Last week, you started laying the groundwork for your UX/UI portfolio website during the Unit 16 Homework by building the wireframe you designed during week 16. This week we will continue to work on your portfolio by applying CSS Flex, CSS Grid, and media queries in order to make your UX/UI portfolio responsive. In this assignment, you will use CSS flex to build a layout showcasing your UX skills. You will also use CSS Grid to create content for the case study you generated during Week 16. On top of this, you will continue to polish your UX/UI portfolio website’s CSS to make your site more attractive and employer-competitive.

**Note:** If you are using a content management system like Wix or Squarespace to build your portfolio, you MUST still complete this assignment for submission.   
  
*If you did not complete the unit 17 homework, or were not happy with the outcome, feel free to use this starter* [*template*](https://drive.google.com/a/2tor.com/file/d/1TXsrSUu6KVyTutg6ZBAXSk7QoJGRTqyh/view?usp=sharing)*. Make sure you style it to match your visual design from unit 16.*

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

## **Week 18: Apply Responsive Website Design to Your Portfolio Page**

**User Story:** As UX/UI professionals, accessibility is key for your portfolio page. You must not operate under the assumption that your site will only be browsed via desktop. In today’s busy world, it's entirely possible that your target audience will not be able to check out your work by desktop and will browse on their mobile devices instead. This week is your chance to illustrate your front-end development skills by building flexible content that shows off your work and makes your UX/UI portfolio responsive.

## **Requirements (8 - 12 hours to complete)**

* **PART ONE: Redline Your Portfolio’s “My UX Skills” and “My Work” Sections (1 hours)**
* **PART TWO: Create a new GitHub repository for this unit's homework. (30 min)**
* **PART THREE: Code the HTML for your “My UX Skills” and “My Work” Sections. (1 hours)**
* **PART FOUR: Code the CSS Style for the “My UX Skills” Section using CSS Flex (4 hours)**
* **PART FIVE: Code the CSS Style for the “My Work” Section using CSS Grid (4 hours)**
* **PART SIX: Code a mobile media query to make your design responsive for mobile devices (2 hours)**
* **PART SEVEN: Upload your coded Website’s index.html, style.css and images folder to GitHub to update GitHub Pages (1 hours)**

