A **designer’s online portfolio** is essential to transitioning to a new career. A designer's online portfolio allows you to communicate who you are, your design process, and, finally, your work! A portfolio is also a place to practice iterating and applying your growing UX, UI, and front-end web design skills.

This homework will be the starting place for the rest of the coding section’s homework (18 - 21), so do your best! Success during this assignment will set you up to create a beautiful design as we continue to iterate on our Portfolio site.

**Optional Portfolio Strategy:** Use a Content Management System (CMS) Platform.

During these weeks, you also have the option to start creating your portfolio using a content management system or website service. Some recommended CMS portfolio platforms:

1. Uxfol.io - <https://uxfol.io/>
2. WIX - <https://www.wix.com/>
3. Squarespace - <https://www.squarespace.com/>
4. Cargo Collective - <https://cargo.site/>
5. Semplice Wordpress - <https://www.semplice.com/>
6. Webflow - <https://webflow.com/>

*Note: Using an optional portfolio strategy does not exempt you from the coding skills homework assignments.*

**Scenario**: Last week, you started the process of creating the homepage of your UX design portfolio during class with your skills assessment and about me wireframe. This week, you will begin to build out this wireframe which will become your UX/UI portfolio website over the next 8 weeks. The purpose of the site is to showcase the case studies and skills you compiled during Unit 16. Get excited! This is the final step of the UX/UI Boot Camp that ends in a new career for you.

*Be sure to read through this document in its entirety.*

## **Week 17: One-Page Website Wireframe**

**User Story:** As a UX/UI Designer/FE Developer, you will want to design and code a one-page web page for hiring managers that allows them to review your work and learn more about you. Your journey begins today with the first coding assignment.

## **Requirements (8-11 hours to complete)**

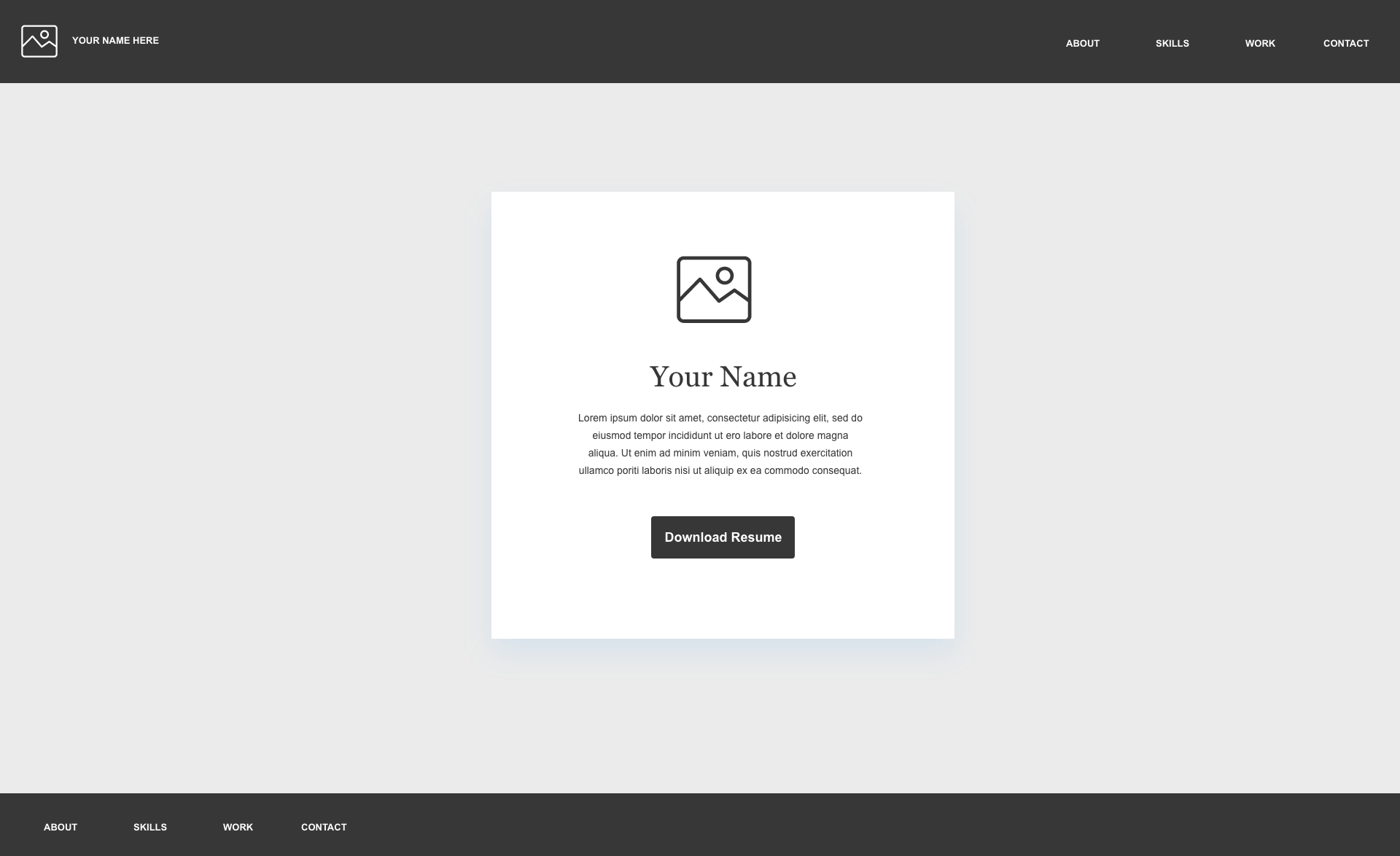
Each of the parts below will guide you through designing and coding the start of your about me wireframe, with suggestions on how you should build and style the structure. (Note the time estimate for completing each step.) There will also be documentation explaining the different HTML elements & CSS properties used should you need help.

