

GIT 418 Final Project

Marti Rosenbeck

Project Requirements

For this project, I was asked to create a single-page website that included a jQuery widget, Ajax/API integration, web storage functionality, and a slideshow/carousel, showcasing the JavaScript I learned in this course.

General Requirements

- Single Page Site:
 - Display properly at 1280px wide.
 - Represent a real, stand-alone website with professional content (no placeholders).
 - Include standard website elements like header, navigation, and footer.
- File Organization and Validation:
 - Well-organized, properly indented, and well-commented files.
 - Valid HTML and CSS without errors or warnings.
 - No inline JavaScript or CSS in HTML files.
 - No visible errors in the browser console.
 - Use `async` or `defer` attribute for script tags.
- Code Integrity:
 - All code must be original, written from scratch.
 - Implement "use strict"; globally in JS.
 - Do not upload files to public online locations without obfuscating JS.

Required Elements

- jQuery Widget/Plugin:
 - Utilize at least one jQuery widget/plugin (e.g., tabs, accordion).
- Ajax/API:
 - Load content from a third-party API or a JSON file.
 - Display at least three pieces of relevant content (e.g., movies, portfolio projects).
- Web Storage:
 - Implement web storage to store/display user info or content.
- Slideshow/Carousel:
 - Include a slideshow/carousel with at least three images using jQuery or another JavaScript library.

Inspiration

The requirements inspired me to create these JavaScript features:

- Accordion FAQ:
 - Displays frequently asked questions about hours, location, and contact details using a jQuery accordion widget/plugin.
- Dinner and Movie Pairing:
 - Integrated a movie API to suggest a movie to pair with a meal.
 - Included a button for users to order their meal online.
- Newsletter Signup:
 - Used web storage to inform users if they've already subscribed to the newsletter.
- Specials Carousel:
 - Displayed images for Thursday and Sunday specials in a slideshow/carousel format.

Goals/Purpose/Users of the Site

Goals:

- Enhance user experience with interactive features.
- Provide essential information like hours, location, and contact details.
- Engage users with movie and meal pairing suggestions.
- Offer newsletter subscription and inform users if they have already subscribed.

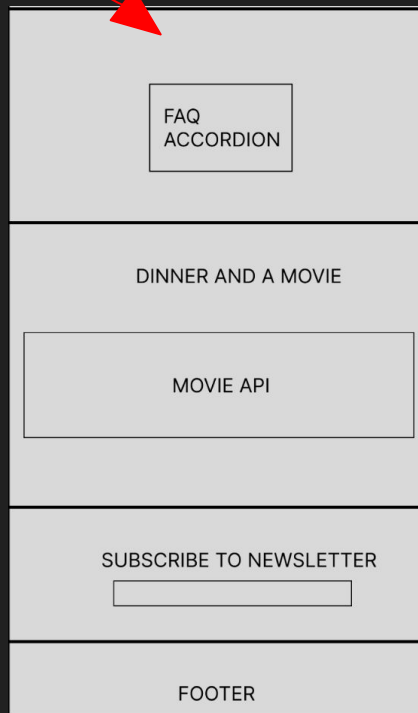
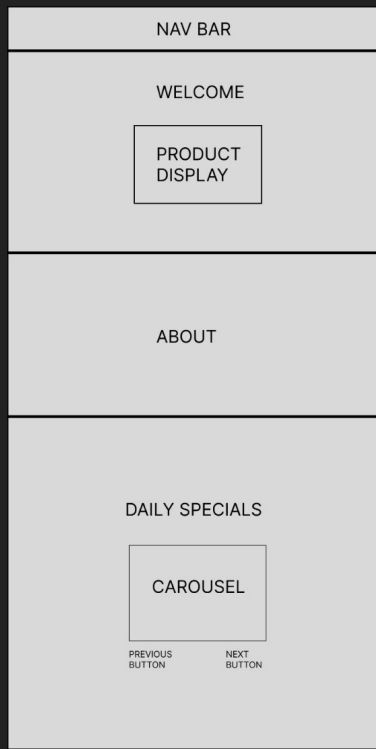
Purpose:

- To provide an informative and interactive website for users.
- To promote the restaurant's offerings and engage with the audience.

Users:

- Local residents and tourists looking for dining options.
- Movie enthusiasts interested in meal and movie pairing.
- Regular customers interested in promotions and updates.

Process: Discovery/Goals/Content Gathering



The process began by examining the single page site I had built for GIT 417 and thinking about how and where I would add the new JavaScript elements. I created the wireframe on the left to finalize the layout of the site.

The main goal of the new website was to add interactivity to the site to make it more engaging and useful for the user.

The Daily Specials content had originally been a 2x2 photo grid and FAQ was a link to a contact page. This content was adapted into JavaScript features such as a photo carousel and an accordion.

