Michael Stiffler
2731139
03/22/2022

CIS 408 LAB 3

Website = http://eecs.csuohio.edu/~mistiffl/mstiff2731139.html

My file structure is set up like this:

Where my index.html is mstiff2731139.html in this case. My main.js file is under the scripts directory and my p3.css file is under my styles directory.

For my lab, I implemented the feature that allows each dropdown item when clicked to show up and then disappear if the user clicks it again. For the following screenshots, I will show this functionality while also explaining how I implemented each part. I also let the JSON be a global object in my code using: Let myObj = {JSON} with JSON being the business JSON data that was given to us.
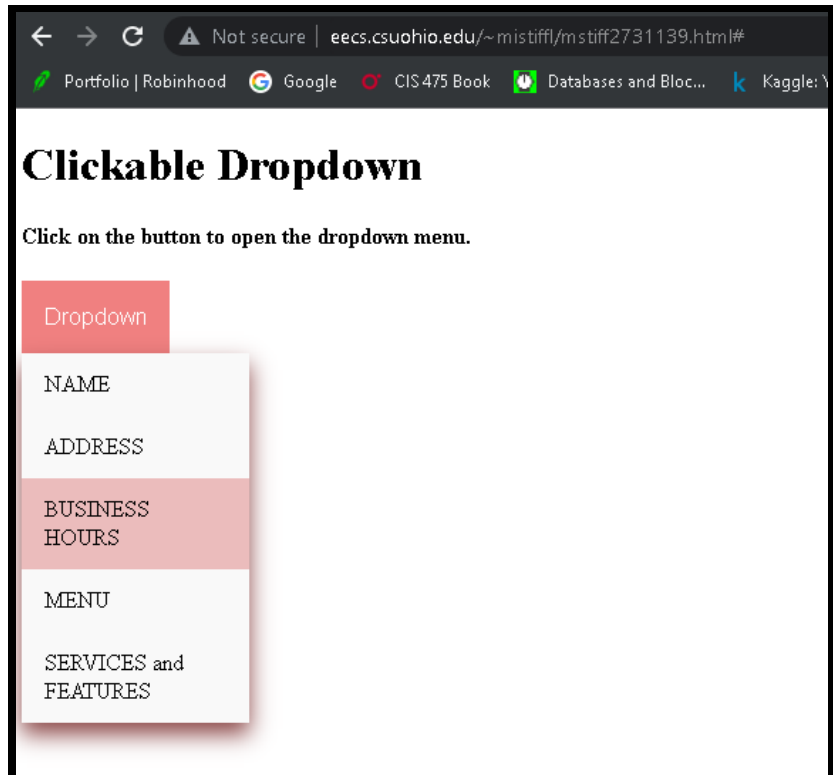
Dropdown:
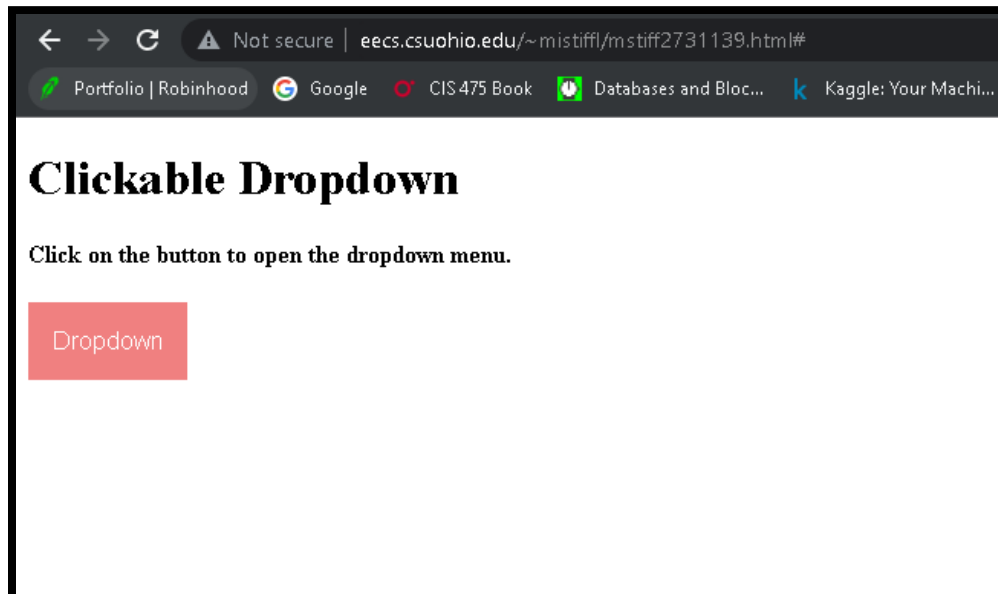
Here is the dropdown with my link in the background. I used the code that Dr. Chung gave us but changed my dropdown to have a coral color.
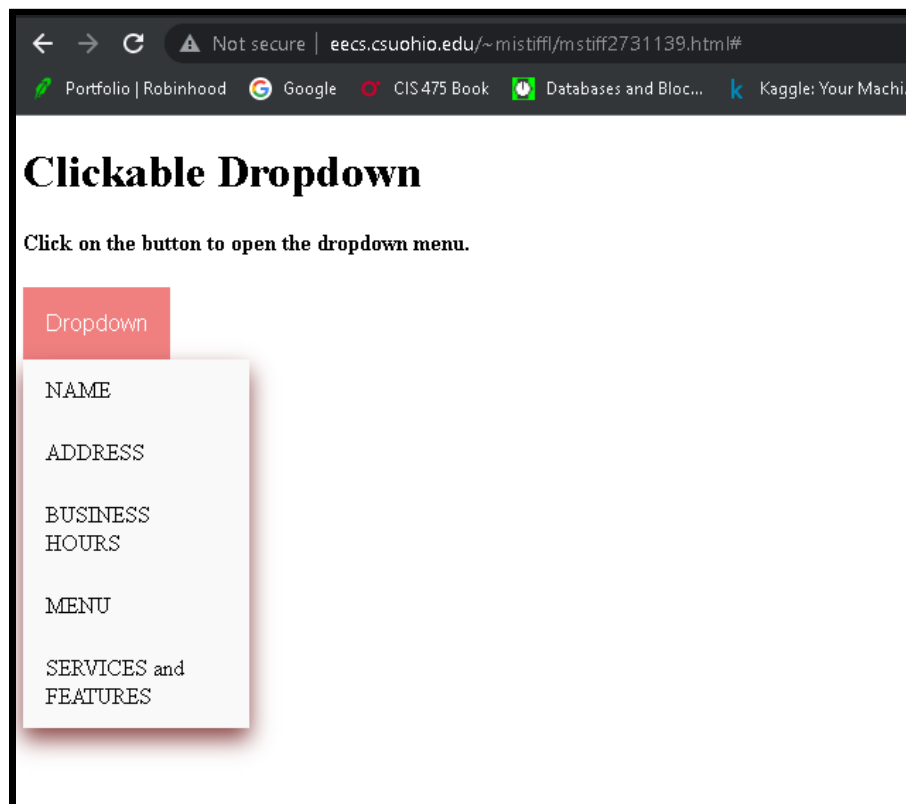

