

Assignment 4: Optimizing Your Site With jQuery

Javier Sin Pelayo

URL (main branch): https://github.com/javisin22/csc372_projects

URL (client_site_v3): https://github.com/javisin22/csc372_projects/tree/client_site_v3

URL (live site): [here](#)

Part I: Review & Optimize Scripts With jQuery

1. What element(s) did you change to be accessed with jQuery instead of JavaScript DOM queries?

I didn't have any elements accessed by JavaScript DOM before, but I added one directly to use with jQuery.

This way, I added a modification for the **CSS** that affects the **<body>** element. I also added a new behavior that I thought of in this week's Discussion, but I'll talk about it in the Events section.

2. How did caching improve performance for frequently accessed elements?

It improved performance as it's accessed two times so the query is just being made once.

3. What jQuery method(s) did you use to update the DOM tree, manipulate elements, and manage content dynamically?

I used the '.css' statement to modify the CSS of the body to change from:

overflow: auto* → *overflow: hidden

I also used another set of methods but related to the event handling and animations. I'll talk about them in the Events section as well.

4. Did you rewrite all the code in your script(s) to use only jQuery? If so, explain why.

I wrote it from scratch, and to be precise, I injected the code in a *useEffect* in the file where it is used itself, not in a separate "js/jquery-script.js" file.

5. If not, explain why you still needed to use JavaScript for certain features. Provide advantages and disadvantages of using jQuery over plain JavaScript in your project.

As mentioned before, I didn't have any separate JavaScript file before, I used it within the files directly inside useEffects or private functions.

However, I know the advantages and disadvantages of these two strategies as the Professor just explained them this week in class.

The main advantages would be:

- a. **Simplified Syntax & DOM Manipulation:** with methods like `.addClass()`, `.css()`, `.append()`, ... it makes DOM selection and manipulation easier and more readable compared to verbose plain JavaScript.
- b. **Cross-Browser Compatibility:** jQuery abstracts away many browser inconsistencies. It ensures that my code works uniformly across different browsers without needing conditional code.
- c. **Rich Ecosystem & Plugins:** There's a vast ecosystem of plugins and extensions available for jQuery, which can quickly add features like animations, form validations, and UI components without writing a lot of custom code.

Disadvantages:

- a. **Increased Bundle Size:** Adding jQuery adds extra kilobytes to my project. In modern web development, plain JavaScript and native browser APIs are often more lightweight.
- b. **Potential for Conflicts with React:** In a React + Next.js project, jQuery's direct DOM manipulation can conflict with React's virtual DOM management, leading to unpredictable behavior if not used carefully. This is why I've just implemented a single simple behavior (the button animation in the Contact form).

Part II: Enhancing Interactivity with jQuery

1. Which event(s) did you modify? How?

I added the functionality of having a button in the "Contact" section, that after clicking it, it fades out (using jQuery's `.fadeOut()` method) and then showing the actual contact form (using the jQuery's `.slideDown()` method).

I used the `.click()` method for the button, and then nested in functions I used the `.fadeOut()` and `.slideDown()` methods as I just mentioned.

2. Why did jQuery's event methods improve your site's performance or maintainability?

Using jQuery's event methods can improve maintainability by abstracting away cross-browser differences and offering a more concise consistent API.

Also, jQuery makes event delegation straightforward with methods like `.on()`, which allows me to attach a single event handler to a parent element to handle events for many child elements. This can improve performance by reducing the number of event listeners attached to individual elements and simplify code maintenance.

3. List the effects and animations you implemented.

I just implemented an animation for the Contact Form, because as I said, using React can be dangerous as the direct DOM manipulation from jQuery against the Virtual DOM approach from React could interfere. This animation basically adds a fade out effect to the button that shows the form, and a slide down effect to the form itself while it appears.

4. How did they improve the user experience on your site?

This animation enhances user experience by providing a smooth, visually appealing transition that clearly communicates the change in interface state. When the "Show Contact Form" button fades out and the form slides down, it helps users understand that their action triggered a new element to appear, reducing cognitive load and making the interaction feel more natural. It also provides immediate visual feedback, which can increase the perceived responsiveness and professionalism of the site.

Part III: Hosting Your Site

The live site is hosted in *Render*. Here's the link to access the live site:

<https://classconnect-18h8.onrender.com>