<u>Part IV:</u> Part V: Project Documentation

A. Overview (1 Page)

Briefly describe your product website in a way anyone can understand.
What does your site do?
What pages or features did you build?
What's the core message, purpose or experience you designed for users?
Avoid heavy technical jargon here. Think like you're explaining your project to someone with no coding background.

My product is a website that helps you find the right gas station for you. After uploading your destination, it analyzes your route, finds gas station options and gives you their gas price, how much time it adds to your route and where it is in your trip. Then you have all the info to choose which one you want to go to and you can start navigation. Makes you save money and time.

B. Coding Approach & Technical Decisions (1-2 Pages)

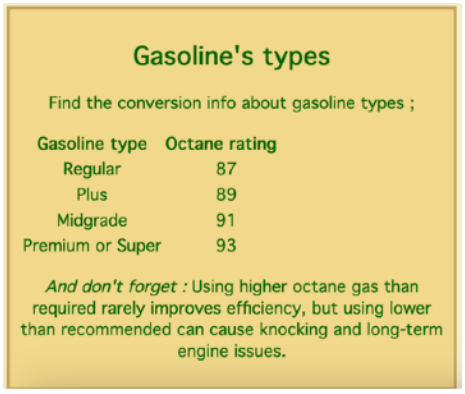
Explain how you built your site and why you coded things the way you did. Consider addressing:
How you organized your HTML, CSS and JavaScript files
Reusable patterns you built (styles, functions, etc.)
Why you chose certain layouts, styles or JS techniques
Any decisions you made to simplify, optimize or improve your code
This section highlights your thought process as a developer.

My website has 4 main pages for now. I started to build the welcome page that gives the user information on how to use it. After, I built the upload page where the user can fill out the form that ask him his route and the gas type he needs for his car. Then, I built the list page that features all the station options that the website has found on his route and the last page shows full info of a specific station. I impleted one main CSS that is quite long. Another one very short is used for the display of the stations in the list page. I added the latest JavaScript files.
I used functions to display the informations of the station that are stored in arrays. One also calculates the time to go based on the distance. I also used it to generates the footer and the header boxes to simplify the modifications you want to make and also avoid repetitions of the same code. And finally, I also added some events, especially to point out if the user didn't select any gas type and be sure it is normal. I also used object to manage the favorite feature that color in read a heart that you can click on for each station.

C. Course Concepts Integration (1 Page)

Describe how you applied concepts from class throughout your site. You may include examples from:
Design (SDLC planning, site map, wireframes)
HTML (semantic structure, content)
CSS (visual hierarchy, layout systems, responsive design)
JavaScript (dynamic updates, interactive features, event listeners, form handling)
Accessibility (alt text, color contrast, keyboard-friendly navigation)
Usability & UX (clear navigation, consistent structure, thoughtful content)
This shows how your project connects to the skills you learned.

HTML : Here we can see the table I created and the HMTL code related. I used the concept seen in class, using table row, data and heading. I also used different layout style like .



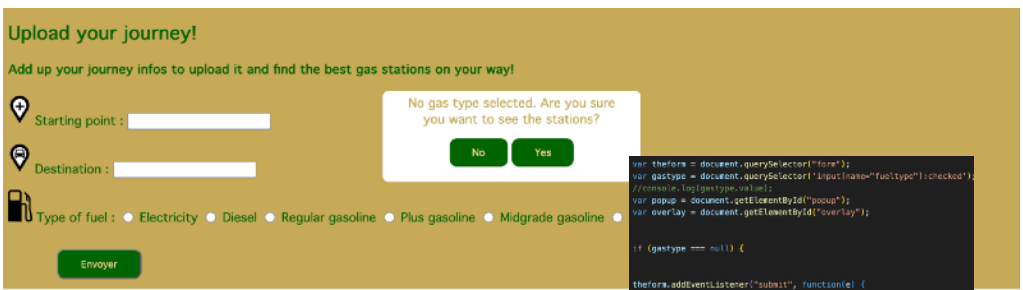

```
<div class="minor">
    <h2>Gasoline's types</h2>
    <p>Find the conversion info about gasoline types ;</p>
        <table>
            <tr><th>Gasoline type</th><th> Octane rating</th></tr>
            <tr><td>Regular </td><td>  87</td></tr>
            <tr><td>Plus </td><td>  89</td></tr>
            <tr><td>Midgrade </td><td>  91</td></tr>
            <tr><td>Premium or Super </td><td>  93</td></tr>

        </table>
        <p><em>And don't forget :</em> Using higher octane gas than required rarely
            improves efficiency, but using lower than recommended can cause knocking
            and long-term engine issues.</p>
</div>
```

