

Bachelor's thesis

Bachelor of Engineering, Information and Communications Technology

2020

LAN NGUYEN

WEBSITE DESIGN AND DEVELOPMENT



Lan Nguyen

WEBSITE DESIGN AND DEVELOPMENT

The aim of this thesis was to present the process of building a website for a coffee shop named "Bokacha" as a front-end developer. The process includes: designing the website and developing the website using HTML5, CSS3, and Javascript.

The project was started at the beginning of January and was expected to finish at the end of February. Bokacha is a startup business and needs a website to achieve particular goals of business branding. The website allows the coffee shop to advertise their business online; provide information such as the address, menu, opening hours; update latest offers, products to customers; and receive feedback from customers.

This thesis consists of two main sections: the first part is defining the concepts of website designing and development, as well as introducing technologies and tools are used; the second section is implementing the website with designed layouts.

The outcome of this thesis was a one-page website that fulfills all the expectations of the Bokacha coffee shop. All the parties were satisfied with the result of the outcome which made this project successful.

KEYWORDS:

Website development, HTML5, CSS3, JavaScript, web design, UX design, UI design.

CONTENT

LIST OF ABBREVIATIONS	5
1 INTRODUCTION	6
2 THEORETICAL BACKGROUND	7
2.1 Website Design	7
2.1.1 User Interface	7
2.1.2 User Experience	8
2.2 HTML5	9
2.3 CSS3	10
2.4 JavaScript	12
3 WEBSITE DEVELOPMENT TECHNOLOGIES AND TOOLS	14
3.1 Web design tool	14
3.2 Text Editor	16
3.3 Chrome DevTools	18
4 WEBSITE PROJECT	23
4.1 Design and layout	23
4.1.1 Website wireframe	23
4.1.2 Color scheme	24
4.1.3 Website design	25
4.2 Development	25
4.2.1 Set up project folder	26
4.2.2 Build the website with HTML, CSS, and JavaScript.	26
5 CONCLUSION	34
REFERENCES	35

FIGURES

Figure 1. Example of Markup Language.	10
Figure 2. Inline method.	11
Figure 3. Internal method.	11
Figure 4. External method.	11
Figure 5. UI design tools.	14
Figure 6. Figma user interface.	15
Figure 7. Visual Studio Code Interface.	17
Figure 8. Chrome DevTools opened.	18
Figure 9. Computed tab opened.	19
Figure 10. Option to force element state.	20
Figure 11. Example of force state styles.	20
Figure 12. Example of error messages in Console tab.	21
Figure 13. Example of files fetched in Network tab.	21
Figure 14. Example of Application tab opened.	22
Figure 15. Bokacha website wireframe.	24
Figure 16. Website color palette.	25
Figure 17. The designs of the website in Figma.	25
Figure 18. Set up project folder.	26
Figure 19. HTML boilerplate.	26
Figure 20. HTML file.	28
Figure 21. HTML without CSS.	28
Figure 22. CSS file.	29
Figure 23. Appearance of the page when CSS file is added.	30
Figure 24. JavaScript file.	31
Figure 25. Adding "id" for each area	31
Figure 26. Scroll function.	32
Figure 27. Create a variable for scrolling distance.	32
Figure 28. Bokacha Website.	33

LIST OF ABBREVIATIONS

CSS	Cascading Style Sheets
HTML	Hyper Text Markup Language
UI	User Interface
UX	User Experience
W3C	World Wide Web Consortium
WWW	World Wide Web

1 INTRODUCTION

Undoubtedly, website design and development has been evolved over the past 20 years. In the early 1990s, websites were very simple. They were basically used for displaying plain text. It becomes clear that today's websites have changed remarkably. Websites in the late 2010s are not only good-looking but also dynamic, functional, and interactive. Also, people nowadays use various devices to browse the Internet such as mobile phones, tablets, laptops, therefore, websites have to be responsive. To achieve that, web technologies and tools have been constantly evolving.

Web development is the process of building and maintaining websites. In other words, web development concerns activities related to website functionalities development. There are two fundamental components in web development: front-end, back-end. In the scope of this thesis, front-end development will be discussed. The goal of the thesis is to give an overall view of website design and development through the process of designing and building a website for a start-up business. Throughout the five chapters of the thesis, the author will discuss the web development and design, introduce website tools and technologies, and implement steps building the website.

- Chapter one is briefly introduces the topic of the thesis.
- Chapter two focuses on the basis of web development which includes HTML, CSS, Javascript. The fundamentals of web design which are UX, UI design also are mentioned in this chapter.
- Chapter three introduces tools that are used to design and build the website. For designing, Figma is considered. Visual Studio Code and Chrome DevTools are also mentioned in this chapter.
- Chapter four explains the process of creating the website from the design phase to the final outcome which is a website.
- Chapter five gives an overview of what the author has learned through the project.

2 THEORETICAL BACKGROUND

As the Internet is commonly used, a website is essential for every business that wants to communicate with its clients. By putting a business on the world wide web, the business owner makes the business available around the world. Therefore, the website development industry has evolved rapidly in the last two decades. There are three specializations in terms of website development: front-end development, back-end development, and full-stack development. This thesis focuses on front-end development which is the practice of using coding languages, which are HTML, CSS, JavaScript to implement a website interface that web users see and interact with. In this chapter, basic concepts relating to web design and front-end web development will be discussed.

2.1 Website Design

In the digital business world, a good website plays a fundamental role in formulating a successful business. A well-designed website helps businesses deliver not only the needed information but also a satisfying experience. In order to form a good web design, two aspects are in consideration: User Interface (UI) and User Experience (UX). While User Interface deals with the aesthetics of the website, User Experience designs a pleasing experience when using the website. User Interface and User Experience are two vital elements in web design; they are two disparate definitions but go hand-in-hand. A good User Interface will draw the users' attentions, then a smooth User Experience will make them stay. It is the job of web designers to make a website catch the users' attention as well as satisfy the users' need at first glance. The two factors will be analyzed in more detail in the following sections.

2.1.1 User Interface

To begin with, the user interface of a website is what visitors use to interact with the website. There are three components UI design covers: visual design, interaction design, and information architecture.

Visual design is concerned with the aesthetics of a website. It includes parts such as layout, color, typography, spacing and other aspects of the look of a website. Visual design is all about choosing the right color palette, font style, the shape of a button, the

spacing between every element, etc. that make the website aesthetically pleasing. Interact design, on the other hand, focuses on how the website functions when interacting with users such as animation, loading indicators, and page transition. According to Zheng (2018), users tend to leave a website within 15 seconds if they do not feel interested in the website. Overall, both of the mentioned points need to be taken into account carefully to adapt to the “15-second rule”.

Another determinant of UI design is information architecture. Information architecture is the way the content of the website is rendered in a logical way that will not force users to spend much time or effort to find what they are looking for. The frame of the layout needs to be planned carefully to be able to achieve that. Besides, designers need to understand their product and know what the customers need. Information architecture is a crucial factor for a website to be considered as a high-quality product because it makes navigation simpler and user-friendly. Moreover, information architecture has a huge impact on user experience as the way the designers built the skeleton of the layout will affect how users utilize the website.

Obviously, how the content is presented visually affects how people perceive it. For example, if a website selling products such as Adidas puts poor quality images of shoes and a lengthy piece of text to promote their new products, the brand will undoubtedly fail in drawing the customers' attention in terms of the online market. Famous product pages usually have a simple but elegant layout in addition to various high-quality pictures.

In most cases, a user interface is the first impression web users have of the website, therefore, bad user interface design drives people away from the website immediately. Users do not spend a long time on a website that makes their eyes hurt because of bad font style, inferior color scheme, or overflowing content. After all, user interface design is all about the way the website communicates with end-users visually.

2.1.2 User Experience

User experience is beyond the functional or aesthetic aspect. It focuses on improving the experience the website creates for users, how users feel, how easy it is for users to achieve an intended purpose when using the website.

