# Web Apps Summative Assignment

## Scenario

Web application to help produce description of second hand books for Oxfam Bookshop.

## Requirements

- Guide through anatomy of a book specialist terminology.
- Guide through specialist terminology for the different types of damage.
- Accessibility audit? E.g. alt text for diagrams / images using to convey definitions.
- Output text result at the end. This must be easy for the user to copy to paste to the online shop listing web app. I.e. based on options user selects throughout page (must work for any combination).
- Clear way for the user to refresh the page to use it for the next item, or start afresh.
- Ideally link to saved role paragraph request to chatgpt, but that is beyond scope of this course.
- Content to be refined - beyond scope of this course.
- General: fast. Easy to use.
- General: visually appealing to use.
- User able to deselect one option clicked on in error - without having to restart the whole process.
- Use on Oxfam shop PCs.
- No confidentiality concerns.

## Design

- User 1: Exisiting volunteer. Speed up workflow.
- User 2: New volunteer. Facilitate this task without specialist terminology knowledge.
- Simplified sections (for user input) to only four. Despite the fact additional specialist terminology is often used to describe books. A large number of sections to check would make the web application difficult to use.
- Font and colours matched to corporate colours.
- Contrasting colour selected for onmouseover / onclick events.

## Development

Drafts of the web application were built up in the following stages:

| Feature | Type of Code | Code Drafted | Reference | Result |
|---|---|---|---|---|
| **HTML page, headings** | HTML essentials (e.g. <body>). HTML headings (e.g. <h1>). | Appendix 3E: Drafted Basic HTML Page in Visual Studio Code: Assignment v1.html | Duckett, *HTML & CSS* (2011) | Page displayed. Headings clear. |
| **External CSS file** | HTML link to external CSS file. CSS: body {}, h1 {}, h2 {}, color:, background-color:, font-family:. Body, headings, and background matched to corporate colours. Also used a similar sans serif font. | Appendix 3D: Drafted External CSS File in Visual Studio Code: Assignment External CSS v1.css | Duckett, *HTML & CSS* (2011) Oxfam Shop (2024). | Font colour changed: HTML link to CSS file worked. |
| **Updated page layout** | HTML: <header>, <div>, comments. CSS: margin:, text-align:, width:, height:, float:, clear:. | Appendix 3A: GitHub repository | Duckett, *HTML & CSS* (2011) | Experimented with CSS to replicate design. Corrected a typing mistake of an extra bracket. |
| **First draft of writing interactive JavaScript for the web page to respond to user input** | JavaScript: getElementById, .innerHTML, function (), onclick. HTML <script> tags. | Appendix 3C: Tested Writing Interactive JavaScript in Visual Studio Code: Test_option_three .html | W3Schools, *JavaScript Where To* (2024). Oxfam Shop (2024). | Success: clicking each button displayed its text at the bottom of the page. First had to make sure the unique function names appeared in all sections of the code. Initially, one button overwrote another because ‘getElementById("sA")’ appeared twice. |

| | | | | |
|---|---|---|---|---|
| | | | | Buttons were selected based on one sample of product listings in the online shop. |
| **Present the text output as one continuous paragraph** (instead of multiple paragraphs) | HTML: <p>, <span>. | Appendix 3A: GitHub repository | W3Schools, *HTML <span> Tag* (2024). | Was able to retain the ‘id’ attributes for use with JavaScript, by replacing the <p> tags with <span> tags.<br><br>Had to add spaces to the end of the text strings. |
| **Reset button at end of form to take user back to start** (for next book) | HTML: id, a href.<br><br>JavaScript: location.reload(), window.open(). | Appendix 3A: GitHub repository | Duckett, *HTML & CSS* (2011).<br>• ‘Linking’ p.88.<br><br>W3Schools, *Window location.reload()* (2024).<br><br>W3Schools, *Window open()* (2024). | Worked.<br><br>Added more space between output and reset.<br><br>Added function to refresh page to return buttons styling & text output to initial values. |
| **Worked out how to get buttons clicked by the user to have a different appearance** | CSS: :active, :hover, background-color, border-radius.<br>JavaScript: .className. | Appendix 3B: Tested Coding of Button Interaction in Visual Studio Code: test_button.html | Duckett, *HTML & CSS* (2011).<br>• ‘Responding to Users’ p.291.<br>• ‘Styling Text Inputs’ p.342.<br><br>Duckett, *JavaScript & jQuery* (2014).<br>• ‘Adding and Removing Properties’ p.112. | ‘:active’ in CSS did change the button styling, but it was not persistent.<br><br>Instead, using JavaScript to change the Class attribute made a persistent change to the button styling. |
| **Tested whether one onclick event would trigger two actions**<br><br>(i.e. button click to both display | JavaScript: function (). | Appendix 3B: Tested Coding of Button Interaction in Visual Studio Code: test_button.html | Duckett, *JavaScript & jQuery* (2014). | Success: separating the two statements with a semi colon meant they worked when included in the same function. |

| | | | | |
|---|---|---|---|---|
| text and change the button styling) | | | | |
| **Enable the user to switch back and forth between a button being selected and deselected** | JavaScript: If … Else. | Appendix 3B: Tested Coding of Button Interaction in Visual Studio Code: test_button.html | W3Schools, *Change the Source of the Lightbulb* (2024). | Success: adapted the If … Else coding example from W3Schools to check the current status of the text display (instead of the image display).<br><br>Initially checking for blank / hidden text did not work. Reversed the order of the statements inside If … Else, and checked for the full sentence text instead. |
| **Display further information when user hovers mouse over diagram** | JavaScript: onmouseover, onmouseout.<br><br>Prepared PNG images in Procreate app. | Appendix 3A: GitHub repository | W3Schools, *JavaScript HTML DOM Events* (2024).<br><br>Oxfam, Anatomy of a Book (2024). | Success: using onmouseover and onmouseout together, was able to switch between images.<br><br>Needed to code the image size, and prepare the image files at a consistent size.<br><br>Maintained the ratio between height and width when coding the <img> display smaller than the pixel measurements of the original image. This prevented distortion. |
| **Display further information when user hovers mouse over button** | JavaScript: onmouseover, onmouseout.<br><br>Prepared PNG images in Procreate app. | Appendix 3A: GitHub repository | W3Schools, *JavaScript HTML DOM Events* (2024).<br><br>Oxfam, Glossary of Terminology (2024). | As above. |

| | | | | |
|---|---|---|---|---|
| | | | Oxfam Shop (2024). | |

## Testing

The ‘Result’ column in the previous section documents the testing of which types of code could deliver the requirements.

Tested the final web application on GitHub Pages – interacts as expected.

For example, adjusts to half screen window.

Tested deployment:

- GitHub Project page was accessible during Week 10 Teams class. (3 Dec)
- Used draft version of the web application (text generated by buttons) on Oxfam shop PC during weekly volunteering. (6 Dec)
- Used draft version of the web application (text generated by buttons) on Oxfam shop PC during weekly volunteering to list a book for sale. (13 Dec)

## Deployment

The web application is deployed on: https://makepfc.github.io/condition/ .

## Evaluation

Strengths:

- User can select and deselect – easier/faster to use. Did not have to use alternative method of ondblclick event, which would be harder for users.
- Class was an efficient way of styling the buttons as selected / deselected, because changing the CSS in one place updates all of the buttons. In contrast, using something like image buttons would have required updating two sets of images for every button to make a style change.
- Reset button speeds up workflow.

Weaknesses:

- Data structure - beyond scope of course.
- Incomplete photos of damage terminology. (buttons)
- No limit on length / readability.

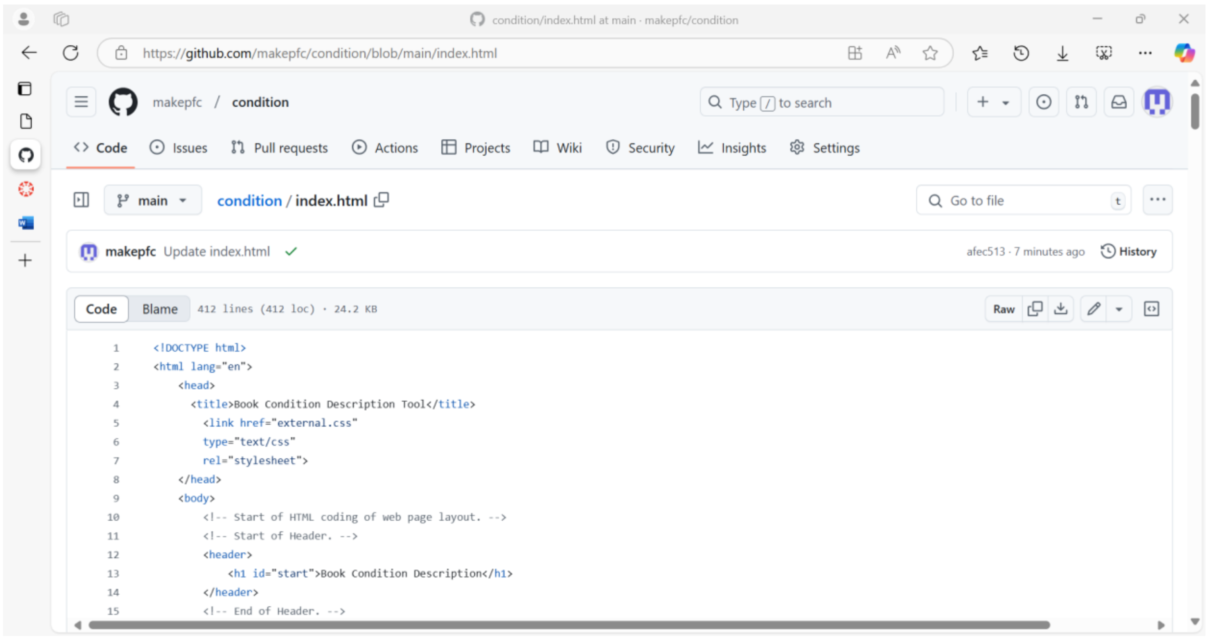
- No logic check: possible to select contradictory descriptions (e.g. press both the 'ink annotations' button and the 'free from annotations' button).
- Duplication of code with the same function uniquely named for every button – there may be a more efficient way to write this code as a shorter version for maintenance.
- Accessibility such as alt text.

## Appendix 1: Copy of Source Code

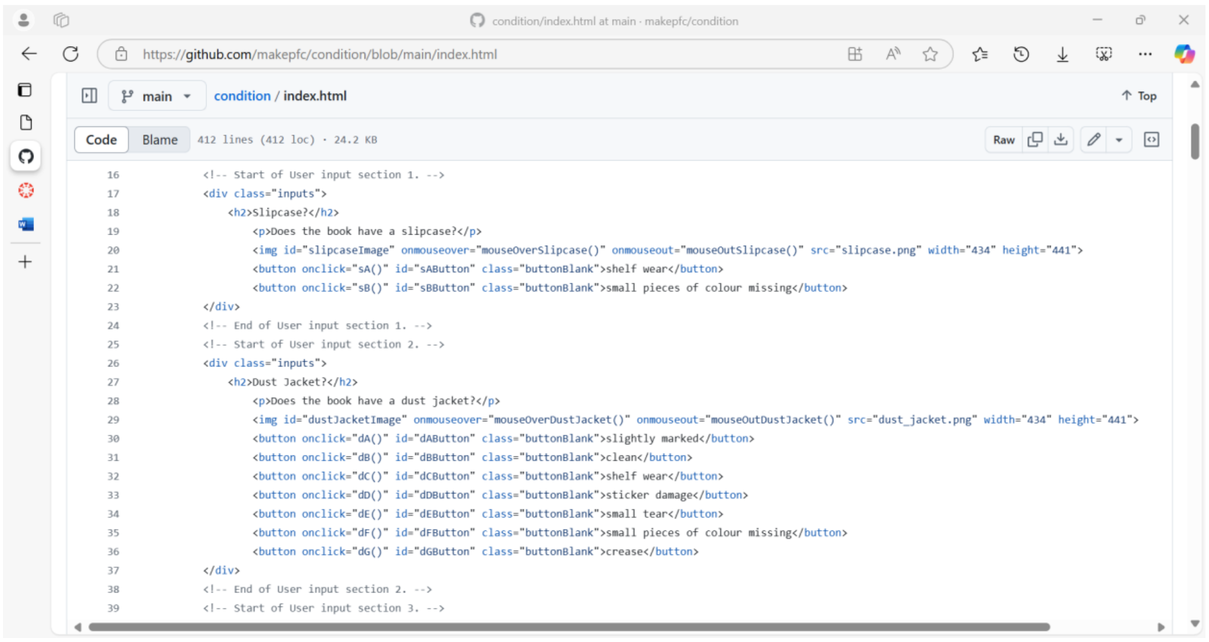
- The web application does not use generated code from templates. All code was typed by hand, using the elements of HTML, CSS & JavaScript learnt from lectures and reading Duckett & W3Schools (see Reference List).

