Front-End Dev
Project Brief

ES1. Responsive Design & ES2. Computational Thinking

Driving Questions:

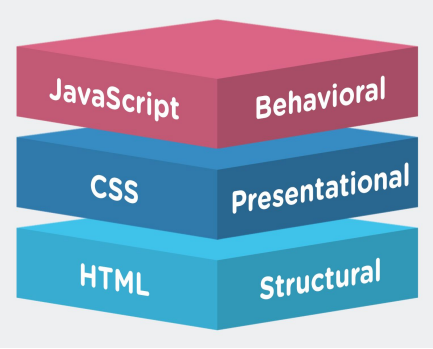
How do we utilize the power of CSS to efficiently customize and style a web page?

How do we develop a responsive web page that changes based on the viewport of the device it is loaded on?

How do we write JavaScript code to add interactive behavior to a web page?

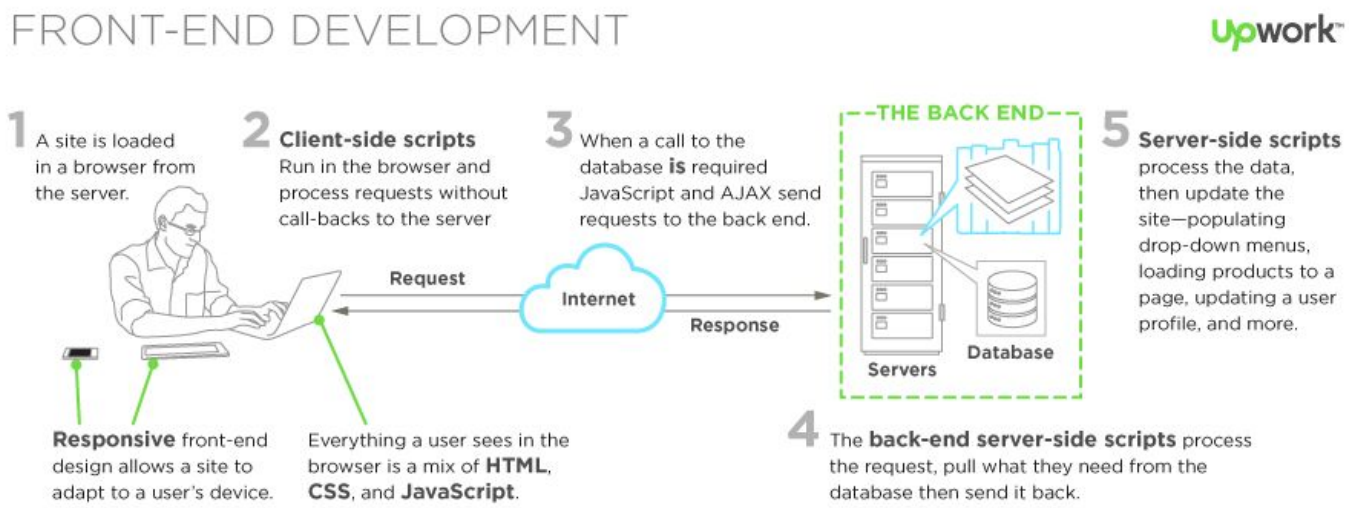
How will the user intuitively understand how to navigate your site?

Project Overview:



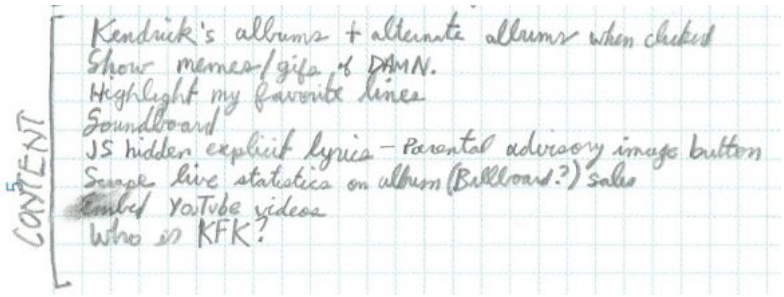
The front end of a website is the part that users interact with. Everything that you see when you're navigating around the Internet is a combo of HTML, CSS, and JavaScript being controlled by your computer's browser. Front-end developers are responsible for a website's user-facing code and the architecture of its immersive user experiences. In order to execute those objectives, front-end devs must be adept at three main languages: HTML, CSS, and Javascript programming.

