, as shown below:

```
.padding-y-2{
    padding-top: 1%;
    padding-bottom: 1%;
}
```

Figure 142: Styling account types section VIII

7. Next, add the "**padding-y-2**" class to the divs, as shown below:

```
<!-- accountTypeSection  -->
<div id="accountTypeSection">
    <div class="green-text padding-y-2">
        BUILT FOR GROWTH
    </div>
    <div class="black-text padding-y-2">
        Banking designed for personal <br>
        and business growth
    </div>
    <div class="grey-text padding-y-2">
        Providing financial solutions that empowering both business and individuals to <br>
        thrive and achieve remarkable growth milestones.
    </div>
    <div>
        <img src="./images/personal_account_icon.png" alt="Personal Account">
        <img src="./images/business_account_icon.png" alt="Business Account">
        <img src="./images/loan_icon.png" alt="Loan Icon">
    </div>
</div>
```

Figure 143: Styling account types section IX

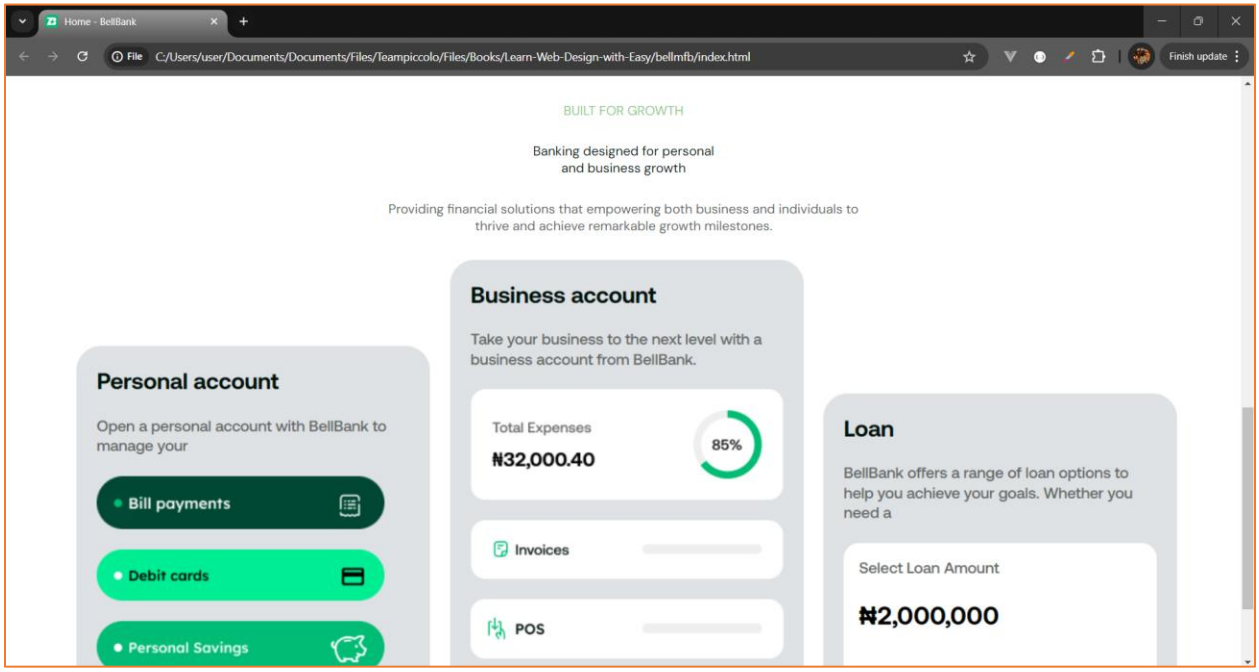


Figure 144: Styling account types section X

8. Next, let's create a custom CSS class that will contain the font-size, we will be creating two classes, as shown below:

```css
.text-heading-2{
    font-size: 2em;
}

.-text-heading-1{
    font-size: 0.8em;
}
```

Figure 145: Styling account types section XI

9. Update the HTML code snippet, as shown below:

```html
<!-- accountTypeSection  -->
<div id="accountTypeSection">
    <div class="green-text padding-y-2 -text-heading-1">
        <b>BUILT FOR GROWTH</b>
    </div>
    <div class="black-text padding-y-2 text-heading-2">
        <b>Banking designed for personal <br>
        and business growth</b>
    </div>
    <div class="grey-text padding-y-2 -text-heading-1">
        <b>Providing financial solutions that empowering both business and individuals to <br>
        thrive and achieve remarkable growth milestones.</b>
    </div>
    <div>
        <img src="./images/personal_account_icon.png" alt="Personal Account">
        <img src="./images/business_account_icon.png" alt="Business Account">
        <img src="./images/loan_icon.png" alt="Loan Icon">
    </div>
</div>
```

Figure 146: Styling account types section XII

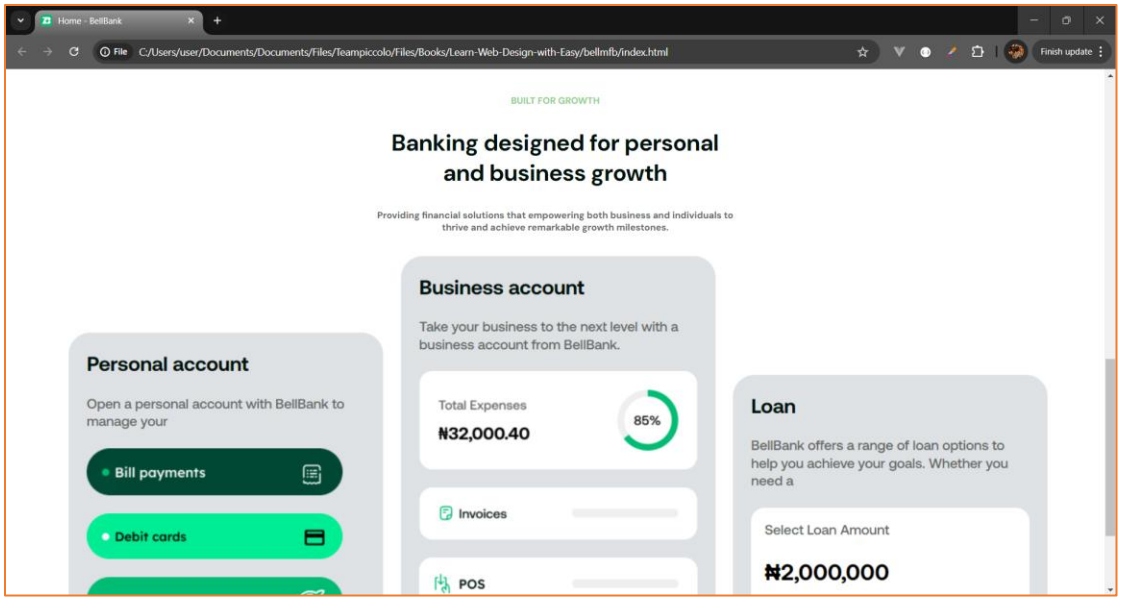


Figure 147: Styling account types section XIII

10. Next, let's work on the images. We will reduce their size using the CSS width property and align the images to the center. Add an ID attribute to the div nesting the images, as shown below:

```
        <div id="accountTypeSectionImage">
            <img src="./images/personal_account_icon.png" alt="Personal Account">
            <img src="./images/business_account_icon.png" alt="Business Account">
            <img src="./images/loan_icon.png" alt="Loan Icon">
        </div>
    </div>
</section>
```

Figure 148: Styling account types section XIV

11. Next, add the following CSS properties to the div, as shown below:

```
#accountTypeSectionImage{
    display: flex;
    justify-content: center;
    align-items: center;
}

#accountTypeSectionImage img{
    width: 25%;
}
```

Figure 149: Styling account types section XV

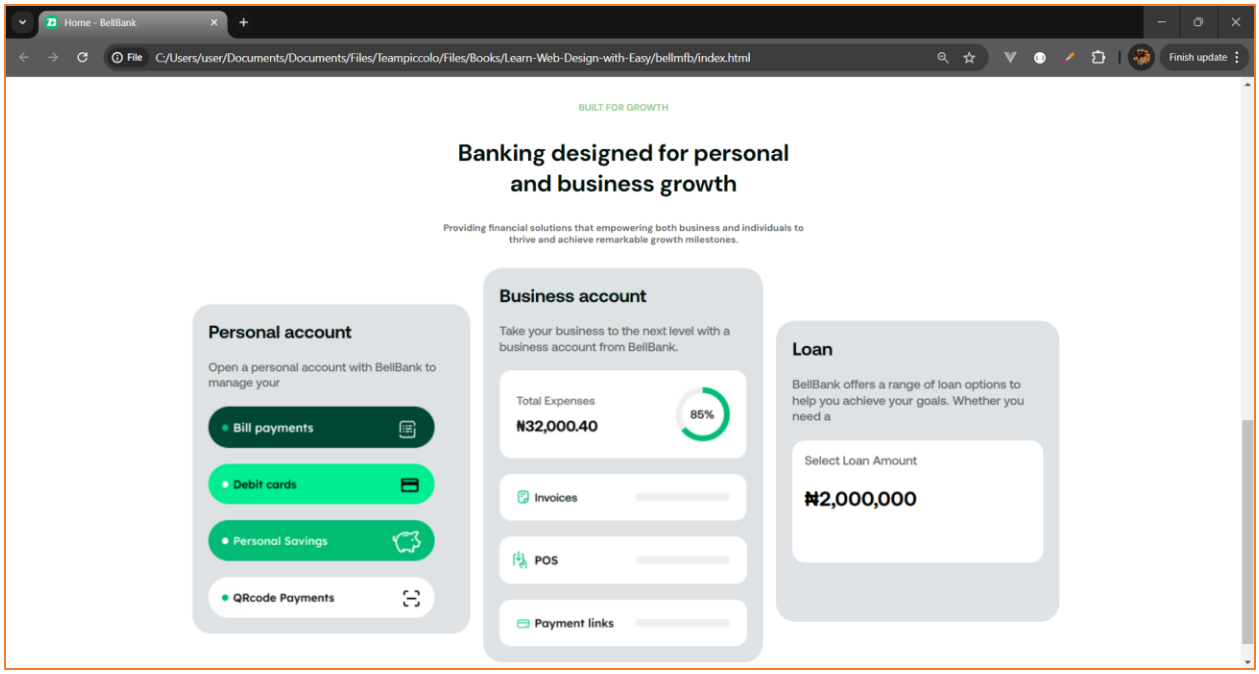


Figure 150: Styling account types section XVI

Styling Features Section

1. Let's create a div inside the "pageSection" element after the "accountTypeSection" div and add an ID attribute to it with a value of "websiteFeatures" as shown below:

```
<!-- pageSection  -->
<section id="pageSection">
    <!-- License and Regulators  -->
    <div id="licenseDiv">···
    </div>
    <!-- accountTypeSection  -->
    <div id="accountTypeSection">···
    </div>
    <!-- websiteFeatures -->
    <div id="websiteFeatures">


    </div>
</section>
```

Figure 151: Styling features section

2. Next, let's create the contents of the div. Below is the code snippet:

```
<!-- websiteFeatures -->
<div id="websiteFeatures">
    <div class="green-text padding-y-2 -text-heading-1">
        <b>FEATURES</b>
    </div>
    <div class="black-text padding-y-2 text-heading-2">
        <b>Experience new banking</b>
    </div>
    <div class="grey-text padding-y-2 -text-heading-1">
        <b>Providing financial solutions that empowering both business and individuals to <br>
        thrive and achieve remarkable growth milestones.</b>
    </div>
</div>
```