### *Appendix 1A: Copy of index.html Deployed on GitHub*

Example screenshots:



```
<!DOCTYPE html>
<html lang="en">
    <head>
      <title>Book Condition Description Tool</title>
        <link href="external.css"
        type="text/css"
        rel="stylesheet">
    </head>
    <body>
        <!-- Start of HTML coding of web page layout. -->
        <!-- Start of Header. -->
        <header>
            <h1 id="start">Book Condition Description</h1>
        </header>
        <!-- End of Header. -->
```


condition / index.html

412 lines (412 loc) · 24.2 KB

```html
        <!-- Start of User input section 1. -->
        <div class="inputs">
            <h2>Slipcase?</h2>
                <p>Does the book have a slipcase?</p>
                <img id="slipcaseImage" onmouseover="mouseOverSlipcase()" onmouseout="mouseOutSlipcase()" src="slipcase.png" width="434" height="441">
                <button onclick="sA()" id="sAButton" class="buttonBlank">shelf wear</button>
                <button onclick="sB()" id="sBButton" class="buttonBlank">small pieces of colour missing</button>
        </div>
        <!-- End of User input section 1. -->
        <!-- Start of User input section 2. -->
        <div class="inputs">
            <h2>Dust Jacket?</h2>
                <p>Does the book have a dust jacket?</p>
                <img id="dustJacketImage" onmouseover="mouseOverDustJacket()" onmouseout="mouseOutDustJacket()" src="dust_jacket.png" width="434" height="441">
                <button onclick="dA()" id="dAButton" class="buttonBlank">slightly marked</button>
                <button onclick="dB()" id="dBButton" class="buttonBlank">clean</button>
                <button onclick="dC()" id="dCButton" class="buttonBlank">shelf wear</button>
                <button onclick="dD()" id="dDButton" class="buttonBlank">sticker damage</button>
                <button onclick="dE()" id="dEButton" class="buttonBlank">small tear</button>
                <button onclick="dF()" id="dFButton" class="buttonBlank">small pieces of colour missing</button>
                <button onclick="dG()" id="dGButton" class="buttonBlank">crease</button>
        </div>
        <!-- End of User input section 2. -->
        <!-- Start of User input section 3. -->
```

Link to full html file:

https://github.com/makepfc/condition/blob/main/index.html

Text (copy of full html file):

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Book Condition Description Tool</title>
      <link href="external.css"
      type="text/css"
      rel="stylesheet">
  </head>
  <body>
      <!-- Start of HTML coding of web page layout. -->
      <!-- Start of Header. -->
      <header>
          <h1 id="start">Book Condition Description</h1>
      </header>
      <!-- End of Header. -->

      <!-- Start of User input section 1. -->

      <div class="inputs">

          <h2>Slipcase?</h2>

            <p>Does the book have a slipcase?</p>

            <img id="slipcaseImage" onmouseover="mouseOverSlipcase()"
onmouseout="mouseOutSlipcase()" src="slipcase.png" width="434" height="441">

            <button onclick="sA()" id="sAButton" class="buttonBlank">shelf wear</button>

            <button onclick="sB()" id="sBButton" class="buttonBlank">small pieces of
colour missing</button>

      </div>

      <!-- End of User input section 1. -->

      <!-- Start of User input section 2. -->

      <div class="inputs">

          <h2>Dust Jacket?</h2>

            <p>Does the book have a dust jacket?</p>

            <img id="dustJacketImage" onmouseover="mouseOverDustJacket()"
onmouseout="mouseOutDustJacket()" src="dust_jacket.png" width="434"
height="441">

            <button onclick="dA()" id="dAButton" class="buttonBlank">slightly
marked</button>

            <button onclick="dB()" id="dBButton" class="buttonBlank">clean</button>

            <button onclick="dC()" id="dCButton" class="buttonBlank">shelf
wear</button>

            <button onclick="dD()" id="dDButton" class="buttonBlank">sticker
damage</button>

            <button onclick="dE()" id="dEButton" class="buttonBlank">small tear</button>

            <button onclick="dF()" id="dFButton" class="buttonBlank">small pieces of
colour missing</button>

            <button onclick="dG()" id="dGButton" class="buttonBlank">crease</button>
      </div>

      <!-- End of User input section 2. -->

      <!-- Start of User input section 3. -->

      <div class="inputs">

          <h2>Cover</h2>

            <img id="coverImage" onmouseover="mouseOverCover()"
onmouseout="mouseOutCover()" src="cover.png" width="434" height="441">

            <button onclick="cA()" id="cAButton" class="buttonBlank">bumped
corners</button>

            <button onclick="cB()" id="cBButton" class="buttonBlank">clean and
unmarked</button>

            <button onclick="cC()" id="cCButton" class="buttonBlank">shelf
wear</button>

            <button onclick="cD()" id="cDButton" class="buttonBlank">small pieces of
colour missing</button>

      </div>

      <!-- End of User input section 3. -->

      <!-- Start of User input section 4. -->

      <div class="inputs">

          <h2>Pages</h2>

            <img id="pagesImage" onmouseover="mouseOverPages()"
onmouseout="mouseOutPages()" src="pages.png" width="434" height="441">

            <button onclick="pA()" id="pAButton" class="buttonBlank"
onmouseover="mouseOverpAButton()"
onmouseout="mouseOutpAButton()">inscribed by previous owner: Ink</button>

            <button onclick="pF()" id="pFButton" class="buttonBlank"
onmouseover="mouseOverpFButton()" onmouseout="mouseOutpFButton()">inscribed
by previous owner: Pencil</button>

            <button onclick="pB()" id="pBButton" class="buttonBlank">clean,
unmarked</button>

            <button onclick="pC()" id="pCButton"
class="buttonBlank">uncreased</button>
```

```
            <button onclick="pD()" id="pDButton" class="buttonBlank">water
damage</button>

            <button onclick="pE()" id="pEButton" class="buttonBlank"
onmouseover="mouseOverPEButton()"
onmouseout="mouseOutPEButton()">foxing</button>

            <button onclick="pG()" id="pGButton"
class="buttonBlank">discolouration</button>

            <button onclick="pH()" id="pHButton" class="buttonBlank">annotations:
Ink</button>

            <button onclick="pI()" id="pIButton" class="buttonBlank">annotations:
Pencil</button>

            <button onclick="pJ()" id="pJButton" class="buttonBlank">free from
annotations</button>

            <button onclick="pK()" id="pKButton" class="buttonBlank">tight
binding</button>

      </div>

      <!-- End of User input section 4. -->

      <!-- Start of Text output section. -->

      <div class="outputs">

         <h2>Condition Description for Shop Listing</h2>

            <!-- Start of Negative outputs from section 1. -->

               <span id="sA"></span>

               <span id="sB"></span>

            <!-- Start of Negative outputs from section 2. -->

               <span id="dA"></span>

               <span id="dC"></span>

               <span id="dD"></span>

               <span id="dE"></span>

               <span id="dF"></span>

               <span id="dG"></span>
```

```
        <!-- Start of Negative outputs from section 3. -->
            <span id="cA"></span>
            <span id="cC"></span>
            <span id="cD"></span>
        <!-- Start of Negative outputs from section 4. -->
            <span id="pA"></span>
            <span id="pF"></span>
            <span id="pD"></span>
            <span id="pE"></span>
            <span id="pG"></span>
            <span id="pH"></span>
            <span id="pI"></span>
        <!-- Start of Positive outputs go at the end (for shop listing). -->
            <span id="dB"></span>
            <span id="cB"></span>
            <span id="pB"></span>
            <span id="pC"></span>
            <span id="pJ"></span>
            <span id="pK"></span>
        <!-- Start of Reset button. -->
            <br />
            <br />
            <br />
            <br />
            <input type="button" onclick="newDoc()" value="Reset">
        <!-- End of Reset button. -->
</div>
<!-- End of Text output section. -->
```

```
    <!-- End of HTML coding of web page layout. -->

    <!-- Start of JavaScript functions. -->

    <script>

      /* Start of Function to display/hide text output & style button as
selected/deselected  (onclick): for Section 1. */

        function sA() {

          var sASelected  = document.getElementById("sA");

          if (sASelected.innerHTML.match("Slipcase has some shelf wear. ")) {

            document.getElementById("sA").innerHTML = "";

            document.getElementById("sAButton").className = "buttonBlank";

          } else {

            document.getElementById("sA").innerHTML = "Slipcase has some shelf
wear. ";

            document.getElementById("sAButton").className = "buttonSelected";

          }

        }

        function sB() {

          var sBSelected  = document.getElementById("sB");

          if (sBSelected.innerHTML.match("Slipcase has some small pieces of colour
missing. ")) {

            document.getElementById("sB").innerHTML = "";

            document.getElementById("sBButton").className = "buttonBlank";

          } else {

            document.getElementById("sB").innerHTML = "Slipcase has some small
pieces of colour missing. ";

            document.getElementById("sBButton").className = "buttonSelected";

          }

        }
      /* Start of Function to display/hide text output & style button as
selected/deselected  (onclick): for Section 2. */
        function dA() {
          var dASelected  = document.getElementById("dA");
          if (dASelected.innerHTML.match("The dust jacket is slightly marked. ")) {
            document.getElementById("dA").innerHTML = "";
            document.getElementById("dAButton").className = "buttonBlank";
          } else {
            document.getElementById("dA").innerHTML = "The dust jacket is slightly
marked. ";
            document.getElementById("dAButton").className = "buttonSelected";
          }
        }
        function dB() {
          var dBSelected  = document.getElementById("dB");
          if (dBSelected.innerHTML.match("The dust jacket is clean. ")) {
            document.getElementById("dB").innerHTML = "";
            document.getElementById("dBButton").className = "buttonBlank";
          } else {
            document.getElementById("dB").innerHTML = "The dust jacket is clean. ";
            document.getElementById("dBButton").className = "buttonSelected";
          }
        }
        function dC() {
          var dCSelected  = document.getElementById("dC");
          if (dCSelected.innerHTML.match("The dust jacket has some shelf wear. ")) {
            document.getElementById("dC").innerHTML = "";
            document.getElementById("dCButton").className = "buttonBlank";
        } else {
            document.getElementById("dC").innerHTML = "The dust jacket has some
shelf wear. ";
            document.getElementById("dCButton").className = "buttonSelected";
        }
    }
    function dD() {
        var dDSelected  = document.getElementById("dD");
        if (dDSelected.innerHTML.match("The dust jacket has some sticker damage.
")) {
            document.getElementById("dD").innerHTML = "";
            document.getElementById("dDButton").className = "buttonBlank";
        } else {
            document.getElementById("dD").innerHTML = "The dust jacket has some
sticker damage. ";
            document.getElementById("dDButton").className = "buttonSelected";
        }
    }
    function dE() {
        var dESelected  = document.getElementById("dE");
        if (dESelected.innerHTML.match("There is a small tear to the dust jacket. ")) {
            document.getElementById("dE").innerHTML = "";
            document.getElementById("dEButton").className = "buttonBlank";
        } else {
            document.getElementById("dE").innerHTML = "There is a small tear to the
dust jacket. ";
            document.getElementById("dEButton").className = "buttonSelected";
        }
    }
```