* **PART ONE: Redline Your Portfolio Wireframe You Created In Week 16 using Adobe XD/Figma (1 hour)**
* **PART TWO: Create and clone a GitHub repository to your desktop (1 hour)**
* **PART THREE: Export assets from your Adobe XD/Figma portfolio wireframe (1 hour)**
* **PART FOUR: Create basic HTML structure for a Navbar, Hero Image and Footer (2 hours)**
* **PART FIVE: Attach a stylesheet and style the HTML of your Navbar, Hero Image and Footer sections (3 hours)**
* **PART SIX: Upload your coded Website’s index.html, style.css and images folder to GitHub and publish via GitHub Pages (2 hours)**

## **Instructions**

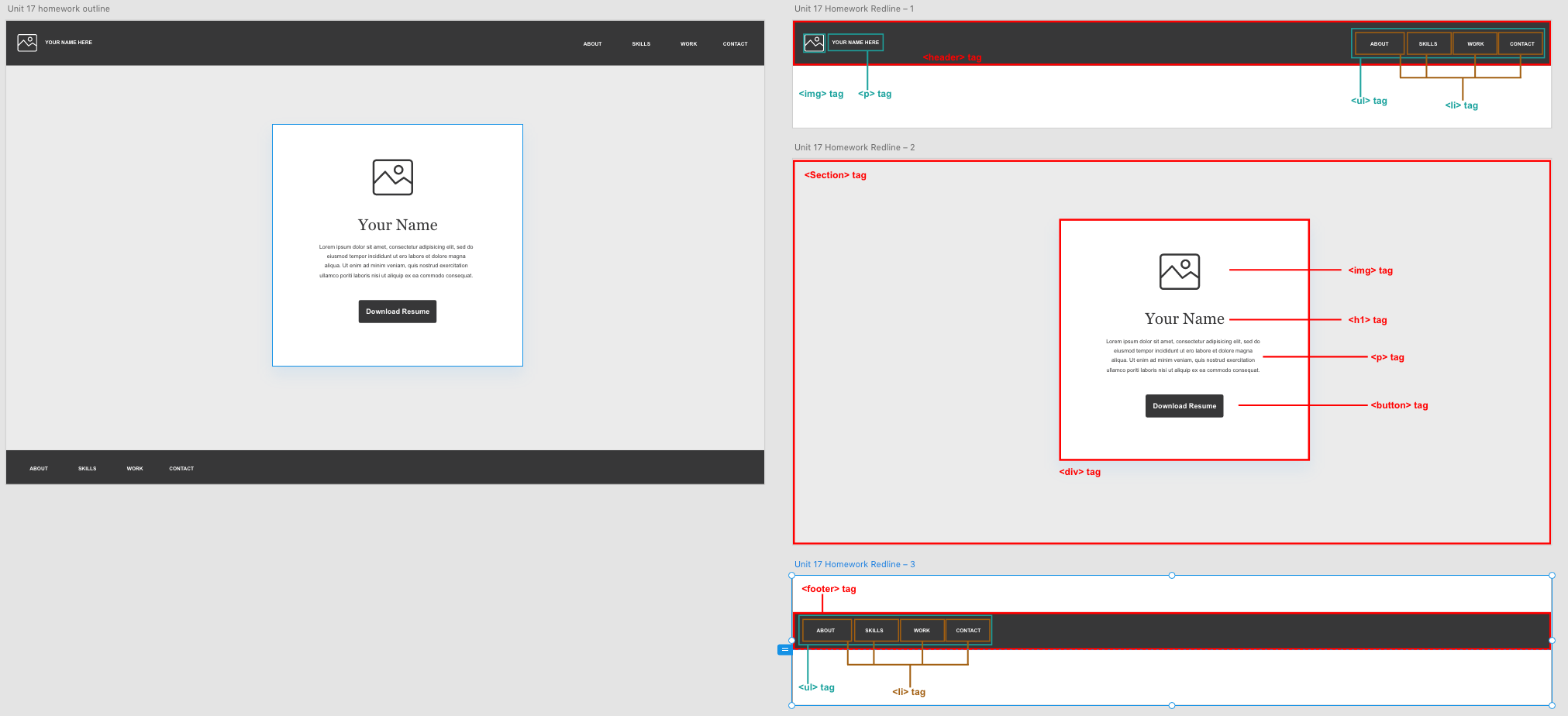
**PART ONE: Redline Your About Me Wireframe You Created In Week 16 using Adobe XD/Figma (1 hour)**

As a front-end developer, this first thing we do is strategically plan out how we want to code a page. We do that by redlining a wireframe or mockup given to us by the design team.