Using the three core languages of the web, you will develop a modern site with interactive features to promote a personal interest of yours that is worth sharing with the CS Pathway. This site will be debugged and optimized for maximum speed, scalability, and UX; both desktop and mobile versions will be designed, programmed, and implemented.

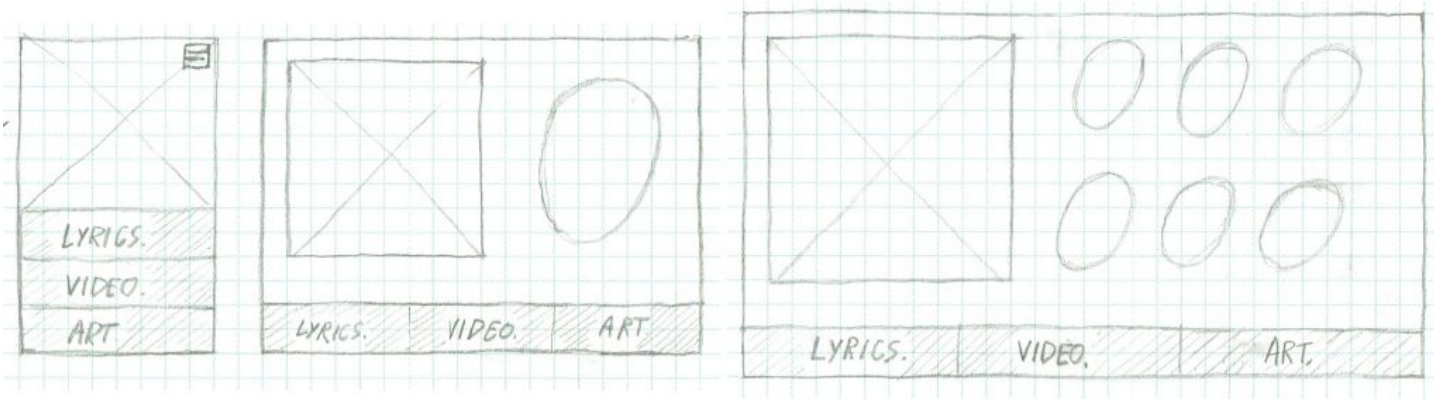


Project Phases:

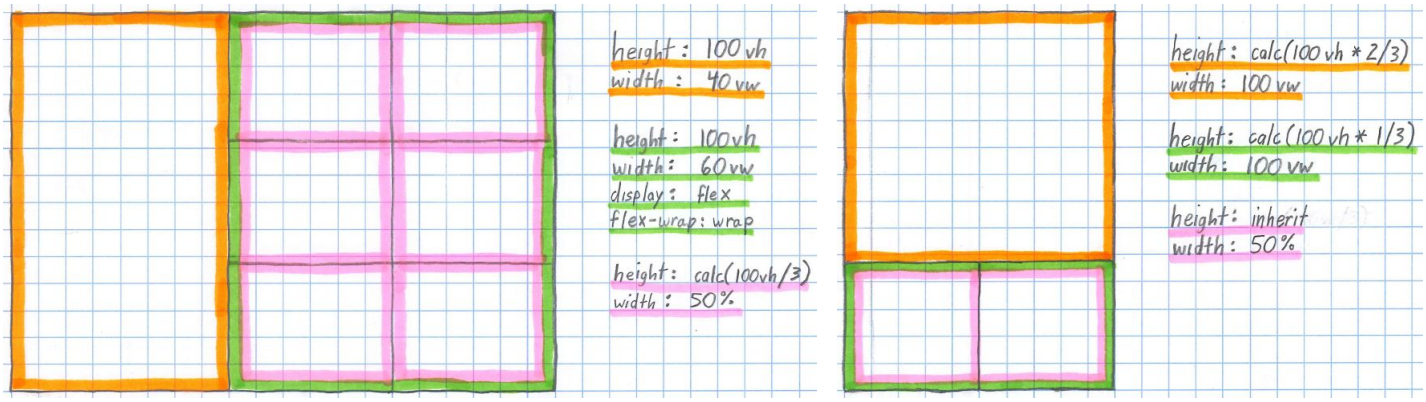
Phase 0: Brainstorm site idea, essential content, and interactive features.



