

**Sang Nguyen**

**BUILDING RESTAURANT WEBSITE FOR BUSINESS USE**

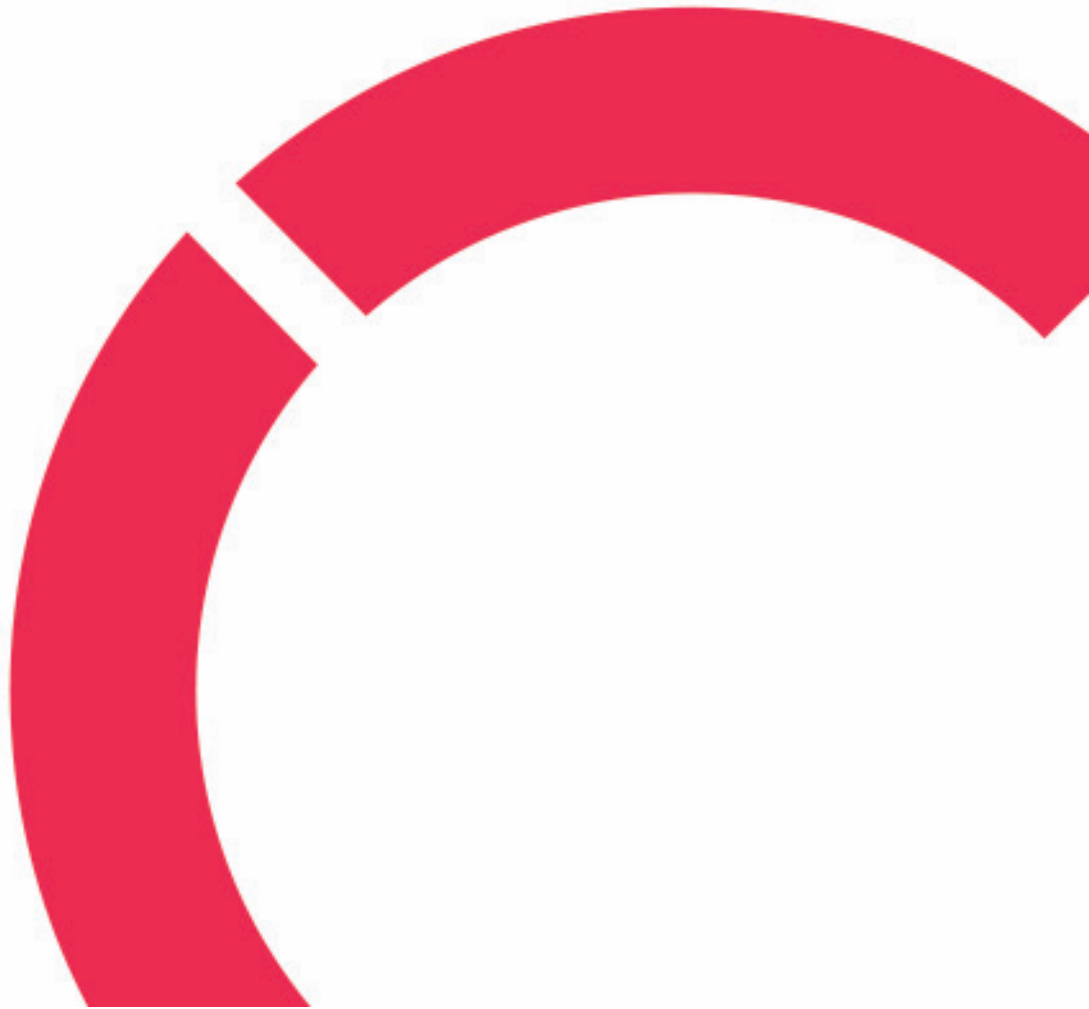
**Case Miyabi Sushi restaurant in Kokkola**

**Thesis**

**CENTRIA UNIVERSITY OF APPLIED SCIENCES**

**Information Technology**

**June 2023**



**ABSTRACT**

<b>Centria University of Applied Sciences</b>	<b>Date</b> June 2023	<b>Author</b> Sang Nguyen
<b>Degree programme</b> Information technology		
<b>Name of thesis</b> BUILDING RESTAURANT WEBSITE FOR BUSINESS USE Case Miyabi Sushi restaurant in Kokkola		
<b>Centria supervisor</b> Jari Isohanni		<b>Pages</b> 48
<p>The purpose of this thesis was to describe the process of developing a website to meet the needs of a business. Specifically, the project involved designing and creating the responsive web in front-end for the Miyabi Sushi restaurant located in Kokkola, Finland, allowing customers to view restaurant information and place online orders.</p> <p>The development of the website was divided into two main stages. The first stage included the researching and planning phases of the web design to align with the business requirements. The second stage concentrated on the actual development process and the web deployment. ReactJS, along with JavaScript, HTML, and CSS, were used to build the web application, supporting the data update dynamically. Subsequently, the local server, with Node.js runtime environment, was utilized to support the deployment and testing processes.</p> <p>As the result of the project, the website was delivered to the client with full functionality. The thesis project serves as a successful example of constructing a web application on client-side for small or medium-sized businesses.</p>		
<p><b>Key words</b> CSS, HTML, JavaScript, ReactJS, Responsive website, Restaurant website</p>		

## **CONCEPT DEFINITIONS**

### **UX**

(User Experience) is a term to depict the feeling of a person in interacting with a product or service.

### **UI**

(User Interface) is a term to depict the visual parts or elements of a product or service that user interacts with.

### **DOM**

(Document Object Mode) is a term to describe the structure of an HTML or XML document presented as node.

### **API**

(Application Programming Interface) is a term to describe a set of rules and protocols allowing communication between software applications.

### **IDE**

(Integrated Development Environment) is a term to indicate a software application providing comprehensive tools and features to support software development.

### **VS Code**

(Visual Studio Code) is a term to indicate source code editor developed by Microsoft.

### **CLI**

(Command Line Interface) is a term to demonstrate the typed commands and text-based interface.

### **GIF**

(Graphics Interchange Format) is a term to demonstrate a specific format used for animated images and short video clips.

### **Nav**

(Navigation) is a common abbreviation to refer the navigation menu or navigation bar on a website.

**ABSTRACT**  
**CONCEPT DEFINITIONS**  
**CONTENTS**

<b>1 INTRODUCTION.....</b>	<b>1</b>
<b>2 BUSINESS WEBSITE PLANNING .....</b>	<b>3</b>
2.1 Objectives.....	3
2.2 Market research .....	5
2.2.1 Target customer .....	6
2.2.2 Competitors research .....	7
2.3 Finalizing website requirements .....	9
<b>3 WEB DEVELOPMENT TOOLS AND TECHNOLOGIES.....</b>	<b>11</b>
3.1 ReactJS.....	11
3.2 JavaScript .....	12
3.3 Redux.....	12
3.4 HTML & CSS.....	13
<b>4 MIYABI SUSHI WEB DESIGN.....</b>	<b>14</b>
4.1 Web layout structure .....	14
4.2 Web content design .....	16
4.2.1 Hero section .....	16
4.2.2 Navigation menu .....	16
4.2.3 Typography .....	17
4.2.4 Colour .....	17
4.2.5 Images and Graphics .....	18
4.3 Mock-up design .....	19
<b>5 MIYABI SUSHI WEB DEVELOPMENT.....</b>	<b>20</b>
5.1 Website type selection .....	20
5.2 Web development environment .....	21
5.3 ReactJS environment setup.....	22
5.4 Project file structure .....	24
5.5 Web construction .....	26
5.5.1 Navigation bar and footer .....	26
5.5.2 Home page .....	29
5.5.3 About page.....	31
5.5.4 Menu page .....	32
5.5.5 Contact page!.....	33
5.5.6 Catering page .....	35
5.5.7 Online Order page .....	36
5.5.8 404-error page .....	41
5.5.9 Website rendering.....	42
<b>6 MIYABI SUSHI WEBSITE FUTURE STEPS AND UPDATE .....</b>	<b>44</b>
<b>7 CONCLUSION .....</b>	<b>46</b>

**REFERENCES.....48**

**FIGURES**

FIGURE 1. Navigation type usage .....5  
FIGURE 2. Miyabi Sushi customer’s review on Google and Foodora app.....6  
FIGURE 3. Layout of the website ..... 15  
FIGURE 4. Visual sitemap (flat structure) of Miyabi Sushi website ..... 15  
FIGURE 5. Primary and secondary colour palettes..... 18  
FIGURE 6. The landing page and menu page mock-up designs of Miyabi Sushi website ..... 19  
FIGURE 7. Recommended extensions to install .....22  
FIGURE 8. React app installation and starting development server .....23  
FIGURE 9. External packages installed .....24  
FIGURE 10. The Miyabi Sushi web project file structure .....25  
FIGURE 11. Miyabi Sushi website – navigation bar on desktop and mobile device views .....27  
FIGURE 12. The footer of Miyabi Sushi website .....29  
FIGURE 13. Miyabi Sushi – landing page design.....30  
FIGURE 14. Miyabi Sushi website – Gallery section .....31  
FIGURE 15. Miyabi Sushi website – About page .....32  
FIGURE 16. Miyabi Sushi website – Menu page.....32  
FIGURE 17. Miyabi Sushi website – Contact page.....33  
FIGURE 18. Email segment input in contact form using label and input elements .....34  
FIGURE 19. Miyabi Sushi website – Catering page .....35  
FIGURE 20. Miyabi Sushi website – Online Order page on desktop and mobile view.....36  
FIGURE 21. Online Order page – Online order item pop-up.....37  
FIGURE 22. Online Order page – Online order cart section.....38  
FIGURE 23. Online Order page – Online order pick-up option.....39  
FIGURE 24. Miyabi Sushi website – 404-error page.....42

**TABLES**

TABLE 1. Similarity of two restaurant websites .....8  
TABLE 2. Difference between two restaurants .....9

**CODES**

CODE 1. Managing open state and responsive screen resize with useState hook and media queries ...28  
CODE 2. Map section implementing Google API for map .....35  
CODE 3. Online Order page – code implementation for controlling section overlay and pop-up.....40  
CODE 4. Online Order page – code implementation for order data update .....40  
CODE 5. Online Order page – code implementation for updating item quantity .....41  
CODE 6. Root component – App.js and application render – index.js .....43

## 1 INTRODUCTION

Nowadays, the modern technologies offer excellent opportunities for start-up companies to easily expand their businesses. The advantages come from the use of automated machines, smart systems, and the Internet of things, which facilitate the expansion of marketing campaigns and sale management. Particularly, in terms of the company's online presence, it brings a significant effect on business growth derived from its online sale channel. Indeed, online marketing through social media platforms and websites plays an essential role in business development. The web platform provides companies with favourable environment to aggressively promote their products and services in the targeted markets. The COVID-19 was a tough challenge for the world economic growth, especially for enterprises offering food-related service. As a result, the demand for online service web applications has increased for businesses of all sizes, including small, medium, and large ones, during and after the COVID-19 pandemic.

The core objective of this thesis project was to create a responsive web application on the front-end side for "Miyabi Sushi", a start-up restaurant in the food industry that offers Asian traditional fusion in the city of Kokkola, located in western Finland. The restaurant size is average size, serving a wide range of sushi and Asian traditional food selections. Moreover, the lunchtime buffet is one of its stand-out features, offering high-quality food and friendly service. A restaurant website will have a significant impact on its visibility among potential customers searching for it online. Increased revenue is not only a result of increased visitors but also of boosting brand awareness. Additionally, the website features promotions and special events, which helps attract new customers and retain customer loyalty.

The first section of this thesis report is about the theoretical background, which includes the website's business requirements and an overview of the web development tools and technologies used. The second section focuses on the implementation processes of the website, with clear indications of the web design, including layout, content design, and mock-up designs. These steps serve as a solid foundation for the later development stage. The website project was developed using ReactJS, a widely used open-source JavaScript library for web development. Redux was utilized for the application's state management, working in conjunction with ReactJS. Meanwhile, JavaScript was manipulated to create various functions related to user interaction and application features. The final version of the website was deployed and tested without database and backend server development in the local development

environment. Finally, the further development and update are discussed after the web construction section.

## **2 BUSINESS WEBSITE PLANNING**

A comprehensive plan can ease the progression of the project by offering a precise framework for the website's design. It serves as the fundamental outline for the ultimate webpage. During this planning phase, the objectives of the business website are established, declaring the future vision for its usage. Also, the market research is conducted to collect the market statistics and improve the comprehensive analysis. This research is carefully invested through two main identifiers, as competitor research and target customer. These steps indicate the website's purpose and identify the business's target audience. The website developing requirements is finalized and served as a crucial precursor, which forms the last section of this phase.

### **2.1 Objectives**

It is believed that a high-quality website, which provides consumers with a valuable experience, can significantly enhance online shopping activities through its implemented functionalities. Additionally, a stable website service is essential in influencing customers' perceptions of service quality and trust. (Liu, Xiao, Lim & Tan 2017, 754-756.) Thus, the objective of building the Miyabi Sushi restaurant website is to create a user-friendly design styled with a navigation bar on top, footer at the bottom, and content displayed in the middle. The Miyabi Sushi Kokkola website will serve as the primary means to raise brand awareness. It will offer crucial information about the restaurant, enabling customers to gain a clear understanding of the available services and offers. Furthermore, the website will allow the owner to advertise new campaigns on the webpage as an effective online marketing tool to boost sales growth. This approach aims to establish a favourable environment that directly impacts customers' perception of the service from their initial access to the website.

Besides, the product distribution on an online store has considerable impact on the product appeal and the customers' suspicion level regarding the product quality. If the products exhibited on a website are favourably evaluated by consumers, this can generate positive emotion throughout the entire website. This leads to increasing the purchase intention of the customers without any concerns about the transactions. To widen the website appeal, there are various studies demonstrating that the increasing of diagnostic signal can enhance the customers' positive attitudes towards product evaluation. In addition,



adding justifiability can serve as an alternative solution for reinforcing consumers' confidence in purchase decisions. This approach effectively convinces customers to make the desired purchase. (Liu et al. 2017, 756-758.)

