Sailesh Rajanala

Team Lead, Team Lotus
Technical Journal

Week #3 | 02/14 - 02/20

Research

This week, my research is about “Transition of a HTML form into a php form”.

I consider this as an important research that benefits the group because a form built using php supports “server-side” validation rather than “client-side” validation.

Sever-side validation makes sure that the client or the user has no control over how their submission is checked for validation. Hence, the user CANNOT bypass the form’s security during submission.

These are the webpages I have been referring to as part of my research on server-side form validation.

https://www.w3schools.com/php/php_form_validation.asp
https://www.w3schools.com/php/php_forms.asp

Lessons & Best Practices

This week, I have learned that sever-side validation is better than client-side validation due to the reason stated above. Therefore, I consider the best practice when creating a form is to create it using sever-side validation.

Assignment Progress Notes

This week, we as a team have completed CODE OF CONDUCT, 2nd SEMESTER WORK STATEMENT, and our first WEEKLY MINUTES assignments.

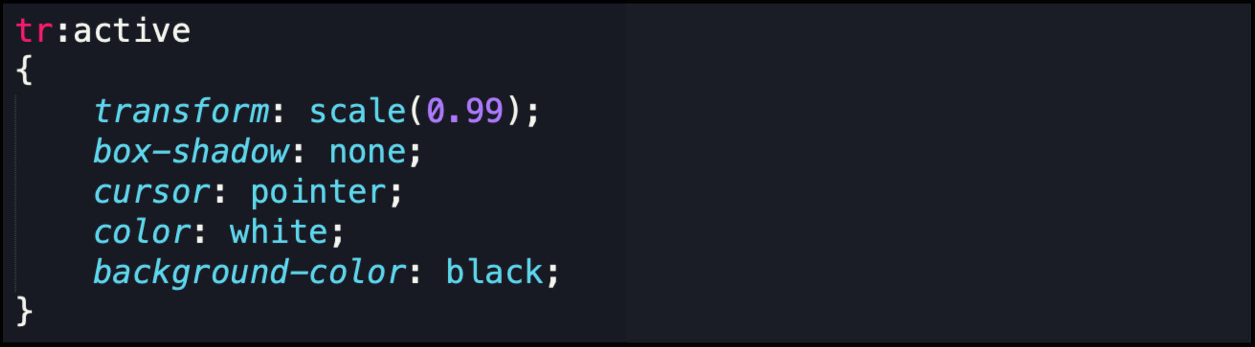
Since, we are new to the concept of Technical Journal that seems to be different from the traditional individual journal from previous semester, we as a team are carefully trying to understand the dos and don’ts regarding writing our Technical Journals.

Prototype Development Notes

This week, I have NOT done any major developments to the prototype other than adding some new CSS 3 features (design features) to the existing version of the prototype. We as a team, decided to start working on the Prototype starting next week.

Some of the code snippets which I contributed this week are showcased below.

Code below is to make a record change color and shrink
when the user clicks and holds it. (Bright mode & Dark mode)



```
tr:active
{
    transform: scale(0.99);
    box-shadow: none;
    cursor: pointer;
    color: white;
    background-color: black;
}
```

Results & Conclusions

There are no research results yet. The research is basically myself trying to understand and get familiar with php form development. The conclusion I had so far is that there is much to learn in php and there are new things I learned this semester so far while doing my research.

Problem

What's the Problem?

The problem this week is that the team has no experience in developing a web form that contains sub forms.

Why's it a problem?

It is a problem because we as a team, has created a web form that is simple and small. We did it with HTML, CSS and Javascript (used only for validation).

The problem is that we need to create a new form that contains various sub forms. (A dynamic form that switches between different fields basing on the user's input)

For example, if the user selects an option stating that they like to be a Guest speaker, then the form has to ask the next few questions that are unique to that option. In other words, the form should be able to ask different questions as per the user's input.

We as a team never done this before. I personally have no experience.

Possible Solution

I think a possible solution is to get familiar with more Javascript applications. I am going to research about php form validation and how to utilize Javascript to show user's input-specific questions. I have experience using javascript to show and hide HTML elements. Moreover, I think it is better to draw the logic on piece of paper, plan it and understand and study the ISME form deeply so that we could develop a dynamic form. Since dynamic forms heavily depend on Javascript, I think it is important to learn more about Javascript.

Week #4 | 02/21 - 02/27

Research

This week, my research is about “Utilizing JavaScript to make the webpage dynamic”.

Javascript is a programming language used in websites. This week, my job is to find a new way to make the web form feel new, unique, and user friendly. This week, we had team meeting where we have discussed or planned the user experience of the web form. One innovative rule that I followed is that I did not start my planning from the programmer’s end and move towards the user’s end rather I started thinking about what the user should see and feel and then slowly going backwards to the programmer’s end. As the result, I have understood that it is important to use Javascript on the programmer’s end to bring expected features to life.

This week I have researched about more Javascript uses like changing a text field’s text value, animating a web element, detecting if the user selected an option.

These are the links I used for my research.

https://www.w3schools.com/jsref/event_onfocus.asp
https://www.w3schools.com/jsref/event_onfocusin.asp
https://www.w3schools.com/jsref/event_onkeypress.asp

Lessons & Best Practices

This week, I have learned that Javascript is the standard programming used for webpage programming. The best practice of using javascript is using it externally (Including script from an external file) rather than internally (write script in the html or php file using script tags) because it promotes reusability and faster loading times for the webpage. This is discovered by me as the result of my research on JavaScript.