Firstly, user experience design is the process of strengthening the usability of a website, improving customer satisfaction when using the website. To be more specific, the

usability of a website is the ease of use of that website, which means a person of average technical skill or experience can use the website seamlessly. It cannot be emphasized enough that designers should have customers' perspectives and try out different combinations to find what is the most suitable.

Secondly, a website should not only be visually appealing but also deliver a pleasurable experience. For instance, if an e-commerce website has a perfectly attractive appearance, but the check-out process is over complicated or buggy, customers will consider they had a bad user experience and hesitate to visit the website next time. In contrast, if the website has an obvious, tempting "Buy Now" button and the check-out is smooth, straightforward and instantaneous; the business will be more successful in turning a visitor into a customer.

Another point to consider is that user experience has a great effect on visitor loyalty. It appears that websites which assure that visitors can find what they need without getting hopelessly frustrated, have a tendency to gaining visitors' loyalty. Indeed, if the user interface is what makes visitors stay, user experience decides whether they come back to the website again. Admittedly, there is a benefit in generating good experiences for visitors. In this technological era, people use web browsers daily to communicate, to connect and to interact with the world, so websites that save web user time and effort are the winners.

To develop a good user experience, UX designers use a method called user-centered design. In simple words, user-centered means putting the end-users' need at the center every step of the process of design (Garrett, 2011). When developing a website, the ultimate goal is to meet users' need, for this reason, users should be always taken into account, e.g., what they need, what they value, and their ability. As a result, UX designers are required a deep understanding of their end-users utilizing research, prototype, testing, and users' feedback.

2.2 HTML5

In general, HTML stands for Hyper Text Markup Language. Hypertext is a computerized text retrieval system that allows users to access cross-reference information via links. In other words, Hypertext is the text displayed on electric devices, e.g. computer, tablet, smartphone, which contains a link that enables a reader to navigate to related

information at another location on the same website or from any documents on the Internet. The second part of the HTML abbreviation, Markup Language, is a computer language used to annotate text in an electronic document so that the computer can structure the format of that document. Most Markup Languages are human-readable rather than programming syntax. For instance, the line of code in Figure 1 is using an example of Markup Language used to create a title on a webpage:

```
<title>This is an example</title>
```

Figure 1. Example of Markup Language.

In the example, the tag <title> is interpreted by the browser and then the browser renders the text as a title on the screen. Markup Language thus consists of two components: the actual text to be displayed and markup elements on how to display that text.

There have been five versions of HTML since the first time this markup language was introduced. At the time of writing the thesis, HTML5 is the latest version of the language, with new elements and JavaScript APIs to improve storage and hardware access. According to Robin Nixon (2009), HTML5 introduced over twenty new elements, some of them just provide machine-readable information to specialist browsers and search engines; still, they perform the same in the browser. To be specific, HTML5 came out with more definite semantic tags in comparison to the earlier versions such as nav, section, article, aside. These new semantic elements help to form a more well-structured page content as they describe what kind of content is in a specific tag.

2.3 CSS3

CSS (Cascading Style Sheets) is a style sheet language, which helps separate the visual appearance of a document from its content. CSS was introduced in 1996 and it has become the foundation of today's web design and development. There are three methods to incorporate a CSS file into HTML: external, internal and inline. Each method has its own advantage. For example, inline style is usually used to quickly change a style of an HTML element and this style rarely appears anywhere else in the HTML file. Another usage of inline method is to override an external or internal style. An inline style is written as Figure 2:

```
<h1 style="color:green;">Hello World!</h1>
```

Figure 2. Inline method.

By using the inline method, developers have to style each element separately without reusing code. Therefore, the inline method is not an efficient and scalable method.

As for internal style, it is usually used in small projects where developers have very few HTML files or even only one file. The styles will be listed right in the <head> tag of HTML file as Figure 3:

```
<head>
  <style>
    Body {
      background-color:#fff;
    }
  </style>
</head>
```

Figure 3. Internal method.

Then, the external style is used when developers have larger projects with complicated project structure and several HTML files. CSS styles will be stored in a separate CSS file and that file is linked to the HTML file as Figure 4:

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

Figure 4. External method.

The external method improves readability and reusability. One stylesheet can be applied to all of the web pages.

As mentioned above, an inline style can override an internal or external style. Sometimes, developers will have many styles that applied to the same element and they might conflict with each other. The style is defined later will override the previous one. One exception is that inline can override all other styles.

There are also numerous CSS libraries and frameworks that are built to save time and efforts for developers. They consist of ready-made components that can be copied or used directly in a project to speed up the process. If developers do not want to use ready-

made components, there are also ready-made styles that they can choose and decorate their own components the way they want. On the other hand, different projects will have different designs. Ready-made components and ready-made styles can not always satisfy the developers' demand. A typical solution for that is to override certain styles of the CSS libraries or frameworks with a separate CSS file. Developers can also modify the CSS libraries or frameworks file to customize them, but that will take lots of time.

Besides CSS libraries and frameworks, CSS pre-processors like SASS (Syntactically Awesome Style Sheets) are also developed. These pre-processors offer syntax advancements and additional features for CSS. For example, with the help of SASS, developers can use conditional statements inside a CSS file. Variables and functions are also levitated in SASS.

2.4 JavaScript

JavaScript is known as a lightweight, high-level compiled programming language. It was created by Brendan Eich of Netscape Communications in 1995. Back then, Netscape Communications wanted to create a scripting language that could not only complement Java but also has similar syntax, excluding existed languages at the time such as Perl, Python, TCL. Eich took only ten days to write a prototype to compete against other proposals. JavaScript's standard is ECMAScript (ECMA stands for European Computer Manufacturer's Association).

JavaScript is used to create more complex features for web pages. What web pages do with HTML and CSS is basically displaying information. With the help of JavaScript, web pages can perform more dynamic actions such as showing timely updated contents, animations, interactive maps, etc. Some basic programming features that JavaScript offers are

- Storing values inside variables.
- Manipulating created variables to perform operations such as joining strings together or calculating a number based on multiple numbers.
- Creating functions that are ready to be called/triggered.
- Reacting to events' triggers such as clicking, hovering, etc.
- Selecting an element and alter the values of the fields.

Another JavaScript's important feature is to send requests and receive data from the server through Application Programming Interfaces (APIs). API is like a communication protocol between devices (browsers, computers, servers, etc.). This feature is the main

difference between a static web page and a dynamic web page, as contents of a static web page are hard-coded while dynamic web page always receives updated contents from servers.

A crucial note when using JavaScript is that not every browser supports the latest syntax of it. An old or outdated browser might fail to respond correctly when handling a JavaScript function, while other new and updated browsers can understand and react accordingly. To solve this problem, various JavaScript libraries and frameworks have been built and developed.

Developers can build a website or application from scratch, but sometimes this is not necessary. There are common features that are used very frequently. In that case, JavaScript libraries offer ready-made solutions that can be used in the process of building a website or application to save time. One of the first libraries is jQuery, which was made in 2006. A JavaScript is not required all the time as sometimes it is heavy to load. For instance, jQuery is famous for selecting elements and their fields. Nevertheless, recently, developers can easily do the same thing with just pure, newly updated JavaScript. So, jQuery is not needed in the mentioned case, except if developers want to use other features.

With the help of JavaScript libraries, developers still had a hard time with handling shared data in projects. That was when JavaScript frameworks stepped in. A framework not only consists of solutions for common problems, but it also defines how developers call and use the code in certain ways to help with organizing the code base. Backbone, Knockout, Ember is a few of the first JavaScript framework. Then, AngularJS was released in 2010, followed by ReactJS and VueJS a few years later. Angular, React and Vue remain the most famous JavaScript frameworks in 2020.

3 WEBSITE DEVELOPMENT TECHNOLOGIES AND TOOLS

3.1 Web design tool

In terms of UI design software, there are various options on the market. They have similarities, still, each of them has unique features. Several main products that designers use are Sketch, Adobe Experience Design (Adobe XD), Figma, InVision Studio, Framer X, etc (Figure 5).