On the other hand, the website of Miyabi Sushi restaurant is mainly designed to promote the product sales. Therefore, its primary objective is set to apply functionalities that support product evaluation for the online customer. For instance, one such function enables users to manipulate product images by zooming in and out. The website should also offer detailed product descriptions to provide customers with a clear information that enhances their knowledge about the products. This again builds up the confidence in product selection. Furthermore, a review section can be added to display other customers' comments on the restaurant's products and services. This feature significantly influences the decision-making process by piquing customers' interest in the featured products and decreasing their negative perceptions of anticipated purchases.

Aside from product exhibition, website developers commonly prioritize the implementation of effective navigation and search systems to support online shopping activities. There are two main types of nav bar in navigation system, known as linear and hamburger. The research demonstrates that linear menu is the more commonly implemented option, accounting for 87.5% of the selections (FIGURE 1). (Satria 2018, 3-4.) Considering the Miyabi Sushi web development, the navigation system functions as a roadmap to simplify the website searching experience, making it easy for users to find all categories of a restaurant's products. Besides, the search system works as a convenient and efficient tool to help users quickly search out desired content online. A well-designed search engine will afford quick response, user-friendly and precise performance in its search results, which allows users to reach the finding content readily.

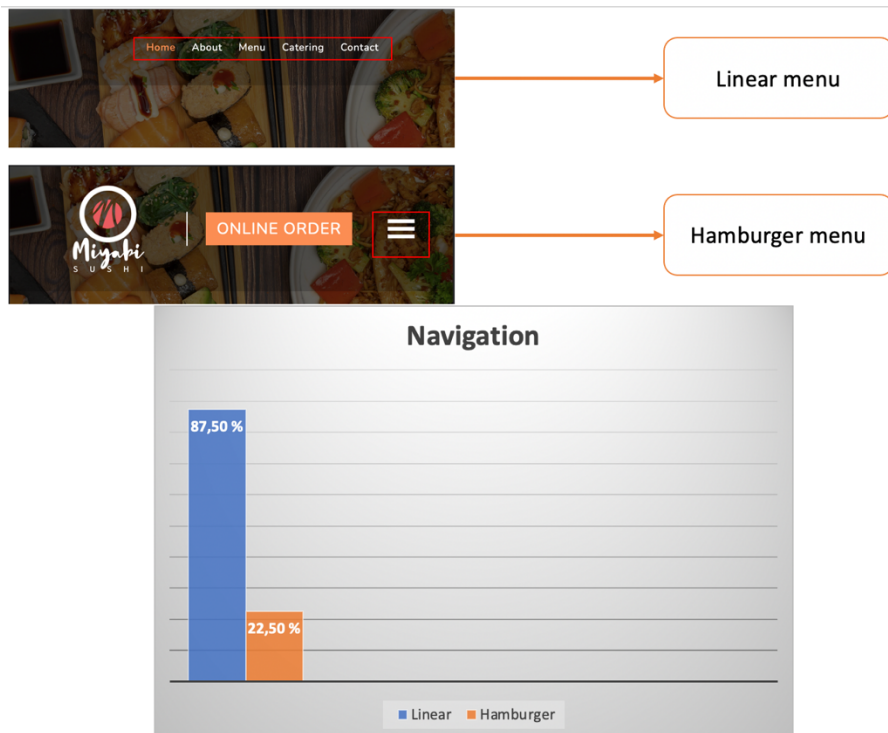


FIGURE 1. Navigation type usage (adapted from Satria 2018, 4)

According to Satria Eri's research (2018, 3) in the IOP Conference Series: Materials Science and Engineering, the white space between website elements plays a decisive role in improving consumers' concentration on important information. Therefore, another key consideration for website developers, in this project, is maintaining a consistent layout across pages with the same design style, from the landing page to other pages such as menu, contact, and about. Consistent formatting of web content, including text and images, throughout the website can gain the user experience. Furthermore, English is the primary language used for all content in the initial version of the website, supporting the engagement of both local and international customers.

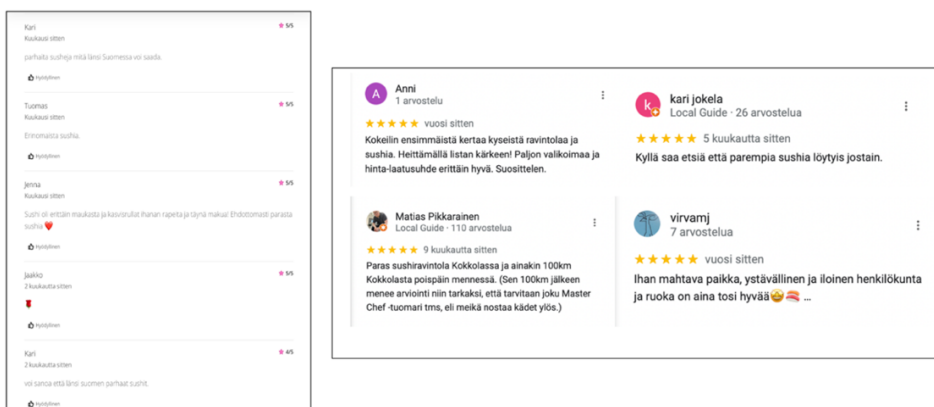
## 2.2 Market research

The process of market investigation is crucial during the planning phase of web development, which identifies the key factors to facilitate website design. Conducting comprehensive market research helps the business owners to verify the needs of their target customers, which directly influences on the position of companies in market competition. By gathering information related to the business field, such as business objectives, potential visitors, and competitor background, it enables recognition of market

needs and customer motivation in website engagement. This generates data insights of users allowing the company to apply effective web strategy with all the required assets, which significantly gains competitive advantages as well as brand exposure. (Dordevic 2023.) In this market research section, two main factors of market analysis are explored, including target customer research and competitor research. These factors support the finalization of specific requirements for Miyabi Sushi web development.

## 2.2.1 Target customer

Conducting market research is vital in identifying the key characteristic to focus on during website development. The website must be constructed in compliance with the needs of consumers and provide them with the information, products, and service they seek. (Campanelli 2019.) Specifically, in the case of Miyabi, the target audience consists of local citizens, food enthusiasts, and Asian fusion food aficionados. Thus, the restaurant's website should provide relevant and valuable content that caters to these potential customers. As a high-quality Asian fusion cuisine provider, Miyabi has received positive feedbacks from customers who express satisfaction with the traditional meals prepared by the restaurant, which is shown below in the Figure 2. According to the consumer preferences, the sushi items are given priority, followed by other types of cuisine and traditional meals. By understanding the market's demands, the Miyabi online service can quickly and efficiently fulfil consumer needs, resulting in easier decision-making and enhancing brand recognition among customers. This underscores the importance of market research in website creation.



Google and Foodora review feedback

FIGURE 2. Miyabi Sushi customer's review on Google and Foodora app (Google 2023; Foodora 2023)

### 2.2.2 Competitors research

As a part of market research, it can be advantageous to have knowledge of the competitors to evaluate the market objectively. This enables the identification of the competitors' strengths and weaknesses in relation to their services, which can serve as a benchmark for improving the overall value of the company's strategy. (Campanelli 2019.) This approach is particularly relevant to the website development, which helps in distinguishing the restaurant from its competitors. The restaurant industry in Kokkola is highly competitive among various types of restaurants, each offering a special range of products and services. Among these, Miyabi Sushi is an Asian restaurant that offers both buffet and à la carte options during the daytime. By examining the online services of other restaurants, key elements can be identified and incorporated into the development of the Miyabi Sushi restaurant's website, thereby enhancing its differentiation from other establishments.

In this study, two competitors are selected for the purpose of comparison. The first restaurant, Madez Lounge, specializes in Asian cuisine, particularly Indian cuisine. The second one is Luckie Fun's sushi buffet, which is known for offering a wide range of high-quality buffet services and is one of the best buffet brands in Finland. In comparison, Miyabi sushi is of similar business size to Madez Lounge, but smaller than Luckie Fun's. Their websites are the main source data for this analysis process, which can be accessed at <https://www.madez.fi> for the Madez website and <https://www.luckiefun.com> for the Luckie Fun website. By using the data collected, a comparative analysis is conducted to identify similarities and differences in web content. The results of this analysis are presented in two tables below, providing a detailed comparison of the selected restaurants for generating design requirements in the next part.

TABLE 1. Similarity of two restaurant websites

Similarity	Description
Web purpose	The main aim of both website is to provide the restaurant information, their sale products, and customer service
Web interface	The design of UI is user-friendly and attractive.
Web content	It is informative including business information and social media links
Web functionality	The web functions offer the simple and easy-to-use navigation.

The main target of Madez Lounge and Luckie Fun's websites is determined as showing their business information, products and offering services. Both webpages share similarities as user can find the high attractive visual on both webpage as well as easily explore the web content via its user-friendly built-in navigation. (TABLE 1.) However, there are some different factors making these sites different from each other. Firstly, the main colour set used on Madez Lounge is illustrated in black, yellow, and red, while the white and red are the primary colours on the website of Luckie Fun's. Each website offers Finnish language in the web content, but users can solely switch to English option on Luckie Fun's web service. Furthermore, the online order service of Madez restaurant relies on a third-party service called Oracle Gloria system, which is different with the built-in order system on Luckie Fun's site. Secondly, the visitors to Luckie Fun online service have access to a wider range of service choices and detail pages compared to the Madez Lounge. (TABLE 2.) The above analysis highlights the valuable elements that can be customized based on the service objectives and applied on the Miyabi Sushi site, allowing the construction of a high-quality website in the later development phase.

TABLE 2. Difference between two restaurants

Difference	Description	
	Madez Lounge	Luckie Fun's
Web theme colour	Black, yellow, and red	White and red
Web language	Only Finnish available	Bilingual languages: Finnish and English
Web online food order	Using Oracle Gloria food ordering system	Having their own order system
Web content	Basic information	Detail information and other service available such as purchasing gift card, feedback, and finding the nearest restaurant.
Web user interface	One page only for the whole content with basic design	Divided into small pages containing valuable information with attractive design.

### 2.3 Finalizing website requirements

Based on the results of a benchmarking analysis, the owner of the business has decided to proceed with the Miyabi web development project, related to client-side showing the official appearance of the website. The official website for Miyabi Sushi will draw inspiration from top-tier restaurant websites analysed above, such as Luckie Fun, Madez, and Nera Steakhouse. The Nera Steakhouse websites can be viewed at <https://steakhouse.neras.fi/homepage-en/>. The business team has specified several requirements related to the design of the navigation bar, social media, and service channels. Specifically, the navigation bar should be located at the top of the page with a standard horizontal layout, and the restaurant's online media presence on platforms such as Facebook and Instagram should be linked to the website. Additionally, the restaurant partners, such as Wolt and Foodora, may be added for providing alternative order methods to customers.

In terms of the development details, the web requirements are determined to focus on the features, functions, and content of the website. These aspects are important to achieve the success of the project. The Miyabi website will include well-designed functional components such as the navigation bar and shopping cart, making it a dynamic website. Moreover, a series of images showcasing restaurant's special dishes must be added, while the Hero section serves as an initial entry point of restaurant website. The website's functions should be optimized for high performance to provide quick responses to user interactions, supporting the smooth operation of the website's features. For the requirements of visual design, Miyabi Sushi must be a responsive website, allowing customer to conveniently access the webpage with different device screen size. Also, the restaurant data must be readily available and accessible, which contains all relevant information, a feedback form, location details, review link, and a list of menu items. Finally, the content will be arranged consistently to display information clearly, with an emphasis on appealing design to attract visitors and encourage engagement with the business.

### 3 WEB DEVELOPMENT TOOLS AND TECHNOLOGIES

Modern technologies provide various uncomplicated approaches for learning and constructing programs without constraints. Nowadays, web development has become accessible for everyone, with numerous tools and software available to ease the evolving process. (Dzhangarov, Pakhaev & Potapova 2021, 2-6.) The current thesis project concentrates on the front-end of website, also referred to as client-side development, to construct a React web application for the Miyabi Sushi restaurant. To achieve positive results, the development process involves utilizing ReactJS, JavaScript, Redux, HTML, and CSS. After completion, the application is deployed and tested through local hosting environment.

#### 3.1 ReactJS

ReactJS, also known as React, is a widely recognized JavaScript library that has gained popularity for building user interfaces in single page applications. This open-source library has valuable effect on the JavaScript community, particularly in the front-end web development. React offers developers and engineers a variety of tools, including composable components, virtual Document Object Model (DOM), and React API, which enables to build user interface applications readily. With the support of React, developers can improve the user experience and reduce the complexity of application operation. The library's asynchronous operations allow the webpage to handle a seamless re-rendering with new data, avoiding any interruptions or delays. Additionally, ReactJS's component architecture serves as the primary unit of an application to break down the app into small and manageable parts. Besides supporting the rendering process, this method also encourages the reuse of components to save time and effort. (Lim 2020, 7-10; Thomas 2018, 5-7.)

In a website, its structure is represented by DOM demonstrating the hierarchical composition of HTML elements. This actual DOM will be automatically updated whenever there are application changes related to content data or state. However, this process is time-consuming and high complexity to be completed. Therefore, the lightweight, in-memory object presentation is introduced as React virtual DOM. This DOM plays the key role in smooth re-rendering process, which effectively overwrites the default browser DOM manipulation. In case of updating DOM, the React app replaces only the component that has changes with the new updated version of that component, which minimizes the



computational operation and enhances the responsiveness of the web application. (Salcescu 2020, 23-24.)

### **3.2 JavaScript**

JavaScript, commonly abbreviated as JS, is an object-oriented programming language that is primarily used for creating websites. Its popularity comes from its ability to facilitate the development of dynamic web pages, which can easily interact with user engagement. JavaScript enables the implementation of diverse web features and easing the dynamic updating of page content. Its widespread use across various web browsers has made it an integral part of enhancing user experience. Together with HTML and CSS, JavaScript forms the essential core of the World Wide Web. (Wilton & McPeak 2010, 1-4.) Incorporating with React components, it helps in programming codebase and creating functions for state management and update (Larsen 2020, 10-15).