Figure 152: Styling features section II

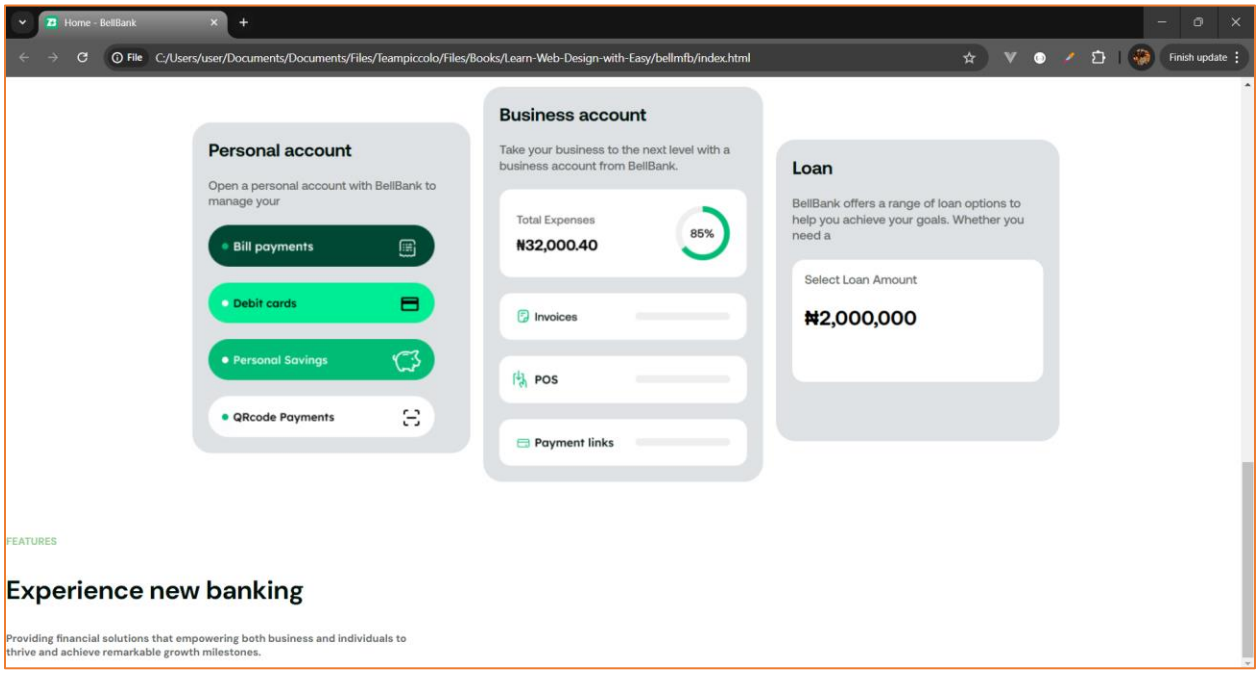


Figure 153: Styling features section III

3. Next, we will learn about the CSS group selector. The group selector allows you to apply the same style definitions to multiple HTML elements. For example, we want the "websiteFeatures" and "accountTypeSection" divs to share the same CSS properties. Since we've already written the CSS for the "accountTypeSection" div, there's no need to rewrite the same code for the "websiteFeatures" div. Instead, we will use the CSS group selector, as shown below:

```
#accountTypeSection, #websiteFeatures{
    text-align: center;
    padding: 2%;
}
```

Figure 154: Styling features section IV

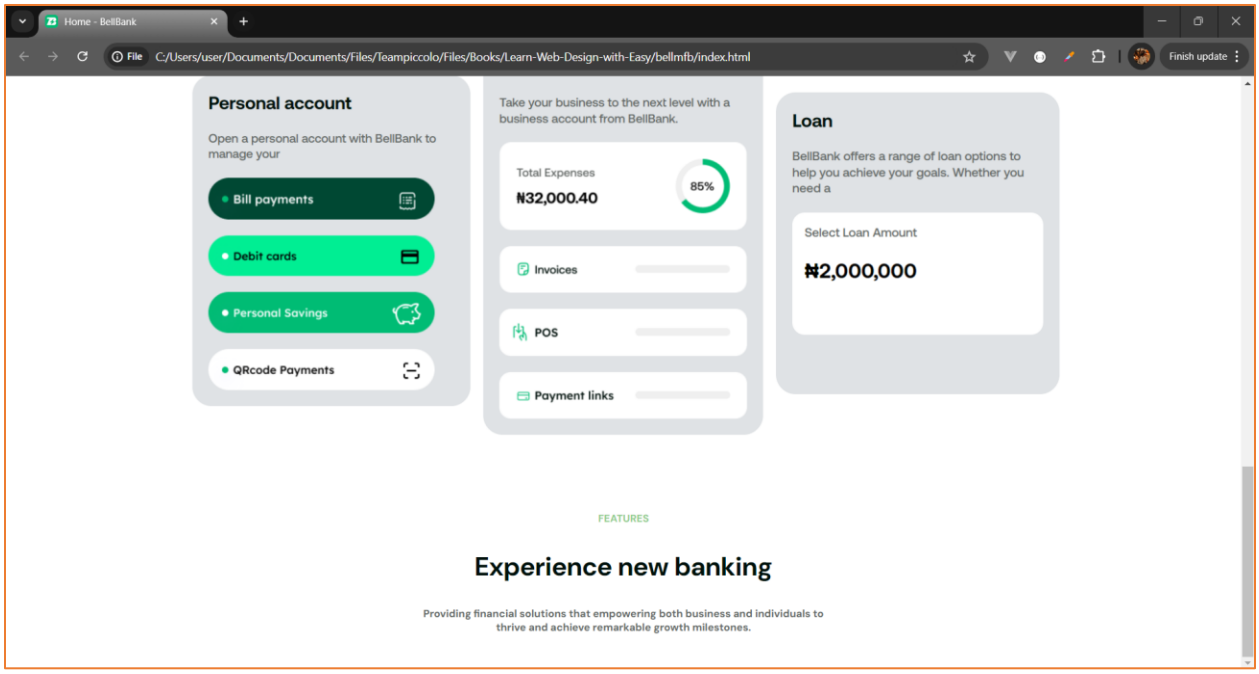


Figure 155: Styling features section V

Styling Debit Card Section

1. Let's create a div inside the "pageSection" element after the "websiteFeatures" div and add an ID attribute to it with a value of "debitCard" as shown below:

```
    <!-- websiteFeatures -->
    <div id="websiteFeatures">···
    </div>
    <!-- debitCardSection -->
    <div id="debitCard">

    </div>
</section>
```

Figure 156: Styling Debit Card Section

2. Next, let's create the contents of the div. Below is the code snippet:

```
<!-- debitCardSection -->
<div id="debitCard">
    <div>
        <div class="green-text padding-y-2 -text-heading-1">
            <b>DEBIT CARD</b>
        </div>
        <div class="black-text padding-y-2 text-heading-2">
            <b>Instant Debit cards that <br>
            always work</b>
        </div>
        <div class="grey-text padding-y-2 -text-heading-1">
            <b>
                If transfers aren't your jam, we get it. Request a debit card and <br>
                have it delivered to you within 48 hours. Activate it in minutes, and <br>
                start using it right away.
            </b>
        </div>
        <div>
            <button>Get Started</button>
        </div>
    </div>
    <div>
        <img src="./images/debit_card_icon.png" alt="Debit Card Icon">
    </div>
</div>
```

Figure 157: Styling Debit Card Section II

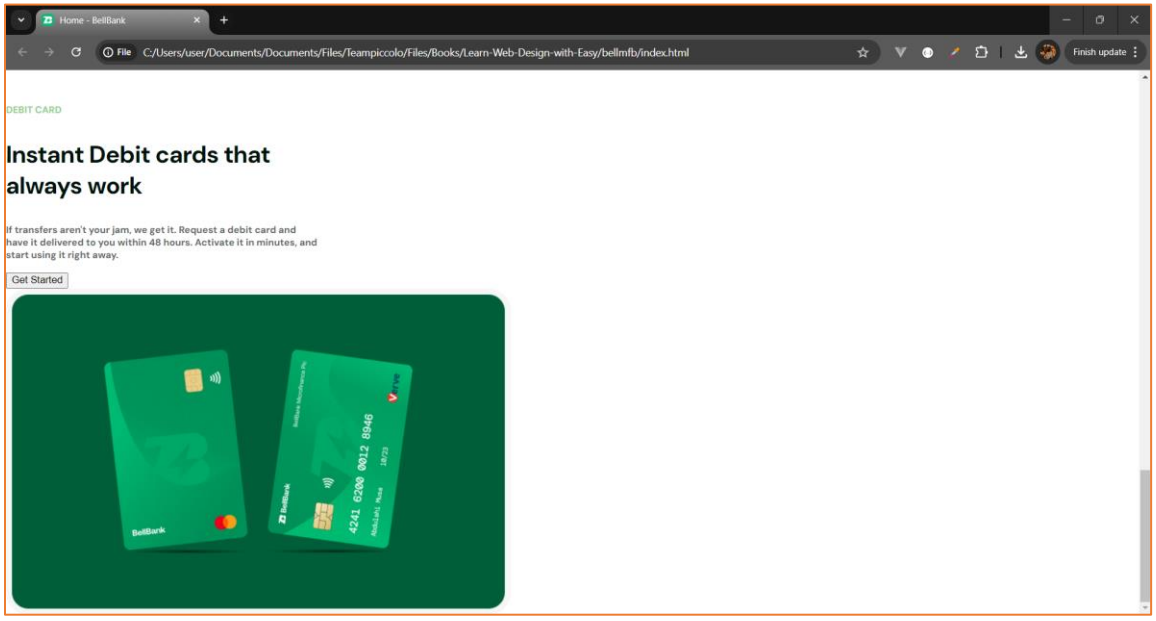


Figure 158: Styling Debit Card Section II

3. Next, let's add more CSS properties to the "debitCard" div in the style.css file, as shown below:

```css
#debitCard{
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-template-rows: auto;
    gap: 20px;
    align-items: center;
    padding: 2%;
}
```

