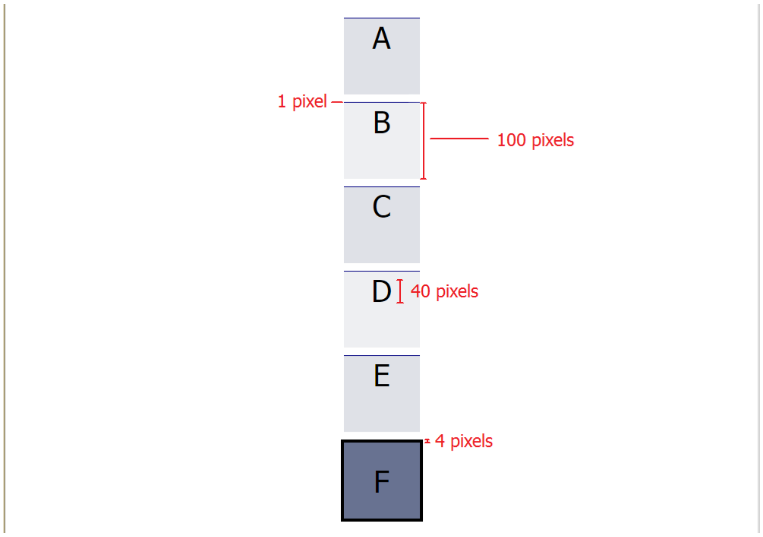
HW1

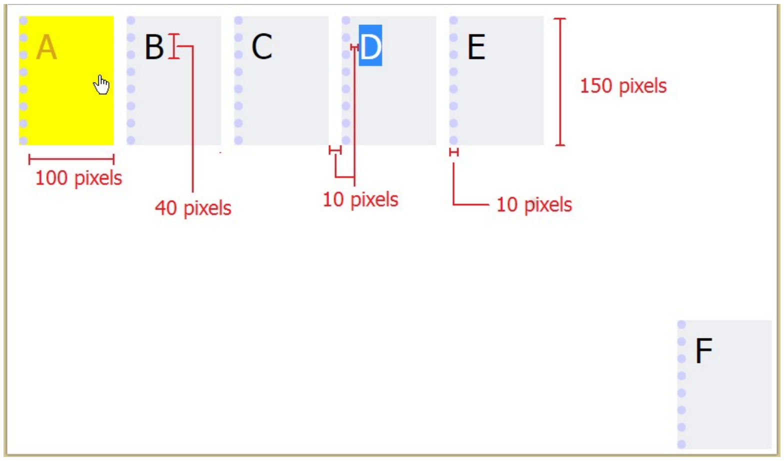
Problem 1 (10 pts)

In this problem, you need to define two HTML files and two CSS files under a folder called problem1. The two HTML files are problem1-firstStyle.html and problem1-secondStyle.html, respectively. In the same folder, create two CSS files, firstStyle.css and secondStyle.css. You link problem1-firstStyle.html to firstStyle.css and problem1-secondStyle.html to secondStyle.css. The contents in problem1-firstStyle.html and problem1-secondStyle.html MUST be exact the same, except for the <link> element.

When I double click problem1- firstStyle.html, I should see something like this:



When I double click problem1-secondStyle.html, I should see something like this:



Note: The D in the picture is highlighted to show you what space the text should occupy. Don't style it with a blue background.

Explanation of the first style:

- The 6 boxes are vertically lined up.
- The 6 boxes are horizontally centered and equally spaced vertically (10 px).
- Every box has the same height (100px) and width (100px), with a 1px top border. Text is always horizontally centered.
- You need to figure out the color code using a color picker.
- “F” is centered both horizontally and vertically in the box.
- The font in all boxes should be 40px and Tahoma.

