** PİRİ REİS UNIVERSİTY**

**ADVANCED WEB DESİGN ASSİGNMENT**

**GROUP MEMBERS:**

**Yağmur Sude YOLCU 20200305036**

**Sinan PATAN 20200305012**

**Salim Can YILDIZ 20200305038**

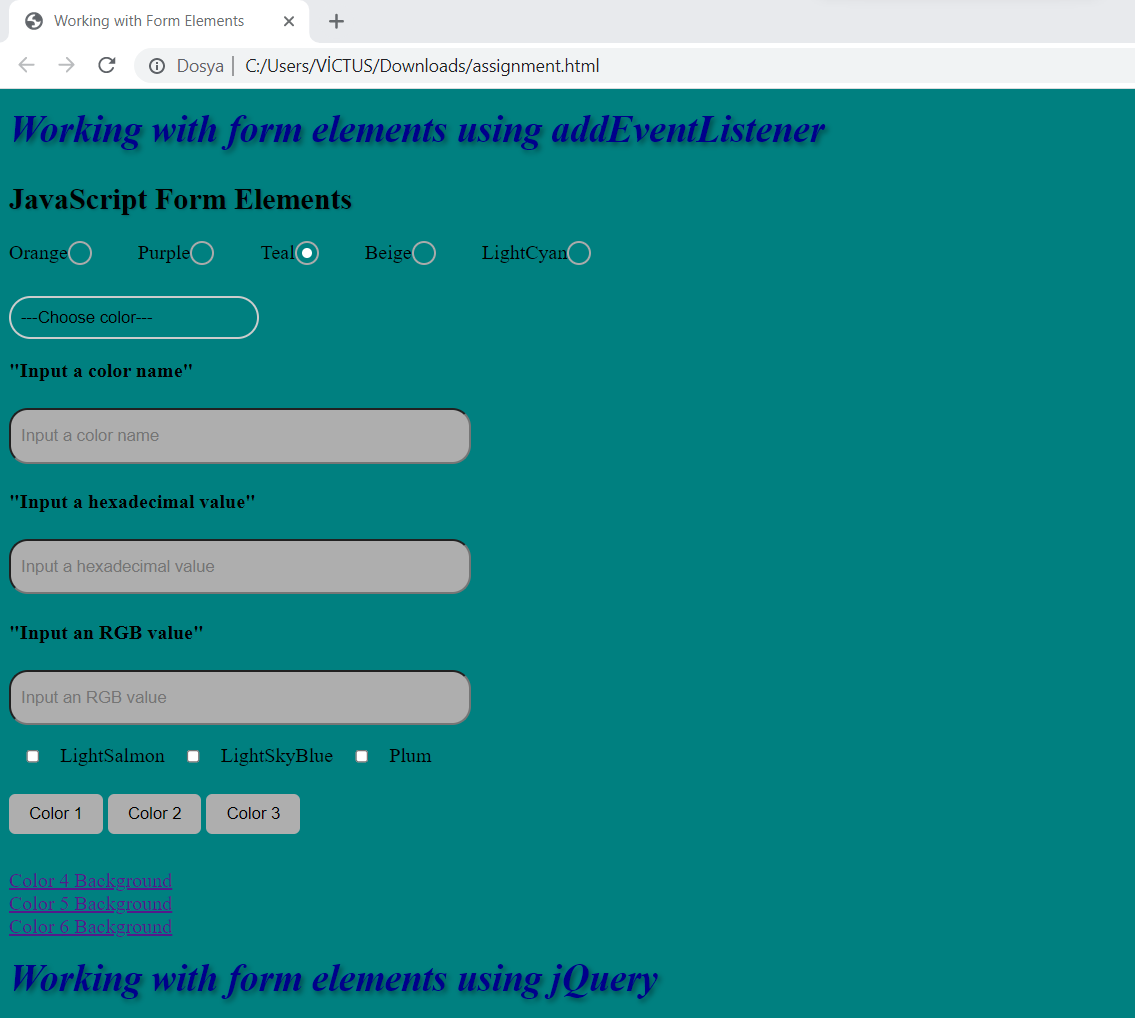
**TOPİC:** **WORK WİTH FORM ELEMENTS USİNG EVENT HANDLERS AND JQUERY.**

**INSTRUCTOR: DR. VOLKAN USLAN**

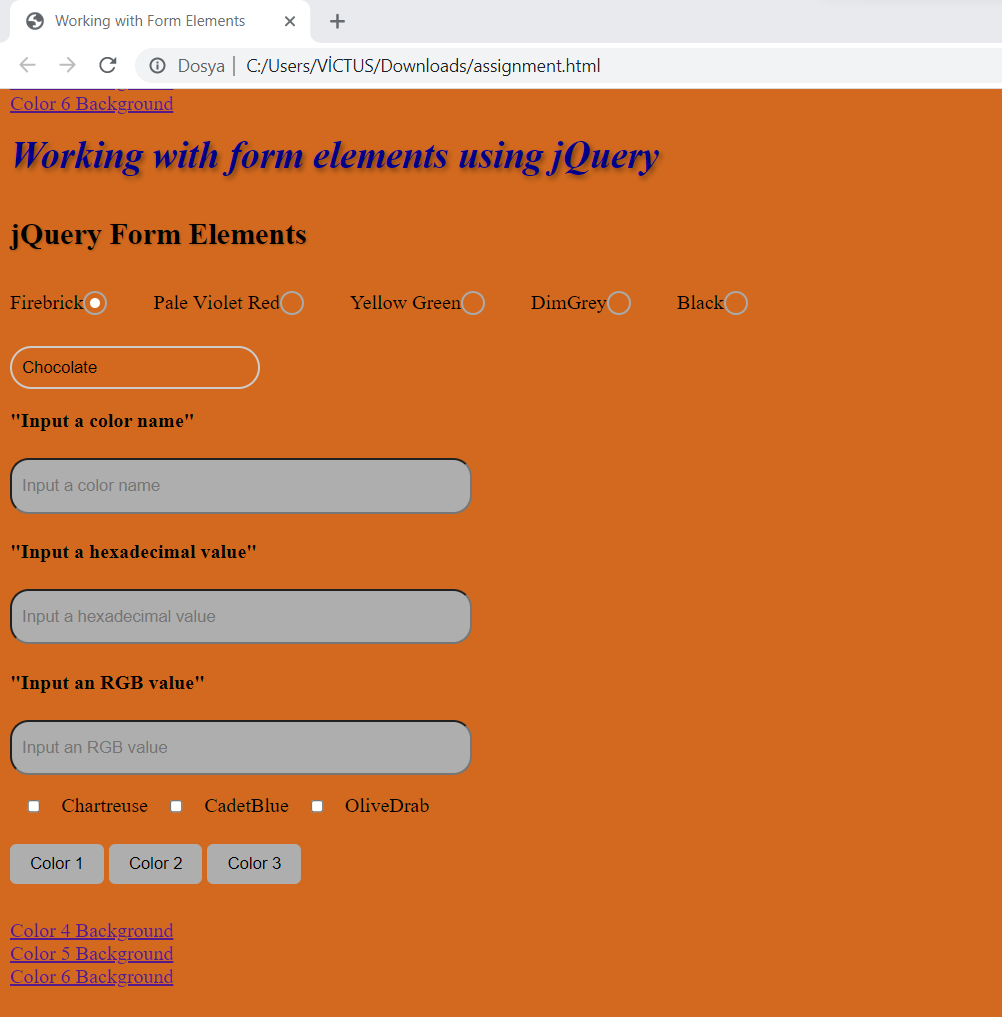
**Our HTML codes demonstrate a web page that allows users to interact with form elements to change the background color dynamically.**

* **The code includes two forms: one for JavaScript form elements and the other for jQuery form elements. Each form contains various input elements such as radio buttons, checkboxes, text inputs, and select dropdowns for selecting colors etc.**
* **The JavaScript part of the code utilizes the addEventListener() function to handle events and change the background color dynamically.**
* **The jQuery part of the code uses event handlers to achieve the same functionality as the JavaScript code, but using jQuery syntax.**
* **The code defines a changeColor() function that can be called to change the background color directly by passing a color value. Both the JavaScript and jQuery parts include event listeners for buttons with specific data-color attributes, allowing the user to select predefined colors.**
* **The code also includes CSS styling for the form elements, including custom radio buttons and checkboxes, as well as the overall appearance of the web page.**
* **The background color changes are applied to the <body> element, resulting in an immediate visual update.**

**In summary, the provided code creates an interactive web page where users can select colors from form elements or predefined buttons to dynamically change the background color.**

