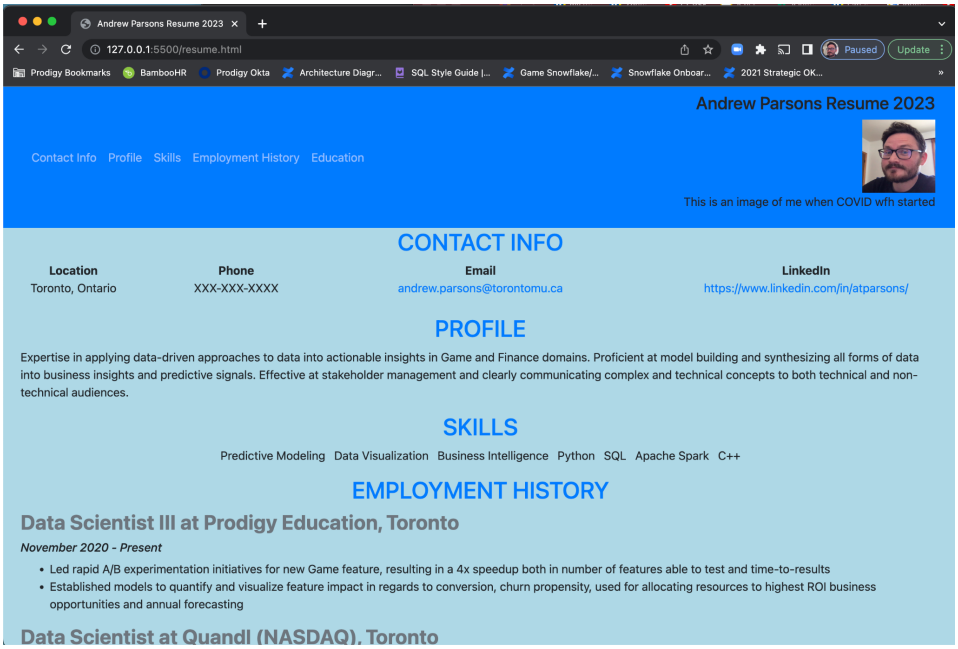
Name: Andrew Parsons
TMU ID: 500992021
Due Date: Feb 25, 2023
Github Repo Link: https://github.com/Parsonswlu/parsonswlu.github.io/tree/main/CCPS530_Lab4

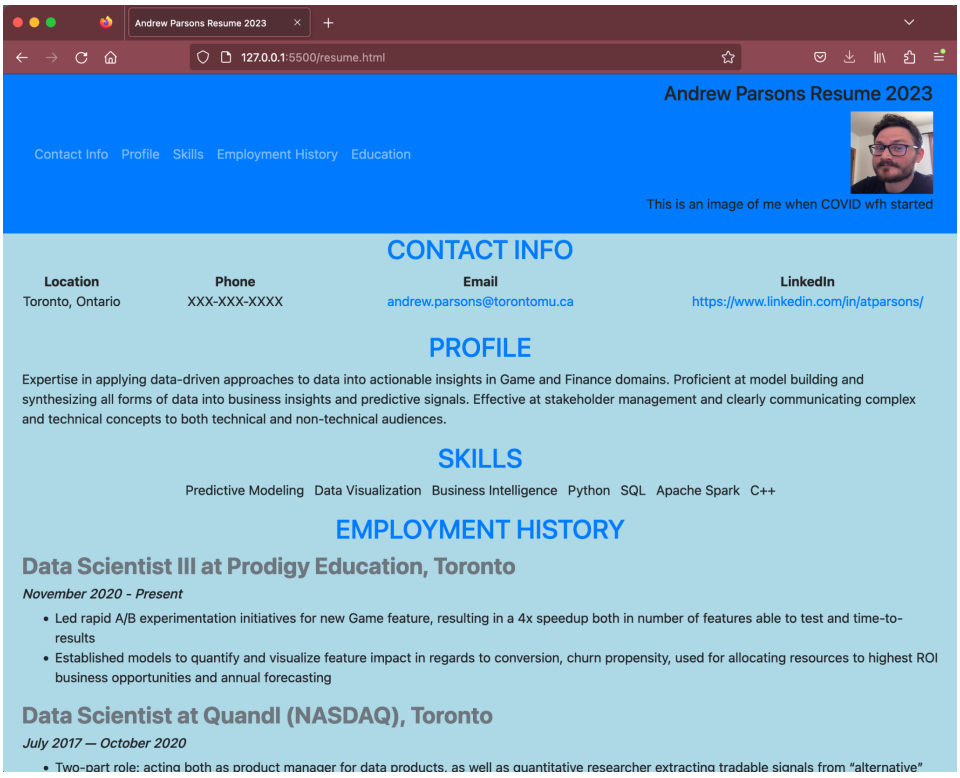
CCPS 530 - Web Systems Development - Lab 4