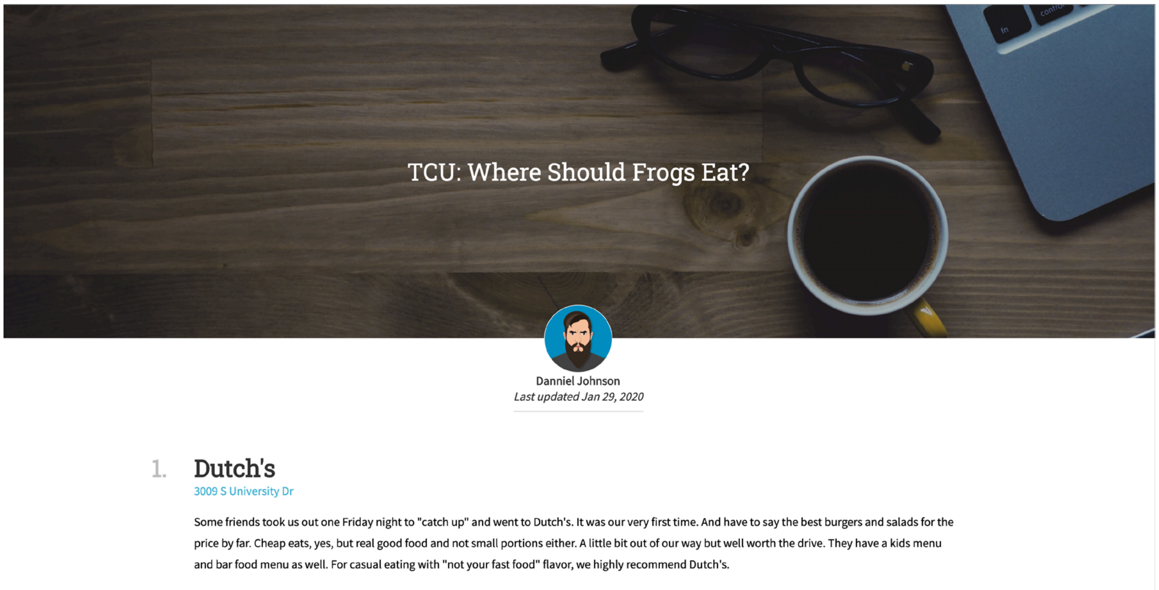
Explanation of the second style

- The 5 boxes are horizontally lined up.
- Box “F” is positioned in the bottom right corner of the window and stays there even when the window resizes.
- A box is 100 px wide and 150 px tall, there is a 10px dotted line on the left side of each box.
- When you mouse over a box, the cursor changes to a hand and the box and font colors change to purple and white, respectively.
- The font is the same as the first style.

Use your best judgement when a property is not mentioned above.

Problem 2 (20 pts)

Recreate the web page: TCU: Where Should Frogs Eat?



TCU: Where Should Frogs Eat?