****

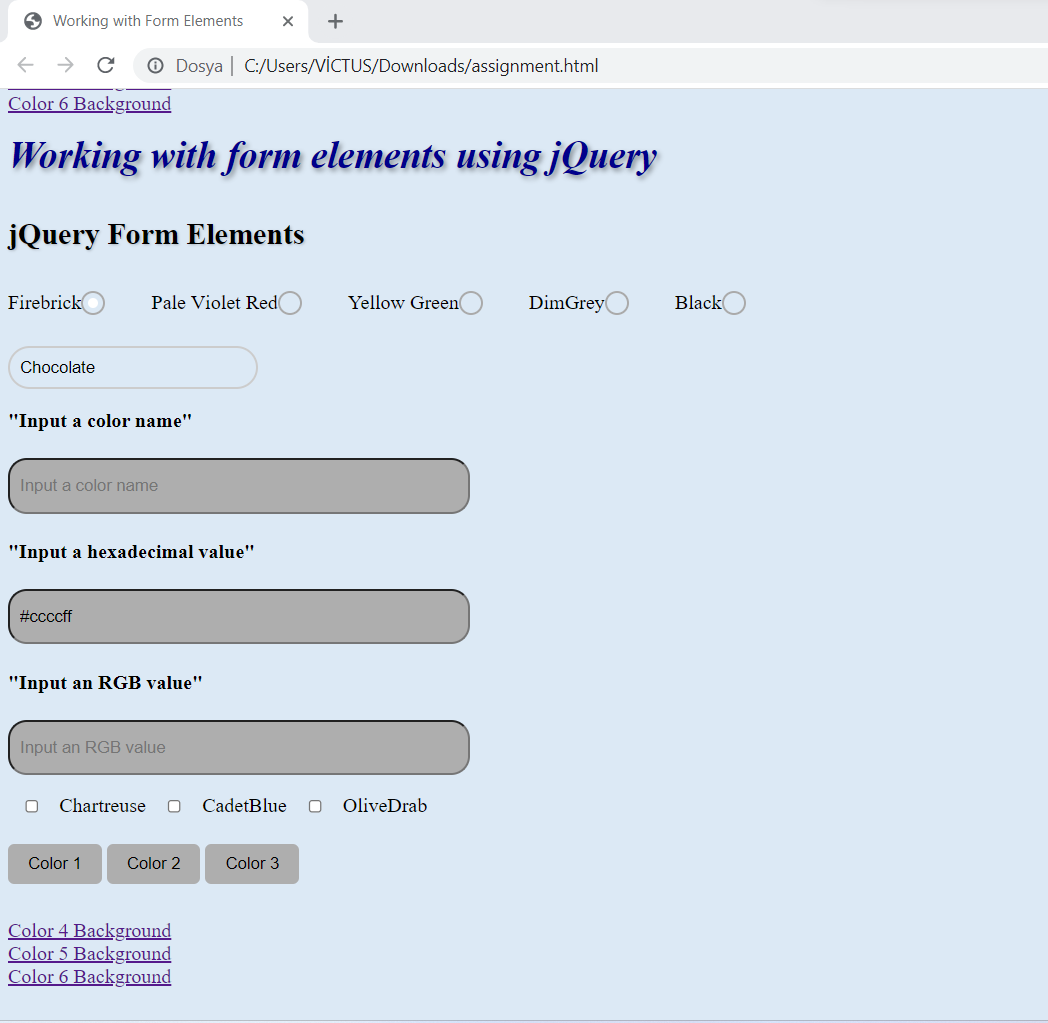
**As you see, when we click Teal radio button, screen displays this color.**

****

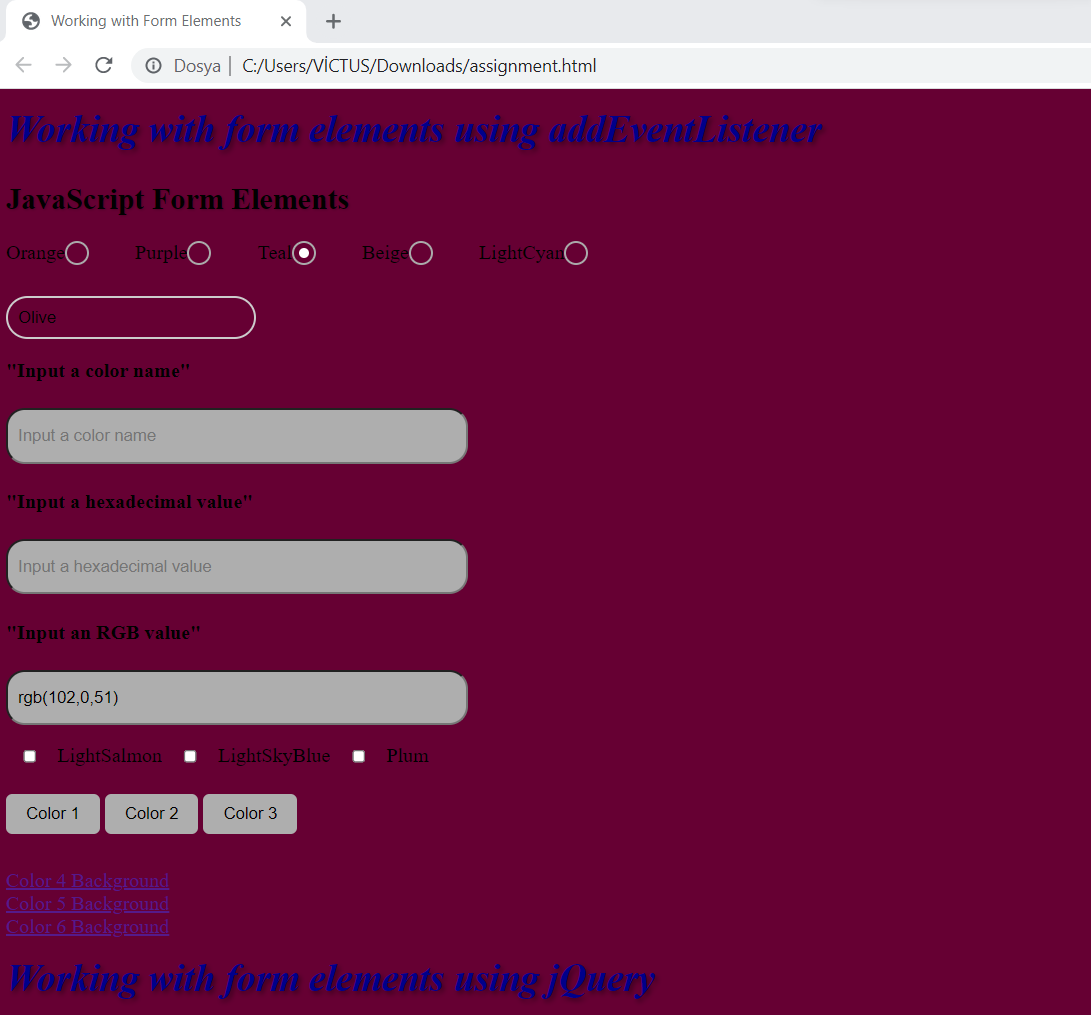
**As you see, when we click selection menu , screen displays this color.**

****

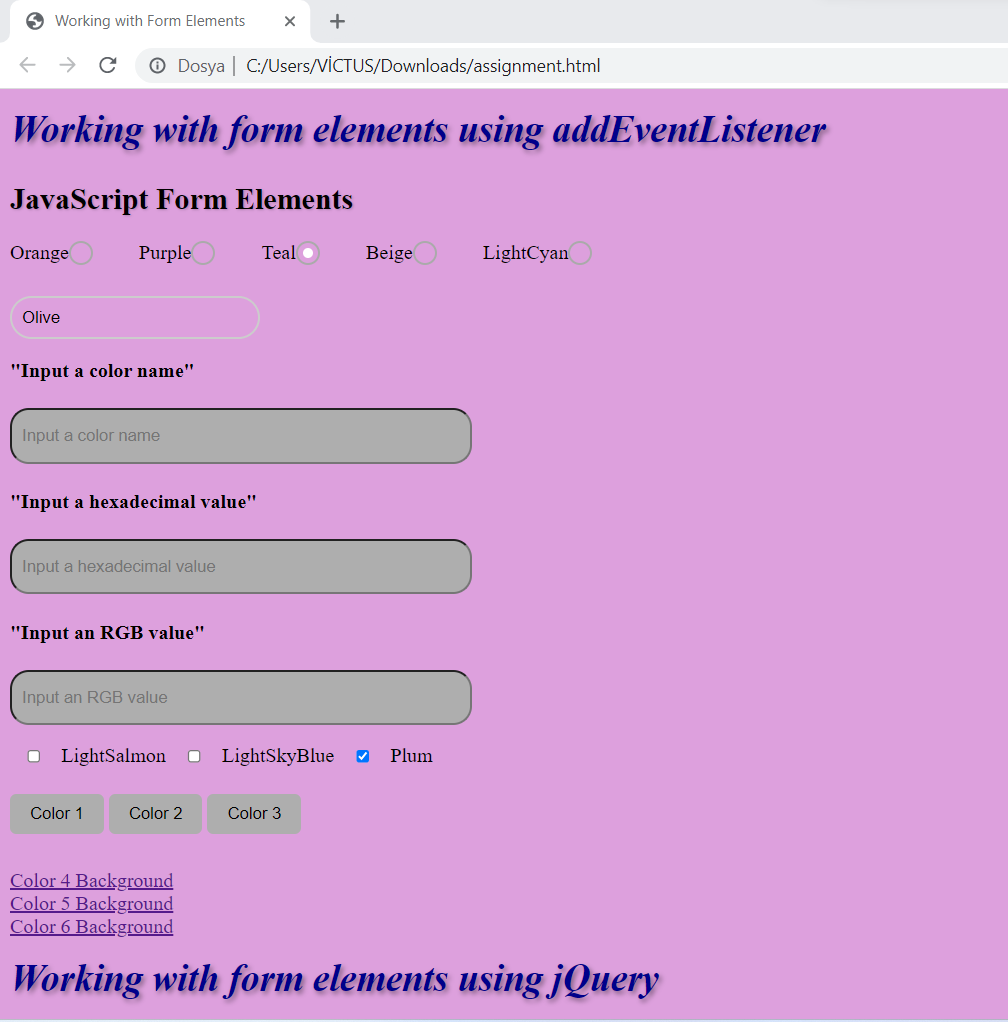
**As you see, when we write a color name in text box, screen displays this color.**



