

System Description

The Online Gallery responsive web app is an app designed to provide a space for artists to hold online exhibitions and for the general public to be able to see these artworks.

More details on the Online Gallery in the project proposal.

The system was built using HTML, CSS and JavaScript. It includes responsive features such as scaling with screen size, hover animations, and feature change depending on screen size.

Challenges

Vue CLI

When attempting to use Vue CLI, some instructions are unclear which caused some problems in setting up Vue. In addition to that, a majority of the project is already made without Vue CLI, and moving the project from the current files to Vue will be too complicated.

Deviations from Proposal

Features Changed

1. The website design has been changed into a slightly cooler monochromatic tone with a new logo.

Features Removed / Pushed to Next Phase

1. Search Bar – This feature is too complex to complete within the time frame.
2. Registration/Log In Database – Although the registration and Log In functions are complete (the website being able to check), it is not yet connected to a registration database and isn't fully functional without an example account
3. User Profile – User profile is also pushed to the next phase
4. Hold Exhibition & Upload Images – Except for one sample Exhibition, both Hold Exhibition and Upload Images will be pushed to next phase as the User Profile is not yet available.
5. Rating – This feature is linked to the user profile and will be pushed to the next phase along with the User Profile.

Coding Concepts

Scaling

This was used in all pages.

The app not only scales to the size of the screen but also changes some features according to the screen size. There are 3 screen sizes available namely PC, Tablet and Mobile Phone. This is especially evident in the homepage as well as the exhibition sample page.

The screen modifications are available at the end of the css file.

Dynamic Log In (Form Input Bindings)

Making use of CSS to make a sleek log in and register page, as well as allowing users to register and log in using three social medias: Facebook, Twitter, and Google. Users are also able to switch between both pages. Even if logging in is not yet an option.

The feature is available at 'Online Gallery Log In.html', 'Online Gallery Register.html', and the css file under form.

Hover Animations (Transition)

Almost all buttons have hover animation, and clickable texts also have hover animations. Although it's accessible in all screen sizes, but hover is only noticeable on PC. It makes use of css' animations such as transition, resize, and hover color.

The feature is available in css under button and links.

Hamburger Menu (Event Handling)

Using the Hamburger Menu, the navigation bar is moved from the top of the screen on PC to hidden on tab or mobile and accessible through the hamburger menu. It makes use of js to add 'active' to a class and css in order to animate.

The feature is available in the js file, and in the css file under hamburger-menu.

Vue Concepts

I implemented Vue without Vue CLI, so some things are not available for me like the SFC and other Vue CLI related things. However, I still managed to use some of Vue's features in my website using the Vue link, javascript and CSS.

Documentations and Basics

- Form Input Binding is used for the Registration and Log In page
- Conditional Rendering is used for a 'Hello User' message in the Log In page
- Event Handling is used for the Registration, Log In, and Transition
- Component Basics are used for Exhibition Description

Components in Depth

- Component Registration is used in order to use the component in the Exhibition page
- Props are used in order to make the message changeable, however without the Hold Exhibition function, the message is still static and only editable through the code
- Custom Events are used for Log in, Registration and Transition, especially the login function which checks the inputted username with the stored username

Transitions and Animations, Reusability & Composition, Tooling and Scaling Up

- Enter / Leave & List Transition is used in order to hide or reveal the Exhibition Description
- State Transition is used in order to make the rating function feel interactive (number animation)
- Unit Testing is used in order to test if each function works, using the testing database is a little bit harder than just run, see the problems, and debug for me though

Reflection

Vue has a lot of useful features, even if I don't end up using the Vue CLI and misses some of the more useful features like the Single File Component. However, even without the Vue CLI, the Vue concepts are still very useful. Though if I have the time to remake everything within Vue CLI it should be easier, and I should reach out for help more. Some concepts are just not applicable without the Vue CLI but I tried implementing as many concepts as I could, which makes the website interactive.