Danniel Johnson

Last updated Jan 29, 2020

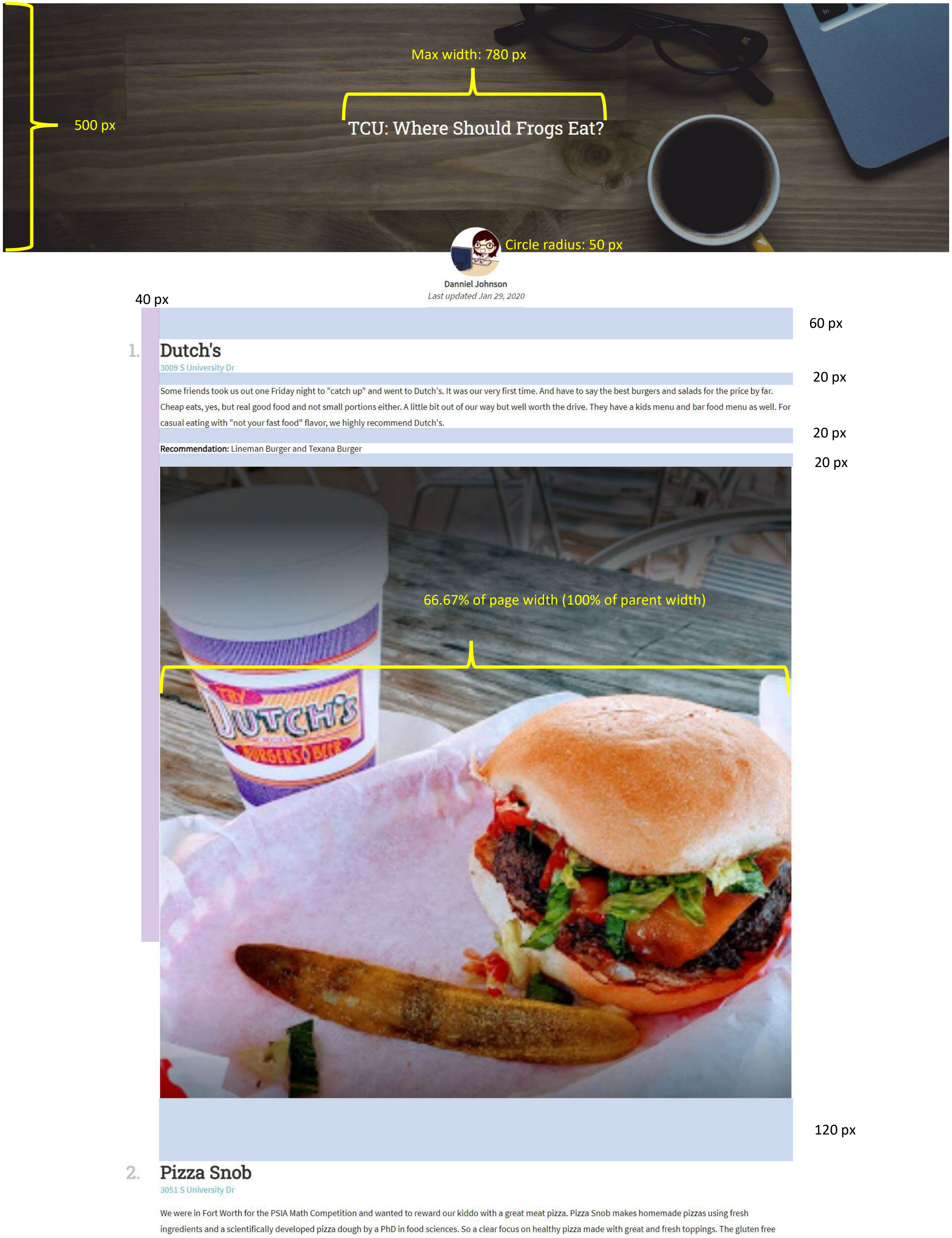
1. **Dutch's**

3009 S University Dr

Some friends took us out one Friday night to "catch up" and went to Dutch's. It was our very first time. And have to say the best burgers and salads for the price by far. Cheap eats, yes, but real good food and not small portions either. A little bit out of our way but well worth the drive. They have a kids menu and bar food menu as well. For casual eating with "not your fast food" flavor, we highly recommend Dutch's.

Starter code is provided. In the folder, you can find all the images you need, a full-size screenshot of the entire page and a tcu.html file. You need to complete tcu.html file and provide a style.css file under CSS folder.

Section measurements



TCU: Where Should Frogs Eat?

Danniel Johnson
Last updated Jan 29, 2020

1. **Dutch's**

3009 S University Dr

Some friends took us out one Friday night to "catch up" and went to Dutch's. It was our very first time. And have to say the best burgers and salads for the price by far. Cheap eats, yes, but real good food and not small portions either. A little bit out of our way but well worth the drive. They have a kids menu and bar food menu as well. For casual eating with "not your fast food" flavor, we highly recommend Dutch's.

Recommendation: Lineman Burger and Texana Burger

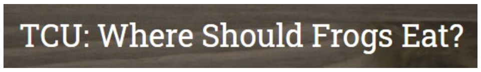
2. **Pizza Snob**

3051 S University Dr

We were in Fort Worth for the PSIA Math Competition and wanted to reward our kiddo with a great meat pizza. Pizza Snob makes homemade pizzas using fresh ingredients and a scientifically developed pizza dough by a PhD in food sciences. So a clear focus on healthy pizza made with great and fresh toppings. The gluten free

Font faces, sizes, and colors

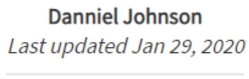
Page title:



TCU: Where Should Frogs Eat?

- The font face is 'Roboto Slab' and the fallback font is serif. This is a Google font. Learn how to use Google font in your project.
- The font size is 36px.
- The font weight is normal (not bold).
- The font color is white.
- The title is vertically and horizontally centered to the header image.

Profile text:

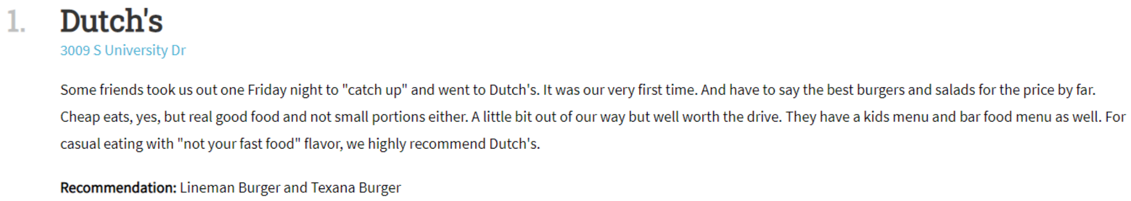


Danniel Johnson

Last updated Jan 29, 2020

- The font face is 'Source Sans Pro' and the fallback font is sans-serif.
- The font size is 18px.
- The font weight for the author name (“Danniel Johnson”) is bold.
- The font style for the last updated (“Last updated Jan 29, 2020”) is italic.
- The font color is rgb(51, 51, 51).
- The line height 24px.
- The gray line under the “Last updated” line is 10px under the text, 2px thick, and the color is #e6e6e6.

Body text:



1. **Dutch's**

3009 S University Dr

Some friends took us out one Friday night to "catch up" and went to Dutch's. It was our very first time. And have to say the best burgers and salads for the price by far. Cheap eats, yes, but real good food and not small portions either. A little bit out of our way but well worth the drive. They have a kids menu and bar food menu as well. For casual eating with "not your fast food" flavor, we highly recommend Dutch's.

Recommendation: Lineman Burger and Texana Burger

Heading

- The font face is 'Roboto Slab' and the fallback font is serif.
- The font size is 36px.
- The font weight is bold.
- The number’s color is rgba(0, 0, 0, 0.25).
- The title color is rgb(51, 51, 51).

Address

- The address under the heading is a link and is described in the Links section.

Paragraphs

- The font face is 'Source Sans Pro' and the fallback font is sans-serif.
- The font size is 18px.
- The line height 32px.
- The font color is rgb(51, 51, 51).