Wireframe Sample:   


1. Before you begin coding, you should mentally map out what html tags you are going to need to use. Open Adobe XD and open your about me wireframe page.
2. Create three new artboards and redline the following sections.
   1. Your header section.
   2. Your Hero Image Section.
   3. Your Footer Section.

Redline Sample:

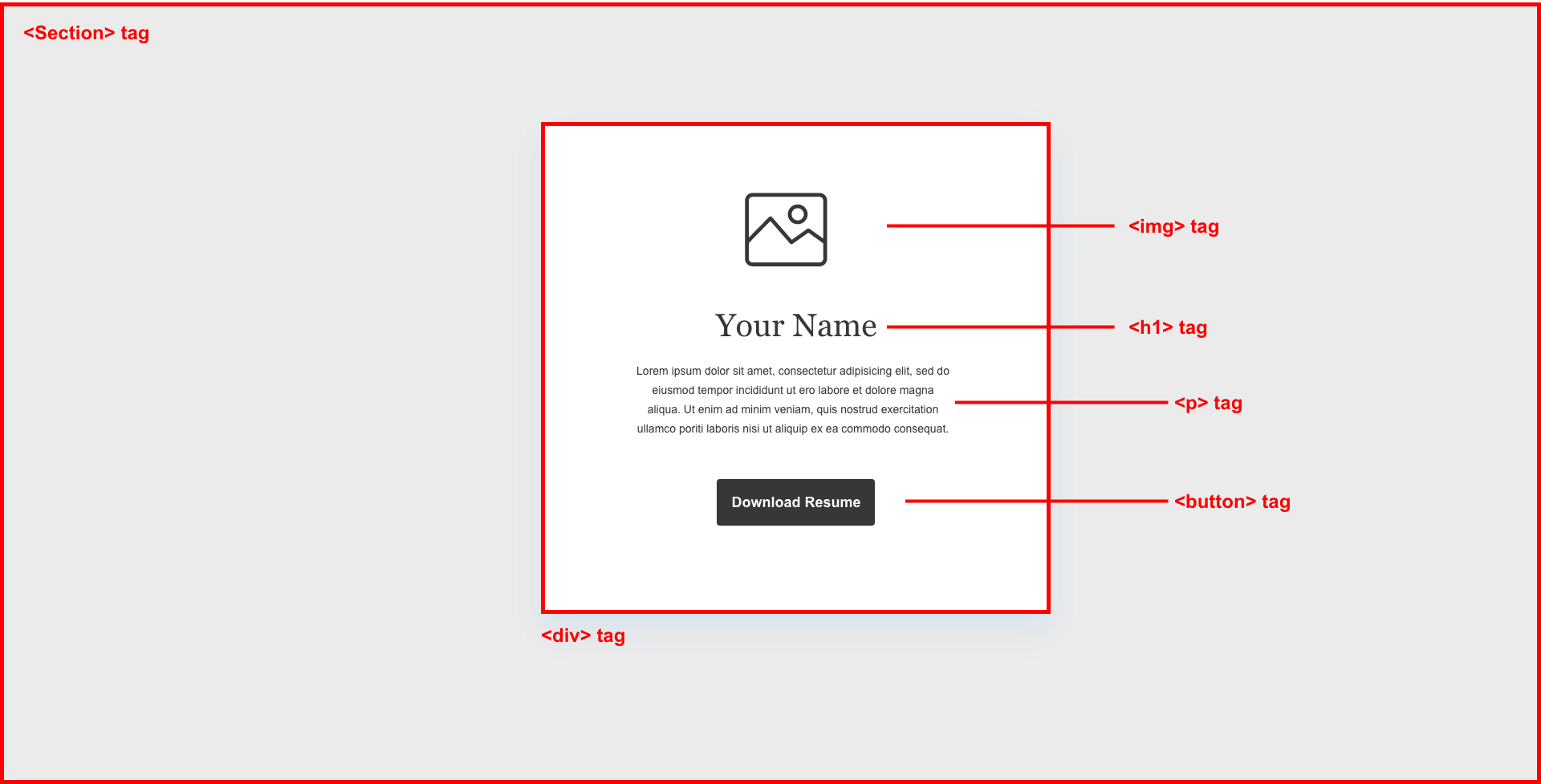


1. Redline your header section. Your header should contain the following tags.
   1. A [<header>](https://www.w3schools.com/tags/tag_header.asp) tag.
   2. An [<img>](https://www.w3schools.com/tags/tag_img.asp) tag.
   3. A [<p>](https://www.w3schools.com/tags/tag_p.asp) tag.
   4. A [<ul>](https://www.w3schools.com/tags/tag_ul.asp) tag
   5. Four [<li>](https://www.w3schools.com/tags/tag_li.asp) tags containing the text:
      1. About.
      2. Skills.
      3. Work.
      4. Contact.