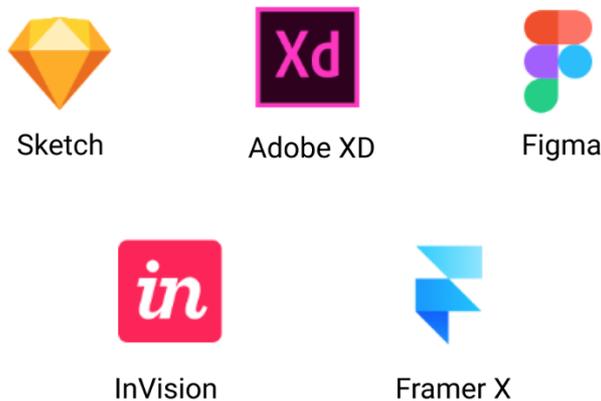


Figure 5. UI design tools.

In this project, the author uses Figma as a design tool to create the website design. This section covers the basics of Figma and its key features.

Figma is a collaborative, cloud-based design tool, which was developed by Dylan Field and Evan Wallace and launched in 2016. The core of Figma is entirely browser-based, therefore it is available on any platform that runs a web browser such as Mac, Windows, Linux, and Chrome OS. Undoubtedly, this feature benefits teams that designers and developers working on different operating systems. Also, Figma is the first real-time collaborative UI design tool, people who work in the same project can view, edit the design at the same time. Hence, a lot of unnecessary steps in the process are eliminated. Additionally, being able to share the most up-to-date version of the project with the sharing feature makes Figma a powerful collaboration tool. Another big advantage of

Figma is in-app commenting. Everyone in a team can easily communicate, give feedback while working on the design. Moreover, Figma concentrates solely on interaction design, its prototype feature is straightforward and intuitive. Unlike the traditional slideshow style prototyping, in order to demonstrate how the design will behave, Figma uses transitions between frame prototyping. This helps users visualize how the design looks and feels once the final product is built. Last but not least, the advantage of using Figma is cloud storage. Every change to a design is saved automatically. But this also brings a drawback. Because Figma is a web application, users cannot access projects offline. In this case, the project can be edited only if the user opened the application before going offline.

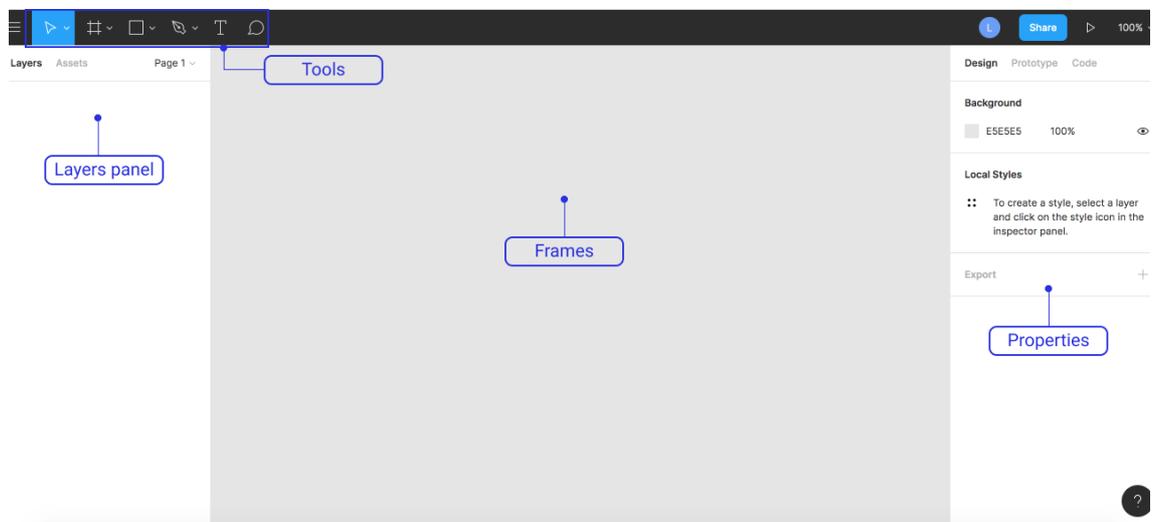


Figure 6. Figma user interface.

Figure 6 shows an overview of the application when opening up a new project. There are four main sections: toolbar, layers panel, canvas, and properties panel. The toolbar locates on the top left of the screen, including the menu and primary design tools. The menu can be expanded to show various functions and preferences. Below menu and toolbar is the page section (layers panel), which contains all the layers and elements of the design. In this section, designers can create, duplicate, delete, or rename pages by right click. The large area in the center of the screen is the artboard - the main working area, where the designs are placed and customized. In Figma, all the work happens and can be viewed inside this artboard. On the top right, the circular avatar(s) indicates people who are working on the project. The next button is the share option. The owner of the project can share a link or send an invitation via email to anybody who wants to view the design. Also, the owner has the right to allow other people to edit the design.

The next symbol is the presentation feature, where the project can be viewed as an interactive prototype. The last one on the top right corner is the view setting menu, which supports tools such as ruler, outline, multiplayer cursor, etc. The panel on the right-hand side is the properties (inspector) panel, which consists of design tab, prototype tab, and code tab. In the design tab, the properties of objects in the design can be adjusted. Prototype tab allows designers to construct and edit of the prototype of a product. The code tab shows how the elements of the design are expressed in code.

All in all, the best design tool is the one that a user finds fast and easy to use, as well as fell most confident with. When it comes to design digital products, Figma is an efficient, simple-to-use tool for designers. Figma is a platform where teams can work together, simplifies the design process.

3.2 Text Editor

The first step to writing code is picking a code writing application. Undoubtedly, code editors help developers speed up the process of writing code since they feature IntelliSense and autocomplete. There are various code editors with different factors such as built-in features, plugins, look and style, workflow, shortcuts, etc. Some of the most popular code editors are Visual Studio Code, Atom, Sublime Text, Brackets, etc. In this project, Visual Studio Code is used to build the website.

Visual Studio Code was created by Microsoft and introduced on April 29, 2015. Visual Studio Code is a free, cross-platform application with open source via MIT License. It is a lightweight but powerful source code editor, which is compatible with a variety of languages and has large libraries of extensions and plugins. Additionally, the application is highly customizable, theme and keyboard shortcuts can be changed to users' preferences. Also, the user interface of Visual Studio Code is simple and intuitive. Figure 7 is a welcome screen when opening Visual Studio Code application.

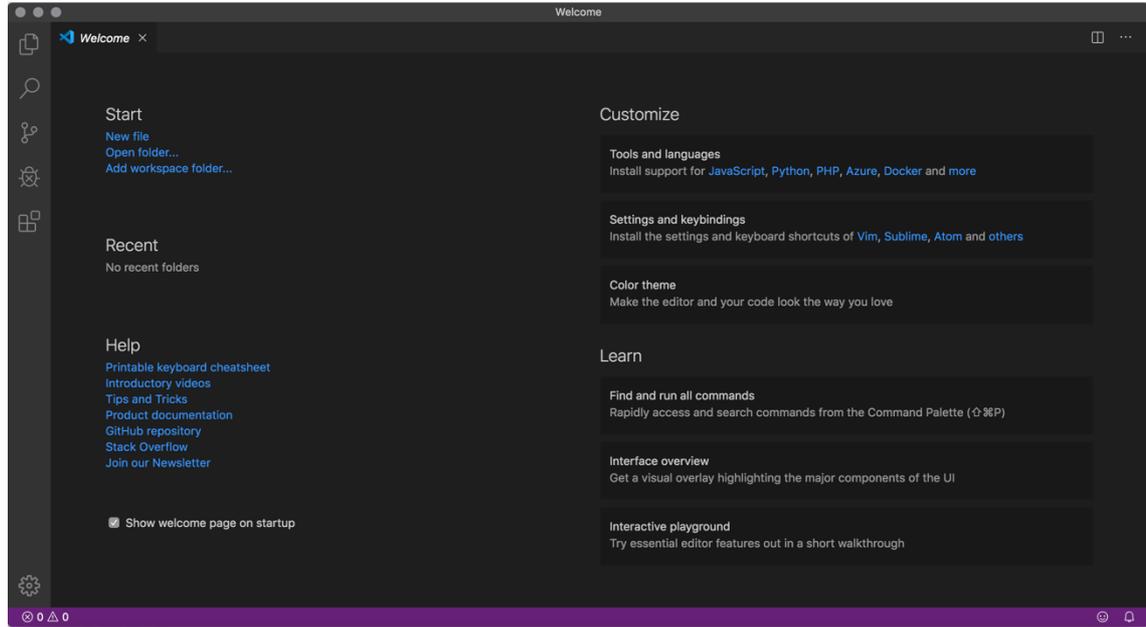


Figure 7. Visual Studio Code Interface.

