# CPSC 332 Web Development

Gonzaga University

Daniel Olivares

## Homework 5 – JavaScript – Interactive Art Gallery with JavaScript

Individual, non-collaborative assignment

### Learner Objectives:

By the end of this assignment, you will be able to:

- **Interact with HTML elements using JavaScript**: Respond to user clicks and dynamically update the webpage.
- **Manipulate HTML and CSS using JavaScript**: Modify element styles and content through event handlers.
- **Dynamically add content**: Use JavaScript to add new elements (art panels) to the webpage.
- **Use external JavaScript and CSS files**: Move inline JavaScript and CSS into external files for better organization and maintainability.

### Prerequisites:

Before starting this assignment, you should:

- Have completed up to and including **Chapter 7** in the class zyBook.
- Be able to create a basic HTML web page using **HTML** and **CSS**.
- Have a basic understanding of **JavaScript** and **event handling**.

### Overview and Requirements:

You will create an **Interactive Art Gallery** webpage where users can "view" famous artworks by clicking on panels. As users view artworks, the background color of the clicked panels will change, and a counter will display how many artworks have been viewed. Additionally, users can add new artworks dynamically, and the gallery can be reset to its original state. You will be provided starter code with the base HTML and CSS. Your task is to add the required JavaScript code to complete the required functionality.

**Functional Requirements:**

1. **Art Panel Click Event**:
    - Clicking on an art panel changes its background to indicate it has been viewed.
    - The number of viewed artworks is tracked and displayed in a counter.
2. **Reset Button**:
    - Adds a button that resets the gallery to its original state and resetting the view counter. Removing dynamically added artworks is not required.
3. **Add Artwork Button**:
    - Dynamically adds new art panels with randomly selected artworks from a predefined list.

4. **Refactor to External Files**:
    - Move all **JavaScript** and **CSS** into external files once the functionality works.
    - You are required to include your script reference in the head section of your HTML document.
    - Hint: refer to **section 7.1** of the zyBook if you encounter issues with the JavaScript not running as expected after moving it externally.

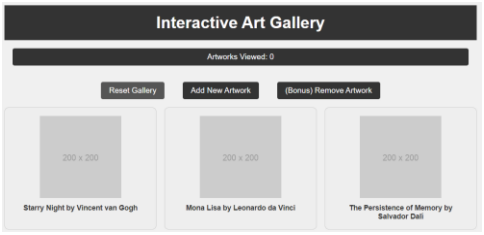
---

**Examples:**

Below are examples of what your **Interactive Art Gallery** should look like at different stages of interaction:

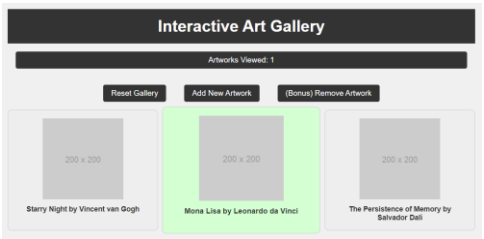
**Initial View (Before Interaction):**

- Three art panels are displayed, and the counter shows "Artworks Viewed: 0."



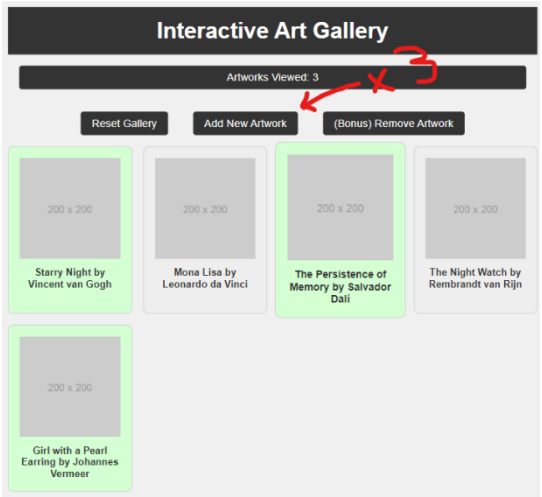

**After Clicking Panels:**

- After clicking on some panels, their background color changes, and the view counter updates accordingly.



