 A screenshot of a computer program

Description automatically generated

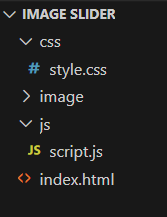
I’m currently working on a University assignment involving building a simple image slider using basic HTML, CSS and JavaScript.

This Slider allows the user to navigate between slides using radio or Arrow buttons manually. It also provides an automatic slide show functionality that transitions between slides after a specified time.

First things first, go ahead and set up a new folder. I’m gonna call it “IMAGE SLIDER” but you can call yours whatever you want. Next add 3 more folders to that one called “js”, “css”, and “image”.

Go ahead and open that folder in the text editor of your choice, I use Visual Studio and make a new file called index.html at the root of your project. next, create a “style.css” file in the css folder and a “script.js” file in the js folder.

Now, you should have a project that looks something like this:



Now, go ahead and add your images to the image folder, it doesn’t matter how many, what size, or what format, but the naming is very important.

let’s dive into the HTML structure of this simple slider:

This HTML code creates a basic image slider with navigation arrows and radio buttons for controlling the slides. The associated CSS and JavaScript files are referenced to provide the styling and interactivity of the slider.

1. The document structure:

The “<html>” tag is the root element.

The “<head>” section contains meta information and external CSS and font-awesome library references.

The “<body>” section contains the content of the webpage.

1. Image Slider:

The image slider is created using a “<div>” element with the class "slider".

It contains a set of radio buttons (<input type="radio">) to control the slides.

Each radio button has a unique ID (“slide1”, “slide2”, etc.) and is associated with a specific slide.

1. Slides:

The slides are wrapped in a “<div>” element with the class "slides".

Each slide is represented by a “<div>” element with the class "slide".

Inside each slide, an “<img>” tag is used to display an image. The “src” attribute specifies the path to the image file, and the “alt” attribute provides alternative text for the image.

1. Navigation:

Navigation arrows are included as “<a>” tags with the classes "left" and "right".

The left arrow contains a font-awesome icon for the left arrow.

The right arrow contains a font-awesome icon for the right arrow.

1. Radio Buttons:

The radio buttons are represented by “<label>” tags with the class "radbtn".

Each “<label>” tag is associated with a specific radio button using the “for” attribute.

1. JavaScript:

The script tag “<script src="./js/script.js"></script>” references an external JavaScript file named "script.js". This file likely contains the logic for controlling the image slider functionality.

Next, we have the CSS, which is central to this front-end project:

This CSS code provides the necessary styles to create an attractive and functional image slider with navigation arrows and radio buttons for controlling the slides.

1. Styling for radio buttons:

The “input[type="radio"]” selector hides the radio buttons using “display: none;”.

1. Styling for the slider container:

The “.slider” class sets the width and height to 100% of the viewport (100vh).

The background color is set to “#1f242d” (a dark shade of blue).

The text color is set to white (#fff).

1. Styling for the slides container:

The “.slider .slides” class sets the slides container to a relative position and sets its width and height to 100%.

The container uses flexbox (display: flex;) for slide alignment.

The “overflow” property is set to hidden to hide any overflow content.

1. Styling for individual slides:

The “.slider .slides .slide” class sets each slide to an absolute position within the container.

The width and height are set to 100%.

The initial opacity is set to 0 to hide the slides.

The “transition” property is used to create a smooth transition effect with a duration of 0.5 seconds.

1. Styling for active slide:

The “.slider .slide .sli” class sets the opacity of the active slide to 1, making it visible.

1. Styling for slide overlay:

The “.slider .slides .slide::after” pseudo-element creates an overlay effect.

It is positioned absolutely at the top-left corner, covering the entire slide.

The background color is set to black (#000).

The opacity is set to 0.1 to create a semi-transparent effect.

The “z-index” property is set to 1 to ensure it appears below the slide content.

1. Styling for slide images:

The “.slider .slide img” class sets the minimum width and height of the slide images to 70%.

The “object-fit” property is set to cover to ensure the image fills the container while maintaining its aspect ratio.

The “transform” and “transition” properties are used for scaling and creating a smooth transition effect.

The “margin” property is set to create spacing around the images.

1. Styling for navigation arrows:

The “.slider .arrow” class styles the navigation arrows.

The position is set to absolute, and the “top” and “transform” properties are used for vertical centering.

The initial opacity is set to 0.5, and the “transition” property is used to create a smooth transition effect.

The “:hover” pseudo-class is used to change the opacity to 1 on hover.

The font-awesome icons are styled using the “.slider .arrow I” selector.

1. Styling for left and right arrows:

The “.slider .left” and “.slider .right” classes position the arrows on the left and right sides, respectively.

1. Styling for radio buttons:

The “.slider .radio” class positions the radio buttons at the bottom of the slider container.

It uses flexbox for centering (display: flex; justify-content: center; align-items: center;).

The “.slider .radio .radbtn” class styles the individual radio buttons.

The width and height are set to create circular buttons with a white border.

The “margin” property creates spacing between the buttons.

The “cursor” property sets the cursor to a pointer on hover.

The “transition” property creates a smooth transition effect on hover.

The “.slider .radio .radbtn:hover” class changes the background color and border radius on hover.

The “.active” class is used to highlight the active radio button by changing its background color.

Lastly, we will look at the JavaScript. The JavaScript code you provided is responsible for the functionality of the image slider.

This JavaScript code provides the necessary functionality to navigate between slides manually using radio buttons or arrow buttons. It also includes an automatic slide transition feature.

1. Variable Declarations:

The “slide” variable stores an array of all slide elements.

The “radiobtn” variable stores an array of all radio button elements.

The “leftArrow” variable stores a reference to the left arrow element.

The “rightArrow” variable stores a reference to the right arrow element.

The “slideInt” variable stores the interval ID for the slide interval.

The “intTime” variable stores the duration of each slide in milliseconds.

1. Radio Button Event Listeners:

The code iterates through each radio button using the “forEach” method.

An event listener is added to each radio button, triggering the “manButtonNav” function when clicked.

The “manButtonNav” function is responsible for handling the navigation when a radio button is clicked.

1. Arrow Button Event Listeners:

Event listeners are added to the left and right arrow buttons.

When clicked, the “nextSlide” or “prevSlide” function is called, respectively.

The “clrInterval” function is also called to clear the slide interval and restart it.

1. “manButtonNav” Function:

This function handles the navigation when a radio button is clicked.

It iterates through all the slides and radio buttons.

If the current index does not match the clicked index, the corresponding slide and radio button are removed from the "active" state.

If the current index matches the clicked index, the corresponding slide and radio button are added to the "active" state.

The “clrInterval” function is called to clear the slide interval and restart it.

1. “nextSlide” Function:

This function handles the transition to the next slide.

It gets the current slide and active radio button using “querySelector”.

The current slide and active radio button are removed from the “sli” and "active" states.

If there is a next sibling slide, if so, it adds the appropriate classes to the next slide and radio buttons.

If there is no next sibling slide, it selects the first slide and radio button to create a loop effects.

1. “prevSlide” Function:

This function handles the transition to the previous slide.

It follows a similar logic to the nextSlide function but handles the previous sibling slide instead.

1. “clrInterval” Function:

This function clears the slide interval using “clearInterval”.

It is called before navigating to the next or previous slide to ensure smooth manual navigation.

After clearing the interval, it sets a new interval using “setInterval” and the “nextSlide” function.

1. Initial Slide Interval:

The slide interval is initially set using “setInterval” and the “nextSlide” function.

This starts the automatic transition between slides based on the specified “intTime” duration.