**As you see, when we write a hexadecimal value in text box, screen displays this color.**

****

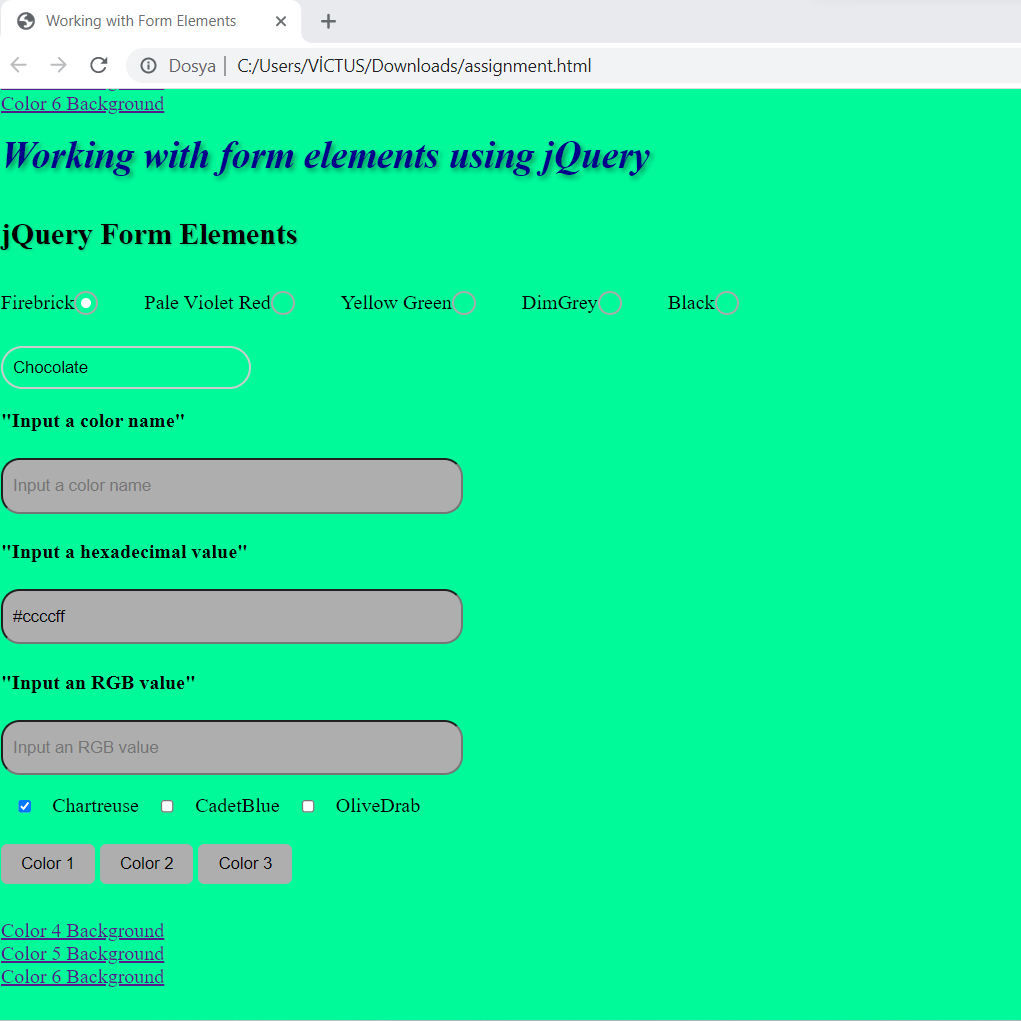
**As you see, when we write a RGB value in text box, screen displays this color.**

****

**When we clicked Plum checkbox button , screen displays this color.**

****

**As you see, when we click Color 3 button, screen displays this color.**

****

**When we click Color 5 Background, screen displays this color.**

**SUMMARY**

**HTML Structure:**

* The code starts with the <!DOCTYPE html> declaration, indicating that it is an HTML5 document.
* The <style> section contains CSS rules for styling the elements.
* The <script> tag includes the jQuery library from a CDN (Content Delivery Network) to utilize jQuery functionality.
* The <body> section contains the visible content of the webpage.

**JavaScript Form Elements:**

* Within the <form> element with the id "js-form," there are various form elements like radio buttons, select dropdowns, text inputs, checkboxes, and buttons.
* Each radio button has a corresponding label and a hidden input field with the type set to "radio." These radio buttons allow users to choose a color.
* The select dropdown allows users to select a color from a predefined list.
* The text inputs are used for entering color names, hexadecimal values, and RGB values.
* The checkboxes allow users to select multiple colors.
* The <input> elements with type="button" are used as buttons to change the background color when clicked.
* The <a> tags with onclick attributes call the changeColor() JavaScript function to change the background color when clicked.

**jQuery Form Elements:**

* Within the <form> element with the id "jquery-form," similar form elements as in the JavaScript section are present, but with a different name attribute ("jquery-color").
* The functionality to change the background color using jQuery is implemented separately from the JavaScript section.
* The jQuery code uses selectors to target the form elements and bind event listeners to them.

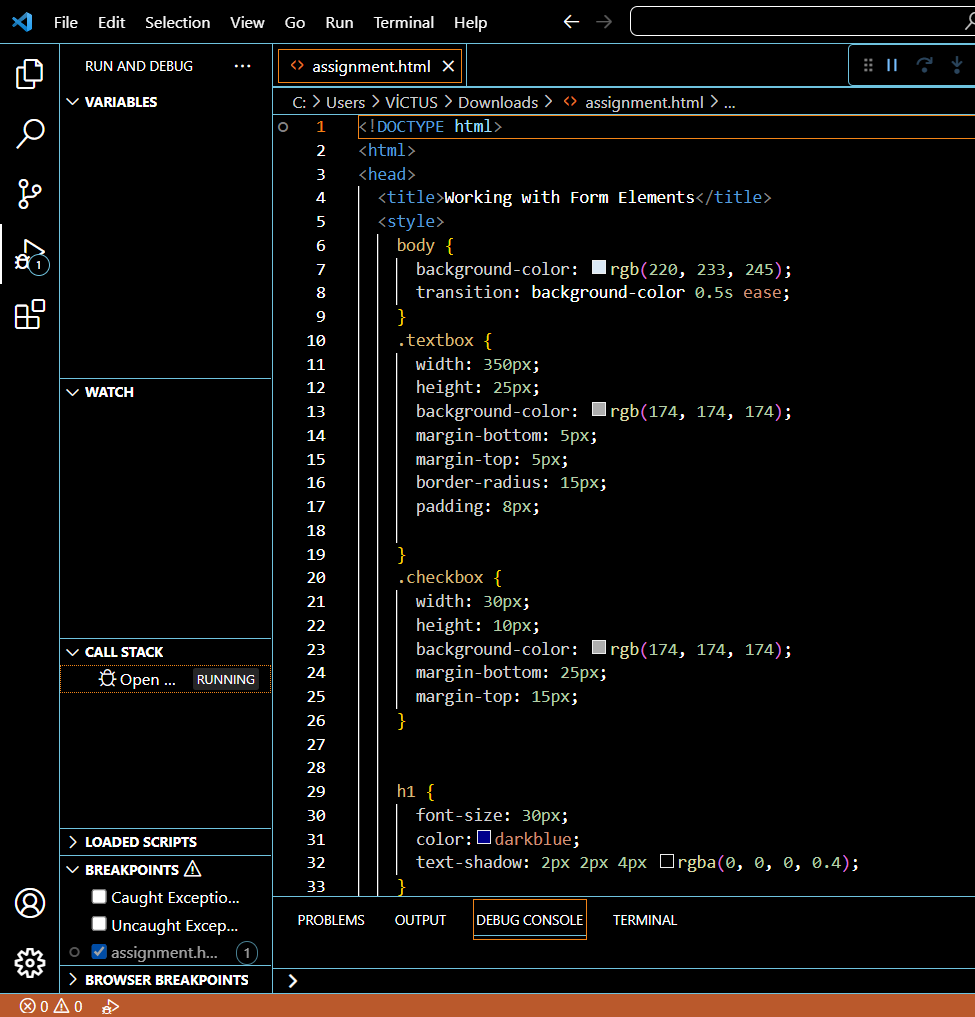
**JavaScript Event Listeners:**

* The JavaScript code wrapped inside the DOMContentLoaded event listener ensures that the JavaScript code runs when the HTML document has finished loading.
* The code selects the form elements using their names and assigns event listeners to them.
* When any of the form elements' values change, the event listener updates the background color of the <body> element.

**changeColor() Function:**

* The changeColor() function is a JavaScript function that takes a color parameter.
* When called, this function changes the background color of the <body> element to the specified color.
* Overall, this code demonstrates how to work with various form elements and manipulate the background color of a webpage using JavaScript and jQuery. It shows different ways to capture user input and respond to events triggered by form elements.

**Let's go through the code and explain each part:**

****

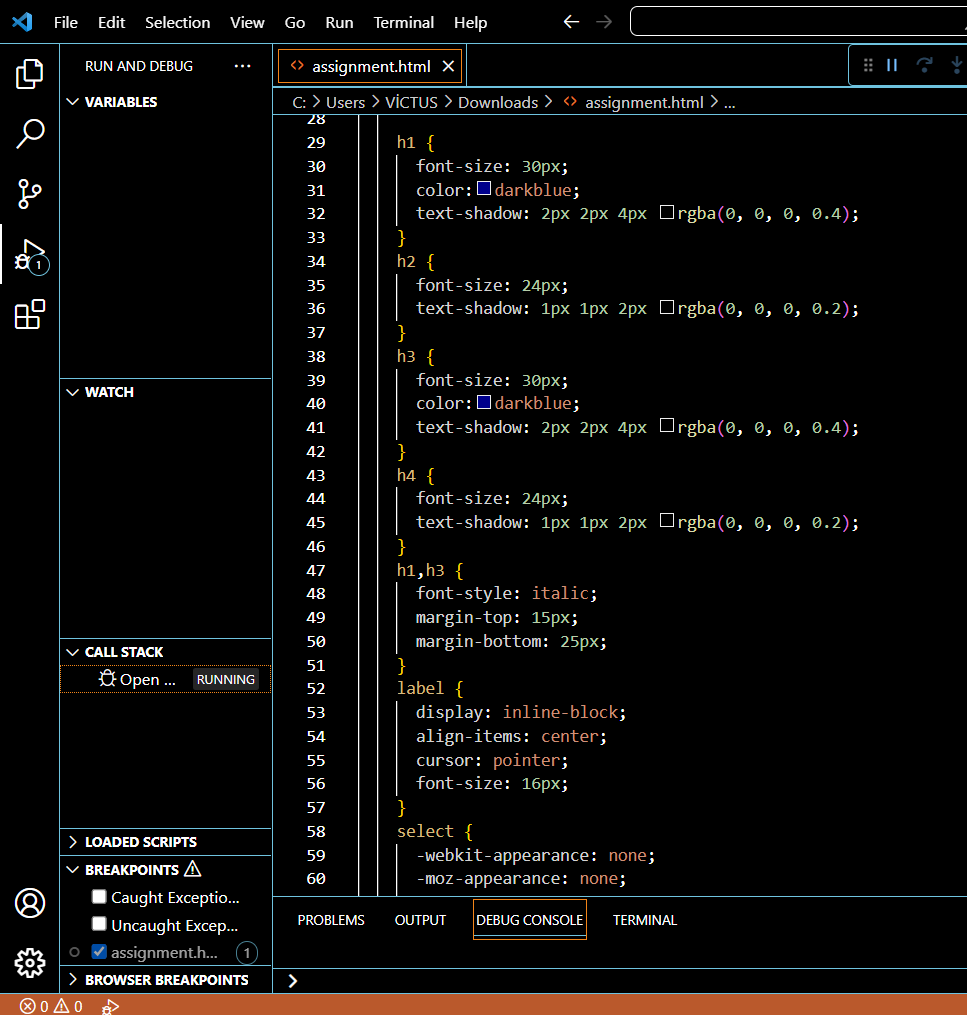
**The HTML code sets up the basic structure of an HTML document with a title and some CSS styles defined in the <style> tag. The CSS code provides styling for the body, textboxes, checkboxes, and <h1> elements.**