Process: Coding

I developed the website using a combination of HTML for structure, CSS for styling, and JavaScript along with jQuery for dynamic functionalities.

I used `defer` in the `

Process: Coding

// Accordion

To create the accordion section, I utilized jQuery's accordion widget.

In the JavaScript file, I used jQuery to select the `#accordion` element from the HTML. Then, I initialized it as an accordion using the `.accordion()` method provided by jQuery UI.

In the CSS, I styled the `#accordion` to have padding, a background color, and to be centered both vertically and horizontally. The text inside the accordion is aligned to the left.

Each question in the accordion is represented by an `<h3>` heading with accompanying `<div>` for the answer. The CSS styles the headings with specific padding and color, and the paragraphs with bold font-weight and padding.

What are your hours?

Wednesday & Thursday | 4 PM - 8 PM

Friday & Saturday | 4 PM - 9 PM

Sunday | 11 AM - 7 PM

Mondays & Tuesdays | Closed

Where are you located?

How can I make a reservation?

Process: Coding

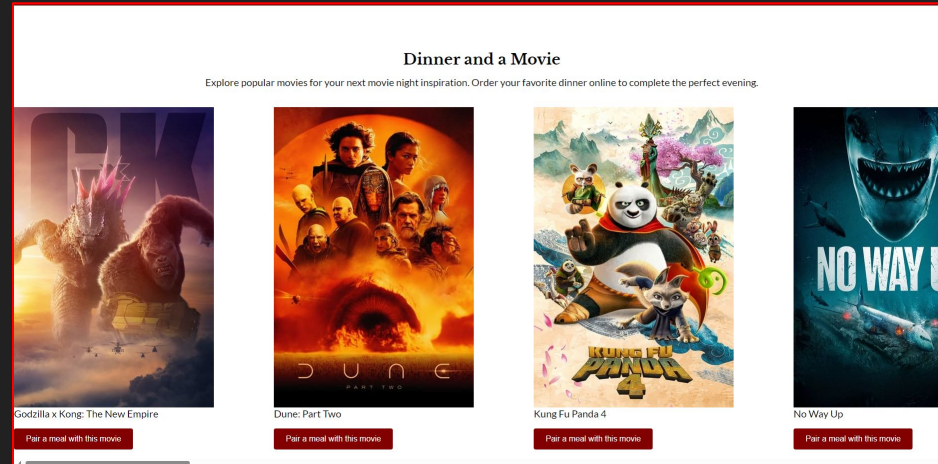
```
// Movies API
```

To create the Movies API section, I used the TMDb API to fetch popular movies and displayed them on the website.

In the JavaScript, I defined a function `loadPopularMovies()` that constructs the API URL and makes an AJAX request to fetch movie data.

In the CSS, I styled the `#dinnerAndMovie` section with centered text and specific padding. I also set up a horizontal scrolling layout for the movie containers using `.movie-container`. I applied styling to the movie poster images to ensure they are responsive and styled the order online button to match the restaurant's color scheme.

Each movie section consists of an image, movie title, and an order online button. The styling ensures the images are responsive, while the titles have padding. The order online button is styled to match the restaurant's color scheme.



Process: Coding

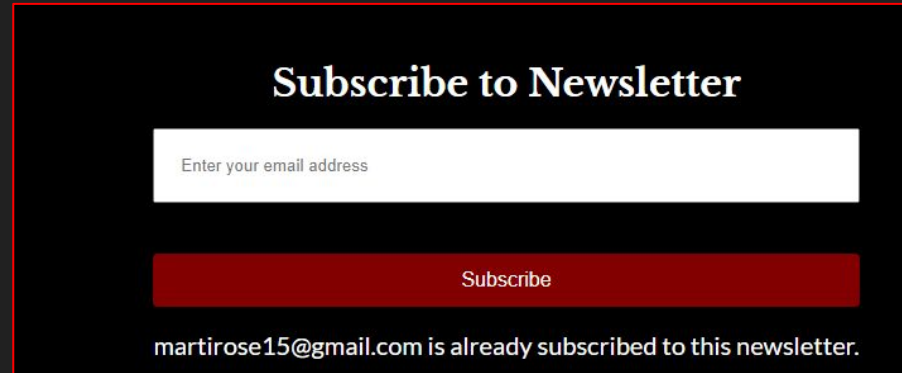
// Web Storage

In the Web Storage section, I utilized local storage to inform users if they had already subscribed to the newsletter.

In the JavaScript, I set up a function `storeEmail(e)` to handle the newsletter signup form. This function first prevents the default form submission and then validates the email format. It checks if the email matches any stored email in `localStorage`. If the email is already stored, a message informs the user they are already subscribed. Otherwise, the email is stored in `localStorage`, and a welcome message is displayed. The email input is then cleared. An event listener on the submit button triggers the `storeEmail` function, and `window.onload` checks for any existing stored email to display a subscription status message.