### **3.3 Redux**

Redux is well-known as an open-source JavaScript library for its ability to manage the state of applications. It is a state container that is easy to use and offers predictability, which makes it an ideal choice for constructing webpages or applications with consistent behaviour. The working method of Redux involves storing all the application's state in a centralized location called the store and returning updated state changes through the dispatching of actions resulting from user interactions. To manage this updating process, the reducer function is necessary in connecting state and action as input and returning new state, if necessary, corresponding to the action. Redux library is often employed alongside other libraries, particularly React, for developing the user interface of JavaScript applications. This collaboration ensures that the state of application components is updated consistently, resulting in predictable and seamless synchronization and application of content changes. (Caspers 2017, 13-14; Salcescu 2020, 70-83.)

### **3.4 HTML & CSS**

The term HTML stands for Hyper Text Markup Language, which is a standard markup language used for building web design and user interfaces. While it is not a programming language, it allows for the creation of the structure of a web page and the organization of its components. HTML comprises various elements that facilitate the display of content on a web browser. However, it is not sufficient to construct a webpage that is attractive and interactive. To address this issue, Cascading Style Sheets (CSS) were developed to aid in the appearance design of HTML documents. CSS allows for the presentation of HTML elements, including fonts, colours, and page layout, in an aesthetically pleasing manner. (Robbins 2012, 12.)

## **4 MIYABI SUSHI WEB DESIGN**

During this phase, the webpage design will be portrayed using three distinct parts consisting of web layout structure, content design, and mock-up design. These procedures are primarily intended to establish the overall appearance of the webpage, which is based on pre-determined objectives aimed at developing a visually attractive and user-friendly website. The web layout concentrates on the visual arrangement to form a wireframe of the webpage, while the content design section describes the organization of information as well as the visual support elements presented on the webpage. Lastly, the mock-up diagram is indicated and referenced in the final section, which describes the fundamental idea about the web appearance.

### **4.1 Web layout structure**

In accordance with the requirements for web design, the concept of layout design focuses mainly on the organization of content, which is presented in a concise and simple manner with full functionality. Additionally, if the layout is designed to be flexible by using media queries in the pages' codebase, it will allow the pages' content to adjust to changes in browser window size (Marcotte 2010). The Miyabi page layout will consist of three main sections: the header, body, and footer. The header includes the website's title or logo, as well as a navigation bar. The main content is displayed in the centre of the page for the website visitor to view. Meanwhile, the footer section contains basic information and contact details. These sections, forming the web layout, will be implemented with media queries to enhance the viewing context in different screen scopes. The detailed description of the page design, referred to in Figure 3, depicts the layout structure as building blocks.

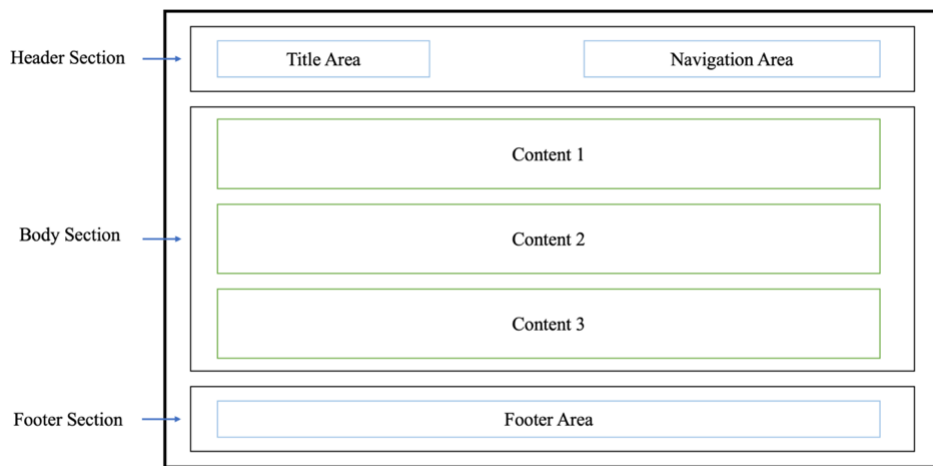


FIGURE 3. Layout of the website (adapted from piyushpilaniya98 2022)

In web development, it is crucial to arrange the information presented on a website. The reason is the effective organization can attract users' attention and reduce the time required for them to find their desired data. (Chaparro & Bernard 1999, 1.) Basing on the constructing requirements and web layout structure, the visual sitemap is generated and shown in the Figure 4 below, which displays a clear overview of the interconnection between pages. The web flat structure illustrates that the Miyabi Sushi website consists of 6 primary pages, including Home, About, Menu, Catering, Contact, and Online Order pages. Additionally, labelled blocks are utilized to showcase the basic content of each page, facilitating ease of navigation for users. Through implementing such visual sitemap, the usability and overall UX in browsing the website can be improved.

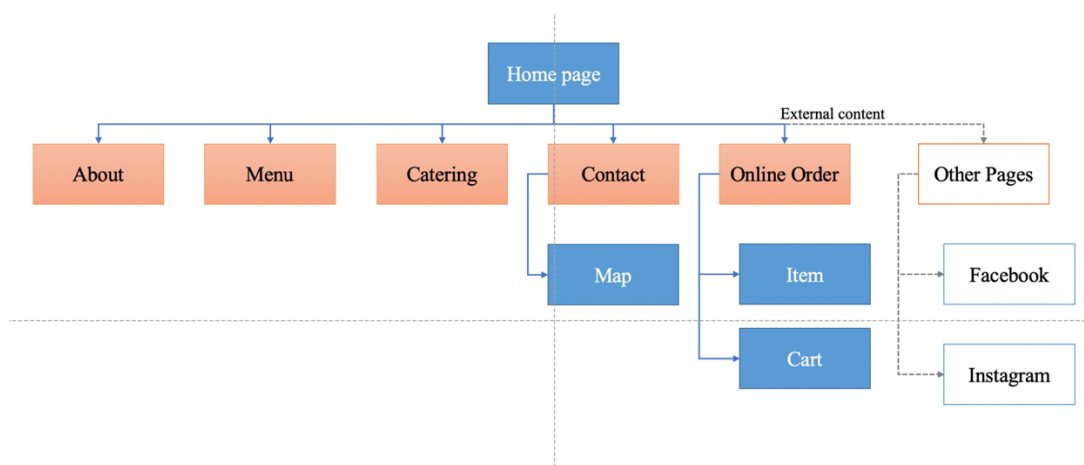


FIGURE 4. Visual sitemap (flat structure) of Miyabi Sushi website (adapted from Chaparro & Bernard 1999, 2)

## **4.2 Web content design**

The design of content is an important part of creating an ordered webpage that meets the users' unique demands of a page to improve the user experience. A good content management may improve navigation functionality while also broadening the visual appeal. (Newman & Landay 2000, 264-268.) The website should ensure that user can easily access and interact with the page content, which reaches the primary objective of offering Miyabi Sushi as a user-friendly and simple-to-use website. This section will introduce the content design of the Miyabi Sushi website through five distinct parts, namely hero section, navigation menu, typography, colour scheme, and photos and graphics. The combination of these elements is the fundamental concept to create the mock-up design in the later section.

### **4.2.1 Hero section**

The hero section, referring to the area of a webpage with the highest level of attraction and viewability, is a commonly used term in the web design community (Gerkules 2021). This part is mainly utilized to feature a hero image or video showcasing demonstrative content intending to encourage customer engagement with products or services. By emphasizing valuable information and minimizing user distraction on other elements, the implementation of this technique can effectively enhance user attention. (Pernice 2013.) The Miyabi webpage could implement a hero section at the top of its homepage, featuring a high-quality photograph highlighting the restaurant's main products – Asian fusion courses. The restaurant's name and slogan will be positioned in the centre of the hero image. Furthermore, a brightly coloured call-to-action button would also be placed beneath to promote the business's unique selling point. The intended purpose of this section is to make a strong initial impression on visitors, which results in reinforcing the restaurant's brand identity.

### **4.2.2 Navigation menu**

Laubheimer (2021) concludes that the ways in which humans seek out and explore new areas in a city are comparable to how web users navigate between pages to fulfil their information needs. Consequently, the navigation bar of a website becomes important in facilitating orientation between various

pages and categories. The navigation menu on the Miyabi Susi website is situated in the middle area of the header section. It is designed to adapt to different screen sizes, appearing as a linear menu in a large screen device and a hamburger menu icon in a small screen device. This menu serves as a directional tool to support in site browsing and exploration, containing a list of main pages such as Home, Menu, About, Contact, Catering and Online Order.

### **4.2.3 Typography**

Typography is a critical element in creating effective digital experiences. It has a significant impact on user experience, which is an essential component in the construction of digital products. The concept of typography encompasses various aspects, including font, font style, font weight, letter spacing, and line break. These factors are combined to improving legibility, enhancing a professional appearance, and strengthening brand identity of a product. (Fessenden 2019.) The chosen typefaces, or font family, for the Miyabi web development project are sans serif fonts, which do not have decorative lines at the bottom of the letters. The main fonts selected are Nunito and Dancing Script, which will be combined with other design elements to create a modern and youthful appearance. Additionally, the Montserrat and Confortaa fonts can be used as alternative options. All these fonts are part of the Google Fonts collection, ensuring that they can be legally used for the project and any commercial purposes after publication (Google Fonts 2023).

### **4.2.4 Colour**

Alongside with typography, colour is a crucial factor of product design serving as a potent tool for designers. The selection of colours in web and mobile design can significantly impact a user's attention, emotions, and perception of a brand. Effective use of colours on a website can bring numerous benefits, including enhancing the legibility of the text content. Moreover, colour combinations can express different emotions, with a combination of modest and soft tone provoking a formal atmosphere, while bright colours evoke a sense of enjoyment. Additionally, vibrant colours and gradients can capture visitors' attention emphasizing essential provided information through visually depth and contrast interfaces. Another commonly used technique is the overlay effect, which can direct attention towards specific design elements. (Babich 2017.)

According to the specification from the Miyabi business team, the colour scheme for the website should convey feelings of happiness, brightness, and freshness. As a result, the selection of primary colours for the website's appearance was done carefully. Two colour palettes were created to represent the main and secondary colour choices used across the website, demonstrated in Figure 5 below. In the first palette, an orangey yellow colour is principally used for the background, while a pastel orange shade is applied to the title header and primary buttons' background. For the contextual elements on the website, deep-sea green and elf green have been chosen, and the footer section background features a carousel pink colour. Additionally, other colours such as white, black, silver chalice, silver, and lemon grass are used as options for the secondary elements.

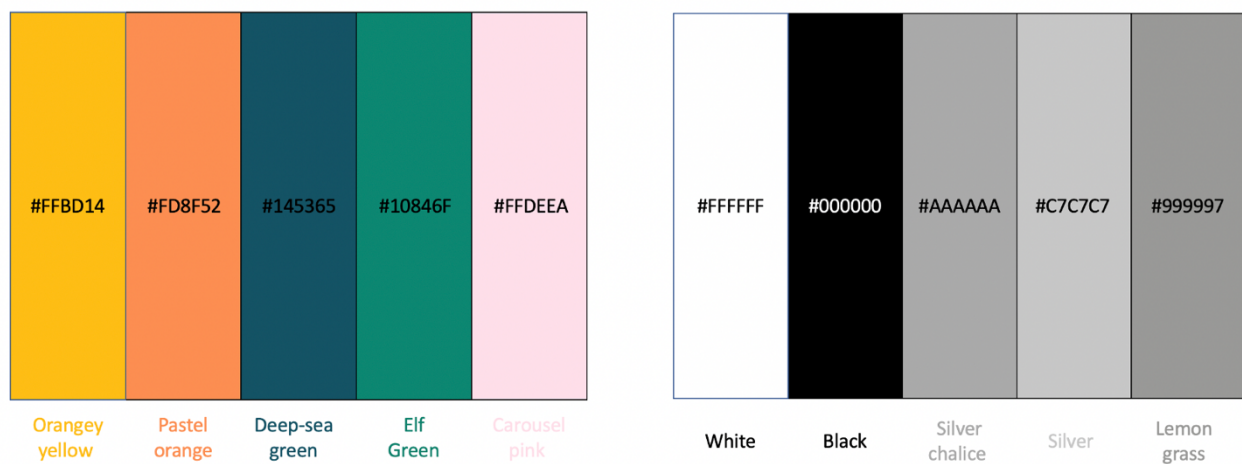


FIGURE 5. Primary and secondary colour palettes

#### 4.2.5 Images and Graphics

Graphic design plays an important role in web content design, particularly in the arrangement of visual elements on a website. By incorporating visual design and context, graphic design can stimulate the user's interest in exploring web content as well as motivate them to interact with provided products and services. (Palombi 2021.) Miyabi Sushi website will feature numerous images and graphics related to the restaurant's products, specifying sushi and Asian fusion cuisine. In addition, the short videos or GIFs can effectively attract user attention and create a visually engaging experience. These visual items have been provided by Miyabi Sushi restaurant owner and are consistently composed with the content on each page to create a more visually appealing and engaging user experience.

### 4.3 Mock-up design

A website mock-up is defined as a visual representation of a website that includes the final visual layout of the main site as well as additional designs such as page components, typeface, context, colour, and photos. Even if it is not the final appearance version, this static prototype allows the reader to view the final webpage's shape without any functionality. (Juviler 2022.) Before moving to the web implementation process, the mock-up designs of the Miyabi home page and menu page are made by using the Wix template and its development platform. Wix is a cloud-based website builder offering user an online environment, along with powerful tools and specialized features, to easily construct a professional website. The Figure 6 below is an example of using Wix builder platform to create the mock-up design. (Wix 2023.) The Miyabi Sushi web design adheres to the fundamental requirements of developing a website with aesthetic appeal, user-friendliness, ease of use, and simple navigation. The mock-up model provides an overview of the Miyabi home page and menu page, which supports to facilitate the web construction procedure later in the chapter.