**HTML:**

* **<!DOCTYPE html>: This is the document type declaration, specifying that the document is an HTML5 document.**
* **<html>: The root element of the HTML document.**
* **<head>: Contains metadata and other elements that define the document.**
* **<title>: Specifies the title of the document displayed in the browser's title bar.**
* **<style>: Used to define CSS styles within the HTML document.**
* **<body>: Contains the visible content of the document.**

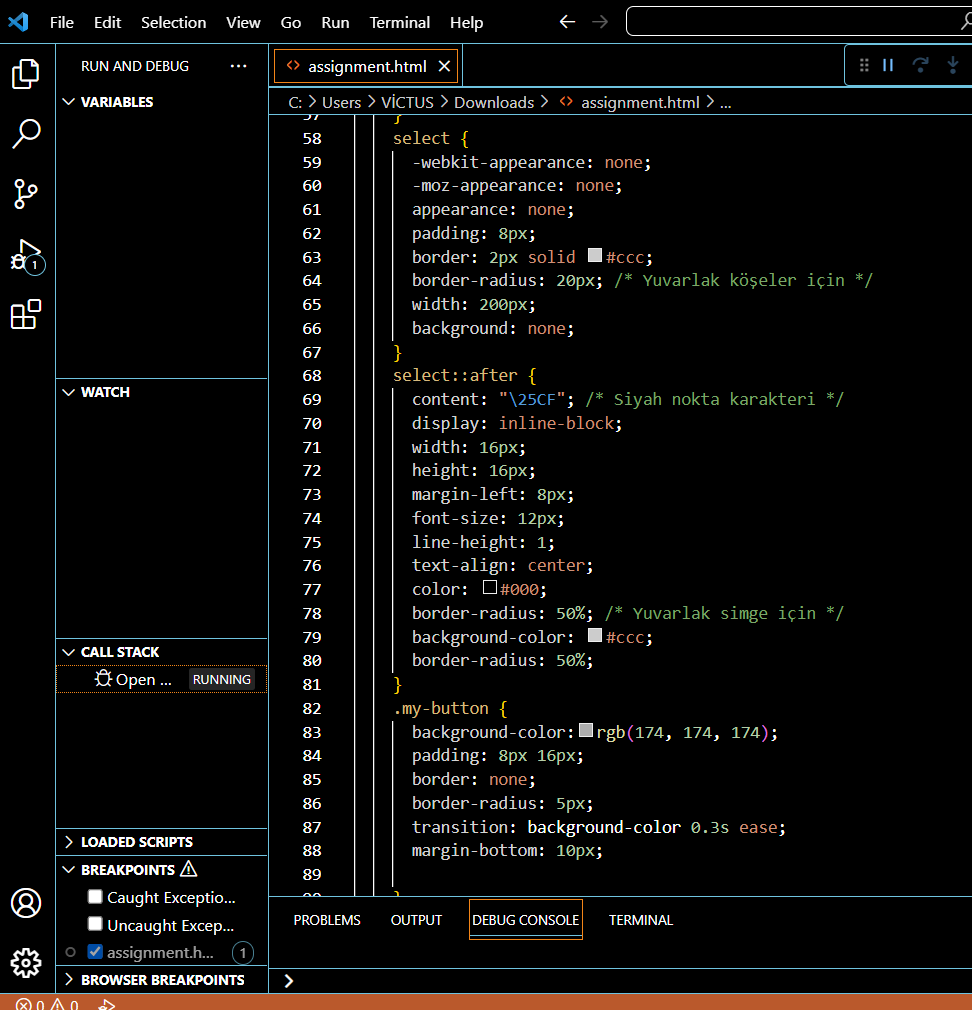
**CSS:**

* **body: Selects the <body> element and sets the background color to rgb(220, 233, 245), which is a light blue shade.**
* **transition: background-color 0.5s ease;: Specifies a transition effect for the background color property with a duration of 0.5s and an easing function.**
* **.textbox: Selects elements with the class .textbox and applies styling properties to them.**
* **.checkbox: Selects elements with the class .checkbox and applies styling properties to them.**
* **h1: Selects <h1> elements and applies styling properties to them, such as font size, color, and text shadow.**
* **font-size: 30px;: Sets the font size of <h1> elements to 30px.**
* **color: darkblue;: Sets the text color of <h1> elements to darkblue.**
* **text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.4);: Adds a text shadow effect to <h1> elements with a horizontal offset of 2px, vertical offset of 2px, blur radius of 4px, and a slightly transparent black color.**

****

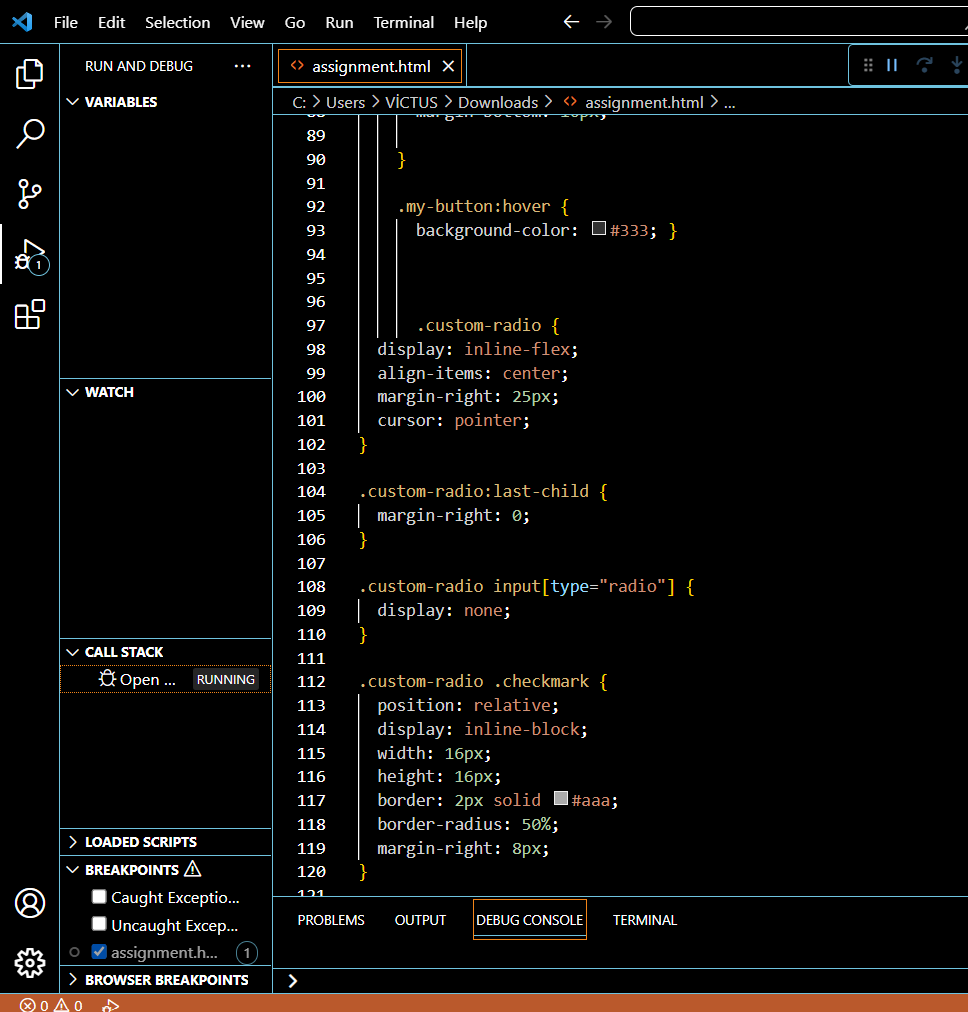
**These CSS code defines styles for various heading elements (<h1>, <h2>, <h3>, <h4>), as well as the <label> element.**