HTML Displayed on Browsers

- Google Chrome:

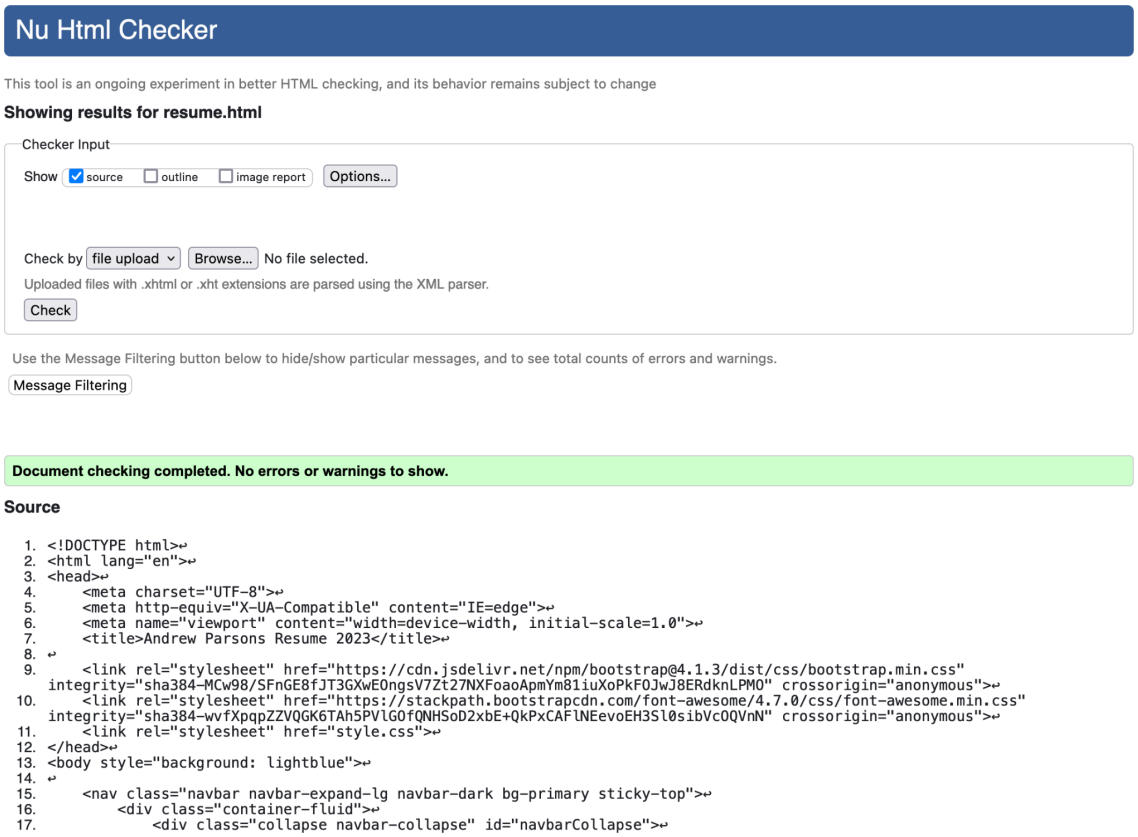


- Firefox:



HTML Checker

- Validator Results from https://validator.w3.org/



Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for resume.html

Checker Input

Show ☑ source ☐ outline ☐ image report Options...

Check by file upload Browse... No file selected.