```
            function dF() {
              var dFSelected  = document.getElementById("dF");
              if (dFSelected.innerHTML.match("The dust jacket has small pieces of colour
missing. ")) {
                document.getElementById("dF").innerHTML = "";
                document.getElementById("dFButton").className = "buttonBlank";
              } else {
                document.getElementById("dF").innerHTML = "The dust jacket has small
pieces of colour missing. ";
                document.getElementById("dFButton").className = "buttonSelected";
              }
            }
            function dG() {
              var dGSelected  = document.getElementById("dG");
              if (dGSelected.innerHTML.match("The dust jacket has a small crease. ")) {
                document.getElementById("dG").innerHTML = "";
                document.getElementById("dGButton").className = "buttonBlank";
              } else {
                document.getElementById("dG").innerHTML = "The dust jacket has a small
crease. ";
                document.getElementById("dGButton").className = "buttonSelected";
              }
            }
          /* Start of Function to display/hide text output & style button as
selected/deselected  (onclick): for Section 3. */
            function cA() {
              var cASelected  = document.getElementById("cA");
              if (cASelected.innerHTML.match("The book has bumped corners. ")) {
                document.getElementById("cA").innerHTML = "";
                document.getElementById("cAButton").className = "buttonBlank";
              } else {
                document.getElementById("cA").innerHTML = "The book has bumped
corners. ";
                document.getElementById("cAButton").className = "buttonSelected";
              }
            }
            function cB() {
              var cBSelected  = document.getElementById("cB");
              if (cBSelected.innerHTML.match("The cover is clean and unmarked. ")) {
                document.getElementById("cB").innerHTML = "";
                document.getElementById("cBButton").className = "buttonBlank";
              } else {
                document.getElementById("cB").innerHTML = "The cover is clean and
unmarked. ";
                document.getElementById("cBButton").className = "buttonSelected";
              }
            }
            function cC() {
              var cCSelected  = document.getElementById("cC");
              if (cCSelected.innerHTML.match("The cover has some shelf wear. ")) {
                document.getElementById("cC").innerHTML = "";
                document.getElementById("cCButton").className = "buttonBlank";
              } else {
                document.getElementById("cC").innerHTML = "The cover has some shelf
wear. ";
                document.getElementById("cCButton").className = "buttonSelected";
              }
            }
```

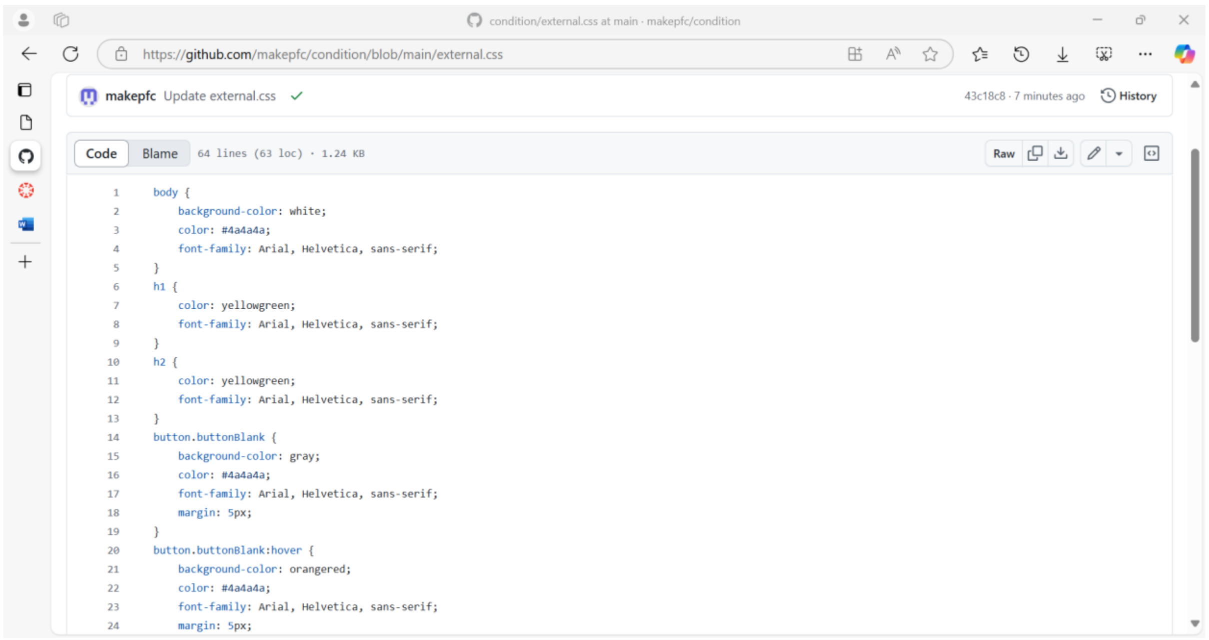
```
        function cD() {
          var cDSelected  = document.getElementById("cD");
          if (cDSelected.innerHTML.match("The cover has some small pieces of colour
missing. ")) {
            document.getElementById("cD").innerHTML = "";
            document.getElementById("cDButton").className = "buttonBlank";
          } else {
            document.getElementById("cD").innerHTML = "The cover has some small
pieces of colour missing. ";
            document.getElementById("cDButton").className = "buttonSelected";
          }
        }
      /* Start of Function to display/hide text output & style button as
selected/deselected  (onclick): for Section 4. */
        function pA() {
          var pASelected  = document.getElementById("pA");
          if (pASelected.innerHTML.match("The book is inscribed by the previous owner
in ink. ")) {
            document.getElementById("pA").innerHTML = "";
            document.getElementById("pAButton").className = "buttonBlank";
          } else {
            document.getElementById("pA").innerHTML = "The book is inscribed by the
previous owner in ink. ";
            document.getElementById("pAButton").className = "buttonSelected";
          }
        }
        function pB() {
          var pBSelected  = document.getElementById("pB");
          if (pBSelected.innerHTML.match("The book has clean, unmarked pages. ")) {
            document.getElementById("pB").innerHTML = "";
            document.getElementById("pBButton").className = "buttonBlank";
          } else {
            document.getElementById("pB").innerHTML = "The book has clean,
unmarked pages. ";
            document.getElementById("pBButton").className = "buttonSelected";
          }
        }
        function pC() {
          var pCSelected  = document.getElementById("pC");
          if (pCSelected.innerHTML.match("The book has uncreased pages. ")) {
            document.getElementById("pC").innerHTML = "";
            document.getElementById("pCButton").className = "buttonBlank";
          } else {
            document.getElementById("pC").innerHTML = "The book has uncreased
pages. ";
            document.getElementById("pCButton").className = "buttonSelected";
          }
        }
        function pD() {
          var pDSelected  = document.getElementById("pD");
          if (pDSelected.innerHTML.match("The pages have some water damage. ")) {
            document.getElementById("pD").innerHTML = "";
            document.getElementById("pDButton").className = "buttonBlank";
          } else {
            document.getElementById("pD").innerHTML = "The pages have some water
damage. ";
            document.getElementById("pDButton").className = "buttonSelected";
          }
        }
        function pE() {
          var pESelected  = document.getElementById("pE");
          if (pESelected.innerHTML.match("The pages show some foxing. ")) {
            document.getElementById("pE").innerHTML = "";
            document.getElementById("pEButton").className = "buttonBlank";
          } else {
            document.getElementById("pE").innerHTML = "The pages show some
foxing. ";
            document.getElementById("pEButton").className = "buttonSelected";
          }
        }
        function pF() {
          var pFSelected  = document.getElementById("pF");
          if (pFSelected.innerHTML.match("The book is inscribed by the previous owner
in pencil. ")) {
            document.getElementById("pF").innerHTML = "";
            document.getElementById("pFButton").className = "buttonBlank";
          } else {
            document.getElementById("pF").innerHTML = "The book is inscribed by the
previous owner in pencil. ";
            document.getElementById("pFButton").className = "buttonSelected";
          }
        }
        function pG() {
          var pGSelected  = document.getElementById("pG");
          if (pGSelected.innerHTML.match("The pages are somewhat discoloured with
age. ")) {
            document.getElementById("pG").innerHTML = "";
            document.getElementById("pGButton").className = "buttonBlank";
        } else {
            document.getElementById("pG").innerHTML = "The pages are somewhat
discoloured with age. ";
            document.getElementById("pGButton").className = "buttonSelected";
        }
    }
    function pH() {
        var pHSelected  = document.getElementById("pH");
        if (pHSelected.innerHTML.match("Some pages contain annotations in ink. ")) {
            document.getElementById("pH").innerHTML = "";
            document.getElementById("pHButton").className = "buttonBlank";
        } else {
            document.getElementById("pH").innerHTML = "Some pages contain
annotations in ink. ";
            document.getElementById("pHButton").className = "buttonSelected";
        }
    }
    function pI() {
        var pISelected  = document.getElementById("pI");
        if (pISelected.innerHTML.match("Some pages contain annotations in pencil.
")) {
            document.getElementById("pI").innerHTML = "";
            document.getElementById("pIButton").className = "buttonBlank";
        } else {
            document.getElementById("pI").innerHTML = "Some pages contain
annotations in pencil. ";
            document.getElementById("pIButton").className = "buttonSelected";
        }
        }
        function pJ() {
          var pJSelected  = document.getElementById("pJ");
          if (pJSelected.innerHTML.match("The interior pages are free from annotations.
")) {
            document.getElementById("pJ").innerHTML = "";
            document.getElementById("pJButton").className = "buttonBlank";
          } else {
            document.getElementById("pJ").innerHTML = "The interior pages are free
from annotations. ";
            document.getElementById("pJButton").className = "buttonSelected";
          }
        }
        function pK() {
          var pKSelected  = document.getElementById("pK");
          if (pKSelected.innerHTML.match("The book has tight binding. ")) {
            document.getElementById("pK").innerHTML = "";
            document.getElementById("pKButton").className = "buttonBlank";
          } else {
            document.getElementById("pK").innerHTML = "The book has tight binding. ";
            document.getElementById("pKButton").className = "buttonSelected";
          }
        }
      /* Start of Functions to display further information on diagrams (onmouseover). */
        function mouseOverSlipcase() {
          document.getElementById("slipcaseImage").src = "slipcase_definition.png";
        }
        function mouseOutSlipcase() {
          document.getElementById("slipcaseImage").src = "slipcase.png";
        }
        function mouseOverDustJacket() {
          document.getElementById("dustJacketImage").src =
"dust_jacket_definition.png";
        }
        function mouseOutDustJacket() {
          document.getElementById("dustJacketImage").src = "dust_jacket.png";
        }
        function mouseOverCover() {
          document.getElementById("coverImage").src = "cover_definition.png";
        }
        function mouseOutCover() {
          document.getElementById("coverImage").src = "cover.png";
        }
        function mouseOverPages() {
          document.getElementById("pagesImage").src = "pages_definition.png";
        }
        function mouseOutPages() {
          document.getElementById("pagesImage").src = "pages.png";
        }
      /* Start of Functions to display further information on buttons (onmouseover): for
Section 1. */
        /* None. */
      /* Start of Functions to display further information on buttons (onmouseover): for
Section 2. */
        /* None. */
      /* Start of Functions to display further information on buttons (onmouseover): for
Section 3. */
```

```
        /* None. */

      /* Start of Functions to display further information on buttons (onmouseover): for
Section 4. */

        function mouseOverPEButton() {

          document.getElementById("pagesImage").src = "foxing.png";

        }

        function mouseOutPEButton() {

          document.getElementById("pagesImage").src = "pages.png";

        }

        function mouseOverpAButton() {

          document.getElementById("pagesImage").src = "inscribed.png";

        }

        function mouseOutpAButton() {

          document.getElementById("pagesImage").src = "pages.png";

        }

        function mouseOverpFButton() {

          document.getElementById("pagesImage").src = "inscribed.png";

        }

        function mouseOutpFButton() {

          document.getElementById("pagesImage").src = "pages.png";

        }

      /* Start of Function for reset button (onclick). */

        function newDoc() {

          location.reload();

          window.open("https://makepfc.github.io/condition/#start", "_self");