* **<h2> elements and sets the font size to 24px and applies a text shadow effect with a horizontal and vertical offset of 1px, a blur radius of 2px, and a semi-transparent black color. This creates a smaller heading with a lighter shadow effect compared to the <h1> elements.**
* **<h3> elements and sets the font size to 30px, the text color to darkblue, and applies a text shadow effect with the same properties as the <h1> elements. It's similar to the style for <h1> but is applied to a different heading level.**
* **This style targets <h4> elements and sets the font size to 24px and applies a text shadow effect with the same properties as the <h2> elements. It's similar to the style for <h2> but is applied to a different heading level.**
* **<h1> and <h3> elements. It sets the font style to italic and adds a top margin of 15px and a bottom margin of 25px. This creates an italicized text style for both heading levels and adds some vertical spacing above and below the headings.**
* **<label> elements. It sets the display to inline-block, allowing the labels to flow horizontally within their containing element. It aligns the items vertically with the align-items property (although it typically applies to flex containers), sets the cursor to pointer, indicating interactivity, and sets the font size to 16px. This style provides some common styling for labels in forms or other interactive elements.**

****

**This CSS code provides styling for <select> elements, adding custom appearance, and for elements with the class .my-button, setting their background color, padding, border, border radius, transition effect, and margin.**

* **select: Selects <select> elements and applies styling properties to them.**
* **-webkit-appearance: none; Removes the default appearance of <select> elements in WebKit-based browsers.**
* **-moz-appearance: none;: Removes the default appearance of <select> elements in Mozilla Firefox.**
* **appearance: none;: Removes the default appearance of <select> elements in other browsers.**
* **padding: 8px;: Adds 8 pixels of padding inside the <select> element.**
* **border: 2px solid #ccc;: Sets a 2-pixel solid border with the color #ccc around the <select> element.**
* **border-radius: 20px;: Sets a border radius of 20 pixels, giving the corners of the <select> element a rounded appearance.**
* **width: 200px;: Sets the width of the <select> element to 200 pixels.**
* **background: none;: Removes the background color from the <select> element.**
* **select::after: Selects the pseudo-element ::after of <select> elements and applies styling properties to it.**
* **content: "\25CF";: Sets the content of the pseudo-element to a black dot character (●).**
* **display: inline-block;: Makes the pseudo-element behave like an inline-block element.**
* **width: 16px;: Sets the width of the pseudo-element to 16 pixels.**
* **height: 16px;: Sets the height of the pseudo-element to 16 pixels.**
* **margin-left: 8px;: Adds 8 pixels of left margin to the pseudo-element.**
* **font-size: 12px;: Sets the font size of the pseudo-element to 12 pixels.**
* **line-height: 1;: Sets the line height of the pseudo-element to 1.**
* **text-align: center;: Centers the content of the pseudo-element horizontally.**
* **color: #000;: Sets the color of the pseudo-element's content to black.**
* **border-radius: 50%;: Sets a border radius of 50%, making the pseudo-element appear as a circle.**
* **background-color: #ccc;: Sets the background color of the pseudo-element to #ccc.**
* **.my-button: Selects elements with the class .my-button and applies styling properties to them.**
* **background-color: rgb(174, 174, 174);: Sets the background color of the elements to rgb(174, 174, 174).**
* **padding: 8px 16px;: Adds 8 pixels of padding on the top and bottom, and 16 pixels of padding on the left and right sides of the elements.**
* **border: none;: Removes the border from the elements.**
* **border-radius: 5px;: Sets a border radius of 5 pixels, giving the corners of the elements a slightly rounded appearance.**
* **transition: background-color 0.3s ease;: Specifies a transition effect for the background color property with a duration of 0.3 seconds and an easing function of ease.**
* **margin-bottom: 10px;: Adds 10 pixels of margin at the bottom of the elements.**

****

**The CSS code provides styles for buttons, radio buttons, and their hover effects.**

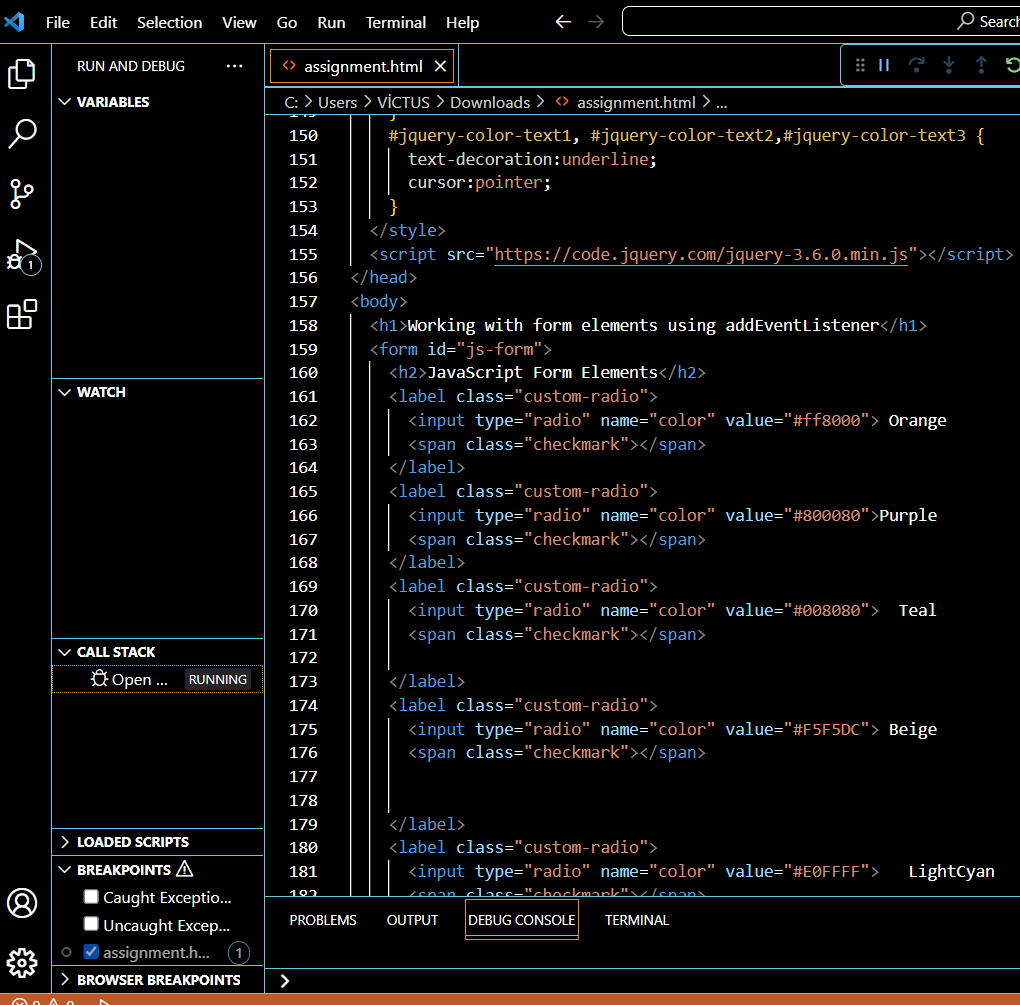
* **.my-button:hover: This selector targets elements with the class .my-button when they are being hovered over.**
* **background-color: #333;: It sets the background color of the hovered elements to #333, which is a dark gray color.**
* **.custom-radio: This selector targets elements with the class .custom-radio.**
* **display: inline-flex;: It sets the display property of the elements to inline-flex, allowing them to be displayed in a horizontal line and flexbox behavior.**
* **align-items: center;: It aligns the items vertically in the center within the flex container.**
* **margin-right: 25px;: It adds a right margin of 25px to create some space between the elements.**
* **cursor: pointer;: It changes the cursor to a pointer when hovering over the elements.**
* **.custom-radio input[type="radio"]: This selector targets input elements of type radio within elements with the class .custom-radio.**
* **display: none;: It hides the radio input elements, making them invisible.**
* **.custom-radio .checkmark: This selector targets elements with the class .checkmark within elements with the class .custom-radio.**
* **position: relative;: It positions the elements relatively, meaning their position can be adjusted using other CSS properties.**
* **display: inline-block;: It displays the elements as inline blocks.**
* **width: 16px; height: 16px;: It sets the width and height of the elements to 16px.**
* **border: 2px solid #aaa;: It sets a border with a width of 2px, a solid style, and a color of #aaa (light gray).**
* **border-radius: 50%;: It gives the elements a circular shape by setting the border-radius to 50%.**
* **margin-right: 8px;: It adds a right margin of 8px to create some space between the elements.**

**metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu**

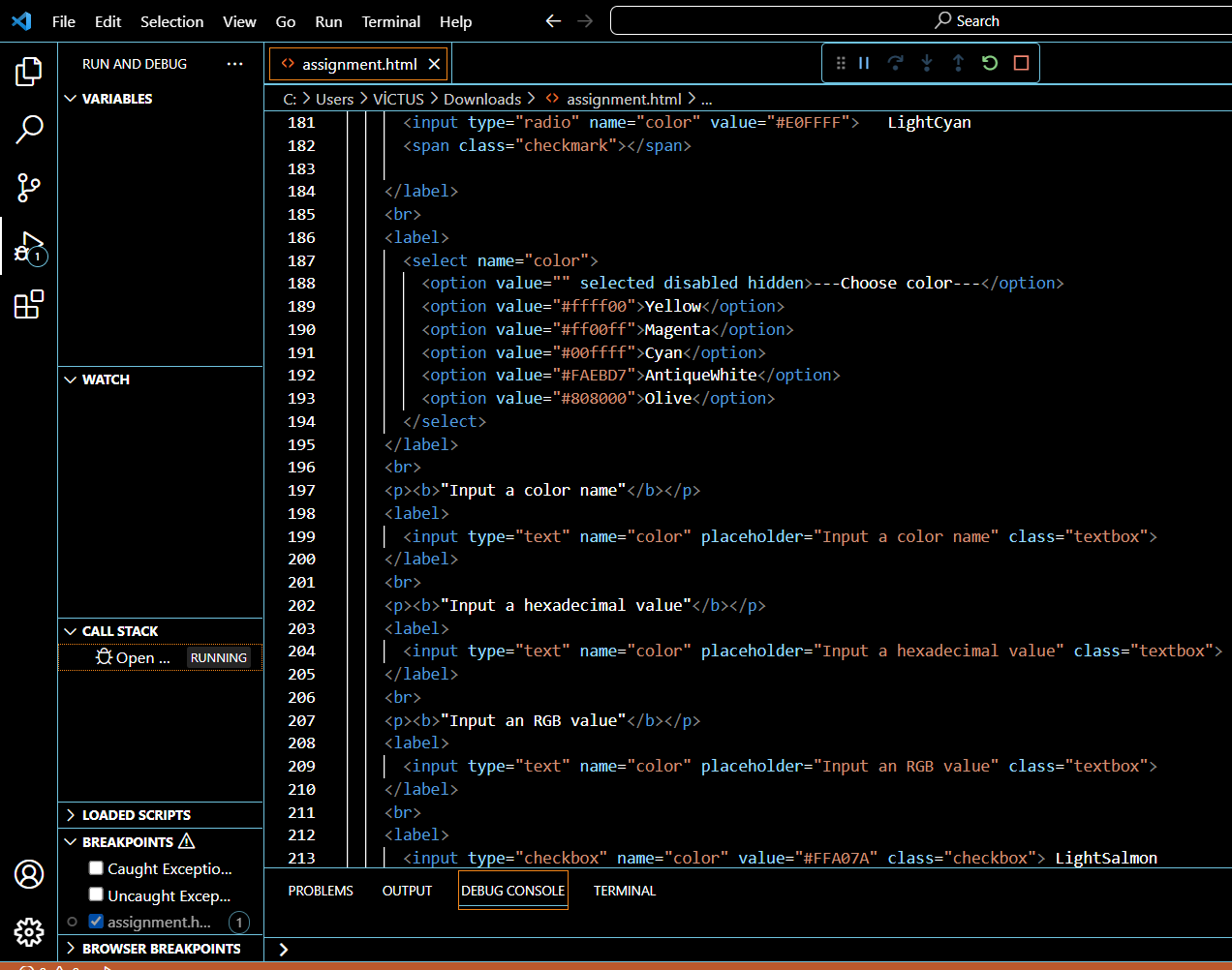
**This CSS code provides styling for radio buttons and their associated labels, select elements with specific names, and elements with specific IDs.**

* **.custom-radio input[type="radio"]:checked + .checkmark::after: Selects the pseudo-element ::after of the .checkmark element that comes immediately after a checked <input type="radio"> element within an element with the class .custom-radio, and applies styling properties to it.**
* **content; Sets the content of the pseudo-element to an empty string.**
* **position: absolute;: Positions the pseudo-element absolutely within its containing element.**
* **top: 4px;: Sets the top position of the pseudo-element to 4 pixels.**
* **left: 4px;: Sets the left position of the pseudo-element to 4 pixels.**
* **width: 8px;: Sets the width of the pseudo-element to 8 pixels.**
* **height: 8px;: Sets the height of the pseudo-element to 8 pixels.**
* **border-radius: 50%;: Sets a border radius of 50%, making the pseudo-element appear as a circle.**
* **background-color: #fff;: Sets the background color of the pseudo-element to white.**
* **.custom-radio label: Selects <label> elements within elements with the class .custom-radio, and applies styling properties to them.**
* **display: inline-block;: Makes the <label> elements behave like inline-block elements.**
* **vertical-align: middle;: Vertically aligns the <label> elements to the middle.**
* **select[name="color"] and select[name="jquery-color"]: Selects <select> elements with the attribute name set to "color" and "jquery-color" respectively, and applies styling properties to them.**
* **margin-top: 25px;: Adds 25 pixels of top margin to the <select> elements.**
* **#color-text1, #color-text2, #color-text3 and #jquery-color-text1, #jquery-color-text2, #jquery-color-text3: Selects elements with the IDs "color-text1", "color-text2", "color-text3", "jquery-color-text1", "jquery-color-text2", and "jquery-color-text3", and applies styling properties to them.**
* **text-decoration: underline;: Adds an underline decoration to the text of the elements.**
* **cursor: pointer;: Changes the cursor to a pointer when hovering over the elements.**

****

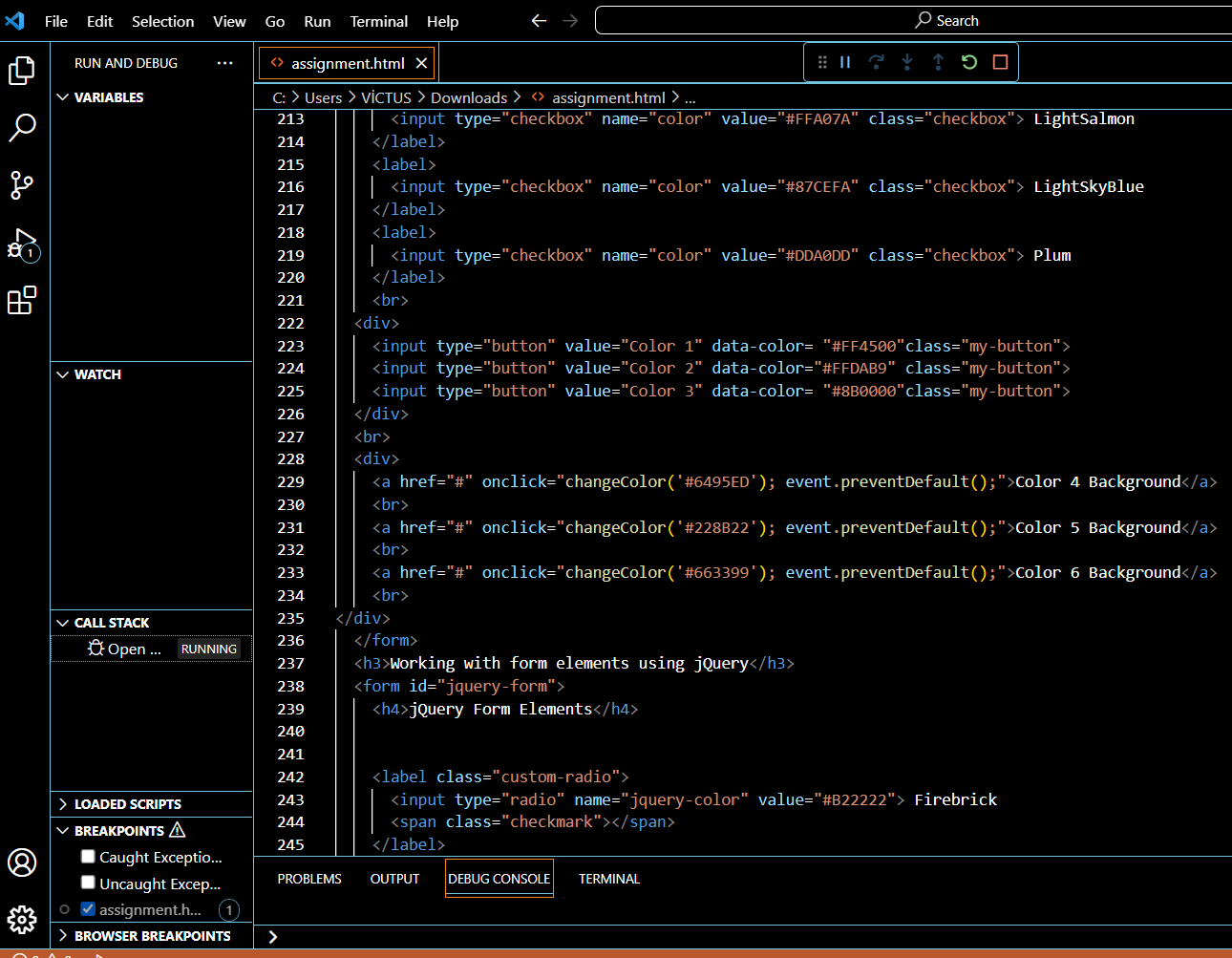
**The purpose of this code creating a form with radio buttons for selecting different colors. The use of the "custom-radio" class and the associated CSS styles is likely to provide customized styling for the radio buttons. The jQuery library is included.**

* **It includes a <script> tag to include the jQuery library. The <body> section contains a heading (<h1>) and a form (<form>) with some form elements inside.**
* **The heading element <h1> displays the text "Working with form elements using addEventListener".**
* **The form element <form id="js-form"> has an ID attribute set to "js-form".**
* **Inside the form, there are several radio buttons (<input type="radio">) with corresponding labels (<label>) and checkmark spans (<span class="checkmark"></span>).**
* **Each radio button has a name attribute set to "color" and a different value attribute representing different colors.**
* **The text content of the labels represents the color names, such as "Orange", "Purple", "Teal", "Beige", and "LightCyan".**