Recommendation

- The phrase “Recommendation:” is bold.

Links

- Each address of the page is a link. The URL needs to be able to link to the restaurant’s website.
- The link color is #42b4d6.
- The link is not bold, and the link is not underlined unless you hover over the link.
- When you hover over the link, the link becomes underlined.

Images

Header image

- The header is a background image (images/header.jpg).
- The background does not repeat.
- The background’s size is set to cover.
- The background position is anchored to the top.
- Its height is 500px.
- There is a semi-transparent color overlay on top of the header image, which is rgba(0, 0, 0, .3)
    - Note: You can achieve this using the following CSS rule: background-image: linear-gradient(rgba(0, 0, 0, .3), rgba(0, 0, 0, .3)), url(header.jpg);

Profile image

- The profile image is images/dj.png
- Its width is set to be 100px.
- It has rounded corners with radius 50%.
- It has a white border that is 1px thick.

Requirements

- Do not use an ordered list element (ol) to create the list layout. There’s nothing wrong with doing this, per se, but it involves some fairly esoteric CSS that we did not cover in class. We recommend you write out the numbers “1.”, “2.”, etc. in the HTML instead of using ol.
- Match the given appearance as closely as possible. It’s fine if your web page is off from our description by a few pixels here and there, but noticeable differences may result in a lower homework score.
- Use the latest Chrome for this project, and for all projects in this class.

For problem 3, 4, and 5. You are only allowed to use vanilla HTML, CSS, and JS. No library or framework is allowed. For problem 6, you must use Tailwind CSS.

Problem 3 (10 pts) Data Binding

A mysterious horned frog invented a weird notation called the mustache template for HTML pages:

Hello {{name}}!

The text inside {{}} is called a property. Usually, a property has a value for it in JSON format.

```
{
        name: 'SuperFrog'
}
```

After processing, the mustache template becomes:

Hello SuperFrog!

In this problem, you are going to implement a processor in vanilla JS.

Open databinding.html and finish the method bind(data). This method must examine all the text on the HTML page only within <div id="app"> and replace any text of the form {{property}} with the corresponding property of the data object.

For example, if you find an HTML element that has the text string:

```
<p>
        Today is the {{day}} day of month {{month}} of the year {{year}}
<p>
```

and the dictionary object argument is

```
{
        month: 'February',
        day: 12,
        year: 2020
}
```

you should update the text of the paragraph to be

```
<p>
        The date is the 30 day of month February of the year 2020
<p>
```

If the text specifies a property that is not defined in the data object the template should be replaced with an empty string. Your function need only handle properly formatted mustache templates.

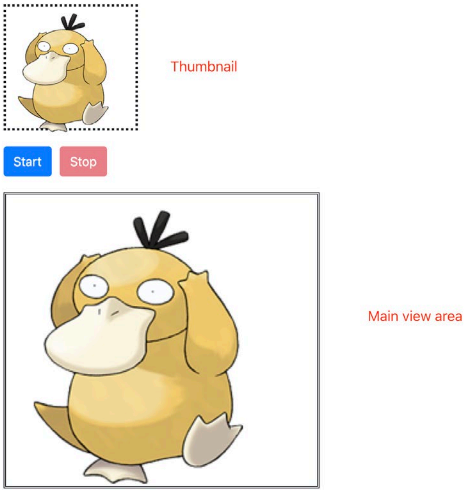
Useful links:

- Regular expression: https://www.w3schools.com/js/js_regexp.asp
- DOM nodeType Property: https://www.w3schools.com/jsref/prop_node_nodetype.asp
- String replace() https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/replace

Your solution shall work on any HTML page with mustache templates and a provided dictionary, you will lose 10 points if your solution is not a general one (in other words, you will lose 10 points if your solution only works for the sample I provided in the html).

