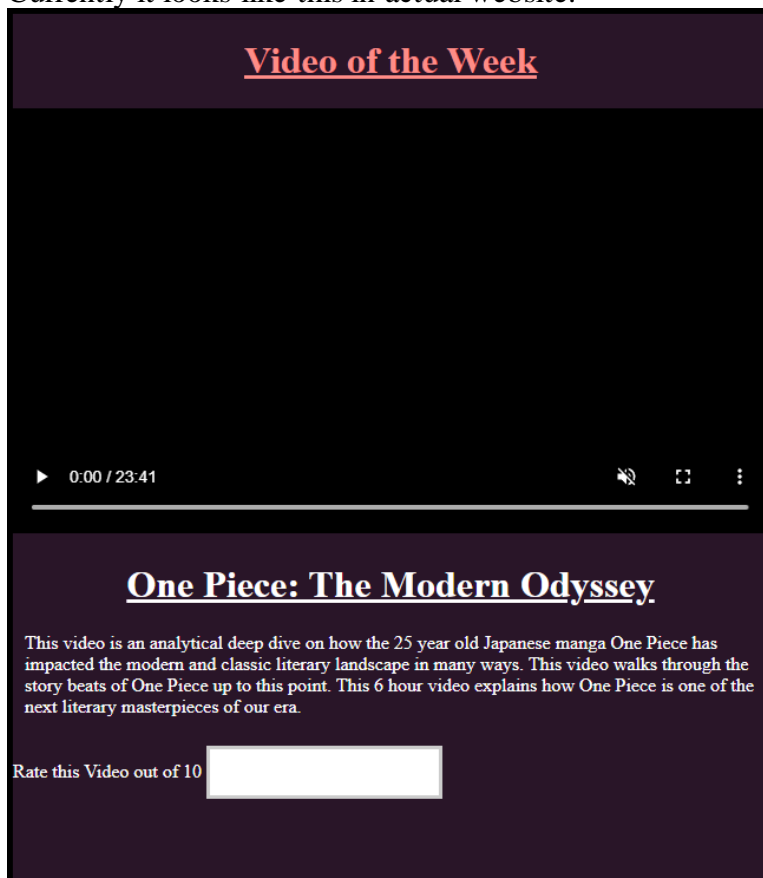


The first set of code I want to edit is the video of the week and change it into a random video player. In Assignment 2, the code looked something like this for the video player:

HTML

```
<div class="video-week">
  <h1 class="title">
    <u>
      <b>Video of the Week</b>
    </u>
  </h1>
  <video width="640" height="360" muted controls>
    <source src="file:Assets/VideoWeak2.mp4" type="video/mp4">
  </video>
```

Currently it looks like this in actual website:



Now the desired outcome is to choose a random YouTube video. Now of course it would be more difficult to get a completely random video so to make this easier it would be a random video in a curated list of videos. By storing what video to show in an iframe in an array it can access it way easier.

The brand-new code looks like this:

HTML

```
<div class="video-week">
  <h1 class="title">
    <u>
      <b>
        Random Video
      </b>
    </u>
  </h1>
  <div id="player">
  </div>
  <div>
    <iframe id="video" frameborder="0" allowfullscreen>
    </iframe>
  </div>
  <button class="center-button" id="regenerate" onclick="document.querySelector('form').reset();">
    Regenerate Video
  </button>
  <script src="Script1.js">
  </script>
```

The new code changes the video and source src to an iframe to make it easier for accessing.