Assignment Progress Notes

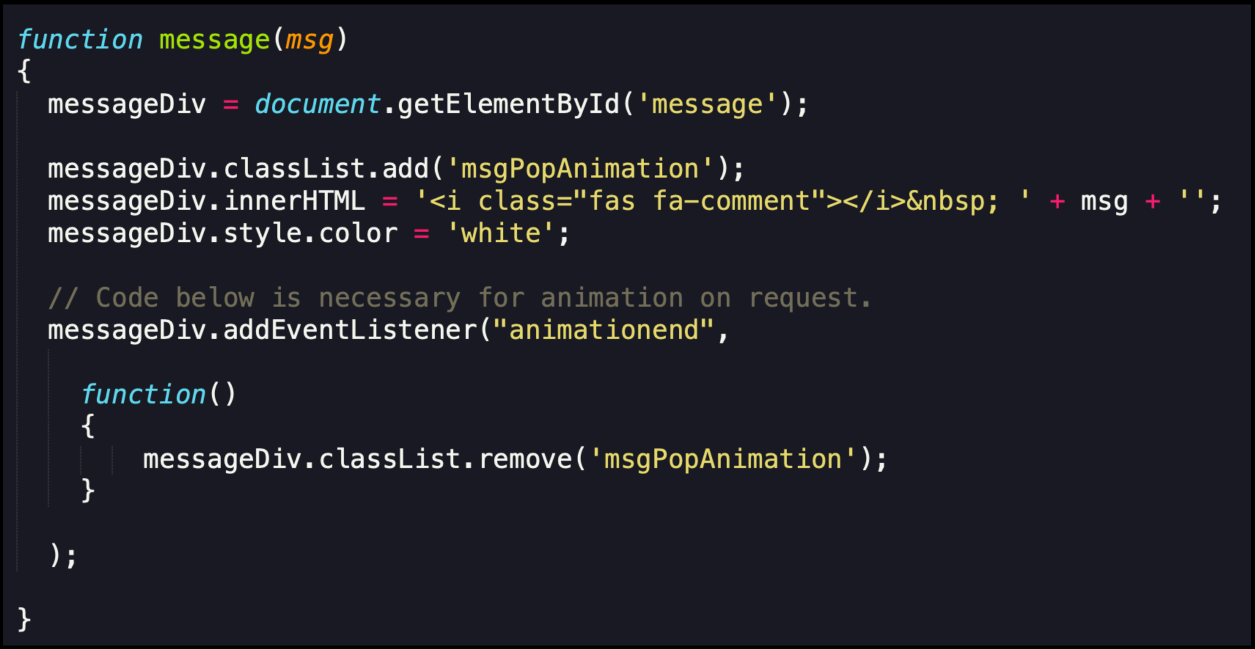
This week, we as a team have finished our Weekly Minutes Document, Project Proposal Write Up, and I have completed my Individual Journal. We as a team had team meetings twice this week where we planned our project’s user interface and one meeting on Friday during class time where we finished all the course assignments for this week.

Prototype Development Notes

This week, I have done some major developments to the web form that include features like displaying messages in the same element where the title is being displayed. The idea is to add new features (weight) to the existing form while painting its responsiveness (lightness and fast processing). I did this by changing nothing to the existing design which is really loved and appreciated but made it even better and responsiveness by simplifying the existing CSS and HTML code.

Some of the code snippets which I contributed this week are showcased below.

Code below represents a function to display message on top of the screen whenever the user clicks on any input field of the form.



```
function message(msg)
{
  messageDiv = document.getElementById('message');

  messageDiv.classList.add('msgPopAnimation');
  messageDiv.innerHTML = '<i class="fas fa-comment"></i>&nbsp; ' + msg + '';
  messageDiv.style.color = 'white';

  // Code below is necessary for animation on request.
  messageDiv.addEventListener("animationend",

    function()
    {
        messageDiv.classList.remove('msgPopAnimation');
    }

  );

}
```

Results & Conclusions

The result of my research regarding whether to choose between creating a new form with new design and new semantics or to follow the old design of existing form but with new semantics is to follow the existing design and semantics of the form but with added features or functionality which is only on top of the existing form design. I have decided to not use any new design but to rather modify or update the foundational designs instead.

Problem

What's the Problem?

The problem this week is that to decide the approach for redesigning and recreating the web from since the requirements were changed by the ISME deliverable.

Why's it a problem?

It is a problem because (in simple words) we cannot use the same old solutions to solve new problems. The new ISME deliverable on which the new web form should be created is demanding applications and heavy usage of web technologies like Javascript.

This problem questions the approach of designing and developing the existing web form. Therefore, it is important to decide whether to re-create the entire web form or to modify the existing web form and build new features on it.

Possible Solution

I believe that the possible solution is to use the existing web form and build new features upon it. This saves time and is important for us to know how this feels as we need to create a training manual for those who will build on or develop the web form after us. Therefore, we have decided to use most of the existing form and just simply add new features on it. We believe that this is possible and is not a huge challenge. I actually tried the development using this approach and I can tell you that it is 4x faster than the approach of creating an entire new foundation for an entirely new web form.

Week #5 | 02/28 - 03/06

Research

This week, my research is about “Redesigning Checkboxes and Radio Buttons”.

In any HTML form, there are always radio buttons and check boxes that are used to take input from the user. But these checkboxes and radio buttons are static and they have the same design and also serve one single purpose that is for the user to select different items in the web form.

This week I have researched about redesigning check boxes and radio buttons. The goal is to obtain a web form that is so simple and straight forward to use. The user should not take time to research or think about how to use the web form itself.
This is my thinking this week to maximize simplicity and utilize design patterns that not only are consistent but also should feel instantly familiar and easy to use.

My research includes learning from web resources listed below.

https://www.w3schools.com/howto/howto_css_custom_checkbox.asp
https://markheath.net/post/customize-radio-button-css

Lessons & Best Practices

This week, It is all about maximizing cohesion and minimizing coupling. In simple words, an important design technique which I usually follow when developing websites is to make the web elements or web modules highly cohesive and extremely less coupled. In simple terms, there are many ways of solving a problem. Brute-Force is not an option for a reliable web design. The best practice or the lesson I learned is to create web elements that are independent, reusable, and in generic form
(open to improvement).