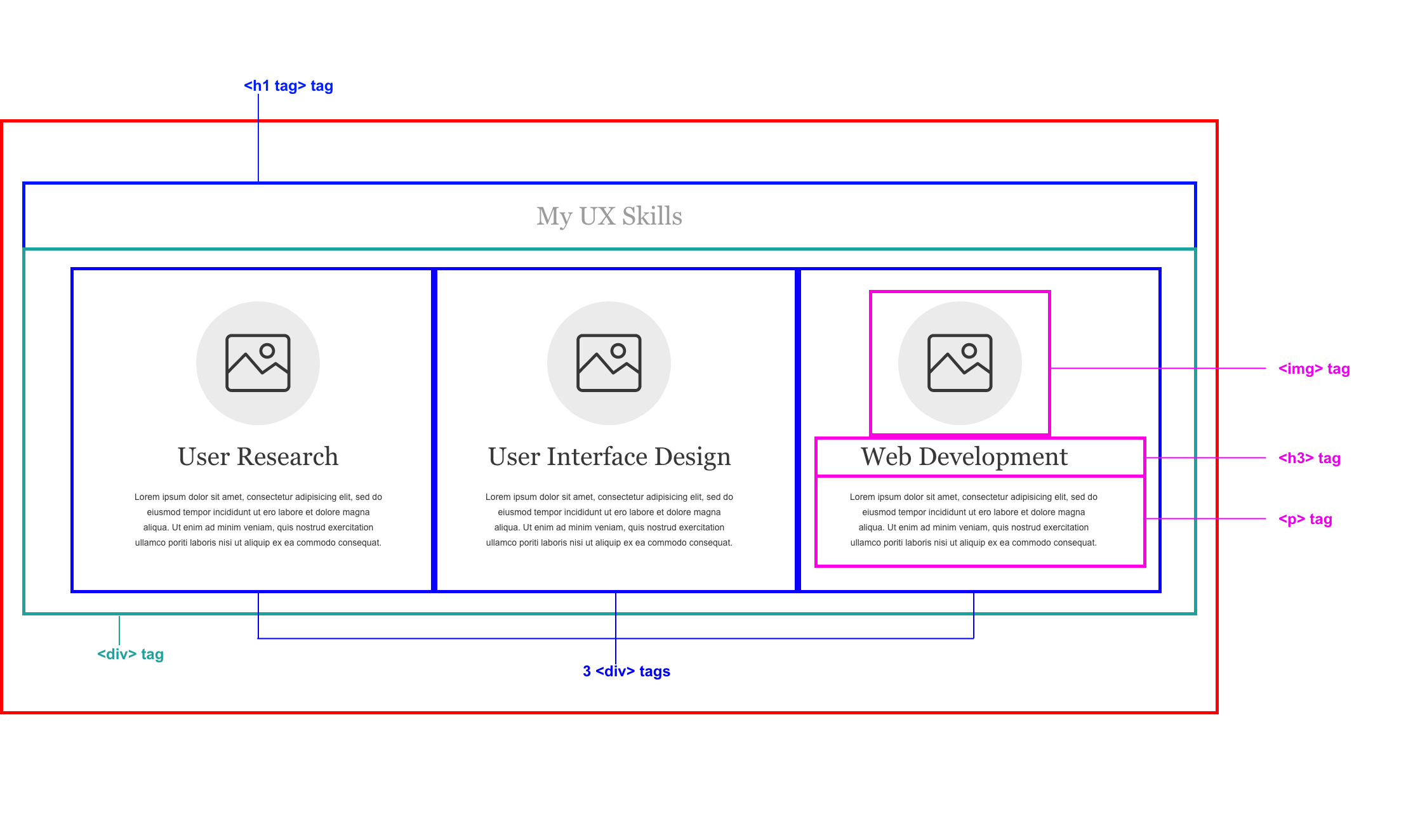
## **Instructions**

**PART ONE: Redline Your Portfolio’s “My UX Skills” and “My Work” Sections (1 hour)**

During the first part of this homework, you will continue to think like a developer by redlining the two sections we will be building using CSS Flex and CSS Grid.  
*Note: If your design does not match the wireframe examples, redline accordingly. Remember this is your portfolio. You have creative freedom over the visual design of your website. If you are not confident in your Front-end Development skills, follow the basic steps here to create a wireframe. Then, style it to visually match your design.*

1. Redline your “My UX Skills” Section. Your redlined design should contain the following tags:
   1. A <section> tag
   2. A <h1> tag
   3. A <div> tag containing the following tags:
      1. Three <div> tags that will serve as our flex children. Each <div> should contain the following tag:
         1. A img tag containing a skill you possess.
      2. A <h1> tag containing the title of the skill.
      3. A <p> tag describing how you apply your skill set.

