PUI Final Project Write-up

Xinzhu Wang
xinzhuwa@andrew.cmu.edu
Dec 9 2022
Website link: https://dandylion99.github.io/xinzhu-pui-assignment/final-project/index.html

Part 1: Website Overview

My website "WESTERN ART MOVEMENTS: 19th & 20th Centuries" is an online art gallery, and aims to introduce major western art movements in the 19th-20th centuries by displaying famous paintings. The information conveyed includes the features, historical context, and time period of each movement, and full images, introduction, artist information of selected famous paintings. I try to make the website interesting and engaging by displaying movements and paintings in a timeline style, using color coding for different movements, and creating elegant animations. The target audience is the general audience who have not much prior knowledge about art history but have an interest and curiosity for art movements.

Part 2: User Interaction

Screen sizes to test my interface

Laptop view: 1280*832 px
Mobile view: iPhone 12 pro, 390*844 px

Interactions

- Hover on an art movement bar with color, see the background changing (contents for gray movement bars haven't been developed)
- Click on this art movement bar, go to the movement page
- Hover on a painting, see it's clickable
- Click on this painting, go to the painting page, read related information
- Click on the arrow on the left, go back to the movement page
- (users can click on other paintings and repeat the above two steps)
- Zoom out using touchpad, or scrolling up using the mouse, or click on the "home" button in the breadcrumb, and then go back to the homepage
- Hover on another art movement bar, see the background changing
- Click on this apartment bar, go to the movement page
- (following interactions are similar to the steps above, namely click into paintings and learn information)

Part 3: External Tools Used

jQuery

Why I chose to use it

I chose to use jQuery because jQuery could simplify codes to complete tasks such as selecting elements, adding animations, adding event listeners, and executing events in sequence (executing the next event after the previous one has finished)

How I used it

I used it in tricky tasks required for the website behaviors, such as cross page animations, and looping through all elements with a specific.

What does it add to my website

jQuery contributed to the simplicity and elegance of my codes. In addition, some of the animation codes are borrowed from online resources, which are written in jQuery codes, so suing jQuery made it easier for me to learn from others.

(1) **Part 4:** Describe how you iterated on your prototypes, if at all, including any changes you made to your original design while you were implementing your website. (4-8 sentences max)

Part 4: Iterations on Prototypes

In my original design, users could zoom in or zoom out using the touchpad to transit between the homepage and the specific movement page, with the zooming animation of the timeline. However, I found the zooming interaction would conflict with the horizontal scrolling interaction when users are viewing the website in a narrower screen, so I decided to abandon the zooming in the homepage to movement page, and only keep the feature of zooming out from the movement page to the homepage.
I devoted most time to polishing visual styles and creating animations, and thus only created contents for two movements with 8 paintings. After asking my TA Venkat, I decided to only display the two completed movements on my website, and gray out other movements on the homepage so that users could understand what's clickable.

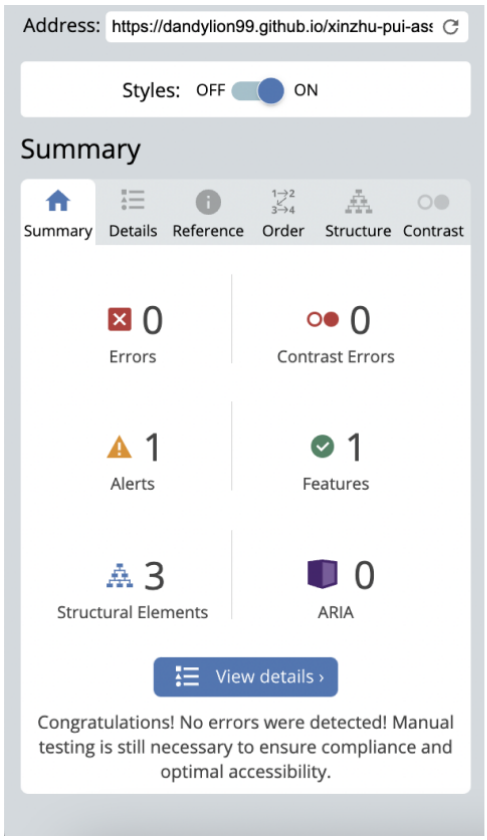
(2) **Part 5:** What challenges did you experience in implementing your website? (2-4 sentences max)

