{ } FE Dev Homework

Unit 17 | Code Your Portfolio Homepage (HTML/CSS)

🕒 ~11 Hours | 👤 Individual | ✎ 6 Steps

(Assignment Due 18.3)

Background

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

An **online portfolio** is essential to transitioning to a new career in design. Your online portfolio communicates who you are, your design process, and your work! A portfolio is also a place to practice iterating and applying your growing UX, UI, and front-end web design skills.

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 follow:

- UXfol.io: https://uxfol.io/
- Wix: https://www.wix.com/
- Squarespace: https://www.squarespace.com/
- Cargo Collective: https://cargo.site/
- Semplice WordPress: https://www.semplice.com/
- 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 creating your UX design portfolio's homepage 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 eight 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.

User story: As a UX/UI designer/front-end developer, you will want to design and code a one-page webpage for hiring managers to review your work and learn more about you. Your journey begins today with the first coding assignment.

Objectives

Here's what instructors want to see that you're capable of:

- ❏ Planning an HTML/CSS layout using annotations from Figma or Adobe XD.
- ❏ Exporting webpage design assets from Figma or Adobe XD.
- ❏ Creating and clone a Github Repository
- ❏ Coding HTML structure and CSS styling to the header > content > Footer of a webpage.
- ❏ Uploading webpage to GitHub pages and share hosted URL.

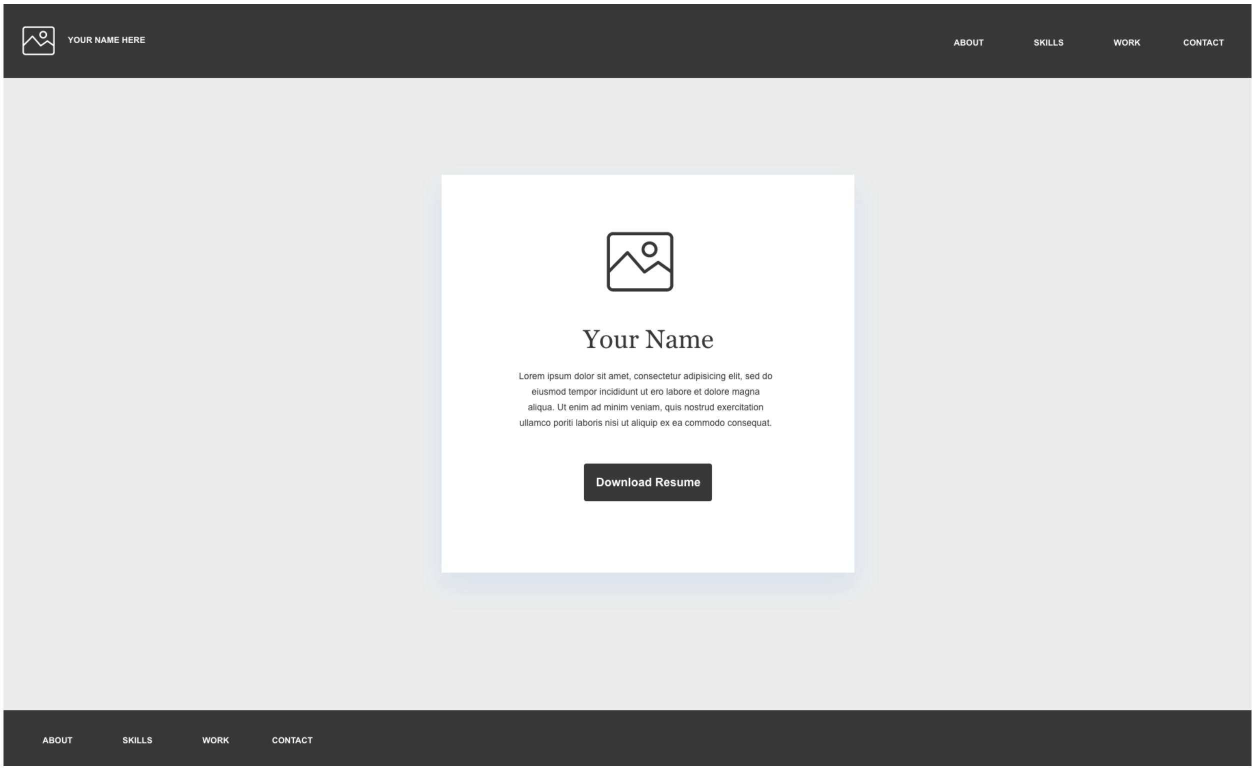
Instructions

Each of the steps 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 that the time estimate for completing each step.) There will also be documentation explaining the different HTML elements and CSS properties in use should you need help.