Figure 159: Styling Debit Card Section III

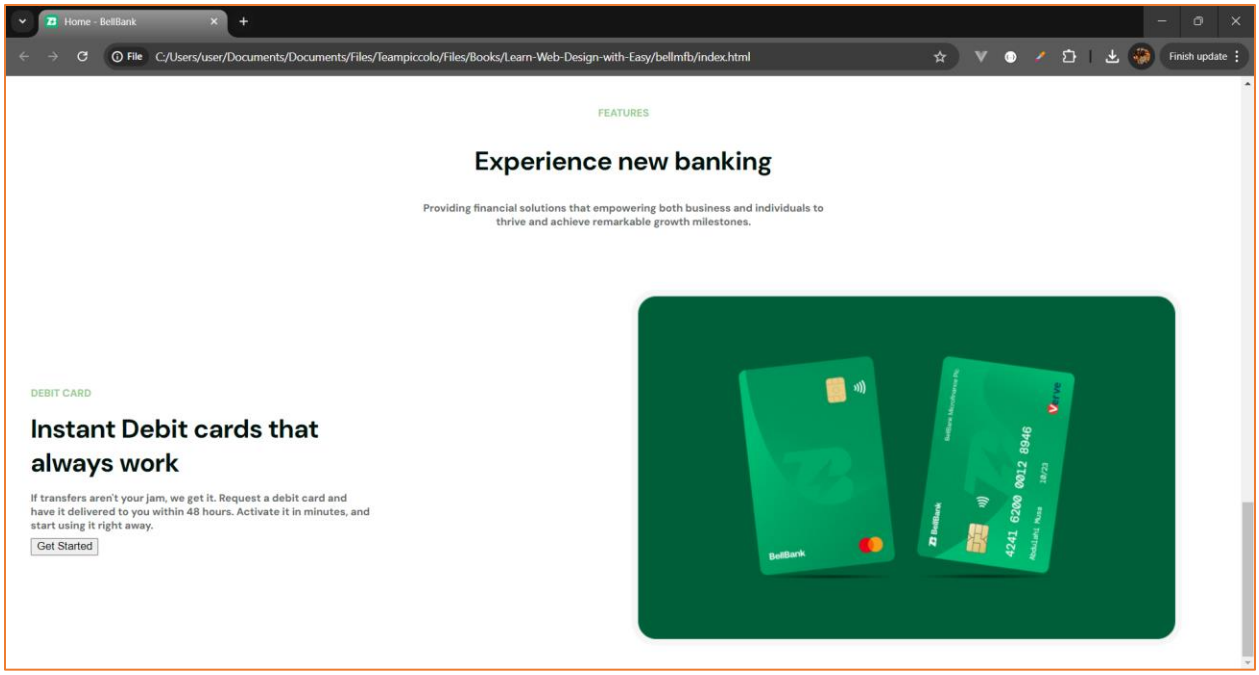


Figure 160: Styling Debit Card Section III

4. Next, let's improve the debit card write-up section and style the button. We will refactor the HTML code and add "**content-text**" class to the div containing the "Debit Card" write-up, add "**padding-y-2**" class to the div containing the button; we will create a new class "**get-start-btn**" class to the button element, below is the update code snippet;

```
<!-- debitCardSection -->
<div id="debitCard">
    <div class="context-text">
        <div class="green-text padding-y-2 -text-heading-1">
            <b>DEBIT CARD</b>
        </div>
        <div class="black-text padding-y-2 text-heading-2">
            <b>Instant Debit cards that <br>
            always work</b>
        </div>
        <div class="grey-text padding-y-2 -text-heading-1">
            <b>
                If transfers aren't your jam, we get it. Request a debit card and <br>
                have it delivered to you within 48 hours. Activate it in minutes, and <br>
                start using it right away.
            </b>
        </div>
        <div class="padding-y-2">
            <button class="get-start-btn">Get Started</button>
        </div>
    </div>
    <div>
        <img src="./images/debit_card_icon.png" alt="Debit Card Icon">
    </div>
</div>
```

Figure 161: Styling Debit Card Section IV

5. Next, let's update the CSS properties, as shown below:

```
#debitCard{
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-template-rows: auto;
    gap: 20px;
    align-items: center;
    padding: 2%;
    margin: 2% 10%;
}

#debitCard img{
    width: 90%;
}

.get-start-btn{
    background-color: #00bc74;
    color: white;
    padding: 1.8% 8%;
    border: 1px solid #00bc74;
    border-radius: 5px;
}

.context-text{
    margin-left: 25%;
}
```

Figure 162: Styling Debit Card Section V

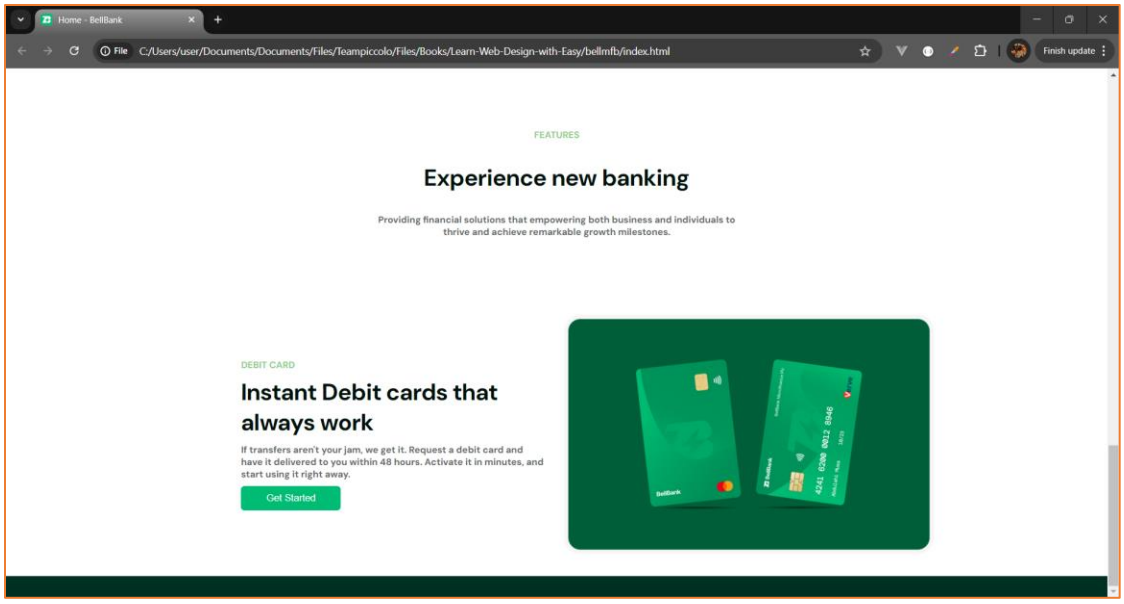


Figure 163: Styling Debit Card Section VI

The UI/UX design is much better now 😊

Styling Personal Account Section

1. Let's create a div inside the "pageSection" element after the "debitCard" div and add an ID attribute to it with a value of "personalAccount" as shown below:

```
<!-- pageSection  -->
<section id="pageSection">
    <!-- License and Regulators  -->
    <div id="licenseDiv">···
    </div>
    <!-- accountTypeSection  -->
    <div id="accountTypeSection">···
    </div>
    <!-- websiteFeatures -->
    <div id="websiteFeatures">···
    </div>
    <!-- debitCardSection -->
    <div id="debitCard">···
    </div>
    <!-- PersonalAccountSection  -->
    <div id="personalAccount">


    </div>
</section>
```

Figure 164: Styling Personal Account Section

2. Next, let's create the contents of the div. Below is the code snippet:

```
<!-- PersonalAccountSection  -->
<div id="personalAccount">
    <div class="context-text">
        <div class="green-text padding-y-2 -text-heading-1">
            <b>PERSONAL ACCOUNT</b>
        </div>
        <div class="black-text padding-y-2 text-heading-2">
            <b>One app for all your <br>
            banking needs</b>
        </div>
        <div class="grey-text padding-y-2 -text-heading-1">
            <b>
                If transfers aren't your jam, we get it. Request a debit card and <br>
                have it delivered to you within 48 hours. Activate it in minutes, and <br>
                start using it right away.
            </b>
        </div>
        <div class="padding-y-2">
            <button class="get-start-btn">Get Started</button>
        </div>
    </div>
    <div>
        <img src="./images/personal_account_section.png" alt="Personal Account Icon">
    </div>
</div>
```

Figure 165: Styling Personal Account Section II

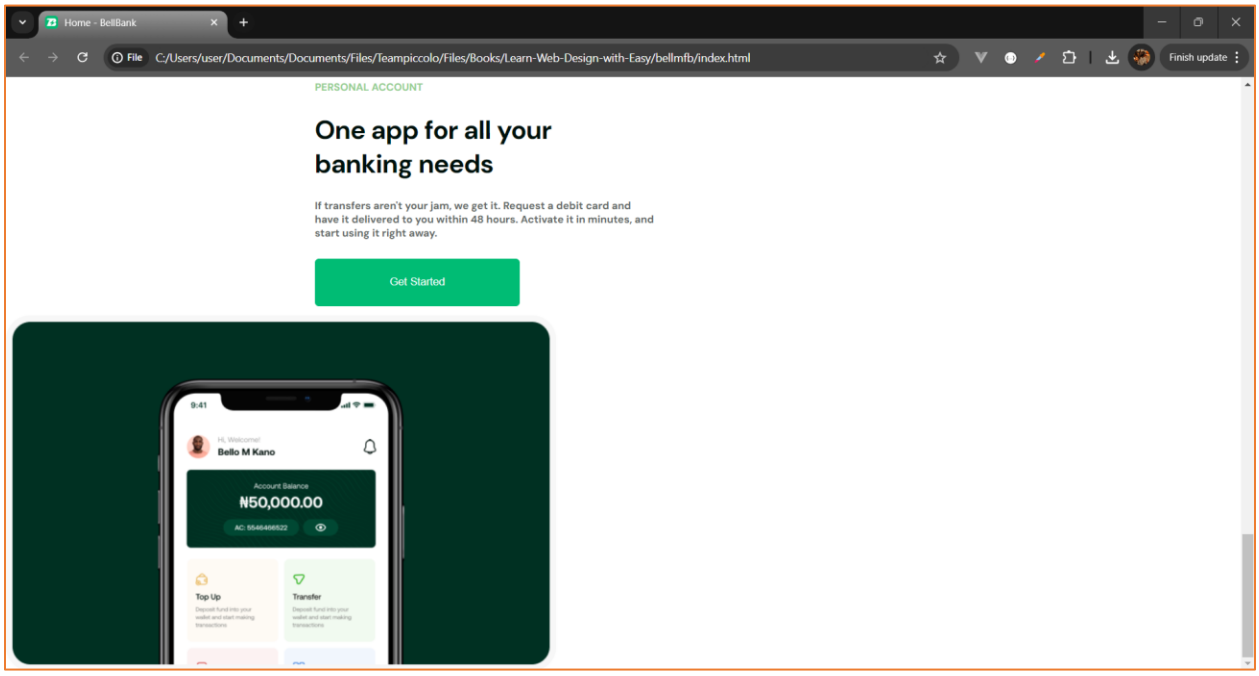


Figure 166: Styling Personal Account Section III

3. Next, let's add CSS properties to the personalAccount div in the style.css file. Since we've learned about the CSS group selector, we'll use it to reduce code redundancy on the website, as shown below:

```css
#debitCard, #personalAccount{
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-template-rows: auto;
    gap: 20px;
    align-items: center;
    padding: 2%;
    margin: 2% 10%;
}

#debitCard img, #personalAccount img{
    width: 90%;
}
```