The sidebar on the left-hand side consists of

- 'File explorer' allows users to find all the project files and directories.
- 'Search' helps with search and replace functionality
- 'Source code management' includes Git integration
- 'Launch and debug' supports debugging function
- 'Manage extensions' lets users add languages, debuggers, and tools.

On the bottom left corner is the menu for settings, keyboard shortcuts, extensions, command palette, etc. Command palette lets users access all of the functionalities of Visual Studio Code, some of the features that are not displayed through the menu or the interface. Status bar on the bottom of the window gives information about the files such as errors and warnings. The main space for editor is maximized and allows users to open more than one editor with the side-by-side feature.

The writing code process is more efficient and quicker when developers use the help of extensions. Picking the right extensions enhances the productivity and workflow of users. In this project, the following extensions are installed:

- Live sever
This extension sets up a local web server so changes can be seen in real-time without having to reload the page.
- Prettier

This is a code formatter, which works for HTML, CSS, JavaScript. It formats code in a consistent style throughout all the files.

- Auto rename tag
When a user wants to change a tag, this extension helps change the ending tag automatically.
- JavaScript (ES6)
This helps speed up JavaScript coding since it contains code snippets for JavaScript in ES6 syntax.
- HTML CSS support
This extension helps with class and id attribute completion.
- Bracket pair colorizer
Blocks of code can be visually organized by identifying matching brackets with the same color.

3.3 Chrome DevTools

Google Chrome DevTools is a very crucial part of the front-end developers' toolkit. It consists of various tools that help developers to debug and build websites faster.

There are different ways to open Chrome DevTools. Developers can press Command+Option+C (Mac) or F12/Ctrl+Shift+J (Window, Linux). Another way is to inspect any element directly by right-clicking on it and choose "inspect" from pop-up options. A window with tools should be opened:

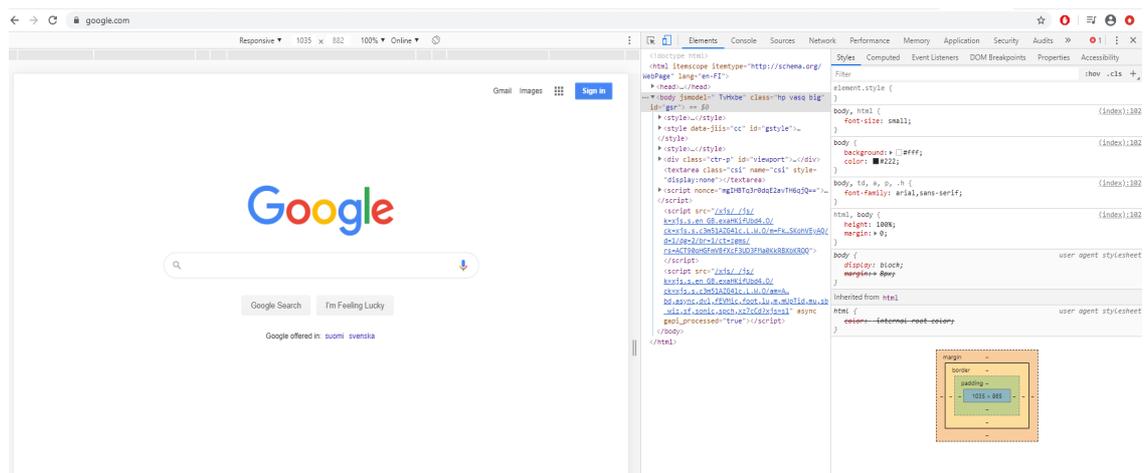


Figure 8. Chrome DevTools opened.

Some important features of DevTool will be mentioned in this section.

As can be seen from Figure 8, an extra window has been opened on the right of the website with various options to choose from. The first tab, which is called “Elements”, lists all the elements and tags that are used. If developers open DevTools by inspecting the element, it will be automatically selected in the list. Any styles that are applied to that element are shown on the right. Developers can easily change those styles like adding new styles or remove existing ones. The styles will take effect instantly and display the changes in the website window. This is a fast way to test if certain styles are working as they should.

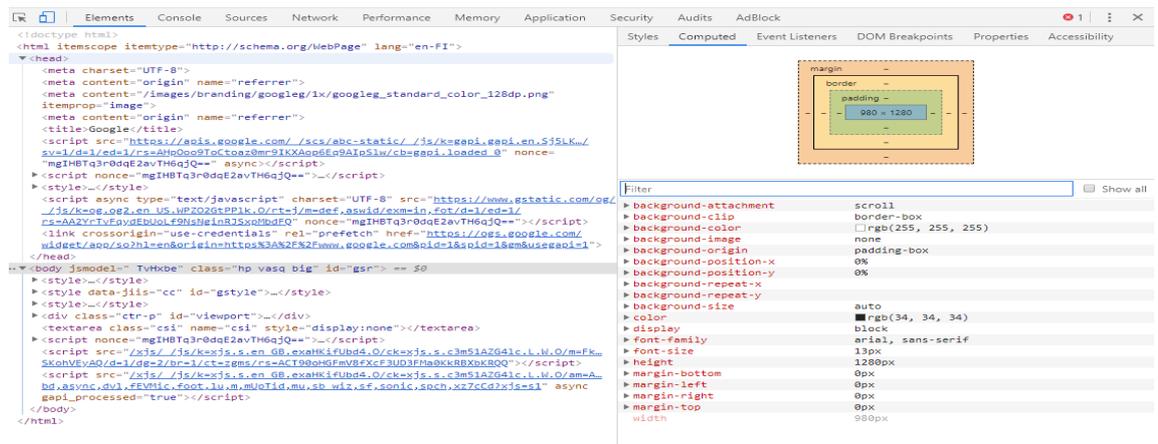


Figure 9. Computed tab opened.

Sometimes an element can have lots of classes that conflict and override each other, making the styles tab very crowded and hard to follow. In that case, developers can levitate the use of the Computed tab. In the Computed tab, there will be a list of final styles that are applied to the selected element. Moreover, developers can always click any style to see which class is storing it. (Figure 9)

Another useful feature is forcing a state of an element. Developers can right-click on the element and choose the option to “force state” (Figure 10)

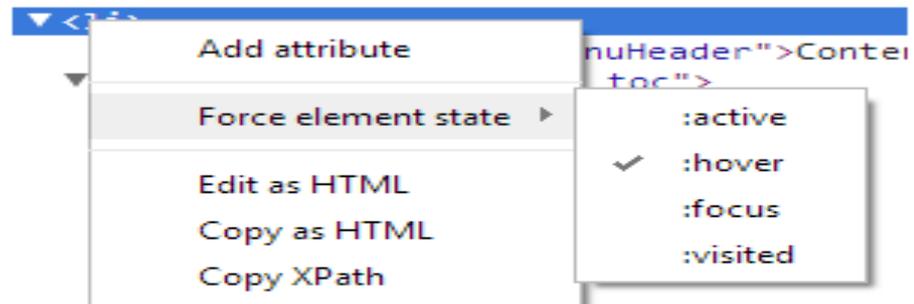


Figure 10. Option to force element state.