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Check

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

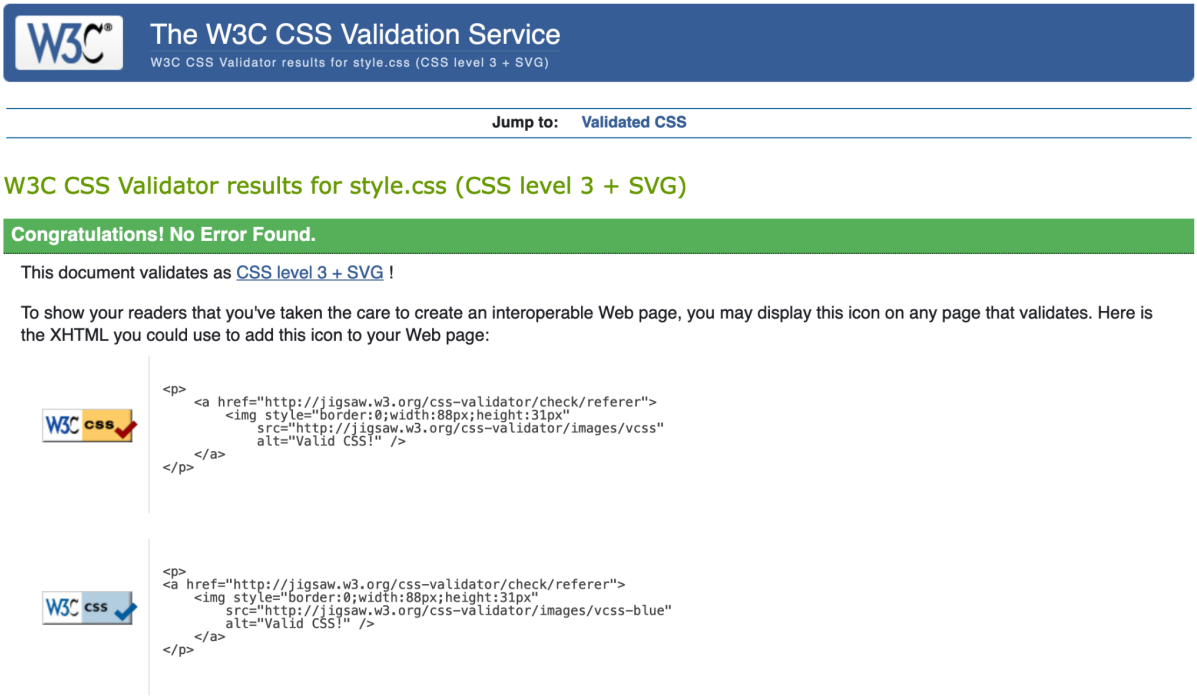
Document checking completed. No errors or warnings to show.

Source

```
<!DOCTYPE html>↵
<html lang="en">↵
<head>↵
    <meta charset="UTF-8">↵
    <meta http-equiv="X-UA-Compatible" content="IE=edge">↵
    <meta name="viewport" content="width=device-width, initial-scale=1.0">↵
    <title>Andrew Parsons Resume 2023</title>↵
↵
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap.min.css"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">↵
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css"
integrity="sha384-wvfXpqpZZVQGK6TAh5PVlGOfQNHSoD2xbE+QkPxCAFlNEevoEH3Sl0sibVcOQVnN" crossorigin="anonymous">↵
    <link rel="stylesheet" href="style.css">↵
</head>↵
<body style="background: lightblue">↵
↵
    <nav class="navbar navbar-expand-lg navbar-dark bg-primary sticky-top">↵
        <div class="container-fluid">↵
            <div class="collapse navbar-collapse" id="navbarCollapse">↵
```

CSS Checker

- Validator Results from https://jigsaw.w3.org/css-validator/



The W3C CSS Validation Service

W3C CSS Validator results for style.css (CSS level 3 + SVG)

Jump to: Validated CSS

W3C CSS Validator results for style.css (CSS level 3 + SVG)

Congratulations! No Error Found.

This document validates as CSS level 3 + SVG !