Problem 4 (10 pts) Random Pokémon Selector

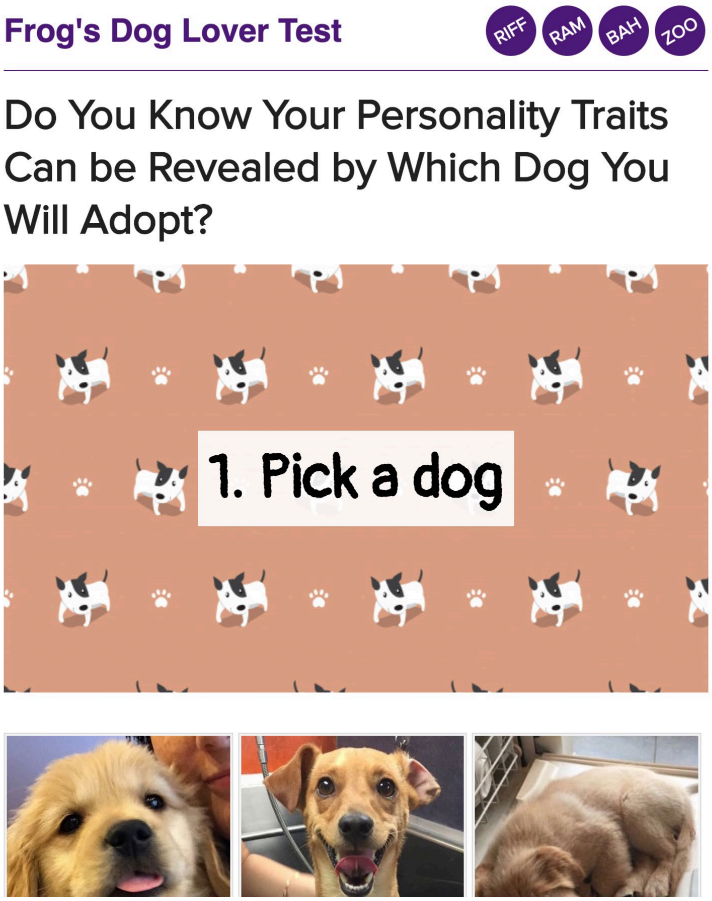
In this problem, you are going to implement a random Pokémon selector. Here is what the selector looks like when you first load the page:



At the beginning, “Start” button is enabled, “Stop” button is disabled. When a user clicks “Start”, the thumbnail picture starts to loop, displaying all Pokémon very fast (take a look at the mp4 file I put in the folder of this problem). During picture looping, “Start” button is disabled, “Stop” button is enabled. When a user clicks “Stop” button, the thumbnail stops looping, and you should display the selected Pokémon in the Main view area. “Start” button is then enabled, “Stop” button is disabled again. A user can click start and stop many times.

Problem 5 (20 pts)

In this problem, you need to finish a website call *Frog's Dog Lover Test*, using HTML, CSS, and JavaScript.



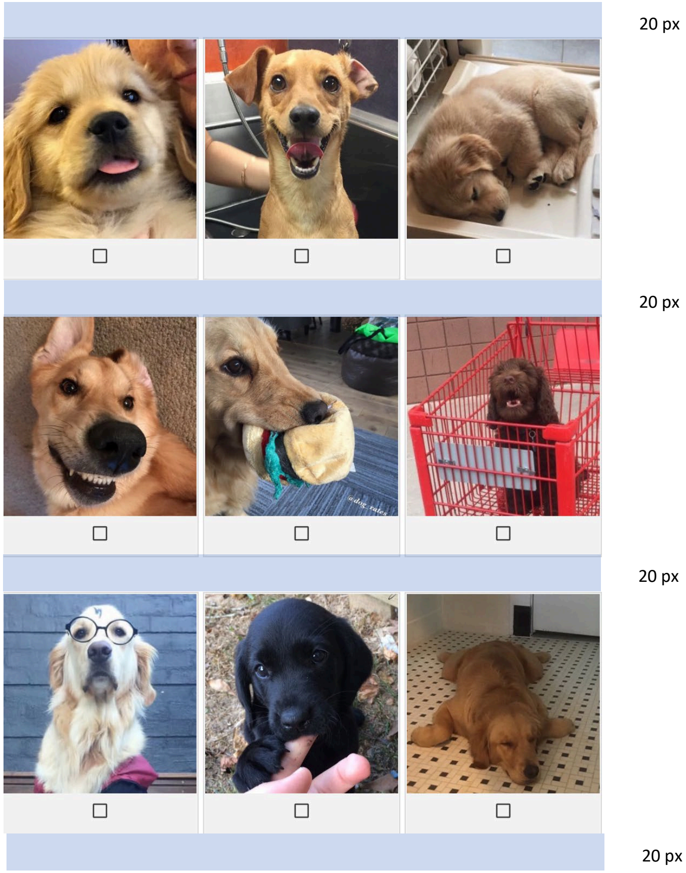
Frog's Dog Lover Test

Do You Know Your Personality Traits Can be Revealed by Which Dog You Will Adopt?

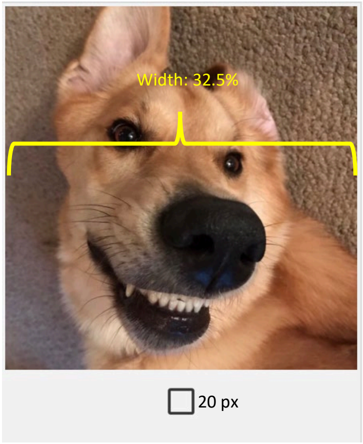
Starter code is provided. In the folder, you can find all the images you need, a full-size screenshot of the entire page, a constants.js file, a script.js file, a provided-style.css, a style.css and an index.html file.

Most HTML and CSS are provided for you already in index.html and provided-style.css (Yeah!), but you need to write JS in script.js, make CSS modifications in style.css and add contents in index.html if necessary.