JavaScript

```
const videoIds = ['ZJ0zJDz1G1M', '8FCAHU93NAK', 'rSvBFm_MuXw', 'bUfift7822g', 'Zi'];

function regenerateVideo() {
  const randomVideoId = videoIds[Math.floor(Math.random() * videoIds.length)];
  const videoUrl = `https://www.youtube.com/embed/${randomVideoId}`;

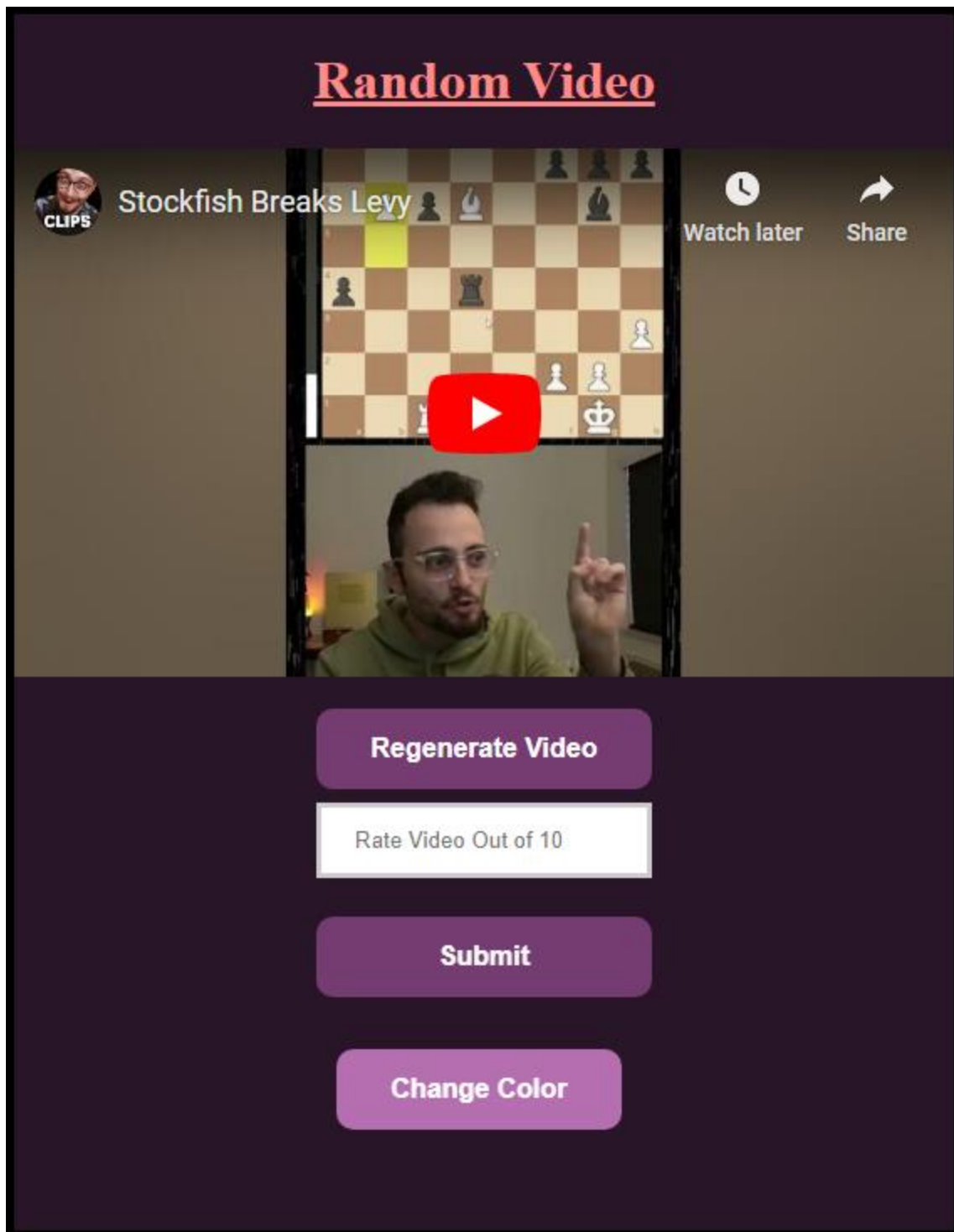
  const videoPlayer = document.getElementById('video');
  videoPlayer.src = videoUrl;
}

const regenerateButton = document.getElementById('regenerate');
regenerateButton.addEventListener('click', regenerateVideo);

regenerateVideo();
```

The VideoIds constant is an array that stores all the possible videos it can get. The function regenerateVideo() is what is called when button is pressed in HTML. With constant randomVideoId it takes the array videoId and using math.floor, math.random, and videoId.length to pick a number in the array at random and assign the value in the array to the randomVideoId. The constant regenerateButton uses the getElementById('regenerate') so that the button in the HTML has the id=regenerate will do the function present in the script. Finally using addEventListener('click', regenerateVideo); This means that on the click of buttons with id of regenerate it will give me a random video again. Another thing is that the onclick function in the button tag shown in the HTML resets the rate out of 10 form featured in the div by using the javascript line querySelector('form').reset(); to empty the rate form.

