# **Project Site Report – V4**

## Student Name: Javaris Taylor Date: 06/15/2020

## Project Version: V4

## See the Project Rubric & Checklist and review the assignment specification to ensure all expected work is included.

# Website URL

Complete the URL below with your home page for your Project submission. This is the address you type in to the browser to get to your site’s home page on GitHub Pages as described in the GitHub instructions in Module 9.

<https://jaytaylor-dev.github.io/project/V4/>

For V4, include your GitHub repository URL as well (see the Module 9 instructions for how to shape that:

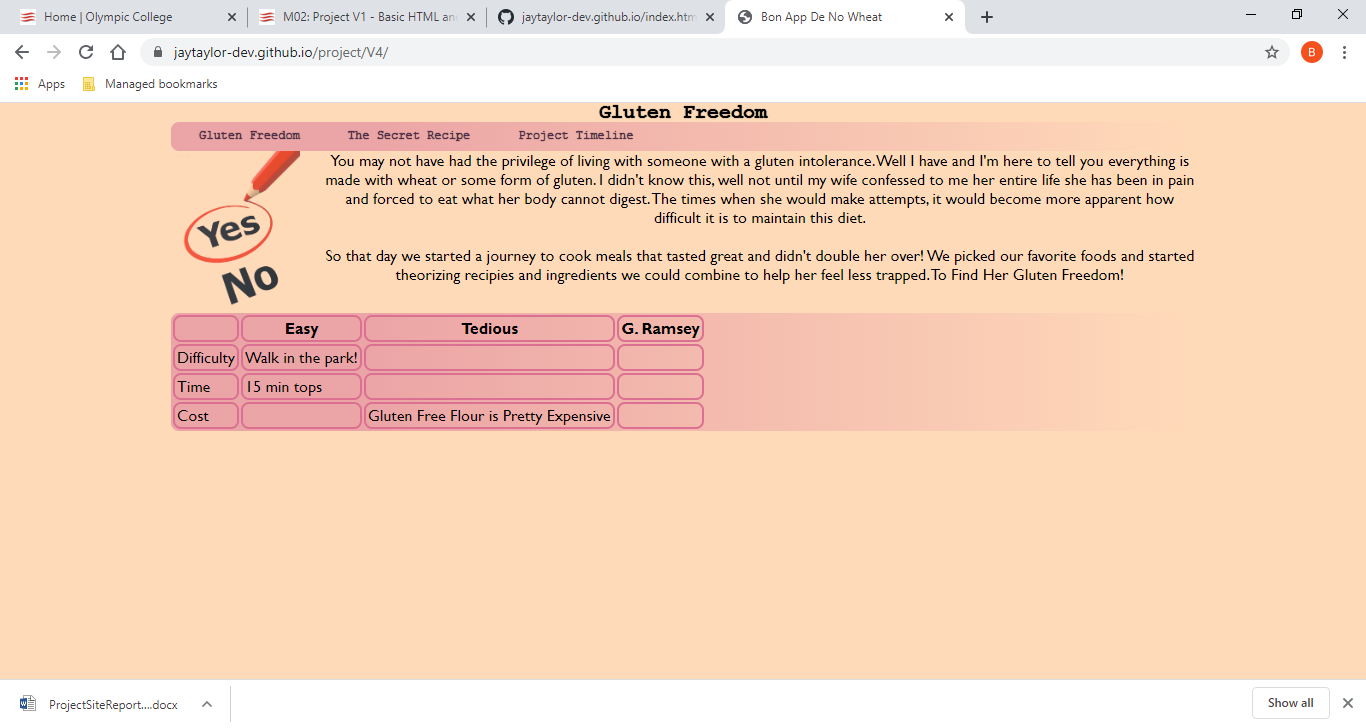
<https://jaytaylor-dev.github.io/project/index.html>

Indicate which of the requested module outcomes were not included in your files and explain why:

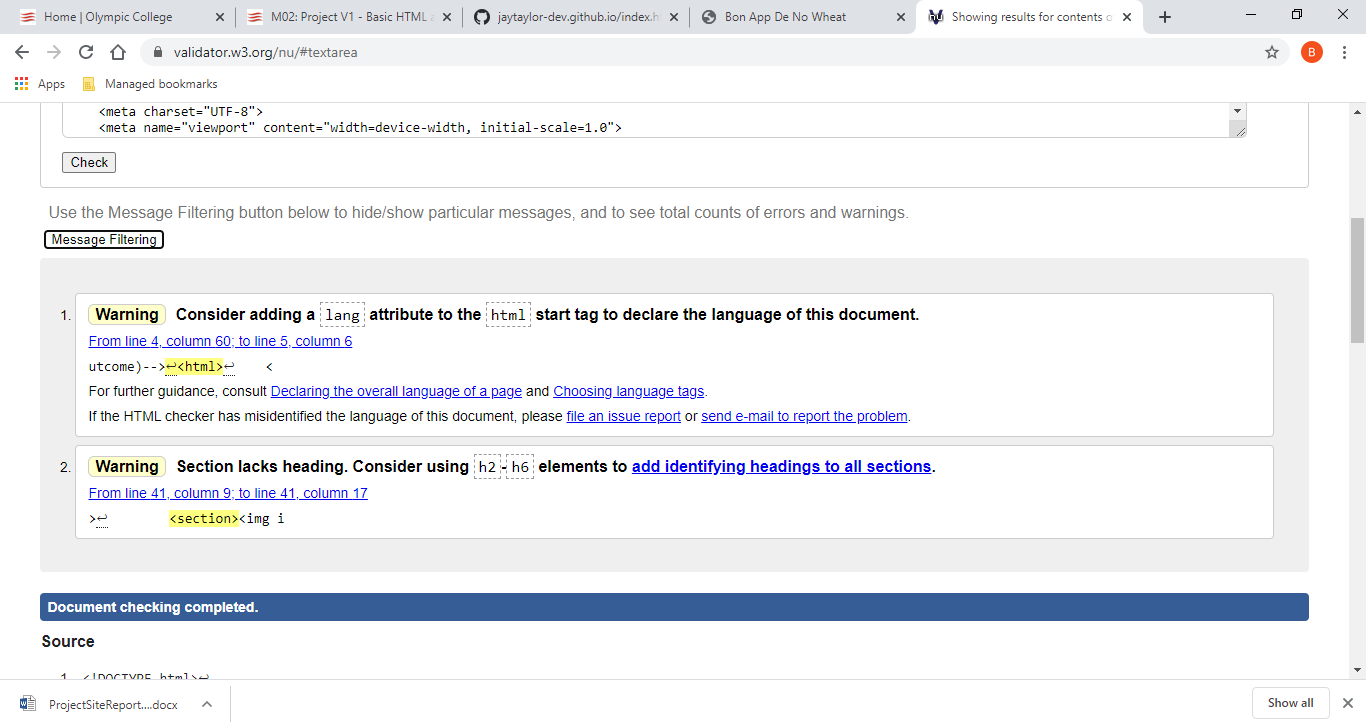
I Left out the audio modules because I did not use media, other than that I utilized everything requested.

# index.html

Include a screenshot of your index.html rendered on github.io (i.e. remote file in browser; include the address bar of the browser in the screenshot):



Include a validation screenshot of **url-based validation** of your index.html from github.io, using the w3c validator (include the url entered for validation in the screenshot):