        }

    </script>

    <!-- End of JavaScript functions. -->
```

```
  </body>
</html>
```

*Appendix 1B: Copy of external.css Deployed on GitHub*

Screenshots:

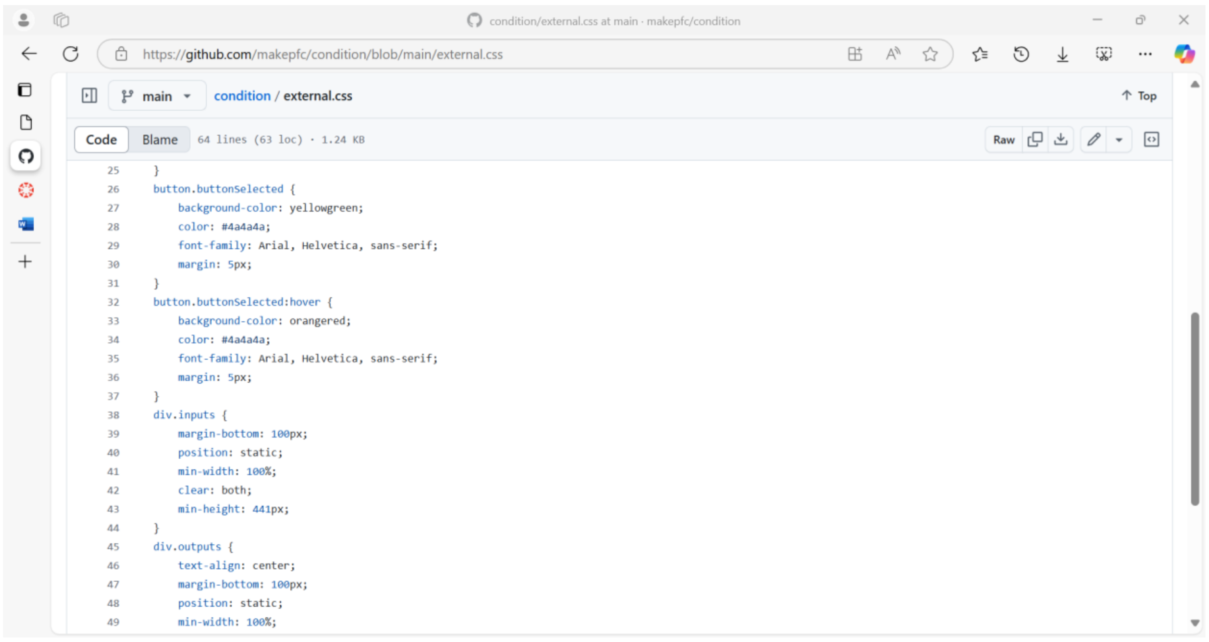

condition/external.css at main · makepfc/condition

https://github.com/makepfc/condition/blob/main/external.css

makepfc Update external.css ✓ 43c18c8 · 7 minutes ago History

Code Blame 64 lines (63 loc) · 1.24 KB Raw

```
body {
    background-color: white;
    color: #4a4a4a;
    font-family: Arial, Helvetica, sans-serif;
}
h1 {
    color: yellowgreen;
    font-family: Arial, Helvetica, sans-serif;
}
h2 {
    color: yellowgreen;
    font-family: Arial, Helvetica, sans-serif;
}
button.buttonBlank {
    background-color: gray;
    color: #4a4a4a;
    font-family: Arial, Helvetica, sans-serif;
    margin: 5px;
}
button.buttonBlank:hover {
    background-color: orangered;
    color: #4a4a4a;
    font-family: Arial, Helvetica, sans-serif;
    margin: 5px;
```

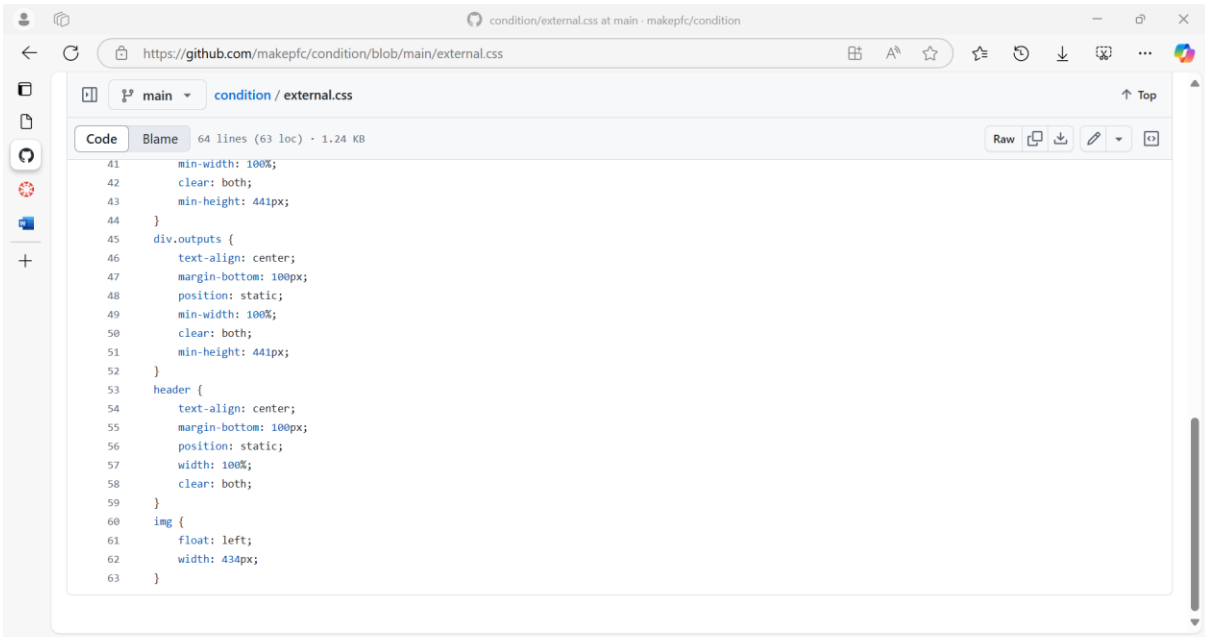

condition/external.css at main · makepfc/condition

https://github.com/makepfc/condition/blob/main/external.css

main condition / external.css Top

Code Blame 64 lines (63 loc) · 1.24 KB Raw

```
}
button.buttonSelected {
    background-color: yellowgreen;
    color: #4a4a4a;
    font-family: Arial, Helvetica, sans-serif;
    margin: 5px;
}
button.buttonSelected:hover {
    background-color: orangered;
    color: #4a4a4a;
    font-family: Arial, Helvetica, sans-serif;
    margin: 5px;
}
div.inputs {
    margin-bottom: 100px;
    position: static;
    min-width: 100%;
    clear: both;
    min-height: 441px;
}
div.outputs {
    text-align: center;
    margin-bottom: 100px;
    position: static;
    min-width: 100%;
```


condition/external.css at main · makepfc/condition
https://github.com/makepfc/condition/blob/main/external.css
main
condition / external.css
Top
Code
Blame
64 lines (63 loc) · 1.24 KB
Raw

```
        min-width: 100%;
        clear: both;
        min-height: 441px;
    }
    div.outputs {
        text-align: center;
        margin-bottom: 100px;
        position: static;
        min-width: 100%;
        clear: both;
        min-height: 441px;
    }
    header {
        text-align: center;
        margin-bottom: 100px;
        position: static;
        width: 100%;
        clear: both;
    }
    img {
        float: left;
        width: 434px;
    }
```

Text:

```
body {
  background-color: white;
  color: #4a4a4a;
  font-family: Arial, Helvetica, sans-serif;
}
h1 {
  color: yellowgreen;
  font-family: Arial, Helvetica, sans-serif;
}
h2 {
  color: yellowgreen;
  font-family: Arial, Helvetica, sans-serif;
}
button.buttonBlank {
  background-color: gray;
  color: #4a4a4a;
  font-family: Arial, Helvetica, sans-serif;
```

```
   margin: 5px;
}
button.buttonBlank:hover {
   background-color: orangered;
   color: #4a4a4a;
   font-family: Arial, Helvetica, sans-serif;
   margin: 5px;
}
button.buttonSelected {
   background-color: yellowgreen;
   color: #4a4a4a;
   font-family: Arial, Helvetica, sans-serif;
   margin: 5px;
}
button.buttonSelected:hover {
   background-color: orangered;
   color: #4a4a4a;
   font-family: Arial, Helvetica, sans-serif;
   margin: 5px;
}
div.inputs {
   margin-bottom: 100px;
   position: static;
   min-width: 100%;
   clear: both;
   min-height: 441px;
}
div.outputs {
   text-align: center;
   margin-bottom: 100px;
   position: static;
   min-width: 100%;
   clear: both;
   min-height: 441px;
}
header {
   text-align: center;
   margin-bottom: 100px;
   position: static;
   width: 100%;
   clear: both;
}
img {
   float: left;
   width: 434px;
}
```

**Appendix 2: Link to Hosted/Deployed Web Application**

https://makepfc.github.io/condition/

**Appendix 3: Additional Development and Testing Documentation**

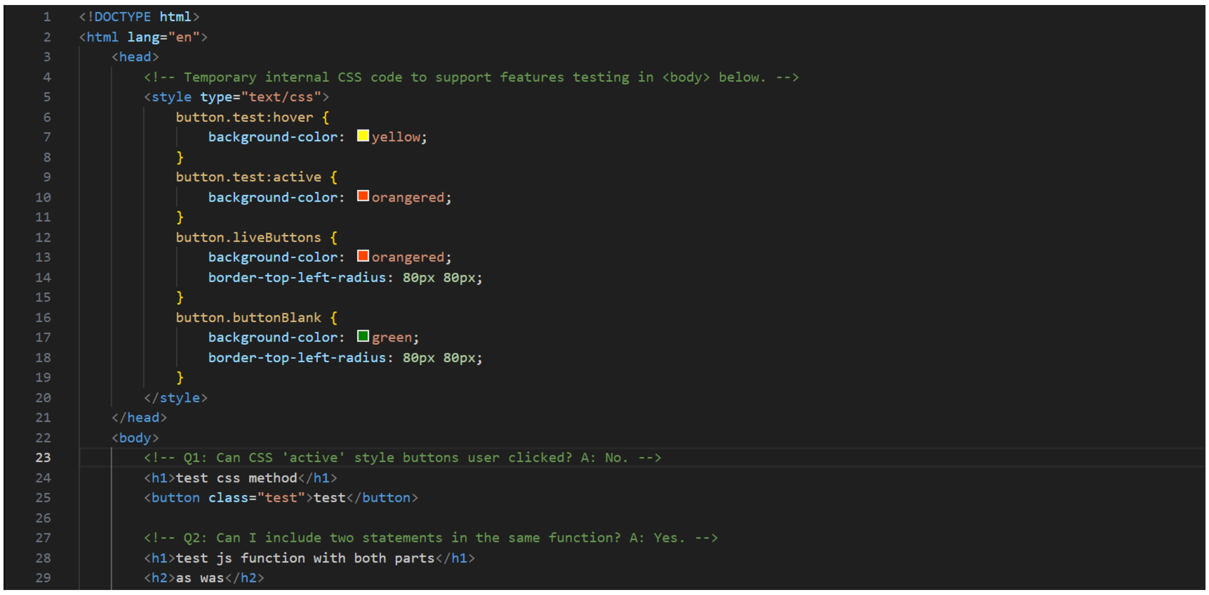
*Appendix 3A: GitHub repository*

https://github.com/makepfc/condition

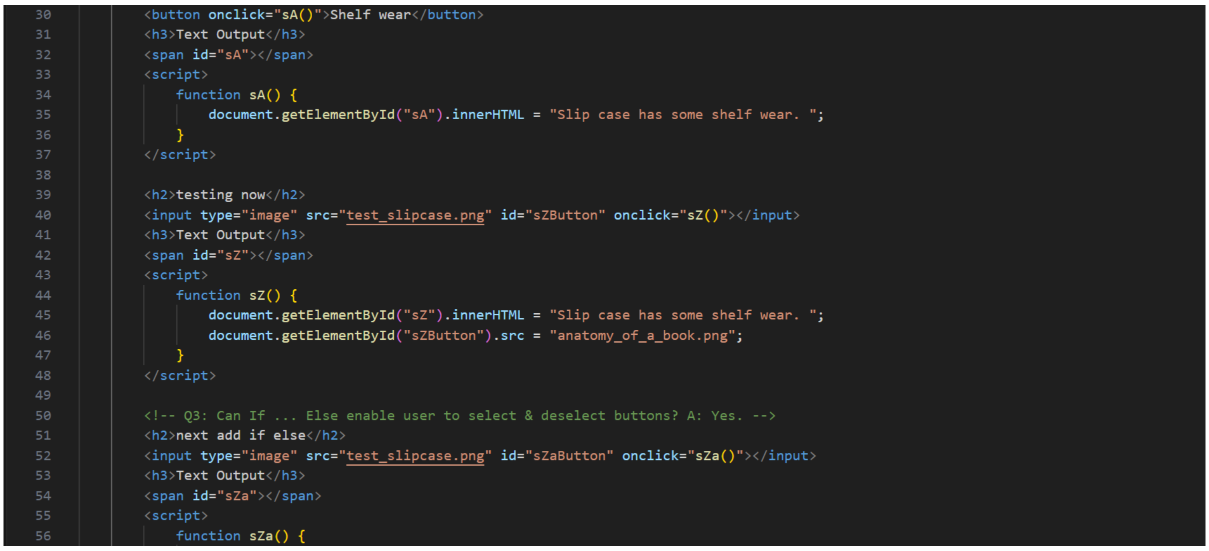
*Appendix 3B: Tested Coding of Button Interaction in Visual Studio Code:*

*test_button.html*

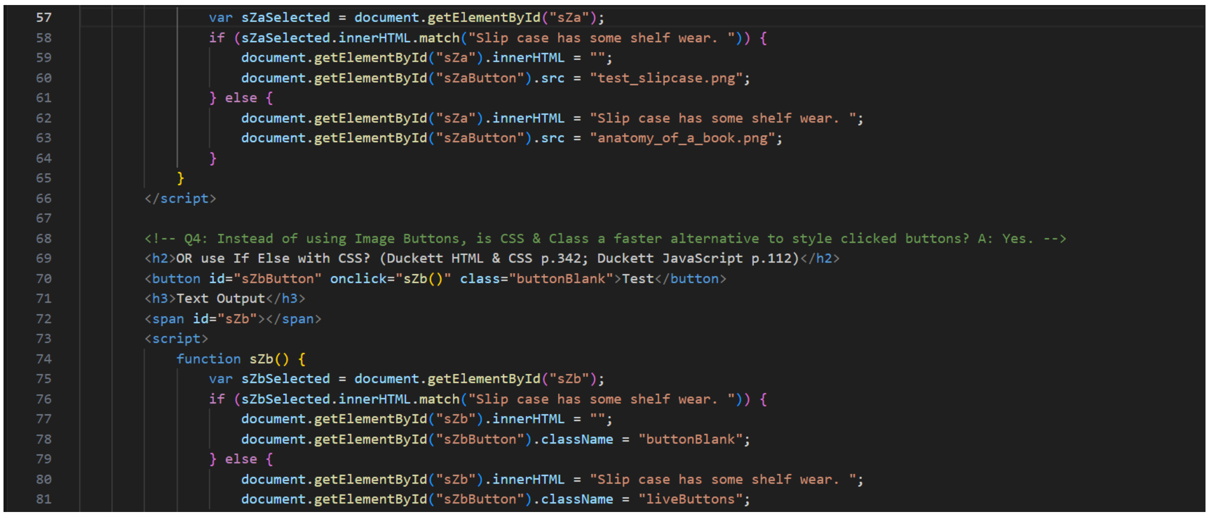
Screenshot:



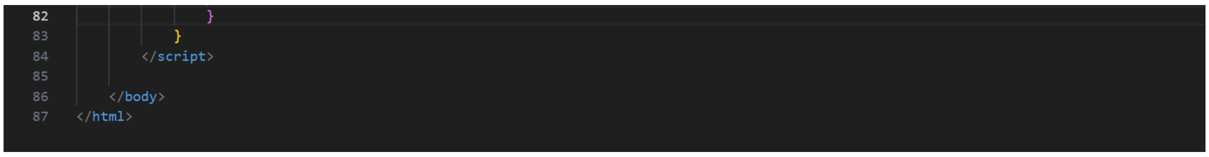
```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <!-- Temporary internal CSS code to support features testing in <body> below. -->
        <style type="text/css">
            button.test:hover {
                background-color: yellow;
            }
            button.test:active {
                background-color: orangered;
            }
            button.liveButtons {
                background-color: orangered;
                border-top-left-radius: 80px 80px;
            }
            button.buttonBlank {
                background-color: green;
                border-top-left-radius: 80px 80px;
            }
        </style>
    </head>
    <body>
        <!-- Q1: Can CSS 'active' style buttons user clicked? A: No. -->
        <h1>test css method</h1>
        <button class="test">test</button>

