By Artur Agalarian
CS463 WebDev

# Journal

## List of features

My personal website consists of HTML, CSS, JavaScript, and Bootstrap library. I've used a single HTML file, a single CSS file, and a single JavaScript file to create the project, meaning that I only have one page, keeping it simple, sort of speak. Note that I do not use JQUERY

There are a lot of features on this website that were shown in class to some degree, most of them highlighted in this section; the features that were not shown in this class are in their separate section down below.

Note that the outside sources used are this course's lectures, homework, and labs, as well as the outside documentation such as Bootstrap documentation, and general questions regarding CSS and HTML.

The links to the outside sources could be found on the bottom of the page.

## Bootstrap

I haven't been too extensive on using Bootstrap in this project, though; the biggest usage examples are the navigation bar and the project section. Examples and more in-depth explanations are below. I really didn't notice how much more CSS I was using compared to Bootstrap until the website was already looking pretty good and serviceable, plus there was more class material on CSS to be implemented.s

### Navbar

The navigation bar is looking similar to its counterpart in one of the homeworks I've been working on.

**Note that I'm using `sticky-top` element that was not covered in the class**. Its inclusion is attributed to how useful having a navigation bar always at the user's disposal is at all times, without having to scroll to the top all the time to access it!

Specifically, the Homework 3 nav bar with a shrinkable list of links to the internal website. Links lead to sections such as “About me”, “Resume”, “Projects”, and “Contact me”. In order to

achieve this, I've utilized a lot of attributes as shown in both the homework 3 assignment as well as the Bootstrap documentation linked down below. Shrinkable navigation bar is outstanding for zoomed-in devices!

Below are the snippets of the expansion on my previous example, showing the primary navigation bar. We're showing the navigation bar itself, the expand button that will be used to open the link menu, and the div containing the actual links. The rule `navbar-expand-md` ensures the collapse rule will be triggered when the website size is set to medium by Bootstrap standards.

## Carousel

**The second example for the extensive use of Bootstrap is something not covered in class, which is the `Carousel` rule**. The Carousel allows to seamless incorporation of the slideshows of the HTML elements.

I've used the Carousel in the section for `Projects Showcase`, where each slide of the Carousel is an individual project. The reasoning behind using it to list the projects is that each of the 2 projects takes up a lot of space on the page, making it a bit tedious to navigate. Using a carousel allows us to conserve space. and give the user a seamless way to interact with projects I've worked on.

My Carousel consists of 3 primary elements: 1) left/previous button, 2) right/next button, and 3) slideshow itself, which are the items of the inner carousel. Note the extensive use of the bootstrap as advertised!

The inner part of the Carousel contains 2 carousel items, which are 2 of my projects. I had to set one of the projects to be `active`, such `<div class="carousel-item active">.  The reason behind this is to signify which slideshow must appear first by default. Each project consists of showcase and an information side. The showcase display the name of the project and the images related to the project. Information lists the description, my participation, and links to the project.`

## Other

Such other minor examples of using Bootstrap include:

- The gradient background present in `<main>`, `<nav>` and some other elements. This is to give the website some style, and I don't recall seing this in the lecture
- I'm using colors such as `bg-dark` and `navbar-dark` in order to give navigation bar some style without changing too many colors

# Accessibility

## Semantic elements

In order to make the website more accessible for the masses, I was sure to include as many semantic elements as possible.

The body element of my website has exactly three semantic elements; which are <nav>, <main>, and <footer>. This ensures that the user/browser is able to partition contents of my websites in the beginning.

The <nav> semantic element does as advertised, it lists the anchor elements that allow users to quickly travel around my website. Such anchor elements primarily allow you to travel inside the <main> element, but there's one link to the footer, which has contact information.

The <main> element has all the primary information for the user, whose only children are "<section>" children: "About me", "Resume", and "Projects". Such a distribution allows users to further narrow down their desired content. Furthermore, each project in the "Projects" section is represented as an <article>, giving more insight to the browser about what kind of information we're displaying. Each image in the project contains the alternative text, and the links are provided to my projects as well as some terms I'm using

The <footer> element only exists to provide the <section> "Contacts information", containing two essential elements to do so. Those elements are the list of my contacts as well as the form for the user to fill out if they would like to reach me personally. Each of my contacts provides the appropriate link, such as my email address being "mailto:king.artur.arv@gmail.com ", automatically redirecting the user to Gmail to send me mail. The contact-me form has some basic information about the user who would like to contact me to fill out; submitting the form will print out the information to the console, as well as spawn the pop-up window with information provided by the user.