Header Sample:



1. Readline your hero image section. Your hero image should contain the following tags.
   1. A [<section>](https://www.w3schools.com/tags/tag_section.asp) tag
   2. A [<div>](https://www.w3schools.com/tags/tag_div.asp) tag that is wrapped by the section tag.
   3. An [<img>](https://www.w3schools.com/tags/tag_img.asp) tag wrapped inside the div.
   4. An [<h1>](https://www.w3schools.com/tags/tag_hn.asp) tag wrapped inside the div.
   5. A [<p>](https://www.w3schools.com/tags/tag_p.asp) tag wrapped inside the div.
   6. A [<button>](https://www.w3schools.com/tags/tag_button.asp) tag wrapped inside the div.

Hero Image Section Sample:

1. Redline your footer section. Your footer section should contain the following tags.
   1. A [<footer>](https://www.w3schools.com/tags/tag_footer.asp) tag.
   2. A [<ul>](https://www.w3schools.com/tags/tag_ul.asp) tag.
   3. Four [<li>](https://www.w3schools.com/tags/tag_li.asp) tags that contain the following text.
      1. About.
      2. Skills.
      3. Work.
      4. Contact.

Footer Section Sample:

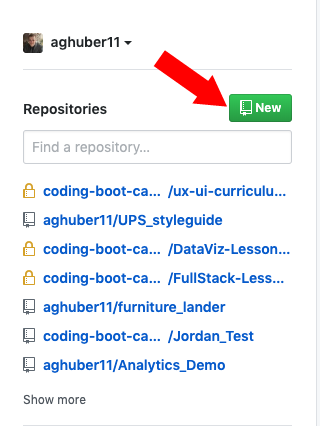


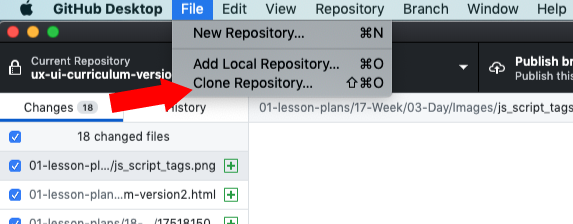
1. Export your artboards to jpeg or png files using Adobe XD’s export tool.
2. Drop these images in a Google drive folder that will be turned in at <https://www.bootcampspot.com>

**PART TWO: Create and clone a GitHub repository to your desktop (1 hour)**

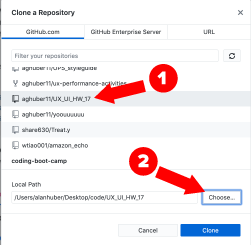
As a front-end developer, we start creating a work space for coding by creating a repository.

Create a GitHub project repo by completing the following:

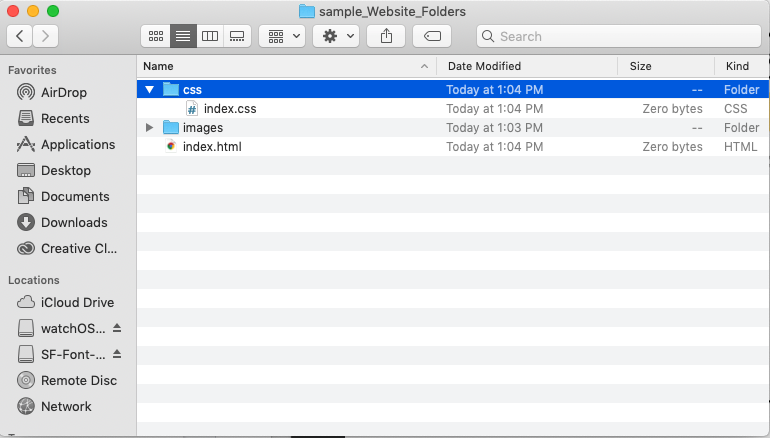
1. Navigate and login to [www.GitHub.com](https://www.github.com/login) and click new in the top left of github.com.  
   
2. Create a new GitHub repo in GitHub named UX\_UI\_HW\_17.
3. Open GitHub Desktop.
4. Clone the empty repo to your local computer in the code folder you created last week in 16.3 by clicking file -> clone.

