

Build a Portfolio Site

[Student Notes](#) [Code Review](#) 14 [Project Review](#) [Project History](#)

Does Not Meet Specifications

Design



SPECIFICATION

The page at minimum includes at least 4 images, title text, regular (paragraph) text, and a logo.

EXCEEDS SPECIFICATION

Reviewer Comments

Very nice work. Great that you've included a navbar and also personalized this page. The one nitpick I have is not all images are same size :(and it might be better you did have them to be same sizes to have a more eye-catching design

[Requirements to exceed the specification >](#)

SPECIFICATION

No <div> tags are without a class or id.

EXCEEDS SPECIFICATION

Reviewer Comments

Very nice use of HTML5 semantics tags

[Requirements to exceed the specification >](#)

SPECIFICATION

The page structure can be interpreted from the DOM.

MEETS SPECIFICATION

Reviewer Comments

Nice.

SPECIFICATION

A grid based layout such that the body has an element that serves as a container for rows and columns is used.

EXCEEDS SPECIFICATION

Reviewer Comments

Awesome

[Requirements to exceed the specification >](#)

Responsive



SPECIFICATION

Viewport meta tag is included in HTML.

MEETS SPECIFICATION

Reviewer Comments

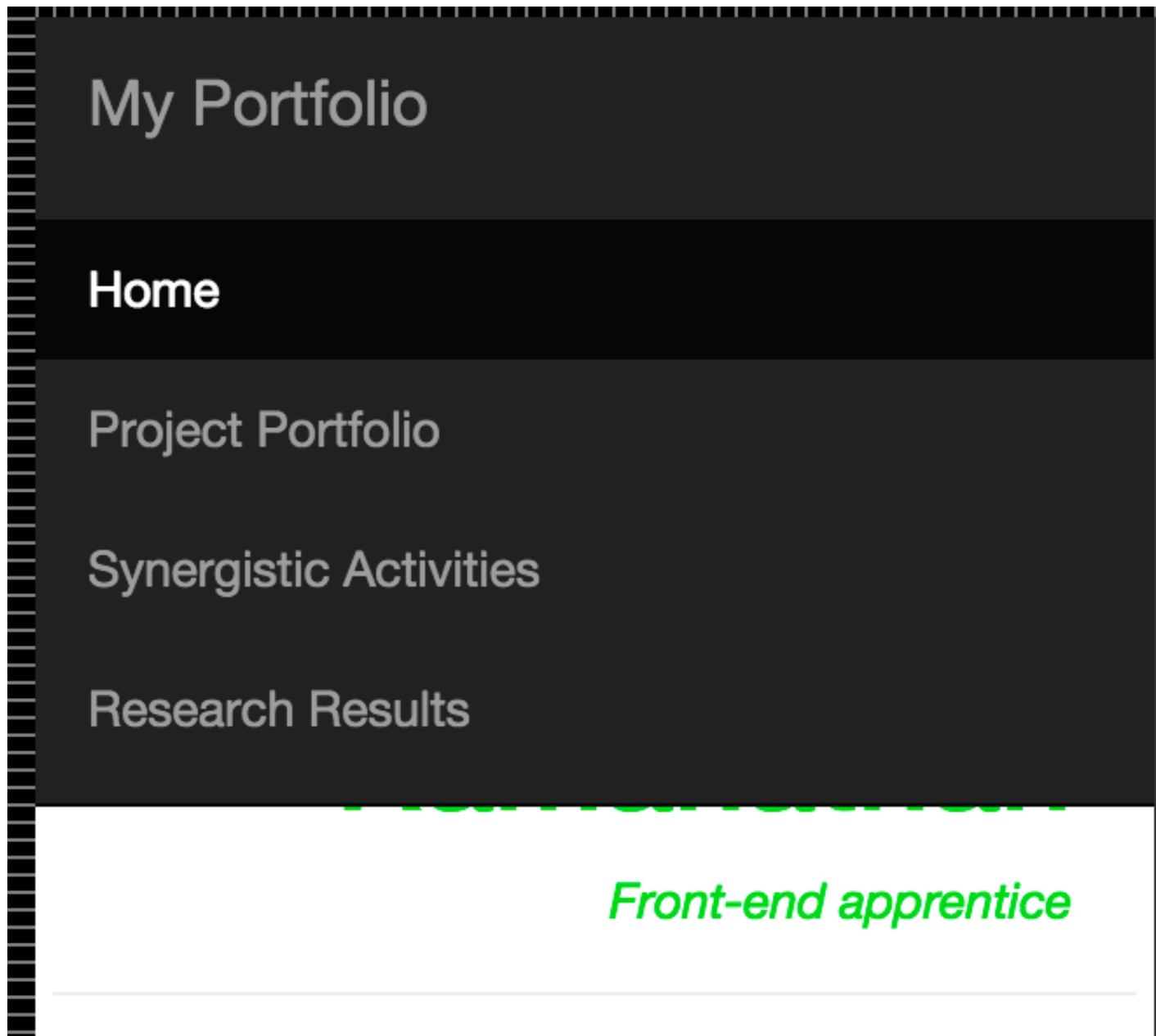
Great

SPECIFICATION

All content is rendered on all three devices. No content should be hidden on mobile devices.