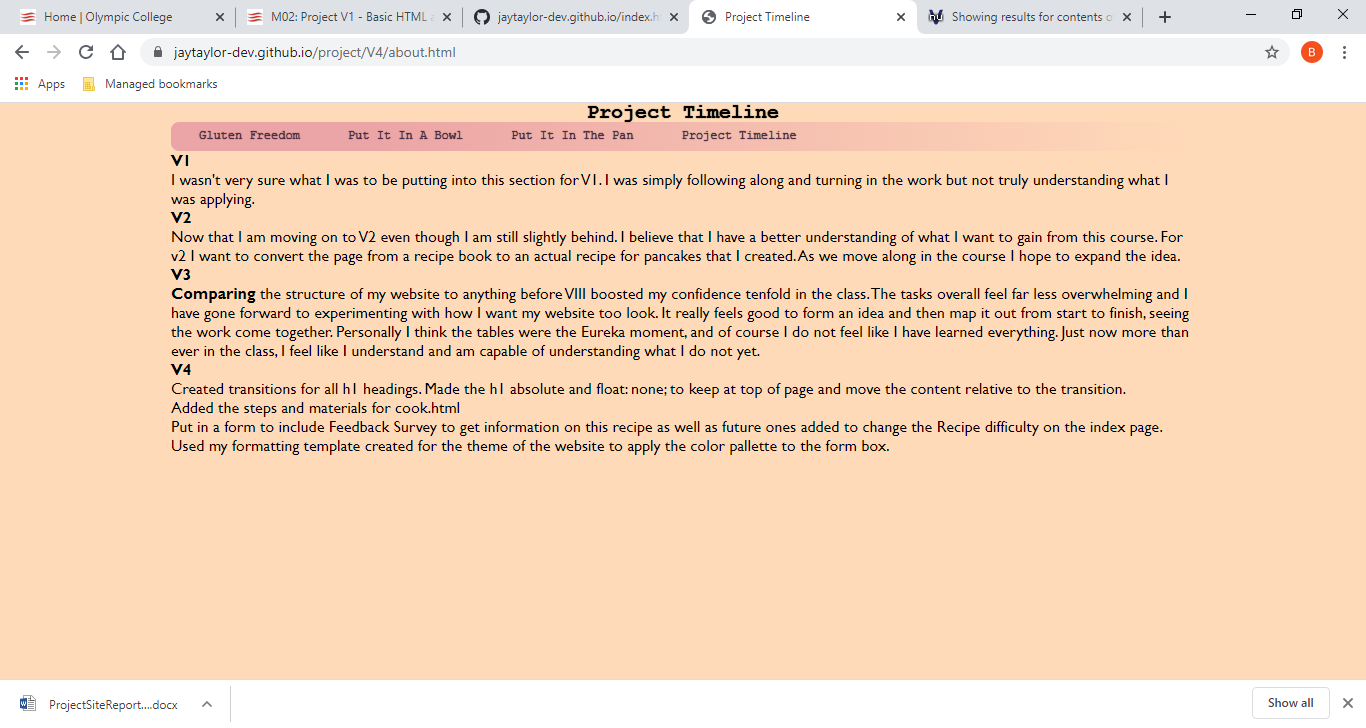


Explain/justify any remaining errors or warnings from the validation:

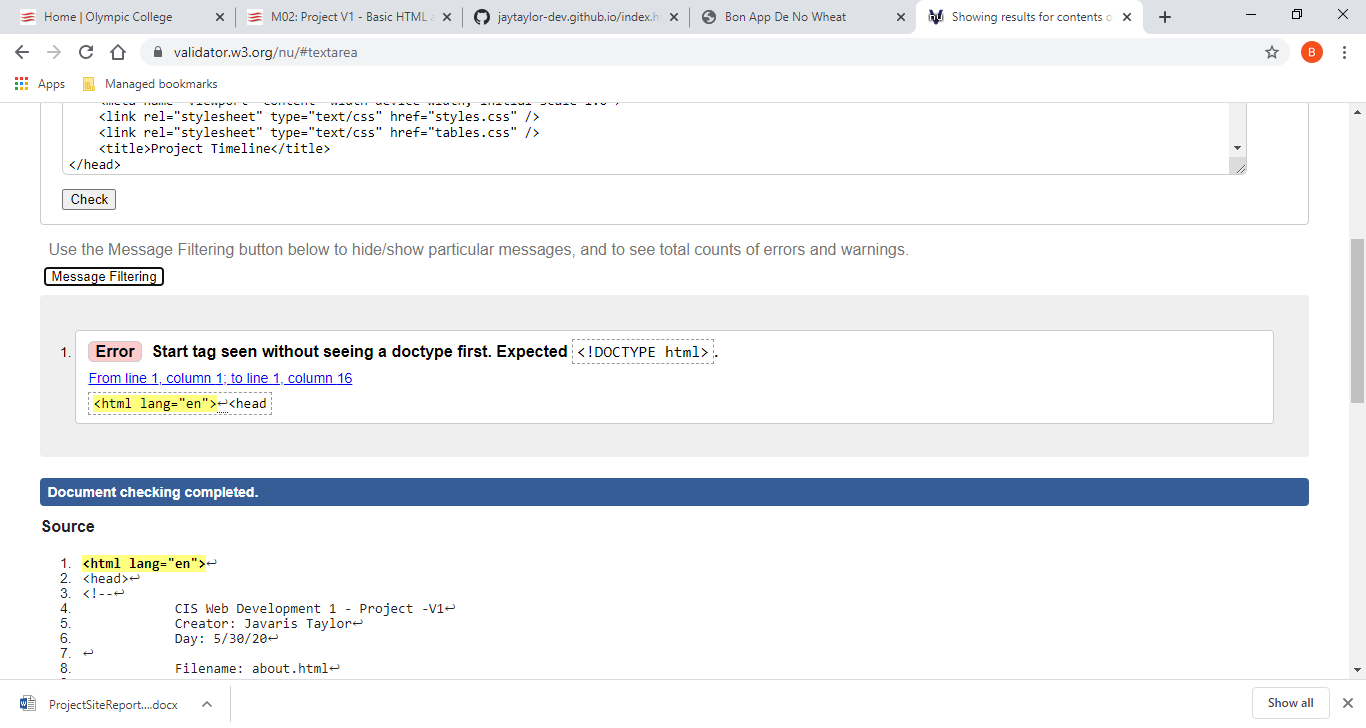
Language setting are not a necessary change as I used the default Visual Code default. Also I have the same format for all my sections and that is why I left them without a title to identify them.