        <!-- Q2: Can I include two statements in the same function? A: Yes. -->
        <h1>test js function with both parts</h1>
        <h2>as was</h2>
```



```html
        <button onclick="sA()">Shelf wear</button>
        <h3>Text Output</h3>
        <span id="sA"></span>
        <script>
            function sA() {
                document.getElementById("sA").innerHTML = "Slip case has some shelf wear. ";
            }
        </script>

        <h2>testing now</h2>
        <input type="image" src="test_slipcase.png" id="sZButton" onclick="sZ()"></input>
        <h3>Text Output</h3>
        <span id="sZ"></span>
        <script>
            function sZ() {
                document.getElementById("sZ").innerHTML = "Slip case has some shelf wear. ";
                document.getElementById("sZButton").src = "anatomy_of_a_book.png";
            }
        </script>

        <!-- Q3: Can If ... Else enable user to select & deselect buttons? A: Yes. -->
        <h2>next add if else</h2>
        <input type="image" src="test_slipcase.png" id="sZaButton" onclick="sZa()"></input>
        <h3>Text Output</h3>
        <span id="sZa"></span>
        <script>
            function sZa() {
```

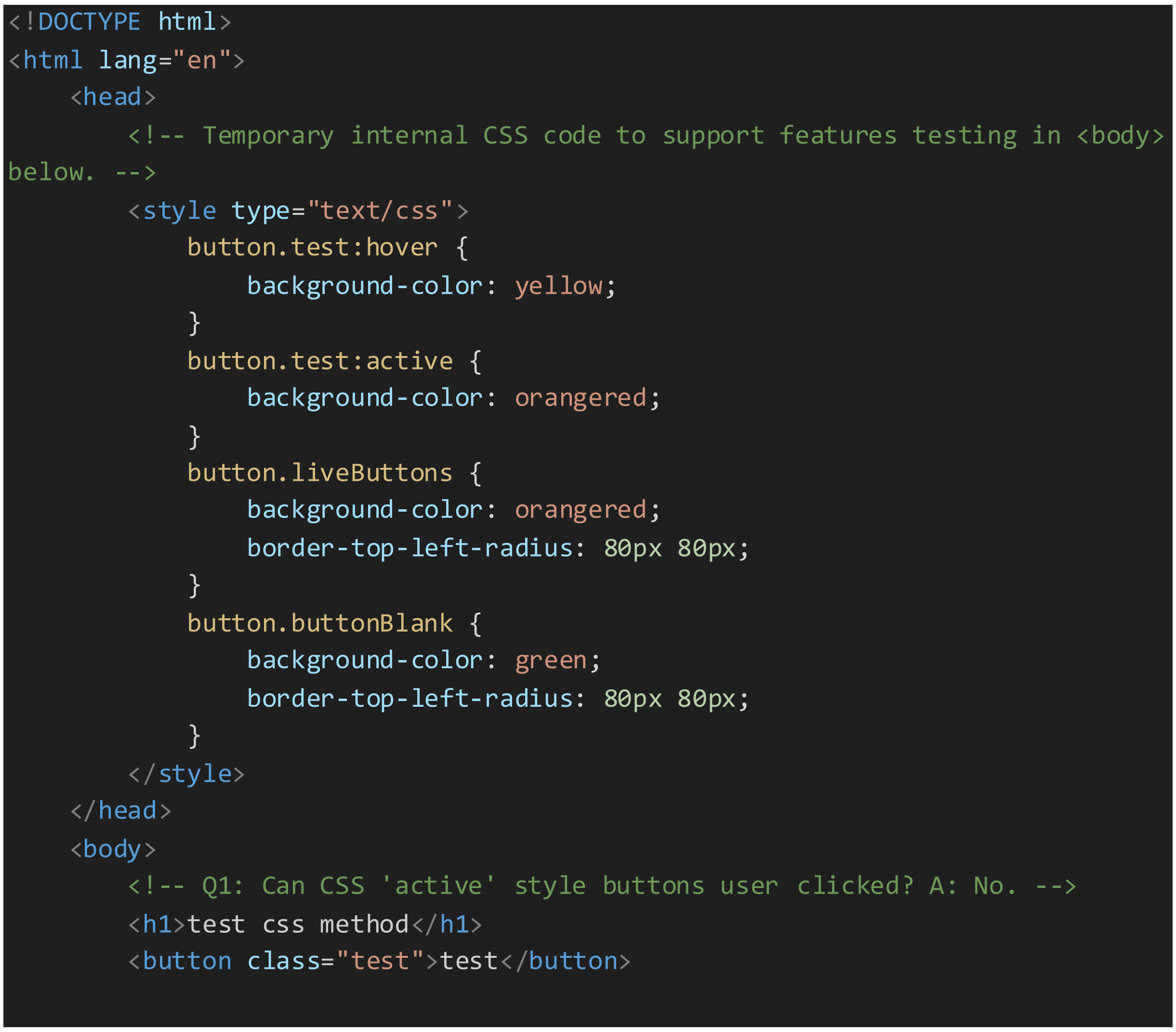


```
                var sZaSelected = document.getElementById("sZa");
                if (sZaSelected.innerHTML.match("Slip case has some shelf wear. ")) {
                    document.getElementById("sZa").innerHTML = "";
                    document.getElementById("sZaButton").src = "test_slipcase.png";
                } else {
                    document.getElementById("sZa").innerHTML = "Slip case has some shelf wear. ";
                    document.getElementById("sZaButton").src = "anatomy_of_a_book.png";
                }
            }
        </script>

        <!-- Q4: Instead of using Image Buttons, is CSS & Class a faster alternative to style clicked buttons? A: Yes. -->
        <h2>OR use If Else with CSS? (Duckett HTML & CSS p.342; Duckett JavaScript p.112)</h2>
        <button id="sZbButton" onclick="sZb()" class="buttonBlank">Test</button>
        <h3>Text Output</h3>
        <span id="sZb"></span>
        <script>
            function sZb() {
                var sZbSelected = document.getElementById("sZb");
                if (sZbSelected.innerHTML.match("Slip case has some shelf wear. ")) {
                    document.getElementById("sZb").innerHTML = "";
                    document.getElementById("sZbButton").className = "buttonBlank";
                } else {
                    document.getElementById("sZb").innerHTML = "Slip case has some shelf wear. ";
                    document.getElementById("sZbButton").className = "liveButtons";
```



```
                }
            }
        </script>

    </body>
</html>
```

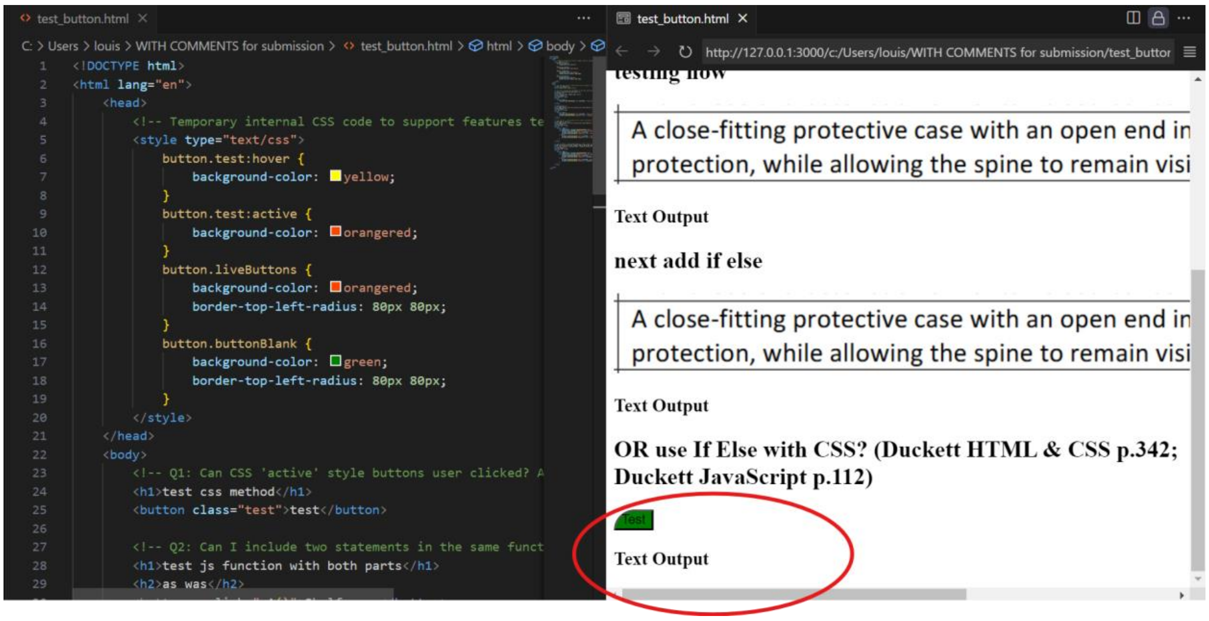
Text:



```
<!DOCTYPE html>
<html lang="en">
    <head>
        <!-- Temporary internal CSS code to support features testing in <body>
below. -->
        <style type="text/css">
            button.test:hover {
                background-color: yellow;
            }
            button.test:active {
                background-color: orangered;
            }
            button.liveButtons {
                background-color: orangered;
                border-top-left-radius: 80px 80px;
            }
            button.buttonBlank {
                background-color: green;
                border-top-left-radius: 80px 80px;
            }
        </style>
    </head>
    <body>
        <!-- Q1: Can CSS 'active' style buttons user clicked? A: No. -->
        <h1>test css method</h1>
        <button class="test">test</button>
        <!-- Q2: Can I include two statements in the same function? A: Yes. --
>
        <h1>test js function with both parts</h1>
        <h2>as was</h2>
        <button onclick="sA()">Shelf wear</button>
        <h3>Text Output</h3>
        <span id="sA"></span>
        <script>
            function sA() {
                document.getElementById("sA").innerHTML = "Slip case has some
shelf wear. ";
            }
        </script>

        <h2>testing now</h2>
        <input type="image" src="test_slipcase.png" id="sZButton"
onclick="sZ()"></input>
        <h3>Text Output</h3>
        <span id="sZ"></span>
        <script>
            function sZ() {
                document.getElementById("sZ").innerHTML = "Slip case has some
shelf wear. ";
                document.getElementById("sZButton").src =
"anatomy_of_a_book.png";
            }
        </script>

        <!-- Q3: Can If ... Else enable user to select & deselect buttons? A:
Yes. -->
        <h2>next add if else</h2>
        <input type="image" src="test_slipcase.png" id="sZaButton"
onclick="sZa()"></input>
        <h3>Text Output</h3>
        <span id="sZa"></span>
        <script>
            function sZa() {
                var sZaSelected = document.getElementById("sZa");
                if (sZaSelected.innerHTML.match("Slip case has some shelf
wear. ")) {
                    document.getElementById("sZa").innerHTML = "";
                    document.getElementById("sZaButton").src =
"test_slipcase.png";
                } else {
                    document.getElementById("sZa").innerHTML = "Slip case has
some shelf wear. ";
```

```
                    document.getElementById("sZaButton").src =
"anatomy_of_a_book.png";
                }
            }
        </script>

        <!-- Q4: Instead of using Image Buttons, is CSS & Class a faster
alternative to style clicked buttons? A: Yes. -->
        <h2>OR use If Else with CSS? (Duckett HTML & CSS p.342; Duckett
JavaScript p.112)</h2>
        <button id="sZbButton" onclick="sZb()"
class="buttonBlank">Test</button>
        <h3>Text Output</h3>
        <span id="sZb"></span>
        <script>
            function sZb() {
                var sZbSelected = document.getElementById("sZb");
                if (sZbSelected.innerHTML.match("Slip case has some shelf
wear. ")) {
                    document.getElementById("sZb").innerHTML = "";
                    document.getElementById("sZbButton").className =
"buttonBlank";
                } else {
                    document.getElementById("sZb").innerHTML = "Slip case has
some shelf wear. ";
                    document.getElementById("sZbButton").className =
"liveButtons";
                }
            }
        </script>

    </body>
</html>
```

Output screenshots:



test_button.html

C: > Users > louis > WITH COMMENTS for submission > test_button.html > html > body >

```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <!-- Temporary internal CSS code to support features te
        <style type="text/css">
            button.test:hover {
                background-color: yellow;
            }
            button.test:active {
                background-color: orangered;
            }
            button.liveButtons {
                background-color: orangered;
                border-top-left-radius: 80px 80px;
            }
            button.buttonBlank {
                background-color: green;
                border-top-left-radius: 80px 80px;
            }
        </style>
    </head>
    <body>
        <!-- Q1: Can CSS 'active' style buttons user clicked? A
        <h1>test css method</h1>
        <button class="test">test</button>

        <!-- Q2: Can I include two statements in the same funct
        <h1>test js function with both parts</h1>
        <h2>as was</h2>
```

test_button.html

http://127.0.0.1:3000/c:/Users/louis/WITH COMMENTS for submission/test_buttor

testing now

A close-fitting protective case with an open end in
protection, while allowing the spine to remain visi

Text Output

next add if else

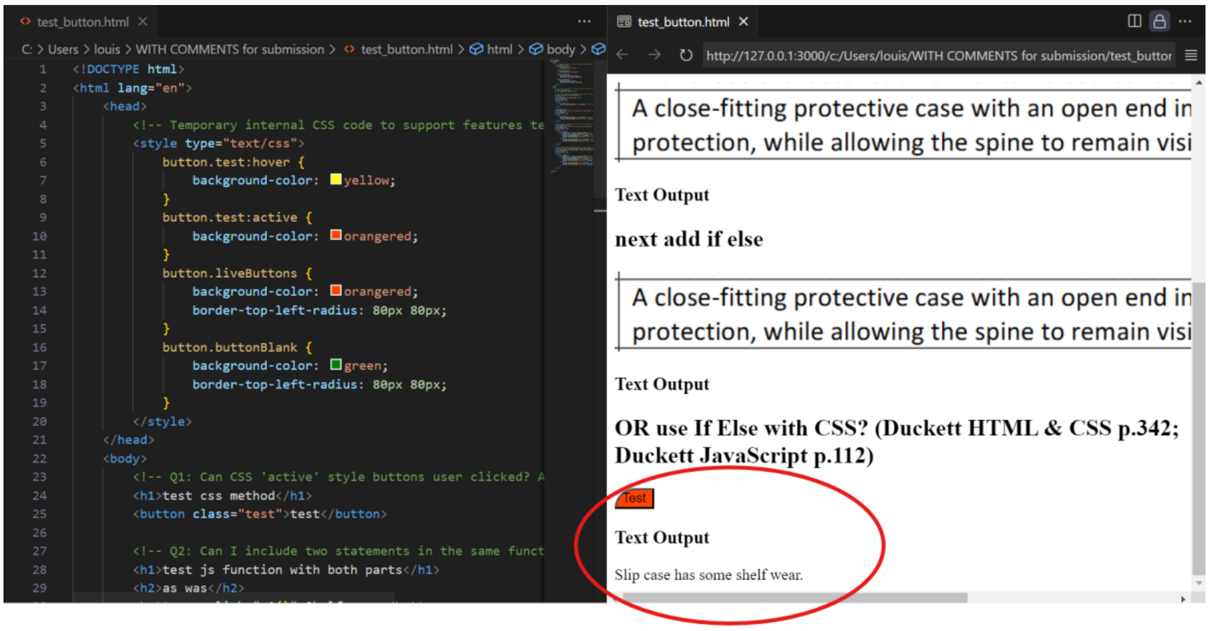
A close-fitting protective case with an open end in
protection, while allowing the spine to remain visi

Text Output

OR use If Else with CSS? (Duckett HTML & CSS p.342; Duckett JavaScript p.112)

Test

Text Output



test_button.html

C: > Users > louis > WITH COMMENTS for submission > test_button.html > html > body >

```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <!-- Temporary internal CSS code to support features te
        <style type="text/css">
            button.test:hover {
                background-color: yellow;
            }
            button.test:active {
                background-color: orangered;
            }
            button.liveButtons {
                background-color: orangered;
                border-top-left-radius: 80px 80px;
            }
            button.buttonBlank {
                background-color: green;
                border-top-left-radius: 80px 80px;
            }
        </style>
    </head>
    <body>
        <!-- Q1: Can CSS 'active' style buttons user clicked? A
        <h1>test css method</h1>
        <button class="test">test</button>

        <!-- Q2: Can I include two statements in the same funct
        <h1>test js function with both parts</h1>
        <h2>as was</h2>
```

test_button.html

http://127.0.0.1:3000/c:/Users/louis/WITH COMMENTS for submission/test_buttor

A close-fitting protective case with an open end in
protection, while allowing the spine to remain visi

Text Output

next add if else

A close-fitting protective case with an open end in
protection, while allowing the spine to remain visi

Text Output

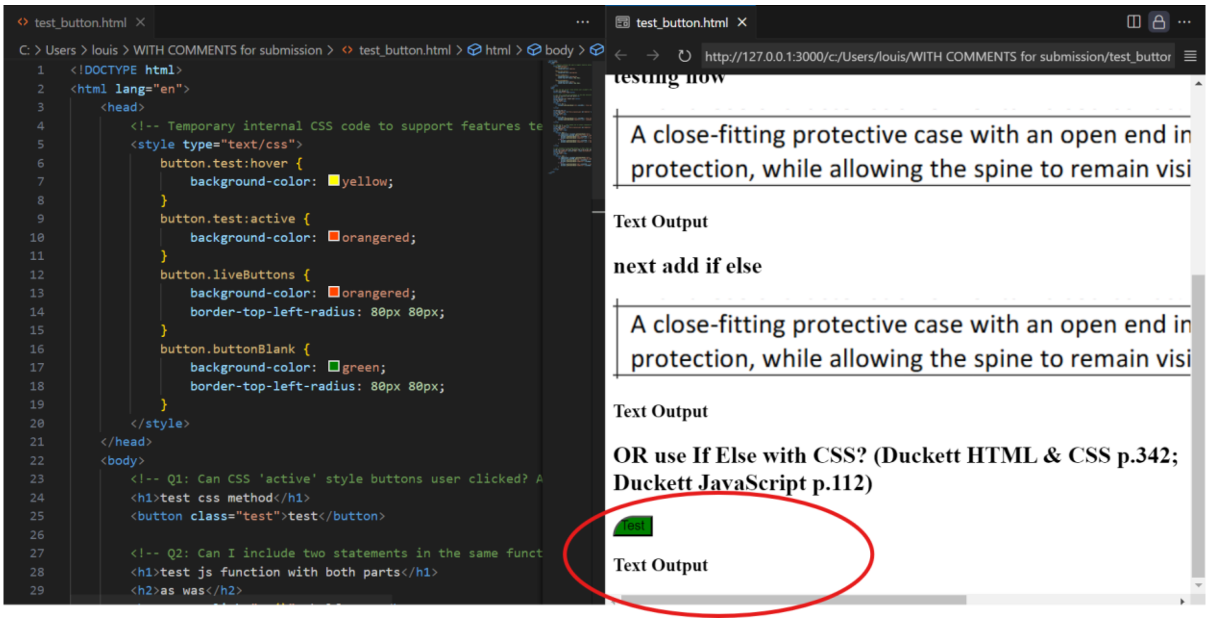
OR use If Else with CSS? (Duckett HTML & CSS p.342; Duckett JavaScript p.112)

Test

Text Output

Slip case has some shelf wear.

*Second click changed button styling and text output back (as required):*



test_button.html

C: > Users > louis > WITH COMMENTS for submission > test_button.html > html > body >

```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <!-- Temporary internal CSS code to support features te
        <style type="text/css">
            button.test:hover {
                background-color: yellow;
            }
            button.test:active {
                background-color: orangered;
            }
            button.liveButtons {
                background-color: orangered;
                border-top-left-radius: 80px 80px;
            }
            button.buttonBlank {
                background-color: green;
                border-top-left-radius: 80px 80px;
            }
        </style>
    </head>
    <body>
        <!-- Q1: Can CSS 'active' style buttons user clicked? A
        <h1>test css method</h1>
        <button class="test">test</button>

