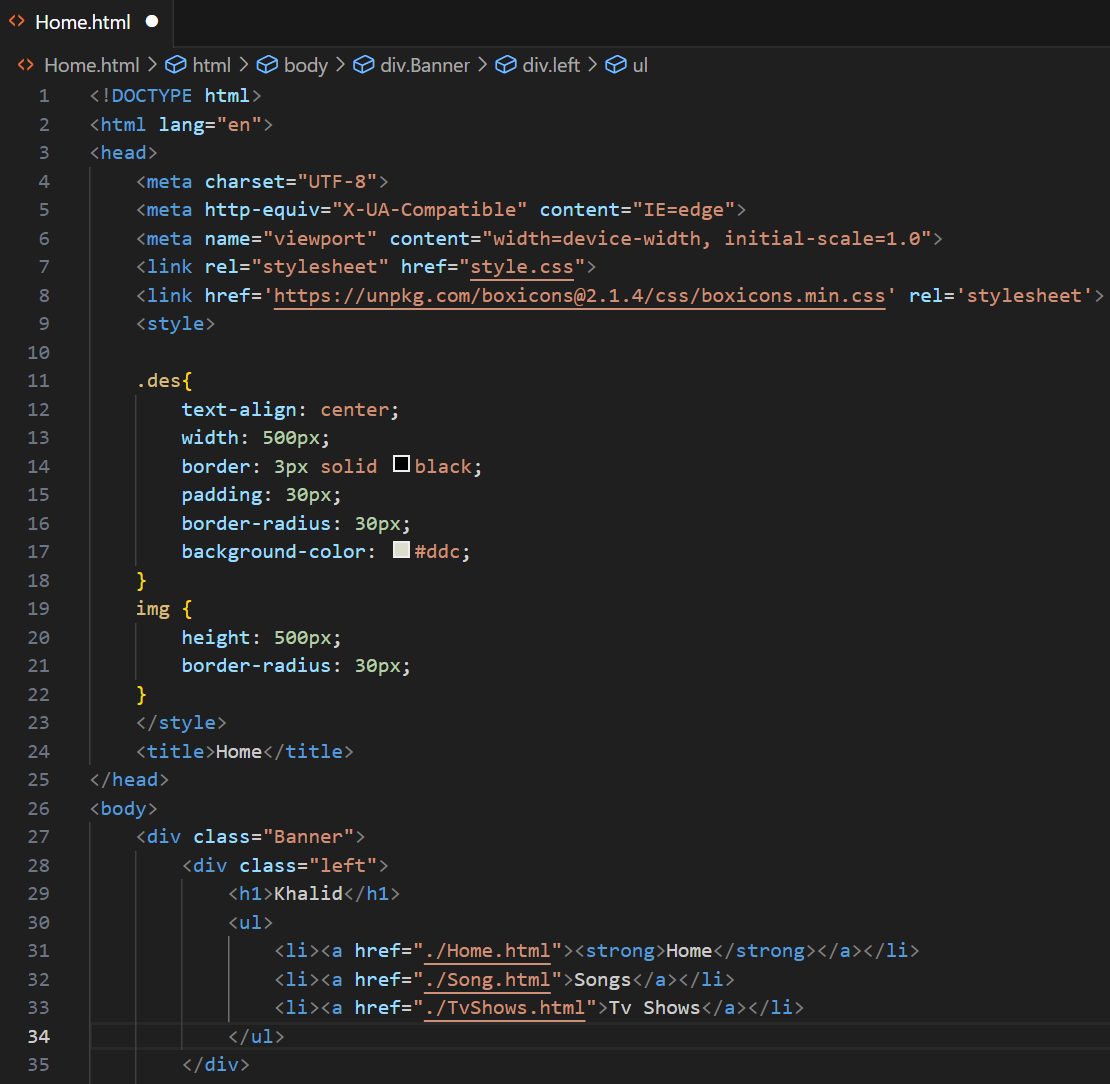
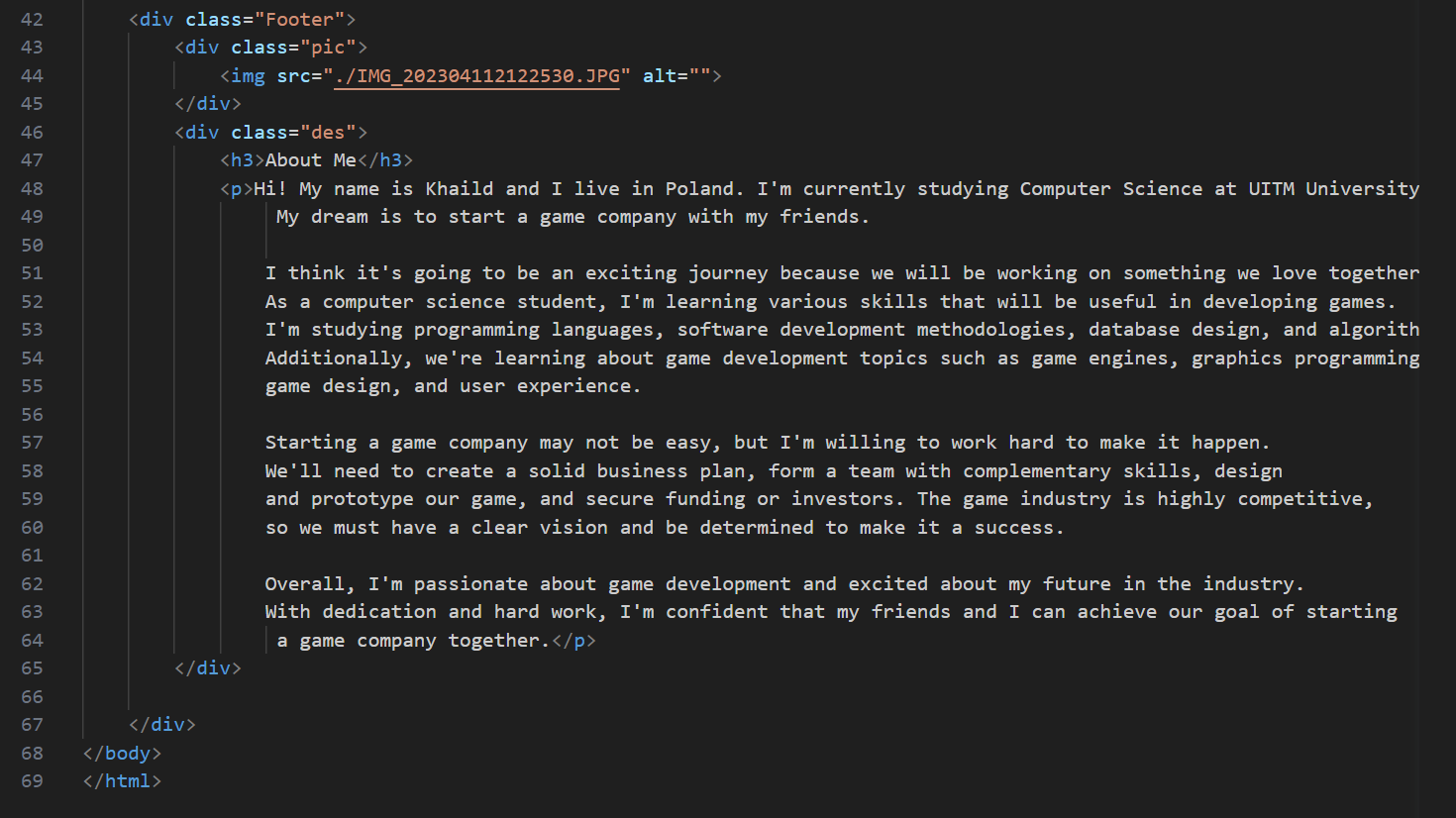
**Website (Khalid Katamato, W68907)**

