

## 1.3 Pass task

### Reflection:

This project involved creating a portfolio website using HTML, CSS, and Vue 3, with a focus on routing and dynamically loading content. Here's how I implemented various concepts to complete the project:

#### HTML and CSS:

I structured the website using HTML5, with sections for the header, navigation, main content, and footer.

CSS was used to style the layout, fonts, and colors of the website. External CSS files were linked to maintain separation of concerns and make the code more organized.

Images were incorporated to enhance the visual appeal and provide context to the content.

#### Vue 3:

I utilized Vue 3 to build a single-page application (SPA) for the portfolio. Vue's reactivity system made it easy to manage and update the content dynamically without the need to reload the entire page.

Vue Router was employed for client-side routing. I created routes for the home page and the task pages (1.1). Each route corresponds to a component that loads specific content.

Vue components were defined to encapsulate and manage different parts of the application, ensuring code modularity and reusability.

Fetching external HTML content was achieved using JavaScript's fetch API within Vue component methods. The loaded content was then rendered using Vue's v-html directive.

#### Routing:

Vue Router allowed me to create a seamless navigation experience. When users click on links in the navigation menu or elsewhere, the router updates the view without full-page reloads.

I defined routes for each task, and when a route is accessed, the corresponding component fetches and displays the relevant HTML content, ensuring efficient content loading.

#### Dynamic Content Loading:

To load content dynamically, I implemented async functions that fetch HTML content from separate files (1.1.html and 1.2.html). These functions are called when navigating to the respective routes.

The fetched content is then displayed within the Vue component using data binding, allowing for a smooth transition between tasks and the home page.

#### Footer and Styling:

The footer section was included to provide copyright information, enhancing the professionalism of the portfolio.

CSS was used to style components, create layouts, and ensure the website is visually appealing and responsive across various devices.

In summary, this project demonstrates the integration of HTML, CSS, and Vue 3 to create a responsive portfolio website with client-side routing and dynamic content loading. The use of Vue Router and Vue components contributes to code maintainability and a seamless user experience when navigating between different tasks and the home page. The project showcases the practical application of modern web development concepts for building interactive and engaging web applications.