Step 1: Annotate the portfolio wireframe you created in Week 16 using Figma or Adobe XD (1-2 hours)

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

Wireframe sample:

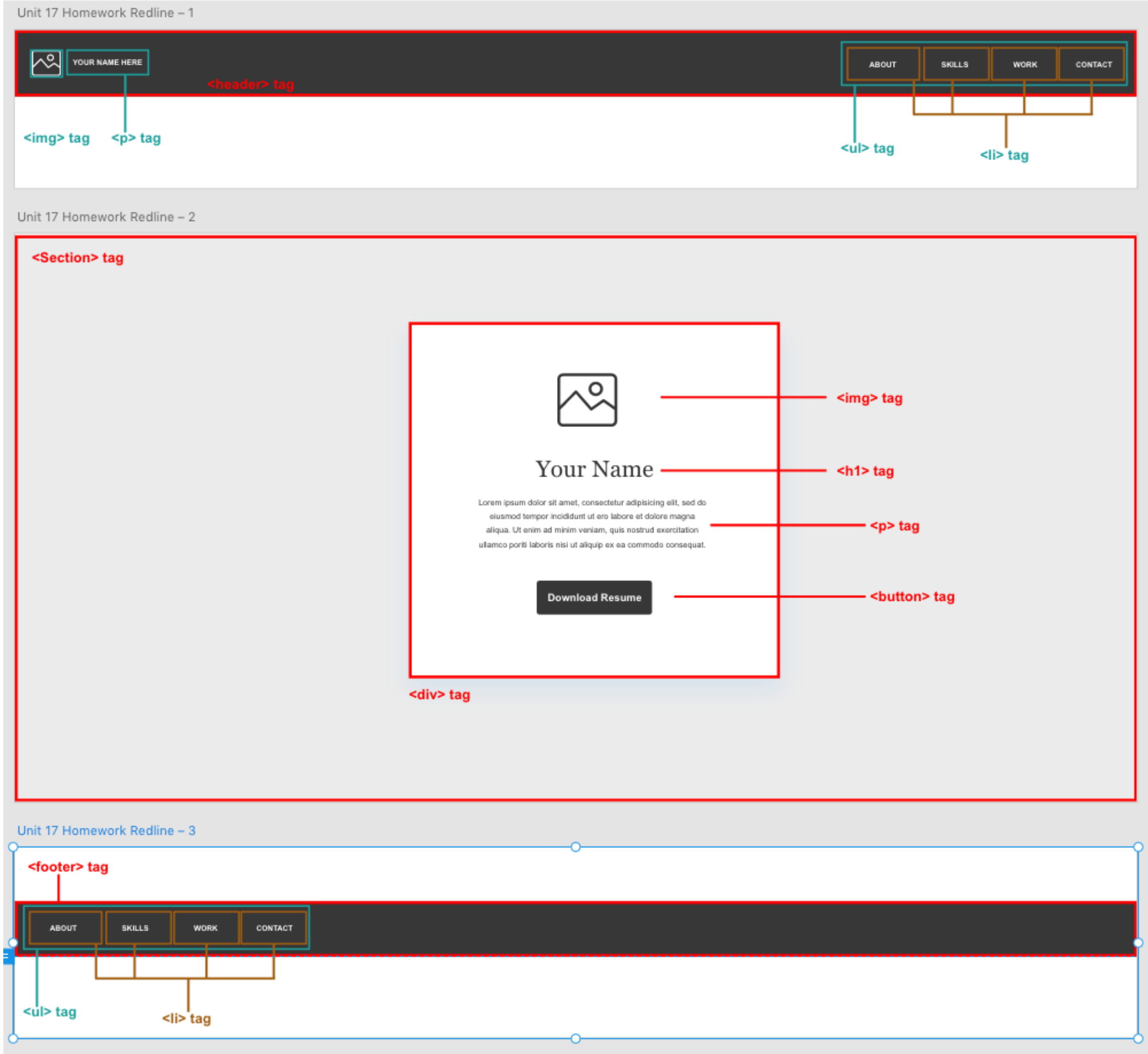


Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut ero labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco poriti laboris nisi ut aliquip ex ea commodo consequat.