DOES NOT MEET SPECIFICATION

Reviewer Comments



When I load your page on a iPhone4, the navbar covers your name :(You should try [responsive](#) navbars and Bootstrap offers capability for this.

All content is responsive and displays on all display sizes (Desktop, Tablet, Mobile).

MEETS SPECIFICATION

Reviewer Comments

Content is responsive and displays on all sized. Good

SPECIFICATION

If a CSS framework is used, classes provided by the CSS framework are used to make images responsive, otherwise media-queries are used to ensure responsiveness of images.

MEETS SPECIFICATION

Reviewer Comments

Good job using Bootstrap for responsive images. You should really consider [srcset](#). Not only does it give you "exceed spec" but it also makes your page more responsive.

[Requirements to exceed the specification >](#)

Separation of Concerns



How satisfied are you with this feedback?

 Resubmit Project

ly of the HTML

document. There are no `<style>` elements in the document.

Note: It is acceptable to include height and width of image elements.

DOES NOT MEET SPECIFICATION

Reviewer Comments

Oh just one style seems to have been missed and included in the HTML. You should move this over. I am marking it in the code review so that you can spot it easily. Good luck. And while you are at it, you might want to rearrange your code into subdirectories. Below is an example:

```
main
|- index.html
```

```
|  
|- css (a folder)  
    |- style.css  
|- js (a folder)  
    |- script.js
```

Code Quality



SPECIFICATION

CSS Formatting Rules -

- Code does not have trailing white spaces.
- Indentation is consistent (either all tabs or all 2 spaces or all 4 spaces etc).
- Code indents all block content, that is rules within rules as well as declarations to reflect hierarchy and improve understanding.
- Code uses a semicolon after every declaration for consistency and extensibility reasons.
- Code always uses a space after a property name's colon, but no space between property and colon, for consistency reasons.
- Code always use a single space between the last selector and the opening brace that begins the declaration block.
- Code always start a new line for each selector and declaration.
- Code always put a blank line (two line breaks) between rules.
- [Optional] Code uses double quotation marks for attribute selectors or property values. Do not use quotation marks in URI values (url()).

DOES NOT MEET SPECIFICATION

Reviewer Comments

There are some rules that still need your attention. Like in the case of HTML, a [beautifier](#) will help a great deal. However, I would still quickly confirm that the beautifier did the job using the [style guide](#) as a guide.

SPECIFICATION

HTML Style Rules -

- HTML documents Use HTML5 `<!doctype html>`.
- Code passes [HTML](#) and [CSS](#) validators.
 - *[Optional] Code does not use entity references unless necessary e.g. characters with special meaning in HTML (like `<` and `&`) as well as control or “invisible” characters (like no-break spaces).
- [Optional] Code omits type attributes for style sheets and scripts.

DOES NOT MEET SPECIFICATION

Reviewer Comments

Comments are great but your syntax for comments isn't correct and that's causing your HTML file to fail validation. I use SublimeText as my editor and I use keyboard shortcuts like `Control+/*` which inserts a comment placeholder and I then fill up details. (Alternatively you can enter text or something and use the keyboard shortcut on that line. SublimeText will convert the text to comments)

Here's how your new comment line will look:

```
<!-- Using bootstrap framework -->
```

SPECIFICATION

HTML Formatting rules -

- All code (HTML element names, attributes, attribute values) is lowercase (except text/CDATA).
- Code does not have trailing white spaces.
- Indentation is consistent (either all tabs or all 2 spaces or all 4 spaces etc).
- Code uses a new line for every block, list or table element and indent every such child element (it's acceptable to put all li elements in one line).
- [Optional] When quoting attribute values, code uses double quotation marks.

DOES NOT MEET SPECIFICATION

Reviewer Comments

The only issue here is that you have some indentation inconsistency problems. I personally use a [beautifier](#) that helps me instantly correct all these issues. Good luck.

SPECIFICATION

General Meta Rules -

- HTML templates and documents Use UTF-8 encoding. (no BOM) i.e. `<meta charset="utf-8">`.
*[Optional] Mark todos and action items with TODO:.

MEETS SPECIFICATION

Reviewer Comments

Nice.

SPECIFICATION

CSS Style Rules -

- Code uses meaningful or generic ID and class names that are as short as possible but as long as necessary.
- Code does not use element names in conjunction with IDs or classes.
- Code uses shorthand properties where possible.
- [Optional] Code omits unit specification after 0 values.
- [Optional] Code includes leading 0s in decimal values for readability.
- [Optional] Code uses 3-character hexadecimal notation where possible.
- [Optional] Code separate words in ID and class names by a hyphen.
- [Optional] Code avoids user agent detection as well as CSS "hacks"—try a different approach first.

MEETS SPECIFICATION

Reviewer Comments

Looks good.



Learn the [best practices for revising and resubmitting your project](#).



Have a question about your review? Email us at review-support@udacity.com and include the link to this review.

INFORMATION

[Nanodegree Credentials](#)
[Udacity for Organizations](#)
[Help and FAQ](#)
[Feedback Program](#)

COMMUNITY

[Blog](#)
[News & Media](#)
[Developer API](#)

UDACITY

[About](#)
[Jobs](#)
[Contact Us](#)
[Legal](#)

FOLLOW US ON