## Wave Extention

In order to meet the established standards of accessibility, I've made sure to run the WAVE Accessibility Extention on my website both using Firefox and Chrome browsers. I made sure to fix all the regular errors, contract errors, and as many alerts as I could. In the end, I only have 2 alerts on my website, such as "Link to PDF document" and "Plugin"; but they should not be a deal breaker because alerts are not the cause of major problems.

I've discovered that my website has around 19 features overall using WAVE! As well as 21 structural elements and 2 ARIA. I wasn't aiming on having this much, but I think a big variety of useful elements will benefit the recognition and quality of the website.

## W3 validator for HTML and CSS

Remembering the requirements for either the lab or homework assignments we've had previously, I've made sure to run the extra validators on both HTML and CSS files I have. I didn't get many errors, and fixing them didn't take much time. The primary HTML errors were regarding my using the ending "px" when specifying the width and height of the image in HTML. The CSS problem was me using commas when specifying all 4 values for the margin.

### Dark theme

This one might be a little silly and subjective, but dark-themed websites are always more preferable to me than light-themed websites. This is primarily because they do not hurt your eyes whenever you're using the website in a dark environment, and dark extensions sometimes do not work properly.

## Responsiveness

### Hoverable buttons/anchors

I've made sure to make major buttons and anchors to be visibliy hoverable throughout the website, so that the user will be able to tell what parts of my website are interactable. Such stunning examples could be seen in the navbar, as each of the links are hoverable and intiuitive to click on; as well as the PDF collapse buttons and carousel buttons being very obvious to notice and hoverable. All of this was done using simple CSS and classes.

To help with it, I've defined the variables that hold the hoverable background colors and regular colors, as well as colors for the links.

### Media Query

My CSS code has a special media attribute that is triggered whenever the website's width is less than 1000px. Triggering this will change the layout of the website to display items vertically because normally, I have the left and right sides for my content. Such a good example is the about me page, normally it would display my picture to the left and my info to the right, but a media query makes the image appear above my info. Similar practice is mirrored for my projects and contact information. This allows users with smaller screens to see all the content of the website clearly. Note that some images are scaled down as well.

### Navigation bar shrinking

This is similar to a media query but it utilizes the Bootstrap library to perform a similar work. Whenever you're zoomed in to Bootstrap's medium or less defined size of the website, the navigation links will be removed, placing the single collapse button instead. Clicking on the collapse button will expand the dropdown menu that will show all the navigation links. Th is was quite simple to implement because the similar idea was demonstrated in Homework 3

Note that I've added that I've made the navigation bar sticky using Bootstrap as well; this just seemed to be a good idea overall, as explained in the Bootstrap section above.

### Collapsable Resume

Having a big PDF file being shown on the page all the time doesn't seem to be quite good for the user experience, especially when the user is done looking at it. Thus, I've implemented the custom collapse button that will collapse the pdf, as well as show it again if the user clicks on the button again. I think this is a good design overal because pdf files are kind of big and awkward to scroll through.

And as a sidenote, I've added a little link just below the resume whenever it is shown, providing the user with the direct link to download the resume if they choose so.

## JavaScript

My JavaScript essentially has the purpose for the two logic loops: a custom collapse button for the resume PDF, and the pop-up window for the form submission. Overall, I've defined three functions in order to achieve all of the above, making sure to call those functions to define the event behavior in the "DOMContentLoaded" event listener; the reasoning behind putting it in "DOMContentLoaded" is to make sure that JavaScript executes only when the DOM is loaded.

First is the collapse button for the resume file. It is a short and sweet click event that either hides the PDF by setting display to none, or shows the PDF by setting the display rule to block. It also changes the messages to "View Resume PDF" and "Hide Resume PDF". I've used query selectors to make the job easier. I do not create any elements in JavaScript at this step.

Second is the form pop-out. After the user submits the Contact-me form, JavaScript creates a pop-up window with all the information the user has provided. Additionally, we're logging the same output to the console. The pop-out window exists in the footer near the form itself, all JavaScript does is to make it display block to appear, and display none to disappear when the user clicks the close button. The only notable parts of this JavaScript is the use of FormData class to retrieve the relevant form data, as well as parsing checkboxes to arrays when at least two has been checked. Note that for each form item input provided, we create an element to be appended to the pop-up window.

## Process

I'm quite pleased with the commit messages I have created during the development of the project. There are around 23 of useful commits throughout, which now I think of it is kind of a lot. I'm not sure if I have overdone it or just making sure to track smaller differences.

### Milestone #1 (June 4th)

When I've started the project, I was sure to give the website the overal structure first in order to understand the scope of the problem. To do that, I've defined the barebone bootstrap collapsabl e yet buggy <nav> element, <main> element that contained the <section>'s to for our project requirements; about me information contained a small paragraph about me and my picture, yet the list of projects was still empty at that point yet already had the bootstrap carousel. As well as the <footer> element to have the Contact me <section> with short list of my contacts.