# about.html

Include a screenshot of your about.html rendered on github.io (i.e. remote file in browser; include the address bar of the browser in the screenshot):



Include a validation screenshot of **url-based validation** of your about.html from github.io, using the w3c validator (include the url entered for validation in the screenshot):

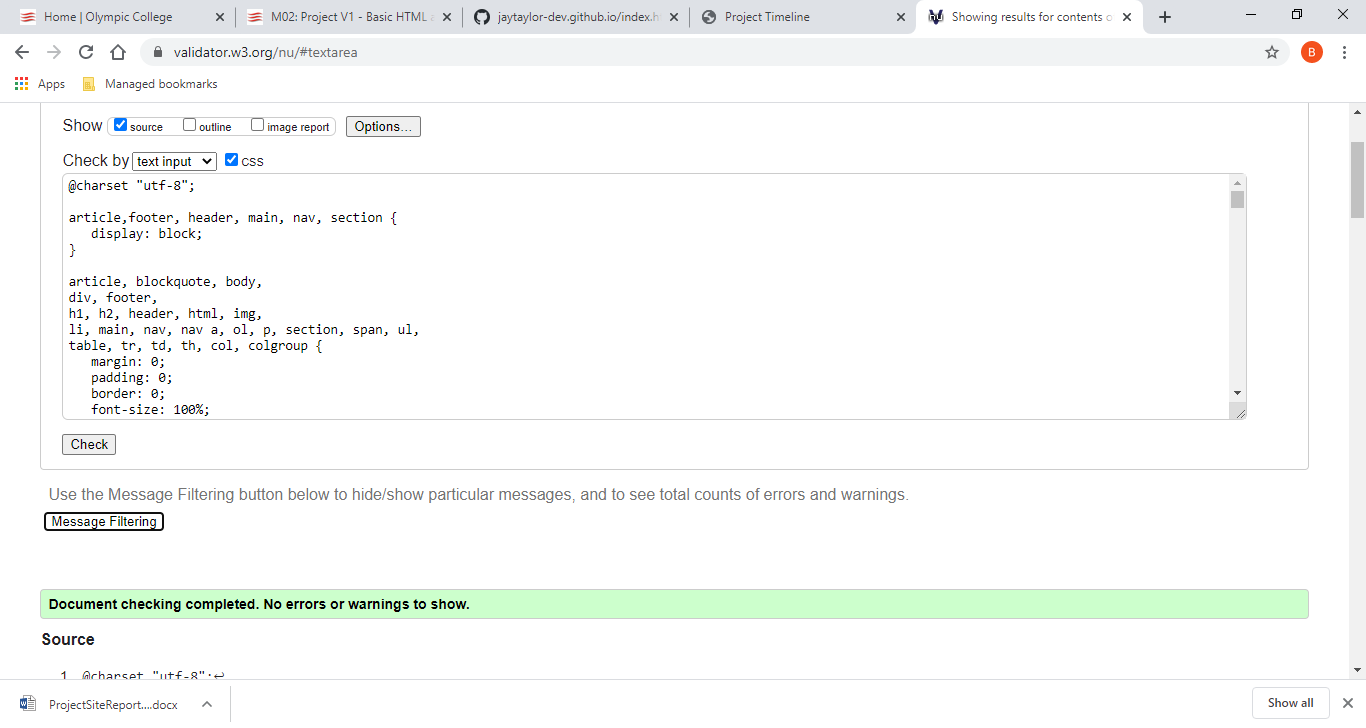


Explain/justify any remaining errors or warnings from the validation:

Just another language warning for the html head.

# styles.css

Include a validation screenshot of **url-based validation** of your styles.css from github.io, using the w3 validator (include the url entered for validation in the screenshot). To get to the CSS URL, use view-source on the HTML and then click on the <link> that has the href to the CSS file. This opens the CSS in its own tab in the browser, get the URL to it from your browser’s address bar:



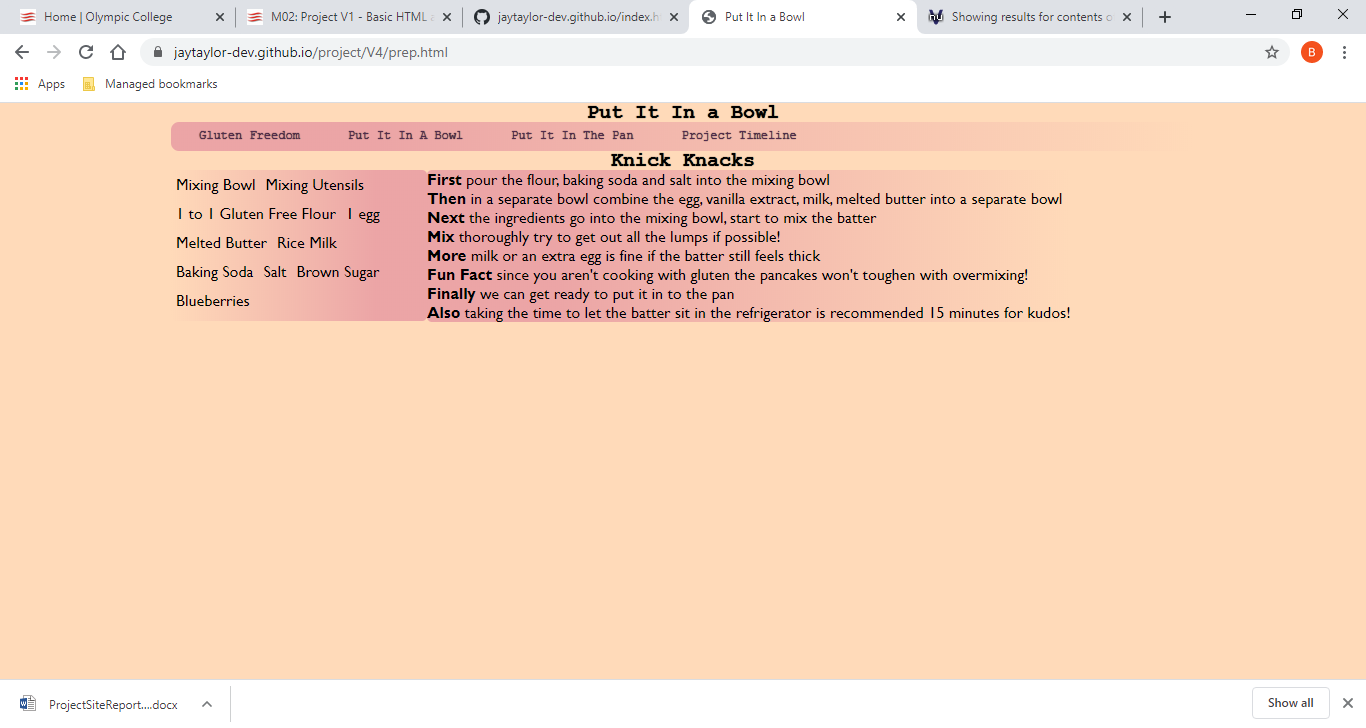
Explain/justify any remaining errors or warnings from the validation:

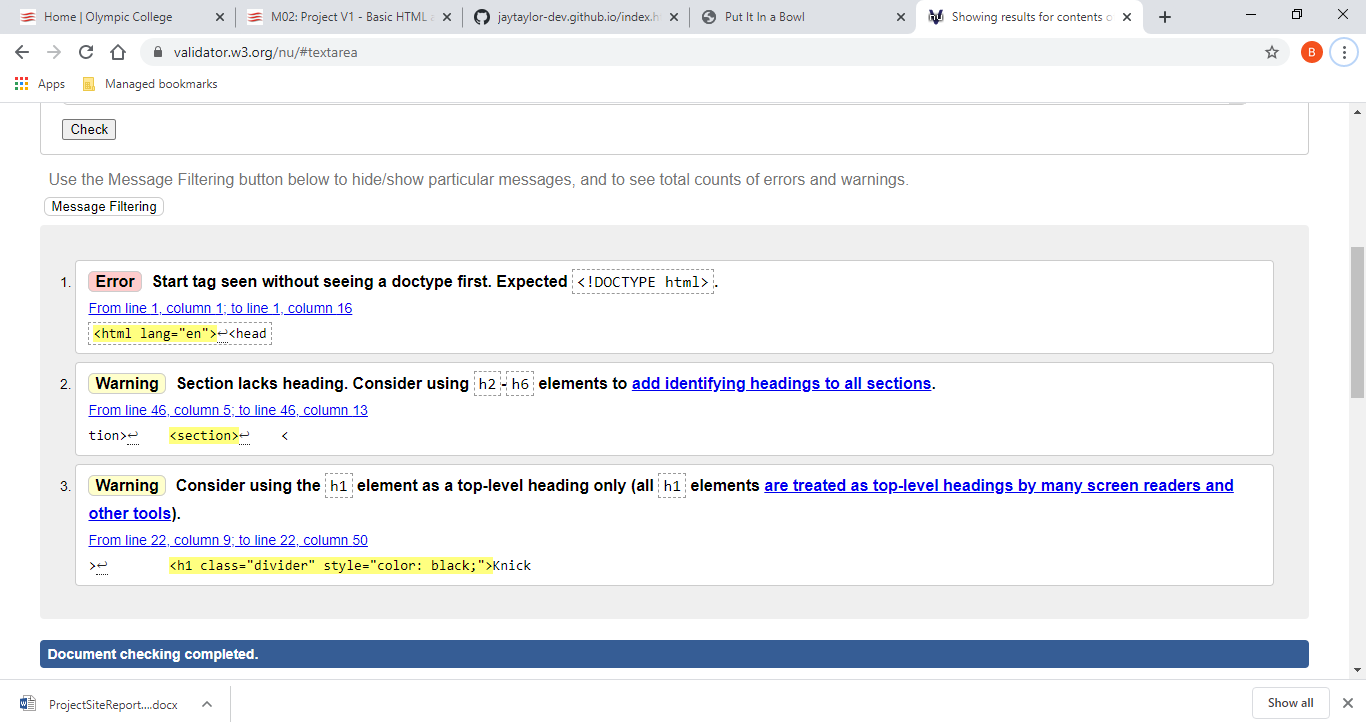
No warnings to show!