Figure 167: Styling Personal Account Section IV

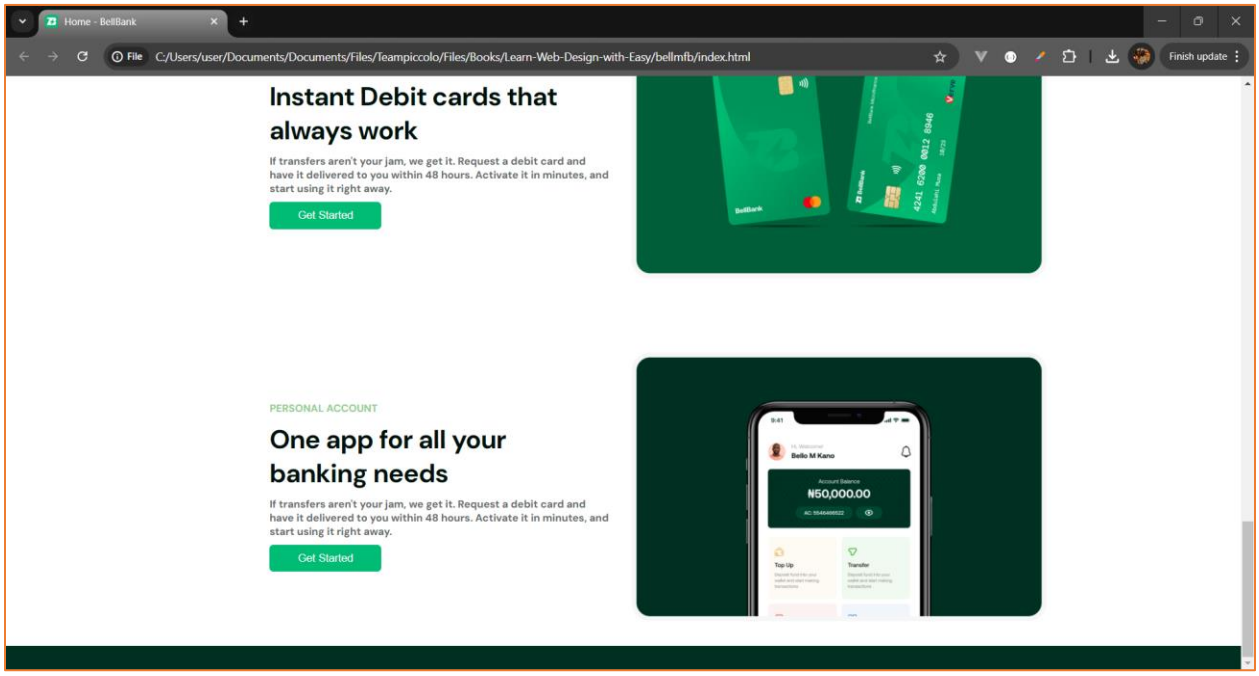


Figure 168: Styling Personal Account Section V

Styling Modular Solution Section

1. Let's create a div inside the "pageSection" element after the "personalAccount" div and add an ID attribute to it with a value of "modularSolution" as shown below:

```
<!-- pageSection  -->
<section id="pageSection">
    <!-- License and Regulators  -->
    <div id="licenseDiv">···
    </div>
    <!-- accountTypeSection  -->
    <div id="accountTypeSection">···
    </div>
    <!-- websiteFeatures -->
    <div id="websiteFeatures">···
    </div>
    <!-- debitCardSection -->
    <div id="debitCard">···
    </div>
    <!-- PersonalAccountSection  -->
    <div id="personalAccount">···
    </div>
    <!-- ModularSolutionSection  -->
    <div id="modularSolution">

    </div>
</section>
```

Figure 169: Styling Modular Solution Section

2. Next, let's create the contents of the div. Below is the code snippet:

```
<!-- ModularSolutionSection  -->
<div id="modularSolution">
    <div class="context-text">
        <div class="green-text padding-y-2 -text-heading-1">
            <b>MODULAR SOLUTION</b>
        </div>
        <div class="black-text padding-y-2 text-heading-2">
            <b>
                A fully integrated suite of <br>
                financial and payments <br>
                products
            </b>
        </div>
        <div class="grey-text padding-y-2 -text-heading-1">
            <b>
                If transfers aren't your jam, we get it. Request a debit card and <br>
                have it delivered to you within 48 hours. Activate it in minutes, and <br>
                start using it right away.
            </b>
        </div>
        <div class="padding-y-2">
            <button class="get-start-btn">Get Started</button>
        </div>
    </div>
    <div>
        <img src="./images/modular_solution_icon.png" alt="Modular Solution Icon">
    </div>
</div>
```

Figure 170: Styling Modular Solution Section II

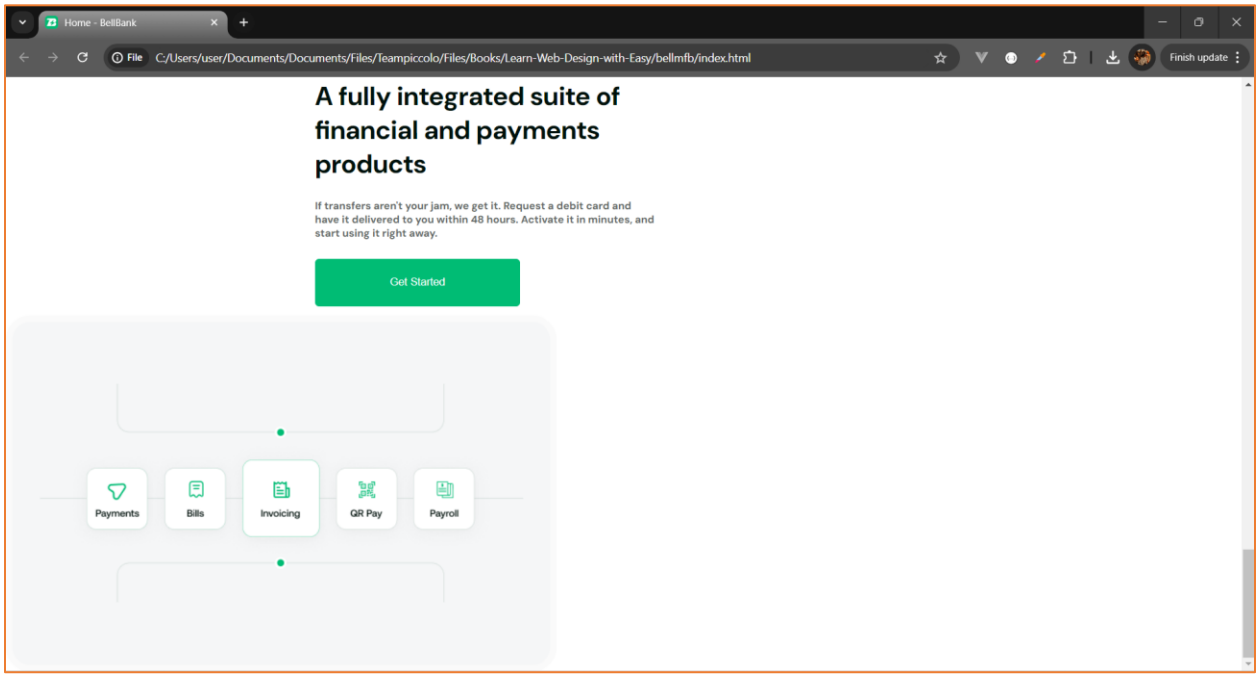


Figure 171: Styling Modular Solution Section III

3. Next, let's update the style.css file and add CSS properties to the div, we will be using the same method we did above, as shown below;

```css
#debitCard, #personalAccount, #modularSolution{
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-template-rows: auto;
    gap: 20px;
    align-items: center;
    padding: 2%;
    margin: 2% 10%;
}

#debitCard img, #personalAccount img, #modularSolution img{
    width: 90%;
}
```

Figure 172: Styling Modular Solution Section IV

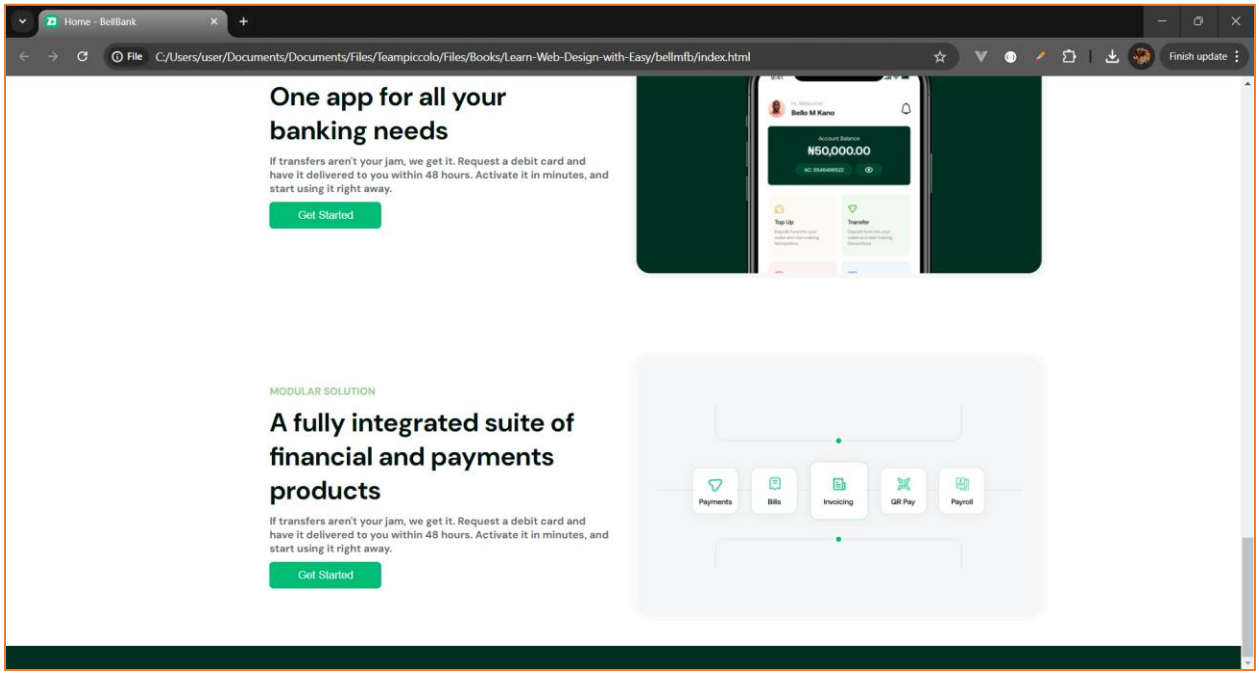


Figure 173: Styling Modular Solution Section V

Styling Why Us Section

1. Let's create a div inside the "pageSection" element after the "modularSolution" div and add an ID attribute to it with a value of "whyUs" as shown below:

```
<section id="pageSection">
    </div>
    <!-- debitCardSection -->
    <div id="debitCard">···
    </div>
    <!-- PersonalAccountSection  -->
    <div id="personalAccount">···
    </div>
    <!-- ModularSolutionSection  -->
    <div id="modularSolution">···
    </div>
    <!-- WhyUsSection  -->
    <div id="whyUs">

    </div>
</section>
```

Figure 174: Styling Why Us Section

2. Next, let's create the contents of the div. Below is the code snippet:

```
<!-- WhyUsSection  -->
<div id="whyUs">
    <!-- First  -->
    <div>
        <div>
            <b>Protecting and powering growth for businesses</b>
        </div>
        <div>
            We provide secure and efficient financial solutions,
            helping you grow with confidence. Our tailored business accounts streamline your finances,
            empower your team, and fuel your success.
        </div>
    </div>

</div>
```

