

HW2

Due March 1, 2023

For problem 1, 2, 3, 4 and 5. You are only allowed to use pure HTML, CSS, and JS. No library or framework is allowed.

For problem 6 and 7, you must use Bootstrap.

Problem 1 (20 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.

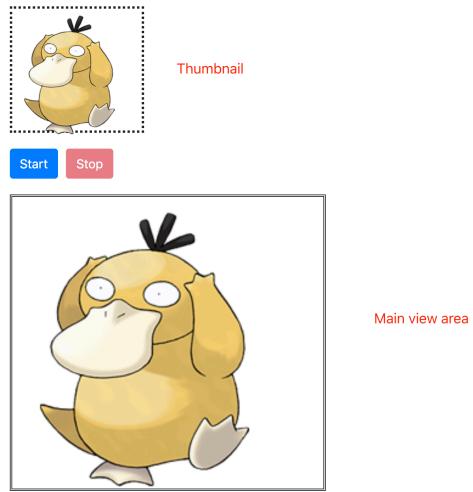
Useful links:

- 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 -15 points if your solution is not a general one (in other words, you will lose 15 points if your solution only works for the sample I provided in the html).

Problem 2 (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 3 (10 pts) Emojis

In this problem, you are going to implement commenting in emojis. Here is what the page looks like when you first load the page:



Please comment by clicking the emoji above: 😊

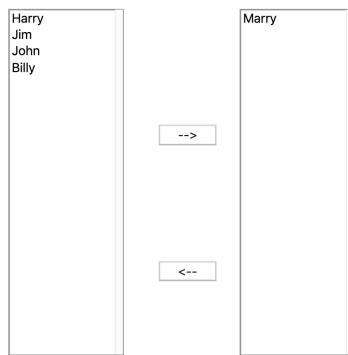
When a user clicks an emoji, it will appear in the comment area.

Take a look at the mp4 file I put in the folder of this problem.

Problem 4 (10 pts) Moving Names

Here is what the page looks like when you first load the page:

Select names and move them to left or right



In this problem, a user can select names on the left panel and move them to the right panel, and vice versa. Press command key (or control key) to select multiple names. Take a look at the GIF file I put in the folder of this problem.

Problem 5 (25 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

RIFF RAM BAH ZOO

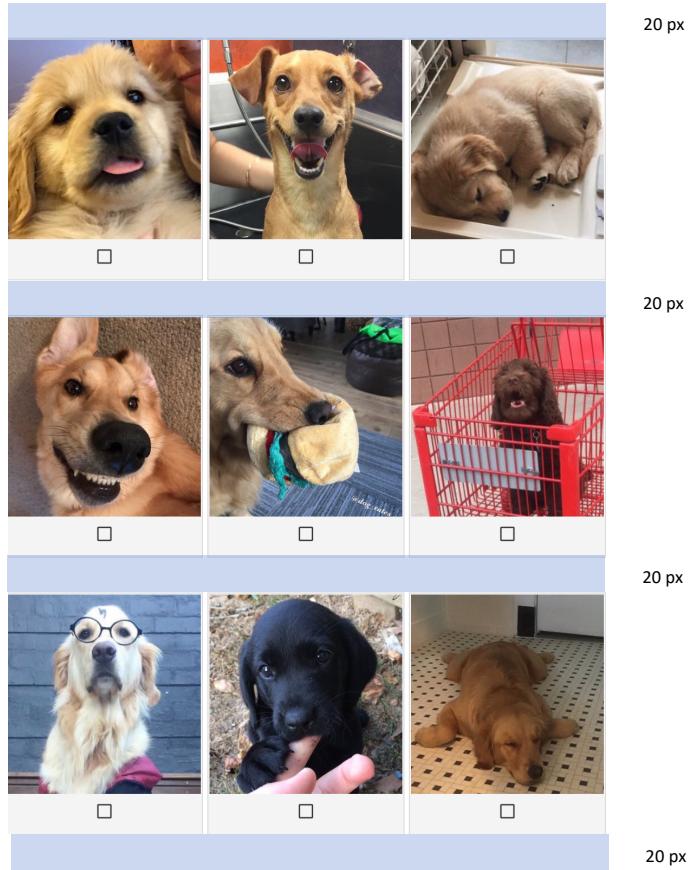
Do You Know Your Personality Traits
Can be Revealed by Which Dog You
Will Adopt?