There are several states to choose from. If any state is chosen, the styles of that state will take effect instantly and can be seen from the website window. In the DevTools window, styles for that state are listed as well. Usually, state's styles will not be listed when an element is inspected unless it is forced to show that state. (Figure 11)



Figure 11. Example of force state styles.

DevTools are not only used for inspecting elements but also for debugging. Any error that occurs while the website is rendered will be shown in the Console tab (Figure 12). If there is something wrong with the website, most likely the reason will be mentioned in that tab. Then, from that error message, developers can investigate more and trace to the root of the problem.

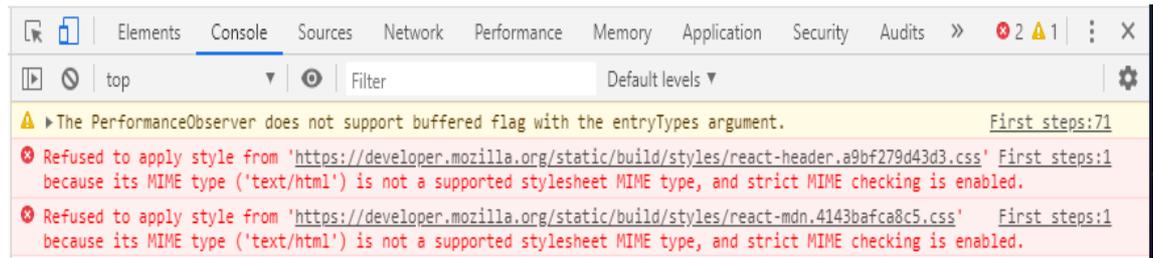


Figure 12. Example of error messages in Console tab.

Another very essential tab is Network (Figure 13). In that tab, there are details of files and data that are used. It could be anything: an image, an extra CSS stylesheet, some data that are fetched from APIs, manifest file, some documents, etc. It is usually used to check if the data is fetched successfully or not.

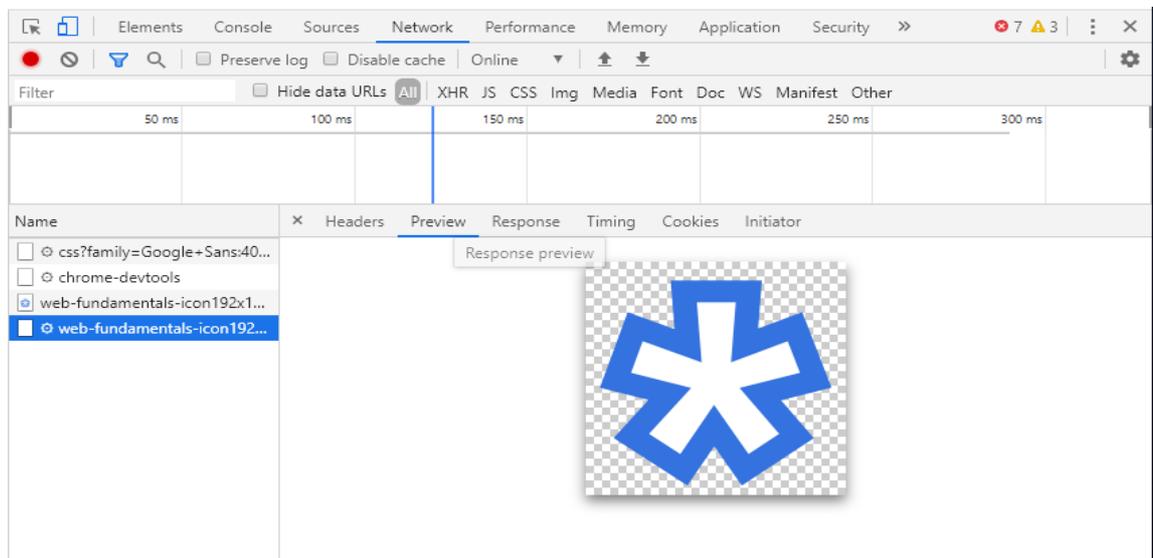


Figure 13. Example of files fetched in Network tab.

The last tab that is mentioned in this section is the Application tab (Figure 14). In this tab, the main features that should be noticed are the storage of cached data, manifest file information and service worker file. Cached data is data that will be stored on the devices when users visit the website for the first time. It will be quickly displayed next time the site is visited. The manifest file offers general information about the website/application. The service worker file is basically a JavaScript file that enhances the possibility to work offline on the website.

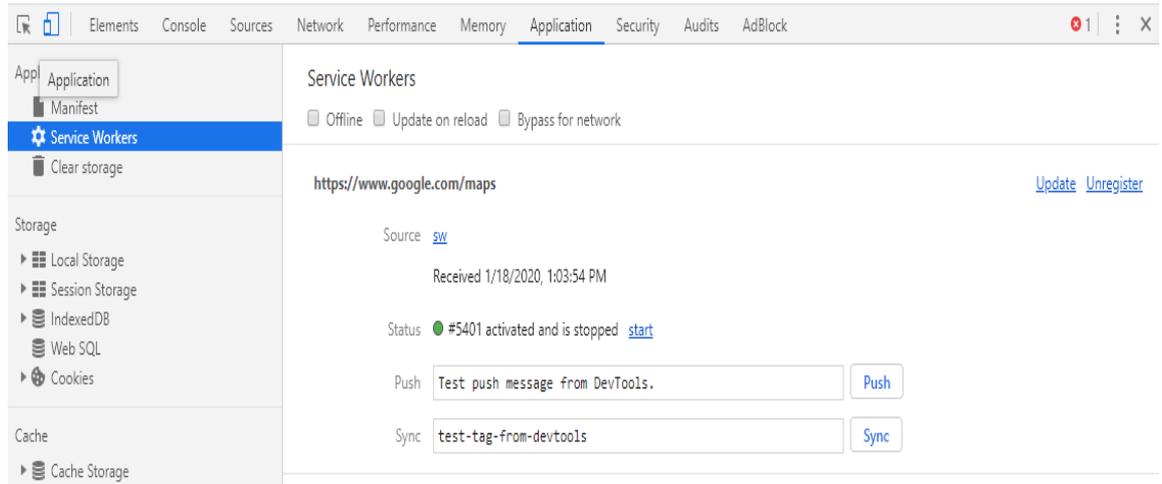


Figure 14. Example of Application tab opened.

4 WEBSITE PROJECT

This chapter describes the main stages of implementation and the outcome of the website.

4.1 Design and layout

4.1.1 Website wireframe

The first stage of the web design project is wireframing (Figure 15). The purpose of wireframing is to organise the content structure and the flow of the web page before worrying about the fonts and colors. This step should be quick and minimal. After gathering information, the wireframe of the website is sketched. The wireframe focuses on

- Arrangement of the website's content
- Functionalities of the website
- Information hierarchy