Figure 175: Styling Why Us Section II

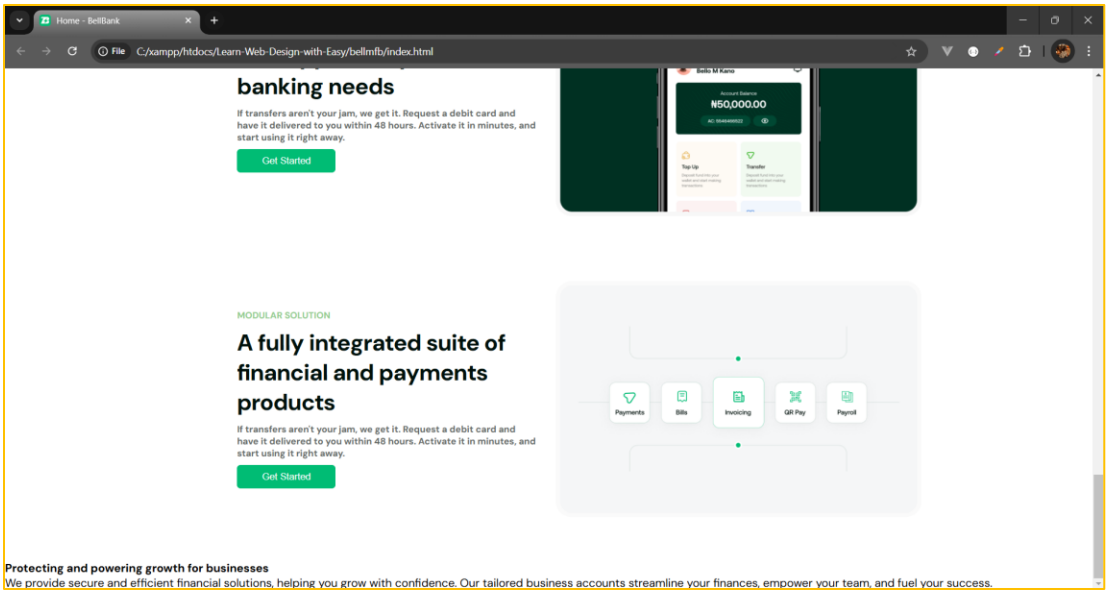


Figure 176: Styling Why Us Section III

3. Next, let's create another div after the first div we created above. Below is the code snippet:

```
<!-- Second  -->
<div>
    <div>
        <!-- Circle  -->
        <div>
            <img src="./images/green_circle_icon.png" alt="Green Circle Icon">
        </div>
        <!-- Heading  -->
        <div>
            <b>Bank-level security</b>
        </div>
        <!-- Text  -->
        <div>
            Fully Licensed by CBN and Insured by NDIC, Be rest assured with our robust and reliable security measures, protecting your business and finances.
        </div>
    </div>
    <div>
        <!-- Circle  -->
        <div>
            <img src="./images/green_circle_icon.png" alt="Green Circle Icon">
        </div>
        <!-- Heading  -->
        <div>
            <b>Regulatory Compliance</b>
        </div>
        <!-- Text  -->
        <div>
            Trust that we're always up-to-date with the latest regulations, ensuring your business is too.
        </div>
    </div>
    <div>
        <!-- Circle  -->
        <div>
            <img src="./images/black_circle_icon.png" alt="Black Circle Icon">
        </div>
        <!-- Heading  -->
        <div>
            <b>White-glove support</b>
        </div>
        <!-- Text  -->
        <div>
            Enjoy personalized, premium support from our expert team, dedicated to your success.
        </div>
    </div>
</div>
```

Figure 177: Styling Why Us Section IV

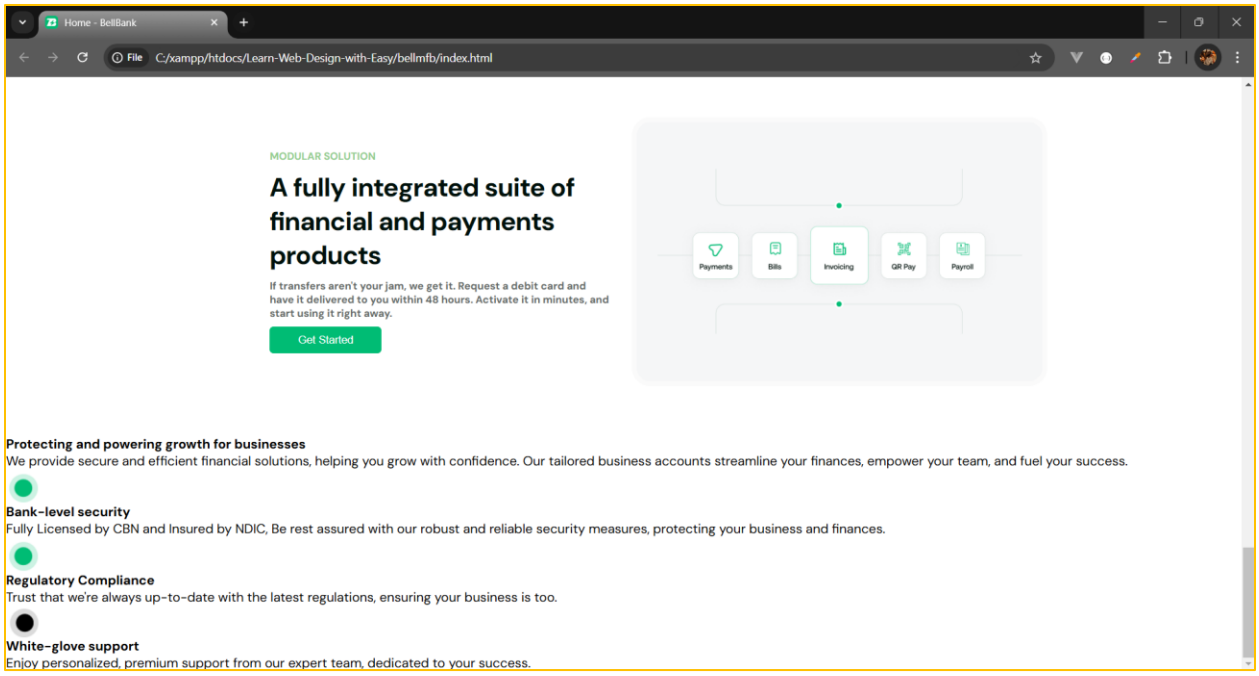


Figure 178: Styling Why Us Section V

4. If you notice, there are two divs nested inside the whyUs section div, and each of these divs contains additional nested divs. Let's start by styling the first div. We will make it a two-column layout using CSS Grid. Add a class attribute to the first div with the value "column-2", as shown below:

```
<!-- WhyUsSection  -->
<div id="whyUs">
    <!-- First  -->
    <div class="column-2">
        <div>
            <b>Protecting and powering growth for businesses</b>
        </div>
        <div>
            We provide secure and efficient financial solutions,
            helping you grow with confidence. Our tailored business accounts streamline your finances,
            empower your team, and fuel your success.
        </div>
    </div>
    <!-- Second  -->
```

Figure 179: Styling Why Us Section VI

5. Add the following CSS properties to the class "column-2" in the style.css, as shown below:

```css
.column-2{
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-template-rows: auto;
    gap: 20px;
    align-items: center;
}
```

Figure 180: Styling Why Us Section VII

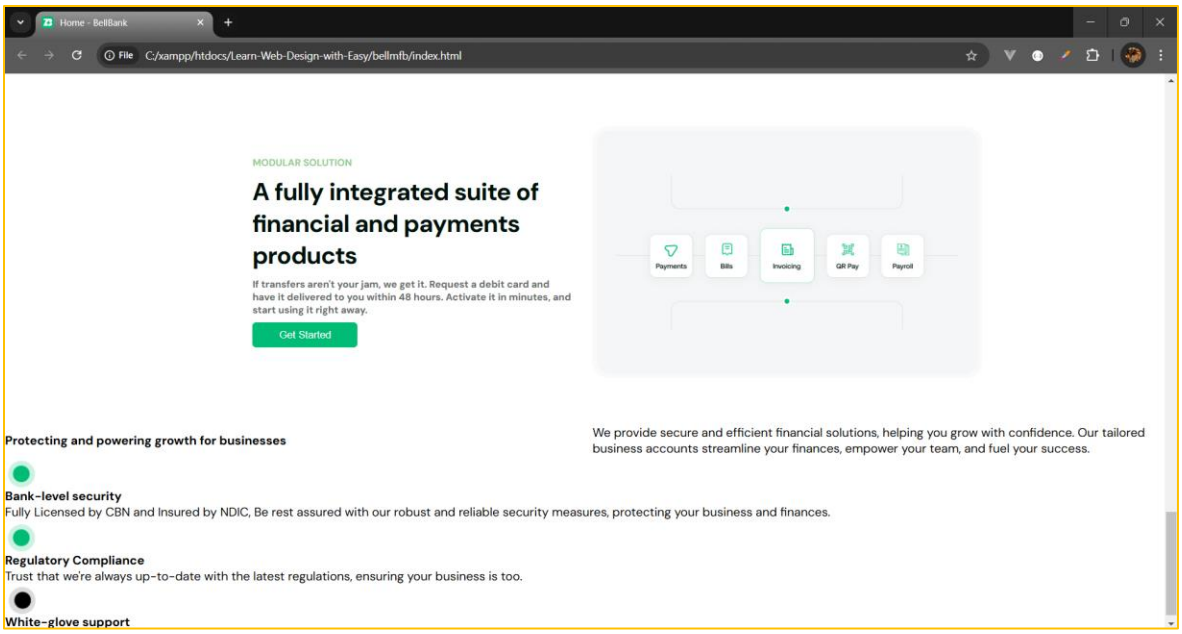


Figure 181: Styling Why Us Section VIII

6. Next, let's add custom CSS classes that we created earlier to improve the design, below is the code snippet:

```html
<!-- First  -->
<div class="column-2">
    <div class="text-heading-2">
        <b>Protecting and powering growth <br> for businesses</b>
    </div>
    <div class="-text-heading-1">
        We provide secure and efficient financial solutions,
        helping you grow with confidence. Our tailored business accounts streamline your finances,
        empower your team, and fuel your success.
    </div>
</div>
<!-- Second  -->
```

Figure 182: Styling Why Us Section IX

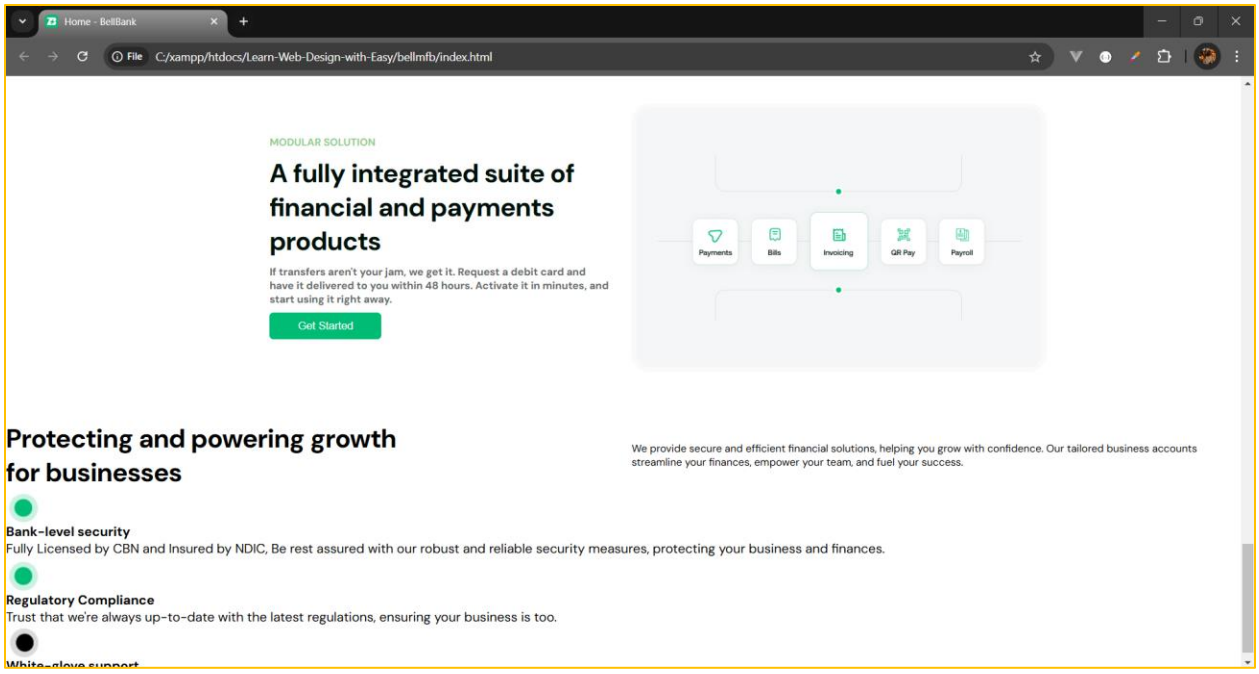


Figure 183: Styling Why Us Section X

7. Next, let's create a custom CSS class called "margin-x-10" in the style.css file to add margin to HTML elements. Below is the code snippet:

```css
.margin-x-10{
    margin-left: 10%;
    margin-right: 10%;
}
```