1. Before you begin coding, you should mentally map out what HTML tags you need to use. Open Figma or Adobe XD and open your About Me wireframe page.

2. Create three new artboards and annotate the following sections.
    a. Your header section
    b. Your hero image section.
    c. Your footer section.

Annotation sample:

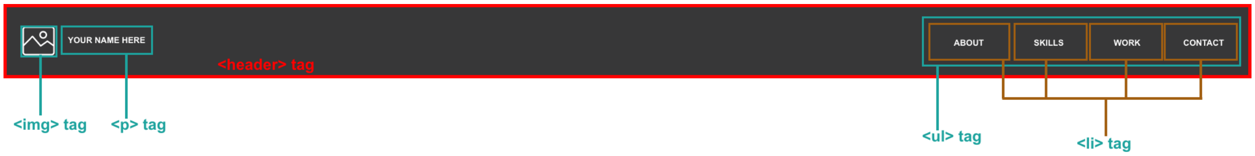


Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut ero labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco poriti laboris nisi ut aliquip ex ea commodo consequat.

3. Annotate your header section. Your header should contain the following tags:
    a. A <header> tag
    b. An <img> tag
    c. A <p> tag
    d. A <ul> tag
    e. Four <li> tags containing the text:
        i. About
        ii. Skills
        iii. Work

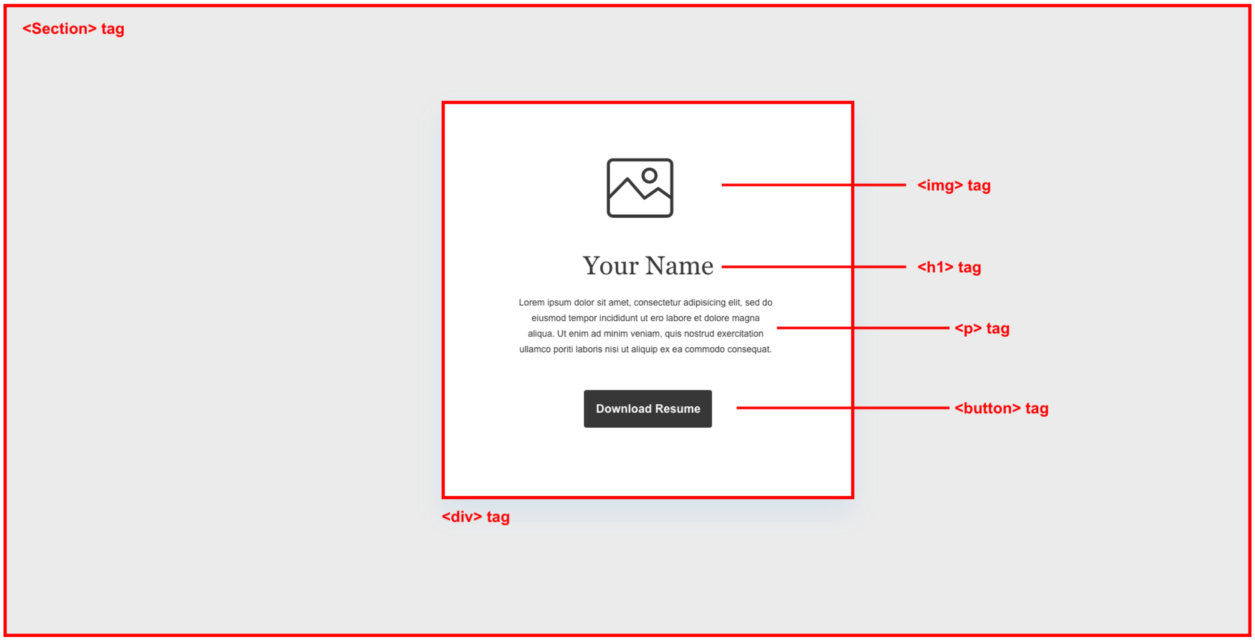
iv. Contact

Header sample:



4. Annotate your hero image section. Your hero image should contain the following tags:
    a. A <section> tag
    b. A <div> tag that is wrapped by the section tag
    c. An <img> tag wrapped inside the div
    d. An <h1> tag wrapped inside the div
    e. A <p> tag wrapped inside the div
    f. A <button> tag wrapped inside the div