Home.HTML:





I started with the `<!DOCTYPE html>` declaration, which specifies the HTML version being used.

The `<html>` tag encloses the entire HTML document.

The `<head>` section contains meta information and links to external resources.

The `<style>` section contains CSS rules that define the appearance of elements in your HTML.

The `.des` class selector styles a `<div>` element with text alignment, width, border, padding, border radius, and background color properties.

The `img` selector styles all `<img>` elements with a fixed height and border radius.

The `<title>` tag sets the title of the webpage to "Home."

The `<body>` section contains the content of the webpage.

The content is divided into two main sections: `<div class="Banner">` and `<div class="Footer">`.

The `<div class="Banner">` section represents the top banner of the webpage.

- It contains two inner `<div>` elements with classes "left" and "right".

The "left" section includes an `<h1>` heading with the text "Khalid" and an unordered list (`<ul>`) with three list items (`<li>`). Each list item is a hyperlink (`<a>`) with a different text.

The "right" section includes three hyperlinks to different social media platforms, each represented by an `<i>` element with a specific class from the "boxicons" icon library.

The `<div class="Footer">` section represents the footer of the webpage.

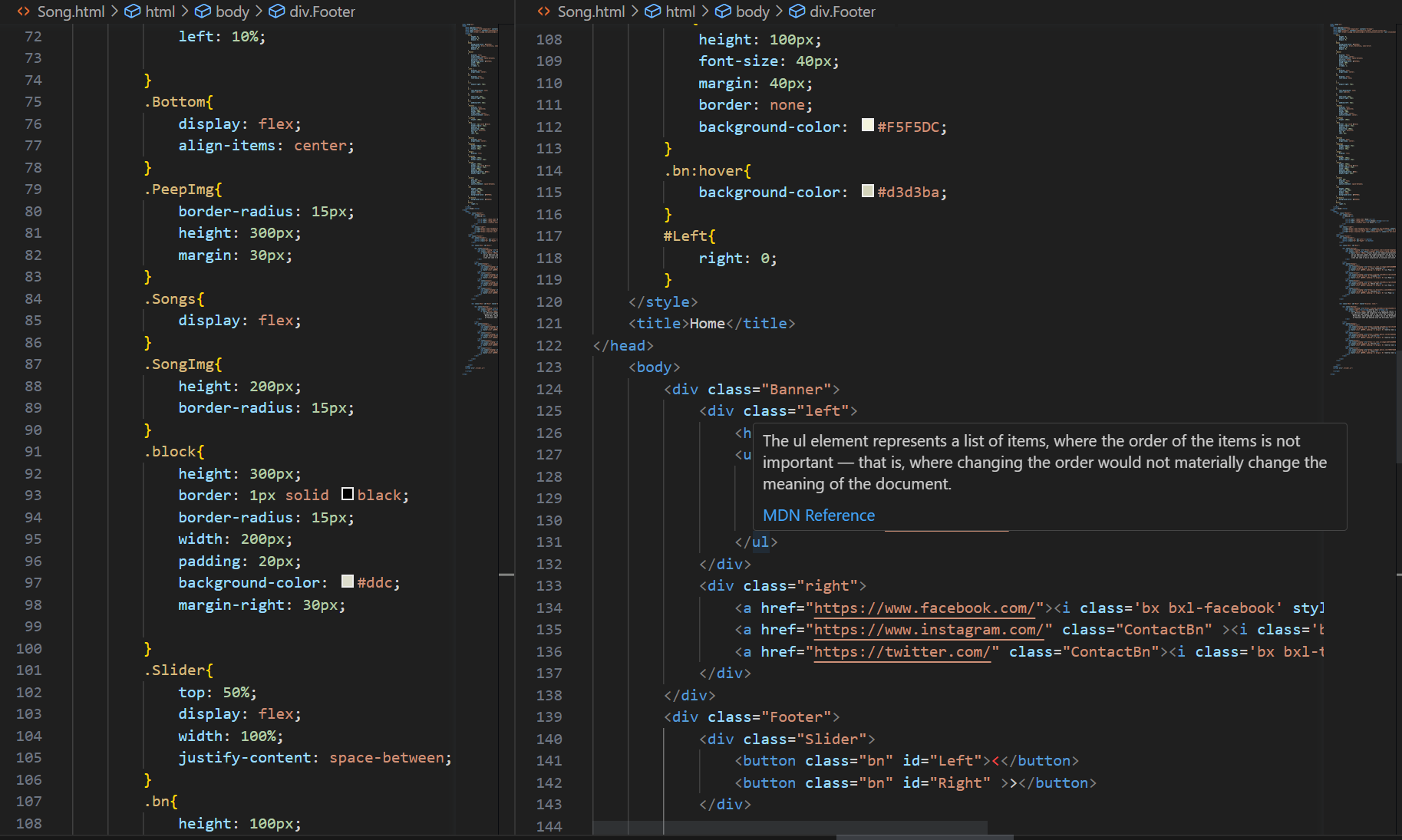
- It contains two inner `<div>` elements with classes "pic" and "des".

