Task 3

Digital copy of website:

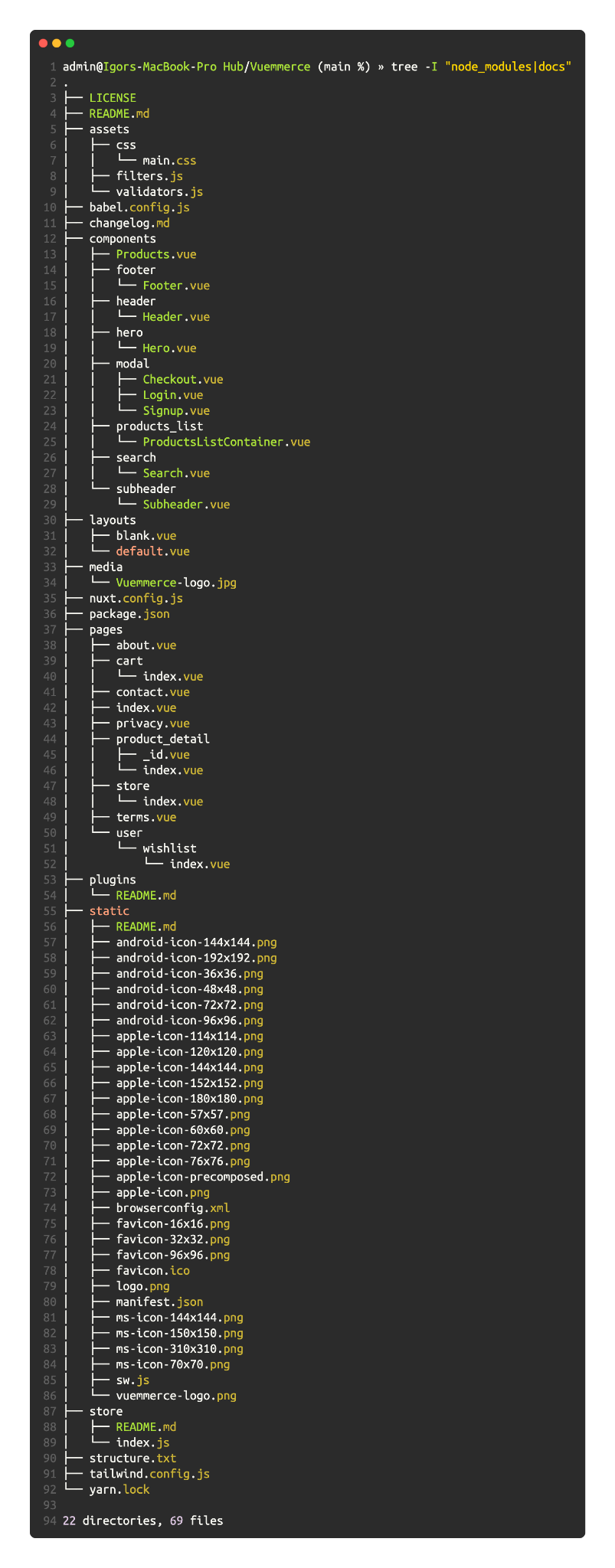
<https://mobileworldstation-website.vercel.app/>

Github with source code:

<https://github.com/denver-code/mobileworldstation_website>

Source Table:

| N | Name | Source | Description |
| --- | --- | --- | --- |
| 1 | logo.png | Selfmade | Logo of MWS company |
| 2 | galaxy-s23-ultra-highlights-design-end-s.jpg | <https://images.samsung.com/uk/smartphones/galaxy-s23-ultra/images/galaxy-s23-ultra-highlights-design-end-s.jpg> | Galaxy S23 Ultra Picture |
| 3 | Apple-iPhone-14-Pro-iPhone-14-Pro-Max-hero-220907\_Full-Bleed-Image.jpg | <https://www.apple.com/newsroom/images/product/iphone/standard/Apple-iPhone-14-Pro-iPhone-14-Pro-Max-hero-220907_Full-Bleed-Image.jpg.xlarge_2x.jpg> | iPhone 14 Pro Max Picture |
| 4 | Xiaomi-Redmi-Note-12-Pro-5G-Plain\_featured-image-packshot-review-1024x691.jpg | <https://cdn.dxomark.com/wp-content/uploads/medias/post-145957/Xiaomi-Redmi-Note-12-Pro-5G-Plain_featured-image-packshot-review-1024x691.jpg> | Redmi Note 12 Pro 5G Picture |
| 5 | TailwindCSS | <https://tailwindcss.com/> | CSS Framework |
| 6 | Vue.js | <https://vuejs.org/> | JavaScript Front-end framework |
| 7 | Nuxt.js | <https://nuxt.com/> | Vue.js Framework |
| 8 | VueX | <https://vuex.vuejs.org/> | Vue.js state management |
| 9 | VueX Store | [https://vuex.vuejs.org/](https://vuex.vuejs.org/guide/) | VueX based store backend |

Full File Structure:

This is a directory tree structure of a Vue.js e-commerce application.