Hero image section sample:

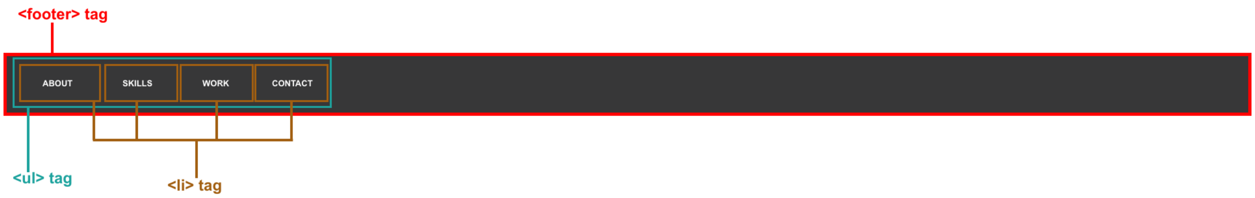


Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut ero labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco poriti laboris nisi ut aliquip ex ea commodo consequat.

5. Annotate your footer section. Your footer section should contain the following tags:
    a. A <footer> tag
    b. A <ul> tag
    c. Four <li> tags that contain the following text:
        i. About
        ii. Skills

iii. Work
iv. Contact

Footer section sample:



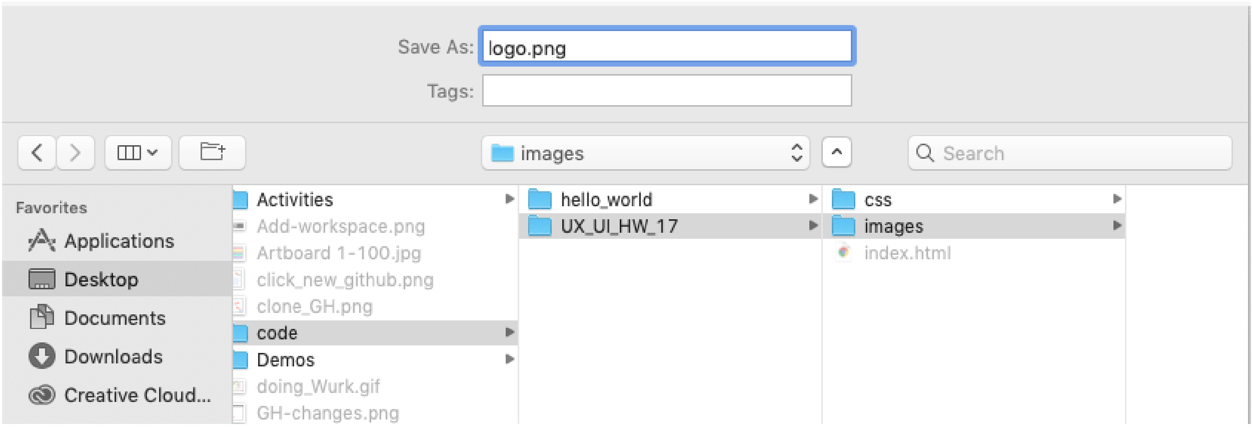
6. Export your artboards to jpeg or png files using the Adobe XD/Figma export tool.
7. Drop these images in a Google Drive folder that will be turned in at https://www.bootcampspot.com.

Step 2: Export assets from your Figma or Adobe XD portfolio wireframe (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 Figma/Adobe XD.

2. Export the following assets using Figma/Adobe XD into your images folder.
   If you need a refresher on how to export images in Adobe XD, please view the following tutorial from Adobe XD documenting the process.
   a. An image of yourself
   b. Your logo

3. Make sure you save the images in the images folder you created in the previous step.



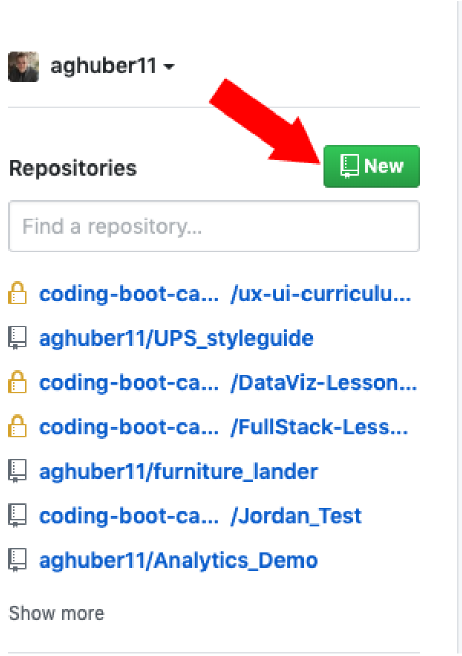
Save As: logo.png
Tags:

Step 3: Create and clone a GitHub repository to your desktop (1 hour)

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

Create a GitHub project repo by completing the following:

1. Navigate and login to www.GitHub.com and click new in the top left of github.com.

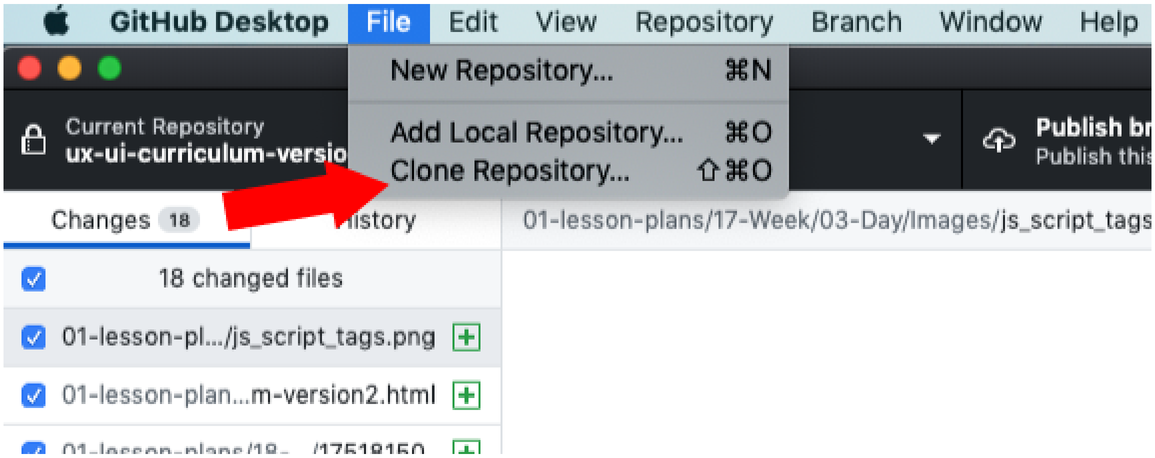


coding-boot-ca... /ux-ui-curriculu...
aghuber11/UPS_styleguide
coding-boot-ca... /DataViz-Lesson...
coding-boot-ca... /FullStack-Less...
aghuber11/furniture_lander
coding-boot-ca... /Jordan_Test
aghuber11/Analytics_Demo

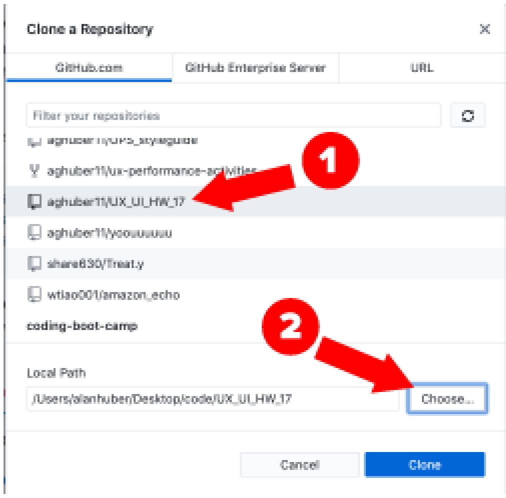
2. Create a new GitHub repo in GitHub named UX_UI_HW_17.

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.