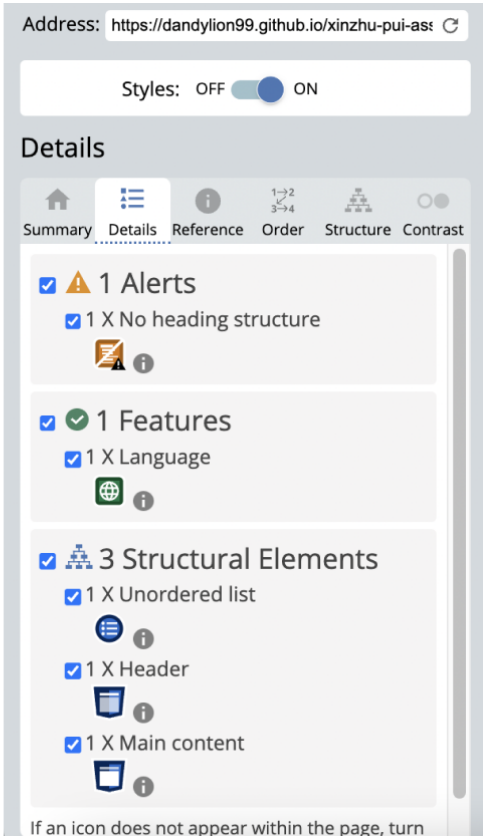
Part 5: Challenges during Implementation

One major challenge is creating the timeline style layout on homepage and the movement page, where each movement or painting box has a customized position and could overlap with the timeline box. I solved this issue by using the “position: absolute” css properties and adding specific position values to each box.
Another big challenge is the animation, especially the cross page animation, which is usually achieved using responsive programming frameworks. However, when i started to create animations, the overall structure of my website is coded in normal html/css/js structure. Considering the time limit and simplicity, I decided to “fake” the cross page animation using css and js, and finally created nearly seamless page transition animations by adding the transition box using JavaScript.

Appendix: Screenshots from WAVE

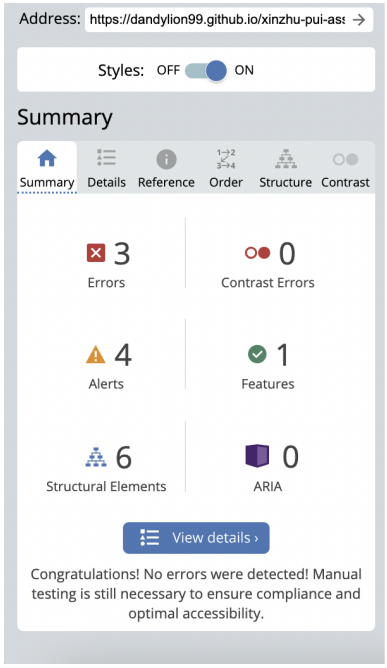
“Home” page

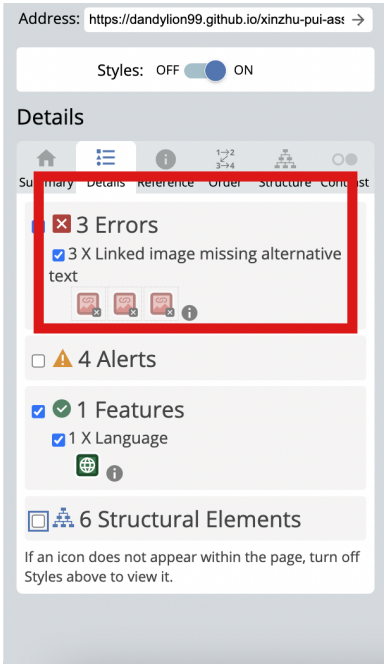




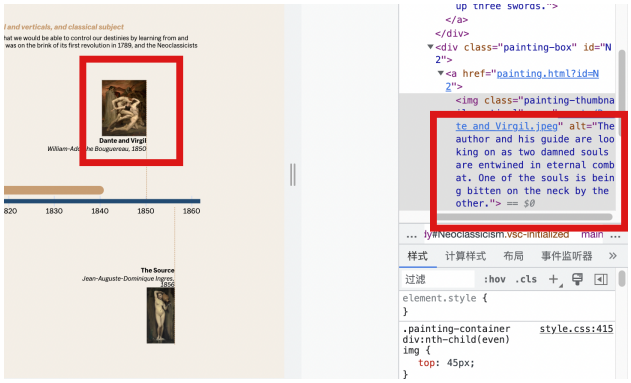
“Movement” page

Note: WAVE shows 3 errors because it does not detect the alt texts I added by JavaScript.