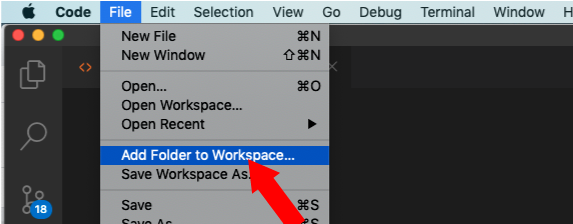


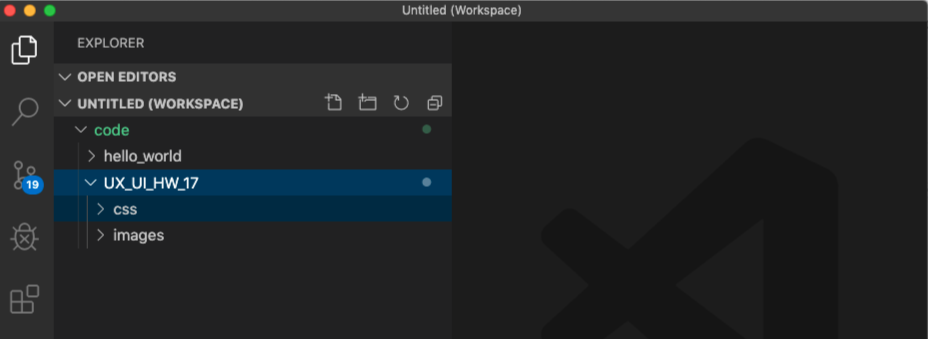
* + Select the repository named UX\_UI\_HW\_17
  + Make sure you choose the code folder in the local path so you can keep track of your work



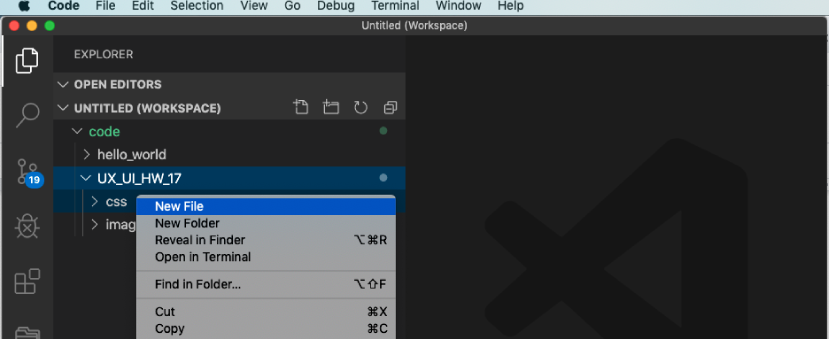
1. Create the proper folder structure inside UX\_UI\_HW\_17 to hold your website assets. Review the folder structure in the following screenshot for a good example.
   * Create two folders inside our UX\_UI\_HW\_17 folder that you just cloned to the code folder on your desktop. Name the first folder css. Name the second folder images.



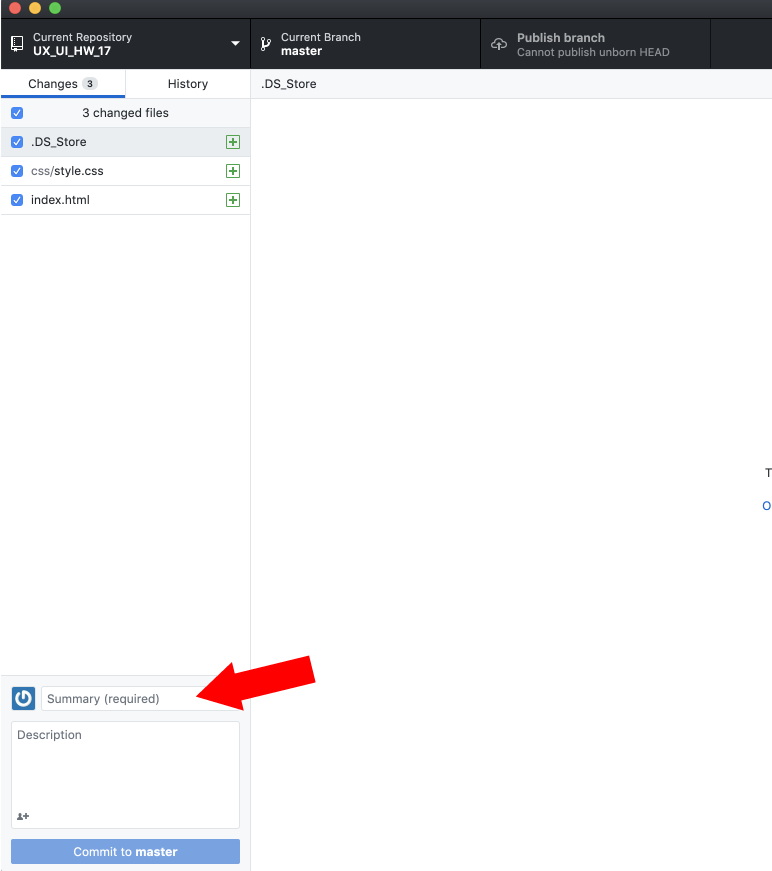
1. Open VSCode.
2. Add your code folder to a visual studio workspace. Click file -> add folder to workspace.  
   