# Additional pages/code files

If you had any additional HTML or CSS files in which you put Module Outcomes, repeat the processes above for them as well.

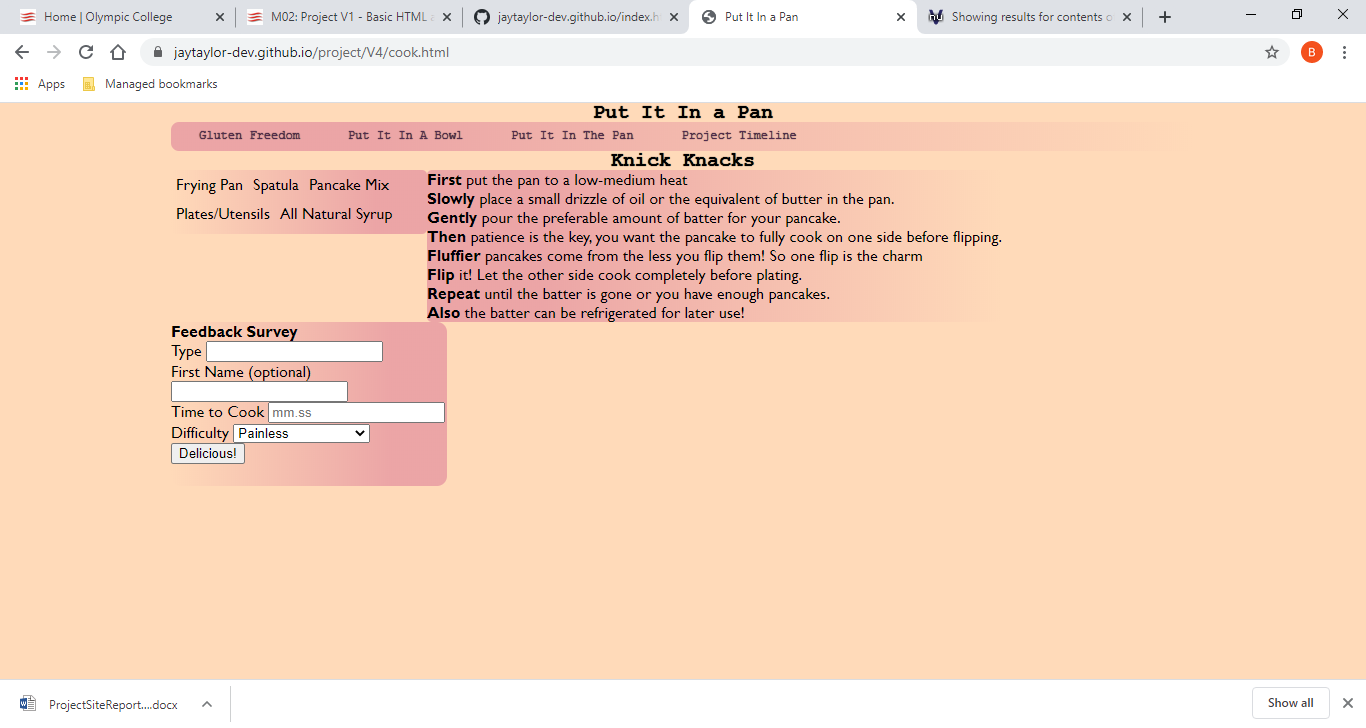
## prep.html

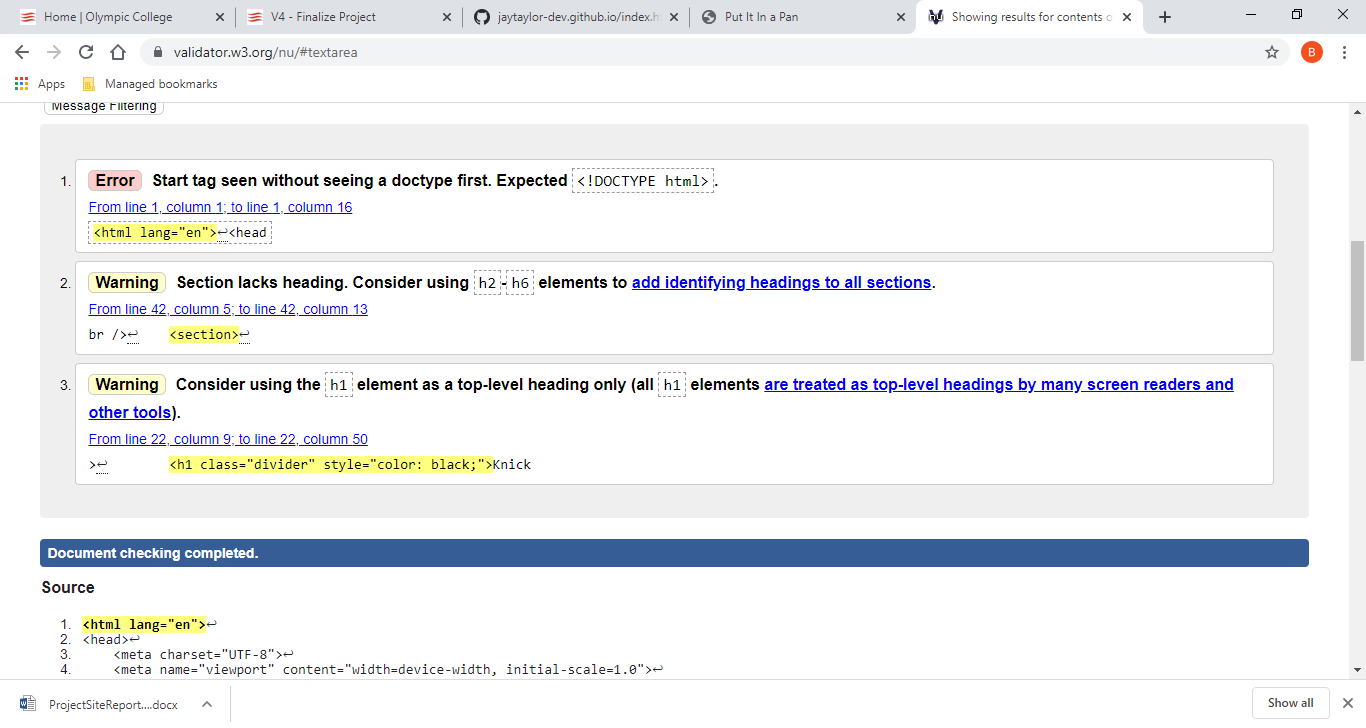