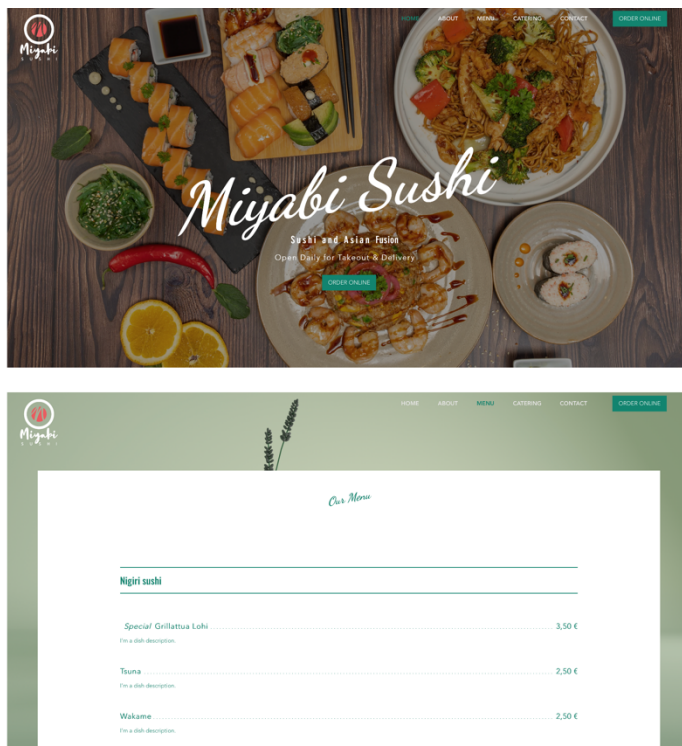


FIGURE 6. The landing page and menu page mock-up designs of Miyabi Sushi website



## 5 MIYABI SUSHI WEB DEVELOPMENT

The Miyabi Sushi website is built in Front-End as a ReactJS application using the technological tools discussed in Chapter 3. Without back-end programming, the project follows the primary purpose to create a responsive design for the application. All procedures linked to web creation are depicted in this chapter by breaking them into several phases. Before web construction process, the website type must be determined to ensure the site offers attractive content and excellent service. It will be followed by the web development environment section, in which a code editor is selected to provide a powerful environment with a collection of tools and capabilities to aid in the development of applications. Second, the ReactJS environment setup part will illustrate the process of installing ReactJS and other required libraries to execute the programming part. Finally, the development of the web pages in accordance with the initial specifications is discussed in detail.

### 5.1 Website type selection

There are two main types of websites as static and dynamic websites, each offering developer different methods to create web pages that meet specific requirements. A static website offers users with fixed content and containing minimum interactivity. It primarily consists of HTML and CSS files, allowing browsers to display content from HTML and CSS files loaded from the server. In this case, any updates to the website are made solely by the website owner. On the contrary, a dynamic website provides a more interactive experience, allowing users to actively engage with various sections on the pages. The content on this website type is dynamically updated based on user interactions and input. It is also connected to a database managed by backend server using programming languages such as PHP or ASP. This enables consistent content updates by retrieving and storing data. (Robbins 2018, 32.)

In this project, the development of the Miyabi Sushi website primarily relies on ReactJS and JavaScript to create a responsive web application. By utilizing React Hooks, the application state facilitates the management of web data changes through user interactivity, which is handled with the support of created JavaScript functions. Therefore, the app's components can be dynamically updated and re-rendered based on the current state of the application. This enables the creation of a dynamic UI and data manipulation, which are essential in building a responsive website and processing data in React app,

respectively. With these features and functionalities, the Miyabi Sushi website aims to be a dynamic website, providing customers with an excellent online service experience.

## 5.2 Web development environment

Before starting to develop applications, it is crucial to select an integrated development environment (IDE) that facilitates code writing and execution of commands. There are two common approaches to choose tools suiting the developer's skill set. The first option is to opt for an all-in-one IDE like Visual Studio Code (VS Code), which provides built-in tools and features. The alternative approach involves combining lightweight tools, such as an editor and a command line interface (CLI). (Wieruch 2017, 4; Wieruch 2022.) Applying to Miyabi Sushi web development, VS Code is chosen as the code editor that is an open-source desktop application developed by Microsoft. It offers a range of tools as well as supporting multiple programming languages and platforms via a rich module marketplace. This marketplace allows developers to customize and enhance their coding experience by installing compatible extension with existing software. (Microsoft Corporation 2023.)

After installing the Visual Studio Code software, it is essential to add several recommended extensions as shown in Figure 7 below. The first extension, ESLint, plays a crucial role in examining and identifying issues within JavaScript code files. Also, it provides suggestions for resolving these problems. Another noteworthy extension is ReactJS code snippets, which provides a selection of predefined code snippets to simplify the coding process by generating React components and patterns. Although not mandatory, it is advantageous to install specific extensions for HTML, CSS, and JavaScript. These software components assist the implementation of additional functionalities, code snippets, and formatting support beyond React components. Examples of these essential extensions include HTML CSS Support, Auto Rename Tag, and JavaScript (ES6) code snippets.

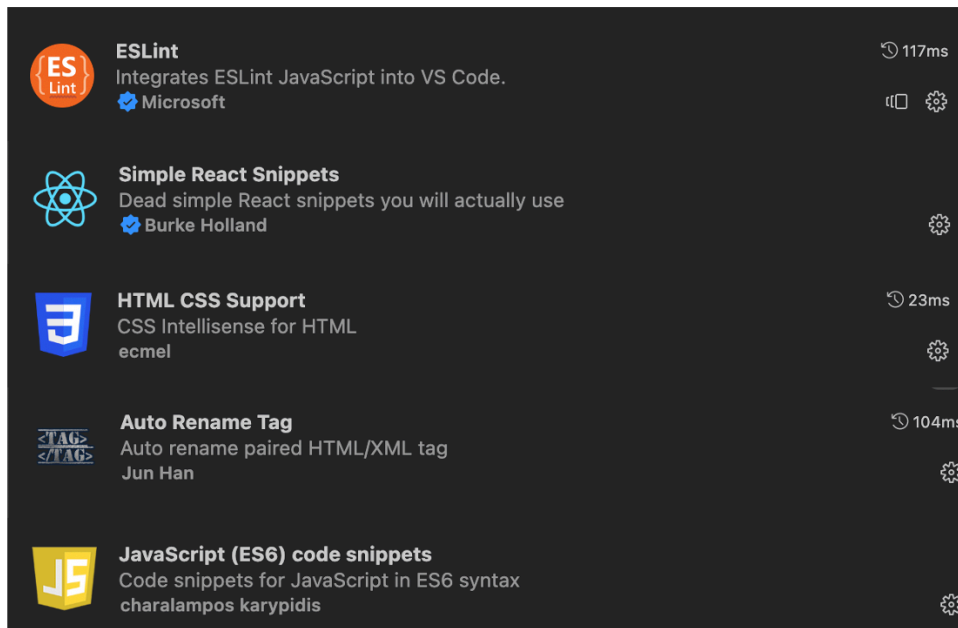


FIGURE 7. Recommended extensions to install (Microsoft Corporation 2023)

### 5.3 ReactJS environment setup

To create and execute a React application effectively, Node.js must be installed providing a runtime environment on the developer machine. Node.js is accompanied by npm, a package manager that facilitates the management of dependencies and project libraries. Node.js is featured to support compatibility across various platforms, including Windows, Linux, Unix, and macOS. Installation of Node.js is effortless, requiring the download of a pre-built installer from its official website that is available at <https://nodejs.org/en>. By utilizing npm through the command line interface, external libraries or packages can be readily installed. In the project's package.json file, all the required node packages are declared wrapped in a repository, which can be effortlessly installed by executing the npm install command. (Salcescu 2020, 10-11; Wieruch 2017, 5-6.)

There are two distinct methods employed in creating a React application using the node package manager. The first option is to execute either "npm init react-app appname" or "npx create-react-app appname", where "appname" refers to the desired name of the application. Once the React application is installed, the development server can be initiated and launched using "npm start" command, which enables the React app to be rendered and displayed in the user's browser. The accompanying Figure 8

demonstrates the utilization of the commands within the VS Code's built-in terminal to create a React application. Upon installation, the application encompasses the necessary files and can apply changes.

The image shows three sequential screenshots of a VS Code terminal window. The first screenshot shows the command `npx create-react-app miyabi_sushi_website` being executed. The second screenshot shows the command `cd miyabi_sushi_website` followed by `ls`, which lists the files and directories: `README.md`, `node_modules`, `package-lock.json`, `package.json`, `public`, and `src`. The third screenshot shows the command `npm start` being executed to start the development server.

FIGURE 8. React app installation and starting development server

To enhance the development process of the Miyabi Sushi webpage, several external packages have been incorporated. Figure 9 displays the contents of the `package.json` file, where all the built-in and additional packages, such as required fonts and icons, are stored. The installation of `react-icons` and `react-router-dom` was facilitated by executing the respective commands: `"npm install react-icons"` and `"npm install react-router-dom"`. The `react-icons` library offers a comprehensive collection of popular icons that can be simply integrated into the code as React components. On the other hand, the Router DOM package eases the navigation development work with its package's components, empowering the client-side routing control. This allows users to navigate between pages and React components smoothly. (Remix Software Inc 2023.) With the support of these packages, the design and functionality construction on Miyabi Sushi webpage are significantly boosted.

```

... {} package.json M X
{} package.json > {} dependencies > web-vitals
1  {
2    "name": "miyabi_sushi_website",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/jest-dom": "^5.16.5",
7      "@testing-library/react": "^13.4.0",
8      "@testing-library/user-event": "^13.5.0",
9      "classnames": "^2.3.2",
10     "react": "^18.2.0",
11     "react-dom": "^18.2.0",
12     "react-icons": "^4.8.0",
13     "react-redux": "^8.0.5",
14     "react-router-dom": "^6.11.0",
15     "react-scripts": "5.0.1",
16     "redux": "^4.2.1",
17     "styled-components": "^5.3.10",
18     "web-vitals": "^2.1.4"
19   },
20   "scripts": {

```

FIGURE 9. External packages installed

## 5.4 Project file structure

The project's file structure is of critical concern in creating an effective codebase. A well-organized and coherent file organization enables developers to easily manage the project elements or entire project, which softens considerable impact on the complexity of project maintenance and update in the future. While there are no strict rules on how project files should be organized, different developers may adopt diverse approaches. For instance, in the example of the Miyabi Sushi project structure illustrated in Figure 10 below, the front-end data source is organized into several essential hierarchical directories. This division into clear and concise subfolders reduces the time and effort required for code management and simplifies future updates.

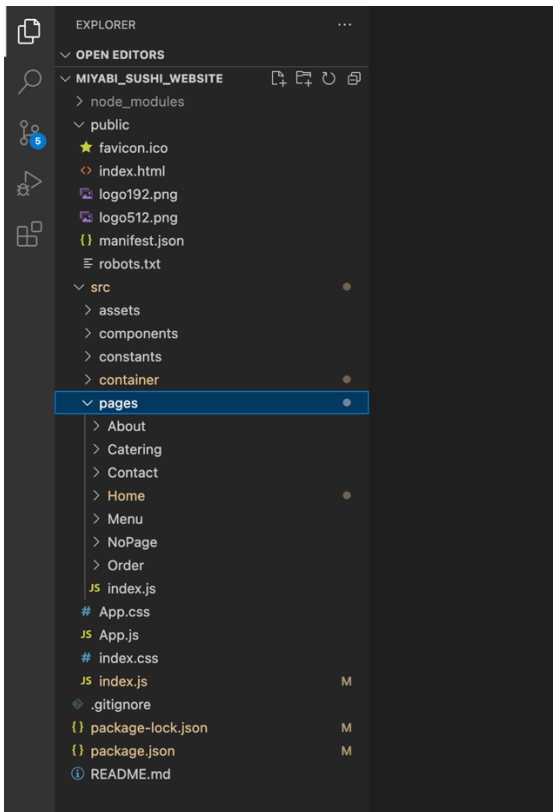


FIGURE 10. The Miyabi Sushi web project file structure

According to the indicated content in Figure 10, the “public” directory stores static files that are accessible to the public, such as `index.html`, `manifest.json`, and logo files. The main repository of the web application is “src”, containing all source code for the React app. Inside this directory, the folder “assets” serves as a storage of image and font files used by the app components, while the “component” folder houses reusable code files that encapsulates specific functionalities and features of app system. It can be further organized into subdirectories based on the purpose of each component. Furthermore, all constant data or related to constants files can be found at “constants” directory providing the assets export files in this case. The following folders, “containers” and “pages”, storing program files that depict specific sections of a page and main pages, respectively. On the other hand, the root component is wrapped in the “App.js” file, allowing the implementation of React app to be rendered in the entry point, “index.js”. Meanwhile, the “App.css” and “index.css” files are used to apply design styles to components and the application. Lastly, the “package-lock.json” and “package.json” file, located at the bottom of file structure, aid in tracking dependency versions and holds metadata for the application, project dependencies, and scripts.

## 5.5 Web construction

A series of necessary steps are followed to achieve a responsive design aligned with the final requirements mentioned in chapter 2. The initial phase involves creating the navigation bar and footer of the website, showcasing the appearance of the navigation in the web view. It also reveals the implementing techniques to switch between different nav menu types on both the desktop screen and mobile screen. Subsequently, the focus shifted to the main pages, namely Home, About, Menu, Contact, Catering, Online Order, and other pages, with each section being analysed in-depth and outlined separately. Finally, the render section encompasses the implementation of the root component, which enables the website to be displayed on a local development server.

### 5.5.1 Navigation bar and footer

In accordance with the sketched layout and web design specifications, the header section of the page contains the restaurant's logo, navigation bar, and a call-to-action button. The navigation bar comprises a menu that facilitates the navigation and access to various pages of the website, including Home, About, Menu, Contact, and Catering. This results in offering a consistent user interface across different pages. As depicted below in Figure 11, the Miyabi logo is positioned on the left-hand side, while the online order button is situated on the opposite side. This design allows swift access to both the home page and the online food ordering page respectively. Furthermore, the background of this component section is set to be black with a 0.2 opacity setting, generating a transparent effect that allows user to see the background behind it.