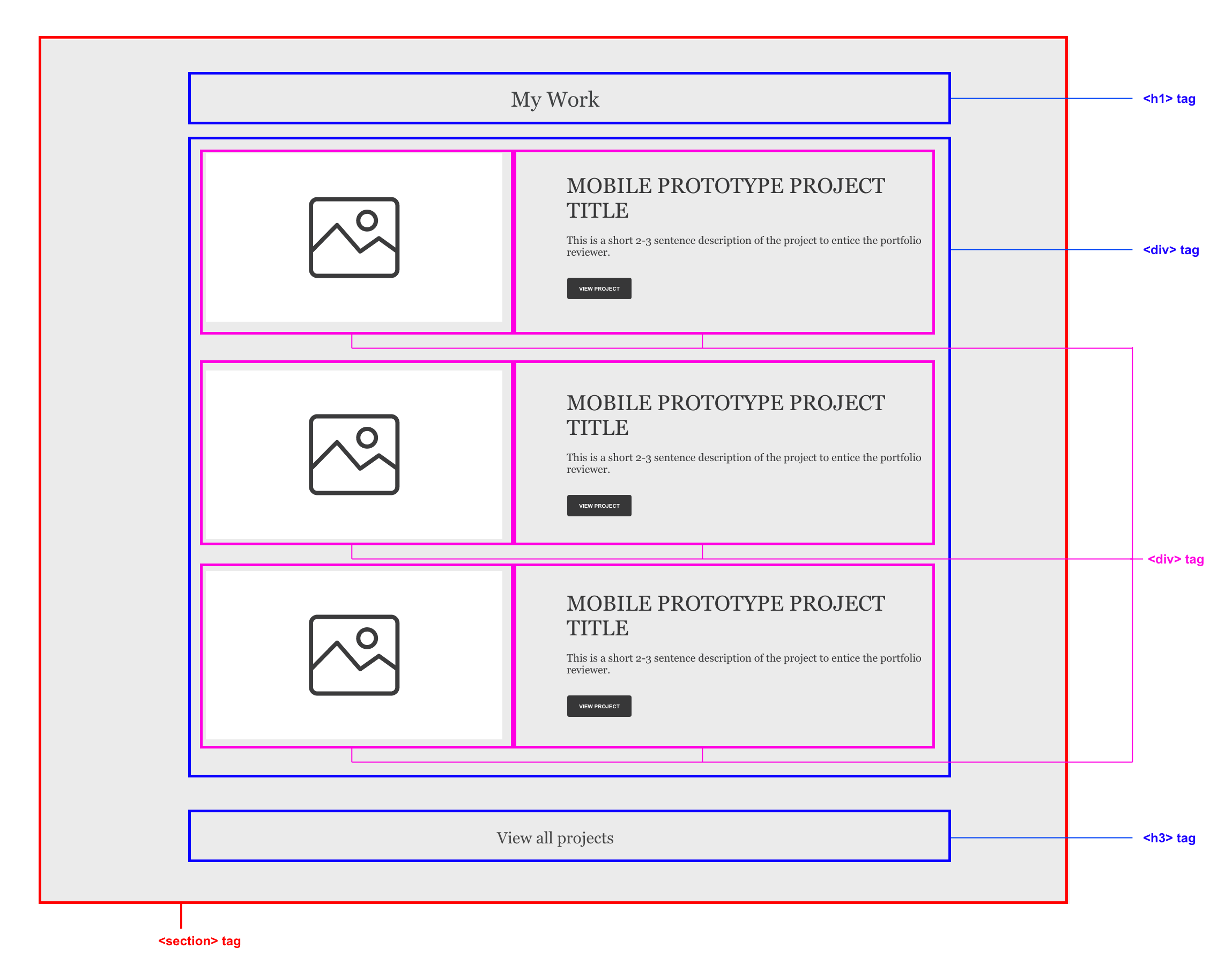
*Below a sample of a redlined “My UX Skills” Section:*



1. Redline your “My Work” Section. Your redlined design should contain the following tags:
   1. A <section> tag.
   2. A <h1> tag.
   3. A <div> tag that contains the following tags:
      1. Two <div> tags side by side.
         1. The first <div> should contain an <img>.
         2. The second <div> should contain a <h4> tag, a <p> tag and a <button> tag.
      2. Two more <div> tags side by side.
         1. The first <div> should contain an <img>.
         2. The second <div> should contain a <h4> tag, a <p> tag and a <button> tag.
      3. Two more <div> tags side by side.
         1. The first <div> should contain an <img>.
         2. The second <div> should contain a <h4> tag, a <p> tag and a <button> tag.

*In total, we should have three rows of two.*

* 1. A <h3> tag.

*Below a sample of a redlined “My Work” Section:* 

Now that we have visually mapped out your layout, let’s create and repo and get coding our sections!