Figure 184: Styling Why Us Section XI

8. Next, add the custom CSS class "margin-x-10" to the "whyUs" div, as shown below:

```html
<!-- WhyUsSection  -->
<div id="whyUs" class="margin-x-10">
    <!-- First  -->
    <div class="column-2">
        <div class="text-heading-2">
            <b>Protecting and powering growth <br> for businesses</b>
        </div>
        <div class="-text-heading-1">
            We provide secure and efficient financial solutions,
            helping you grow with confidence. Our tailored business accounts streamline your finances,
            empower your team, and fuel your success.
        </div>
    </div>
```

Figure 185: Styling Why Us Section XII

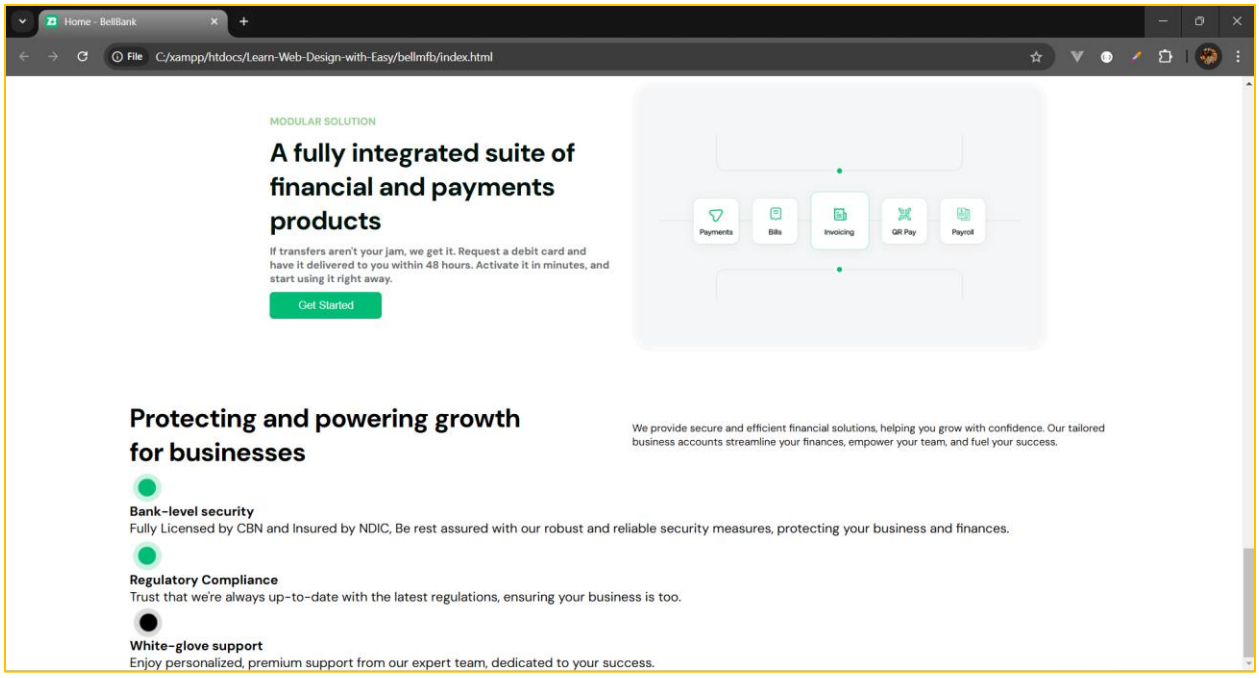


Figure 186: Styling Why Us Section XIII

9. Now, let's work on the second div. We will make it a three-column layout using CSS Grid. Add a class attribute to the second div with the value "column-3", as shown below.

```
            We provide secure and efficient financial solutic
            helping you grow with confidence. Our tailored bu
            empower your team, and fuel your success.
        </div>
    </div>
    <!-- Second  -->
    <div class="column-3">
        <div>
            <!-- Circle  -->
            <div>
                <img src="./images/green_circle_icon.png" alt
            </div>
```

Figure 187: Styling Why Us Section XIV

10. Add the following CSS properties to the class "column-3" in the style.css, as shown below:

```css
.column-3{
    display: grid;
    grid-template-columns: 1fr 1fr 1fr;
    grid-template-rows: auto;
    gap: 20px;
    align-items: center;
}
```

Figure 188: Styling Why Us Section XV

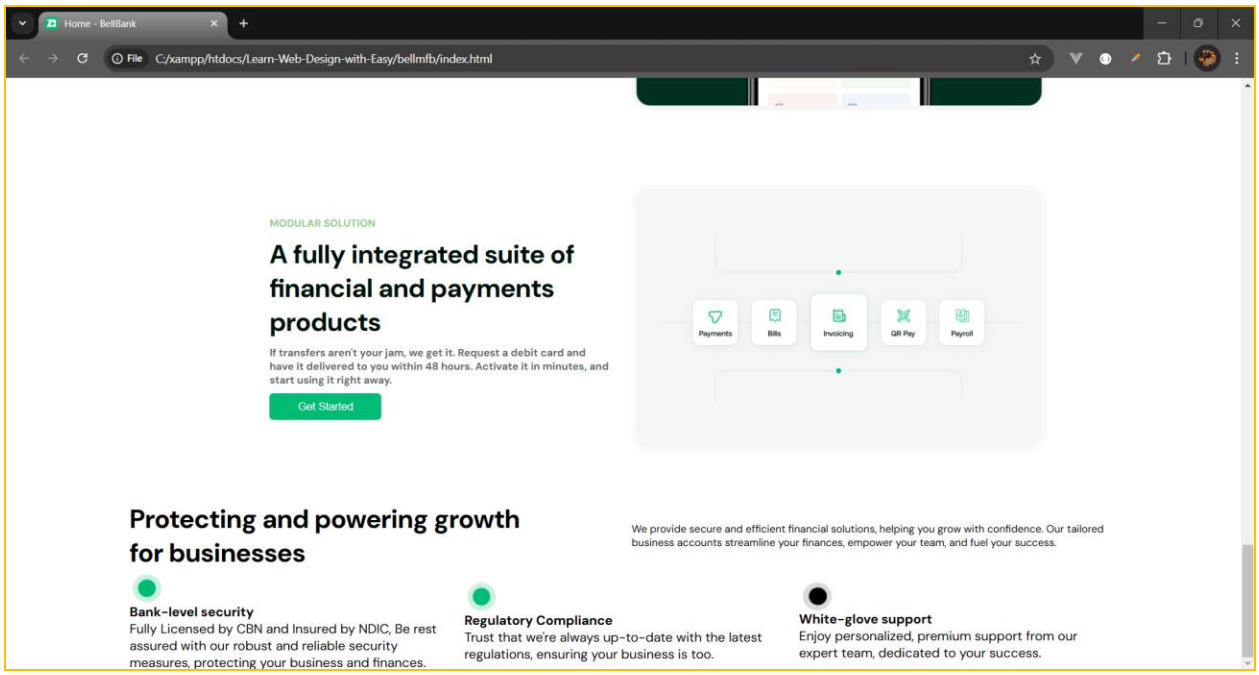


Figure 189: Styling Why Us Section XVI

12. Next, let's create a custom CSS class called "padding-y-2" in the style.css file to add padding to HTML elements. Below is the code snippet:

```css
.padding-y-2{
    padding-top: 2%;
    padding-bottom: 2%;
}
```

Figure 190: Styling Why Us Section XVII

13. Next, add the custom CSS class “padding-y-2” to the child divs of the second div, as shown below:

```
<!-- Second  -->
<div class="column-3">
    <div>
        <!-- Circle  -->
        <div class="padding-y-2">
            <img src="./images/green_circle_icon.png" alt="Green Circle
        </div>
        <!-- Heading  -->
        <div class="padding-y-2">
            <b>Bank-level security</b>
        </div>
        <!-- Text  -->
        <div class="padding-y-2">
            Fully Licensed by CBN and Insured by NDIC, Be rest assured
        </div>
    </div>
    <div>
```

Figure 191: Styling Why Us Section XVIII

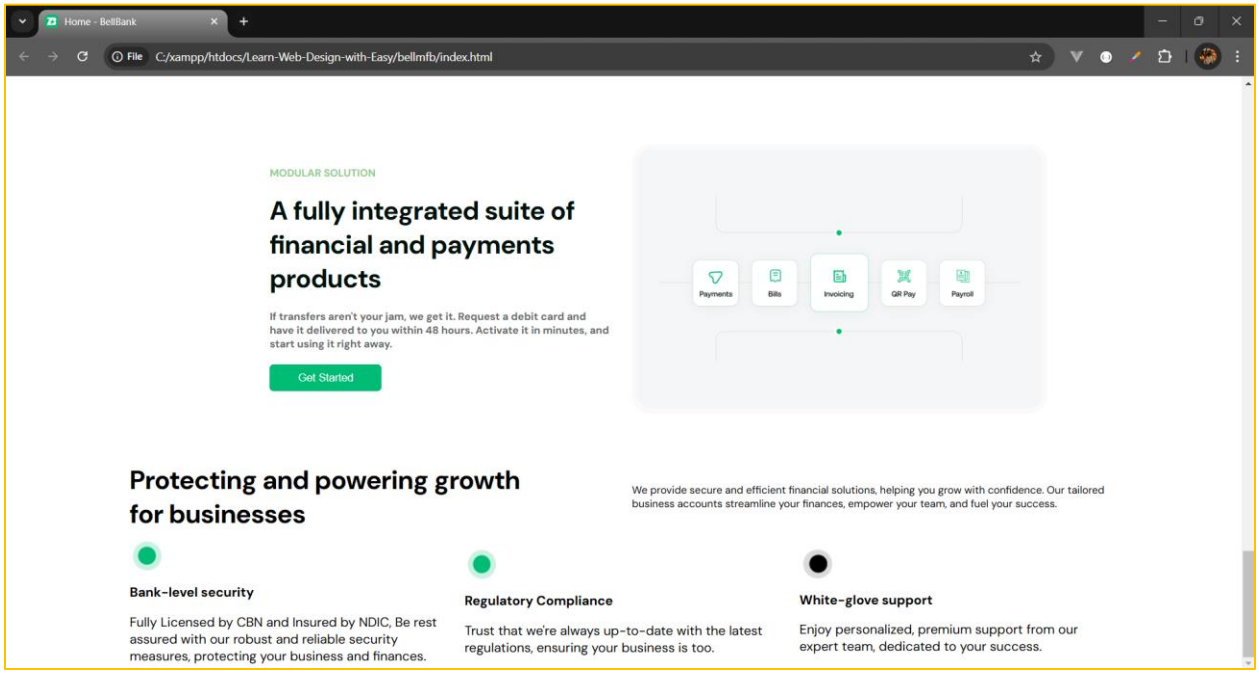


Figure 192: Styling Why Us Section XIX

14. Next, let's create a custom CSS class called "margin-y-2" in the style.css file to add margin to HTML elements. Below is the code snippet:

```
.margin-y-2{
    margin-top: 2%;
    margin-bottom: 2%;
}
```

Figure 193: Styling Why Us Section XX

15. Next, add the custom CSS class "margin-y-2" to both the first and second div within the whyUs div. Below is the code snippet;

```
<!-- WhyUsSection  -->
<div id="whyUs" class="margin-x-10">
    <!-- First  -->
    <div class="column-2 margin-y-2">···
    </div>
    <!-- Second  -->
    <div class="column-3 margin-y-2">···
    </div>
</div>
```