Choice grid requirements



Choice



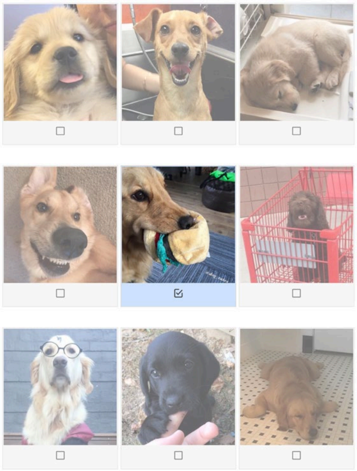
For each choice, background color is #f4f4f4, border is 1px and color is #dcdcdc, padding is 10px. Each image has different width and height, you need to figure out a way to get the images to “shrink” inside its container.

Picture needs to stay in the div nicely, otherwise -5 points

Behaviors

Once you finish the CSS, you need to add behaviors to this page by writing some JS code in script.js. There are three questions on this page, a user needs to select one item per question. Once all three questions are answered, at the bottom of the page, you should reveal the personality traits based on the three choices.

Question 1: pick a dog



For the selected item:

- The checkbox image should change from unchecked to checked (in this example, we use a pic instead of a HTML checkbox)
- Background color is #cfe3ff
- The checked image is can be found in images folder

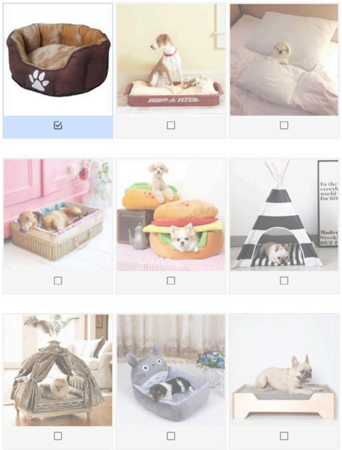
For the unchosen items:

- They should be made semi-transparent by setting their opacity: 0.6;
- Only the unchosen items for this question should change the style.

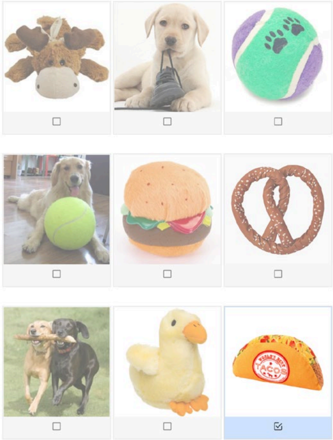
Changing an answer in one question:

- If the user has not completed the test (i.e. there is at least one unanswered question), they should be able to change their answer to a question by clicking a different answer.
- After the user has answered every question, the answer choices should lock and it should no longer be possible to change an answer, until the user clicks “Restart quiz” or refreshes the page.

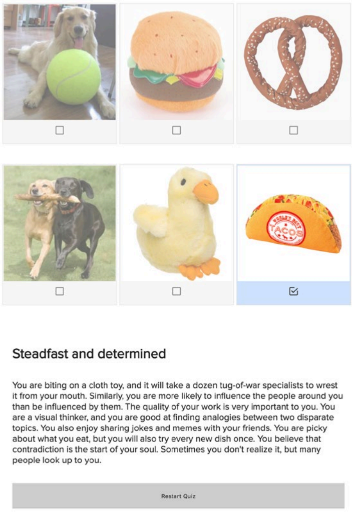
Question 2: pick a bed



Question 3: pick a toy



The following text only appears when a user selects all three choices. You can decide the style of the result section (margin, padding, color, etc) by yourself.



Steadfast and determined

You are biting on a cloth toy, and it will take a dozen tug-of-war specialists to wrest it from your mouth. Similarly, you are more likely to influence the people around you than be influenced by them. The quality of your work is very important to you. You are a visual thinker, and you are good at finding analogies between two disparate topics. You also enjoy sharing jokes and memes with your friends. You are picky about what you eat, but you will also try every new dish once. You believe that contradiction is the start of your soul. Sometimes you don't realize it, but many people look up to you.

Result button

- Background color is #cecece
- On hover, the color should change to #e0e0e0

Scoring the test

Take a look at index.html, you will find that each div has a data-choice-id and data-question-id attribute. There are 9 different values for the data-choice-id: blepping, happy, sleeping, dopey, burger, cart, nerd, shy and sleepy. They correspond to the RESULTS_MAP predefined in constants.js.

When the test is done, you should count the number of times a particular choice-id is selected. E.g., if a user selects happy in question 1, dopey in question 2 and happy again in question 3. The winner is happy, you should output RESULTS_MAP['happy'] as the result in the end of this page.

If there is a tie, i.e., if the user selects all distinct data-choice-ids, the answer to the question 1 should win.