   * Your window will now look like this:



1. Create an index.html file in your root folder and an index.css file located in your CSS folder (illustrated below) using VSCode.
   * Right click on your UX\_UI\_HW\_17 and click new file. Name it index.html
   * RIght click on your CSS folder and right click -> select new file. Name in style.css  
     *If you are a beginner at using a text editor, please review the following* [*intro videos*](https://code.visualstudio.com/docs/getstarted/introvideos) *about VSCode.*

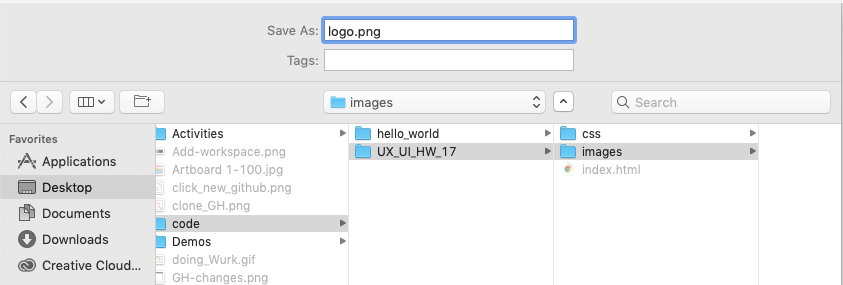
**

1. Open GitHub Desktop. If you need a refresher on how to use GitHub Desktop, view the following [documentation](https://help.github.com/en/desktop/getting-started-with-github-desktop).
2. GitHub Desktop can see that you have added files and is tracking your repo. Make sure you write a summary message and click commit to master.



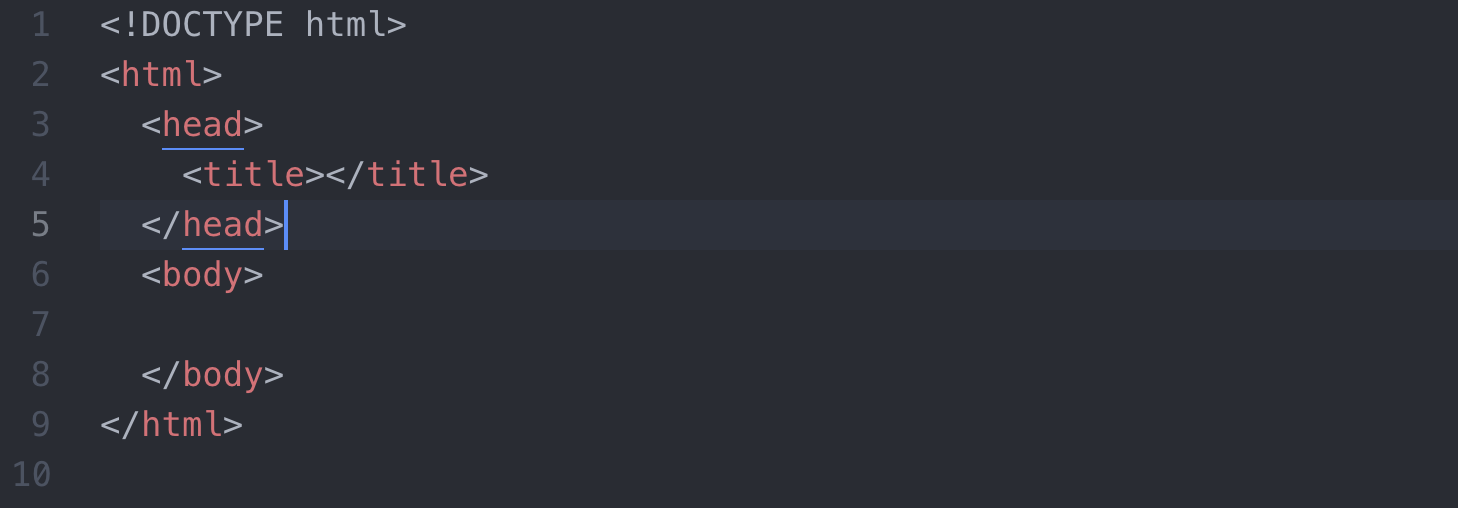
1. Push these pages up to your UX\_UI\_HW\_17 GitHub repo to sync your local repo with your cloud repo.

**PART THREE: Export your image assets from your wireframe to your images folder (1 hour)**

1. Open the design file you created last week during unit 16 where you designed a wireframe of an about me page using Adobe XD.
2. Export the following assets using Adobe XD into your images folder.   
   *If you need a refresher on how to export images in adobe XD, please view the following* [*tutorial*](https://helpx.adobe.com/xd/help/export-design-assets.html) *from Adobe XD documenting the process.*
   1. An image of yourself.
   2. Your logo.
3. Make sure you save the images in the images folder you created in the previous step.  
   