Phase 1: Sketch out multiple desktop + mobile wireframes considering how content will reflow.

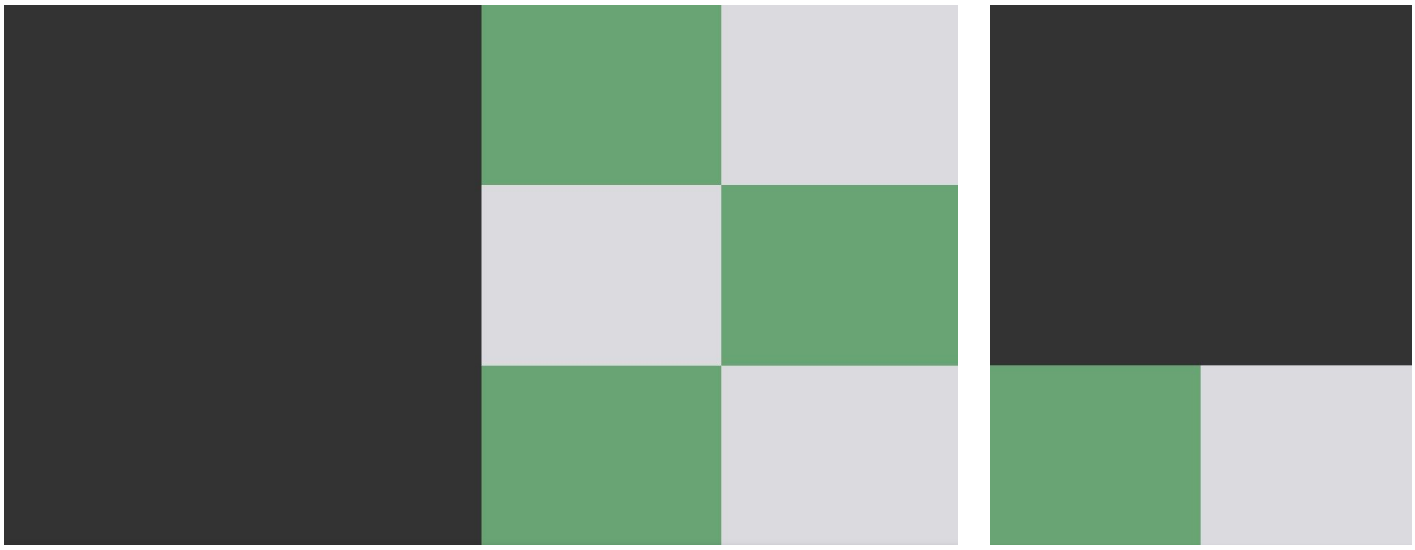


Phase 2: Plan your DIVs thinking about how your page will reflow and being specific about the heights, widths, and display properties you will use.



Phase 3: Code the desktop layout (no content, only colors) using DIVs, classes, Flexbox.

Phase 4: Set your breakpoints using @media queries to reorganize DIVs.



Phase 5: Add content + styling details and debug.

Phase 6: Insert two pieces of tested JavaScript code and debug. Add comments in your .js file explaining the functionality of all pieces of your code.

Phase 7: Fine-tune the UX (user experience) details considering the questions below.