CSS : I used the box model seen in class to create for example the header. The class searcher sets the green color and justify-content makes the different links evenly spaced across the page. The links are styled and if hovered, a border appears to give feedback to the user by darkening it. The other class .active change the style of the active page to show the user which page he is currently on.


```
/*Searchbar definition */
.searchbar{background-color:darkgreen;
    display:flex;
    padding:15px;
    justify-content:space-between;
    width:100%;
    margin: 0px;
    box-sizing: border-box;
}
.searchbar a{color:#f1d88a;
    font-size: 20px;
    text-decoration:none;
    padding:12px;
    text-align: center;
    font-weight: bolder;
    margin:0px 30px;
}
.searchbar a:hover {color:#C6A45D;
        border:solid 1px #C6A45D;
        font-size: 20px;
        cursor : pointer;
}
.active{ color:#C6A45D;
    border:solid 1px #f1d88a;
    font-size: 20px;
    text-decoration:none;
    padding:100px;
    text-align: center;
}
```

JavaScript : Here, it is shown how the pop-up box was coded. I used an EventListener to trigger the box to appear when the button submit is hit. For the No and Yes buttons, I used Event Handling that call 2 different functions that display or hide the pop-up.




```
<div id="errorMessage"></div>

<div id="popup">
No gas type selected. Are you sure you want to see the stations? <br>
<div class="smallbutton" onclick="closePopup()">No</div>
<div class="smallbutton" onclick="submitform()">Yes</div>
</div>

</div>
```

```
var theform = document.querySelector("form");
var gastype = document.querySelector('input[name="fueltype"]:checked');
//console.log(gastype.value);
var popup = document.getElementById("popup");
var overlay = document.getElementById("overlay");


if (gastype === null) {


theform.addEventListener("submit", function(e) {
        e.preventDefault(); // Prevent actual form submission
        popup.style.display = "block";
        overlay.style.display = "block";
    });


function closePopup() {
        popup.style.display = "none";
        overlay.style.display = "none";
    }

function submitform() {
        theform.submit();
        console.log("form submitted");
    }

}
```

My website adapts to the width of the screen and some fonts shrink at the same time. Also, the stations in the list stacks when it becomes to small, as well for the 3 footers on the upload page.
My website's photo all have clear alternative text and the website is navigable using tab.

D. Challenges & Problem-Solving (1 Page)
Walk through the most significant challenges you faced while building or improving your site. For each challenge, explain:
What the problem was
What you tried
How you ultimately solved or improved it
What you learned from the process
This is where you demonstrate real growth and persistence.

One main challenge that I faced was setting up the filter based on the type of fuel selected. This block of code's purpose is to hide the gas stations that do not have the fuel type selected by the user. To do that, we check if the price for that fuel type is 0 (meaning not available) and we hide the whole div card for that gas station. One challenge was to find how to convert a string to a variable name to access the right array of prices. I found that using window[] works and converts the string to the variable name. Another challenge was to hide the whole div card, which I did by selecting all the divs with the class "list" and changing their display style to 'none'.
This challenge made me learn how to troubleshoot more efficiently by testing smaller parts of the code one at a time instead of trying to fix everything at once. This made the debugging process much easier and helped me see exactly where things were going wrong. This way of debugging code is really educative in my daily basis because it is very rational and makes you isolate smaller parts to find the problem and fix it.

```
  var selectedFuel = document.querySelector("#fuel-filter");
  console.log(selectedFuel.value + " selectedFuel");

/ we check which fuel type is selected and update the price accordingly by displaying the correct variable from the
  if (selectedFuel.value === 'All'){
      for (var k=0; k<names.length; k++){
          document.querySelectorAll('p.price')[k].innerText = "Select a fuel type to see prices";}
  }

  else  { // a fuel type is selected

      console.log(window[selectedFuel.value] + "FuelPrice");  //window[] converts string to variable name
     // document.querySelectorAll('p.price')[0].innerText =  "Syoooo";
      for (var j=0; j<names.length; j++){
          console.log("j=" + j);
          if  (window[selectedFuel.value][j]=="0") {
                  document.querySelectorAll("div.list")[j].style.display = 'none';
                  console.log("hiding card");
          }
          else  {
              document.querySelectorAll('p.price')[j].innerText = selectedFuel.value + " price : $" + window[selec

      }
  }
```

E. Strengths & Areas For Improvement (1 Page)

Reflect honestly on your final site:
What part of your project are you most proud of? Why?
What part feels weaker or needs more work?
If you had more time, what would you continue improving? Recommend realistic ideas for moving your site forward.

I'm most proud of my site's clean layout and the interactive features, especially the filtering system, the ability to favorite stations, and the detailed info pages. At the same time, the site is still improvable, particularly for the other pages that I didn't have time to develop such as the favorites page and the settings page. I would also improve the lack of real station data, and several peers suggested adding actual photos to make the stations more recognizable, which I just did and it adds realism to the website. If I had more time, I would continue connecting it to a real database so users can browse actual stations along their route.