Assignment Progress Notes

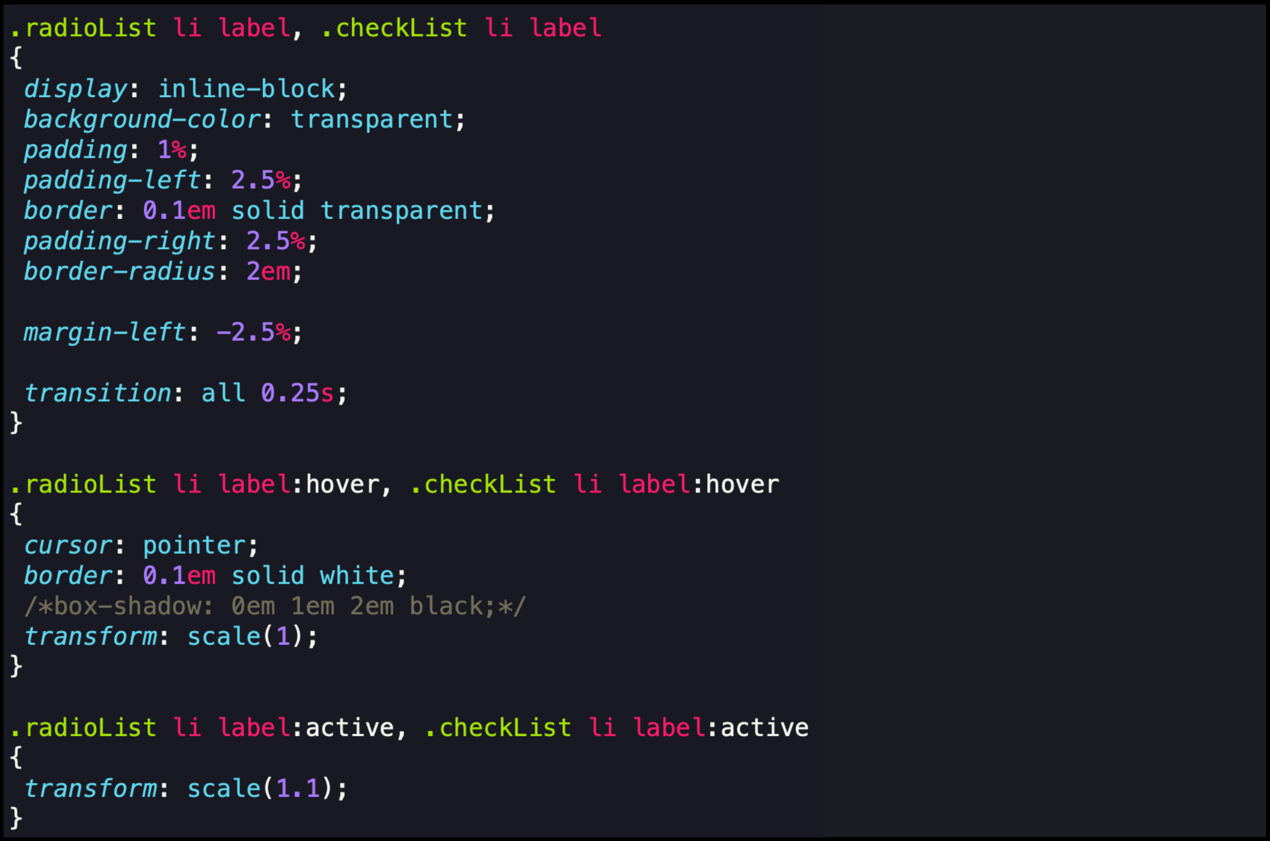
This week, we as a team had multiple team meetings and during our meetings, we wrote code for most of the time. During the team meeting on Thursday, we completed the weekly minutes document together and I completed my Individual Journal.

Prototype Development Notes

This week, I have done changes to CSS of radio buttons and checkboxes in the form. Apart from that, I have also added new javascript functions or modules as part of the features of the form. The end result is a new UI for radio buttons and checkboxes which is simple, consistent, and easy to use.

Some of the code snippets which I contributed this week are showcased below.

Code below represents CSS properties of the new radio buttons and check boxes.



```
.radioList li label, .checkList li label
{
 display: inline-block;
 background-color: transparent;
 padding: 1%;
 padding-left: 2.5%;
 border: 0.1em solid transparent;
 padding-right: 2.5%;
 border-radius: 2em;

 margin-left: -2.5%;

 transition: all 0.25s;
}

.radioList li label:hover, .checkList li label:hover
{
 cursor: pointer;
 border: 0.1em solid white;
 /*box-shadow: 0em 1em 2em black;*/
 transform: scale(1);
}

.radioList li label:active, .checkList li label:active
{
 transform: scale(1.1);
}
```

Results & Conclusions

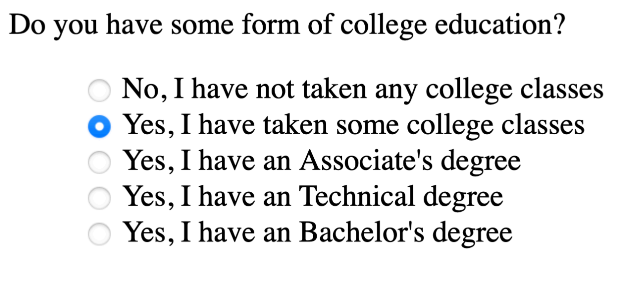
This week, my research results conclude the “merging” of radio buttons and checkboxes as one single UI element. This could be good or bad. Good thing is that it is simple since it is one single thing that serves two different purposes and it is super easy to use. A problem could arise which questions their functionality.

Checkboxes and Radio Buttons have two different purposes.

Check Boxes = to select ***ONE or MORE*** options

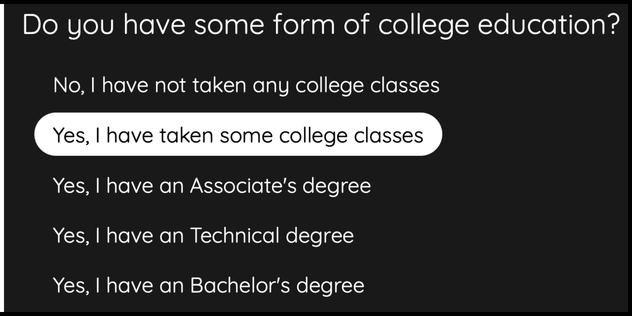
Radio Buttons = to select ***ONLY ONE*** option

Now, if we pay close attention, one thing they both have in common is ***SELECT***
This is the key idea for innovation. This is the idea behind my redesign or merging both input elements. With this, the user has only one goal : ***TO SELECT***



Do you have some form of college education?

○ No, I have not taken any college classes
● Yes, I have taken some college classes
○ Yes, I have an Associate's degree
○ Yes, I have an Technical degree
○ Yes, I have an Bachelor's degree

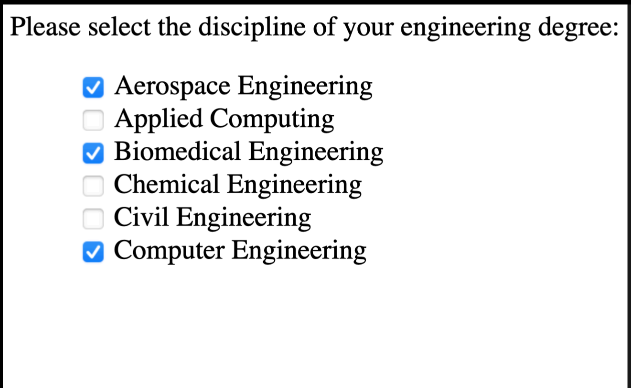


Do you have some form of college education?

