**Jacob’s Top 5 TV shows Website**

The website I decided to make was a ranking of my top five favorite TV shows that I have watched over the years. I made this because I wanted to work on a site that adheres to my own interests and would be exciting to work on. The five pages I created each had one of the TV shows I like and some brief descriptions about the shows. The main page and the second also contain tables in lists relating to the shows. For example, my first page has a table containing all of the main characters in One Piece and their roles on the pirate crew they are a part of. All of the pages contain 1-2 images from the show which are either the cover art to advertise the show or an image of a scene from the show.

**HTML elements used:**

* <!doctype html> → Declares that the document is HTML5, so browsers know how to interpret it.
* <html lang="en"> → Wraps the entire page and specifies the language for accessibility and search engines.
* <head> → Contains metadata about the page, like encoding, stylesheet, and title.  
  + <meta charset="utf-8"> → Ensures the page can handle special characters.
  + <link rel="stylesheet" href="style.css"> → Links to an external CSS file for styling.
  + <title> → Sets the title that appears in the browser tab.
* <body> → Contains all visible content on the page.
* <header> → Site-wide header with the main title (<h1>) and navigation.  
  + <nav> + <ul> + <li> + <a> → Creates a navigation menu with links to other pages.
* <main> → Holds the main content of the page.  
  + <section> → Groups introductory content into a logical block.
  + <article> → A standalone piece of content (One Piece review).  
    - <header><h2> → Introduces the article’s topic.
    - <button> → Provides interactivity for toggling the crew table and inverting colors.
    - <img> → Adds visuals of the show and characters, with alt text for accessibility.
    - <p> → Paragraphs for descriptive content and explanations.
    - <table> → Displays crew members and their roles in a clear, structured way.
* <footer> → Contains site-wide info, like copyright.

**CSS elements used:**

* body → Sets up a flexbox layout to center and stack content vertically.  
  + Removes default margins, applies a clean font (Arial), and gives a soft background color for readability.
* header → Given a gradient background, padding, rounded corners, and a subtle shadow to stand out as the page’s banner.
* header h1 → Larger font size with text-shadow for emphasis.  
  + Makes the header look polished and eye-catching.
* nav ul → Removes bullets, turns the list into a horizontal flex menu, and uses gap for spacing.
* nav ul li a → Styled as buttons with padding, rounded corners, background transparency, and hover effects (color + scale transform).  
  + Improves usability and interactivity of navigation links.
* main → Keeps the content centered and limited in width for readability.
* Uses flexbox to stack section and article elements cleanly.
* section → Styles introductory text with increased line spacing and centered alignment.  
  + Makes the intro easy to read and welcoming.
* article → White semi-transparent background with padding, rounded corners, and shadow.
* Hover effect (translateY) adds subtle interactivity.
* Centers content and ensures consistency in width.
* article img → Scales images responsively (max-width: 100%), centers them, adds rounded corners and shadows.
* Hover effect slightly enlarges the image with a stronger shadow and border, making images interactive and modern-looking.
* table → Full width, centered, with collapsed borders and rounded edges.
* th cells → Highlighted with a themed background (#ff6f61).
* footer → Similar gradient to the header for visual consistency.
* @media (max-width: 600px) → Adjusts padding, font sizes, and spacing for mobile screens.  
  + Ensures the site remains readable and user-friendly on small devices.
* ol, ul → Centers list items, removes default padding, and aligns bullets/numbers inside.
* li → Adds spacing between items for better readability.  
  + Keeps lists neat and aligned with the site’s centered layout.

**JavaScript code used:**

**Functions:**

* toggleCrewTable() → Shows or hides the crew table.
* invertColors() → Inverts the page’s colors for fun or accessibility.

**Event Listeners:**

* Attached to the buttons to make the page interactive when clicked.

**CSS validation link** -> https://jigsaw.w3.org/css-validator/validator$link

**HTML validation link** -> Validator.nu/LV https://validator.w3.org/services