In the CSS, I styled the `#newsletter` section with a dark background and added padding for visual appeal. The form is displayed using a grid layout with centered content. Input fields and messages are styled with specific padding and color to maintain the design consistency.

The newsletter section includes an email input field, a submit button, and message placeholders. The styling ensures a visually appealing and user-friendly form, while the JavaScript uses local storage to provide feedback on subscription status.



Subscribe to Newsletter

Enter your email address

Subscribe

martirose15@gmail.com is already subscribed to this newsletter.

Process: Coding

// Image Carousel

To create the image carousel section, I used JavaScript to handle the slide transitions and jQuery for DOM manipulation.

In the JavaScript code, I first set up variables to keep track of the current slide and the total number of slides. Initially, all the images in the carousel are hidden except for the first one. For the next and previous button click events, I used jQuery to hide the current image, update the current slide index, and then show the new image. The index is looped back to the first or last image if the user tries to go past the end or beginning.

In the CSS, I utilized CSS Grid and Flexbox to center-align and layout the images and buttons. The images have a maximum width of 100% to ensure they are responsive, and they have a subtle box shadow for a bit of depth.

The HTML structure includes a container div for the carousel images, with each image wrapped in its own container div. Below the image carousel, there are next and previous buttons for navigation.



Previous

Next

Process: Testing

I evaluated the site's design by viewing the index file across various web browsers to ensure consistency and appearance. If something was off, I used developer tools to fix it and change the code in VS code. Additionally, I validated the code's functionality using validators and monitored the browser console for any errors.



Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for contents of text-input area

Checker input

Show source outline image report

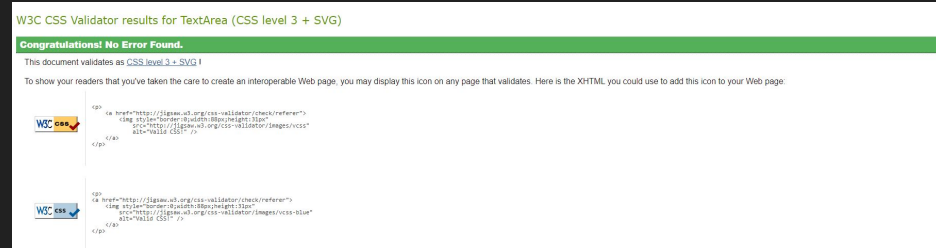
Check by **text input** CSS

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="description"
      content="The Grill on Main has been serving the Mercer County community since 2006. We have a large selection of fresh seafood, unique entrees, and a variety of draft beer. Dine in with us or order online today!">
    <meta name="author" content="Martí Rosenbäck">
    <meta name="keywords"
      content="Grill on Main, Restaurant, Family Diner, Fresh Seafood, Large Selection of Seafood, Tennessee Chicken, Chicken, Ohio Restaurant, Colwater Ohio, Mercer County, Online Ordering, InstantOrder Online Ordering, Restaurant Online Ordering with InstantOrder">
    <title>The Grill on Main | Colwater, Ohio</title>
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings

Message Filtering

Document checking completed. No errors or warnings to show.



W3C CSS Validator results for TextArea (CSS level 3 + SVG)

Congratulations! No Error Found.

This document validates as **CSS level 3 + SVG 1**

To show your readers that you've taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the XHTML you could use to add this icon to your Web page:

```
<img alt="W3C CSS logo" data-bbox="508 688 538 708" style="vertical-align: middle;"/>
<script src="http://jigsaw.w3.org/css-validator/check/referer">
</script>
<script src="http://jigsaw.w3.org/css-validator/checked/1.0.0">
</script>
```

```
<img alt="W3C CSS logo" data-bbox="508 772 538 792" style="vertical-align: middle;"/>
<script src="http://jigsaw.w3.org/css-validator/check/referer">
</script>
<script src="http://jigsaw.w3.org/css-validator/checked/1.0.0">
</script>
```

Tools and Sources

Tools/Technologies

- VS Code
- Figma
- Developer Tools
- W3C Validation Service

All images are my own except for the wooden background image. The background image was licensed for free from Adobe Stock.

Stock ID: 401061075 Author: Dmytro

Reflection

Given more time, I would have dedicated effort to enhancing the visual design of the JavaScript features, particularly by adding more CSS elements to the FAQ accordion. I had issues with the web storage function on the site, and couldn't figure out how to automatically clear the message for users who already signed up for the newsletter. Despite multiple attempts, I was unable to find a solution for this issue. However, I thoroughly enjoyed working on implementing the movies API; it was a rewarding aspect of the project. Overall, I'm pleased with the progress made, but I recognize that there's room for improvement and refinement in future iterations.