-10 if web page behaves weirdly

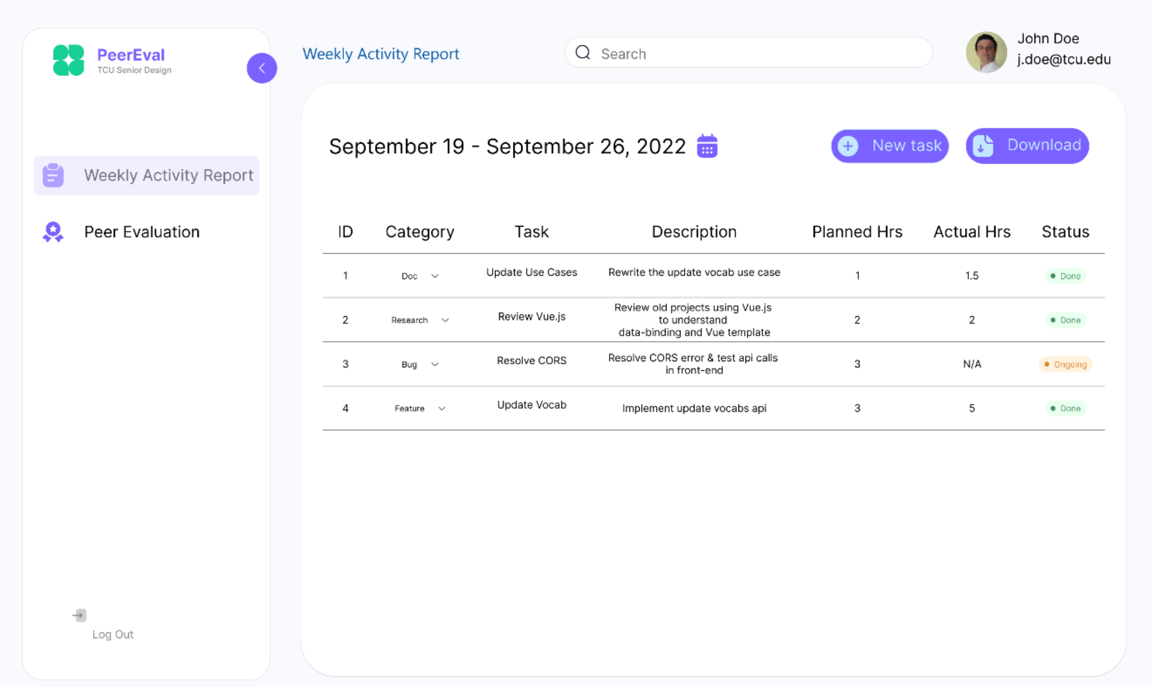
Resetting the test

If the user clicks the "Restart quiz" button, the page should reset to its original state and scroll back to question 1. Check out https://developer.mozilla.org/en-US/docs/Web/API/Element/scrollIntoView

Problem 6 (20 pts) Tailwind Project

In this problem, you need to use Tailwind to design three html pages.