- How will the user know how to interact with your site without being given instructions?
- What feedback will the user receive from your site?
- How does the user know where they are within your site?
- How does the user know that a feature is interactive?

Timeline:

Date	Mile-stone	Deliverables	Event
10.09 - 10.13	**0**	Initial Brainstorm	Project Roll Out
10.16 - 10.20	**1**	Images of **Phase 0**, **Phase 1** posted on Google Sheet.	Planning Documents Completed
10.23 - 10.27	**2**	Working link to **Phase 2** posted on Google Sheet.	Programming Begins
10.30 - 11.02	**3**	Working links to **Phase 3** & **Phase 4** posted on Google Sheet.	Field Trip to DSC! Field Trip to CPP!
11.06 - 11.09	**4**	Working links to **Phase 3** & **Phase 4** posted on Google Sheet.	New CS Lab!
11.13 - 11.17	**5**	Photo editing in **Photoshop** & audio editing in **Audacity** Working link to **Phase 5** posted on Google Sheet.	UX Testing @ Exhibition
11.20 - 11.24	**Thanksgiving Break**		
11.27 - 12.01	**6**	Learn because it is fun.	JavaScript coding.
12.04 - 12.08	**7**	Working link to **Phase 6** posted on Google Sheet.	JavaScript debugging.
12.11 - 12.13	**8**	Link to **Final Deliverable** posted on Google Sheet.	CS POL Presentations

Honors Requirements:

Detail & Quality	*UX Optimization*
● No content is overflowing, overlapping, illegible, not visible, or distorted; clear attention to detail. ● At the mobile breakpoint the layout, images, and text all change in an expected way. ● Site load-time is minimized by minimizing size of all content.	● You have gathered data from 10 individuals who fit the profile of users who would visit your site; this data is from both direct questions and observations you make while the user interacts with your site. ● One page explanation of choices that you made in your HTML/CSS that optimize the UX of your website and how you implemented feedback gathered during the testing phase.

Exhibition Event:

At the DVC Exhibition on November 14th, students will individually present their Super Fan Interactive Experience web page and conduct User Experience testing with guests at Exhibition. The goal of this UX testing to gather feedback about ways that we can fine-tune our sites and make them more intuitive for the user.

Code Requirements:

0. Code in all **HTML**, **CSS**, and **JS** files are tabbed/organized according to all conventions outlined in the Google Style Guide ensuring optimal human readability and file size.

1. All code/files included in your website are **contained** within your GitHub project repository.

2. Blocks of code taken from other sites/libraries must be **cited** in your code by writing a comment explaining the functionality and the URL of your source. Do not copy lines of code you do not understand, you will be penalized for not understanding your code or including redundant code.

<!-- *HTML comment* **-->** **/*** *CSS comment* ***/** **//** *JS comment*

3. Final design must act **responsively**, reflowing based upon the viewport width of the device/browser. At the narrowest viewport breakpoint, all essential content reflows into a single column of stacked divs.

4. All JavaScript code included in your site must be explained with **comments** you have written.

Essential Knowledge Requirements:

ES1. Responsive Development: using only **display: flex**, **display: block**, and **flex-wrap: wrap** to position elements/divs with relative measurements, setting intuitive breakpoints using @media queries.

ES2. Computational Thinking: building reusable code for future use, avoiding unnecessary repetition by understanding inheritance and the cascade, using #ids and classes to target specific elements/divs or specific groups of elements/divs, creating interactive features using JavaScript, using the calc() function to calculate heights and widths.

HoM1. Professional Communication: high quality wireframes for desktop and mobile drawn, clear plan of how divs will reposition/reflow when loaded on a mobile viewport, details and UX of site are improved after testing and feedback is received.

SUPERFAN Interactive Experience Requirement Checklist

HoM. Professional Communication

Accurately communicate problems, solutions, and goals with your clients, your team, and yourself. Revise your work using an iterative process in response to feedback. Support your communication with visual evidence including sketched wireframes, annotations of functionality, and digital mockups of design goals.