No, I have not taken any college classes
[✓ Yes, I have taken some college classes]
Yes, I have an Associate's degree
Yes, I have an Technical degree
Yes, I have an Bachelor's degree

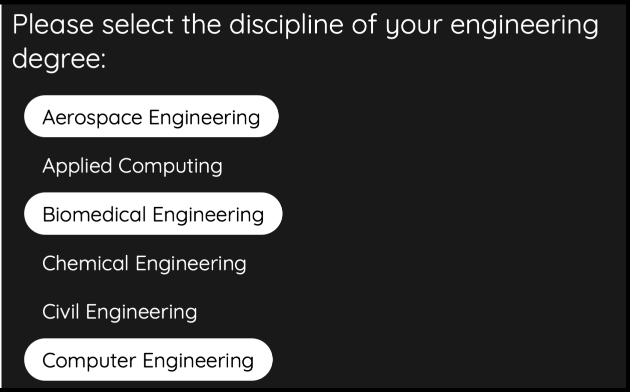
Ordinary Radio Button when selected

My New Radio Button when selected



Please select the discipline of your engineering degree:

☑ Aerospace Engineering
□ Applied Computing
☑ Biomedical Engineering
□ Chemical Engineering
□ Civil Engineering
☑ Computer Engineering



Please select the discipline of your engineering degree:

[✓ Aerospace Engineering]
Applied Computing
[✓ Biomedical Engineering]
Chemical Engineering
Civil Engineering
[✓ Computer Engineering]

Ordinary Check Boxes when selected

My New Check Boxes when selected

No matter how well the new check boxes and radio buttons look, their functionality is unchanged. This means that we have all the new innovative ideas plus the original and unchanged functionality, all together in the new input elements. This means that the developer has to not worry about learning any thing new rather they simply use these like ordinary radio buttons and check boxes.

I have created radio buttons unlike any radio buttons and check boxes unlike any check boxes. This is the result of my research and I am very happy about it.

Problem

What’s the Problem?

The problem this week is regarding design elements that did not really feel familiar or useful to the user experience. In simple words the problem is over thinking about UI and making it over complicated.

Why’s it a problem?

It is a problem because the User Interface should be attractive, easy to use, and powerful (should make sense and simplify things) but not complicated and hard to use or code. There are lots of ways to create a web form and built a unique user interface.

Basing on personal experience, I always wish to have someone knowledgable with me to help me whenever I fill any professional form. The goal of our user interface is to solve this problem. Its purpose is to help the user but not confuse the user. The form should provide frequent feedback to the user so that the user at the end of the day should have no worries or questions about what they have submitted or whether they have submitted exactly what is required or something else.

Possible Solution

Building a User experience is key. Design is not how something looks, it is how something exists or actually works. I have changed the UI element called `messageDiv` many times (> 50). The reason is to make it stand out, appear, and to be used only when necessary. Finally, taking a look at the UI again after some time, made me look back and understand what exactly each element’s purpose is.

With this attitude, I was able to design elements that actually represent some sort of connection and the connection between elements is what the user instantly notices and gets familiar with. Moreover, the solution for such problems is to slow down during the design phase, think carefully, rethink, and use the UI many times to detect imperfections and fix them until the entire design or UX/UI makes sense.

Week #6 | 03/07 - 03/13

Research

This week, my research is about “Reinventing the User Experience”.

In any HTML form, there are simple fields like Name, email, phone number as well as complex fields (hard to understand) like Employer (name? or company?). Therefore one problem a modern UI should solve is to make the purpose of the Web Form clear and easy to understand for the user. The next level UI should help the user by acting like a guide from start till the end of filling the form. Not only that, the UI should be powerful as well and should dynamically check the input provided by the user and give the user frequent and updated feedback as the user modifies their input.

This week I have researched about reinventing the user experience using JavaScript’s `append()` and CSS’s `@keyframes` (tool to create and control animations).

My research includes learning from web resources listed below.

https://developer.mozilla.org/en-US/docs/Web/API/ParentNode/append
https://www.w3schools.com/jsref/met_node_appendchild.asp
https://www.w3schools.com/jsref/event_onkeyup.asp
https://developer.mozilla.org/en-US/docs/Web/CSS/@keyframes
https://www.w3schools.com/cssref/css3_pr_animation-keyframes.asp

Lessons & Best Practices

This week, It is all about creating jaw-dropping and smooth animations to make the Web Form distinguishable from all the other forms. The trick is to use only simple tools to create powerful modules. We create these extraordinary modules from scratch and by using simple inbuilt Javascript event listeners and functions to keep the simplicity and functionality of JavaScript alive while improving and adding new features to the form at the same time.

Animation Best Practices (for SMOOTH animations)

<> use `transform: scale()`
instead of changing width or height properties of the HTML element.

<> use `transform: translateY()`
instead of changing margin or padding properties of the HTML element.

Assignment Progress Notes

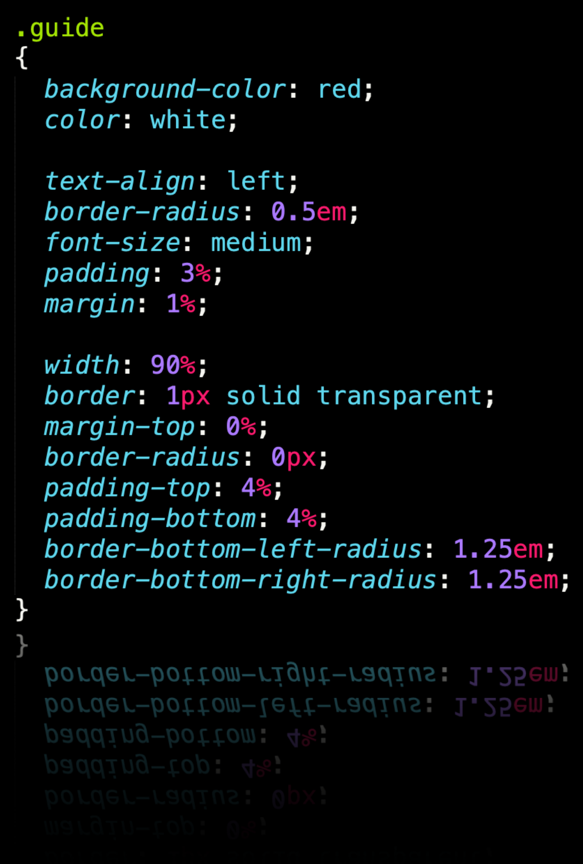
This week, we as a team had a team meeting on Friday where we discussed about the Team Peer Reviews Package, completed the weekly minutes document together and I completed my Individual Journal.

Prototype Development Notes

This week, as I listed above, I have worked on updating animations and also updating the overall theme or UI of the Web Form using CSS 3 and JavaScript.

Some of the code snippets which I contributed this week are showcased below.

