# Content

* Unique domain (URL) that is professional and ideally includes your name
* Contact Info: email, github, linkedin, twitter (optional)
  + If you decide to use a contact form, also include a direct link to your email
* Compelling About Me that shows passion for the web, wanting to be a programmer, *and* personality outside of coding.
* All 4 projects in the project section, should include:
* Title
* Short description
* Skills used (what is the stack?)
* Compelling screenshot (use a placeholder until your project’s design has been approved so that you’re choosing the best image)
  + Alternatively you could use an icon, photo, or illustration that represents your project
* Link to GitHub repo
* Link to live, working app
* GitHub README
  + The name of your app at the top of the file
  + A link to the live project
  + Documentation of your API.
  + Screenshot(s) of your app. This makes your app description much easier to understand.
  + A summary section. This should have a concise explanation of what your app does. Try to frame this from the standpoint of what the user does, or what the app enables for the user.
  + A section on the technology used.
  + Student examples: <https://github.com/jon-wade/pw-vault>, <https://github.com/jon-wade/incite-news>, <https://github.com/bvellek/aeropress-me-app#aeropressme-%EF%B8%8F>

# Design

Portfolios are reviewed on the following criteria:

* **mobile-first design**
* **semantic, valid markup**
* **built from scratch (do not use templates)**
* responsiveness from 320px to 1600px width screens
* attention to typographical detail and complimentary color scheme
* content clarity and readability
* contact functionality is required
* **cross-browser functionality (Chrome, Firefox, Safari)**

Projects are reviewed on the following criteria:

* **mobile-first design**
* **semantic, valid markup**
* responsiveness from 320px to 1600px width screens
* attention to typographical detail and complimentary color scheme
* content clarity and readability
* projects can’t be cloned from the curriculum or appear to be tutorial driven
  + No standard todo lists, calendars, calculators, etc
* apps must have strong concepts and useful or interesting functionality
* apps should have a landing page explaining what they do and who they’re for
* if apps require downloading or login, a demo version must be available

# Code Quality

* Resources:
  + [Starting With Code Reviews Lesson](https://courses.thinkful.com/fewd-201v2/project/1.1.2)
  + <https://github.com/airbnb/javascript>
  + <https://github.com/airbnb/javascript/tree/es5-deprecated/es5>
* General
  + Does the code work? Does it perform its intended function, the logic is correct etc.
  + Consistency through the file (attention to detail here)
    - Indentation (tabs vs spaces)
    - Font URLs (double vs single quotes)
    - Semicolons
  + Are there tests throughout the code?
  + Is all the code easily understood? Is documentation needed?
  + Do variable names describe what is stored in them?
  + Does it conform to consistent code styling? (indentation, quotes, etc)
  + Is it DRY? Is there any redundant or duplicate code?
  + Can any global variables be replaced?
  + Do loops have a set length and correct termination conditions? (No infinite loops)
  + Can any logging or debugging code be removed?
  + Do comments exist and describe the intent of the code?
  + Are all functions commented?
  + Is any unusual behavior or edge-case handling described? (is the author handling unexpected user behaviour)
  + Is there any incomplete code? If so, should it be removed or flagged with a suitable marker like ‘TODO’?
* HTML
  + Are you using semantic HTML and avoiding “div soup?”
  + Are you wrapping all input elements in form tags?
  + Are you including alt tags on images and adhering to other accessibility best practices?
    - <http://a11yproject.com/checklist.html>
* CSS
  + Class names should be descriptive
  + Never target elements with ID selectors
  + Asset files should have clear names
  + Font-family should have fallback options via font stacks
  + Prefixes should be used appropriately
    - <https://github.com/postcss/autoprefixer>
  + Hierarchy of element selectors should be clear
* JS
  + Follow ES5/6 standards
    - <https://github.com/airbnb/javascript>
    - <https://github.com/airbnb/javascript/tree/es5-deprecated/es5>
  + Caching DOM elements in variables
  + Avoid overly nested if statements
  + Node: structuring the project in an clearly organized manner, consistent amongst all projects
  + React: use classes appropriately. (Sometimes you don’t need a react component, just use a plain vanilla function)

# Resources

Another good overview of what your portfolio should have:

<https://gist.github.com/tholex/67eb8f86435ae77f3ad5>

Thinkful bootcamp student portfolios:

* <http://www.robbykim.com> (Full Time)
* <http://www.jbenjaminy.com/> (Full Time)
* <http://www.sierragregg.com> (Full Time)
* <http://surbhiposwalia.com/> (Full Time)
* <http://www.seanjun.com> (Full Time)
* <https://paulchong.net/> (Full Time)
* <http://www.simonattfield.com/> (Full Time)
* <http://shaykennedy.io/> (Full Time)
* <http://www.timpaik.co/>
* <http://jonwade.codes/>
* <http://equimper.github.io/>
* <http://www.cuttybang.audio/>
* <http://crtnywllr.github.io/>
* <http://www.melaniebrgr.com/>
* <http://www.linanguyenportfolio.com/>
* <https://keithunderwood.com/>
* <https://noux-nova.github.io/portfolio/>
* <http://rakeshrpatel.com/>
* <http://jasondobrowner.com/>

UX:

* <http://uxmyths.com/>
* <https://signalvnoise.com/posts/1926-a-shorthand-for-designing-ui-flows>
* <https://designshack.net/category/articles/critique-articles/>
* <https://www.useronboard.com/onboarding-teardowns/>
* <https://www.smashingmagazine.com/2014/12/design-principles-visual-weight-direction/>

HTML/CSS fundamentals and responsive design:

* <https://codepen.io/mi-lee/post/an-overview-of-html5-semantics>
* <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>
* [http://tympanus.net/codrops/css\_reference](http://tympanus.net/codrops/css_reference/)
* <https://code.tutsplus.com/tutorials/the-30-css-selectors-you-must-memorize--net-16048>
* <https://css-tricks.com/centering-in-the-unknown/>
* <https://css-tricks.com/absolute-positioning-inside-relative-positioning/>
* <https://css-tricks.com/dont-overthink-it-grids/>
* <http://bradfrost.com/blog/post/html-wireframes/>
* <http://bradfrost.com/blog/post/7-habits-of-highly-effective-media-queries/>
* <https://bradfrost.github.io/this-is-responsive/patterns.html>
* <http://bradfrost.com/demo/ish/>

Typography:

* <http://fontpair.co/>
* <https://femmebot.github.io/google-type/>
* <http://www.typogui.de/>
* <https://prowebtype.com>
* [https://www.typewolf.com](https://www.typewolf.com/)
* <http://practicaltypography.com/typography-in-ten-minutes.html>

Color, Gestalt, and Imagery:

* <https://coolors.co/app>
* <https://css-tricks.com/nerds-guide-color-web/>
* <https://material.google.com/style/color.html>
* <https://material.google.com/style/imagery.html#imagery-principles>
* <https://www.smashingmagazine.com/2014/09/design-principles-connecting-and-separating-elements-through-contrast-and-similarity/>
* <https://color.adobe.com/create/color-wheel/>
* <http://javier.xyz/cohesive-colors/>
* <https://toolness.github.io/accessible-color-matrix/>
* <http://webaim.org/resources/contrastchecker/>
* <http://www.colourco.de/>

Inspiration:

* <http://uxarchive.com/>
* <http://land-book.com/>
* <http://niceportfol.io/>
* <http://hoverstat.es/>
* <https://templated.co/>
* <https://www.designernews.co/badges/design>