Figure 194: Styling Why Us Section XXI

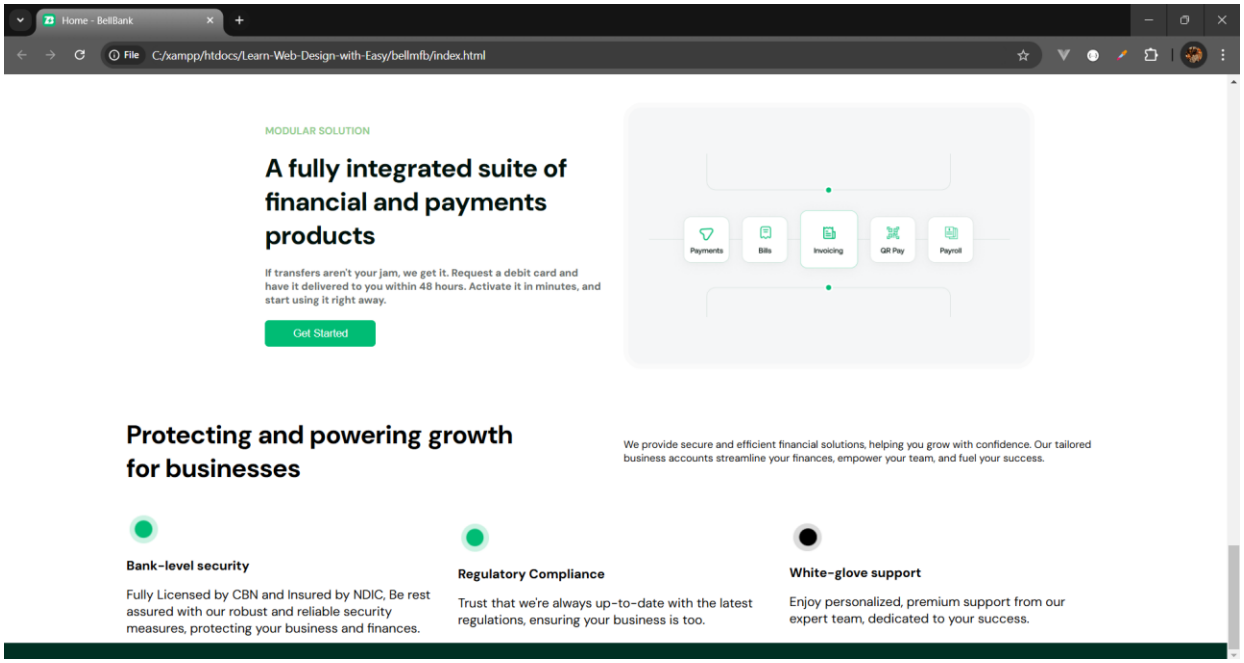


Figure 195: Styling Why Us Section XXII

Exercise 3 (Practical)

1. Create a custom CSS class called padding-y-5 and give it padding-top of 5% and padding-bottom of 5%.
2. Change the font-color of the headings “Bank-level security” and “Regulatory Compliance” to custom CSS class “text-green”.
3. Add the custom CSS class “padding-y-5” to the whyUs div.

Answer

1. Solution

```
.padding-y-5{
    padding-top: 5%;
    padding-bottom: 5%;
}
```

Figure 196: Exercise Solution I

2. Solution

```
        <!-- Heading  -->
        <div class="padding-y-2 green-text">
            <b>Bank-level security</b>
        </div>
        <!-- Text  -->
        <div class="padding-y-2">
            Fully Licensed by CBN and Insured by NDIC, Be rest
        </div>
    </div>
    <div>
        <!-- Circle  -->
        <div class="padding-y-2">
            <img src="./images/green_circle_icon.png" alt="Gree
        </div>
        <!-- Heading  -->
        <div class="padding-y-2 green-text">
            <b>Regulatory Compliance</b>
        </div>
        <!-- Text  -->
        <div class="padding-y-2">
            Trust that we're always up-to-date with the latest
        </div>
```

Figure 197: Exercise Solution II

3. Solution

```
<!-- WhyUsSection  -->
<div id="whyUs" class="margin-x-10 padding-y-5">
    <!-- First  -->
    <div class="column-2 margin-y-2">···
    </div>
    <!-- Second  -->
    <div class="column-3 margin-y-2">···
    </div>
</div>
```

Figure 198: Exercise Solution III

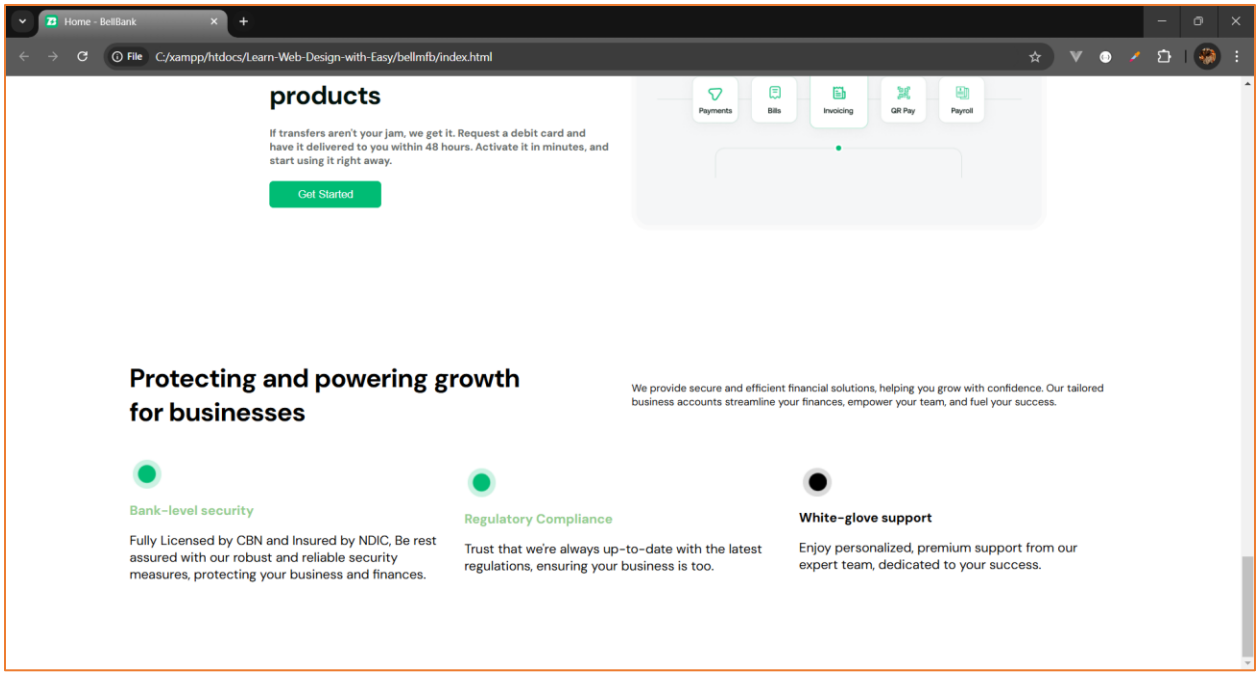


Figure 199: Styling Why Us Section XXIII

Styling Download App Section

1. Let's create a div inside the "pageSection" element after the "whyUs" div and add an ID attribute to it with a value of "downloadApp" as shown below:

```
    <!-- WhyUsSection  -->
    <div id="whyUs" class="margin-x-10 padding-y-5">…
    </div>
    <!-- DownloadAppSection  -->
    <div id="downloadApp">


    </div>
</section>
```

Figure 200: Styling Download App Section

2. Next, let's create the contents of the div. Below is the code snippet:

```
<!-- DownloadAppSection  -->
<div id="downloadApp">
    <div>
        <div class="text-heading-2">
            Get an account that <br> suits you.
        </div>
        <div class="-text-heading-1">
            At BellBank, we understand that every individual and business is <br>
            unique. That's why we offer a range of accounts tailored to suit <br>
            your specific needs. Whether you're looking for a personal <br>
            account, a business account, or something in between, we've got <br>
            you covered.
        </div>
        <!-- Download Icon  -->
        <div>
            <span class="downloadIcon"><img src="./images/apple_white_icon.png" alt="Apple Icon"></span>
            <span class="downloadIcon"><img src="./images/playstore_white_icon.png" alt="Android Icon"></span>
        </div>
    </div>
</div>
```

Figure 201: Styling Download App Section II

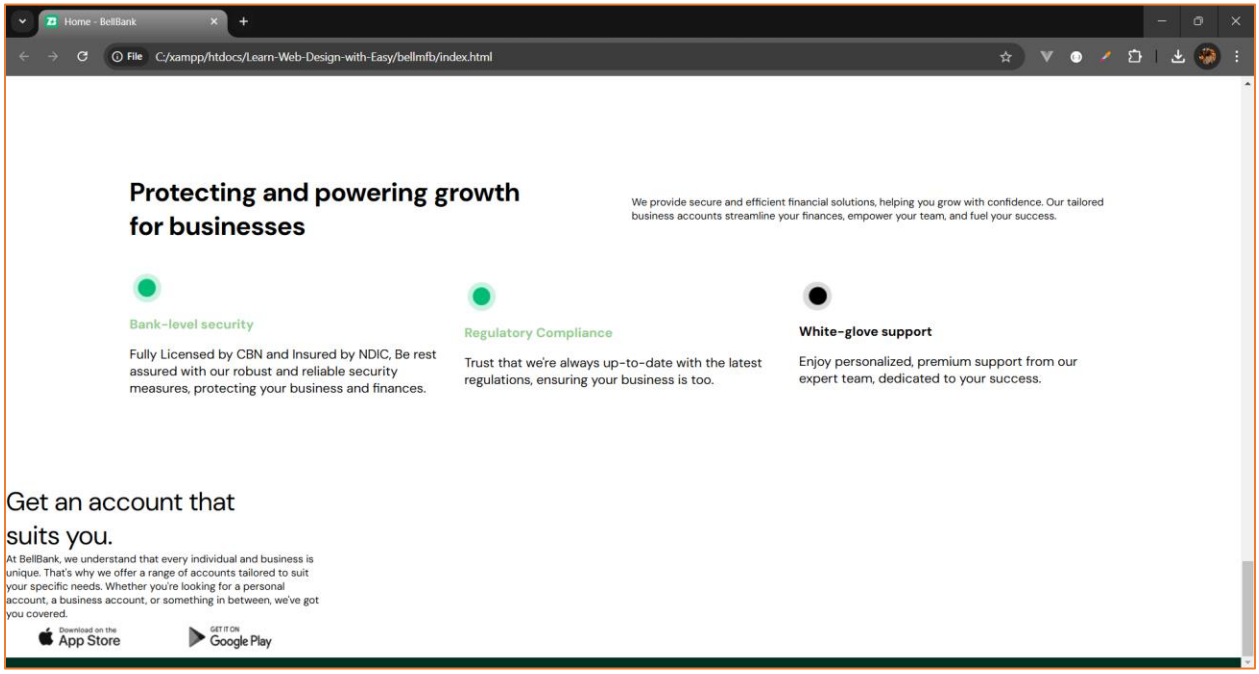


Figure 202: Styling Download App Section III

3. Add the following CSS properties to the “downloadApp” section in the style.css, as shown below:

```css
#downloadApp{
    width: 60%;
    background-image: url('../images/bg_one.png');
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center;
    margin: 2% auto;
    padding: 2% 5%;
    color: white;
    border-radius: 25px;
}
```