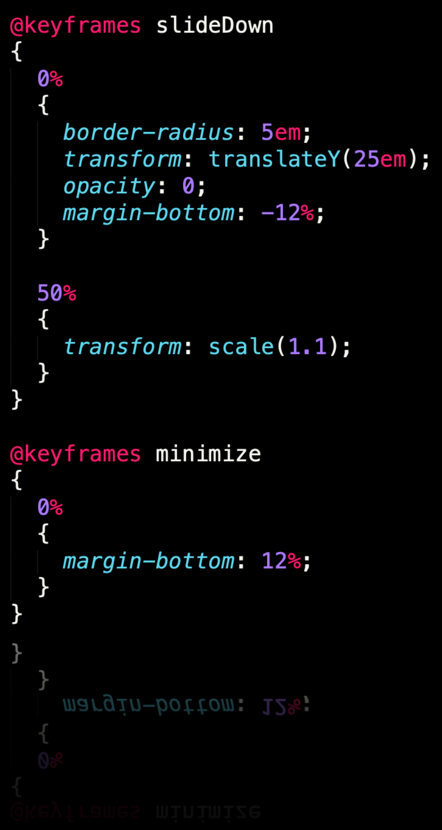
Code (left) represents CSS properties of the user Interface for “guide”.
Code (right) represents animations for “guide”.



```
.guide
{
    background-color: red;
    color: white;

    text-align: left;
    border-radius: 0.5em;
    font-size: medium;
    padding: 3%;
    margin: 1%;

    width: 90%;
    border: 1px solid transparent;
    margin-top: 0%;
    border-radius: 0px;
    padding-top: 4%;
    padding-bottom: 4%;
    border-bottom-left-radius: 1.25em;
    border-bottom-right-radius: 1.25em;
}
```



```
@keyframes slideDown
{
    0%
    {
        border-radius: 5em;
        transform: translateY(25em);
        opacity: 0;
        margin-bottom: -12%;
    }

    50%
    {
        transform: scale(1.1);
    }
}

@keyframes minimize
{
    0%
    {
        margin-bottom: 12%;
    }
}
```

Results & Conclusions

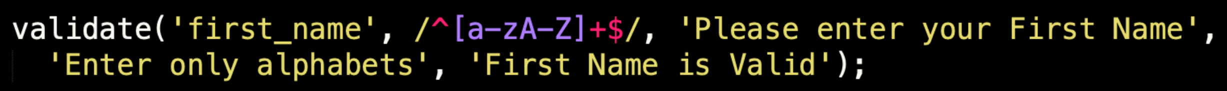
This week, the results of my research is a JavaScript function() that is powerful and independent which can be used to validate any textfield in the Web Form.

The function works in three stages.

Stage 1 : When there is no input from the user yet.
Stage 2 : When there is a valid input.
Stage 3 : When there is an invalid input.

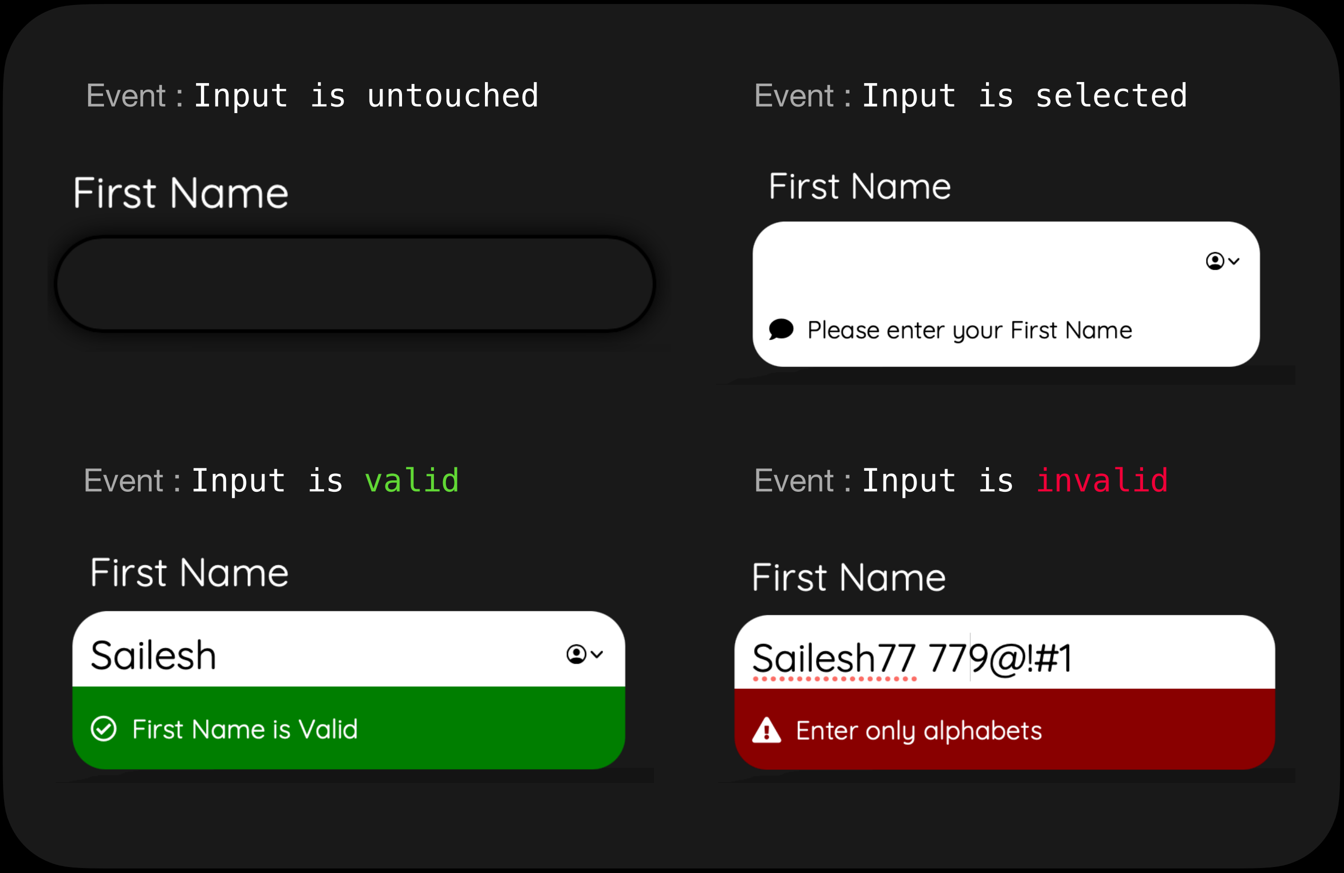
To use the function, all the programmer needs to do is write only one line of code.