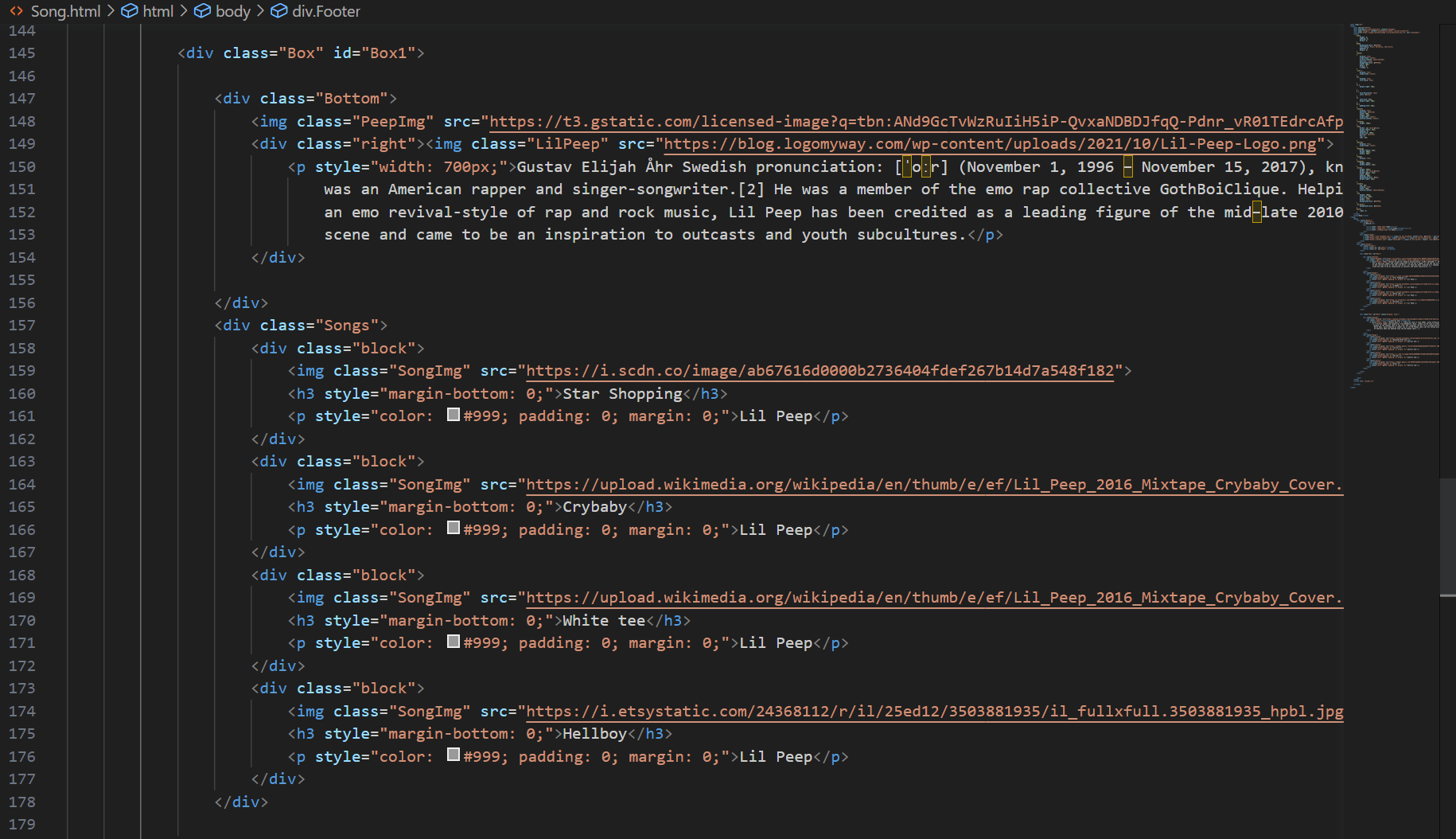
The "pic" section includes an `<img>` element that displays an image specified by the `src` attribute.