Here you can see when you hover over the dropdown button, the list of clickable items is displayed.

Here is an example of when you hover over an item in the list of items from the dropdown. It makes the color behind the text darker with some alpha value so the user knows what they're about to select.
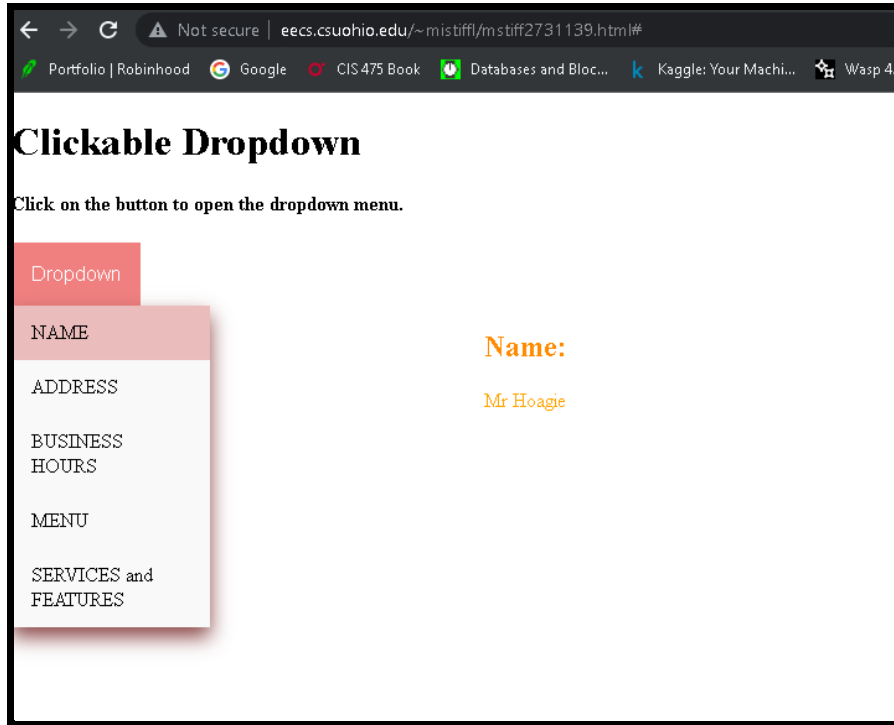
Here is what it will look like when the user clicks the name attribute in the list. My program will run the onclick() function for the button and go into the showName() function in my main.js file. If the text is not showing then I grab an h2 element and fill it with "Name:" and then I