Example <>



```
validate('first_name', /^[a-zA-Z]+$/, 'Please enter your First Name',
  'Enter only alphabets', 'First Name is Valid');
```

Code in action <>



Event : `Input is untouched`

First Name

Event : `Input is selected`

First Name

Please enter your First Name

Event : `Input is valid`

First Name

Sailesh

✓ First Name is Valid

Event : `Input is invalid`

First Name

Sailesh77 779@!#1

⚠ Enter only alphabets

Problem

What’s the Problem?

The problem this week is simplifying complex solutions or powerful animations. Everyone loves great graphics. Intense Graphics always come with a price that is demanding high performance of the machine that results in draining the battery life. How can we fix this?

Why’s it a problem?

It is a problem because the a Web Form with great features is not reliable if it has rough animations or transitions. Animations being smooth and light is necessary for best performance of any website. In the case of web form, it is true that the number of features are inversely proportional to good web performance of the web form. If performance is bad, then the user experience is terrible.

Possible Solution

Time is key and is our solution. Instead of running multiple tasks at once. We could run them at separate times. The human eye ignores the time intervals in terms of milli seconds and does not notice them. Therefore, algorithms like Round Robin process scheduling are inspiration here. When an animation is being run, it must is the only animation that is being run. All the animations in the website take 0.25s and this is the standard I follow to make animations fast and noticeable. Slower or faster than 0.25 takes away the liveliness of the Web Form. Therefore, there is a possibility for us to run 4 animations in 1 second. All animations associated with a HTML element could be completed in terms of seconds instead of milliseconds which means fewer concurrent or parallel animations which automatically saves up the performance and space of the machine.

We can achieve this by utilizing

`animation-delay: {time in seconds}` (CSS 3) and

starting new animations only when a current animation completes by utilizing

`addEventListener(“animationend”, function() {} );` (JavaScript)

This is the solution I found after some research and it helps in improving the performance of graphically intense webpages.

Week #7 | 03/14 - 03/20

Research

This week, my research is about
“Fusing Radios with Text Fields and Checkboxes with Text Fields”.

In any HTML form, if there is a question asking the user to select an option among the list of options and if there is an option called “other” (usually provided for the user to enter or create their own option), when the user selects the other option, then the user is asked a different question and is now displayed with a different form element usually a text box to get the new value of the user’s own option.

This week I have researched about sing a radio button with a text box as well as mixing a check box with a text box. The result of this research simplifies a complicated solution or simplifies a complicated procedure that has been followed since many years.

These are the web pages where I got the information for my research.

https://developer.mozilla.org/en-US/docs/Web/API/HTMLFormElement/elements
https://developer.mozilla.org/en-US/docs/Web/API/HTMLFormElement
https://www.w3schools.com/jsref/coll_form_elements.asp
https://www.w3schools.com/jsref/prop_radio_checked.asp
https://www.w3schools.com/jsref/prop_checkbox_checked.asp

Lessons & Best Practices

This week, the question is about the functionality of the “other” option. The question is that what should happen when the user selects the other option among all the provided options. This means that the user wants to provide their own opinion or create their own option. This simple logic made us do exactly that which resulted in simplifying the overall functionality of the “other” option among list of options.

While programming such functionality, we had to make sue of functions in JavaScript. We want the programmer to utilize the same function but in many ways and to accomplish different tasks, eventually making the function itself a variable. We did this using a feature called “Default parameters”. Using default parameters for a function makes the function reusable for different cases. Therefore we think whenever we create a function, it is best practice to utilize the power of default parameters for that function.

Assignment Progress Notes

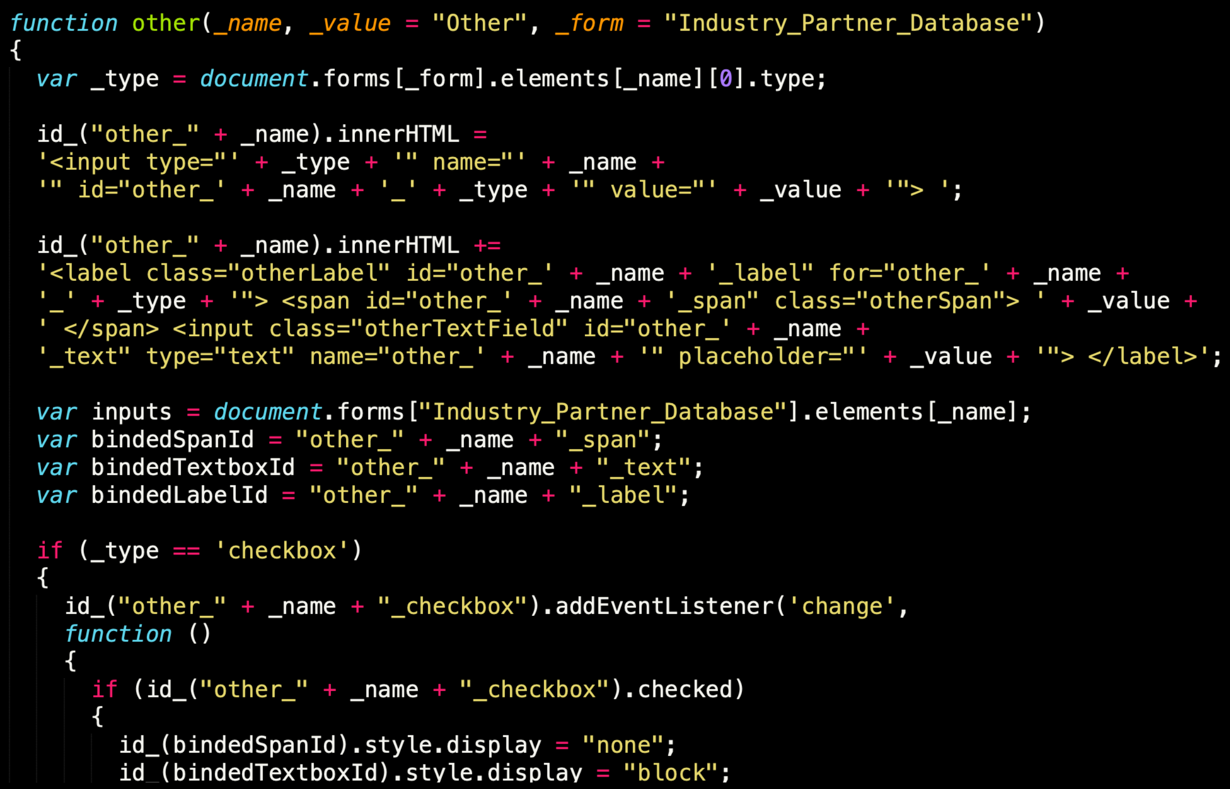
This week, we as a team had multiple team meetings during which we discussed the logic of various HTML elements' functionalities and I wrote some CSS 3 and JavaScript code. We also discussed about the Project Team Review Package and have created and uploaded new files to the GitHub repository we submitted last week. During the team meeting on Thursday, we completed the weekly minutes document together and I completed my Individual Journal.