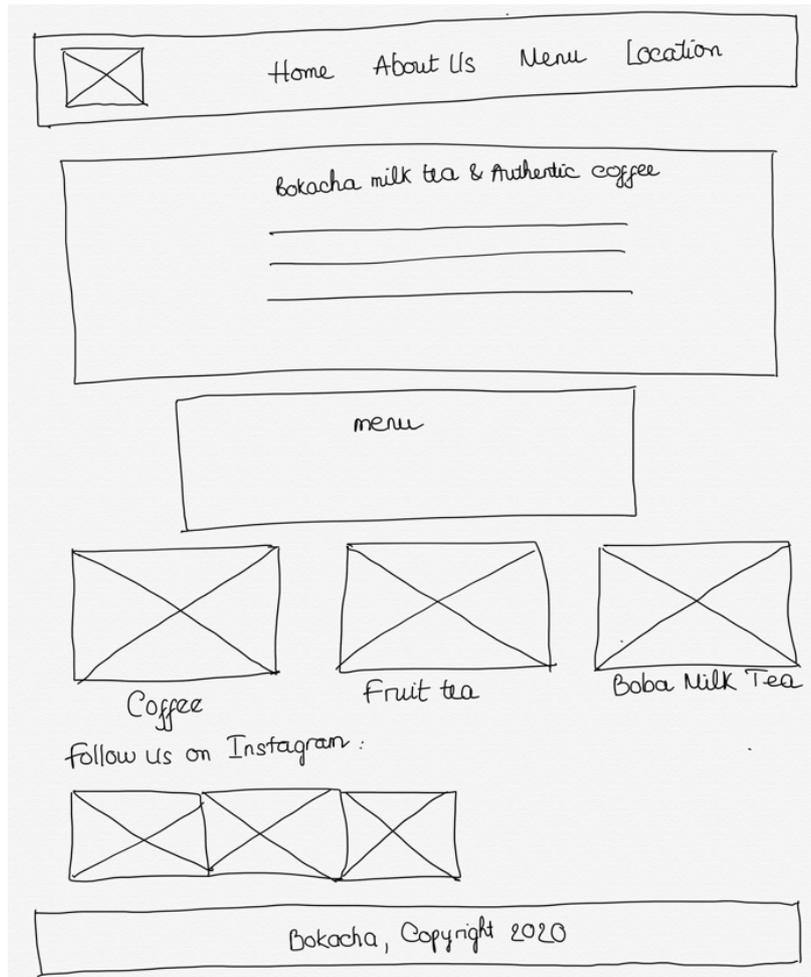


Figure 15. Bokacha website wireframe.

4.1.2 Color scheme

It is clear that color is one of the most important parts of a design. Having a good color palette helps strengthen the design of the website. The color of a website should not be too bright, too dark or too contrasty. The website color scheme is chosen based on the color theme of Bokacha coffee shop. There are four colors as shown in Figure 16. They are analogous colors on the color wheel. This combination makes a monochromatic look as well as a harmonious feel for the website.



Figure 16. Website color palette.

4.1.3 Website design

The visual design is created according to the wireframe and following the color scheme. In this step, every element such as logo, navigation item, header, button is illustrated in detail. Since the content structure was arranged in the wireframing step, the task basically is putting text and placing images into the design. Figma is used to outline the design. (Figure 17)

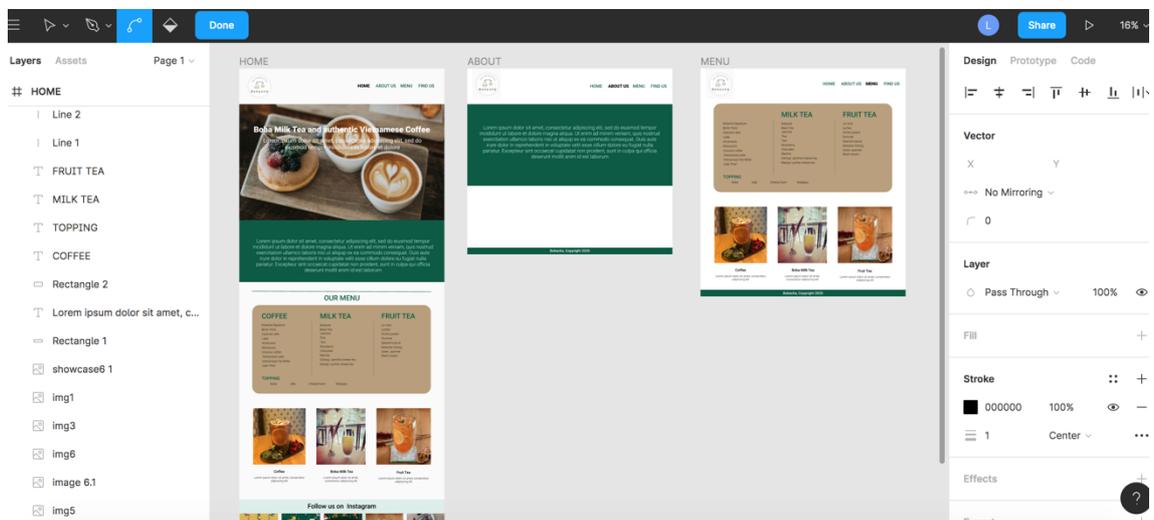


Figure 17. The designs of the website in Figma.

4.2 Development

Once the finished design is ready, the next stage is the practice of using HTML, CSS, JavaScript.

4.2.1 Set up project folder

The first step of building a website is to set up the project folder. The website consist of there main files (Figure 18)

- Index.html is the core structure of the website. It includes every element and links to other files.
- Style.css stores external styles for the elements in the index.html file.
- Main.js includes all the functions that are called or triggered by elements in the index.html file.

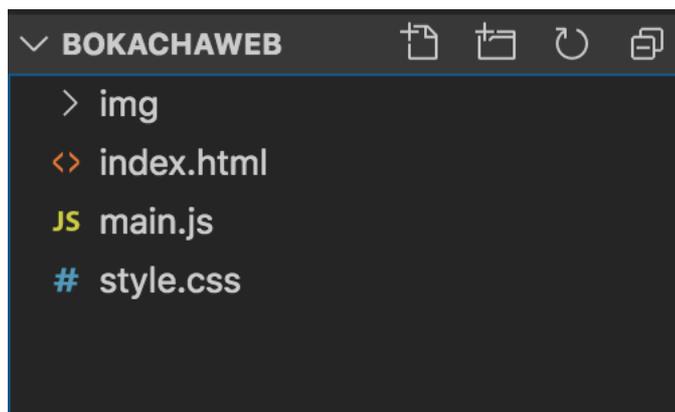


Figure 18. Set up project folder.

4.2.2 Build the website with HTML, CSS, and JavaScript.

Creating HTML file:

To start with, an HTML file is built with Visual Studio Code editor. An HTML boilerplate is created by exclamation [!] (Figure 19)



Figure 19. HTML boilerplate.

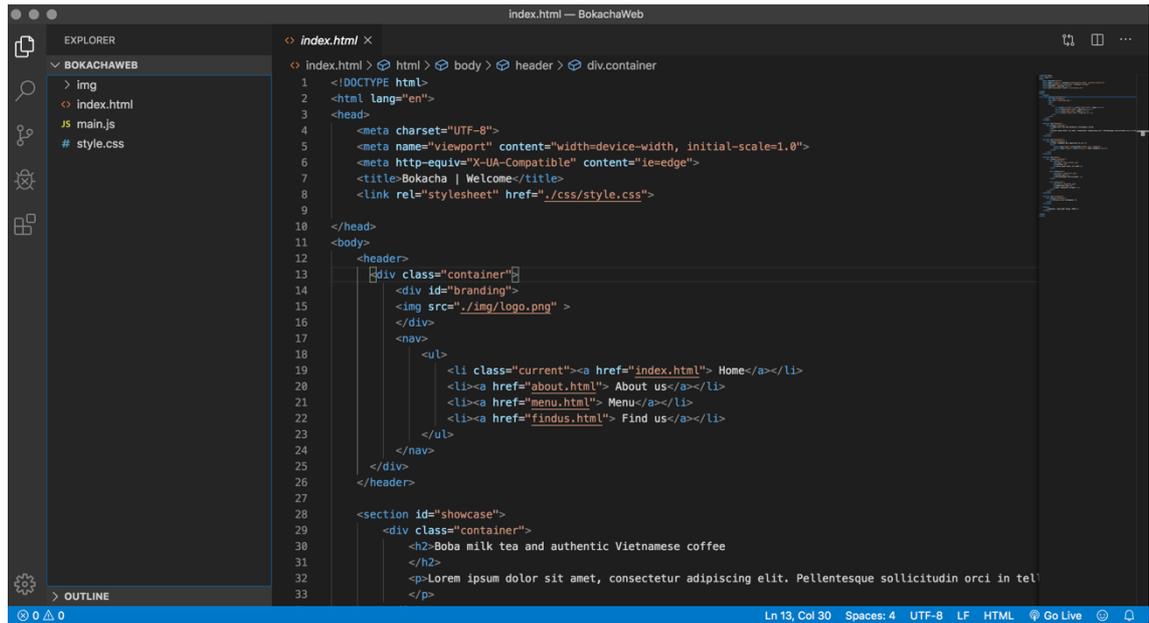
A basic HTML website requires the following elements

- `<!DOCTYPE html>` this is a declaration that defines the language markup of the document to be HTML.
- `<html>` is the core element of a web page. Everything is contained within this element.
- `<head>` this is the first element within the `<html>` tag. The head element contains metadata which is information about the document. It also contains links to style sheets and javascript file.
- `<title>` this element specifies a title for the document.
- `<body>` contains the page content which is visible to the web users.

