Hello All! My name is <Abby> (introduce jenna and linnea too).
We chose bootstrap as our software framework that we wanted to use to demonstrate one of the key concepts of this course: “Code as Data” The concept of "code as data" refers to treating code itself—including functions, variables, and structures—as manipulable, reusable, and dynamic entities. While our project primarily involves HTML and CSS, this principle is subtly present in the reusable and data-driven design of our code. Let’s analyze this principle in our existing code and how it ties directly to the idea of "code as data” as we walk through how you can integrate bootstrap to make a remarkable website.

Bootstrap is a popular open-source front-end framework used for building responsive, mobile-first websites and web applications. Created by Twitter developers Mark Otto and Jacob Thornton in 2011, it offers a collection of CSS and JavaScript tools for creating sleek, modern web designs. Bootstrap includes pre-designed components like navigation bars, buttons, modals, forms, and carousels, which developers can easily customize and integrate into their projects. It also includes a grid system for layout structure, making it easier to create responsive designs that automatically adjust to different screen sizes and devices. The framework supports various JavaScript plugins for features like tooltips, popovers, and image sliders. Bootstrap promotes consistency, as it reduces the need to write custom CSS from scratch, streamlining the web development process. It is widely used due to its ease of use, flexibility, and the large community of developers and contributors supporting it. Today we will be mainly utilizing bootstraps buttons. *Show the bootstrap website*

With that being said, let’s start by creating a new HTML file. In order to include bootstrap into our code, we will have to link the bootstrap content directly into our code by doing this: *linnea creates the document and links the bootstrap thingy at the top*

Now that we have that out of the way, our next step is to start coding! What we’re looking to make is a colorful website inspired by the recent BlockBuster film “Wicked”, we want to make a website that has an interactive link that sends you to the specific song’s spotify link of your choosing. In order to do this, we will incorporate a button for each song on the soundtrack. We are creating a “Button” class that is already made for us in the bootstrap library. From here we can add each individual song link and a title for the button by creating a reference. Our buttons link directly to each individual song through the spotify link. Along with this, we include a title that is displayed on the top of each button with the title of the song: *Linnea makes some really cool button stuff and that will be epic as shit*

The Bootstrap buttons themselves are an excellent example of how code can function as data. Each button is defined by a simple class, such as `btn` or `btn-primary`, and these classes act as data that dictate the button's style and behavior. For instance, the `btn-primary` class contains data about the button’s default color, padding, and font size. When we apply this class to a button in our HTML, we’re essentially passing data to that button, telling it how to look and behave. By changing or adding new classes, such as `btn-large` or `btn-warning`, we alter the data that the button uses, which results in a completely different appearance or interaction. This makes the button code reusable and flexible, as we can simply manipulate the data (the

class names) to create a variety of buttons with different styles, sizes, and functions, all without needing to rewrite or duplicate code. In this way, the Bootstrap button is a data-driven element that adjusts based on the values (or "data") we assign to it. For our code, we used a custom button class so that we could customize our buttons and make them even fancier.

Now, since we have the buttons made we can also create a fancy little header with glitter and sparkles, just because we can: *Linnea makes the header*

Since we want our website to be pretty, we are also going to create a CSS file that will help us with styling our page. We are going to create a root that will define the rest of the variables that we use throughout the rest of the file. In here we will create our first and second color, as well as the blur for our background. Now, we can specify our colors for the hovers and button colors as well as other styles: *Linnea does the funky style stuff here.*

CSS variables serve as a key example of how code can act as reusable data. They allow us to define values once and reuse them throughout the codebase, enabling flexibility and dynamic styling changes without repetitive code. This is shown in our color definitions in first color and second color. Here, `--first-color` and `--second-color` act as **data** that control the behavior of the CSS rules. By assigning the value of `--first-color`, for instance, we dynamically affect every button styled that calls the first color variable. This abstraction is fundamental to "code as data". When we use a variable such as `--second-color,` it looks up the variable value that we defined earlier in the code. It treats this section of code as data and a color. This then gets implemented directly into our bootstrap code as the button's style depends directly on this.

Now that we have all of our code, let's take a look at our fancy website! As you can see it is really pretty. Now, let's look at where our code created certain things. By using bootstrap, we were able to create buttons that utilized the data-like code in order to make something beautiful. As you can see here, when you hover your mouse over one of the buttons, created by bootstraps, it takes in the data provided by the code that we specified in our CSS file. Because of this, when we hover over the buttons, it is able to change the color because it takes in the data! As you can see, our website is stunning and functional and perfectly implements bootstrap's frameworks to make it look the way it does. Not only this, but it clearly implements basic concepts of programming languages.

When looking at our website we can see many different elements all together. We see that the heading is glittery and unique based on our specifications. Beyond this, each of our buttons directly links to the songs that are listed.

Reflection:

Hey this is Jenna here. I am going to talk a little bit about our reflection as a group on this project. One thing I have learned over the years is that the best way to learn or to apply your learning is by using what in application with something you have done before or do on the daily. Applying something we have learned from this class to something we already know how to do helps us see how what we learned is applied in our current toolset. As soon as we saw we were able to do bootstrap, we jumped on that idea because we have all worked with it previously and we felt we could use it to incorporate things we have learned in this class. So since we have already coded websites and used bootstrap before, we felt this was a good way to go.

A little piggy back off this idea is also the fact that we chose to make our webpage about the soundtrack of Wicked. Which is a new movie we all have watched and enjoyed. This was a great idea because working with something you love or have passion for can accelerate learning the material as well as getting the project done.

Overall, this project helped us understand how the idea of “code as data” works in frameworks like Bootstrap. By using reusable stuff like buttons and classes, we saw how this makes coding easier and more flexible. For example, we could change the look and behavior of buttons and styles without rewriting everything, which saved time and effort. This made our website look cool and work well while also showing how useful reusable code can be. It also showed us how simple things like class names or CSS variables act as “data” to make big visual changes happen. In the end, this project gave us a hands-on way to learn how to use a framework, tweak it to our needs, and see how basic programming concepts actually work in real projects.

In order to get this project completed, Linnea worked on making our wonderful code and website. Jenna and Abby both worked on the script, integration of the core concepts, as well as the Readme. All of us worked hard together to create this wonderful project and tutorial.