The new code helps it look like this:

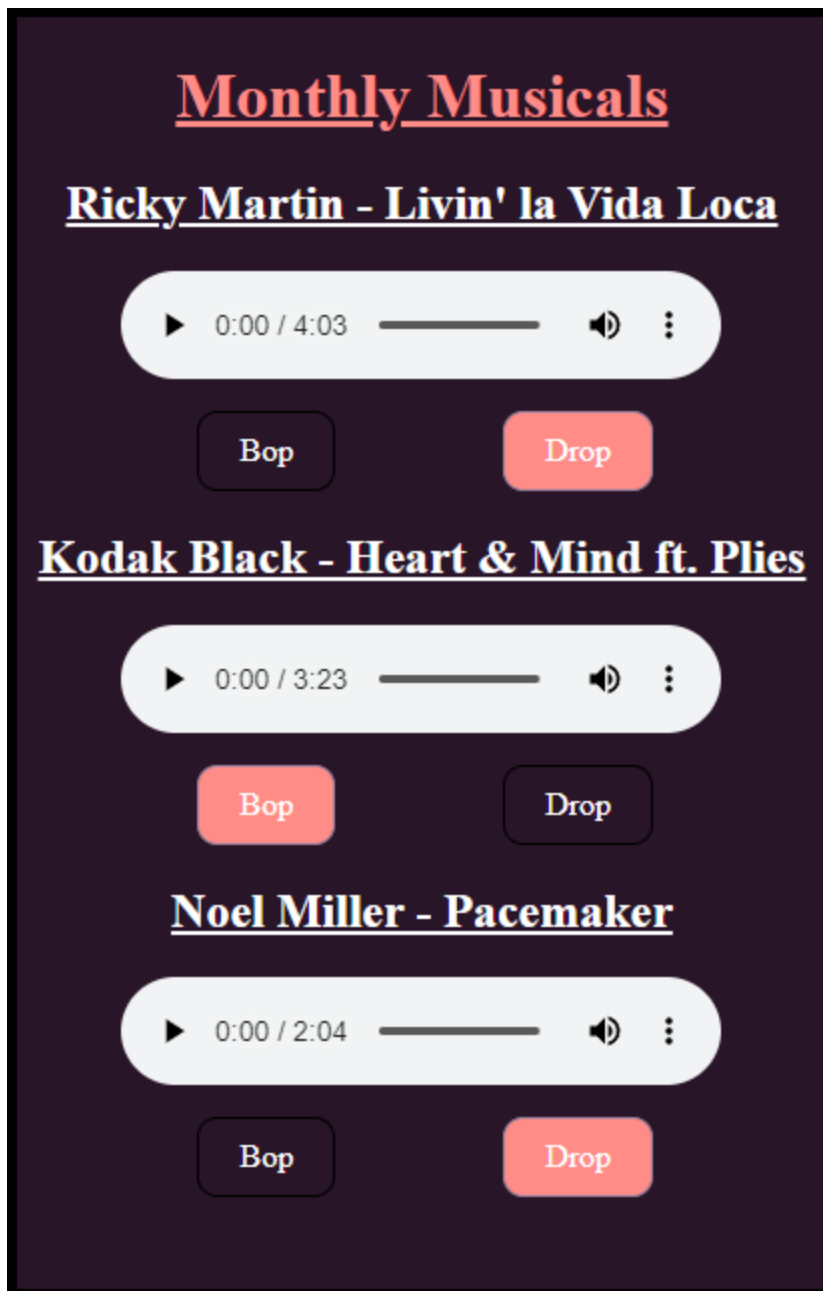


The second set of code I want to change is the monthly musicals code. In Assignment 2 the HTML code looked like this:

HTML

```
<div class="music-month">
  <h1 class="title">
    <u>
      <b>Monthly Musicals</b>
    </u>
  </h1>
  <h2 class="subtitle">
    <u>
      <b>Ricky Martin - Livin' la Vida Loca </b>
    </u>
  </h2>
  <p>
    <audio controls>
      <source src="file:Assets/Song1.mp3">
    </audio>
    <form>
      <input id="bop1" name="toggle" type="radio">
      <label class="BoporDrop" for="bop1">Bop</label>
      <input id="drop1" name="toggle" type="radio">
      <label class="BoporDrop" for="drop1">Drop</label>
    </form>
  </p>
  <h2 class="subtitle">
    <u>
      <b>Kodak Black - Heart & Mind ft. Plies</b>
    </u>
  </h2>
  <p>
    <audio controls>
      <source src="file:Assets/Song2.mp3">
    </audio>
    <form>
      <input id="bop2" name="toggle" type="radio">
      <label class="BoporDrop" for="bop2">Bop</label>
      <input id="drop2" name="toggle" type="radio">
      <label class="BoporDrop" for="drop2">Drop</label>
    </form>
  </p>
  <h2 class="subtitle">
    <u>
      <b>Noel Miller - Pacemaker</b>
    </u>
  </h2>
  <p>
    <audio controls>
      <source src="file:Assets/Song3.mp3">
    </audio>
    <form>
      <input id="bop3" name="toggle" type="radio">
      <label class="BoporDrop" for="bop3">Bop</label>
      <input id="drop3" name="toggle" type="radio">
      <label class="BoporDrop" for="drop3">Drop</label>
    </form>
  </p>
</div>
```