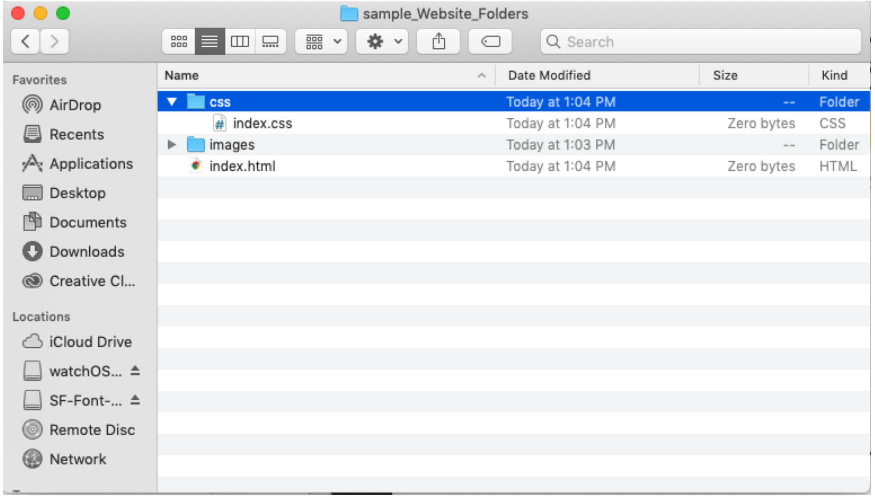
- Select the repository named UX_UI_HW_17.
- Make sure you choose the code folder in the local path to keep track of your work.



Local Path

/Users/alanhuber/Desktop/code/UX_UI_HW_17

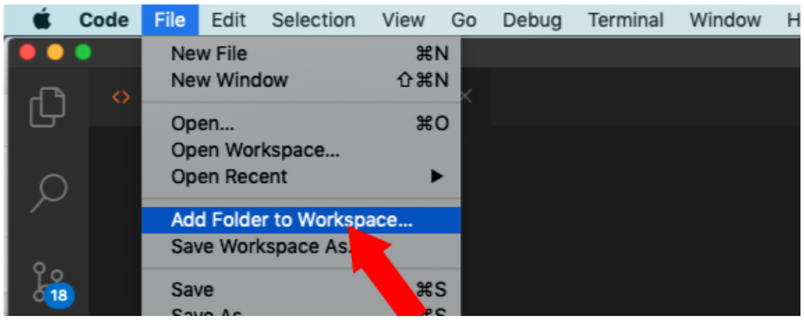
5. Create the proper folder structure inside UX_UI_HW_17 to hold your website assets. Review the folder structure in the following screenshot for a good example.
    - Create two folders inside our UX_UI_HW_17 folder that you just cloned to the code folder on your desktop. Name the first folder CSS. Name the second folder images.



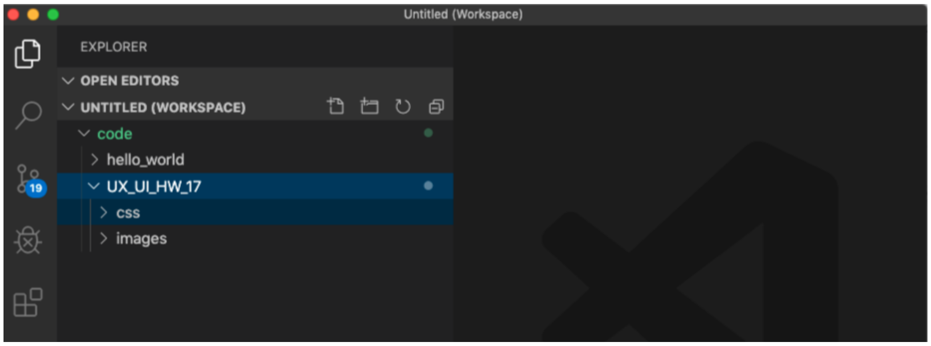
Name	Date Modified	Size	Kind
css	Today at 1:04 PM	--	Folder
index.css	Today at 1:04 PM	Zero bytes	CSS
images	Today at 1:03 PM	--	Folder
index.html	Today at 1:04 PM	Zero bytes	HTML