The next step is adding semantic elements to create the layout as sketched in the wireframing step, which are:

- `<header>`: represents the introductory content. This is a container for heading components which are logo and navigation bar.
- `<div>`: defines a division in HTML document. This tag is used for the purpose of wrapping or grouping some elements together, then style those elements with CSS. It acts as a common container with no specific meaning, not like `<nav>` or `<footer>`.
- `<nav>`: defines a set of navigation links.
- ``: defines an unordered list.
- `<section>`: describes a section of the document.
- `<h1>`, `<h2>`, `<h3>`: define HTML headings.
- `<p>`: defines a paragraph.
- `<footer>`: represents the footer of the page. A footer contains information about copyright and author information.

After having the frame of the website, the content such as text and images are filled in this HTML file (Figure 20). Technically, images are not stored in the HTML file, the `` tag is used to link the images from "img" folder to HTML page.



```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Bokacha | Welcome</title>
8   <link rel="stylesheet" href="./css/style.css">
9
10 </head>
11 <body>
12   <header>
13     <div class="container">
14       <div id="branding">
15         
16       </div>
17       <nav>
18         <ul>
19           <li class="current"><a href="index.html"> Home</a></li>
20           <li><a href="about.html"> About us</a></li>
21           <li><a href="menu.html"> Menu</a></li>
22           <li><a href="findus.html"> Find us</a></li>
23         </ul>
24       </nav>
25     </div>
26   </header>
27
28   <section id="showcase">
29     <div class="container">
30       <h2>Boba milk tea and authentic Vietnamese coffee
31     </h2>
32     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque sollicitudin orci in tel
33   </p>

```

Figure 20. HTML file.

The code results in the picture shown in Figure 21.

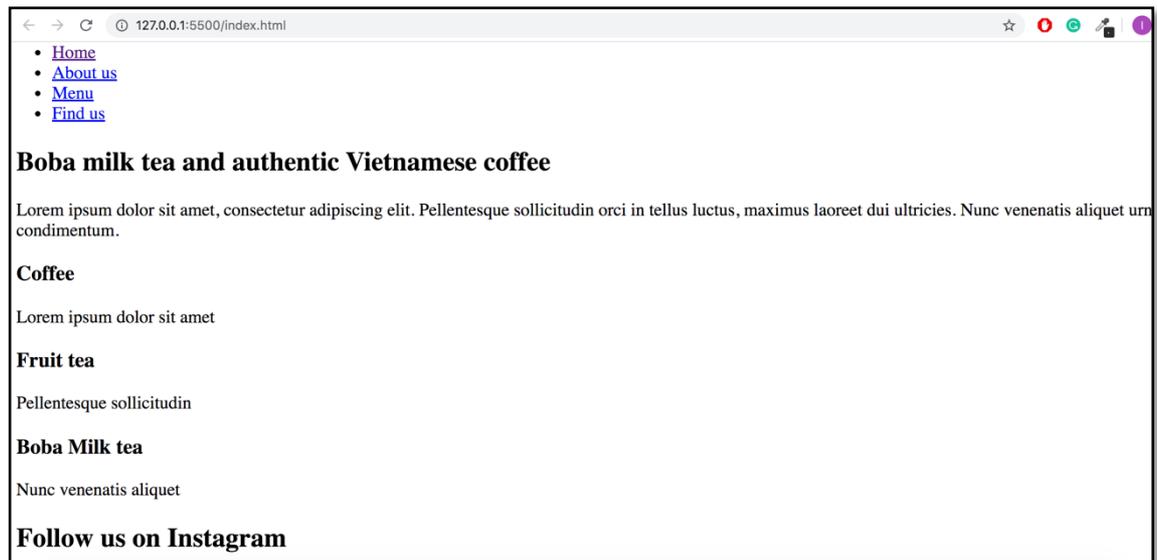
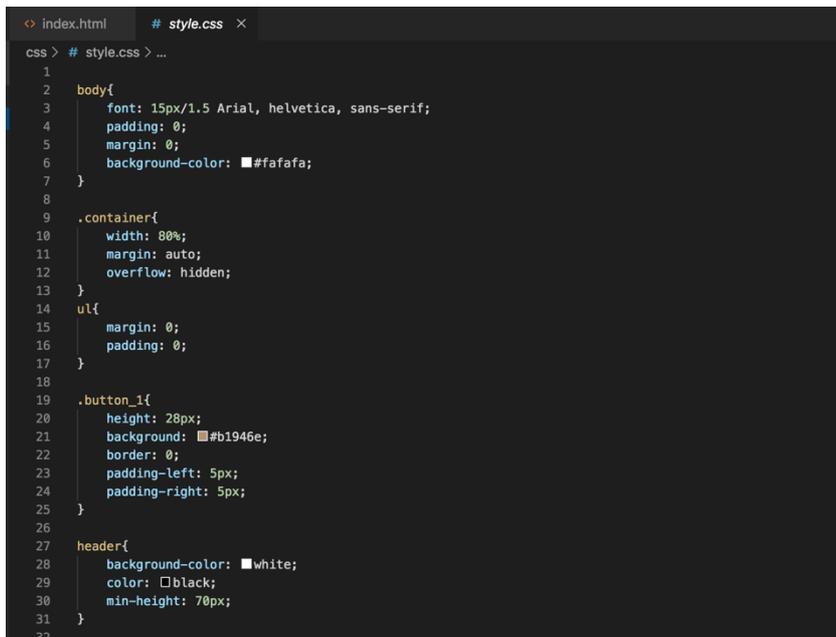


Figure 21. HTML without CSS.

Adding CSS file:

The next step of the process is writing CSS code. As shown in Figure 21, the result of HTML file is a plain page with only text, in order to decorate the web, CSS file is added. CSS is responsible for the colors, fonts, and positioning of the content.

A screenshot of a code editor window showing the content of a file named 'style.css'. The editor has a dark background and a light-colored text. The code is written in a standard CSS syntax, defining styles for the body, a container, a list, a button, and a header. The code is as follows:

```
css > # style.css > ...
1
2 body{
3   font: 15px/1.5 Arial, helvetica, sans-serif;
4   padding: 0;
5   margin: 0;
6   background-color: #fafafa;
7 }
8
9 .container{
10  width: 80%;
11  margin: auto;
12  overflow: hidden;
13 }
14 ul{
15  margin: 0;
16  padding: 0;
17 }
18
19 .button_1{
20  height: 28px;
21  background: #b1946e;
22  border: 0;
23  padding-left: 5px;
24  padding-right: 5px;
25 }
26
27 header{
28  background-color: white;
29  color: black;
30  min-height: 70px;
31 }
32
```

Figure 22. CSS file.

CSS file (Figure 22) controls the appearance of the web page using CSS properties:

- Color of the text and background: `color`, `background-color`.
- Text manipulation: `text-align`, `text-decoration`, `text-transform`.
- Size and style of the font: `font-weight`, `font-family` and `font-size`.
- Box model (margin, border, padding, width, height): `margin`, `border`, `padding`, `height`, `width`.
- Alignment of container element: `float`.

As a result, Figure 23 is how the page is rendered after applying CSS code.