With this code the website looks like this:



The desired outcome is that when the a button is pressed the other button becomes darked out and that, when the bop button is pressed it will take you to the YouTube video from where the song originates.

Here is the new HTML and JavaScript to help achieve this plan:

HTML

```
<div class="music-month">
  <h1 class="title">
    <u>
      <b>
        Monthly Musicals
      </b>
    </u>
  </h1>
  <h2 class="subtitle">
    <u>
      <b>
        DM DOKURO - Roar of The Jungle Dragon
      </b>
    </u>
  </h2>
  <p>
    <audio controls>
      <source src="assets/Song1.mp3">
    </audio>
    <div class="button-container">
      <button id="bop-btn1" onclick="song1()">Bop</button>
      <button id="drop-btn1" onclick="drop1()">Drop</button>
    </div>
  </p>
  <h2 class="subtitle">
    <u>
      <b>
        Bobby Caldwell - What You Won't Do for Love
      </b>
    </u>
  </h2>
  <p>
    <audio controls>
      <source src="assets/Song2.mp3">
    </audio>
    <div class="button-container">
      <button id="bop-btn2" onclick="song2()">Bop</button>
      <button id="drop-btn2" onclick="drop2()">Drop</button>
    </div>
  </p>
  <h2 class="subtitle">
    <u>
      <b>
        nelward - Innuendo
      </b>
    </u>
  </h2>
  <p>
    <audio controls>
      <source src="assets/Song3.mp3">
    </audio>
    <div class="button-container">
      <button id="bop-btn3" onclick="song3()">Bop</button>
      <button id="drop-btn3" onclick="drop3()">Drop</button>
    </div>
  </p>
</div>
```