Prototype Development Notes

This week, I have considered solving the “other” option functionality problem. Therefore, I have come up with various code snippets that include creation of various functions, event listeners and their corresponding HTML elements.

This week, I have created a new module that handles the “other option” problem. I consider this as an innovative task as we are creating our own form element that is a combination of a radio button and a text box or the combination of a check box and a text box.

I have created the function below during our team meetings along with the help of my teammates.



```
function other(_name, _value = "Other", _form = "Industry_Partner_Database")
{
  var _type = document.forms[_form].elements[_name][0].type;

  id_("other_" + _name).innerHTML =
  '<input type="' + _type + '" name="' + _name +
  '" id="other_' + _name + '_' + _type + '" value="' + _value + '"> ';

  id_("other_" + _name).innerHTML +=
  '<label class="otherLabel" id="other_' + _name + '_label" for="other_' + _name +
  '_' + _type + '"> <span id="other_' + _name + '_span" class="otherSpan"> ' + _value +
  ' </span> <input class="otherTextField" id="other_' + _name +
  '_text" type="text" name="other_' + _name + '" placeholder="' + _value + '"> </label>';

  var inputs = document.forms["Industry_Partner_Database"].elements[_name];
  var bindedSpanId = "other_" + _name + "_span";
  var bindedTextboxId = "other_" + _name + "_text";
  var bindedLabelId = "other_" + _name + "_label";

  if (_type == 'checkbox')
  {
    id_("other_" + _name + "_checkbox").addEventListener('change',
    function ()
    {
      if (id_("other_" + _name + "_checkbox").checked)
      {
        id_(bindedSpanId).style.display = "none";
        id_(bindedTextboxId).style.display = "block";
```

Results & Conclusions

This week, my research resulted in the creation of a highly cohesive JavaScript function() that is very powerful and can be used to dynamically display and activate a text box when the user selects the “other” radio button or the “other” check box.

The function has the following features

Feature 1 : Automatic detection of the given HTML element.
(Determines whether the given id belongs to a radio button or to a checkbox)

Feature 2 : Element specific form handling.
(Makes the resulting text box act and behave either like a radio button or a checkbox)

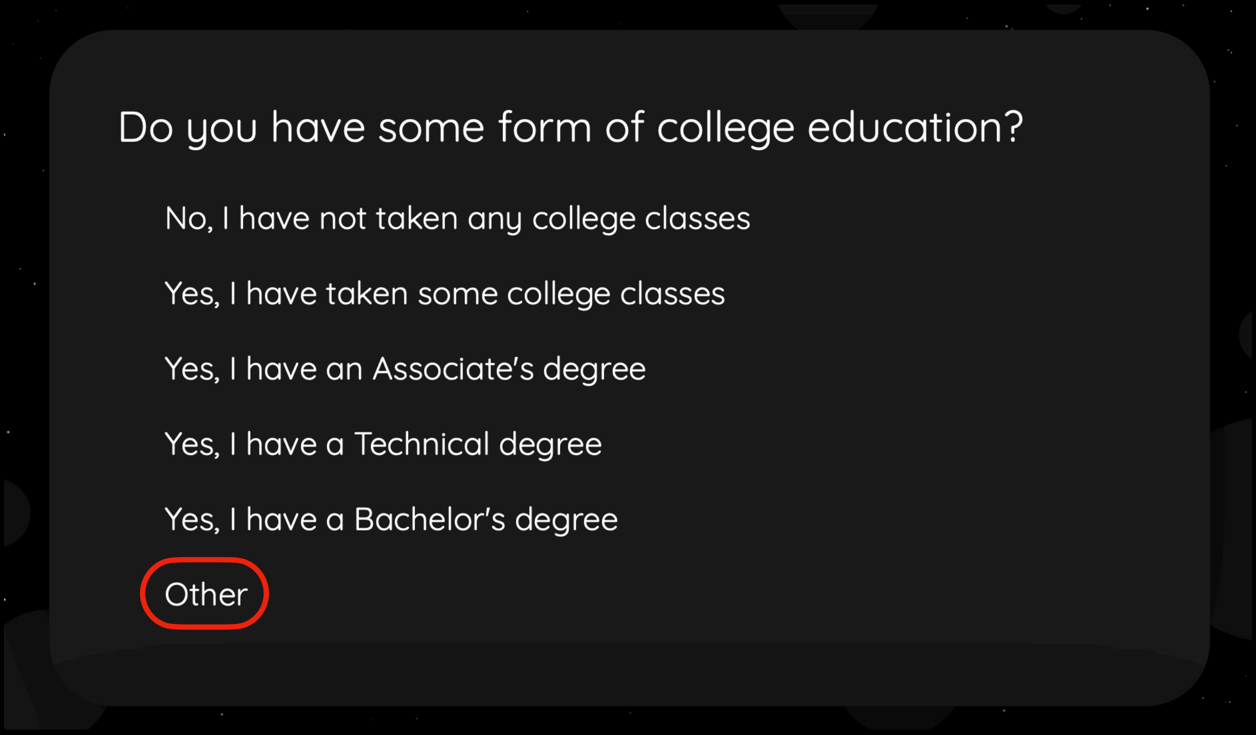
Stage 3 : Highly Cohesive with Minimal Coupling.
(Programmer can use this function to create the “other” option just by a function call)

Example <>

```
other("college_education");
```

Result <>

The creation of “Other” option that is a text box but behaves like a radio button.



Do you have some form of college education?