**PART FOUR: Code basic HTML structure for a Navbar, Hero Image and Footer (2 hours)**

To practice being a front-end developer, let’s start coding in our GitHub repository using VScode.

1. Open the index.html file you created in VScode.
2. Create the basic HTML structure for our HTML document.  
   

**Code the Navbar HTML:**

1. Inside the [body tag](https://www.w3schools.com/tags/tag_body.asp), create the following div tags to hold our content.
   1. A [<header> tag](https://www.w3schools.com/tags/tag_header.asp) with the class of navigation. Create the following HTML structure inside of it:
      1. A [div tag](https://www.w3schools.com/tags/tag_div.asp) with the class of inline-block and logo. Wrapped inside this div tag, create a:
         1. [Img tag](https://www.w3schools.com/tags/tag_img.asp) with a logo. Use the image of your logo you exported during the second step of this homework.
      2. A [nav tag](https://www.w3schools.com/tags/tag_nav.asp) with the class of [inline-block](https://www.w3schools.com/css/css_inline-block.asp). Wrapped inside of the nav tag create:
         1. An [unordered list tag](https://www.w3schools.com/html/html_lists.asp) that wraps four [li tags](https://www.w3schools.com/tags/tag_li.asp).
            1. Give the li tags the class of inline-block.

Inside your first li tag, add the text “about”.

Inside your second li tag, add the text “skills”.

Inside your third li tag, add the text “work”.

Inside your fourth li tag, add the text “contact”.

**Code the Hero Image HTML:**

1. Under your header, create a section tag with the class of “heroImg”.
2. Wrapped inside the hero image section, create:
   1. A div with the class of content. Inside this div, create the following HTML structure:
      1. An [<img> tag](https://www.w3schools.com/tags/tag_img.asp) with a picture of yourself that you exported in the second step of this homework.
      2. A [<p> tag](https://www.w3schools.com/tags/tag_p.asp) that contains a paragraph about yourself!
      3. A [<button>](https://www.w3schools.com/tags/tag_button.asp) tag with the words “Download Resume”

**Code the Footer HTML:**

1. Under your section with the class of “heroImg,” create a [<footer>](https://www.w3schools.com/tags/tag_footer.asp) tag.
2. Wrapped inside the footer section create:
   1. An [<ul> tag](https://www.w3schools.com/html/html_lists.asp) that wraps four [<li> tags](https://www.w3schools.com/tags/tag_li.asp).
      1. Give the li tags the class of inline-block.
         1. Inside your first li tag, add the text “about”.
         2. Inside your second li tag, add the text “skills”.
         3. Inside your third li tag, add the text “work”.
         4. Inside your fourth li tag, add the text “contact”.

**PART FIVE: Attach a CSS stylesheet and add CSS style to the HTML of your Navbar, Hero Image, and Footer sections (3 hours)**

As a front-end developer, now that you have some HTML structure, you can select and style your elements via a CSS stylesheet.

1. In VSCode, open your style.css file located in your css folder.
2. Create a [<link> tag](https://www.w3schools.com/tags/tag_link.asp) to connect your HTML file to your CSS file.

**Code Navbar CSS:**

1. To build out our navbar, first we need all the elements to line up next to each other. In our style.css file, create a [selector](https://www.w3schools.com/css/css_selectors.asp) that targets our elements with the class of inline-block.
   1. Give this selector the CSS property of [display: inline-block;](https://www.w3schools.com/css/css_inline-block.asp) This will cause our navbar to line up next to each other.
   2. Give this selector the property of [vertical-align: top;](https://www.w3schools.com/cssref/pr_pos_vertical-align.asp) to ensure all elements align next to one another.
2. Next we need our divs to fill the width of our container.
   1. Create a [CSS selector](https://www.w3schools.com/css/css_selectors.asp) that targets our div with the [class](https://www.w3schools.com/html/html_classes.asp) of logo.
      1. Set this element’s [width](https://www.w3schools.com/cssref/pr_dim_width.asp) to 23%.
      2. Give this element a [padding](https://www.w3schools.com/css/css_padding.asp) of 1%.
   2. Create a CSS selector that targets our HTML element nav. If you don’t remember how to target base html elements, read this [documentation](https://www.w3schools.com/cssref/css_selectors.asp)
      1. Set this element’s [width](https://www.w3schools.com/cssref/pr_dim_width.asp) to 73%.
      2. Give this element a [padding](https://www.w3schools.com/css/css_padding.asp) of 1%.
3. Let’s align our unordered list in its container to the right.
   1. Target the base HTML ul element we created earlier.
      1. Give it a [width](https://www.w3schools.com/cssref/pr_dim_width.asp) of 50%.
      2. Center it using [margin-left: auto;](https://www.w3schools.com/css/css_margin.asp).
4. Let’s center our images inside their containers also. In our css selector that targets the class of logo:
   1. Give this element the CSS property [text-align: center;](https://www.w3schools.com/cssref/pr_text_text-align.ASP)