****

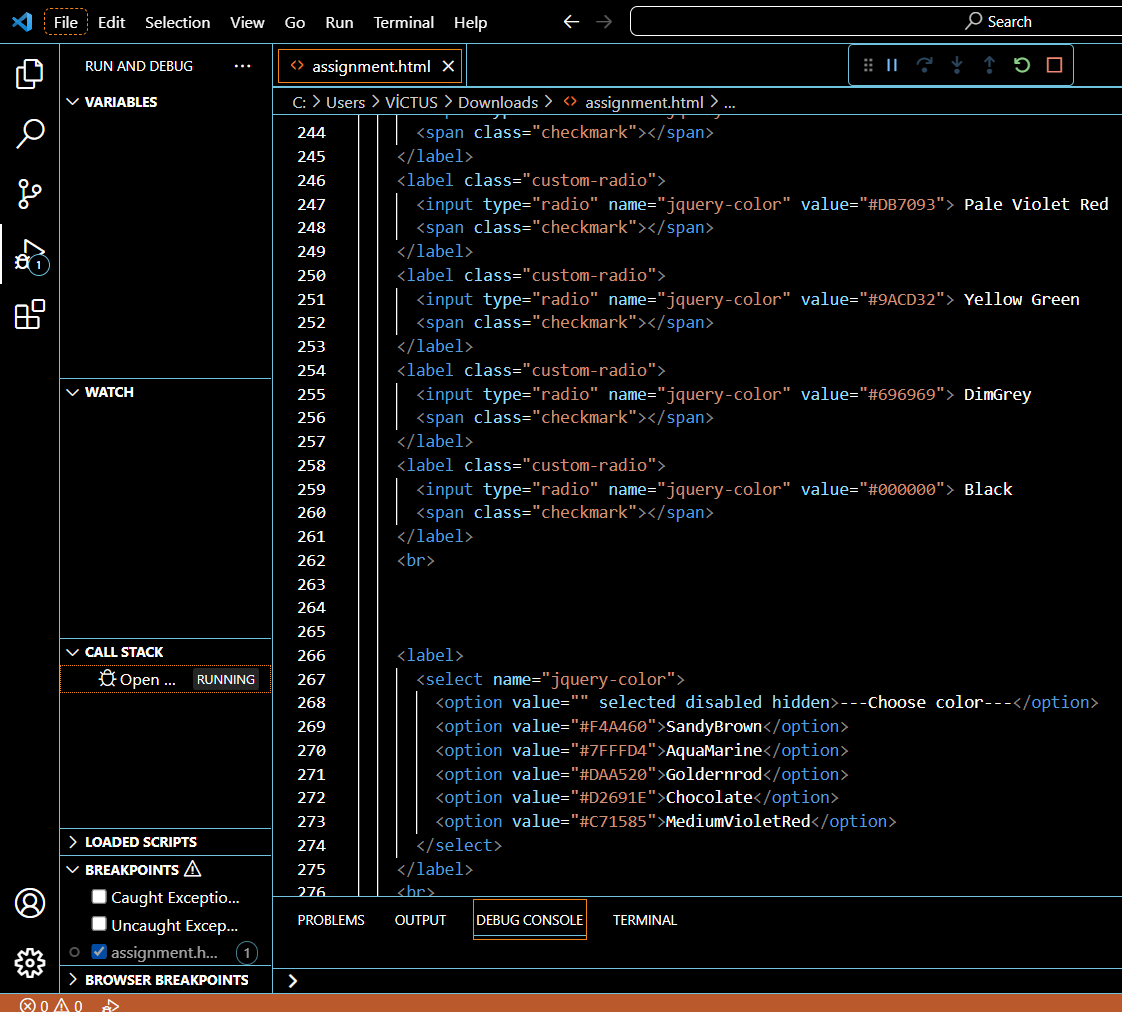
**The purpose of this code is providing various options for selecting or inputting color values in different formats (color names, hexadecimal, and RGB) within a form.**

* **The first element is a <select> dropdown menu with the name "color". It contains several <option> elements representing different color choices. The first option is selected by default and displays the text "---Choose color---". Each option has a corresponding value attribute that represents a color in hexadecimal format.**
* **Following the dropdown menu, there is a heading <p> element that says "Input a color name". After the heading, there is an <input> element of type "text" with the name "color" and a placeholder attribute set to "Input a color name". It has the class "textbox" which applies some styling to the input field.**
* **Similarly, there are two more <p> headings and <input> elements below. The second <input> has a placeholder for entering a hexadecimal value, and the third <input> has a placeholder for entering an RGB value. Both of these inputs also have the class "textbox".**
* **Lastly, there are two <label> elements with checkboxes <input type="checkbox"> for selecting additional colors. The checkboxes have the name "color" and a value attribute representing different colors. They also have the class "checkbox" for styling purposes.**

****

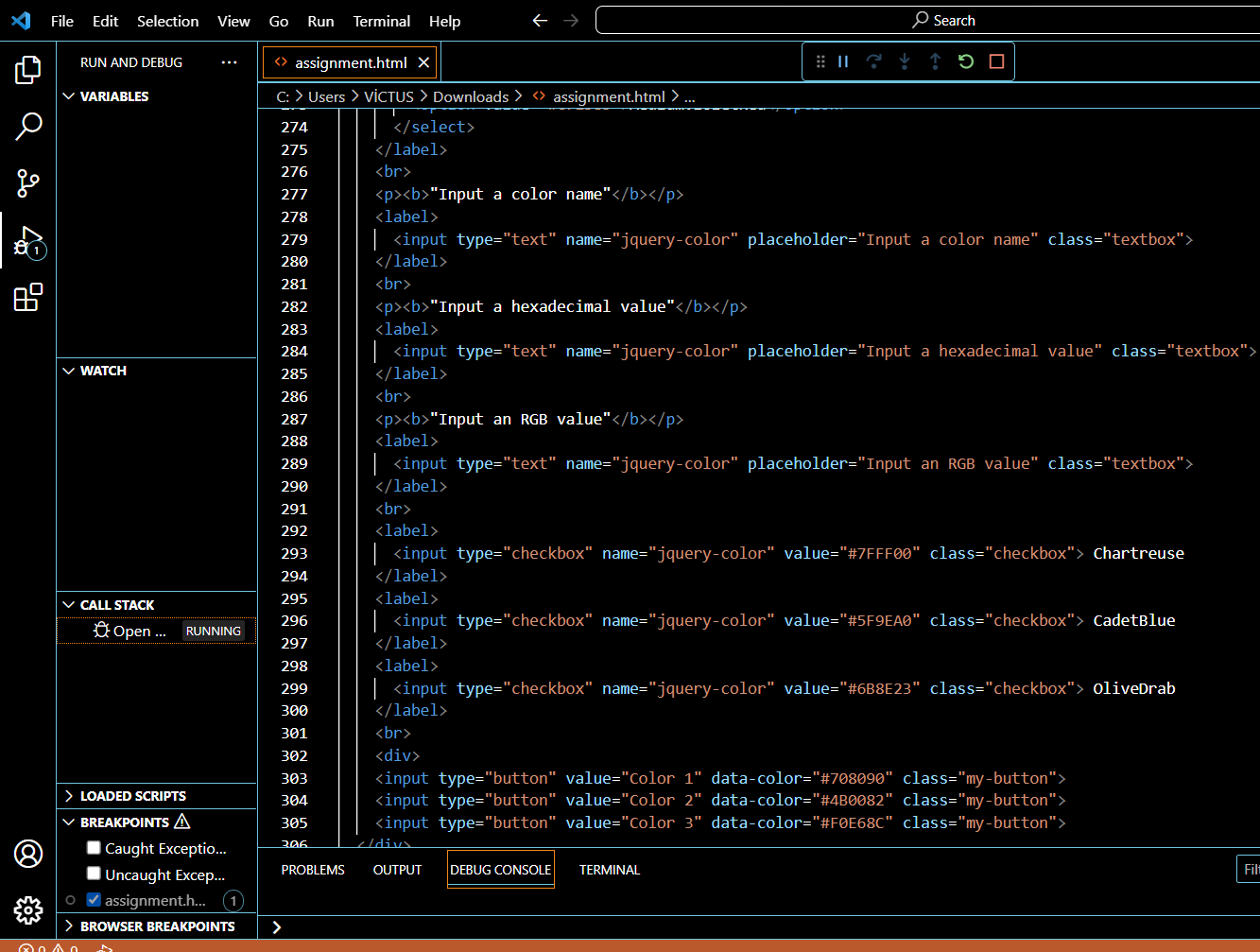
**The code demonstrates working with form elements using jQuery. It includes checkboxes, buttons, and radio buttons with associated colors. The buttons and anchors have interactive behavior to change the background color using JavaScript functions.**

* **The code includes a checkbox <input> element wrapped inside a <label>. It has the name "color", a value attribute set to "#DDA0DD" (representing the color Plum), and the class "checkbox".**
* **Following that, there is a <div> element containing three <input> elements of type "button" with different values and data-color attributes. These buttons have the class "my-button". The data-color attribute is used to associate specific colors with each button.**
* **Next, there is another <div> element containing three <a> (anchor) elements with different background color values assigned to their onclick events. When clicked, these anchors trigger a JavaScript function changeColor() with the corresponding color as an argument. The event.preventDefault() prevents the default behavior of the anchor, which is usually redirecting to a new page.**
* **After the first form, there is a heading <h3> and another form with the id "jquery-form". Inside this form, there are two radio button <input> elements with the name "jquery-color" and different values. These radio buttons are wrapped inside <label> elements and have the class "custom-radio".**