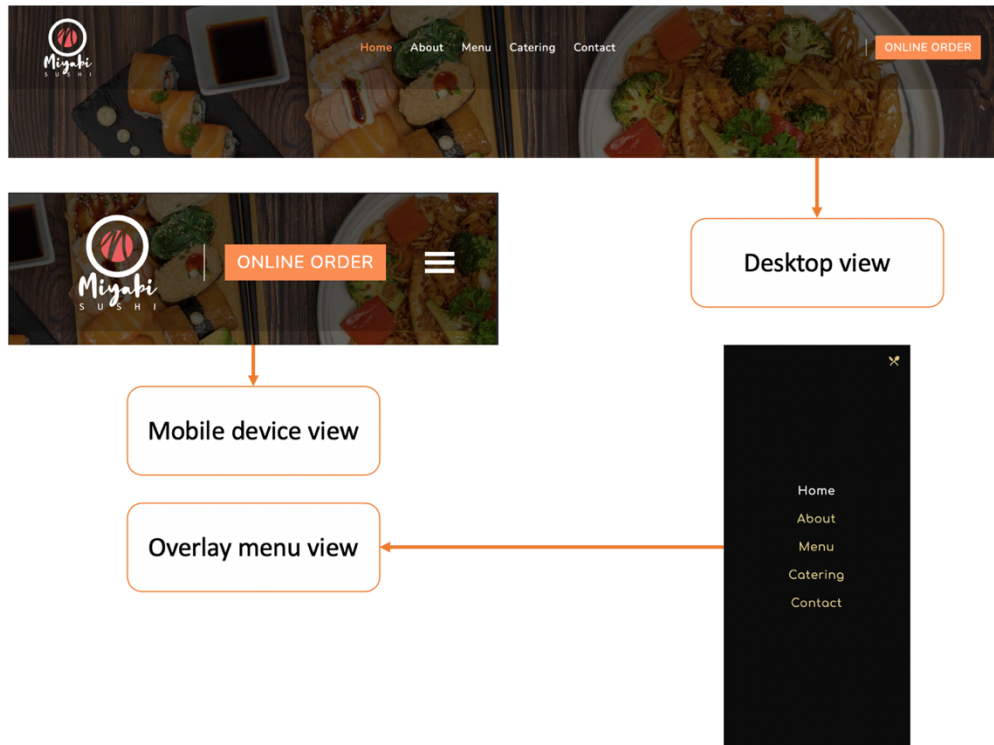


FIGURE 11. Miyabi Sushi website - navigation bar on desktop and mobile device views

The size of the header section is adjusted dynamically based on the user's current viewport, as illustrated in Figure 11. In the desktop interface, the entire navigation menu is centred within the parent section, enabling convenient access and navigation between pages. On the other hand, this menu will automatically transform into a hamburger menu icon when user resizes the browser window or accesses the pages on a mobile device. This approach simplifies the user interface and improves readability on devices with limited screen space, contributing to making the Miyabi Sushi website more user-friendly. By clicking on the hamburger menu icon, a navigation menu list overlay appears on top of the current screen view, empowering the user to interact with the page menu once again.

To attain nav bar design and functionalities, the React Hook - `useState` is utilized to create a `handleMenuOpen` function, which manages the open state of the overlay navigation menu on a small screen browser. As implemented in “Navbar.jsx”, the `useState` hook is imported on the top of the program before declaring the `toggleMenu` state variable. Ultimately, the `handleMenuOpen` function was employed to control and update the current open state of the overlay navigation by using `toggleMenu` as a variable and `setToggleMenu` function for update the value of `toggleMenu`. (CODE 1.) Besides, the collapsing of category menu into hamburger icon is handled through the media queries, as shown in the Code 1 below within the “Navbar.css” file.



```

import React, {useState} from 'react';
import {GiHamburgerMenu} from 'react-icons/gi';
import {MdOutlineRestaurantMenu} from 'react-icons/md';
import {Link, useLocation} from 'react-router-dom';
import images from '../constants/images';
import './Navbar.css';

const Navbar = () => {
  // Define variable to store the opening state of hamburger Menu overlay screen
  const [toggleMenu, setToggleMenu] = useState(false); // Switch for opening the small screen menu
  // Define variable to store the location of the current page
  const location = useLocation(); // Checking current location (which page?) to add new class

  // Handle open menu overlay in the order online page
  const handleMenuOpen = () => {
    setToggleMenu(!toggleMenu);
  }
}

```

Navbar.jsx

```

/* menu position and style for mobile */
.app_navbar-smallscreen {
  display: none;
}

.app_navbar-smallscreen_overlay {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100vh;
  background: var(--color-black);
  transition: 0.5s ease;
  flex-direction: column;
  z-index: 6;
}

/* The style and size for diffent screen dimension */

@media screen and (min-width: 2000px) {
  .app_navbar-logo img {
    width: 180px;
  }
}

@media screen and (max-width: 1150px) {
  .app_navbar-links {
    display: none;
  }
  .app_navbar-smallscreen {
    display: flex;
  }
}

```

Navbar.css

CODE 1. Managing open state and responsive screen resize with useState hook and media queries

The website footer is specifically designed to hold all essential information about Miyabi Sushi restaurant. As shown in Figure 12, the footer section consists of the contact details of Miyabi Sushi, including the address, email, and phone number. Moreover, the opening hours of the restaurant are positioned in the right corner, while the icons in the middle provide links to the restaurant's social media platforms, granting customer easy connect with the restaurant through these channels. Situated beneath, the restaurant's partner logos provides the association with alternative online order system, offering home delivery option. Lastly, the same technique of using media queries is applied to enhance the responsiveness of the footer between different screen dimensions.

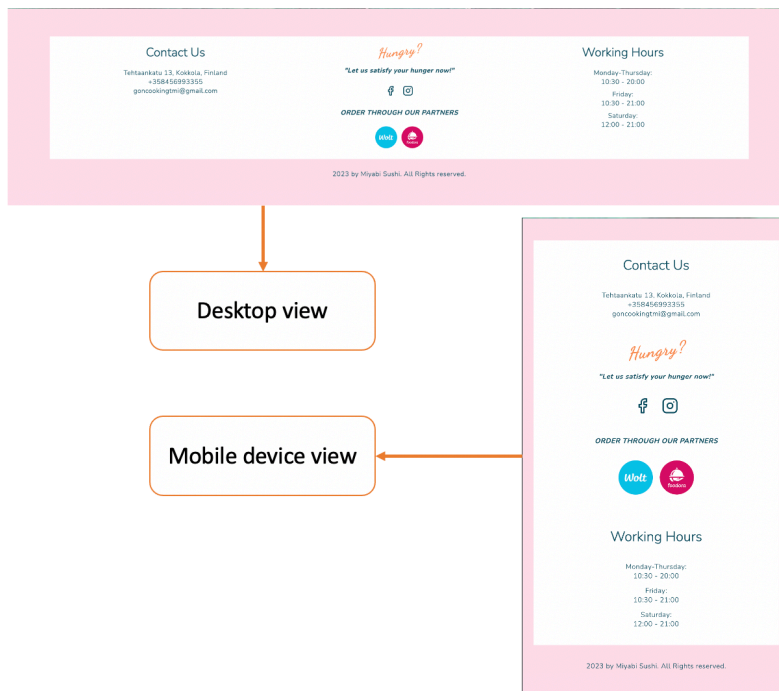


FIGURE 12. The footer of Miyabi Sushi website

## 5.5.2 Home page

The Home page serves as the primary page or landing page of the website. It provides visitors with general information about the restaurant, including the menu, a gallery of food items, and options for catering services. The information on the page is organized into separate blocks, arranged vertically from top to bottom of the page. Each section is implemented using distinct components, as depicted in Figure 13. The hero section is placed on the top, acting as the user's entry point upon accessing the website. It showcases the restaurant's brand name and main products at the section core. Moreover, the hero image, applied as the background of this section, displays appetizing dishes that enhances the overall attractiveness of the website and stimulating customers' appetites. In the implemented code, the Home.jsx file brings together all the section components, such as the Header, OurProducts, Gallery, and OurCatering, in a sequential manner.

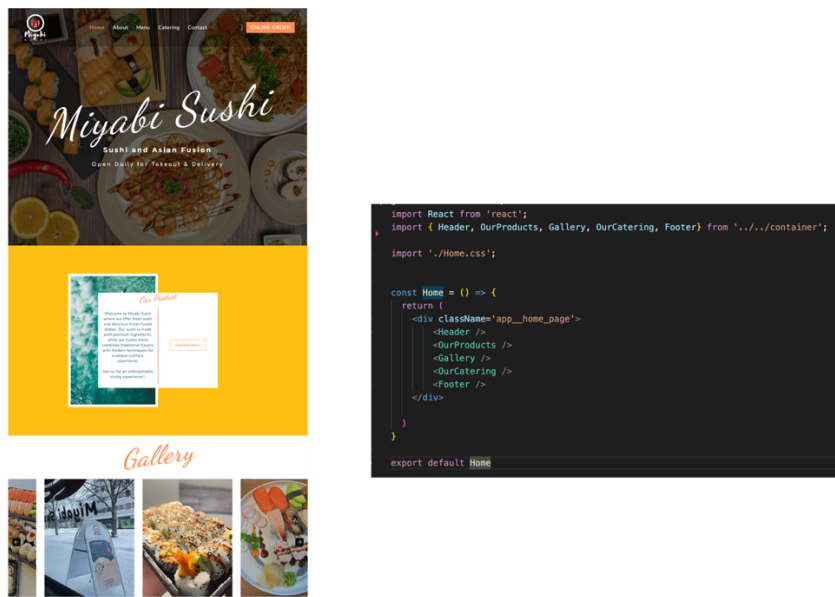


FIGURE 13. Miyabi Sushi – landing page design

The image files need to be imported before the program can access the photo files for the gallery and construct the "galleryImages" collection. Each declared picture is then placed in an individual box, and all of them is wrapped within a gallery container, as depicted in Figure 14 presented below. The useRef hook is utilized to retain a reference to the container element, entitling control over the current view of the gallery container. Consequently, customers can navigate through the gallery images by utilizing arrow buttons to move forwards or backwards. This helps visitor to view more picture in the collection. Moreover, when hovering over an image, an Instagram link becomes visible and establishes a connection to the restaurant's Instagram profile.

## Gallery



```

1 import React, {useRef} from 'react';
2 import {BsInstagram, BsArrowLeftShort, BsArrowRightShort} from 'react-icons/bs';
3 import { SpecialSubHeading } from '../././components';
4 import { images } from '../././constants';
5 import './Gallery.css';
6
7 // Defining an array of gallery images to store image references
8 const galleryImages = [images.miyabi_gallery01, images.miyabi_gallery02, images.miyabi_gallery03,
9   images.miyabi_gallery04, images.miyabi_gallery05, images.miyabi_gallery06, images.miyabi_gallery07,
10  images.miyabi_gallery08, images.miyabi_gallery09, images.miyabi_gallery10];
11
12 const Gallery = () => {
13   // Assign a variable to store the specific element reference by using useRef Hook
14   const scrollRef = useRef(null);
15
16   // Function is defined to handle scrolling the gallery left or right
17   // based on the provided direction parameter
18   const scroll = (direction) => {
19     const { current } = scrollRef;
20     if (direction === 'left') {
21       current.scrollLeft -= 300;
22     } else {
23       current.scrollLeft += 300;
24     }
25   };

```

FIGURE 14. Miyabi Sushi website – Gallery section

### 5.5.3 About page

In Figure 15 below, the About page presents an introduction to the Miyabi Sushi restaurant, encapsulated in a centrally positioned paragraph to capture the viewer's attention effectively. The passage provides information about the restaurant's establishment, its primary services and mission. Additionally, it draws up the long-term strategy and direction of the restaurant in the next few years. The background design follows a simple style with two decorative pictures positioned behind the text content container, which mostly exploits the CSS features. This ensures that the introductory paragraph box stands out distinctly from other elements on the page.

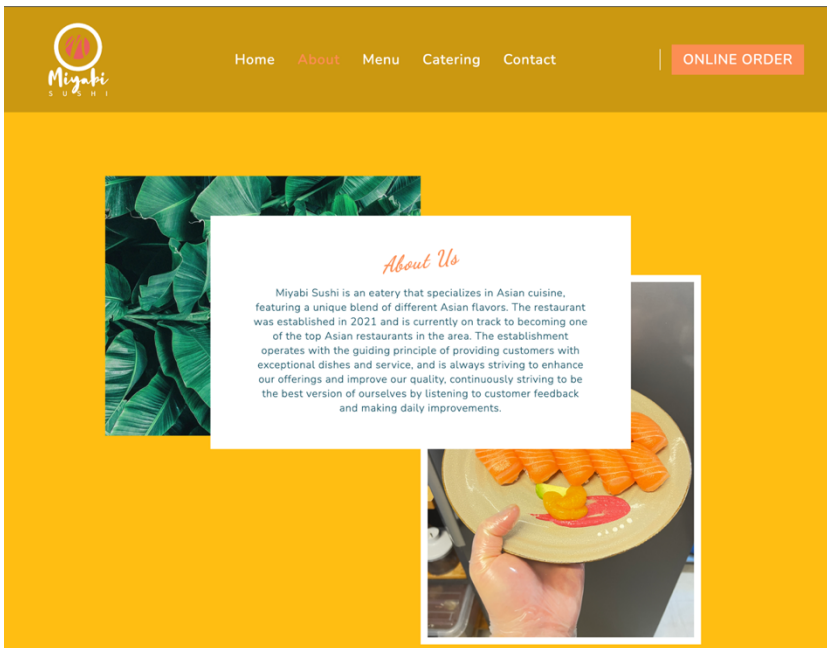


FIGURE 15. Miyabi Sushi website – About page

## 5.5.4 Menu page

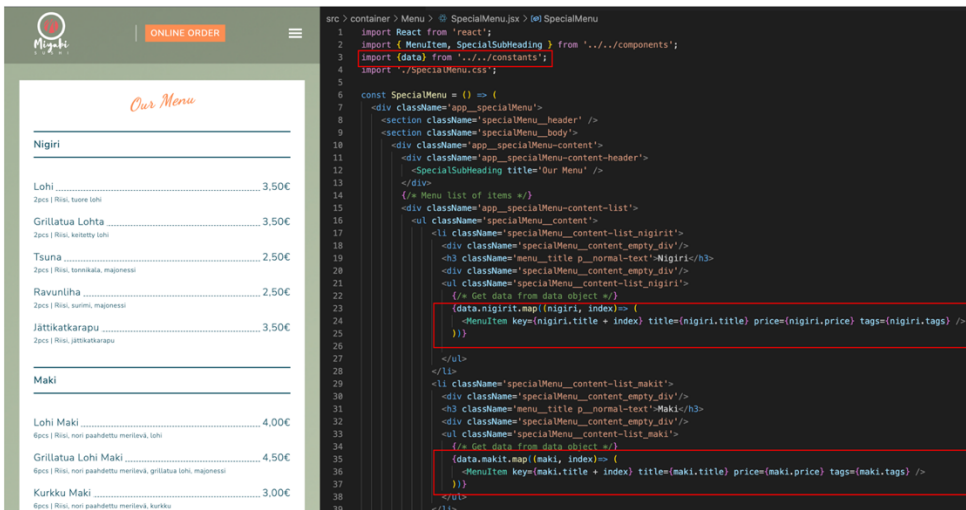


FIGURE 16. Miyabi Sushi website – Menu page

On the Menu page, customers can explore the list of all the products available at the restaurant along with detailed description. Each item is presented in a segmented format on the menu, showcasing its name, portion size, ingredients, and price. Similar food items are grouped together forming specific lists that are retrieved from a demo data file, which can be replaced by a real database in the backend

development process. By utilizing the built-in "map" function, the item information is iterated upon and passed into the MenuItem component from the imported data file, which results in a complete menu list. The MenuItem component is then implemented to display the item list with a professional designed style, as illustrated in Figure 16 above.