It contains various directories and files related to the application's components, layouts, pages, store, plugins, and static assets.

The "assets" directory includes CSS files, filters, and validators. The "components" directory includes various reusable components of the application such as product list, header, footer, hero, and subheader. The "layouts" directory includes the default and blank layouts for the application.

The "pages" directory contains various pages of the application such as the home page, cart, contact, about, product detail, store, terms, and user wishlist. The "static" directory includes static assets such as icons, images, logos, and manifest files.

The "store" directory includes the Vuex store for the application. The "plugins" directory includes readme files related to the plugins used in the application.

Other files in the directory include the babel configuration, changelog, Nuxt.js configuration, package.json, tailwind configuration, and yarn.lock.

Pre-made code and Snippets & Configuration:

We used a VueX(Nuxt) Store built-in backend support folder for creating logic for internal routes and creating Business Logic Components and managing user requests like add to cart or remove from cart, register user, authorise user.

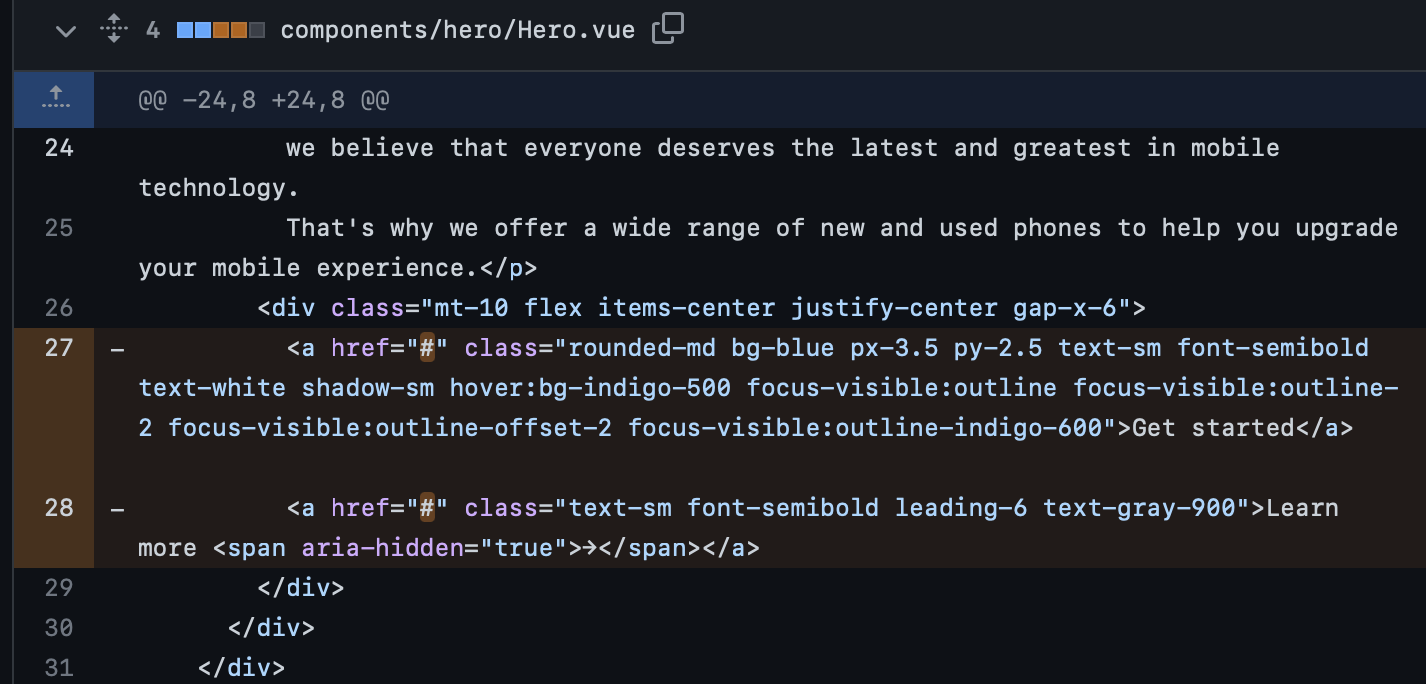
Full code of internal backend you can check here - <https://github.com/denver-code/mobileworldstation_website/blob/main/store/index.js>

Products of a store looks like that on backend:

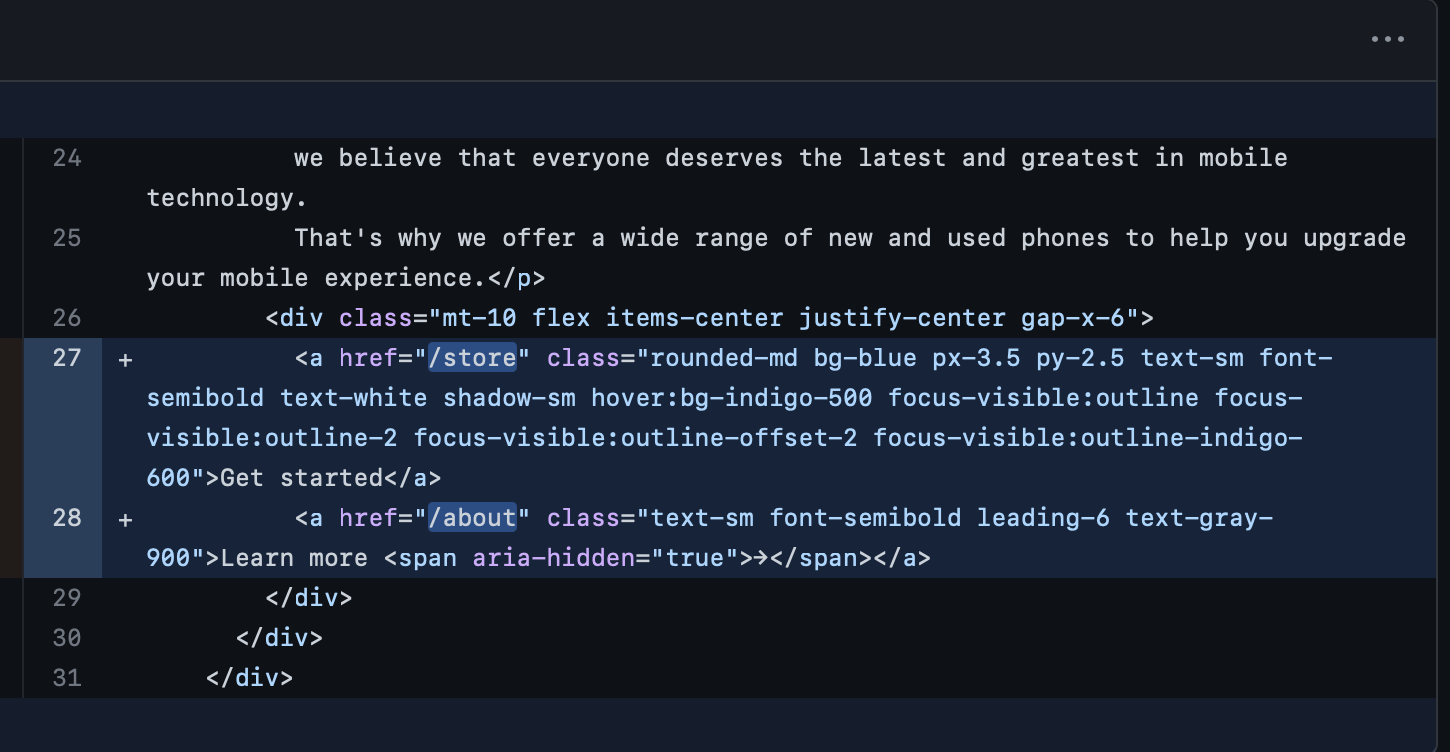
Test Plan Results:

| N | Action | Description | Expected result | Actual result | Fix |
| --- | --- | --- | --- | --- | --- |
| 1 | Click on Logo | Every page contain logo in navbar, so user can click to this logo | User should be redirected to the home page. | User redirected to home page. | - |
| 2 | Sign up button pressed | To process with cart you should have account | User account will be created and stored in local DB | User account has been created successfully. | - |
| 3 | Click on Get Started button | On the main page we have a “Get Started” button which should redirect users. | User should be redirected to the store page | User wasn’t redirected to store page | User was redirected to store page |
| 4 | Query entered to the search product field | Users may want to look up their own key-codes or phones with specific words in model name. | Only products related to the query will be displayed | Only products related to the query has been displayed | - |
| 5 | Add to cart button click | While users looking for phone - they may choose one and decided to buy this phone, but before they should add it to the cart | Product will be added to the cart | Product added to the cart | - |

Test N3 Fix:

Before:

After:



Commit with fix - <https://github.com/denver-code/mobileworldstation_website/commit/408d2d61dcba684c531c15faa886206c889f2083>