grab a paragraph element and fill in the correct JSON name attribute into it. If the text is already showing then I grab the div that the h2 and p element live under and change it's style display to "none". Here is what it will look like if I click the button again:
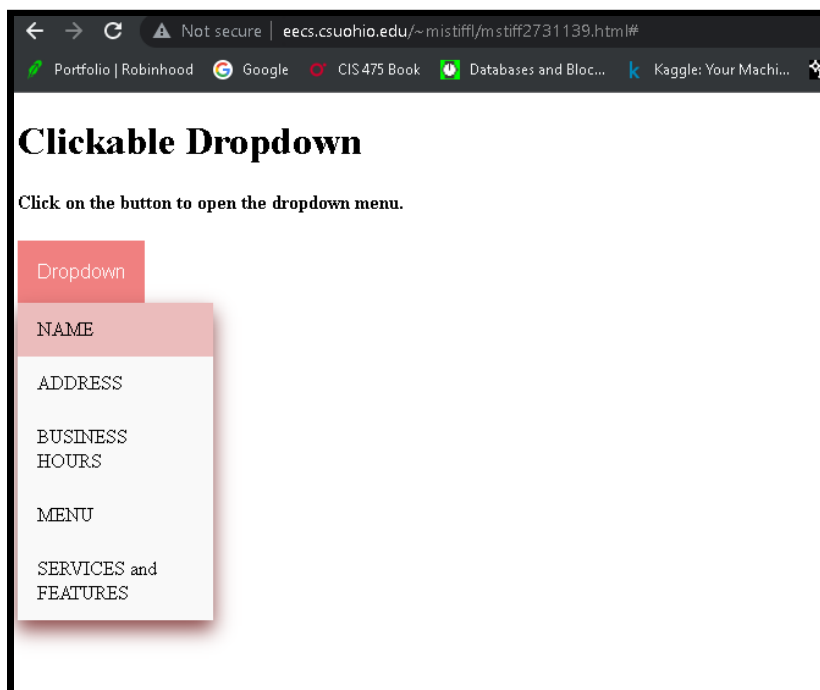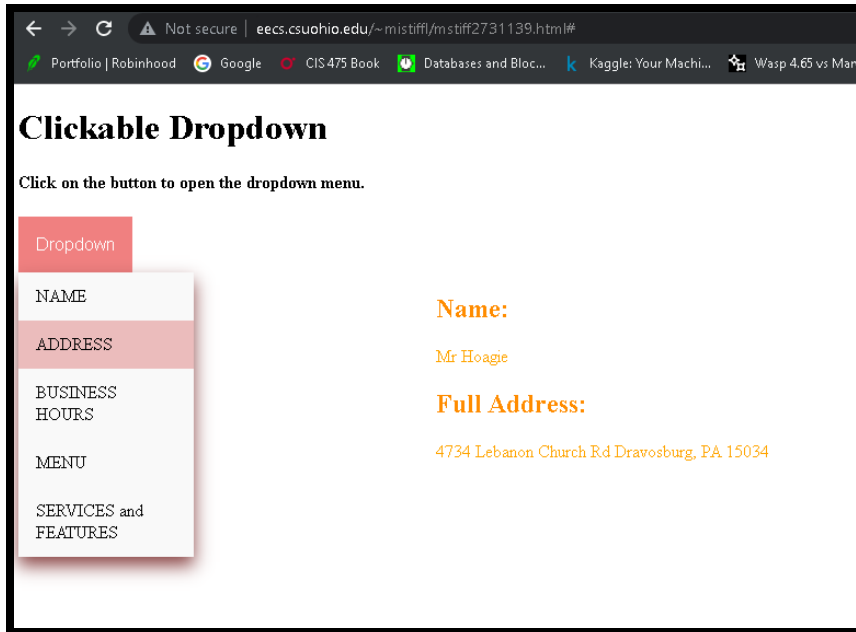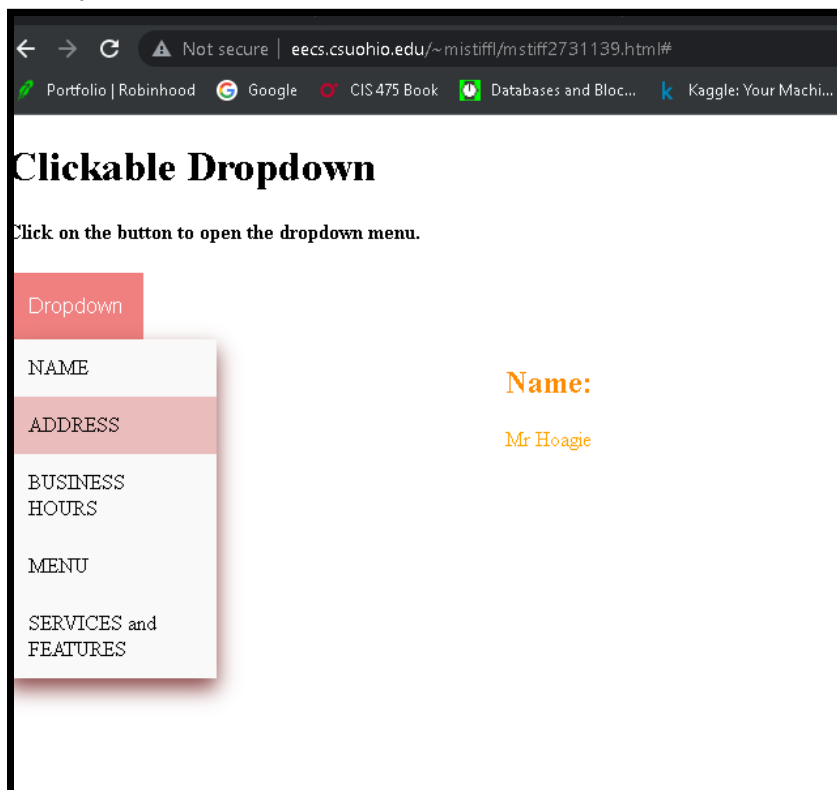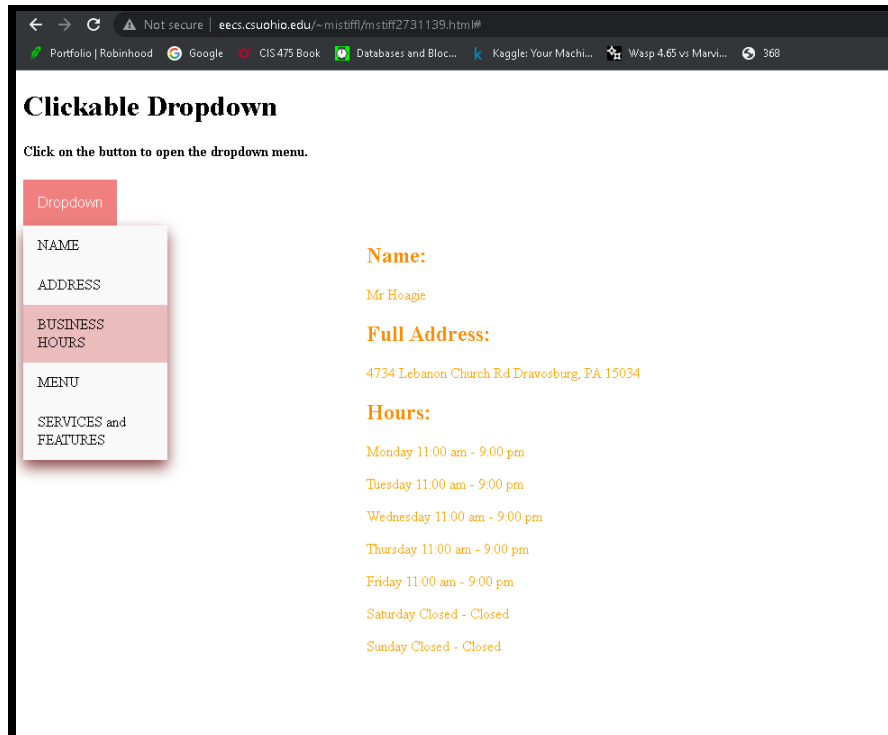
Here is what it will look like when the user clicks the address attribute in the list. My program will run the onclick() function for the button and go into the showAddress() function in my main.js file. If