        <!-- Q2: Can I include two statements in the same funct
        <h1>test js function with both parts</h1>
        <h2>as was</h2>
```

test_button.html

http://127.0.0.1:3000/c:/Users/louis/WITH COMMENTS for submission/test_buttor

testing now

A close-fitting protective case with an open end in
protection, while allowing the spine to remain visi

Text Output

next add if else

A close-fitting protective case with an open end in
protection, while allowing the spine to remain visi

Text Output

OR use If Else with CSS? (Duckett HTML & CSS p.342; Duckett JavaScript p.112)

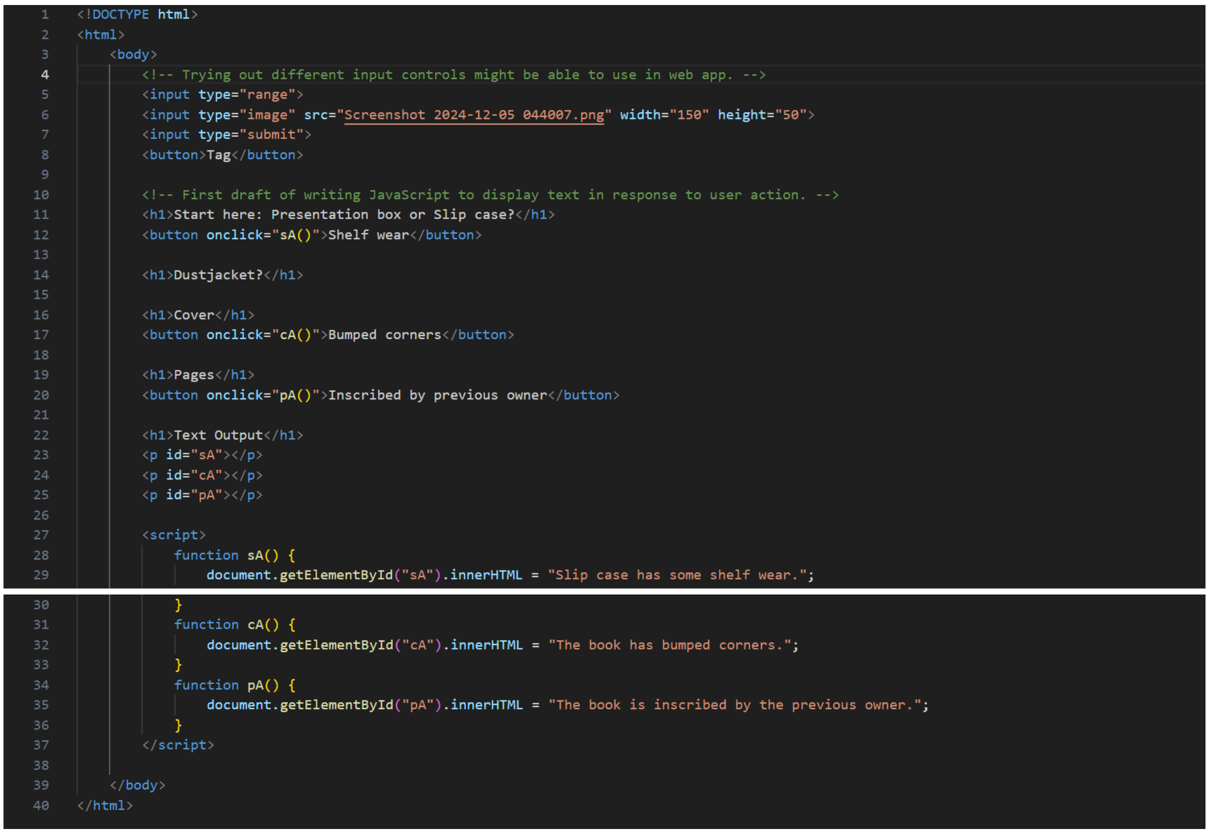
Test

Text Output

*__Appendix 3C: Tested Writing Interactive JavaScript in Visual Studio Code:__*

*__Test_option_three.html__*

Screenshot:

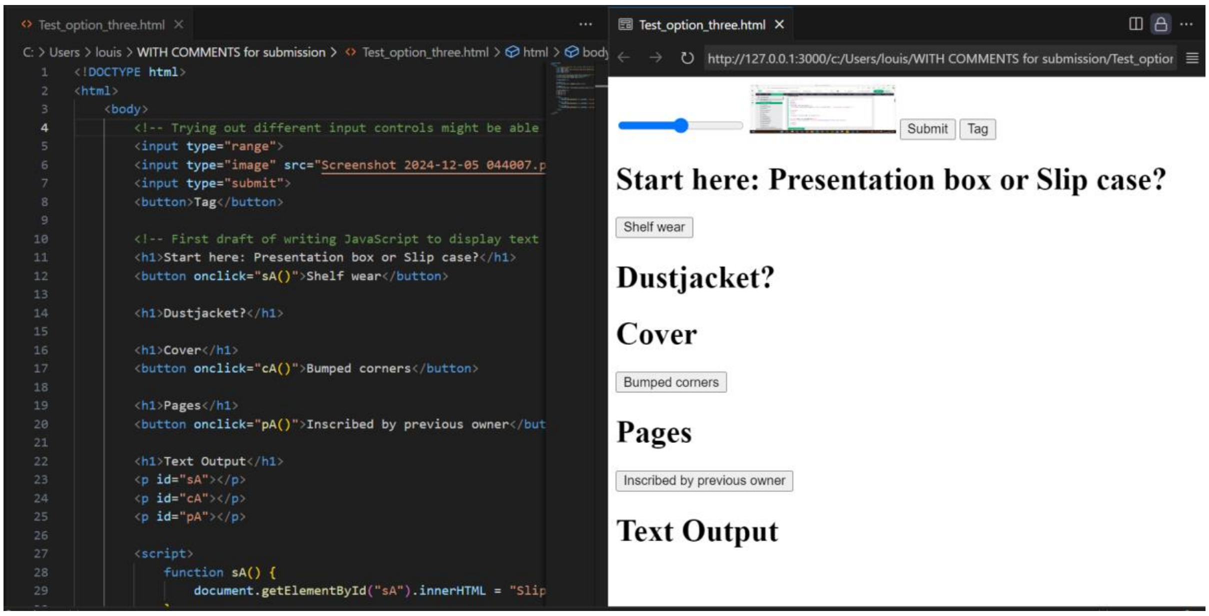


```html
<!DOCTYPE html>
<html>
    <body>
        <!-- Trying out different input controls might be able to use in web app. -->
        <input type="range">
        <input type="image" src="Screenshot 2024-12-05 044007.png" width="150" height="50">
        <input type="submit">
        <button>Tag</button>

        <!-- First draft of writing JavaScript to display text in response to user action. -->
        <h1>Start here: Presentation box or Slip case?</h1>
        <button onclick="sA()">Shelf wear</button>

        <h1>Dustjacket?</h1>

        <h1>Cover</h1>
        <button onclick="cA()">Bumped corners</button>

        <h1>Pages</h1>
        <button onclick="pA()">Inscribed by previous owner</button>

        <h1>Text Output</h1>
        <p id="sA"></p>
        <p id="cA"></p>
        <p id="pA"></p>

        <script>
            function sA() {
                document.getElementById("sA").innerHTML = "Slip case has some shelf wear.";
            }
            function cA() {
                document.getElementById("cA").innerHTML = "The book has bumped corners.";
            }
            function pA() {
                document.getElementById("pA").innerHTML = "The book is inscribed by the previous owner.";
            }
        </script>

    </body>
</html>
```

Text:

```html
<!DOCTYPE html>
<html>
    <body>
        <!-- Trying out different input controls might be able to use in web app. -->
        <input type="range">
        <input type="image" src="Screenshot 2024-12-05 044007.png" width="150" height="50">
        <input type="submit">
        <button>Tag</button>

        <!-- First draft of writing JavaScript to display text in response to user action. -->
        <h1>Start here: Presentation box or Slip case?</h1>
        <button onclick="sA()">Shelf wear</button>

        <h1>Dustjacket?</h1>

        <h1>Cover</h1>
        <button onclick="cA()">Bumped corners</button>

        <h1>Pages</h1>
        <button onclick="pA()">Inscribed by previous owner</button>

        <h1>Text Output</h1>
        <p id="sA"></p>
        <p id="cA"></p>
        <p id="pA"></p>

        <script>
            function sA() {
                document.getElementById("sA").innerHTML = "Slip case has some shelf wear.";
            }
            function cA() {
                document.getElementById("cA").innerHTML = "The book has bumped corners.";
            }
            function pA() {
                document.getElementById("pA").innerHTML = "The book is inscribed by the previous owner.";
            }
        </script>
    </body>
</html>
```

Output screenshots:



Test_option_three.html

C: > Users > louis > WITH COMMENTS for submission > Test_option_three.html > html > body

```
<!DOCTYPE html>
<html>
    <body>
        <!-- Trying out different input controls might be able
        <input type="range">
        <input type="image" src="Screenshot 2024-12-05 044007.p
        <input type="submit">
        <button>Tag</button>

        <!-- First draft of writing JavaScript to display text
        <h1>Start here: Presentation box or Slip case?</h1>
        <button onclick="sA()">Shelf wear</button>

        <h1>Dustjacket?</h1>

        <h1>Cover</h1>
        <button onclick="cA()">Bumped corners</button>

        <h1>Pages</h1>
        <button onclick="pA()">Inscribed by previous owner</but

        <h1>Text Output</h1>
        <p id="sA"></p>
        <p id="cA"></p>
        <p id="pA"></p>

        <script>
            function sA() {
                document.getElementById("sA").innerHTML = "Slip
```

http://127.0.0.1:3000/c:/Users/louis/WITH COMMENTS for submission/Test_optior

Submit Tag

**Start here: Presentation box or Slip case?**

Shelf wear

**Dustjacket?**

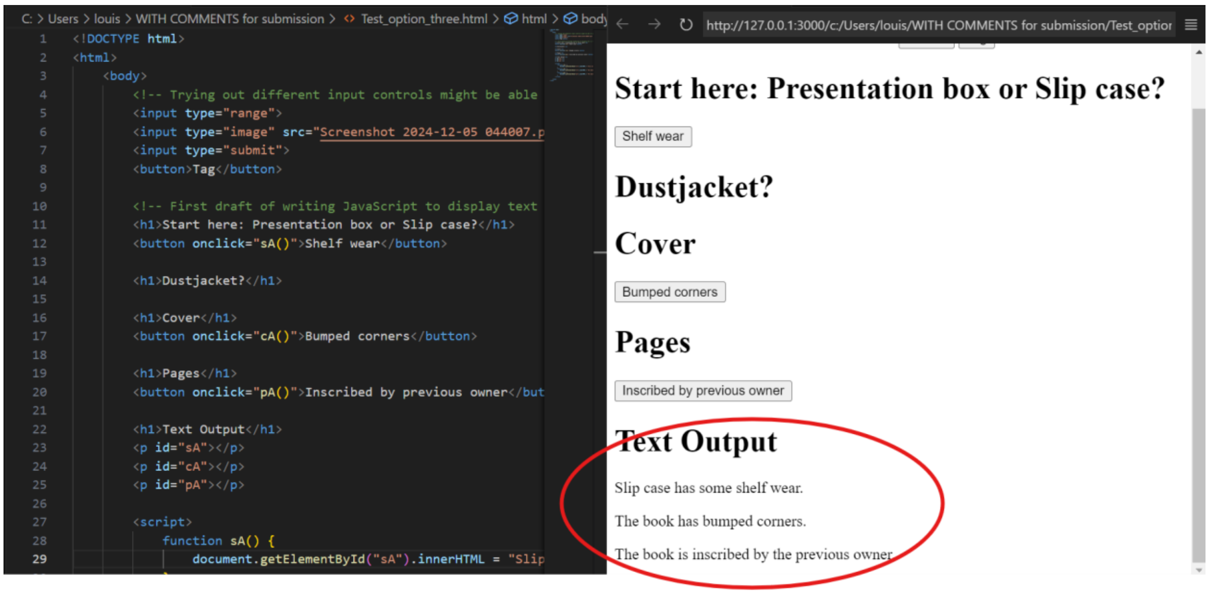
**Cover**

Bumped corners

**Pages**

Inscribed by previous owner

**Text Output**



C: > Users > louis > WITH COMMENTS for submission > Test_option_three.html > html > body

```
<!DOCTYPE html>
<html>
    <body>
        <!-- Trying out different input controls might be able
        <input type="range">
        <input type="image" src="Screenshot 2024-12-05 044007.p
        <input type="submit">
        <button>Tag</button>

        <!-- First draft of writing JavaScript to display text
        <h1>Start here: Presentation box or Slip case?</h1>
        <button onclick="sA()">Shelf wear</button>