Interactive Features:

| Screenshot | Action |
| --- | --- |
|  | Redirect users to the index page |
|  | Redirect users to the store page |
|  | Redirect users to the contact page |
|  | Redirect users to the about page |
|  | Redirect users to the cart page |
|  | Redirect users to the store page |
|  | Redirect users to the about page |
|  | Redirect users to the feedback/contact page |
|  | Redirect users to the store page |
|  | Redirect users to the cart page |
|  | Redirect users to the privacy policy page |
|  | Redirect users to the terms and conditions page |
|  | Redirect users to the facebook page |
|  | Redirect users to the instagram page |
|  | Redirect users to the twitter page |
|  | Open an a menu with auth options |
|  | Open an a popup for registration |
|  | Open an a popup for a authorisation |
|  | Add item to a wishlist |
|  | Add item to a cart |
|  | Set Amount of items you going to buy |
|  | Remove item from cart |
|  | Search items by users query |
|  | Show popup to payment etc to finish payment and receive items |
|  | Remove item from cart |
|  | Redirect users to the Product details page |
|  | Grab user email and password and process information after click on blue button |
|  | Grab user first name, email, password, re-entry password to process information after click on blue button |
|  | Close popup |
|  | Grabs user first name, email and message, then on click - send all this data to support click |

Page Screenshots

| N | Page Screenshot | Details |
| --- | --- | --- |
| 1 |  | This page should contain items that users should see when they enter the website.  So that’s a Title and short slogan, short description who are we and what we doing, and intuitive buttons/links for fast navigation to main functionality of website |
| 2 |  | This page may contain products which users would like to buy or just check what we have. Also we need functionality for case if user decide to buy something, so we have stuff like Price, Title/description/rating, quantity select and “Add to Cart” button |
| 3 |  | This page should contain all information and tools how user can contact us such a Form for feedback or reporting problem and info like address, email and phone |
| 4 |  | This page should contain all information about the company, who we are and what we are doing. |
| 5 |  | This page should contain products which the user added to the cart and wanted to buy.  So we have a retrieve all stuff in user’s cart and display Buy button to make users able to buy their stuff |
| 6 |  | Privacy Policy |
| 7 |  | Terms & Conditions |
| 8 |  | This page must contain product details like description, title, price, rating, picture for case when user wants to get to know better about a specific product. Also for some cases it’s a Wish-list button if form of a heart, and “Add to Cart” button |

Feedback:

**Feedback 1:**

Regarding the test plan, it is good to see that you are already testing the website before deployment. However, the test plan seems to be lacking in detail. It would be helpful if you could provide more information on the test environment, the steps to reproduce the issue, and the expected result. Additionally, it would be beneficial to have a list of all the pages on the website that should be tested, as well as any potential edge cases that should be considered.

**Suggestion:** To improve the test plan, you could start by creating a list of all the pages on the website, and then create a test case for each page. For each test case, you could include the test environment, the steps to reproduce the issue, and the expected result. It would also be useful to include any edge cases that should be tested. Once you have created the test cases, you could assign them to a tester or testing team and track the progress of each test case using a test management tool.

**Response:** Thank you for your feedback. I appreciate your suggestions on how to improve the test plan. I will create a detailed list of all the pages on the website and create a test case for each page. For each test case, I will include the test environment, the steps to reproduce the issue, and the expected result. Additionally, I will include any edge cases that should be tested. I will assign the test cases to a tester or testing team and track the progress of each test case using a test management tool.

**Feedback 2:**

The user requirements you have listed are comprehensive and seem to cover all the necessary aspects of the website. However, it would be helpful to have a prioritisation of these requirements, as it may not be possible to implement all of them within the given timeline and budget. Additionally, it would be beneficial to conduct user research to understand the target audience's preferences and pain points, which could inform the design and development process.

**Suggestion**: To prioritise the user requirements, you could create a matrix that evaluates each requirement based on its importance and feasibility. This matrix could help you identify the most critical requirements that should be implemented first. Additionally, conducting user research through surveys, focus groups, or interviews could help you understand the target audience's preferences and pain points, which could inform the design and development process.

**Response**: Thank you for your feedback. I agree that prioritising the user requirements is essential to ensure that we deliver a high-quality website within the given timeline and budget. I will create a matrix that evaluates each requirement based on its importance and feasibility, which will help us identify the most critical requirements that should be implemented first. Additionally, I will conduct user research through surveys, focus groups, or interviews to understand the target audience's preferences and pain points, which will inform the design and development process.

Brief and more Feedbacks Summary

Feedback 1:

Testers reported that the login page took too long to load, which caused frustration and a poor user experience. 5 testers participated in this test. We accepted this feedback and addressed the issue by optimising the login page by reducing the amount of assets and changing layout.

Feedback 2:

The client requested that we add a chatbot feature to the website to improve customer support. However, due to budget constraints and limited development resources, we rejected this feedback. We explained the reasons to the client and proposed alternative solutions that would still provide efficient customer support.

Feedback 3:

Testers reported that the checkout process was confusing and unclear, causing them to abandon their shopping carts. 10 testers participated in this test. We accepted this feedback and re-designed the checkout process to make it more straightforward and intuitive.