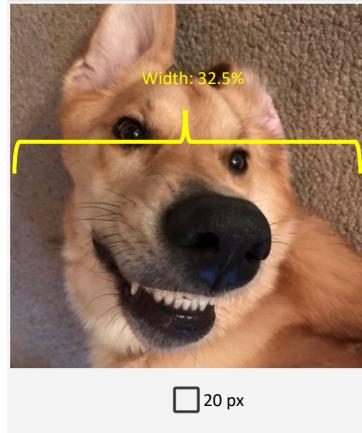
Starter code is provided. In the folder, you can find all the images you need, a full-size screenshot of the entire page, a constants.js file, a `script.js` file, a provided-style.css, a `style.css` and an `index.html` file.

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



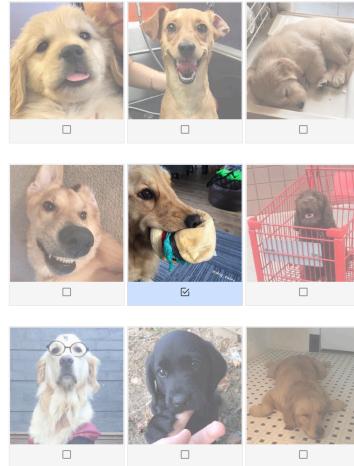
For each choice, background color is #f4f4f4, border is 1px and color is #dcdcdc, padding is 10px. Each image has different width and height, you need to figure out a way to get the images to “shrink” inside its container.

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

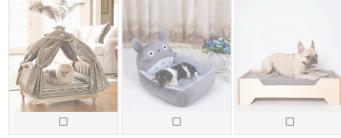
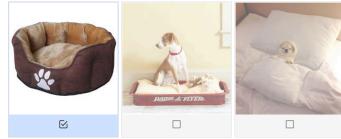
For the unchosen items:

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

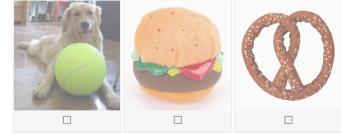
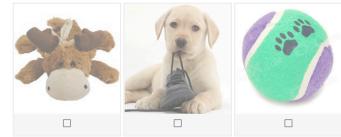
Changing an answer in one question:

- 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**.

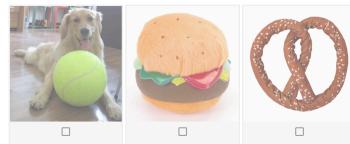
Question 2: pick a bed



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 like to enjoy sharing jokes and memes with your friends. You are picky about what you eat, but you 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.

[Restart Quiz](#)

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.

-15 if behave 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 (10 pts)

Design a web page to display 10 users' information requested from <https://jsonplaceholder.typicode.com/users>. (4 pts) For more information about this API, see <https://jsonplaceholder.typicode.com/> You must use fetch API.