Yes	No	
		High quality wireframes for desktop and mobile are **annotated** with relative measurements.
		Wireframes demonstrate how DIVs will **reposition/reflow** when loaded on a mobile viewport.
		The most important content of this site is **emphasized** in both desktop and mobile designs.
		Blocks of code **(1)** taken from other sites/libraries or **(2)** written with substantial help of a peer is **cited** by writing a comment explaining the functionality and the URL of your source.

ES1. Responsive Development

Design and build a website that is responsive to the viewport in which it is rendered. Develop fluency with CSS properties and utilize the cascade/parent-child inheritance relationships to efficiently apply styling.

Yes	No	
		Only **display: flex**, **display: block**, and **flex-wrap: wrap** are used to position elements/DIVs.
		Relative measurements are used to size content on the page.
		The **calc() function** is used to calculate heights and widths of DIVs with dimensions that are repeating decimals when written as percents/viewport units.
		Final design must act **responsively**, reflowing based upon the viewport width of the device/browser.
		The **breakpoints** have been set using @media queries and occur at an **intuitive** viewport width.
		At the mobile breakpoint the layout, images, and text all change in an **expected** way.
		At the narrowest viewport breakpoint, all essential content reflows into a **single column** of stacked DIVs or an equivalent layout that works for your site.
		No content is overflowing, overlapping, illegible, not visible, or distorted; clear attention to **detail**.

ES2. Computational Thinking

Develop, implement, and analyze algorithms/abstraction to improve readability and avoid repetition using Python and JavaScript; use and identify mathematical/logical concepts to improve the functionality of programs.

Yes	No	
		Two interactive JavaScript features are coded and triggered by a user click.
		All JS is explained using **comments** specifying how HTML, CSS, & JS interact to make each feature work.
		Reuse of code is demonstrated by calling a single function from multiple elements on the page.
		Reuse of code is demonstrated by using a single class to style multiple elements on the page.
		Student can demonstrate how the **cascade** is useful when adding/toggling classes on specific elements.
		Student can demonstrate how properties are **inherited** and do not need to be re-specified on child DIVs.
		No **unused** lines of code (this includes classes or single properties within a class).
		Code in all HTML, CSS, and JS files are **tabbed/organized** according to all conventions outlined in the Google Style Guide ensuring optimal human readability and file size (you can also use Eric/Andy's examples as references).
		All code/files included in your website are **contained** within your GitHub project repository (ex: no links to external images).

SUPERFAN POL Presentation Agenda

1. While presenter sets up, the rest of the class spends 1 minute previewing the presenter's site using the link posted on the Project #2 Google Sheet:
    - **a.** Try to figure out the purpose of the site.
    - **b.** Observe how the page reflows when the mobile breakpoint/breakpoints is/are hit.
    - **c.** Find both JavaScript interactive features and observe how cleanly and consistently they work.

2. Presenter gives Eric/Andy their self-scored checklist of requirements then shows their site to the class:
    - **a.** Briefly explains the SuperFan topic and why they chose it
    - **b.** Shows annotated Phase 3 desktop/mobile wireframes and explains how they used relative measurements, display:flex and display:block
    - **c.** Shows how their page reflows at their breakpoint(s) and explains how/why this makes sense
    - **d.** Explains how their JavaScript functions work by walking the class through all three files in the site (HTML Before Click, JS, CSS, HTML After Click) using the "inspect" tool. [Eric/Andy may interrupt with clarifying questions to check your understanding]
    - **e.** Briefly discusses how they want to continue improving their site

3. Presenter leaves the room while the audience discusses the UX score.

    "As the user I understand the purpose of this site and it is clear how I navigate the page."
    "The desktop and mobile pages are enjoyable/professional/legible and the JS features work as expected."

    1 = Strongly Disagree 2 = Disagree 3 = Neutral 4 = Agree 5 = Strongly Agree