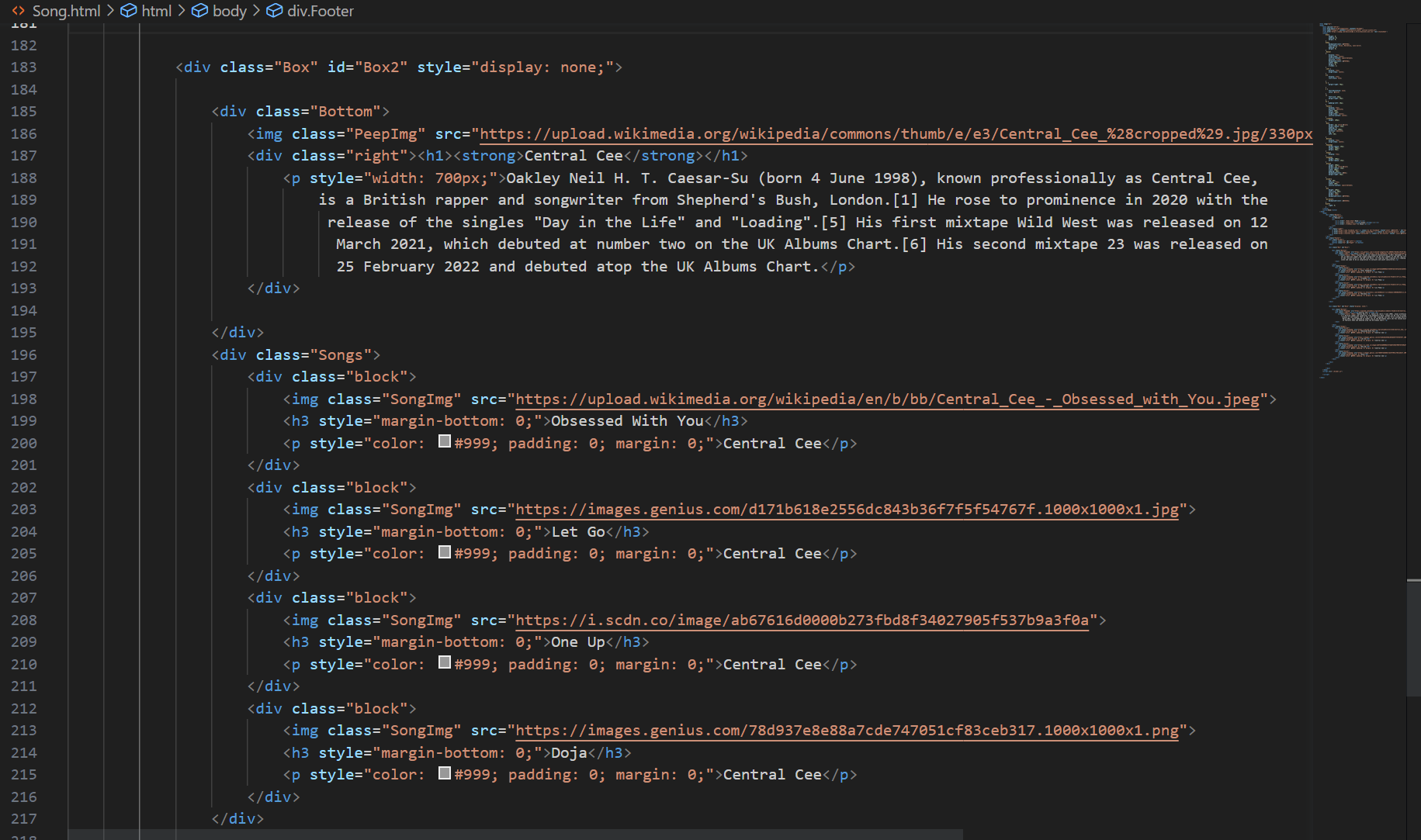
The "des" section includes an `<h3>` heading and a `<p>` paragraph with text describing myself and my aspirations in the field of game development.

Song.HTML:









1. I Set the language of the document to English using the `<html>` tag attribute `lang="en"`.

2.I Added a `<head>` section to include metadata and the document title.

3. and I Included some external CSS stylesheets from the internet using the `<link>` tag with the `href` attribute pointing to the CSS file.

4.I Added internal CSS styles within the `<style>` tags to define the appearance of different elements in the document.

5.I Created the overall structure of the webpage in the `<body>` section.

6.I Created a banner at the top of the page using the `<div>` element with the class "Banner". It contains two child `<div>` elements with the classes "left" and "right".

7. Inside the "left" `<div>`, I added a heading (`<h1>`) with the text "Khalid" and an unordered list (`<ul>`) with three list items (`<li>`), each containing a link (`<a>`) to different HTML pages.

8. Inside the "right" `<div>`, I added three social media links represented by icons (`<i>`) wrapped in anchor tags (`<a>`) with their respective URLs.

9.I Created a footer section at the bottom of the page using the `<div>` element with the class "Footer".

10. Inside the footer, I added a slider section (`<div>` with class "Slider") containing two buttons (`<button>`) with classes "bn" and IDs "Left" and "Right".