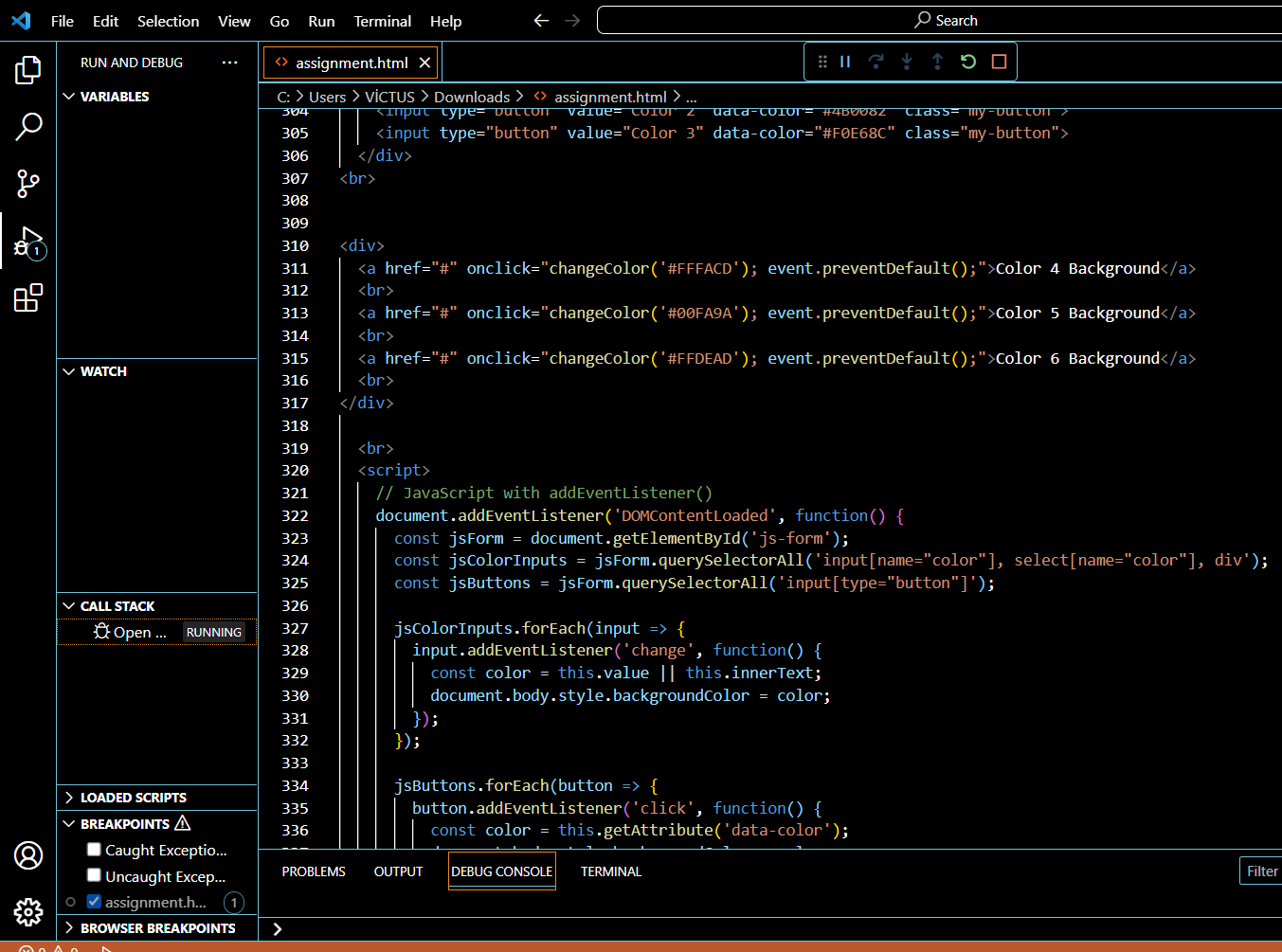
****

**These elements provide the user with different ways to select colors within the "jquery-form" form.**

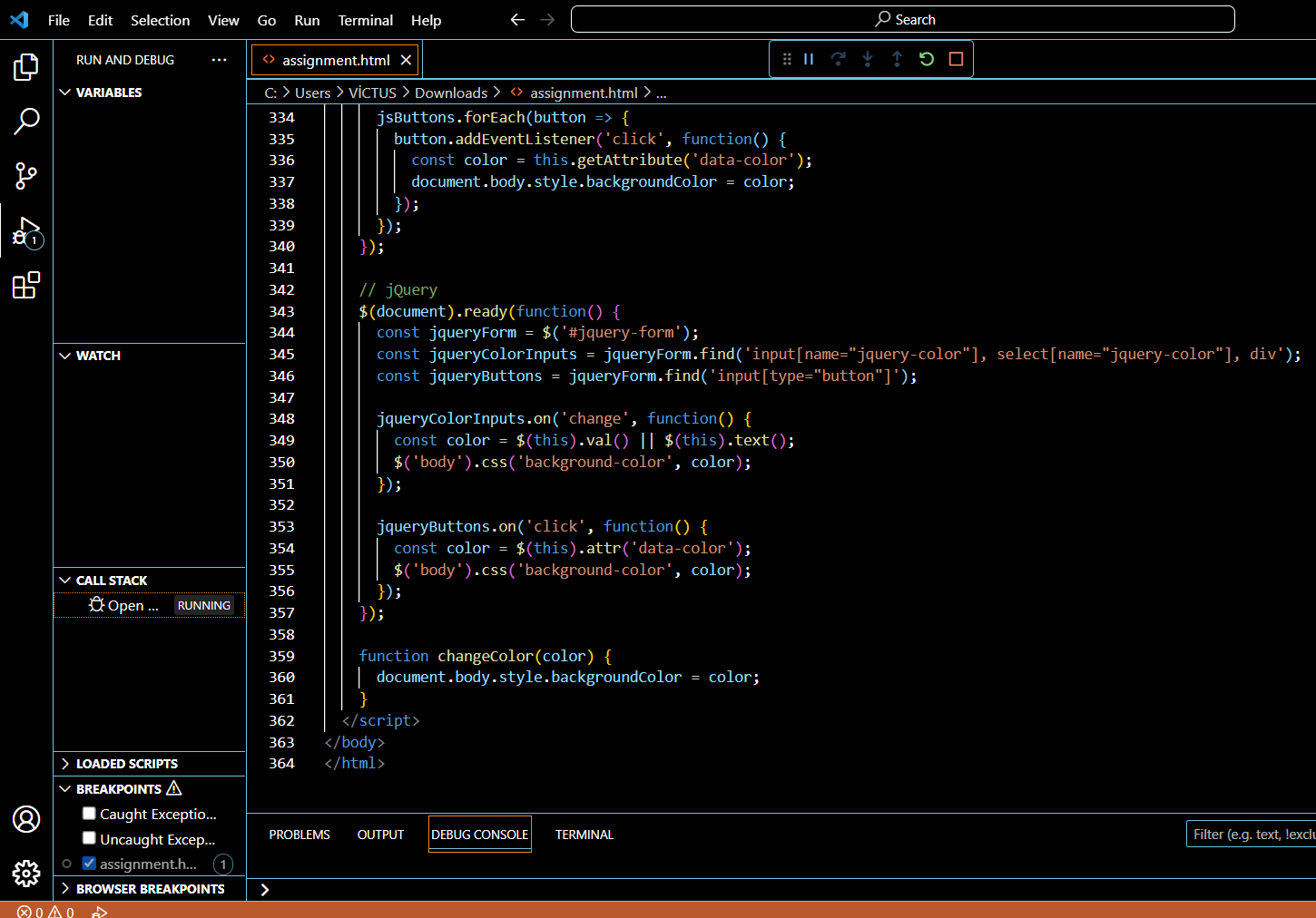
* **The code includes three radio button elements wrapped inside <label> elements with the class "custom-radio". Each radio button has a different value assigned to the name "jquery-color". The options are "Yellow Green" with the value "#9ACD32", "DimGrey" with the value "#696969", and "Black" with the value "#000000". These radio buttons allow the user to select a color option.**
* **Below the radio buttons, there is a <select> element wrapped inside a <label> element. The select element has the name "jquery-color". It includes multiple <option> elements, one of which is selected by default and displays "---Choose color---" to prompt the user to make a selection. The other options include "SandyBrown" with the value "#F4A460", "AquaMarine" with the value "#7FFFD4", "Goldernrod" with the value "#DAA520", "Chocolate" with the value "#D2691E", and "MediumVioletRed" with the value "#C71585". The user can choose a color from the dropdown list.**

****

* **There are three input elements of type "text" that allow the user to input color-related information. Each input element has the name "jquery-color" and a specific placeholder text. The first input is for entering a color name, the second is for entering a hexadecimal value, and the third is for entering an RGB value. These inputs have the class "textbox".**
* **There are three checkboxes wrapped in <label> elements. Each checkbox has the name "jquery-color" and a specific value assigned to it. The checkboxes represent color options that the user can select. The options are "Chartreuse" with the value "#7FFF00", "CadetBlue" with the value "#5F9EA0", and "OliveDrab" with the value "#6B8E23". These checkboxes have the class "checkbox".**
* **Below the checkboxes, there is a <div> element containing three <input> elements of type "button". Each button has a specific value assigned to it using the "data-color" attribute. The buttons are labeled as "Color 1", "Color 2", and "Color 3". When clicked, these buttons trigger a JavaScript function and pass the respective color value as data. These buttons have the class "my-button".**

****

* **The <div> element wraps the three <a> elements, which serve as links for changing the background color of the page.**
* **Each <a> element has an onclick attribute that triggers the changeColor() function when clicked. The function call is followed by event.preventDefault() to prevent the default behavior of the link, which is navigating to another page.**
* **The changeColor() function is a custom JavaScript function defined elsewhere in the code (not shown in the provided snippet). It takes a color value as an argument and sets the background color of the <body> element to the specified color.**
* **The purpose of these <a> elements is to allow the user to change the background color of the page by clicking on the respective links. The changeColor() function is responsible for applying the selected color to the background.**

****

**The jQuery code ensures that when the form elements or buttons change, the background color of the page is updated accordingly using either the selected value or the specified color. The changeColor() function is used as an alternative method for changing the background color directly from JavaScript.**

* **The $(document).ready() function ensures that the code inside it is executed once the document (DOM) has finished loading.**
* **jqueryForm variable stores the jQuery object representing the form element with the id 'jquery-form'.**
* **jqueryColorInputs variable stores the jQuery object containing all input elements, select elements, and div elements with the name 'jquery-color' within the form.**
* **jqueryButtons variable stores the jQuery object containing all input elements with the type 'button' within the form.**
* **The .on('change', function() { ... }) method is used to attach a change event listener to the jqueryColorInputs. When any of these inputs change, the associated function is executed.**
* **Inside the change event listener function, the selected color value is retrieved using $(this).val() or $(this).text() depending on the type of input element, and then the background color of the <body> element is set using $('body').css('background-color', color).**
* **The .on('click', function() { ... }) method is used to attach a click event listener to the jqueryButtons. When any of these buttons are clicked, the associated function is executed.**
* **Inside the click event listener function, the color value is retrieved from the data-color attribute of the clicked button using $(this).attr('data-color'), and then the background color of the <body> element is set using $('body').css('background-color', color).**
* **The changeColor() function is a custom JavaScript function defined outside the jQuery code. It takes a color value as an argument and sets the background color of the <body> element to the specified color**