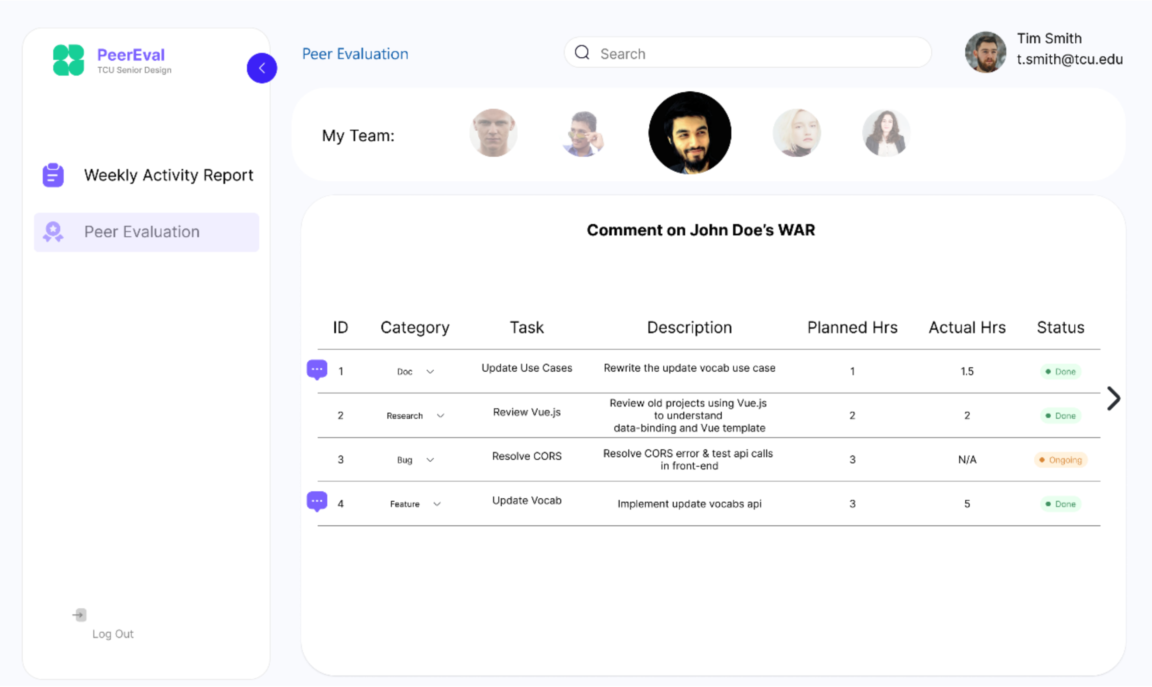
war.html



September 19 - September 26, 2022

ID	Category	Task	Description	Planned Hrs	Actual Hrs	Status
1	Doc	Update Use Cases	Rewrite the update vocab use case	1	1.5	● Done
2	Research	Review Vue.js	Review old projects using Vue.js to understand data-binding and Vue template	2	2	● Done
3	Bug	Resolve CORS	Resolve CORS error & test api calls in front-end	3	N/A	● Ongoing
4	Feature	Update Vocab	Implement update vocabs api	3	5	● Done

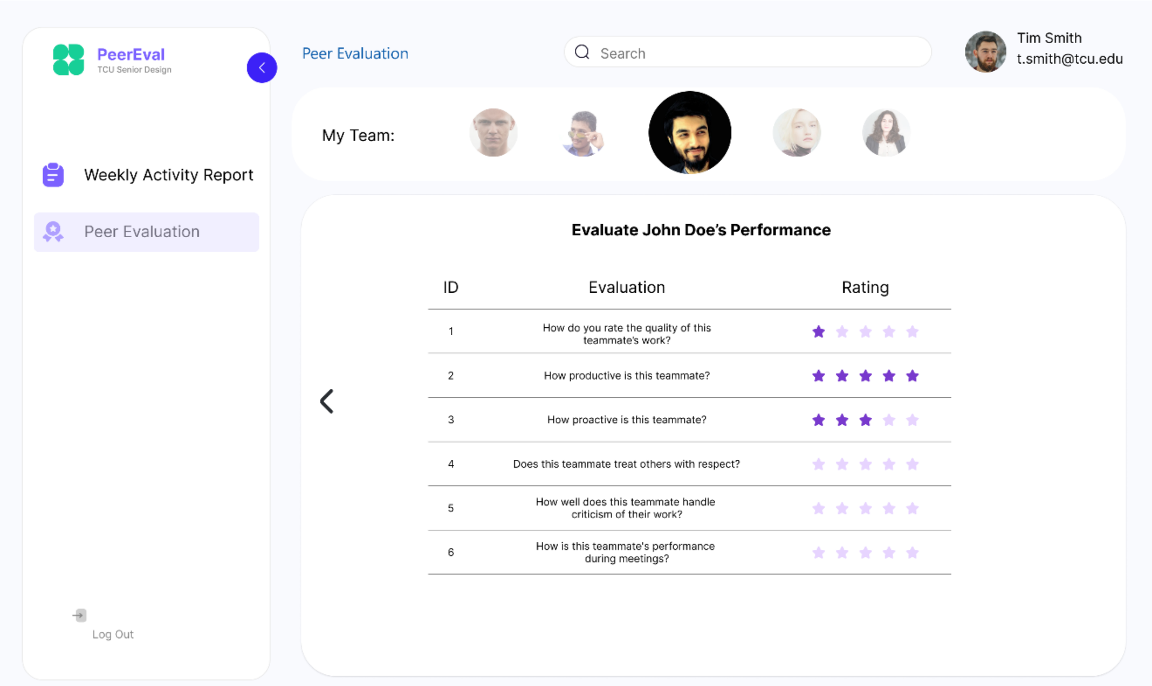
peer1.html



Comment on John Doe's WAR

ID	Category	Task	Description	Planned Hrs	Actual Hrs	Status
1	Doc	Update Use Cases	Rewrite the update vocab use case	1	1.5	● Done
2	Research	Review Vue.js	Review old projects using Vue.js to understand data-binding and Vue template	2	2	● Done
3	Bug	Resolve CORS	Resolve CORS error & test api calls in front-end	3	N/A	● Ongoing
4	Feature	Update Vocab	Implement update vocabs api	3	5	● Done

peer2.html



Evaluate John Doe’s Performance

ID	Evaluation	Rating
1	How do you rate the quality of this teammate's work?	★☆☆☆☆
2	How productive is this teammate?	★★★★★
3	How proactive is this teammate?	★★★☆☆
4	Does this teammate treat others with respect?	☆☆☆☆☆
5	How well does this teammate handle criticism of their work?	☆☆☆☆☆
6	How is this teammate's performance during meetings?	☆☆☆☆☆

Font Awesome

You may have noticed some lovely icons I put on the page, e.g., the download icon, star icon etc.

Learn how to include icons in your page https://fontawesome.com/

How to turn in this homework?

Create a folder called “HW1-Firstname-Lastname” under this folder, I should see six subfolders, Problem1 to Problem6.

Zip HW1-Firstname-Lastname and submit to TCU Online.