## ***Goal/Purpose of the Application***

Since the purpose of the assignment revolved around our interests, I decided that I would do a game review that would showcase some of the games that I have been playing recently. Allowing the user to interact with the website in some way was my first goal to begin with. I had to start brainstorming some interactive items that I could put onto my website, in order to draw the user in.

## ***How to Use the Application***

Since I wanted the user to be intrigued with the website, I had used different shades of pink for the color palette so it would seem inviting. As the user puts their cursor over the cards I have listed above, it will show how actively I play the game. For some added flair, if the user decides to click on the game card, the application will proceed to show the review I have left for the game. If the user scrolls further down, I had left a little note for each game. I wanted to put the personal notes, as it felt like a connection back to my life. These little notes help make this project feel more like myself.

## ***RUBRIC***

1. Interface is Authentic, Professional, Balanced Interface design is authentic, looks professional, is balanced across the web page

- For my styling, I had used the file *styling.css*. Intentionally, I had put my styling onto my HTML document, yet I found that putting everything onto a separate file gave my codes a clean and concise look.

1. Interface links to functional jQuery jQuery is included in application source

- <script src = "jquery-3.7.1.js"></script> | **this is included in *index.html***

1. Events are Effective, jQuery jQuery is used to bind events to event handlers<br/> The event handling mechanism works to achieve goals.

- $("p").eq(index).html(`<b>${games[index].game\_name}</b><br>${games[index].genre}<br><i> Release Date- ${games[index].release\_date}`);

- **This statement labels the front of the card, so the user is able to identify it easily.**

1. DOM Access, jQuery The DOM is accessed with jQuery

const showInformation = () => {

setUpBoxes();

jQuery.each($(".box"),establishGameInfo);

- **In this statement, I had access jQuery to help create some boxes for my layout, as you see with the setUpBoxes()**

**- Another great example would be the mouse rollover events that changes the contents of the card everytime the user passes over it. A border also shows around the boxes afterwards.**

$(".box").on("mouseover", function () {

$(this).css("border", "2px dotted white");

let indexGame = $(".box").index(this);

$(this).find("p").text(`Actively played?- ${games[indexGame].actively\_played}<br><i> Release Date- ${games[indexGame].release\_date}</i>`);

});

1. DOM Dynamically Built, jQuery The DOM is dynamically updated using jQuery calls

const setUpBoxes = ()=> {

let wrapperRef = $('.container');

games.forEach((games)=>{

let boxMarkup= ` <div class="box">

<img class="game\_pic" src="./images/${games.game\_name.toLowerCase()}.png" />

<p class="review"></p>

</div>`;

wrapperRef.append(boxMarkup);

**- The class ‘container’ is holding all of the boxes displayed.**

1. Style, jQuery In at least one instance, style is modified or applied using jQuery

Mouseover → $(this).css("border", "2px dotted white");

Mouseout → $(this).css("border", "2px dotted pink");

**- In these statements, the cards will display a new border as the user moves their cursor over the card, allowing the user some interactivity with the website.**