

Mock up 1

JavaScript



Michal Krawczyk 12/10/2023

# Screenshots

Showcase

A black and white website with a black and white image

Description automatically generated

Dark/Light Mode

A screenshot of a website

Description automatically generated

Show more/less text

A screenshot of a website

Description automatically generated

**JavaScript Functionality for Mockup 1**

In the design of our mockup, we incorporated three key functionalities using JavaScript:

* Image Showcase
* Dark/Light Mode Toggle
* Expandable Text (Read More/Less)

My JavaScript structure follows a modular approach, where the main app.js file orchestrates specific functionalities from individual module files.

**Image Showcase:**

Positioned at the top of the webpage, I implemented a gallery feature that consists of thumbnails and a primary display image.

Implementation:

I initialized our functionality using the initShowcase() function.

Using this function, I target the main elements of the gallery: the display image, the associated caption text, and the set of thumbnails.

For interactivity, an event listener is attached to each thumbnail. Upon clicking a thumbnail, it:

Updates the source of the primary display image by manipulating the thumbnail's source to match the larger image's filepath.

On desktop versions, the associated text from the thumbnail is made visible beneath the main display image. This text remains hidden on smaller device sizes using CSS for a cleaner user experience.

**Dark/Light Mode Toggle:**

In the layout, I integrated a Dark/Light mode toggle feature to enhance the user experience by allowing users to choose their preferred viewing mode.

Implementation:

The function setMode() was developed to handle the theme switching.

Two key elements are targeted in this feature: the sun icon (representing the light mode) and the moon icon (representing the dark mode).

The primary method of toggling between modes is by manipulating the class list of the root HTML element.

When the sun icon is clicked:

The class light-mode is added to the root element using the method classList.add(). This shifts the webpage to its light theme version.

Conversely, when the moon icon is clicked:

The class light-mode is removed from the root element using classList.remove(), reverting the page back to its dark theme.

This functionality ensures users have the flexibility to adapt the website's appearance according to their preference or ambient lighting conditions.

**Read More / Less Functionality**

For the Read More/Less functionality, the following approach was taken:

Collect Necessary Elements:

All the text toggling links, denoted by the class show-text, are selected and stored in the showHideTextLinks NodeList.

An array fullTexts is initialized to store the full, un-truncated text of each associated text element.

Iterate Through Each Toggle Link:

For every link in showHideTextLinks, a function is invoked for each item. This function takes in both the link itself and its index in the NodeList.

For each link, the preceding sibling (which is the text to be toggled) is selected and stored in textElement.

The complete text of textElement is then stored in the fullTexts array at a position corresponding to the link's index.

Truncate Initial Text (if necessary):

If the length of the text from textElement exceeds 100 characters, it's truncated to the first 100 characters followed by an ellipsis (...).

This ensures that, by default, only a shortened version of the text is shown.

Attach Event Listener for Toggling:

A click event listener is added to each showHideTextLink.

On clicking a link:

If the currently displayed text ends with an ellipsis (...), indicating it's in its truncated state, the full text (from the fullTexts array) is shown, and the link's text is updated to "Show less".

Otherwise, the text is truncated again, and the link's text changes back to "Click here for more informations".

Text Truncation Logic:

The truncateText function is a utility function that receives a text and returns its first 100 characters followed by an ellipsis (...).