Figure 203: Styling Download App Section IV

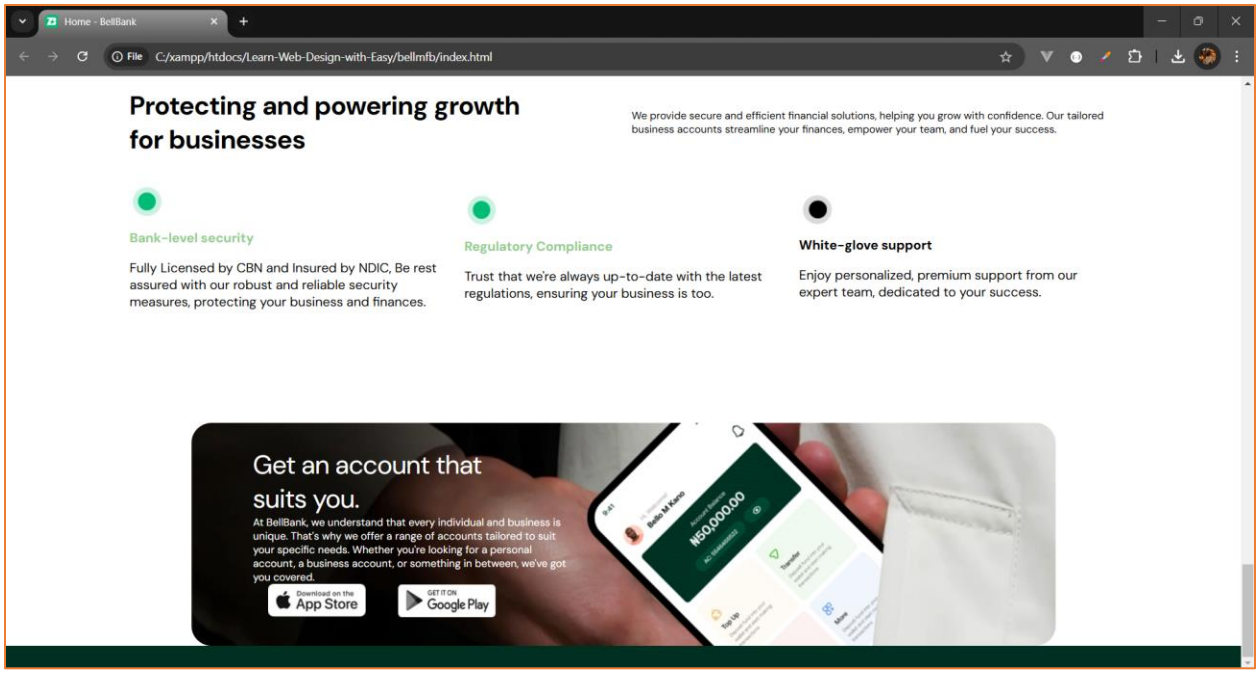


Figure 204: Styling Download App Section V

4. Let's improve the design by adding some padding within the texts. Below is the code snippet;

```
<!-- DownloadAppSection  -->
<div id="downloadApp">
    <div>
        <div class="text-heading-2 padding-y-2">
            Get an account that <br> suits you.
        </div>
        <div class="-text-heading-1 padding-y-2">
            At BellBank, we understand that every individual and business is <br>
            unique. That's why we offer a range of accounts tailored to suit <br>
            your specific needs. Whether you're looking for a personal <br>
            account, a business account, or something in between, we've got <br>
            you covered.
        </div>
        <!-- Download Icon  -->
        <div class="padding-y-2">
            <span class="downloadIcon"><img src="./images/apple_white_icon.png" alt="Apple Icon"></span>
            <span class="downloadIcon"><img src="./images/playstore_white_icon.png" alt="Android Icon"></span>
        </div>
    </div>
</div>
```

Figure 205: Styling Download App Section VI

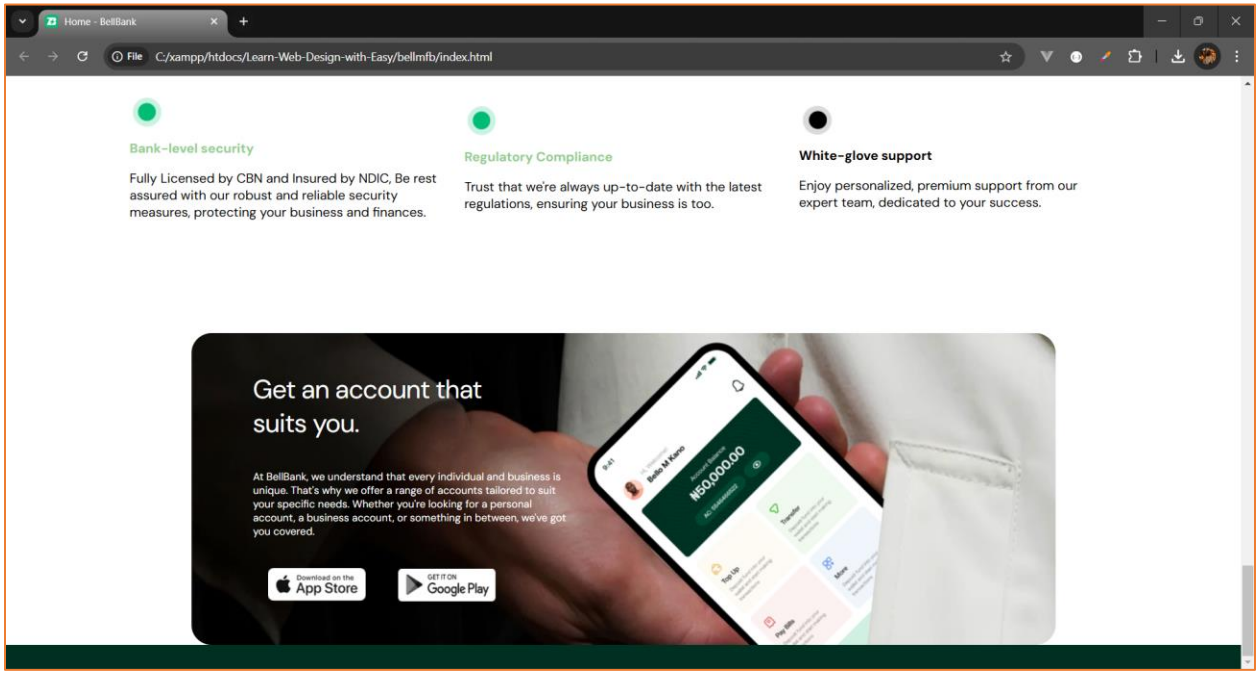


Figure 206: Styling Download App Section VII

Exercise 4 (Practical)

1. Make the text “Get an account that suits you” bold using HTML and not CSS.
2. Create some spacing around the downloadApp div container.

Answer

1. Solution

```
<div class="text-heading-2 padding-y-2">
    <b>Get an account that <br> suits you.</b>
</div>
```

Figure 207: Exercise 1 Solution

2. Solution

```
#pageSection{
    background-color: white;
    padding-bottom: 5%;
}
```

Figure 208: Exercise 1 Solution II

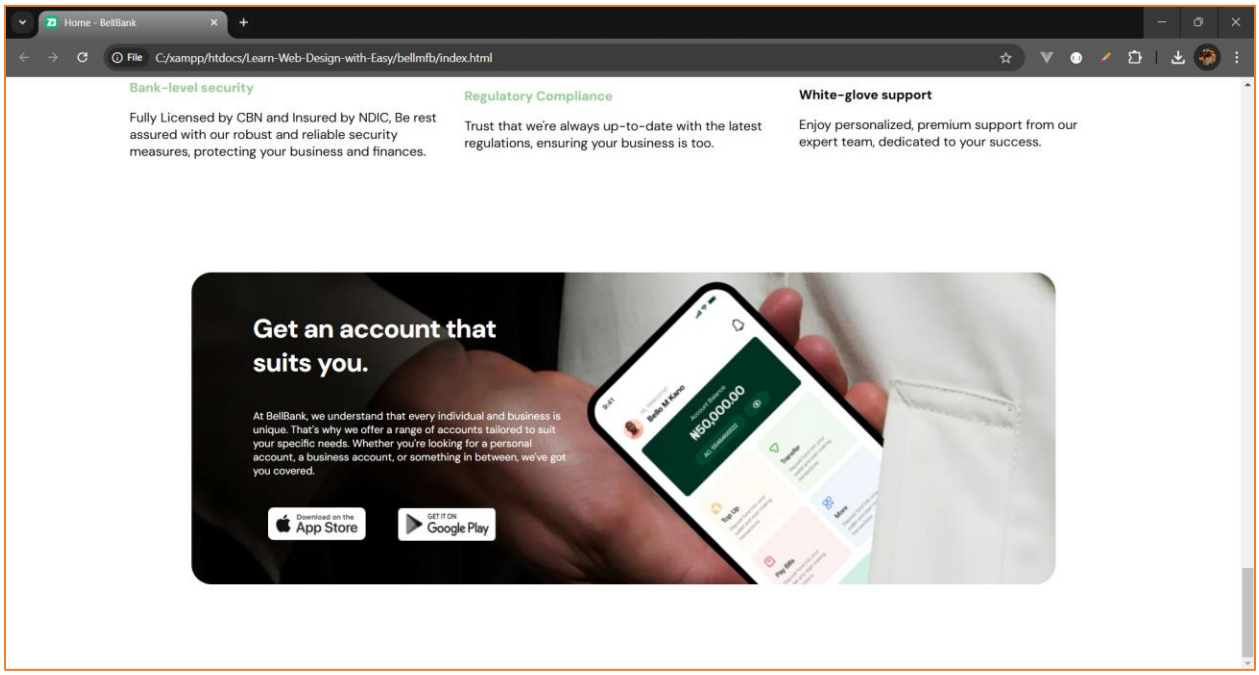


Figure 209: Styling Download App Section VIII

Congratulations on making it this far! We've covered and practiced a lot. Remember, the more you learn and practice, the closer you get to becoming world-class. Next, we'll start working on the footer section of the website. Congratulations again, and don't forget to smile! 😊

References

1. https://developer.mozilla.org/enUS/docs/Learn/Common_questions/Web_mechanics/How_does_the_Internet_work
2. https://developer.mozilla.org/enUS/docs/Learn/Common_questions/Web_mechanics/Pages_sites_servers_and_search_engines
3. https://developer.mozilla.org/enUS/docs/Learn/Common_questions/Web_mechanics/What_is_a_domain_name
4. https://developer.mozilla.org/enUS/docs/Learn/Getting_started_with_the_web/HTML_basics
5. https://developer.mozilla.org/enUS/docs/Learn/Getting_started_with_the_web/CSS_basics
6. https://www.w3schools.com/html/html_elements.asp
7. https://www.w3schools.com/html/html_head.asp
8. https://www.w3schools.com/html/html_attributes.asp
9. https://www.w3schools.com/html/html_favicon.asp
10. https://www.w3schools.com/tags/tag_body.asp
11. https://developer.mozilla.org/enUS/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure
12. https://www.w3schools.com/css/css_background.asp
13. https://www.w3schools.com/css/css_dimension.asp
14. https://www.w3schools.com/css/css_display_visibility.asp
15. https://www.w3schools.com/css/css3_flexbox_container.asp
16. https://www.w3schools.com/css/css_link.asp
17. https://www.w3schools.com/css/css_list.asp
18. https://www.w3schools.com/css/css_selectors.asp
19. https://www.w3schools.com/css/css_colors.asp
20. https://www.w3schools.com/css/css_margin.asp
21. https://www.w3schools.com/css/css_padding.asp
22. https://www.w3schools.com/css/css_border.asp
23. https://www.w3schools.com/css/css_font.asp

24. https://www.w3schools.com/css/css_font_google.asp
25. https://www.w3schools.com/css/css_text.asp
26. https://www.w3schools.com/css/css_grid.asp