Language warning and heading warning. I have all my headings and sections set up for a theme and easy styling.

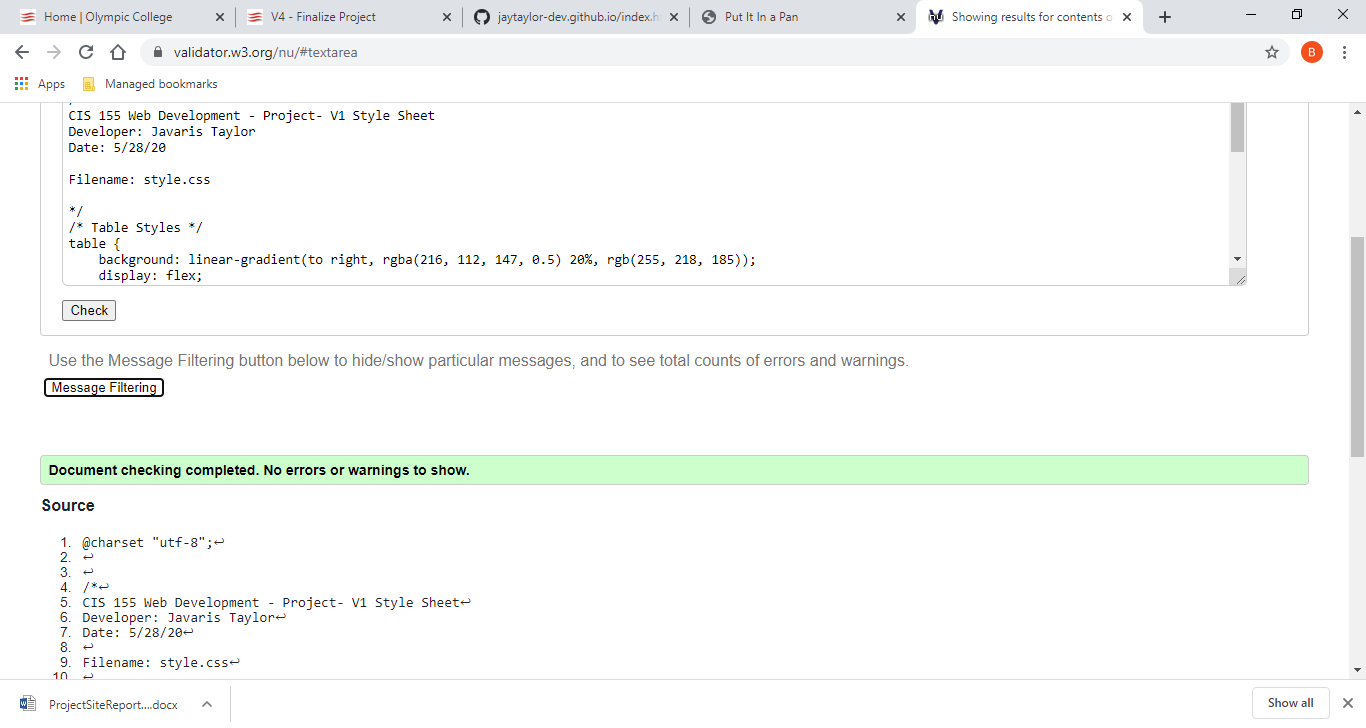
## cook.html





The same errors as before it is requesting that I diversify the headings, but that is how I am keeping the formatting and theme consistent.