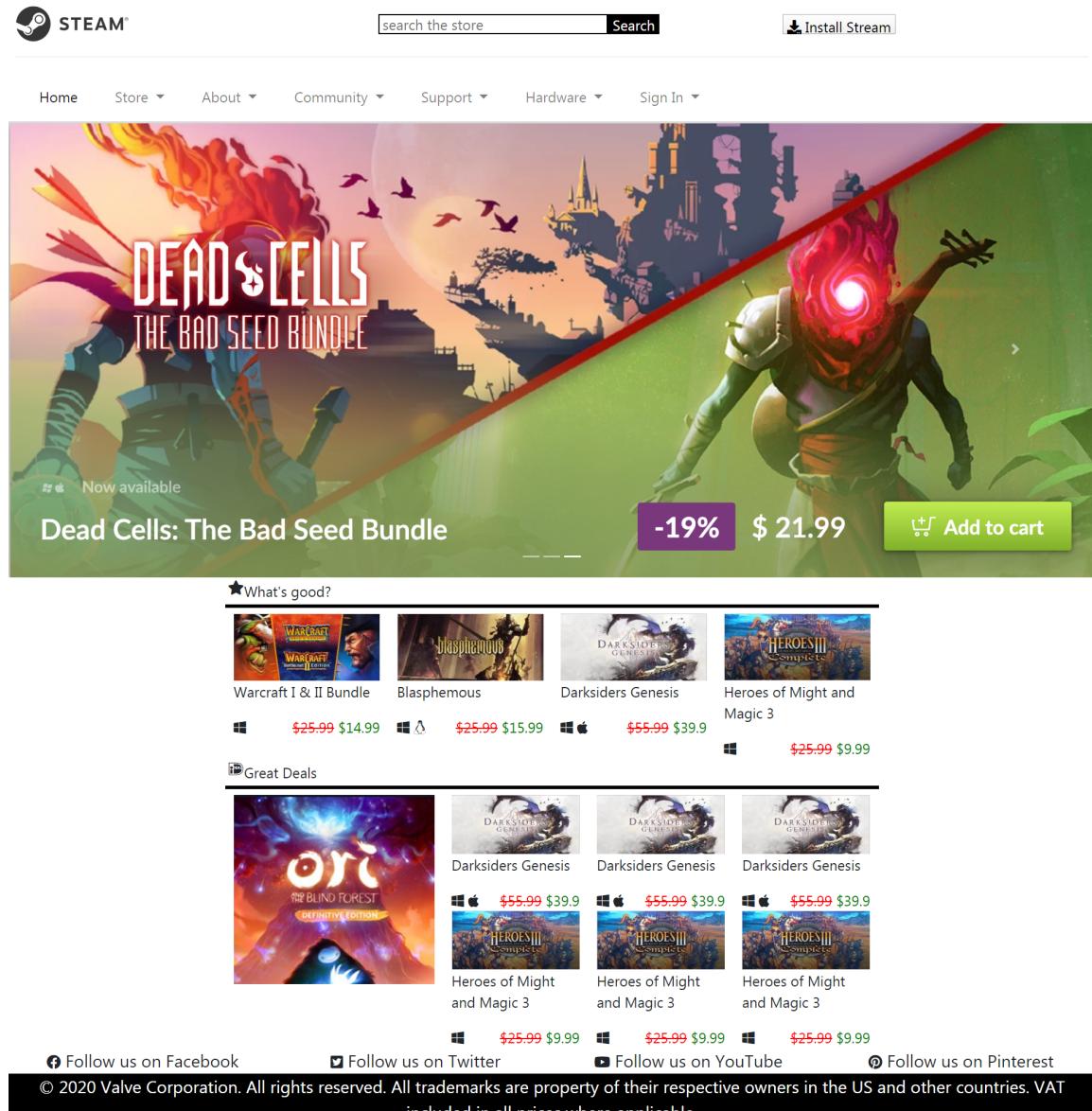
When displaying Emails, if the user's email ends with ".biz" use Bootstrap **success badge** style, otherwise, use **danger badge** style. (2 pts). See <https://getbootstrap.com/docs/5.1/components/badge/>

You shall also implement the "Sort by name" functionality. That is, display users in alphabetical order. (4 pts)

User Management				
Sort by name				
ID	Name	Email	Website	Company
1	Leanne Graham	Sincere@april.biz	hildegard.org	Romaguera-Crona
2	Ervin Howell	Shanna@melissa.tv	anastasia.net	Deckow-Crist
3	Clementine Bauch	Nathan@yesenia.net	ramiro.info	Romaguera-Jacobson
4	Patricia Lebsack	Julianne.OConner@kory.org	kale.biz	Robel-Corkery
5	Chelsey Dietrich	Lucio_Hettinger@annie.ca	demarco.info	Keebler LLC
6	Mrs. Dennis Schulist	Karley_Dach@jasper.info	ola.org	Considine-Lockman
7	Kurtis Weissnat	Telly.Hoeger@billy.biz	elvis.io	Johns Group
8	Nicholas Runolfsdottir V	Sherwood@rosamond.me	jacynthe.com	Abernathy Group
9	Glenna Reichert	Chaim_McDermott@dana.io	conrad.com	Yost and Sons
10	Clementina DuBuque	Rey.Padberg@karina.biz	ambrose.net	Hoeger LLC

Problem 7 (15 pts) Bootstrap Project

In this problem, you need to use **Bootstrap 5.1** to design this fake STEAM website.



Your work MUST include, Bootstrap Navbar, Carousel, Grid system, icons. You can add your own CSS if necessary.

All images are provided.

Font Awesome

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

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

How to turn in this homework?

Create a folder called “HW2_Firstname-Lastname” under this folder, I should see 7 subfolders, problem1 to problem7.

Zip HW2_Firstname-Lastname and submit to TCU Online.