No, I have not taken any college classes

Yes, I have taken some college classes

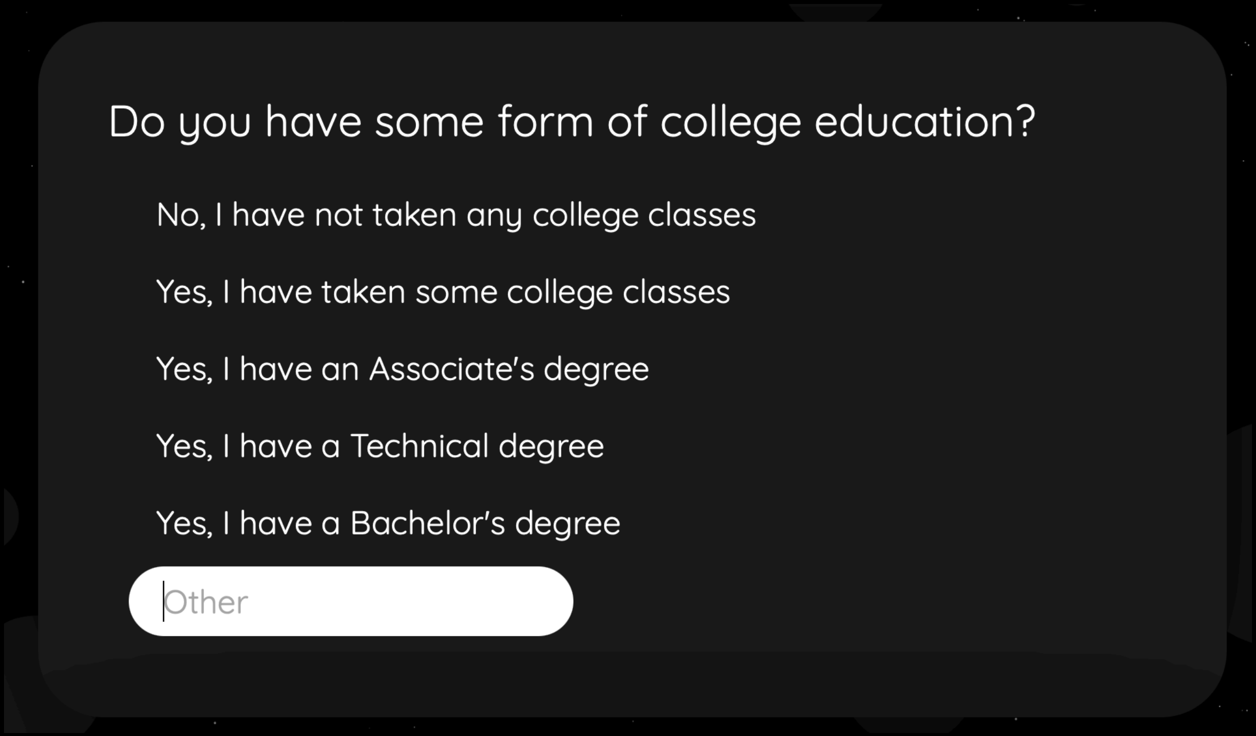
Yes, I have an Associate's degree

Yes, I have a Technical degree

Yes, I have a Bachelor's degree

[✓ Other]

When the user selects the “Other” option, it transforms into a text-box.



Do you have some form of college education?

No, I have not taken any college classes

Yes, I have taken some college classes

Yes, I have an Associate’s degree

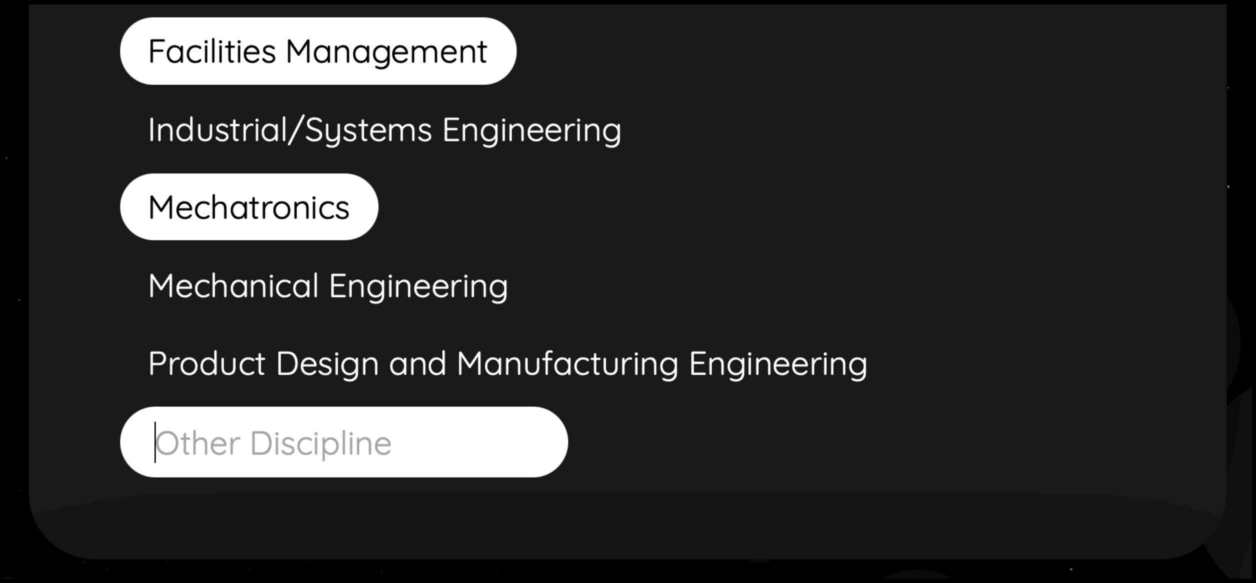
Yes, I have a Technical degree

Yes, I have a Bachelor’s degree

Other

Now, the user is given the freedom of creating their own option among the list of options. This is the power of calling the `other()`

Now, the “Other” in this case, is a text box but behaves like a checkbox. Therefore, `other()` can make a textbox behave as a radio button or as a checkbox.



[✓ Facilities Management]

Industrial/Systems Engineering

[✓ Mechatronics]

Mechanical Engineering

Product Design and Manufacturing Engineering

Other Discipline

Problem

What’s the Problem?

The problem this week is about coming up with a reliable and a unified solution to make a text box behave either like a radio button or as a checkbox.

Why’s it a problem?

It is a problem because we need to meet the standards of the our HTML elements or the bar that we set so high with our way of utilizing singular HTML elements like textbooks and selects and we need to be able to deliver the same quality of treatment or utilization to non-singular HTML elements like a group of radio buttons or a group of checkboxes.

Possible Solution

A possible solution is to understand the requirements and then come up with some innovative idea. Therefore, I stood back and took time to understand the details or demands or requirements of what it is like to create the best two-in-one substitute for the ‘other’ option.

Requirement #1
We need a text box so the user can type their own option.

Requirement #2
If we are making the user create their own option, then the new option should respect and follow the rules of its domain (radio button or checkbox)

Requirement #3
We should also be able to later develop on the current foundation of the function

Solution
Create a HTML element that contains both [radio or checkbox] and [textbox] and display only the [radio or checkbox] until selected by the user but display only the [textbox] version when the user selects it. Create a JavaScript function that automatically switches between different versions of input. Use the same CSS rules for both the [radio or checkbox] and [textbox] version of the HTML element. Use *`eventListeners`* for events like *`keyUp, change, focusin, and focusout.`* Finally, use Default Parameters to make the function more cohesive and less coupled so to enable the programmer build new features on its current foundation.