PART I | **AUDIENCE & GOALS**

I created a “My process” page for my forthcoming portfolio. Note that this is NOT going to be the index page of my site once it is built out next semester. AFTER recruiters and prospective employers have viewed my projects and resume, the goal of this page is to give these audiences more insight into my unique skillset and approach to design.

For me, the MHCI program represents a mid-career shift. I already have a decade of professional experience in design-tangential domains. During that time I have cultivated a set of skills that enhance my effectiveness as a designer and leader. I represent these skills as the nodes of my radar chart. Through the page’s interactions, I want to showcase those unique qualities, my love of data visualization, and the diversity of project types I have worked on. When flipping through the project carousel, the radar chart’s changes quickly and visually model my flexibility and array of experience for prospective employers.

PART II | **INTERACTION AFFORDANCES**

On load, the page uses progressive disclosure to call attention first to the radar chart and “my design process” header; then to the projects; and finally, to the “why a radar chart?” button.

- The user then hovers over the button, phrased as a question to provide a call to action and a guide to interpreting the page.
    - When hovering over the blue button, a tooltip appears.
    - This directs the user to hover over the radar chart nodes for more detail on each leadership quality, and then to hover over projects to see the radar chart change.
- When the user hovers over each radar chart node located near the chart labels (denoted by a light gray circle), each circle expands and a tooltip appears with more detail on the leadership attribute.
    - Hover over each node of the chart to reproduce the interaction.
- The project carousel auto-cycles through project thumbnails, displaying each project. This will be more interesting when I have more than three projects to display!
    - Click the arrows on each side of the carousel to advance or go back.
    - Hover over each project to see which skills I used - the radar chart will change accordingly.
    - Hover over “my design process” to reset the chart.
- On small or extra-small screens, a hamburger menu appears.
    - Click the menu to expand or detract the menu.
    - Click on resume to view my resume.

PART III | **TOOLS**

Javascript libraries

- Chart.js
    - I chose to use this library because it provided clean chart code. I tried several data visualization libraries with radar charts, but many were outdated or had errors. As this is not a widely-used chart type like a bar or line chart, my options were more limited. I found it much more beginner-friendly than D3, as I needed to customize my tooltips and labels.
    - I used it to implement the radar chart. I created two data arrays, with the first as a dummy set (all data points at 100) so I could create the node hover capability. The second data set is reflected in the yellow polygon, which is redrawn using Javascript each time a project or the “reset” is hovered over.
    - This enables the main visual impact of the page: the ability to quickly scan projects and visualize my diverse skillset. The built-in ability to change the node radius on hover (the gray circles) also provides feedforward and feedback for users that the nodes are interactive.
- Bootstrap (includes JQuery and Popper)
    - I chose to use Bootstrap for its powerful functionalities including responsive resizing, hamburger menu open and collapse, tooltips, and the image carousel.
    - I used it to:
        - Implement a responsive grid system
        - Create a responsive top navigation menu
        - Minimize the information shown on the page with a tooltip, not revealed unless hovering over the blue button
        - Create a carousel displaying my projects
    - The carousel and responsive aspect of Bootstrap enable the user to see the entire page above the fold on both desktop and mobile. For the interaction to have impact, the user needs to see the projects and radar chart concurrently. This was a design challenge at first, but Bootstrap helped greatly. Bootstrap also enables viewing on all sizes of displays.

CSS animations

- Animate
    - I chose to use animate to gradually disclose sections of the page on load.
    - I used animation-delay to progressively disclose the top half, bottom half, and then the callout button.
    - This allows users to process information in the order I desire.
- Transition
    - I used CSS transitions to create the overlay effect on projects in the carousel.
    - I used transitions to initially hide the overlay with a project description and “See Project” button. On hover, the overlay appears and the image behind is reduced to 10% opacity.

- This adds more progressive disclosure, keeping the look of the page clean and only providing information when it is queried.

PART IV | **ITERATION**

Though I implemented much of my page the way I planned in my mid-fidelity mockup, I made a few changes:

1. Instead of creating new text divs when the user hovers over radar chart nodes, I used Chart.js tooltips. This was challenging to implement, as I needed to customize what was included in the tooltips themselves using Javascript.
2. I initially loaded all my project thumbnails on the lower half of the page and created a half-scroll. After showing my page to various users for feedback, one recommended implementing a carousel for a cleaner look.
3. I talked to several users about how to frame how I was using the radar chart. That conversation led me to eliminate subheaders and implement the blue “Why a radar chart?” button.

PART V | **CHALLENGES**

Though the final output looks simple, this project stretched my skills significantly. I was reminded once again of the importance of local vs. global variables in Javascript functions; when I needed to pass new arrays into the chart on project hover, I had to reorder my functions to ensure I could call the needed variables. I struggled more than expected with Chart.js and Bootstrap documentation, which was more minimal than I would have liked when I was trying to create custom solutions like the node hover tooltips on the chart. I also struggled with my top navigation breaking into a new line on collapse until I found the issue: flex-wrap, a built-in Bootstrap attribute!