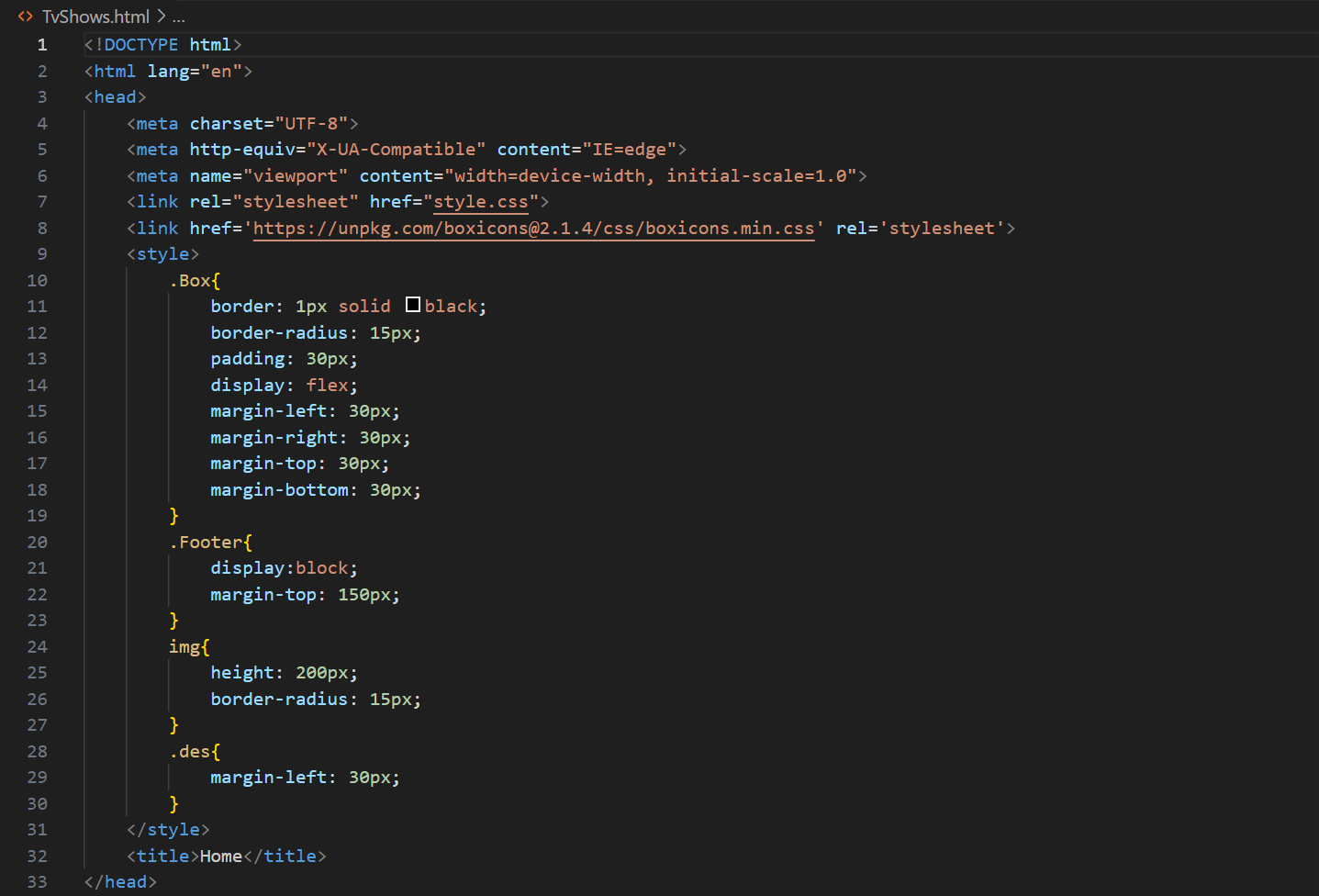
11. Below the slider, I added two `<div>` elements with the class "Box" and IDs "Box1" and "Box2". The "Box1" is initially visible, and "Box2" is hidden (using the `style="display: none;"`).