**Code Hero Image CSS:**

1. To begin, give our section with the class of heroImg a [height](https://www.w3schools.com/cssref/pr_dim_height.asp).
   1. Give this element the CSS property of [height](https://www.w3schools.com/cssref/pr_dim_height.asp): 600px;
   2. Give this element a [padding](https://www.w3schools.com/css/css_padding.asp) of 25px;
2. Create a CSS selector that targets the class of content.
   1. Give this element a [width](https://www.w3schools.com/cssref/pr_dim_width.asp) of 500px;
   2. Center this element using [margin: 0 auto;](https://www.w3schools.com/css/css_margin.asp)

**Code Footer CSS:**

1. To build out your footer, write a [CSS selector](https://www.w3schools.com/css/css_selectors.asp) that targets the base html footer element.
   1. Give this selector a [background color](https://www.w3schools.com/cssref/pr_background-color.asp) of #373738
   2. Use padding to center your [<ul>](https://www.w3schools.com/tags/tag_ul.asp) and [<li>](https://www.w3schools.com/tags/tag_li.asp) elements
2. Write a [CSS selector](https://www.w3schools.com/css/css_selectors.asp) that targets your [<ul>](https://www.w3schools.com/tags/tag_ul.asp) nested inside of our footer element.
   1. Set this element to display as an [inline-block](https://www.w3schools.com/css/css_inline-block.asp) level element.
   2. Give this selector the property of [vertical-align: top;](https://www.w3schools.com/cssref/pr_pos_vertical-align.asp) to ensure all elements align next to one another.
   3. Use padding to center the text inside of our [<li>](https://www.w3schools.com/tags/tag_li.asp) elements.
3. That’s it! We now have a basic wireframe of our portfolio site for you to practice your CSS skills on.

**Code BONUS:**

1. Now it’s up to you to create and practice styling this page. Do your best to make this template look like the design you created during week 16.   
     
   When you are finished styling, make sure you reupload your modified CSS files to GitHub so the world can see your work!  
     
   Here are a couple links to some CSS properties that you should experiment with.  
   <https://www.w3schools.com/cssref/pr_background-image.asp>  
   <https://www.w3schools.com/cssref/css3_pr_box-shadow.asp>  
   <https://www.w3schools.com/cssref/pr_font_font-family.asp>  
   <https://www.w3schools.com/html/html_colors.asp>

**Strategies for applying visual design to your portfolio:**

* Apply [background-colors](https://www.w3schools.com/cssref/pr_background-color.asp) to your navbar and the associated unordered lists and list items contained inside.
* Add a background image to your hero image. Make sure to use [background-size: cover](https://www.w3schools.com/cssref/css3_pr_background-size.asp) and [background-repeat: no-repeat](https://www.w3schools.com/cssref/pr_background-repeat.asp) to make your background span the full width of your page.
* Apply a [drop shadow](https://www.w3schools.com/cssref/css3_pr_box-shadow.asp) to the div contained inside your hero image.
* Add a [font-family](https://www.w3schools.com/cssref/pr_font_font-family.asp) to the text elements in your LI and p tags.

**PART SIX: Upload your Website to GitHub and publish via GitHub Pages (2 hours)**

As a front-end developer, getting practice publishing your work and hosting it online will help you submit your assignment but also get you into the habit of getting feedback and practicing GitHub desktop.

1. Upload your finished webpage to GitHub using GitHub Desktop.
2. Publish your site on GitHub Pages (if you haven’t already).
3. Make sure you copy the URL. You will need it to submit your homework via Bootcamp Spot.

## **Tips**

The following tips are general developer workflow tips to help speed up your development process when building a webpage:

* Start by coding the skeleton of the wireframe first. Use outline: **1px solid red;** to see what you're doing on your page with your CSS. Remember to use the web inspector to preview changes to your CSS.
* Style your content inside your wireframe.
* Google a lot! There are many nuances to front-end development and you will undoubtedly encounter problems when practicing coding. Using Google to search for answers is a normal part of all web developers’ workflow.
* Create an account at <https://stackoverflow.com/>. Stack Overflow is a community forum for developers. It is a great resource to help answer all your developer questions.

### **How to Submit Your Homework**

Use Google Drive (G-drive) and Github to turn in your homework assignment (make sure to double-check that your sharing permissions are set to “Anyone with the link can view”).

Submit the following links to <https://www.bootcampspot.com>:

* A Link to a G-drive folder containing images of your portfolio wireframe, and the redline wireframe mapping out the HTML elements:
  + The original design of your portfolio wireframe.
  + The redlined header, hero image and footer.
* A link to your hosted GitHub Pages.
* A link to your GitHub repo.