6. Open Visual Studio Code.

7. Add your code folder to a visual studio workspace. Click file -> add folder to workspace.

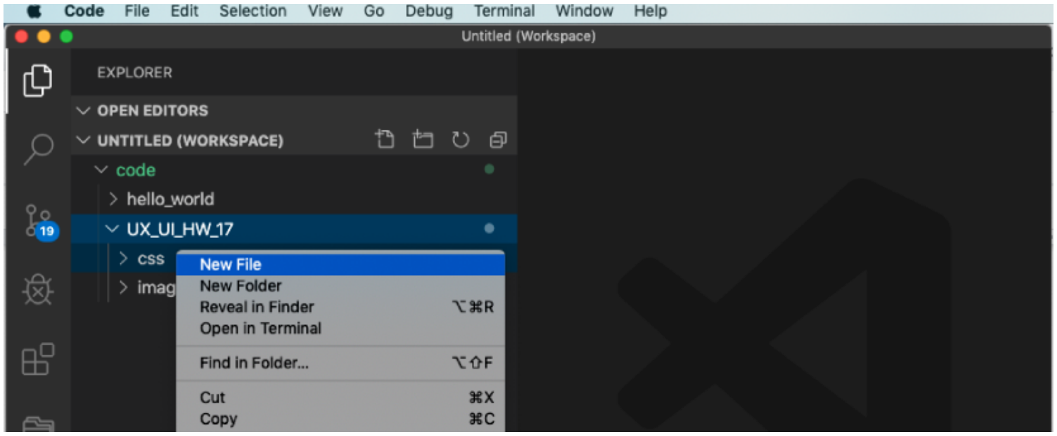


- Your window will now look like this:



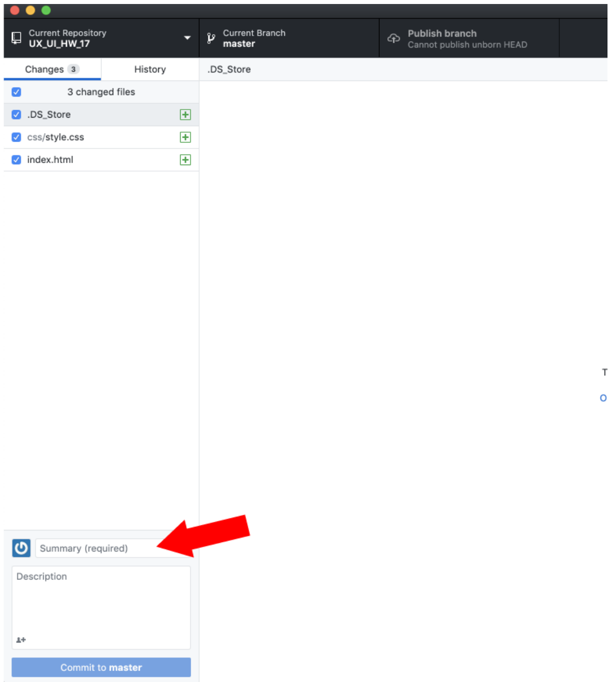
8. Create an index.html file in your root folder and an index.css file located in your CSS folder (illustrated below) using VSCode.

- 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 these intro videos about Visual Studio Code.*



9. Open GitHub Desktop. If you need a refresher on how to use GitHub Desktop, view the following documentation.

10. 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.

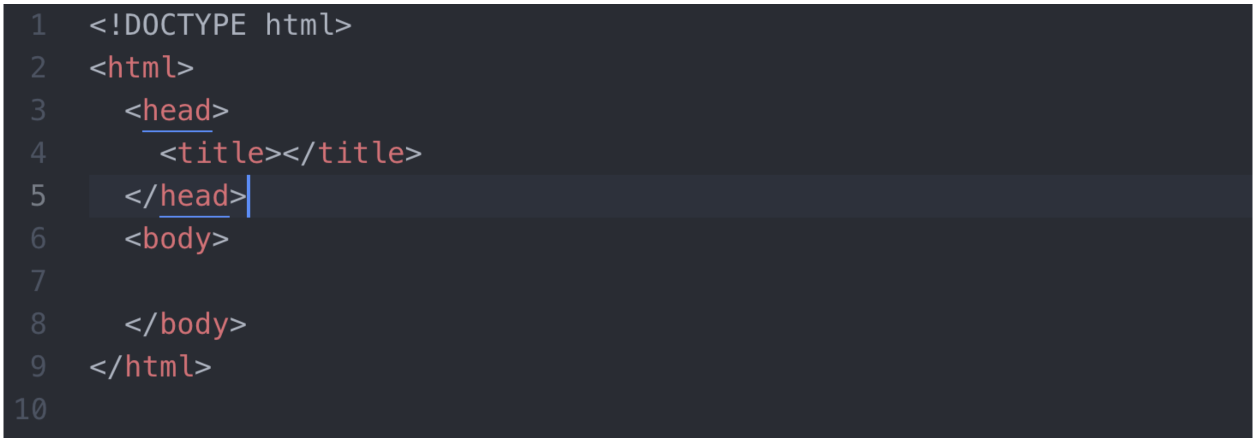


11. Push these pages up to your UX_UI_HW_17 GitHub repo to sync your local repo with your cloud repo.

Step 4: Create a basic HTML structure for a navbar, hero image, and footer (1-2 hours)

To practice being a front-end developer, let's start coding in our GitHub repository using Visual Studio Code.