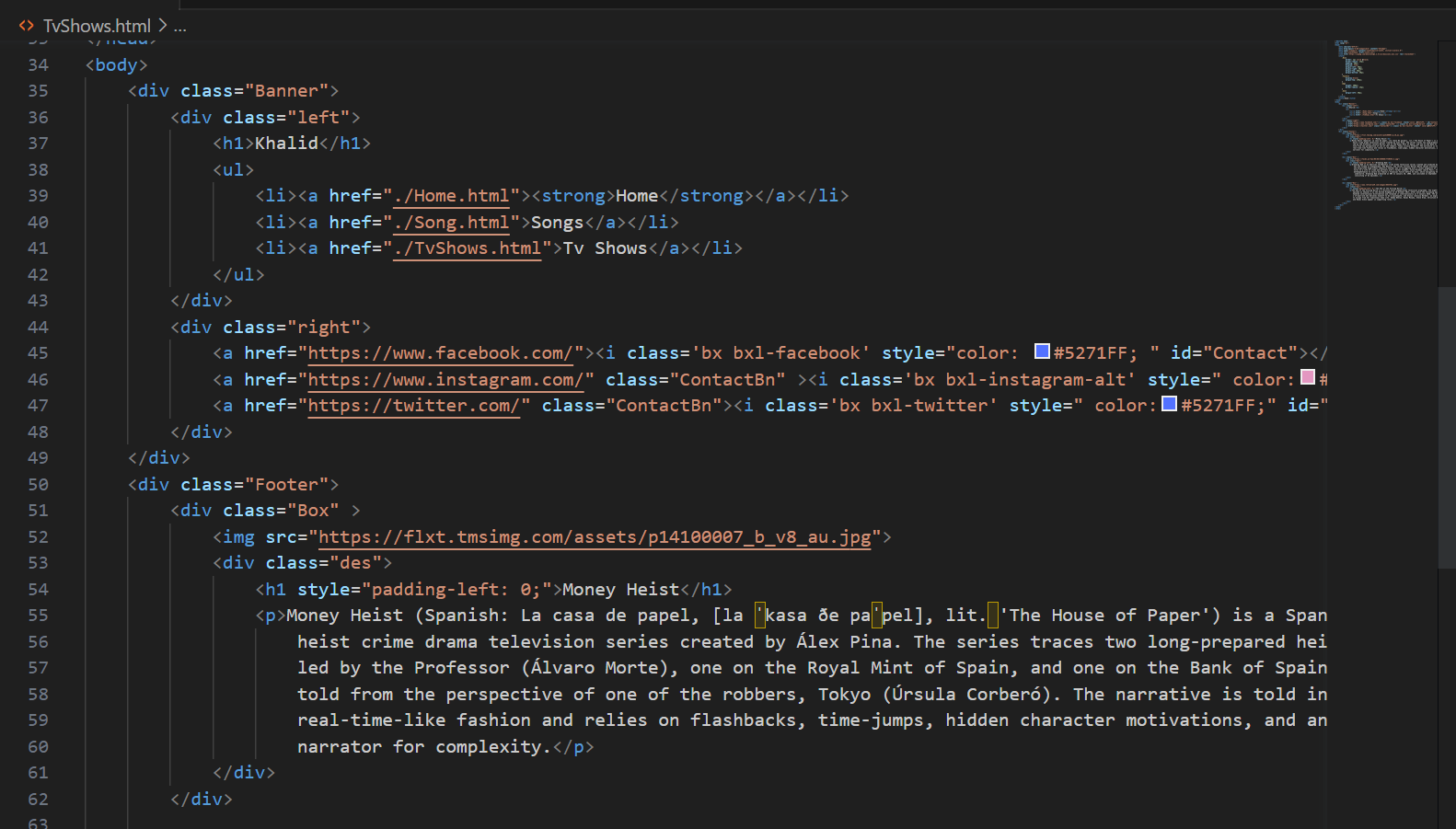
12. Inside "Box1" and "Box2", I added a content section (`<div>`) with the class "Bottom" containing an image (`<img>`) and a `<div>` with class "right" that includes text content.

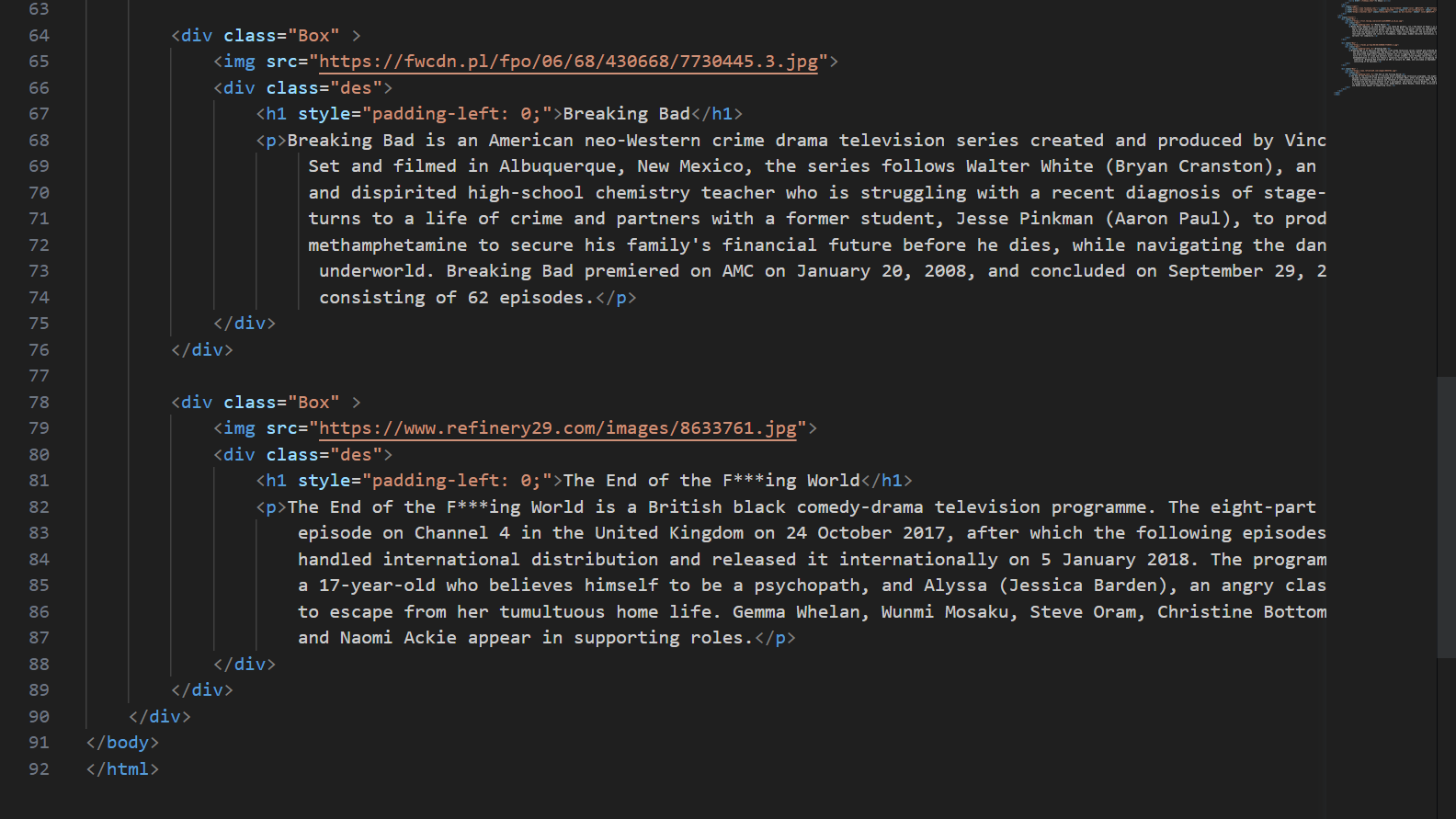
13. Inside each content section, I added a song section (`<div>`) with class "Songs". Each song is represented by a block (`<div>`) with the class "block" containing an image, song title, and artist name.