the text is not showing then I grab an h2 element and fill it with "Full Adress:" and then I grab a paragraph element and fill in the correct JSON address attribute into it. If the text is already showing then I grab the div that the h2 and p element live under and change its style display to "none". Here is what it will look like if I click the button again:
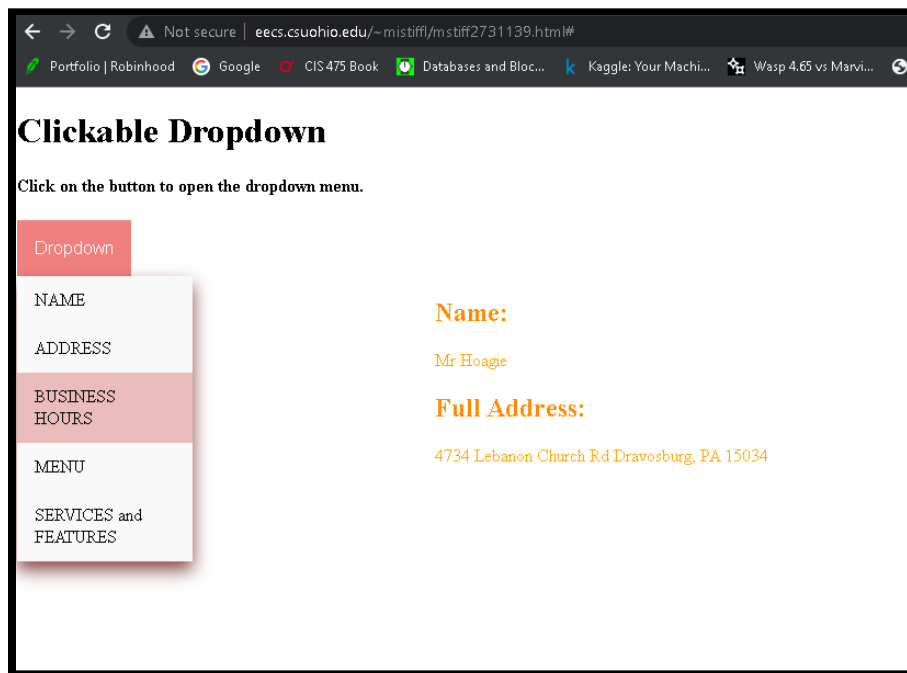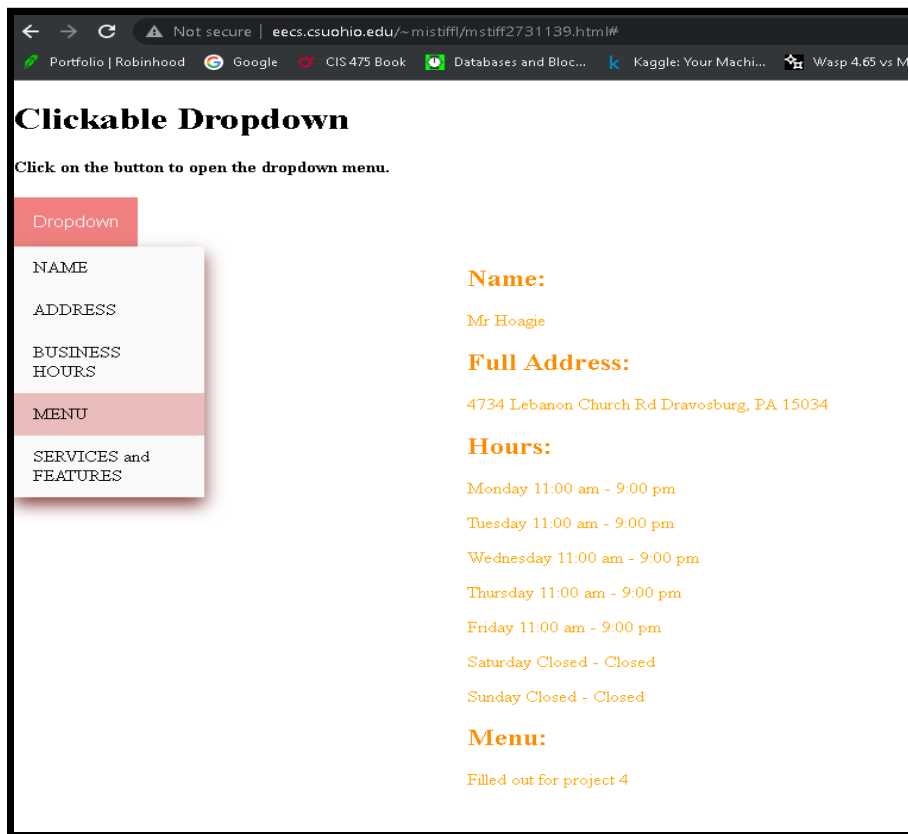
Here is what it will look like when the user clicks the hours attribute in the list. My program will run the onclick() function for the button and go into the showHours() function in my main.js file. If the text is not showing then I grab an h2 element and fill it with "Hours:" and then I grab a paragraph element and fill in the correct JSON hours attribute into it. For the hours, I made a new object:

```
let cleaned_hours = {
    "Monday": {"close": "Closed", "open": "Closed"},
    "Tuesday": {"close": "Closed", "open": "Closed"},
    "Wednesday": {"close": "Closed", "open": "Closed"},
    "Thursday": {"close": "Closed", "open": "Closed"},
    "Friday": {"close": "Closed", "open": "Closed"},
    "Saturday": {"close": "Closed", "open": "Closed"},
    "Sunday": {"close": "Closed", "open": "Closed"}
}
```