**PART TWO: Create a new GitHub repository to hold this unit's homework. (30 mins)**

1. Before we start coding, create a new repository on [Github](https://www.github.com) and name it UX\_UI\_HW\_18.  
   If you need a reminder of how to create and clone a repository, please reference last week’s [homework](https://docs.google.com/document/d/1BTveuMj619aBGFXeR6c-jarNLz5JXVh0dTgpHCZS1mM/).
2. Clone the blank repository to your desktop using Github Desktop.
3. Copy and paste your homework files from UX\_UI\_HW\_17 into UX\_UI\_HW\_18. You want to be able to show how your code abilities have grown during the course of the code section. Having multiple versions will allow you to show employers how your skills have grown.
4. Update your UX\_UI\_HW\_18 repository with the new files.
5. Publish your site on GitHub pages. Click Settings on your repository’s homepage.
6. Scroll down to the GitHub Pages area in Settings and select the master branch.
7. Check the URL after the page refreshes. You're ready to code!

**PART THREE: Code the HTML for your “My UX Skills” and “My Work” Sections. (1 hour)**

**Code the “My UX Skills” HTML**

1. Create a <[section>](https://www.w3schools.com/tags/tag_section.asp) tag with the [class](https://www.w3schools.com/html/html_classes.asp) of mySkills. Inside of it, create the following tags:
   1. A [<h1>](https://www.w3schools.com/tags/tag_hn.asp) tag that contains the text “My UX Skills”
   2. A [<div>](https://www.w3schools.com/tags/tag_div.asp) tag with the [class](https://www.w3schools.com/html/html_classes.asp) of flexContainer that will serve as our flex container. This div will contain the following tags:
      1. Three [<div>](https://www.w3schools.com/tags/tag_div.asp) tags that will serve as our flex children. Each div should contain:
         1. A [<img>](https://www.w3schools.com/tags/tag_img.asp) tag. Pick or create an icon that will fit the skill.
         2. A [<h3>](https://www.w3schools.com/tags/tag_hn.asp) tag that contains text to describe the skill.
         3. A [<p>](https://www.w3schools.com/tags/tag_p.asp) tag containing a paragraph about how you use the skill.

That’s it for the HTML of our “My UX Skills” section. Let’s move on to the “My Work” Section.

**Code the “My UX Skills” HTML**

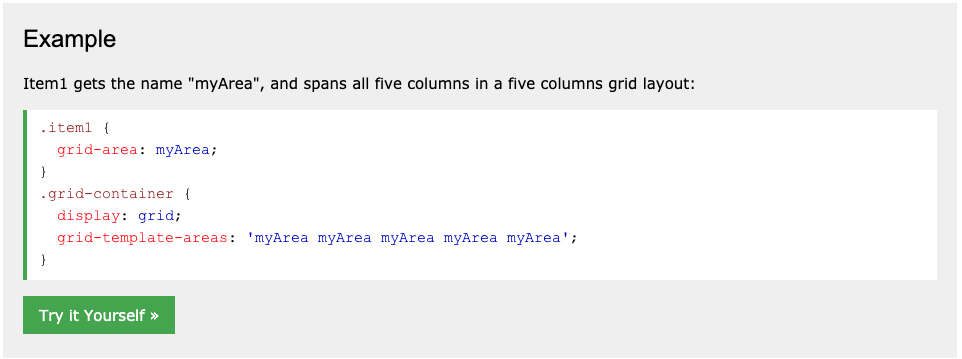
1. Create a <[section>](https://www.w3schools.com/tags/tag_section.asp) tag with the [class](https://www.w3schools.com/html/html_classes.asp) of myWork. This section should contain the following tags:
   1. A [<h1>](https://www.w3schools.com/tags/tag_hn.asp) tag containing the text “My Work”.
   2. A [<div>](https://www.w3schools.com/tags/tag_div.asp) tag with the [class](https://www.w3schools.com/html/html_classes.asp) of workGrid. This div should contain the following tags:
      1. Two [<div>](https://www.w3schools.com/tags/tag_div.asp) tags. The farthest first [<div>](https://www.w3schools.com/tags/tag_div.asp) will contain an [<img>](https://www.w3schools.com/tags/tag_img.asp) tag and the second will contain:
         1. A [<h4>](https://www.w3schools.com/tags/tag_hn.asp) tag.
         2. A [<p>](https://www.w3schools.com/tags/tag_p.asp) tag.
         3. A [<button>](https://www.w3schools.com/tags/tag_button.asp) to view the project.
      2. Two more [<div>](https://www.w3schools.com/tags/tag_div.asp) tags underneath the other two. The farthest first [<div>](https://www.w3schools.com/tags/tag_div.asp) will contain an [<img>](https://www.w3schools.com/tags/tag_img.asp) tag and the second will contain:
         1. A [<h4>](https://www.w3schools.com/tags/tag_hn.asp) tag.
         2. A [<p>](https://www.w3schools.com/tags/tag_p.asp) tag.
         3. A [<button>](https://www.w3schools.com/tags/tag_button.asp) to view the project.
2. Under our grid, create a [<h3>](https://www.w3schools.com/tags/tag_hn.asp) tag tag that contains the text “View all projects”.