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



```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>

  </body>
</html>

```

Code the navbar HTML:

1. Inside the body tag, create the following div tags to hold our content.
    a. A <header> tag with the class of navigation. Create the following HTML structure inside of it:
        i. A div tag with the class of inline-block and logo. Wrapped inside this div tag, create an:
            1. Img tag with a logo. Use the image of your logo you exported during the second step of this homework.
        ii. A nav tag with the class of inline-block. Wrapped inside of the nav tag create:
            1. An unordered list tag that wraps four li tags.
                a. Give the li tags the class of inline-block.
                    i. Inside your first li tag, add the text "about."
                    ii. Inside your second li tag, add the text "skills."
                    iii. Inside your third li tag, add the text "work."
                    iv. 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:
    a. A div with the class of content. Inside this div, create the following HTML structure:
        i. An <img> tag with a picture of yourself that you exported in the second step of this homework.
        ii. A <p> tag that contains a paragraph about yourself!
        iii. A <button> tag with the words “Download Resume.”

Code the footer HTML:

1. Under your section with the class of “heroImg,” create a <footer> tag.
2. Wrapped inside the footer section create:
    a. An <ul> tag that wraps four <li> tags.
        i. 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.”

Step 5: Attach a stylesheet and style the HTML of your navbar, hero image, and footer sections (2-3 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 Visual Studio Code, open your style.css file located in your CSS folder.
2. Create a <link> tag to connect your HTML file to your CSS file.

Code navbar CSS:

1. First, we need all the elements to line up next to each other to build out our navbar. In our style.css file, create a selector that targets our elements with the class of inline-block.
    a. Give this selector the CSS property of display: inline-block; This will cause our navbar to line up next to each other.
    b. Give this selector the property of vertical-align: top; to ensure all elements align next to one another.

2. Next, we need our divs to fill the width of our container.
    a. Create a CSS selector that targets our div with the class of logo.
        i. Set this element’s width to 23%.

        ii. Give this element a padding of 1%.
    b. Create a CSS selector that targets our HTML element nav. If you don’t remember how to target base HTML elements, read this documentation
        i. Set this element’s width to 73%.
        ii. Give this element a padding of 1%.

3. Let’s align our unordered list in its container to the right.
    a. Target the base HTML ul element we created earlier.
        i. Give it a width of 50%.
        ii. Center it using margin-left: auto;.

4. Let’s center our images inside their containers also. In our CSS selector that targets the class of logo:
    a. Give this element the CSS property text-align: center;

Code hero image CSS:

1. To begin, give our section with the class of heroImg a height.
    a. Give this element the CSS property of height: 600px;
    b. Give this element padding: 25px;

2. Create a CSS selector that targets the class of content.
    a. Give this element width: 500px;
    b. Center this element using margin: 0 auto;

Code footer CSS:

1. To build out your footer, write a CSS selector that targets the base HTML footer element.
    a. Give this selector a background color: #373738
    b. Use padding to center your <ul> and <li> elements.

2. Write a CSS selector that targets your <ul> nested inside of our footer element.
    a. Set this element to display as an inline-block level element.
    b. Give this selector the property of vertical-align: top; to ensure all elements align next to one another.
    c. Use padding to center the text inside of our <li> 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 of 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 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 and background-repeat: no-repeat to make your background span your page's full width.
- Apply a drop shadow to the div contained inside your hero image.
- Add a font-family to the text elements in your li and p tags.

Step 6: Upload your coded website's index.html, style.css, and images folder to GitHub and publish via GitHub Pages (2 hours)

As a front-end developer, getting practice publishing your work and hosting it online will not only 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 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 of your developer questions.

Required Deliverables

Google doc with the following sections:

1. A link to your annotated Figma webpage.
2. A link GitHub repository
3. A link to your Github pages website URL.

Submission

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:

Here's what to submit to Bootcamp Spot:

1. **A link to your Google Doc with required deliverable links**