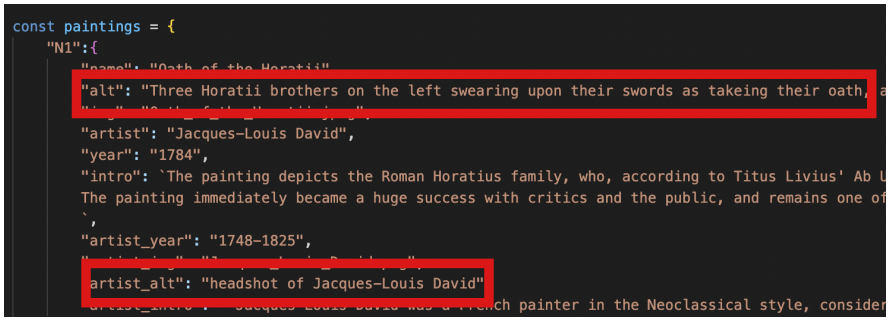




From the screenshots below, you can see that there actually are alt texts for painting images. I stored all alt texts for all images in the paintings_info.js file, and added them in html pages using JavaScript functions.



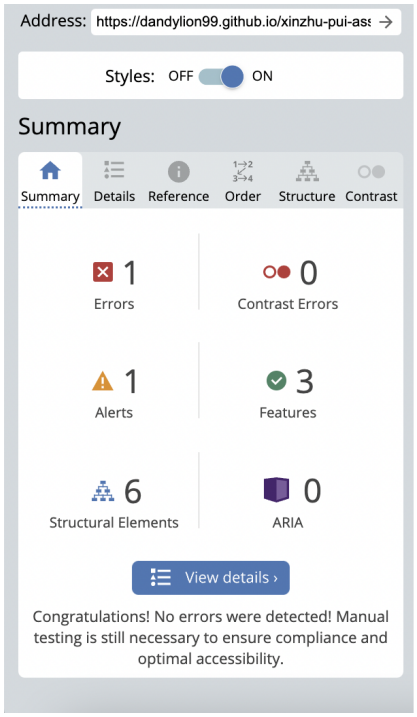
```
window.onload = () => {
    ID = $('body').attr('id');

    // set theme color
    $(':root').css('--theme-color', movements[ID]["theme_color"]);
    $(':root').css('--main-bg-color', movements[ID]["main_bg_color"]);

    // add alt text for each painting img
    $('.painting-box').each(function(i, obj) {
        ID = obj.id
        console.log($("#"+ID+" img"));
        $("#"+ID+" img").attr('alt', paintings[ID]['alt']);
    });
```

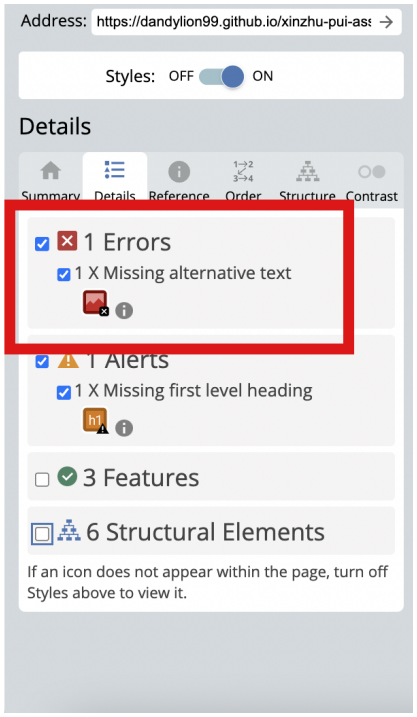


```
const paintings = {
    "N1":{
        "alt": "Three Horatii brothers on the left swearing upon their swords as takeing their oath,
        "artist": "Jacques-Louis David",
        "year": "1784",
        "intro": `The painting depicts the Roman Horatius family, who, according to Titus Livius' Ab U
        The painting immediately became a huge success with critics and the public, and remains one of
        `,
        "artist_year": "1748-1825",
        "artist_alt": "headshot of Jacques-Louis David"
```

“Painting” page

Note: Similarly, WAVE shows 1 errors because it does not detect the alt texts I added by JavaScript.





From the screenshots below, you can see that there actually are alt texts for painting images. I stored all alt texts for all images in the paintings_info.js file, and added them in html pages using JavaScript functions.


```
// update the painting page based on id
function updatePainting(paintingID,currentPainting){
    let painting = currentPainting;
    let imgElement = document.querySelector(".painting-full");
    imgElement.src = "assets/"+painting['img'];
    imgElement.alt = painting['alt'];
```