Data Science and Data Visualization

Lab 1 – Introduction to HTML, CSS, SVG, and Javascript

Reference:

HyperText Markup Language (HTML): https://developer.mozilla.org/en-US/docs/Web/HTML,

Cascading Style Sheets (CSS): https://developer.mozilla.org/en-US/docs/Web/CSS

Scalable Vector Graphics (SVG): https://developer.mozilla.org/en-US/docs/Web/SVG,

JavaScript: https://developer.mozilla.org/enUS/docs/Web/JavaScript

Instructions

- Implement a HTML with additional files if needed (e.g. one for CSS, and one for JavaScript)
- Do not use D3 or any other libraries
- Read chapter 3 of the Interactive Data Visualization for the Web

Details

1. **Info (HTML & CSS)**

   Create an HTML web page with the title "Lab 1". It contains the following text:

   - Your name
   - Your student ID
   - The course tile "Data science and data visualization"
   - The lab title "Lab 1 – Introduction to HTML, CSS, SVG, and JavaScript"
   - The text "This is all my own work. I did not copy the code from any other source"

All of these lines except the last should be heading elements. The first line should be a level-3 header, and the other three should be a level-4 header. The final line should be a paragraph element.

You should then style the text using CSS so that your name is display in blue, your student id and the course title are rendered in bold text, and the lab name in is italics only (not bold). You must do this using CSS; do not use any HTML attributes except id and class.

2. **Initials (SVG)**

   In the same web page, create an SVG graphic (https://developer.mozilla.org/en-US/docs/Web/SVG) that displays your three initials using SVG. For example, consider the name "Nguyen Van A": its initials are "NVA". You do not need to draw periods after each letter.

You may only use SVG lines, polylines, rectangles, or paths. You do not need to use curves; you may use straight lines to render letters. (See this font (http://www.dafont.com/ds-digital.font) for ideas on how to draw letters without curves). **Do not draw your initials using SVG or HTML text elements**.

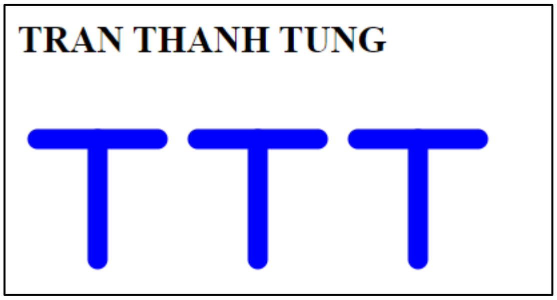
You should draw all content in a single svg element that has dimension 250x150, e.g.

```
<svg id=”initials” width=”250” height=”150”>

      <!—YOUR SVG HERE →

</svg>
```

For example



Hints:

- SVG coordinates start from the top-left part of the element
- Investigate the SVG fill and stroke attributes https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Fills_and_Strokes) to style your initials
- You can style SVG elements with CSS in a similar manner to HTML elements

3. Histogram (JavaScript + SVG)

In the same web page, create a JavaScript function that creates a histogram that shows the distribution of characters in a string. Use this function to draw a bar chart for the letters in your given name (without accents).

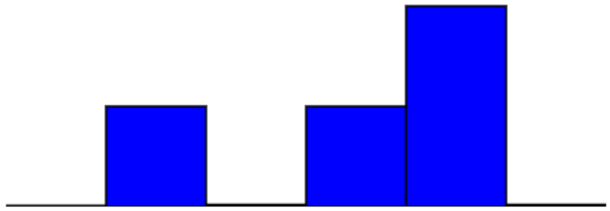
Your function should take an svg element and a string and update the svg element so that it displays a histogram of the string with six bins (A-D, E-H, I-L, M-P, Q-U, V-Z). The number of characters that are in those ranges will determine the height of the bars for each bin. Each bin should be 50 pixels wide and each letter occurrence should contribute 50 pixels of height to the bar. The bars should be filled with a blue color and the lines drawn in black.

```
<svg id=”histogram” width=”300” height=”400”>

      <!—YOUR SVG HERE →

</svg>
```

You should define a function createHistogram such that a call createHistogram(<svgElt>, "Tung") will create the histogram



Hints:

- You may use the following function to add the bar to the svg tag

```
function addEltToSVG(svg, name, attrs)
{
  var element =  document.createElementNS("http://www.w3.org/2000/svg", name);
  if (attrs === undefined) attrs = {};
  for (var key in attrs) {
    element.setAttributeNS(null, key, attrs[key]);
  }
  svg.appendChild(element);
}
```

- Make sure your JavaScript code occurs after the svg elements are defined. HTML is processed sequentially. You may also use an onload function (https://developer.mozilla.org/enUS/docs/Web/API/GlobalEventHandlers.onload)
- Give the zero bin a height of 1 pixel so that it will show on the screen
- You can access a character of a string in javscript via subscript
- You can compare characters with standard comparison operators (e.g. `<','>=') as in Java.
- JavaScript's String.toUpperCase (https://developer.mozilla.org/enUS/docs/Web/JavaScript/Reference/Global_Objects/String/toUpperCase) or String.toLowerCase (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/toLowerCase) methods should make your code more robust.
- Use JavaScript's document.getElementById (https://developer.mozilla.org/enUS/docs/Web/API/document.getElementById) function to get a reference to the svg element.

(Not required) For a challenge, add text labels to the bins.