## tables.css



# Wireframes

If this version requested any wireframes, include them here along with a discussion of how close you got to your original wireframe when coding your site.

# Congratulations on producing a great project! It will continue to evolve as we continue our learning.

See the Rubric and Checklist on the pages that follow, use them to check your work prior to submitting it. You are not required to write anything on the pages below, there are there to assist you in completing your assignment.

# Project Rubric

|  |
| --- |
| Process: What objectives are you demonstrating? (10 points) Comment your code to identify the learning objectives shown with your code. These are the learning objectives supplied in the Canvas course shell and identified in the assignment. The comment must include the module #, learning objective #, quote the learning objective, and state how the code shows it has been achieved. These are HTML or CSS comments in your code files. |
| Product: Are requirements met? (40 points) Do the files and rendered pages meet the requirements of the assignment (see the overview, the checklist below, and the particular assignment)?  Is there consistent & appropriate navigation between the pages?  Do all links work and all images appear?  Is it reasonably well organized and reasonably efficient in its design?  Does the code work for the right reasons? Is the code valid HTML5/CSS3 per the w3c validators? |
| Professionalism: Is your code/project well documented, appropriate, and on-time? (20 points) Is your code laid out and commented in a way that explain key / unique / complex elements of the document? This is beyond commenting the objectives.  Is white space used to show containment and to make it easy to scan the code?  Does the code and the report use professional English both in rendered content and comments, and are spelling, grammar, and punctuation correct?  Is the rendered content consistent with the site topic and project requirements?  Have you fully completed the Site Report?  Were w3c validator warnings sufficiently justified in the report? If sufficient justification is given for a warning, it will be permitted - contact me to determine that beforehand. Note, errors cannot be justified, they are errors and should be corrected in the code.  Were the Site Report and Site for Peer Review submitted on time? |

Unless otherwise noted your assignments are individual assignments and what you submit must be your own work. At any point during the term, you should be able to explain or duplicate any of your solutions for your assignments. I reserve the right to refuse to accept any assignment if you cannot demonstrate the ability to perform similar work when asked, or if you cannot explain your answer or approach that you have used. See [WAC 132C-120-065(1)](http://apps.leg.wa.gov/wac/default.aspx?cite=132C-120-065).

# Project Checklist

These are general requirements you may want to verify once you believe your project is complete:

1. Put comments near the top of any file that you code with your name, the date, the assignment, and an overview of the file’s purpose.
2. Use valid HTML5/CSS3 in your files. Even if a browser renders your page, that does not necessarily mean it is valid. Be sure to do these for a good start:
   1. Remember to put a HTML5 DOCTYPE at the top of each html file.
   2. HTML and CSS both specify the UTF-8 character set.
   3. HTML semantic tags are used in body to identify key elements such as header, nav, main, section, article, and footer.
3. HTML and CSS code uses indentation to show containment of elements, content, and properties.
   1. all child elements indent the same amount relative to their parent.
   2. all sibling elements indent to line up at the same level.
   3. consistent indentation: two parents that are siblings indent their children the same amount.
4. Nest all marked up page content within the body element; this includes header, footer, etc. No page content should appear above or below the body tags or within the head element.
5. Appropriate metadata is in the HTML head element: charset, title, author, keywords, description.
6. Add the required comments for the learning objectives being demonstrated; the comment should include the identifier, the objective, and explain how the nearby code shows your understanding of the objective.
7. Add any other comments to explain any code in your file that would need explanation if handed to another web developer. At a minimum, CSS should have area comments to note which types/sections of HTML the CSS applies to, as seen in the tutorials.
8. Appropriate navigation exists between the HTML pages, potentially with placeholders (href="#") for planned future pages. All pages should link to the About Page and potentially also to the Home page. How they relate to each other, depends on your website layout.
9. Pages share a common CSS file so the site has the same general appearance across its pages. A page may have its own additional CSS file or embedded CSS as well for features local to that page (not found on other pages).
10. Test your code locally prior to uploading it to GitHub, but test it again on GitHub. GitHub is “case-sensitive” for file names, so if your code does not agree with your exact file names, images and links may stop working. Once you have uploaded your files to GitHub, you get to your pages by navigating in a browser from <http://github.io/> to your personal account and repository name by manually changing the address bar by adding the names directly. This will bring up your index.html file, and you can navigate from there to your other pages.
11. Follow the instructions and requirements in the course shell for the project and this version of it.
12. Complete your site report: this includes information, screenshots and discussion as requested in the report and the assignment. V4 has you add wireframes and a discussion of them.

# Thank you for checking your work!