# COMP 488 Assignment 2 Submission Write-up

## Submission Links

• GitHub Repository: https://github.com/yaredabebel/comp488-vercel-app

• Live Website: https://comp488-vercel-app.vercel.app/

## Project Overview (300–500 words)

### What I Created

For this assignment, I developed a responsive website that combines HTML, CSS, and JavaScript into a cohesive application. The site includes an introduction, an interactive feature powered by JavaScript, and a showcase of technologies used. The design emphasizes a clean, modern layout with gradients and smooth transitions while ensuring accessibility through semantic HTML. Responsive design was a priority, with Flexbox, CSS Grid, and media queries enabling the layout to adapt seamlessly across devices.  
  
The interactive component — a button that toggles hidden content — provided a simple but effective demonstration of DOM manipulation and event handling. This feature, paired with a well-structured design, shows how front-end technologies can work together to create both functional and visually engaging web applications.

### Experience with Git, GitHub, and Vercel

I’ve worked with Git and GitHub on prior projects, so the version control workflow felt familiar. Committing often, writing clear messages, and maintaining a clean history helped keep the project organized. GitHub’s repository tools made it easy to track progress and manage the codebase.  
  
Vercel, which I’ve used before, streamlined the deployment process. The GitHub integration made it straightforward — each push to the repository automatically triggered a new build and updated the live site. This continuous deployment workflow reinforced what I already appreciate about modern hosting platforms: the ability to focus on coding while letting the platform handle delivery.

### Learning Outcomes and Challenges

One of the biggest takeaways from this project was improving the responsive design. While I had some experience with CSS Grid and Flexbox, fine-tuning layouts for different screen sizes required additional attention. Testing on smaller viewports exposed areas where spacing and alignment needed adjustments.  
  
A key learning was refining the separation of concerns between structure, style, and interactivity. While I already understood the concept, applying it in a real deployment cycle reinforced how much cleaner and maintainable code becomes when HTML, CSS, and JavaScript are kept distinct.  
  
The most challenging part was ensuring consistent cross-browser behavior and polishing the user experience across different devices. Iterating through these challenges not only solved immediate issues but also built stronger habits for testing and debugging in future projects.  
  
Overall, this project was less about learning everything from scratch and more about building on prior knowledge. It gave me the chance to refine workflows I’ve used before, apply best practices, and see the entire development cycle — from writing code locally to deploying a live site — executed in a professional, streamlined way.