### 5.5.5 Contact page

The appearance of Contact page, as depicted in Figure 17, is created by integrating three distinct sections: a contact form, a map, and a GIF. The contact form allows users to send feedback or directly request contact to the restaurant email. To emphasize the mandatory input requirement, the email address field is marked with an asterisk symbol, preventing the missing of communication means. Furthermore, the map section exhibits the restaurant's location by utilizing the Google API, which displays the precise address and pinpointing the location on the map. Also, the review section is attached to the address information box, creating a connection linked to the google review page. Below the map section, a GIF is implemented with the purpose of enhancing the visual appeal of the page and expressing gratitude from the restaurant team to the customers.

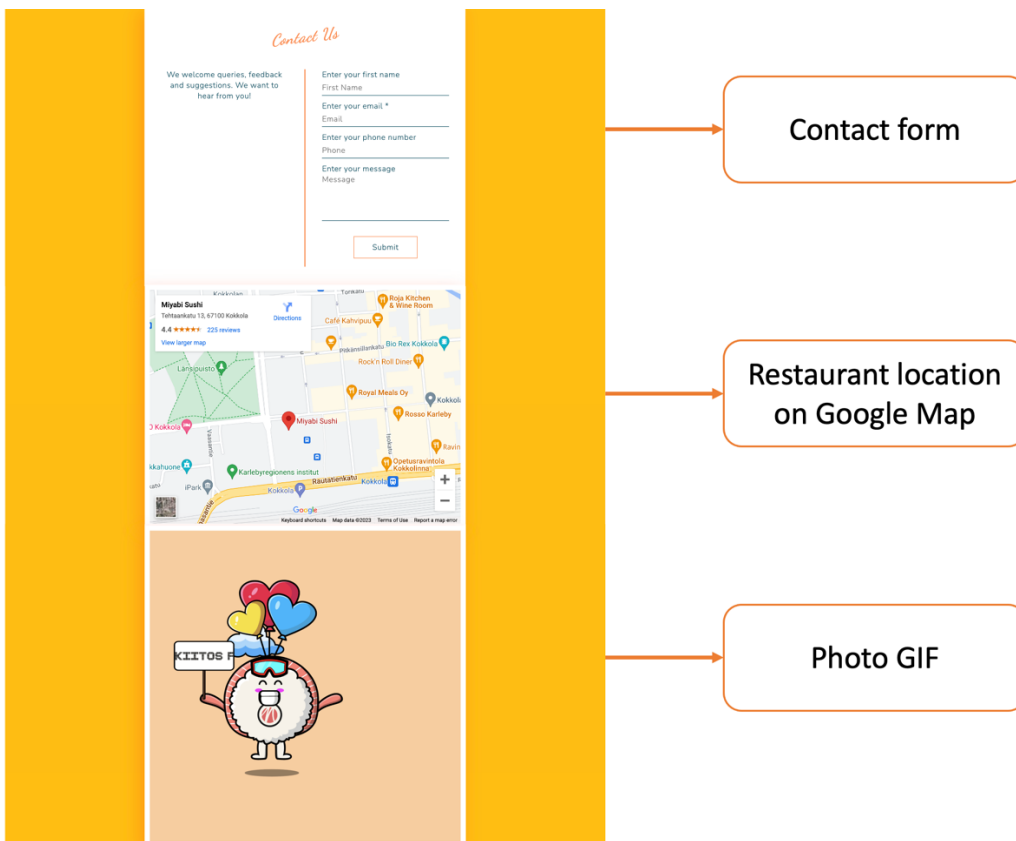


FIGURE 17. Miyabi Sushi website – Contact page.

The implementation of the contact form primarily involves the employment of input elements. To associate the title of input with the respective input field, the label element is exercised. This approach improves the accessibility and usability of the form, enabling other assistive technologies to enhance the user experience. When a user clicks on the label text, the connected input field receives focus, indicating its active state and readiness for the next action. As depicted in the example in Figure 18, clicking on the label for the email input field triggers a design transformation. This leads to changes in the background colour, turning it pink, as well as adds the border style to the field box.

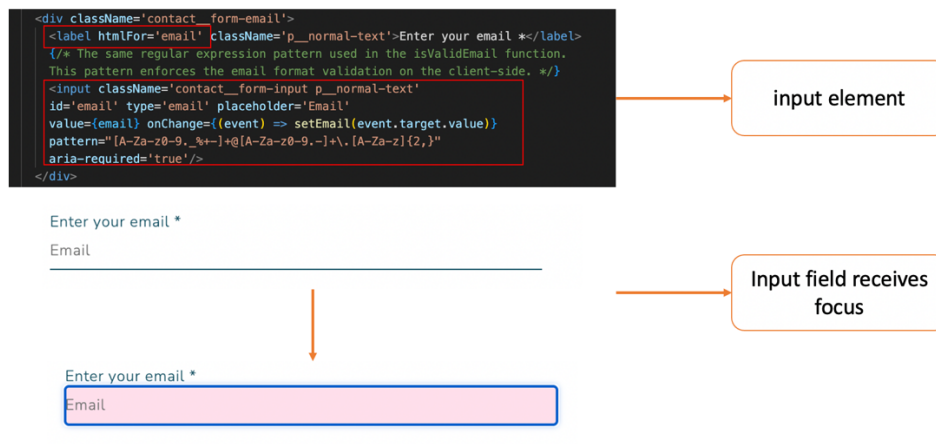


FIGURE 18. Email segment input in contact form using label and input elements

As introduced earlier, the map section includes `<iframe>` element that embeds the Google API to display the restaurant's location. In Code 2 below, the inline frame, commonly referred to as an `iframe` in web development, works as a container for integrating an external source into a specific section of the webpage. Consequently, the map is rendered as an HTML document, covering a particular area within the main view. Additionally, various attributes such as `style`, `size`, `title`, and functional settings also need to be added within the `iframe` element to make certain the correct visual presentation. For instance, the width and height values inside the `iframe` are adjusted to 600 and 450 respectively to fit the dimensions of the outer map section container (CODE 2).

```

/* Ref attribute is used to assign a reference to the element */
<div className='app_contact-map' ref={mapContainerRef}>
  <div className='app_contact-map-content' ref={mapContainerContentRef}>
    <iframe src='https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d1759.2857523330215!
    2d23.124912916258438!3d63.8358835834557!2m3!1f0!2f0!3f0!3m2!111024!21768!4f13.1!3m3!1m2!
    1s0x46874892f88fb80d%3A0xb87d2988ebc51151!2sMiyabi%20Sushi!5e0!3m2!1sen!2sf1!4v1684099938753!5m2!1sen!2sf1'
    width='600' height='450' style={{border:0}} allowFullScreen='' loading='lazy'
    referrerPolicy='no-referrer-when-downgrade' title='miyabi map location'></iframe>
  </div>
</div>
/* Photo GIF section */
<div className='app_contact-photo' ref={photoContainerRef}>
  <div className='app_contact-photo-box' ref={photoContainerBoxRef}>
    <img src={img.miyabi_kiitos_gif} alt='miyabi kiitos' />
  </div>
</div>

```

Embed map API

## CODE 2. Map section implementing Google API for map

### 5.5.6 Catering page

The Catering page provides comprehensive information about the restaurant's catering options, offering customers an overview of the available services. The service types are categorized into a la carte, buffet, and online order, presented separately below the introductory paragraph box as described in Figure 19. Each service category container holds a content box at the centre, which provides the category information in detail. In addition, these containers are visually differentiated by varying background designs, highlighting the unique characteristics of each individual service category. To produce this design, HTML and CSS features are mainly applied to implement the necessary content and elements' style, improving the visual appeal of the page.

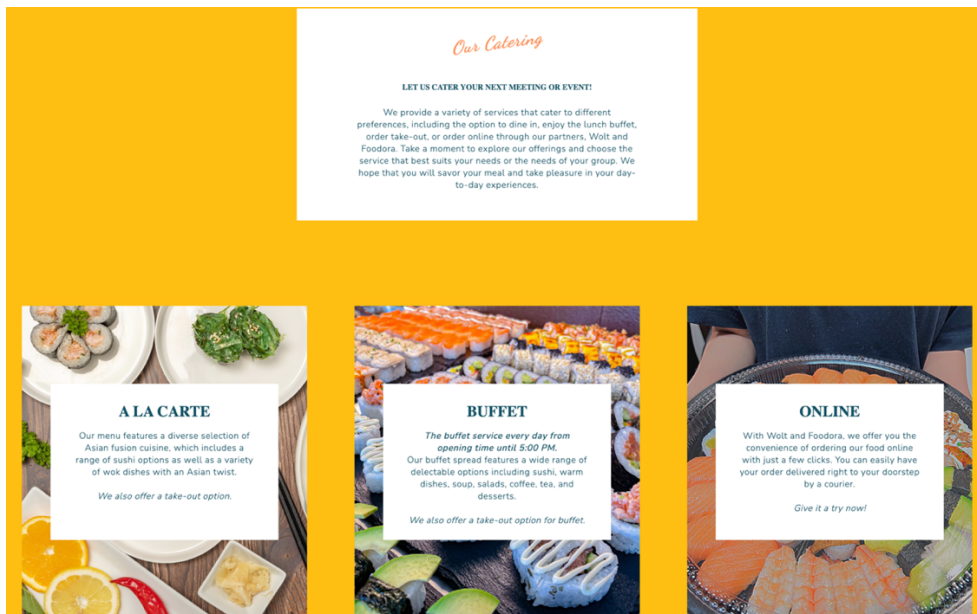
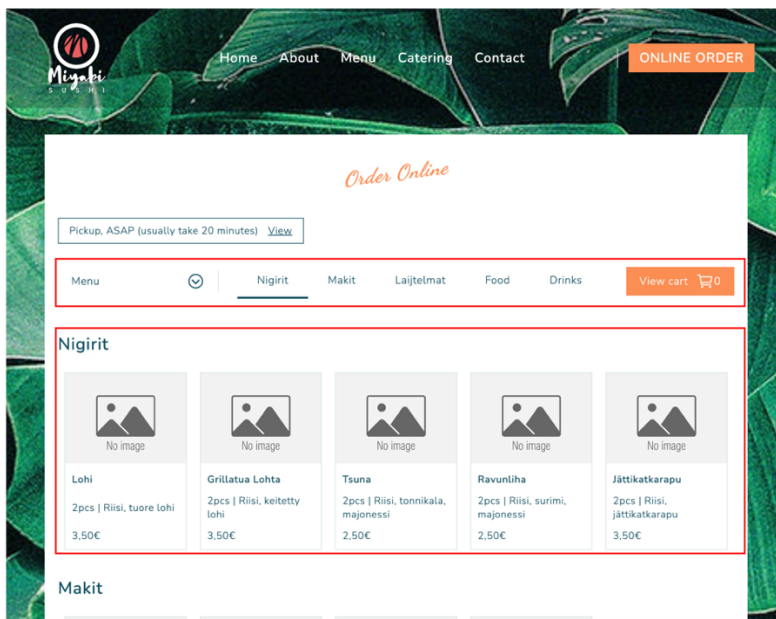


FIGURE 19. Miyabi Sushi website – Catering page

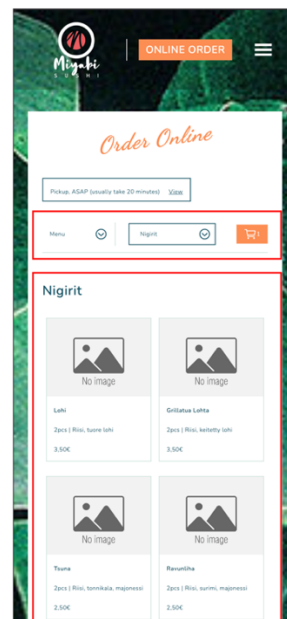


### 5.5.7 Online Order page

The Online Order is an essential page in building an e-commerce website enabling customer to place order online conveniently. In this project, this page features the same item list as seen on the Menu page. However, each item here is enveloped within a flexible container box that modifies dynamically with the screen dimension. As can be seen in Figure 20 below, the Nigirit sushi category section illustrates the items in different boxes, which is followed by other categories such as Makit, Laijtelmat, Foods, and Drinks. To broaden user experience and accessibility, a sub-menu component has been incorporated. This sub-menu bar offers the possibility of changing the menu selection, navigating between categories, and displaying the online shopping cart. The categories on nav bar provides a list of product types offered by the restaurant. When a category name is clicked, the browser screen will automatically scroll to the selected category's position. This feature facilitates users' understanding of the online order menu structure and offers user-friendly navigation within the responsive application.