JavaScript

```
function song1() {
    var bop = document.getElementById("bop-btn1");
    var drop = document.getElementById("drop-btn1");
    bop.style.backgroundColor = "#FF8C87";
    bop.style.color = "white"
    drop.style.backgroundColor = "#0a050a";
    drop.style.color = "#3f3f3f"
    window.open("https://www.youtube.com/watch?v=5GcPA3C0QeI", "_blank");
}

function song2() {
    var bop = document.getElementById("bop-btn2");
    var drop = document.getElementById("drop-btn2");
    bop.style.backgroundColor = "#FF8C87";
    bop.style.color = "white"
    drop.style.backgroundColor = "#0a050a";
    drop.style.color = "#3f3f3f"
    window.open("https://www.youtube.com/watch?v=n9DmdAwUbx", "_blank");
}

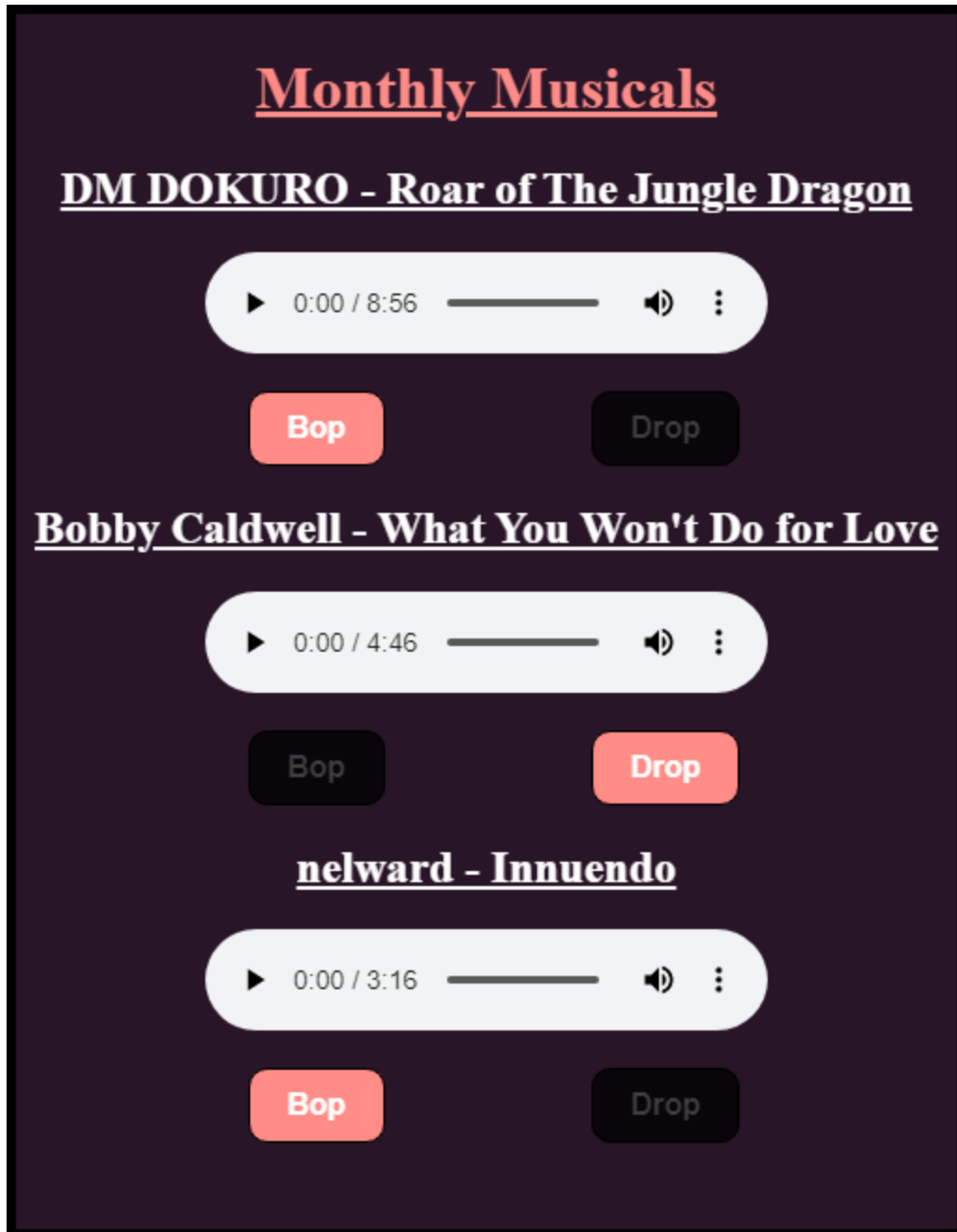
function song3() {
    var bop = document.getElementById("bop-btn3");
    var drop = document.getElementById("drop-btn3");
    bop.style.backgroundColor = "#FF8C87";
    bop.style.color = "white"
    drop.style.backgroundColor = "#0a050a";
    drop.style.color = "#3f3f3f"
    window.open("https://www.youtube.com/watch?v=jq6kaCcFE7c", "_blank");
}

function drop1() {
    var bop = document.getElementById("bop-btn1");
    var drop = document.getElementById("drop-btn1");
    drop.style.backgroundColor = "#FF8C87";
    drop.style.color = "white"
    bop.style.backgroundColor = "#0a050a";
    bop.style.color = "#3f3f3f"
}

function drop2() {
    var bop = document.getElementById("bop-btn2");
    var drop = document.getElementById("drop-btn2");
    drop.style.backgroundColor = "#FF8C87";
    drop.style.color = "white"
    bop.style.backgroundColor = "#0a050a";
    bop.style.color = "#3f3f3f"
}

function drop3() {
    var bop = document.getElementById("bop-btn3");
    var drop = document.getElementById("drop-btn3");
    drop.style.backgroundColor = "#FF8C87";
    drop.style.color = "white"
    bop.style.backgroundColor = "#0a050a";
    bop.style.color = "#3f3f3f"
}
```

The new code removes the 4-line input and label code for a far simpler 2 line button code. By using functions and ids it is possible to have all the buttons access the different functions to take to new videos depending on the song. The style. Commands from JavaScript helps change the color of buttons on click as well so it helps. Currently this is what the website now looks like:



The buttons now blacken out to create a completed button press look.