        <h1>Dustjacket?</h1>

        <h1>Cover</h1>
        <button onclick="cA()">Bumped corners</button>

        <h1>Pages</h1>
        <button onclick="pA()">Inscribed by previous owner</but

        <h1>Text Output</h1>
        <p id="sA"></p>
        <p id="cA"></p>
        <p id="pA"></p>

        <script>
            function sA() {
                document.getElementById("sA").innerHTML = "Slip
```

http://127.0.0.1:3000/c:/Users/louis/WITH COMMENTS for submission/Test_optior

**Start here: Presentation box or Slip case?**

Shelf wear

**Dustjacket?**

**Cover**

Bumped corners

**Pages**

Inscribed by previous owner

**Text Output**

Slip case has some shelf wear.

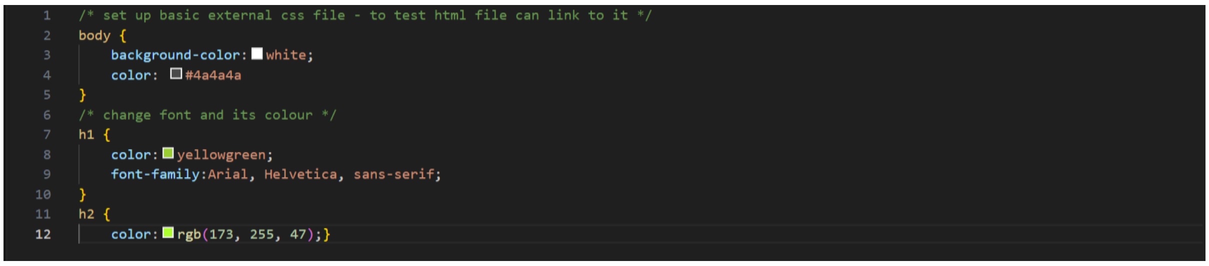
The book has bumped corners.

The book is inscribed by the previous owner

*__Appendix 3D: Drafted External CSS File in Visual Studio Code:__*

*__Assignment External CSS v1.css__*

Screenshot:



```
/* set up basic external css file - to test html file can link to it */
body {
    background-color:white;
    color: #4a4a4a
}
/* change font and its colour */
h1 {
    color:yellowgreen;
    font-family:Arial, Helvetica, sans-serif;
}
h2 {
    color:rgb(173, 255, 47);}
```

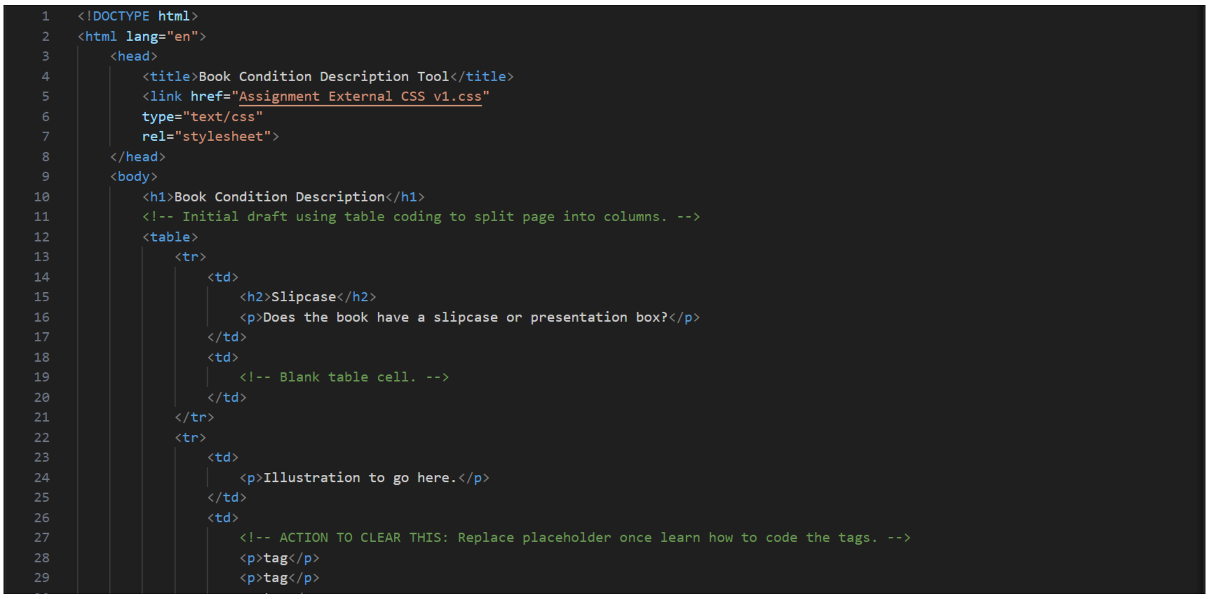
Text:

```
/* set up basic external css file - to test html file can link to it */
body {
    background-color:white;
    color: #4a4a4a
}
/* change font and its colour */
h1 {
    color:yellowgreen;
    font-family:Arial, Helvetica, sans-serif;
}
h2 {
    color:rgb(173, 255, 47);}
```

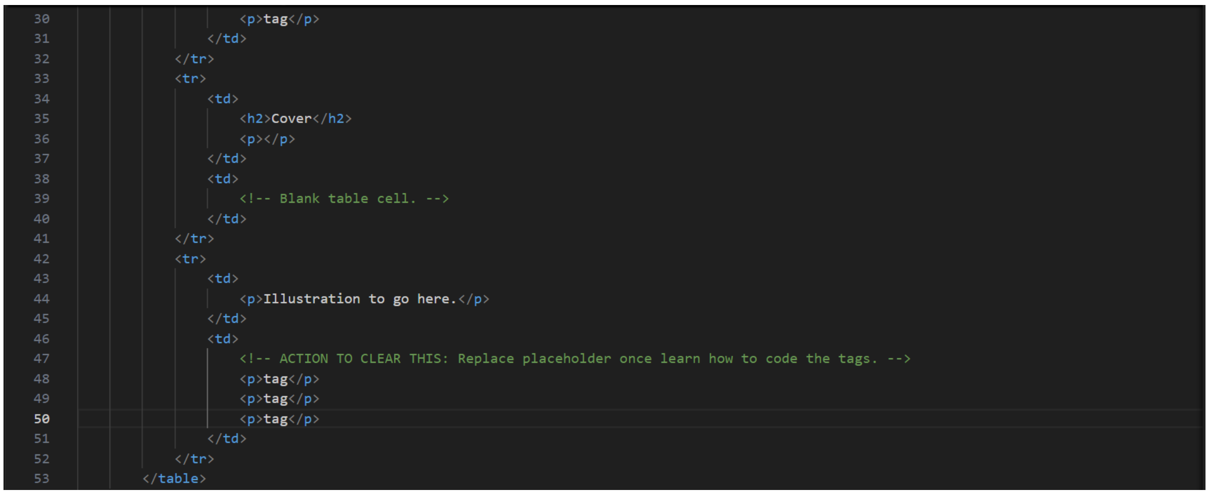
*Appendix 3E: Drafted Basic HTML Page in Visual Studio Code:*

*Assignment v1.html*

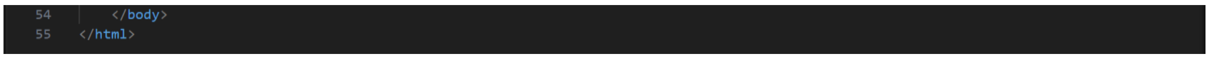
Screenshot:



```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Book Condition Description Tool</title>
        <link href="Assignment External CSS v1.css"
        type="text/css"
        rel="stylesheet">
    </head>
    <body>
        <h1>Book Condition Description</h1>
        <!-- Initial draft using table coding to split page into columns. -->
        <table>
            <tr>
                <td>
                    <h2>Slipcase</h2>
                    <p>Does the book have a slipcase or presentation box?</p>
                </td>
                <td>
                    <!-- Blank table cell. -->
                </td>
            </tr>
            <tr>
                <td>
                    <p>Illustration to go here.</p>
                </td>
                <td>
                    <!-- ACTION TO CLEAR THIS: Replace placeholder once learn how to code the tags. -->
                    <p>tag</p>
                    <p>tag</p>
```



```html
                    <p>tag</p>
                </td>
            </tr>
            <tr>
                <td>
                    <h2>Cover</h2>
                    <p></p>
                </td>
                <td>
                    <!-- Blank table cell. -->
                </td>
            </tr>
            <tr>
                <td>
                    <p>Illustration to go here.</p>
                </td>
                <td>
                    <!-- ACTION TO CLEAR THIS: Replace placeholder once learn how to code the tags. -->
                    <p>tag</p>
                    <p>tag</p>
                    <p>tag</p>
                </td>
            </tr>
        </table>
```



```html
    </body>
</html>
```

Text:

```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Book Condition Description Tool</title>
        <link href="Assignment External CSS v1.css"
        type="text/css"
        rel="stylesheet">
    </head>
    <body>
        <h1>Book Condition Description</h1>
        <!-- Initial draft using table coding to split page into columns. -->
        <table>
            <tr>
                <td>
                    <h2>Slipcase</h2>
                    <p>Does the book have a slipcase or presentation box?</p>
                </td>
                <td>
                    <!-- Blank table cell. -->
                </td>
            </tr>
            <tr>
                <td>
                    <p>Illustration to go here.</p>
                </td>
                <td>
                    <!-- ACTION TO CLEAR THIS: Replace placeholder once learn
how to code the tags. -->
                    <p>tag</p>
                    <p>tag</p>
                    <p>tag</p>
                </td>
            </tr>
            <tr>
                <td>
                    <h2>Cover</h2>
                    <p></p>
                </td>
                <td>
                    <!-- Blank table cell. -->
                </td>
            </tr>
            <tr>
                <td>
                    <p>Illustration to go here.</p>
                </td>
                <td>
                    <!-- ACTION TO CLEAR THIS: Replace placeholder once learn
how to code the tags. -->
                    <p>tag</p>
                    <p>tag</p>
                    <p>tag</p>
                </td>
            </tr>
        </table>
    </body>
</html>
```

## Reference List

Duckett, Jon, *HTML & CSS: Design and Build Websites* (Indianapolis: John Wiley & Sons, 2011).

Duckett, Jon, *JavaScript & jQuery: Interactive Front-End Web Development* (Indianapolis: John Wiley & Sons, 2014).

GitHub, *GitHub Pages* (2024). <https://pages.github.com/> [accessed 3 December 2024].

Oxfam, *Anatomy of a Book* (2024). <https://oxfam.app.box.com/s/m5oqgllxrynenjos5dv1u2snejslyq77> [accessed 29 October 2024].

Oxfam, *Glossary of Terminology: inc. Condition Descriptors* (2024). <https://oxfam.app.box.com/s/lkfr3hz4tojf1onikt659djinv9eo7qu> [accessed 29 October 2024].

Oxfam, *Making the Most of Books Online* (2024). <https://oxfam.app.box.com/s/tdcdtxxk5r6wp9aziusk5xzf94jk9apq/file/1677815490401> [accessed 29 October 2024].

Oxfam Shop, *Browse a Selection of Products From This Shop Online* (2024). <https://onlineshop.oxfam.org.uk/searchresults?N=&type=search&Nf=sku.listPrice|GT+0&Ns=product.creationDate|1&Ntk=&Nr=AND(product.active:1,NOT(sku.listPrice:0.000000),product.ox_shopid:F8081)&No=0&Nrpp=30> [accessed 13 December 2024].

W3Schools, *Change the Source of the Lightbulb* (2024). <https://www.w3schools.com/js/tryit.asp?filename=tryjs_lightbulb> [accessed 4 December 2024].

W3Schools, *HTML Block and Inline Elements* (2024). <https://www.w3schools.com/html/html_blocks.asp> [accessed 17 December 2024].

W3Schools, *HTML <span> Tag* (2024). <https://www.w3schools.com/tags/tag_span.asp> [accessed 15 December 2024].

W3Schools, *JavaScript HTML DOM Events* (2024). <https://www.w3schools.com/js/js_htmldom_events.asp> [accessed 4 December 2024].

W3Schools, *JavaScript Where To* (2024). <https://www.w3schools.com/js/js_whereto.asp> [accessed 4 December 2024].

W3Schools, *Window location.reload()* (2024). <https://www.w3schools.com/jsref/met_loc_reload.asp> [accessed 17 December 2024].

W3Schools, *Window open()* (2024). <https://www.w3schools.com/jsref/met_win_open.asp> [accessed 17 December 2024].