Online order page on Desktop view



Online order page on Mobile view

FIGURE 20. Miaybi Sushi website – Online Order page on desktop and mobile view

To place an order, customers can first select an item box. Corresponding user interaction, an overlay item container will pop-up over the current page view, providing a full view of the item's description

and an add-to-cart section. In the add-to-cart section, users can find the quantity selection and a confirm button. The total price of a selected item is also displayed within the “Add to my order” button, which is automatically updated based on the adjusted quantity. Following that, the data of the selected item, along with the confirmed quantity and total price, will be sent to the Cart component. Figure 21 below illustrates an example, showcasing the "grillatua lohta sushi nigiri" option available for purchase.

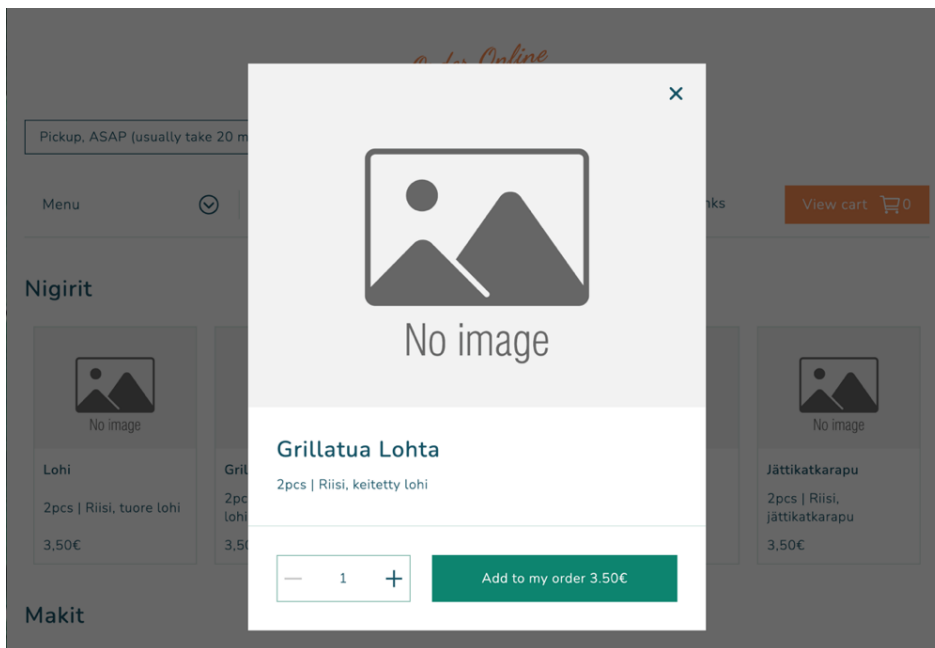


FIGURE 21. Online Order page – Online order item pop-up

In the subsequent step, users can access the online shopping cart to review the currently selected products, as depicted in Figure 22. The Cart component is imported and displayed on top of the current view, demonstrating an overview of the order's information. This enables customers to review their order before proceeding to the next stage, the checkout process. Besides, customers can make changes on their selections again, such as increasing or decreasing the quantity of an item, removing a selected product, adding a private note, and viewing the subtotal price for the entire order. However, as this project primarily focuses on client-side web development or Front-End development, the checkout stage has not been implemented, which shows the notification above the checkout button.

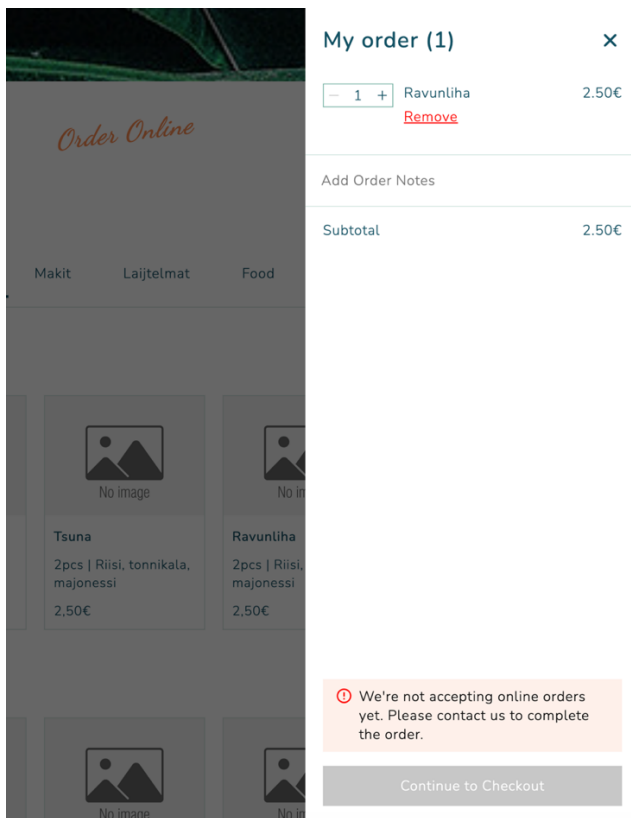


FIGURE 22. Online Order page – Online shopping cart section

Alongside, the page also offers users the opportunity to modify the pick-up option from a box located above the sub-menu bar. By default, only the estimated order processing time is displayed within the box to inform the customers about the amount of time the restaurant requires to complete the order. Upon clicking the "view" button, as depicted in Figure 23 below, a white box will appear overlaying the current page view, showing the detailed information about the pick-up service. This feature allows customers to change the current service point to another branch location for order collection. However, since Miyabi Sushi restaurant has only one branch, the option to change the service point is announced as unavailable.

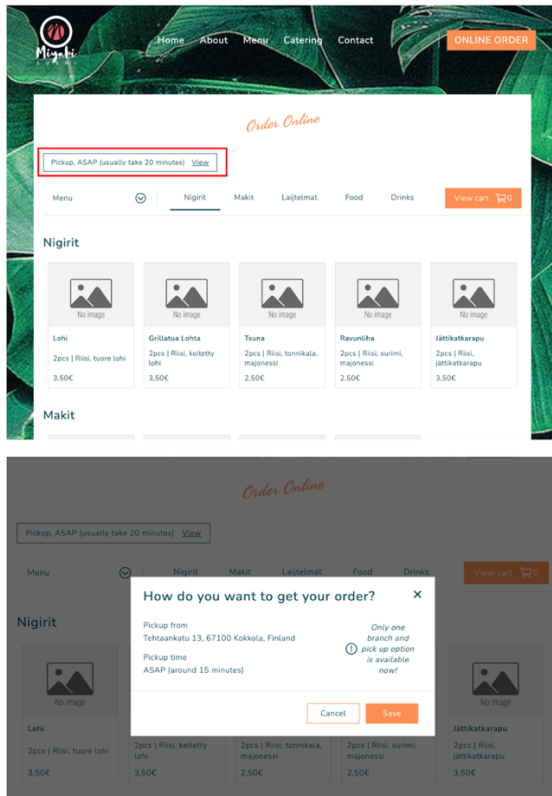


FIGURE 23. Online Order page – Online order pick-up option

To achieve the desired result illustrated above, various techniques are employed to construct the Online Order page. In the page content, there are multiple hidden sections that can be revealed through user engagement. These sections utilize different setups, including menu dropdowns, pop-ups, and overlays, to enhance the functionality of the page. Like the approach used for the navigation bar and footer components, the `useState` hook is imported to primarily manage the open state of each overlay section in the codebase. The code snippet in Code 3 describes the import statement at the beginning of the file, which includes the required built-in hooks, external data, and components. Moreover, inside the “OrderOnline” function, there is the declaration of the open state variables for each specific section, along with their initial values. These variables are updated through their respective “set” functions in response to changes made by visitors.

```

OrderOnline.jsx ×
src > container > OrderOnline > OrderOnline.jsx > OrderOnline > toggleDropdown
1 import React, {useState, useEffect} from 'react'
2 import {IoIosArrowDropdown} from 'react-icons/io';
3 import {IoCartOutline} from 'react-icons/io5';
4 import { OrderItem, PickupOption, SpecialSubHeading, SubMenuNav, Cart } from '../components';
5 import {data} from '../constants';
6 import './OrderOnline.css';
7
8 const OrderOnline = () => {
9   // Define the state for storing the state of dropdown menu
10  const [isOpen, setIsOpen] = useState(false);
11
12  // Define the state for storing the state of pick up option box
13  const [isOpenPickup, setIsOpenPickup] = useState(false);
14
15  // Define cart items to store selected items
16  const [cartItems, setCartItems] = useState([]);
17
18  // Define the variable to control the ?open state of the Cart component
19  const [isOpenCart, setIsOpenCart] = useState(false);
20

```

CODE 3. Online Order page – code implementation for controlling section overlay and pop-up

The order listing item data is stored in the local storage, which are loaded whenever the user accesses the Online Order page using the `useEffect` hook. This data is then assigned to an array variable called `cartItems`, which is controlled by the `useState` hook, enabling the possibility of later updating variable value later. As depicted in Code 4 below, `cartItems` is defined and updated through the `handleUpdateItems` function. The `addToCart` function, declared in the `OrderOnline` component, is passed to its child component, the `OrderItem` component. This allows the application to trigger a callback function to add a new item to the order list. Additionally, another call-back function from `OrderOnline`, `handleUpdateItems`, is passed to the `Cart` component as a parameter. This ensures that the selected items list is updated using the `useEffect` hook whenever the user makes a new change, thereby affecting the original item list passed in the online shopping cart section.

<pre> // Define cart items to store selected items const [cartItems, setCartItems] = useState([]);  const addToCart = (title, quantity, price, totalPrice) =&gt; {   const newItem = {     title: title,     quantity: quantity,     price: price,     totalPrice: totalPrice   };   setCartItems([...cartItems, newItem]); };  const handleUpdateItems = (updatedItems) =&gt; {   setCartItems(updatedItems); };  const toggleCart = () =&gt; {   setIsOpenCart(!isOpenCart); };  useEffect(() =&gt; {   // Fetches saved cart items from local storage   const savedCartItems = localStorage.getItem('cartItems');   if (savedCartItems) {     // Parse and set the cart items from local storage     setCartItems(JSON.parse(savedCartItems));   }, []); </pre>	<pre> const OrderItem = (imgUrl, title, price, tags, addToCart) =&gt; {   // Function to get the total amount according to the product quantity   const handleAddToOrder = () =&gt; {     const totalPrice = (numericPrice * quantity).toFixed(2);     addToCart(title, quantity, price, totalPrice);   }; </pre>	<pre> const Cart = ({items, onClose, onUpdateItems}) =&gt; {   // Define the cart items array to store the passed in items array   // From parent component (in this case is Online Order)   const [cartItems, setCartItems] = useState(items);    // Hook to update the saved array as well as pass-in array   useEffect(() =&gt; {     onUpdateItems(cartItems);     localStorage.setItem('cartItems', JSON.stringify(cartItems));   }, [cartItems, onUpdateItems]); </pre>
OrderOnline component	OrderItem component	OrderItem component

CODE 4. Online Order page – code implementation for order data update

Likewise, the updated quantity function is employed to implement the changes in OrderItem component and Cart component made by the customer, as described in the Code 5 below. Although the two code snippets may differ, the underlying idea and logical approach remain the same. Customers can modify the quantity of a product by clicking the minus and plus button icons or by directly entering a new quantity number into the input field. In the OrderItem component, each update type is managed by a separate function, while the Cart component uses a single function to handle all changes. Additionally, before any modifications are applied, certain conditions are evaluated to ensure the logical correctness within the function.

```
// Define quantity variable to store the state of the quantity
const [quantity, setQuantity] = useState(1);
// Extracting numeric value from price prop
const numericPrice = parseFloat(price.replace(',', ''));

// Function to increase the quantity
const incrementQuantity = () => {
  setQuantity(quantity + 1);
};

// Function to decrease the quantity
const decrementQuantity = () => {
  if (quantity > 1) {
    setQuantity(quantity - 1);
  }
};

// Function to get the total amount according to the product quantity
const handleAddToOrder = () => {
  const totalPrice = (numericPrice * quantity).toFixed(2);
  addToCart(title, quantity, price, totalPrice);
};

// Function to update the quantity change
const handleQuantityChange = (event) => {
  const newQuantity = parseInt(event.target.value);
  // Check if the new quantity is valid (not empty, 0, or NaN)
  if (!isNaN(newQuantity) && newQuantity > 0) {
    setQuantity(newQuantity);
  }
};
```

OrderItem - update the item quantity

```
// Function to handle the quantity change after user adjusts the value
const handleQuantityChange = (index, newQuantity) => {
  setCartItems(prevItems => {
    const updatedItems = prevItems.map((item, i) => {
      // Change and update the data of an item
      if (i === index) {
        // Check if the new quantity is valid (not empty, 0, or NaN)
        if (newQuantity && !isNaN(newQuantity) && newQuantity !== 0) {
          // const newQuantity = item.quantity + change;
          const newTotalPrice = newQuantity * parseFloat(item.price.replace(',', ''));
          return {...item, quantity: newQuantity, totalPrice: newTotalPrice};
        }
      }
      return item;
    });
    return updatedItems;
  });
};
```

Cart - update the item quantity

CODE 5. Online Order Page – code implement for updating item quantity

### 5.5.8 404-error page

If a customer attempts to access a non-existent path of the website, the web app will automatically navigate to the 404-error page, indicating that the current link is invalid. This method helps in appropriately responding to viewers when encountering an invalid or broken link, which ensures the overall performance of the website as well as the smooth browsing experience. As shown in Figure 24, the visual design of this error page is a combination of vector graphics obtained from Freepik website. The premium licenses provided by the Miyabi Sushi owner allow for unlimited usage without the need of attribution. These vectors were modified to create the final design, featuring a background image with an error message. To resolve this situation, users can either use the correct website link or navigate to other valid pages using the navigation bar menu.