To show your readers that you've taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the XHTML you could use to add this icon to your Web page:

```
<p>
    <a href="http://jigsaw.w3.org/css-validator/check/referer">
        <img style="border:0;width:88px;height:31px"
            src="http://jigsaw.w3.org/css-validator/images/vcss"
            alt="Valid CSS!" />
    </a>
</p>
```

```
<p>
<a href="http://jigsaw.w3.org/css-validator/check/referer">
    <img style="border:0;width:88px;height:31px"
        src="http://jigsaw.w3.org/css-validator/images/vcss-blue"
        alt="Valid CSS!" />
    </a>
</p>
```

(close the img tag with > instead of /> if using HTML <= 4.01)

Technical Report

1. What HTML5 tags you used and why you chose them?

The following are HTML5 tags included (also described in same question from Lab 1 writeup)

- **html, head, body** tags
  - Default when creating the **`html:5`** scaffolding in Visual Studio.
  - Why: Minimum requirements for displaying content on a web page.
- **title** tag
  - Default when creating the **`html:5`** scaffolding in Visual Studio.
  - Located within the **head** section of the HTML
  - Why: Displays the title of the web page on the browser's tab ("Andrew Parsons Resume 2023")
- **link** tag
  - Used to link other files to the HTML file, in this case a "stylesheet" CSS file as well Bootstrap styles via CDN
  - Located within the **head** section of the HTML
  - Why: Separate HTML content from the style components, importing from a separate CSS file "style.css"
- **img** tag
  - Used to provide an image reference to display on the web page
  - Located within the **body** section of the HTML
  - Why: To include an image at the top of the page - putting a face to a name
- **a** tag
  - Used to create a clickable link to a (potentially) different web page, email address or file
  - Located within the **body** section of the HTML
  - Why: To provide a "mailto:" email and "https://" LinkedIn link on the page
- **p** tag
  - Used to separate content on the page by creating a new paragraph
  - Located within the **body** section of the HTML
  - Why: Isolate different content and style sections
- **h1 - h6** tags
  - Used to create different header levels
  - Located within the **body** section of the HTML
  - Why: Partly as a requirement of the Lab, partly to separate the largest header size / style (h1) to represent the most important section at the top ("Andrew Parsons Resume 2023") vs the smaller sections throughout ("Employment History", "Education", "Skills", etc.)
- **ul, li** tags
  - Used to create an unordered list (**ul**) on the page and a series of list items (**li**) in that list
  - Located within the **body** section of the HTML
  - Why: To present a list of items under the same topic - i.e. skills, responsibilities at a former company, etc.

New tags added since Lab 1 also include:

- **div** tag
  - Used to separate the various sections of the page
  - Located within the **body** section of the HTML

    - Why: Allows for easy separation of various sections and ability to reference and assign specific classes, attributes to that entire section.
- **nav** tag
    - Used to create a navbar across the page
    - Located within the **body** section of the HTML
    - Why: using the **nav** tag allows for easy reference to the top navigation panel of the page, including the ability to easily add links, buttons, etc. Adding the “sticky” modifier allows for it to always render at the top of the page regardless of how far you scroll on the page.

2. Compare and contrast JQuery vs plain JavaScript usage when it comes to using AJAX and navigating through the DOM of your HTML page.

- Using jQuery to navigate the DOM of the HTML allowed for much simpler syntax than pure JavaScript
    - I.e. referencing the class I needed to rotate the images could be done simply by $(‘.load-image’) rather than traversing the DOM from the top of the document (i.e. document.getElementsByClassName(".load-image"))
- The syntax to make an AJAX call in jQuery was also very straightforward
    - I.e. using $.ajax({url: .., type: .., dataType: .., success: ...}) in jQuery, versus the overhead of creating an XMLHttpRequest object and all the subsequent steps involved with actually executing on the request object.

3. Which web browsers did you use to view your page and are there rendering differences between browsers? You may add screenshots to your report.

- I tested the page with both Google Chrome and Firefox browsers
- I could not find any rendering differences between the browsers
- Screenshots are provided on page 1 of this report

4. How long did you spend on this lab? The length of time includes readings and research and code experimentation. State time involved in readings and research as well as code experimentation sessions.

- I spent approximately 6 hours working on this lab.
- It took over 5 hours to determine the correct syntax and structure to load in data from a json file, construct a loop that could cycle through various json element references, and wrap it all in an ajax GET request.
- It then took about another 30 minutes to do the validation, take screenshots, write up the technical report, etc.