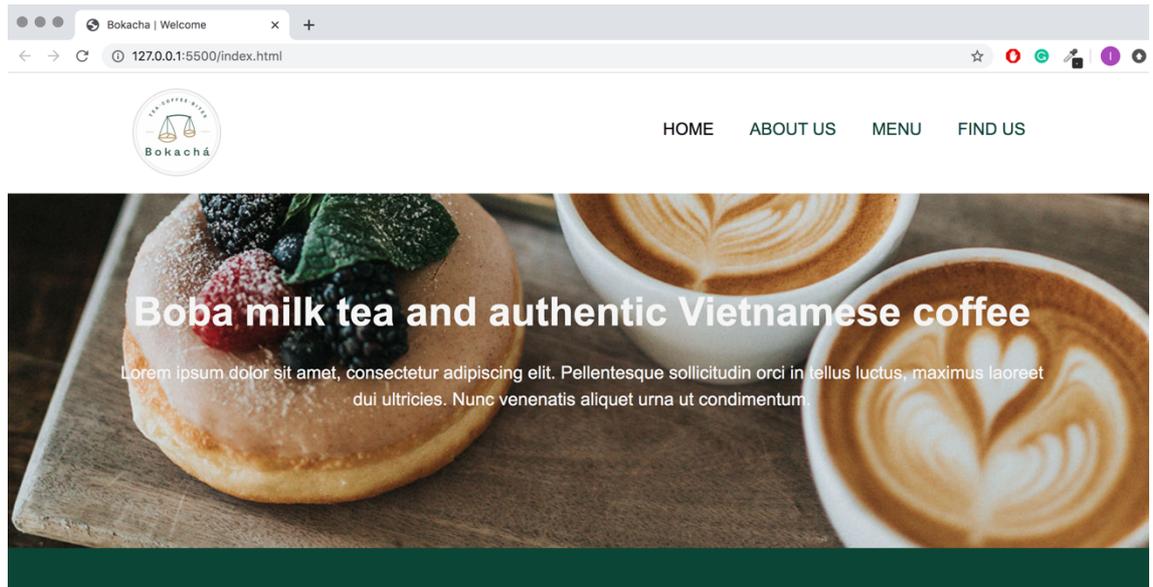


Figure 23. Appearance of the page when CSS file is added.

JavaScript file:

Finally, the JavaScript file is added to make the website functional and interactive.
(Figure 24)

```

main.js x
1  $(window)
2  .scroll(function() {
3      var scrollDistance = document.documentElement.scrollTop;
4
5      // Assign active class to nav links while scrolling
6      if (scrollDistance >= 0) {
7          $("#nav-about-us").removeClass("active");
8          $("#nav-menu").removeClass("active");
9          $("#nav-find-us").removeClass("active");
10         $("#nav-home").addClass("active");
11     }
12     if (scrollDistance >= $("#about-us").position().top - 100) {
13         $("#nav-home").removeClass("active");
14         $("#nav-about-us").addClass("active");
15     }
16     if (scrollDistance >= $("#menu").position().top - 100) {
17         $("#nav-about-us").removeClass("active");
18         $("#nav-menu").addClass("active");
19     }
20     if ($(window).scrollTop() + $(window).height() == $(document).height()) {
21         $("#nav-menu").removeClass("active");
22         $("#nav-find-us").addClass("active");
23     }
24 }
25 }
26 .scroll();

```

Figure 24. JavaScript file.

The navigation items suppose to take users to various areas of the website based on which item that the users chose. That could be done by adding an id to each area (Figure 25). And then in each navigation item, a link to that id will be stated through the href attribute

```

<a class="active" href="#home">Home</a>
<a href="#about-us">About us</a>
<a href="#menu">About us</a>
<a href="#find-us">About us</a>

```

Figure 25. Adding "id" for each area

Next, a function needs to be created to highlight the chosen navigation item. If the users do not click any navigation item but scroll down to the area of any navigation item, that item will be highlighted as well. The function will be triggered when the website is scrolled (Figure 26)

```
$(window).scroll(function() {})
```

Figure 26. Scroll function.

A variable is created to calculate the space that has been scrolled (Figure 27)

```
var scrollDistance = document.documentElement.scrollTop;
```

Figure 27. Create a variable for scrolling distance.

The next step is to compare that variable to the actual position of each area. As can be seen from the above figure, there are a couple of condition statements that are created to work on that. The first condition is to compare the position of the navigation bar, which is 0px from the top. If the condition is met, the “active” class is added to the element (the navigation item). On the other hand, that class will be removed from all other navigation items. The second one is the comparison between the variable and the position of “about-us” area. 100px is reduced from the position since the navigation bar is sticky. It needs to be calculated to show the beginning of the chosen area. The way of rest statements works similarly.

The final outcome of the project is the Bokacha website as shown in Figure 28.

[HOME](#) [ABOUT US](#) [MENU](#) [FIND US](#)

Boba Milk Tea and authentic Vietnamese Coffee

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

OUR MENU

COFFEE	MILK TEA	FRUIT TEA
<ul style="list-style-type: none"> Bokacha Signature Berlin Tonic Caramel Latte Latte Americano Mocha Coconut coffee Vietnamese Latte Vietnamese Flat White Cafe "Phin" 	<ul style="list-style-type: none"> Bokacha Black Tea Jasmine Thai Taro Strawberry Chocolate Moksha Oolong/ Jasmine cheese tea Mango/ Lychee cheese tea 	<ul style="list-style-type: none"> Li' mint Lychee Perfect peach Summer Special tropical Bokacha Oolong Green Jasmine Black Assam
TOPPING Boba Jelly Cheese foam Bobabop		

Coffee
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Boba Milk Tea
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Fruit Tea
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Follow us on Instagram

Bokacha, Copyright 2020

Figure 28. Bokacha Website.

5 CONCLUSION

Having a website, a business can represent itself in the World Wide Web. Indeed, a website is an essential source of information about a business. Therefore, web development is continually evolving to adapt to the importance of websites. The basic building blocks of web development are HTML, CSS, JavaScript. HTML organizes the structure of the website, CSS is responsible for the appearance, and JavaScript adds functionalities and interactivities for the website.

In summary, there were five main phases in this project:

- Analysis: this phase involved talking with the customer to gathering information. The author identified the goals and requirements of the website.
- Planning: the author determined the objectives of the project in terms of time, methods, technologies and tools used.
- Design: in this phase, the appearance of the website was created following design principles.
- Development: this is the main focus of the project, which is the practice of using coding languages (HTML, CSS, JavaScript) to implement the website interface.

As a result of this project, the author achieved a better understanding in terms of web design and development. Also, the author gained more experience by building a website from scratch.

REFERENCES

Figma help center, [online] Available at: <https://help.figma.com/hc/en-us/categories/360002042553-Using-Figma>. [Accessed January 2020]

Garrett, J. J. 2011. The elements of user experience: User-centered design for the Web and beyond. 2nd ed. New Riders: Berkeley, CA.

Google developers, [online] Available at: <https://developers.google.com/web/tools/chrome-devtools>. [Accessed February 2020]

Constine, J. 2015. 23-Year-Old's Design Collaboration Tool Figma Launches With \$14M To Fight Adobe. Tech Crunch. [online] Available at: <https://techcrunch.com/2015/12/03/figma-vs-goliath/>. [Accessed January 2020]

Nixon, R. 2009. Html5. BarCharts: Boca Raton.

Visual Studio Code docs, [online] Available at: <https://code.visualstudio.com/docs/getstarted/userinterface>. [Accessed January 2020]

Wikipedia. JavaScript. [online] Available at: <https://en.wikipedia.org/wiki/JavaScript>. [Accessed February 2020]

Wikipedia. Website. [online] Available at: <https://en.wikipedia.org/wiki/Website#History>[Accessed January 2020]

Wikipedia. Web development. [online] Available at: https://en.wikipedia.org/wiki/Web_development. [Accessed February 2020]

World Wide Web Consortium. A little history of the World Wide Web, [online] Available at: <https://www.w3.org/History.html>. [Accessed January 2020]

Zheng, D. 2018. The 15 Second Rule: 3 Reasons Why Users Leave a Website. [online] The daily egg. Available at: <https://www.crazyegg.com/blog/why-users-leave-a-website/>. [Accessed January 2020]