The styles had primarily focused on establishing the main colors of the website as variables, and trying to apply them on the website elements. I've had trouble with making the website dark-themed because my top-level elements couldn't be colored. To address that, I've added the "!important" rule to two of my top-level elements' colors to do so.

I've imported Bootstrap, created HTML, CSS, and JavaScript files, as well as added my profile picture. JavaScript was an empty file at that point, though.

### Milestone #2 (June 5th)

In this milestone, I've made sure to *" repurpose carousel from displaying sections 'previous work' and 'projects' to instead be inside 'projects' section; now displaying a list of projects"* as said in the commit message (I'll keep on quoting my commits).

Furthermore, I've added the pdf of my resume to the resume section, as well as providing the collapse button using JavaScript to collapse and expand the resume as the user desires. Carousel was quite buggy because I've messed up the classname and button positioning; classnames were in general pretty messed up for each major section, so I've fixed them by naming them appropriately.

### Milestone #3 (June 6th)

This day was pretty good, I was able to fix and refactor a lot of minor inconsistencies, reconsidering some style choices for my website.

I was able to complete all the required information about the Stellnull Project. This was both in regards to the content, images, links, as well as the CSS styles. I was pretty impressed at how much I can write about my past projects.

I've had a lot of minor changes, such as renaming the website title to have my name, expanding the color variables to include hoverable and link colors, and fixing most of the internal navbar links to point to the correct sections of my website. Also, my contact information now had the list of contacts instead of them being in paragraphs. And the buttons now are visible and hoverable with more-or-less consistent colors.

## Milestone #4 (June 7th)

I have made quite some progress on this milestone, having a larger number of commits than before within a day.

First of all is the big one, I've completed the second and final project, Playverse, which didn't really take me that much time because CSS classes were already defined for my first project.

I've expanded on my about–me section, listing out a lot of different experiences I've had throughout Computer Science both on and off campus.

Basic media queries were added as well into CSS to account for website width below 1000 pixels! From the start, I was aiming at making all the content of the website centered and make the overall direction to be column-based whenever the website was too zoomed in.

I've pushed my changes to the Github repository, being able to deploy my website with ease in the main branch. I had to resolve some merge conflicts, but it was just a matter of removing old changes.

I've added a form to the contact-me section, allowing the user to fill out the form with their information if they would like to contact me i any way. I made sure to utilize the basics of the inputs such as text, email, radio, checkboxes, as well as submit and reset buttons. I've made sure to add JavaScript to output the result of the form into the console.

## Milestone #5 (June 8th)

The last day of working on the website, I was dedicated to finishing the project a little earlier by starting early too! Though as a sidenote, It did take me a lot of time each day to make progress; I think I should've started even earlier to have more free time, oh well.

The pop-up window is now displayed whenever we submit the form! It was as simple as adding a new HTML element into the footer, as well as define some logic in JavaScript to dynamically append new elements.

I've done a lot of work in order to meet the accessibility and responsiveness requirements, specifically on this day. Such examples would be:

1. Using Wave to fix contrast errors in "contacts" and repetitive links in the "projects", and fixing the "Projects" navbar link to point to projects instead of contacts
2. Allowing the user to have a link to download the PDF directly.

3. Running the W3 validator to fix width and height values defined in HTML now do not have "px" ending, fixing the resume pdf file to not have spaces in the name, and fixing css bug where margin values had been comma-separated

I was able to complete the short Readme.md file as was expected of us, listing out some information about the project and replication.

And for some reason, I had to fix a but where the resume button saying "Hide Resume PDF" when the pdf wasn't shown, and to meet the completion, I've added a short paragraph just before the resume to make the section look a bit less empty.

## Sources


1. Bootstrap documentation: https://getbootstrap.com/docs/5.3/getting-started/introduction/
2. CS463 on canvas (including lectures and examples): https://canvas.pdx.edu/courses/101623/modules
3. StellNull project: https://noodlestrudel.itch.io/stellnull-jetball
4. Playverse-chatapp project: https://canvas.pdx.edu/courses/101623/modules
5. Link to github-pages introduction: https://pages.github.com/
6. CS463 Homework repo: https://github.com/lowercasewords/webdev-homework
7. CS463 Lab repo: https://github.com/lowercasewords/webdev-lab-notebook
8. W3 HTML validator: https://validator.w3.org/#validate_by_upload
9. W3 CSS validator: https://jigsaw.w3.org/css-validator/
10. Firefox WAVE Accessibility Extension: https://addons.mozilla.org/en-US/firefox/addon/wave-accessibility-tool/?utm_source=addons.mozilla.org&utm_medium=referral&utm_content=search