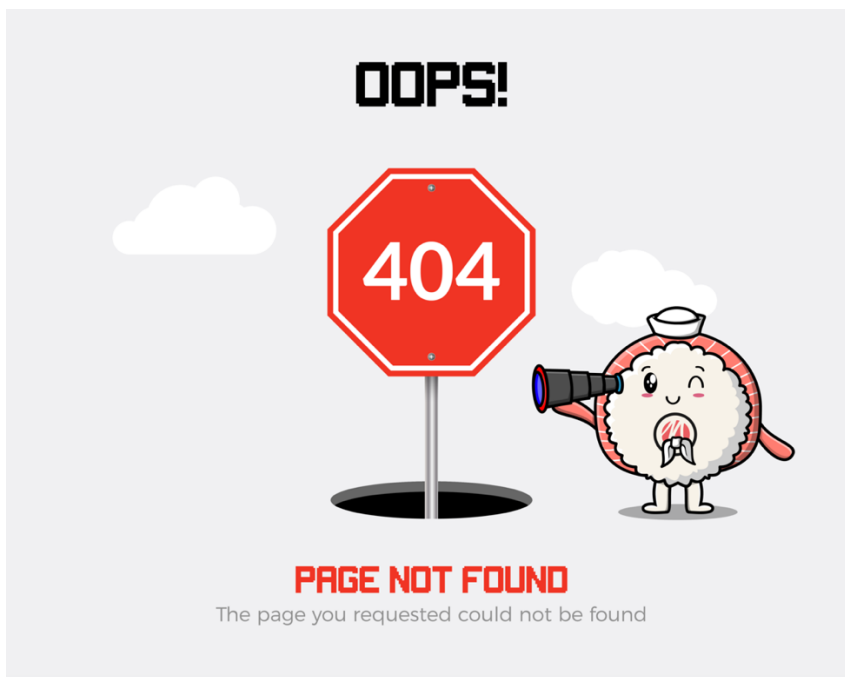


FIGURE 24. Miyabi Sushi website – 404-error page (Freepik company 2023)

### 5.5.9 Website rendering

Once all the pages have been built, the root component, here is `App.js`, is programmed to incorporate all the page components within the `app__root` div. Showing below in the code snippet in Code 6, the `react-router-dom` library is imported to provide the `BrowserRouter` component, which establishes the functionality of client-side routing to permit user to navigate between pages. The `BrowserRouter` component serves as the top-level component in the application, allowing developers to apply the `Route` component to define the different routes. This also maps routes to specific elements. The page paths are linked to their corresponding page components, which will be rendered when the selected path matches a page. Additionally, the 404-error page will be displayed if the requested page cannot be found, as indicated by the code snippet with `path=""`. In the `index.js` file, the necessary dependencies, including the `App.js` component and `createRoot` function, are imported into the `index.js` file to enable the invocation of the `render` method. Finally, the website is deployed and tested on the local development environment, which locally runs the web application on `http://localhost:3000`.

```

App.js
src > App.js > @ default
1 import React from 'react';
2 import { NavBar } from './components';
3 import './App.css';
4 // eslint-disable-next-line
5 import { BrowserRouter as Router, Routes, Route, BrowserRouter } from 'react-router-dom';
6 import { Home, About, Catering, Contact, Menu, Order, NoPage } from './pages'
7
8
9 const App = () => {
10   return(
11     <BrowserRouter>
12       <div className="app_root">
13         <NavBar />
14         <Routes>
15           <Route exact path="/" element={Home} /> />
16           <Route path="/about" element={About} /> />
17           <Route path="/contact" element={Contact} /> />
18           <Route path="/menu" element={Menu} /> />
19           <Route path="/catering" element={Catering} /> />
20           <Route path="/order-online" element={Order} /> />
21           <Route path="/" element={NoPage} /> />
22         </Routes>
23       </div>
24     </BrowserRouter>
25   )
26 }
27
28
29 export default App;

```

```

index.js M X
src > index.js
1 import React from 'react';
2 import { createRoot } from 'react-dom/client';
3 import './index.css';
4 import App from './App';
5
6 createRoot(document.getElementById('root')).render(< App />);

```

CODE 6. Root component - App.js and application render - index.js



## 6 MIYABI SUSHI WEBSITE FUTURE STEPS AND UPDATE

As introduced earlier, the Miyabi Sushi web project primarily focuses on developing the client-side of the web application. However, before the website can be published, there are a few additional steps that need to be taken, including setting up the backend server and web hosting. The web application has been successfully built and tested on the local development environment, which launched Miyabi Sushi web app with full functionalities and demo data file related to the online shopping service. To continue with the development, it is crucial to establish a server backend to handle these functions and facilitate data updates dynamically. The server now plays an important role in intermediating connectivity between the web application and other data sources. This entitles the application performing database read and write requests, as well as integration with Google APIs. (Mead 2018, 32-33.)

The process of constructing the server backend involves selecting a suitable backend programming language and utilizing its frameworks such as JavaScript with Node.js, PHP with Laravel, or Python with Django. The next step, once the server is programmed, is to choose a hosting provider that can support the web hosting on its cloud infrastructure. This helps to facilitate the connection between the web app and the server, ensuring its accessibility on the internet. There are several reputable web hosting providers available, each offering a range of features and subscription options, such as Bluehost, HostGator, and DreamHost. Additionally, these hosting providers often offer built-in database options to manage the flow of web data and enhance the overall functionality of the website. (Natural Intelligence Ltd 2023.)

In the current version of the Miyabi Sushi web application, customers can smoothly access information about the restaurant and place orders. In future updates, the user authentication will be a useful supplement. This introduces a valuable functionality that provides customers with private accounts for logging into the restaurant's website. This feature will enable the restaurant to create a loyalty program, effectively manage customer loyalty, and track customer engagement statistics. Besides, the user accounts contribute to a user-friendly experience by facilitating order management and the payment process. (Mutai 2023.)

Furthermore, the addition of online reservation service is a potential update for the web application. This factor will save time with booking tables in advance by providing relevant details such as date, time, and group size. Implementing this service will significantly enhance the convenience of customer

service by ensuring that the restaurant's booking system is always accessible. Lastly, the customer database, including information from registrations, orders, and bookings, will provide valuable data insights for the restaurant's business team. With the comprehensive view derived from customer figures, it enables the company team to generate essential strategies for growing business revenue. (The Square Editorial Team 2023.)

## 7 CONCLUSION

The aim of this thesis is to create a responsive web application for Miyabi Sushi restaurant located in Kokkola, Finland. This is primarily related to the front-end side, also known as client-side, with the support of ReactJS, Redux, JavaScript, HTML, and CSS. Through the web construction, most initial objectives were achieved by following the website requirements. The website was also tested successfully in the local development environment, which simulated the webpage, its features, and functionalities. The current website version, without backend server development, offers the restaurant a concise and modern appealing design with online order functionality on the client-side. Additionally, the website can become the main restaurant's marketing channel by offering products and service publicly. From the site, the customer can find all the restaurant's information from the company foundation to contact form, and online order offers. The web visitor can interact with different elements on the webpage, such as viewing product's detail or adding a product to the shopping cart.

In the development stage, Redux was initially applied to manage the application state and data, which promotes the scalability of the web app and simplified state updates. However, as the project size was assessed, it was determined that Redux was redundant. Subsequently, the React app was constructed without the use of Redux, employing a dynamic website by combining distinct components created independently. These components were implemented inside the root component with flexible manner to be ready for display rendering. This helped in generating the responsive website allowing to adapt to user's device and screen size, such as desktop and mobile screens. With the CSS and ReactJS support, the elements on a webpage were styled to be self-adjusting corresponding the user's current device view. This brings positive advantages to improve user experience as well as save development time and cost.

As the result, the Miyabi Sushi web project was successfully completed, following the requirements listed in chapter 2. The customer now can access all the information provided via the restaurant on the website, containing the gallery section showing the restaurant product and marketing images. Furthermore, the feedback form enables visitor to contact the restaurant customer service for sharing or requesting information. The cart functionality is also available to offer online ordering, providing great convenience to the customers and helping the restaurant team with order tracking and sales boosting. Besides, with the responsive design implemented, user can get access the webpage on different devices, avoiding interruption and error. In the local development environment, the web app was tested

indicating the smooth operation and being prepared to start the backend development. This thesis project was recorded in detail, consisting of theoretical concepts and the development process, facilitating future development or similar project development.

For further studies, the backend and database implementation should be the next development phase. The incorporation of these components supports the content and functionality management, allowing to store, retrieve, and manipulate data dynamically. The server-side will handle the request from client-side to perform data processing and other operation. Meanwhile, the database serves as the data storage of the web app containing all the content and supporting retrieval and manipulation, which works hand in hand with the backend program to deliver required functionality and user experience. Subsequently, the additional features, such as user authentication and table reservation, are possible to build on the implemented server.

## REFERENCES

- Babich, N. 2017. *Playing With Color: Vibrant Options For Apps And Websites*. Available: <https://www.smashingmagazine.com/2017/09/vibrant-colors-apps-websites/>. Accessed: 05 June 2023.
- Campanelli, D. 2019. *THE ART OF MARKET RESEARCH IN WEBSITE CREATION*. Available: [https://www.builderonline.com/building/operations/the-art-of-market-research-in-website-creation\\_o](https://www.builderonline.com/building/operations/the-art-of-market-research-in-website-creation_o). Accessed: 30 April 2023.
- Caspers, M. 2017. React and Redux. *Ausarbeitungen zum Seminar: Rich Internet Applications w/HTML and JavaScript*.
- Chaparro, B. & Bernard, M. 1999. Sitemap Design: Alphabetical or Categorical?. *Usability News* 1. 1.
- Dzhangarov, A. Pakhaev, K. & Potapova, N 2021. Modern web application development technologies. *IOP Conference Series: Materials Science and Engineering* 1155.
- Dordevic, B. 2023. Executing Constructive Market Research for Website Design. Available: <https://alphaefficiency.com/market-research-for-website-design>. Accessed: 07 June 2023.
- Fessenden, T. 2019. *Typography Terms Cheat Sheet*. Available: <https://www.nngroup.com/articles/typography-terms-ux/>. Accessed: 05 June 2023
- Freepik 2023. Available: <https://www.freepik.com>. Accessed: 11 June 2023.
- Foodora. 2023. *Miyabi Sushi*. Available: <https://www.foodora.fi/restaurant/npuz/miyabi-sushi>. Accessed: 07 June 2023.
- Gerkules, M. 2021. *How to Design a Hero Image: Best Practices and Examples*. Available: <https://shorturl.at/eiH12>. Accessed: 05 June 2023.
- Google. 2023. *Miyabi Sushi*. Available: <https://tinyurl.com/miyabisushigoogle>. Accessed: 07 June 2023
- Google Fonts. 2023. Available: <https://fonts.google.com>. Accessed: 04 April 2023.
- Juviler, J 2022. *What Is a Website Mockup? [+ 4 Steps to Make One]*. Available: <https://blog.hubspot.com/website/website-mockup>. Accessed: 05 June 2023.
- Lim, G. 2020. *Beginning React with Hooks*. 1<sup>st</sup> Edition. Publisher: Independently published.
- Liu, F. Xiao, B. Lim, E. & Tan, C. 2017. The art of appeal in electronic commerce: Understanding the impact of product and website quality on online purchases. *Internet Research* Vol. 27. 752-771.
- Laubheimer, P. 2021. *Local Navigation Is a Valuable Orientation and Wayfinding Aid*. Available: <https://www.nngroup.com/articles/local-navigation/>. Accessed: 05 June 2023.
- Larsen, J. 2020. *React Hooks in action*. Shelter Island: Manning Publications Co.

- Microsoft Corporation. 2023. *Download Visual Studio Code*. Available: <https://code.visualstudio.com>. Accessed: 05 June 2023.
- Marcotte, E. 2010. Responsive web design. *A list apart* 306.
- Mead, A. 2018. *Learning NodeJS Development*. Birmingham, UK: Packt Publishing Ltd.
- Mutai, J. 2023. *The Pros and Cons of Offering User Accounts on Your Website*. Available: <https://shorturl.at/cjGO7>. Accessed: 06 June 2023.
- Newman, M. & Landay, J. 2000. Sitemaps, Storyboards, and Specifications: A Sketch of Web Site Design Practice. *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*. 264-268.
- Natural Intelligence Ltd. 2023. *Best Web Hosting Services of 2023*. Available: <https://shorturl.at/fqLYZ>. Accessed: 06 June 2023.
- Pernice, K. 2013. *Carousel Usability: Designing an Effective UI for Websites with Content Overload*. Available: <https://www.nngroup.com/articles/designing-effective-carousels/>. Accessed: 05 June 2023.
- Palombi, N 2021. *What is graphic design? With examples for beginners*. Available: <https://web-flow.com/blog/what-is-graphic-design>. Accessed: 05 June 2023.
- Piyushpilaniya98. 2022. *CSS | Website layout*. Available: <https://www.geeksforgeeks.org/css-website-layout/#article-meta-div>. Accessed: 15 April 2023.
- Remix software. 2023. *Feature Overview*. Available: <https://reactrouter.com/en/main/start/overview>. Accessed: 06 June 2023.
- Robbins, J. 2018. *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics*. 5<sup>th</sup> Edition. Canada: O'Reilly Media, Inc.
- Salcescu, C. 2020. *Functional React: Quick start with React Hooks, Redux and MobX*. 2<sup>nd</sup> Edition. Publisher: Independently published.
- Satria, E. 2018. Content website of small and medium enterprises. *IOP Conference Series: Materials Science and Engineering* 434.
- The Square Editorial Team. 2023. *Top 8 Benefits of Online Booking Systems for Business*. Available: <https://squareup.com/au/en/townsquare/benefits-of-online-booking-systems>. Accessed: 06 June 2023.
- Thomas, M. 2018. *React in action*. 1<sup>st</sup> Edition. Manning Publications Co.
- Wilton, P. & McPeak, J. 2010. *Beginning JavaScript*. 4<sup>th</sup> Edition. Indianapolis: Wiley Publishing, Inc.
- Wieruch, R. 2017. *The Road to learn React*. Publisher: Independently published.
- Wieruch, R. 2022. *Web Development Setup for Beginner*. Available: <https://www.robinwieruch.de/developer-setup/>. Accessed: 06 June 2023.

Wix. 2023. Available: <https://www.wix.com/about/us>. Accessed 07 June 2023.