So that when I parse through the actual JSON object, I can just fill each day if it has a closed or open value for that day. Any other day will be closed if it is not in the list of hours from the JSON. I also run a function on each value to convert it to standard time using a function convertToStandard(time) with a library called moment. As you can see from the json, Saturday and Sunday were not present in the file so that's where it will print closed. If the text is already showing then I grab the div that the h2 and p element live under and change its style display to "none".
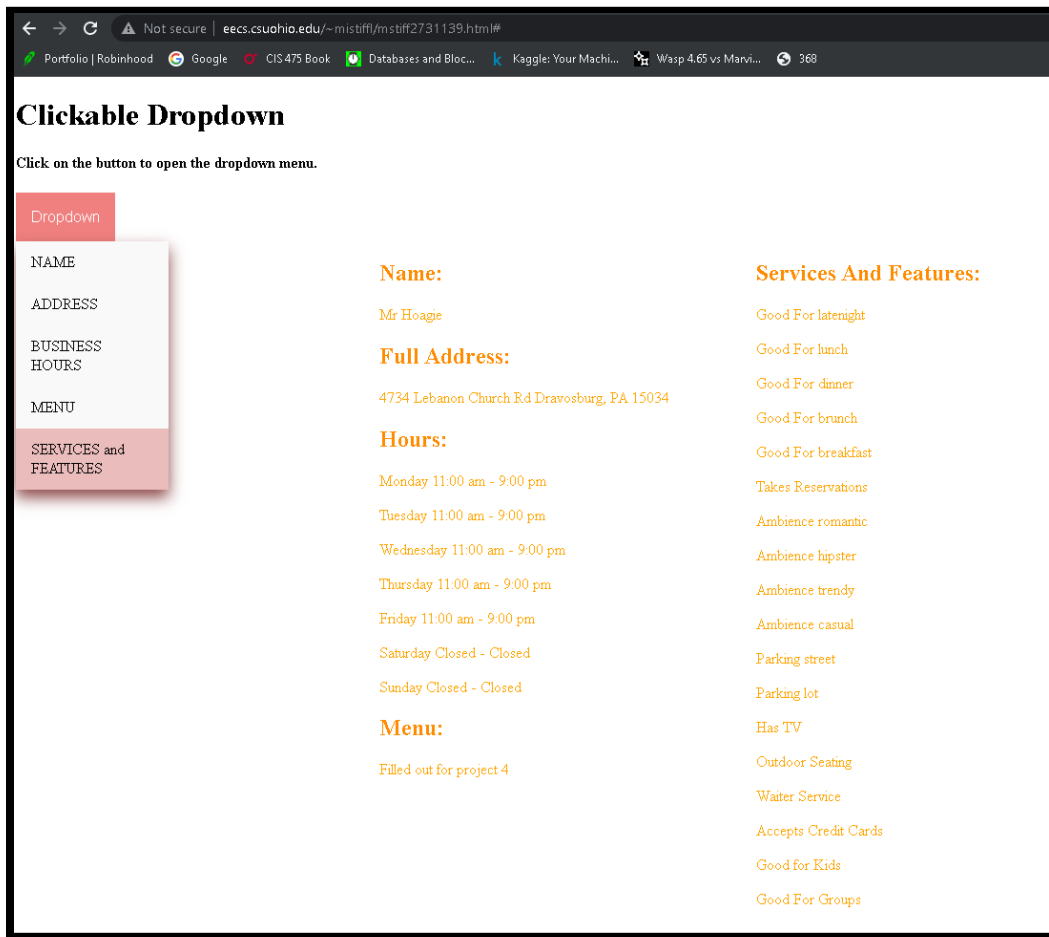
Here is what it will look like if I click the hours button again:

Here is just to show that the menu button works. It incorporates the same functionality as name and address when you click the button it goes to it's onclick() function in the javascript file and if it is not shown then I show the text, otherwise, I hide the text by making its parent div's style display "none". Here is what it looks like when you click the menu button again:
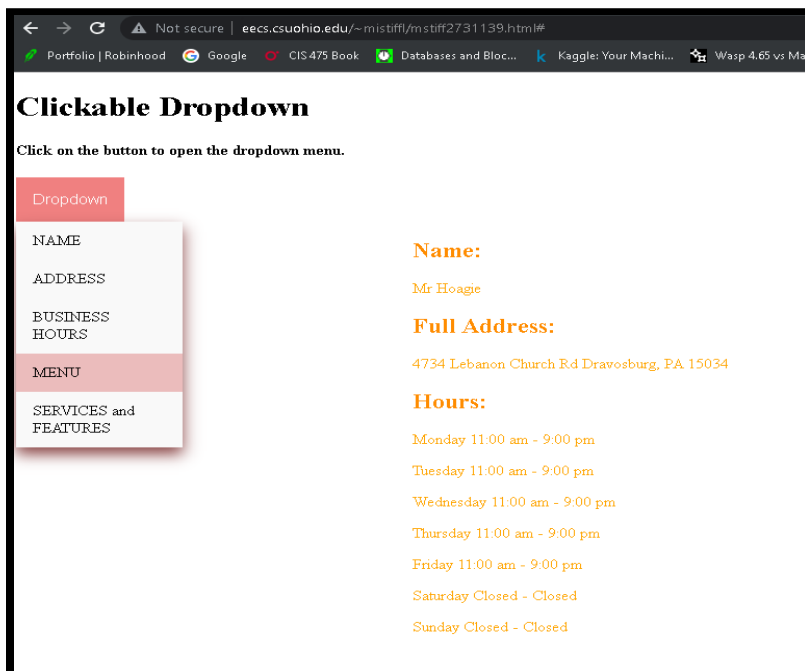
Here is what it will look like when the user clicks the Services and Feature attribute in the list. My program will run the onclick() function for the button and go into the showServicesAndFeatures() function in my main.js file. I only parse the attributes variable in the JSON one time. I iterate over each value in the object and if that value is not an object, I check if its value is equal to true and if it is, I add it to a list. If the value within the attribute object is an object, I run yet another loop to go through that object and check its values for either true like I did above, and add them to the list if they are. I then create a p element for each of the values in the list of true attributes and insert them before their parents as siblings.

If the text is not showing then I grab an h2 element and fill it with "Service And Features:" and also show all the p elements I created. Otherwise, if the text is showing I just set the parent div with all the p elements and h2 element to style display "none". Here is what that will look like:

Clickable Dropdown

Click on the button to open the dropdown menu.

Dropdown

NAME

ADDRESS

BUSINESS
HOURS

MENU

SERVICES and
FEATURES

Name:

Mr Hoagie

Full Address:

4734 Lebanon Church Rd Dravosburg, PA 15034

Hours:

Monday 11:00 am - 9:00 pm

Tuesday 11:00 am - 9:00 pm

Wednesday 11:00 am - 9:00 pm

Thursday 11:00 am - 9:00 pm

Friday 11:00 am - 9:00 pm

Saturday Closed - Closed

Sunday Closed - Closed

Menu:

Filled out for project 4