14. Finally, I included a JavaScript file (`<script>`) with the source attribute pointing to "./Slider.js".

TvShows.HTML:







1. I started the code with the HTML doctype declaration `<!DOCTYPE html>` indicating that the document is an HTML5 document.

2. The `<html>` element is the root element of the HTML document and contains all other elements.

3. The `<head>` section contains meta information and the title of the webpage.

4. Within the `<head>` section, there are several `<meta>` tags that specify the character encoding, compatibility, and viewport settings for the webpage.

5. The `<link>` tag is used to include external CSS stylesheets. In this code, I have linked a stylesheet named `style.css`.

6. Another `<link>` tag is used to include an external CSS library called Boxicons.

7. The `<style>` section contains CSS code for defining the styles of various elements in the webpage.

8. The `<body>` section contains the visible content of the webpage.

9. Inside the `<body>`, there is a `<div>` element with a class of "Banner" that represents the header section of the webpage.

10. Within the "Banner" section, there are two `<div>` elements, one with a class of "left" and another with a class of "right". These divisions create a two-column layout.

11. The "left" division contains an `<h1>` heading with the text "Khalid" and an unordered list `<ul>` with three list items `<li>`. Each list item contains an anchor `<a>` element that links to different HTML pages.

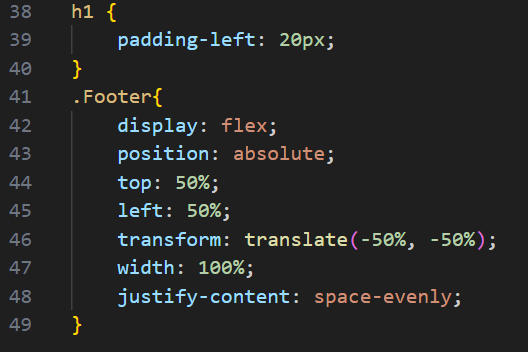
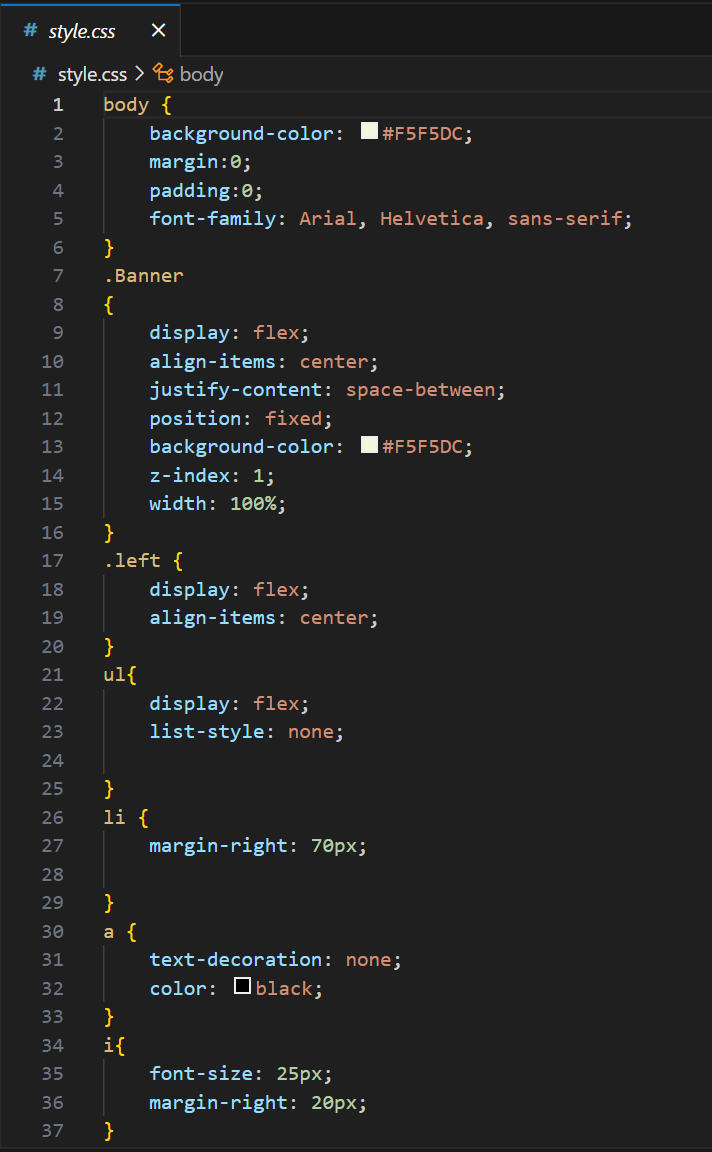
12. The "right" division contains three social media icons represented by anchor `<a>` elements. Each anchor element has an `<i>` element with a specific Boxicon class.

13. After the "Banner" section, there is a `<div>` element with a class of "Footer" that represents the footer section of the webpage.