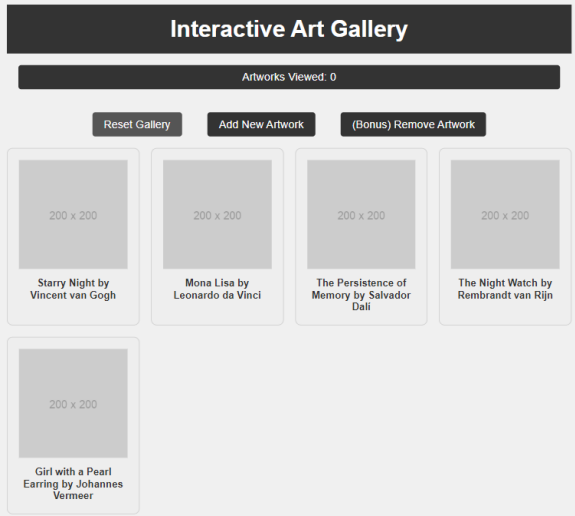

**After Adding New Artwork:**

- New artworks are dynamically added to the gallery by clicking the "Add New Artwork" button. Note the layout should automatically adjust based on viewport width (this should be in the starter code).



**After Resetting the Gallery:**

- The gallery returns to its original state and the view counter reset. Note the dynamically added artwork was not removed from the gallery.




## Bonus (5 points)

Add an additional button titled “Remove Artwork” that toggles the functionality of clicking an artwork from highlighting it to removing it. That is, rather than highlighting the clicked artwork, it will remove it from the page.

Clicking this button should also modify the style of the button to indicate it is “active” and inform the user that any clicks will remove the clicked artwork. Clicking the button again should revert click functionality back to highlighting any clicked artwork.

**Note: You are required to add a Canvas comment stating you have completed this task to get credit.**


## Submitting Assignment:

Submitting your assignment requires submitting in two steps:

1. Submit your assignment to the appropriate GitHub Classroom repository (commit to the repository!). You’ll find the invitation URL as part of this assignment on the Canvas assignment page.

2. *Submit two URLs*

    i. Submit your GitHub Classroom repository URL to Canvas (Line 1).

    ii. *Submit your GitHub Pages URL to Canvas (Line 2)*

        i. *Enable Pages via the settings in the GitHub Classroom repository created as part of this assignment.*

## Grading Guidelines:

You will be graded based on the following criteria:

1. **Basic JavaScript Interaction** (20 points)
    - 10 points: Correctly implements event handling for clicking on art panels.
    - 5 points: Updates the view counter correctly based on panel clicks.
    - 5 points: Visual feedback (background color change) when an artwork is viewed.
2. **Reset and Add Artwork Functionality** (20 points)
    - 10 points: Reset button restores the gallery to its original state (removing dynamically added panels is not necessary but you must reset their viewed state).
    - 10 points: Add New Artwork button correctly adds new art panels dynamically.
3. **CSS Styling and Design** (10 points)
    - 5 points: Good use of CSS for styling the art panels, buttons, and page layout.
    - 5 points: CSS properly linked in an external file.
4. **Refactoring to External Files** (20 points)
    - 10 points: JavaScript code successfully moved to an external .js file.
    - 10 points: The external JavaScript file functions correctly as an external file.
5. **HTML Validation** (5 points)
    - 5 points: HTML code must validate without errors using the W3C HTML Validation Service.
6. **GitHub Classroom Repository Commit History** (10 points)
    - 5 points: Regular, meaningful commits made throughout the development process (minimum of 3 commits).
7. **Working GitHub Pages URL** (5 points)
    - 5 points: Submission includes a working GitHub Pages URL that correctly displays the interactive webpage. URL is generated using the correct repository (the assignment repository).
8. **Completion of Requirements** (10 points)
    - 5 points: All specified features are implemented and functional (counter, reset, add artwork, external file refactoring).

**Total: 100 points**