14. Inside the "Footer" section, there are three `<div>` elements with a class of "Box". Each "Box" represents a box containing an image and a description of a TV show.

15. Each "Box" contains an `<img>` element with a source URL pointing to an image of a TV show poster and a `<div>` element with a class of "des" that contains a heading `<h1>` and a paragraph `<p>` describing the TV show.

Style.CSS:



1. The `body` selector sets the background color, margin, padding, and font family for the entire body of the page.

2. The `.Banner` selector defines the styles for a banner element. It uses flexbox (`display: flex`) to align its content vertically (`align-items: center`) and distribute the space between its child elements evenly (`justify-content: space-between`). The banner is positioned fixed (`position: fixed`) to stay at the top of the viewport. It has a background color of `#F5F5DC`, a z-index of 1 (to ensure it appears above other elements), and a width of 100%.

3. The `.left` selector is nested within the `.Banner` selector and defines the styles for a left-aligned element inside the banner. It also uses flexbox (`display: flex`) to align its content vertically.

4. The `ul` selector defines the styles for an unordered list. It uses flexbox (`display: flex`) to align its list items horizontally.

5. The `li` selector defines the styles for list items. It sets a right margin of 70 pixels.

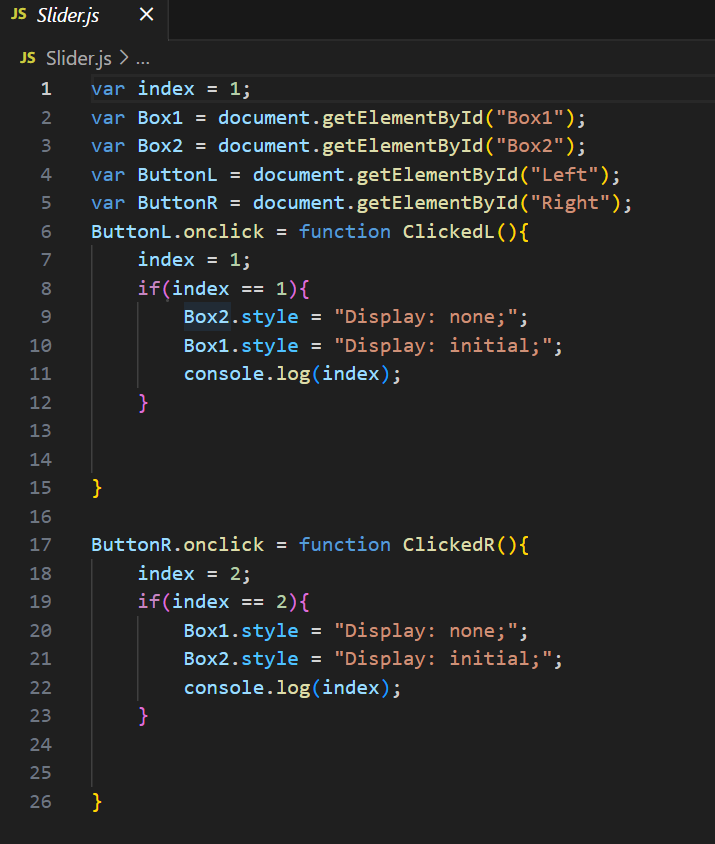
6. The `a` selector defines the styles for links. It removes the default underline (`text-decoration: none`) and sets the text color to black.

7. The `i` selector defines the styles for icons. It sets the font size to 25 pixels and adds a right margin of 20 pixels.

8. The `h1` selector defines the styles for heading elements. It adds padding to the left side (20 pixels).

9. The `.Footer` selector defines the styles for a footer element. It uses flexbox (`display: flex`) to horizontally align its content. The footer is positioned absolutely (`position: absolute`) and centered vertically and horizontally using `transform: translate(-50%, -50%)`. It has a width of 100% and evenly distributes the space between its child elements (`justify-content: space-evenly`).

Slider.JS:



1. `var index = 1;`: Initializes a variable `index` and assigns it the value 1.

2. `var Box1 = document.getElementById("Box1");`: Retrieves the HTML element with the ID "Box1" and assigns it to the variable `Box1`.

3. `var Box2 = document.getElementById("Box2");`: Retrieves the HTML element with the ID "Box2" and assigns it to the variable `Box2`.

4. `var ButtonL = document.getElementById("Left");`: Retrieves the HTML element with the ID "Left" and assigns it to the variable `ButtonL`.

5. `var ButtonR = document.getElementById("Right");`: Retrieves the HTML element with the ID "Right" and assigns it to the variable `ButtonR`.

When the `ButtonL` element is clicked, the `ClickedL` function is executed. Inside the function, the `index` variable is set to 1. Then, there is a condition `if(index == 1)` that checks if `index` is equal to 1. If it is, it sets the `display` style property of `Box2` to "none" (hiding it) and the `display` style property of `Box1` to "initial" (showing it). Finally, it logs the value of `index` to the console.

When the `ButtonR` element is clicked, the `ClickedR` function is executed. Inside the function, the `index` variable is set to 2. Then, there is a condition `if(index == 2)` that checks if `index` is equal to 2. If it is, it sets the `display` style property of `Box1` to "none" (hiding it) and the `display` style property of `Box2` to "initial" (showing it). Finally, it logs the value of `index` to the console.

In summary, the code sets up two buttons (`ButtonL` and `ButtonR`) and two boxes (`Box1` and `Box2`). Clicking `ButtonL` sets the `index` variable to 1 and shows `Box1` while hiding `Box2`. Clicking `ButtonR` sets the `index` variable to 2 and shows `Box2` while hiding `Box1`.