**PART FOUR: Code the CSS Style for the “My UX Skills” Section using CSS Flex (4 hours)**

1. If your design differs from the HTML content we constructed earlier, create HTML tags to match your redlined design file for your CSS flex container. This will vary from design to design; each design is different.
2. The first step to creating your “My UX Skills” Section is to set a flex container. Set [display: flex](https://www.w3schools.com/css/css3_flexbox.asp) to your <div> with the class of .flexContainer.
3. Now that you have a flex container created, we need to set the widths of the three <div> tags wrapped inside of it.
   1. Apply width 33% to each <div> wrapped inside to create a fully responsive row.  
      *Note: Don’t forget to apply margin and padding to these <div> tags, creating a layout that is evenly spaced.*
4. Write an [HTML comment](https://www.w3schools.com/html/html_comments.asp) above your section tag that says:  
    <!-- My UX Skills section coded using CSS flex -->
5. Now that you have the basic structure and style created for your My UX Skills Section, your job is to style it to match your visual design! Here are a couple tips to get you working:
   1. Use [flex direction](https://www.w3schools.com/cssref/css3_pr_flex-direction.asp) on your flex container to set it to display as either a column or a row.
   2. If you are using the placeholder content for this section, fill it with your own content from your design you created during the week 16 homework.
   3. If your design differs from the placeholder content, add in your HTML elements to make it match your design and style accordingly.

**PART FIVE: Code the CSS Style for the “My Work” Section using CSS Grid (4 hours)**

1. If your design differs from the HTML content we constructed earlier, create HTML tags for your CSS grid. This will vary from design to design, each design is different.   
   *Note:* For a comprehensive guide on how to use CSS Grid, check out this [tutorial](https://css-tricks.com/snippets/css/complete-guide-grid/).
2. Apply [display: grid;](https://www.w3schools.com/css/css_grid.asp) to your div with the class of workGrid to convert this <section> tag into a [grid container](https://www.w3schools.com/css/css_grid_container.asp).
3. Apply named [grid areas](https://www.w3schools.com/cssref/pr_grid-area.asp) to your divs wrapped inside your grid container.



1. On your grid container, create [grid-template-rows](https://www.w3schools.com/cssref/pr_grid-template-rows.asp) and [grid-template-columns](https://www.w3schools.com/cssref/pr_grid-template-columns.asp) to match the visual design of your page. Keep in mind, you can use percentages, pixels, or fraction units to accomplish this.
2. Build out your [grid-template-areas](https://www.w3schools.com/cssref/pr_grid-template-areas.asp) in your grid container to match your design.
3. Apply CSS styles to your CSS grid to make sure the visual design matches your wireframe that you created during the Unit 16 homework.
4. Create a [HTML comment](https://www.w3schools.com/html/html_comments.asp) above your “My Work” Section that reads:   
   <!-- My Work Section Constructed using CSS Grid -->

**PART SIX: Code a mobile media query to make your design responsive for mobile devices (2 hours)**

1. Write a mobile [media query](https://www.w3schools.com/cssref/css3_pr_mediaquery.asp) at the bottom of your style.css file. Set this mobile media query to take effect at 600 pixels or less using max-width.  
     
   *Scroll to the next page.*

zA close up of a sign

Description automatically generated

1. Inside your mobile media query’s brackets, rewrite the flex-direction of your [flex container](https://www.w3schools.com/cssref/css3_pr_flex-direction.asp) to display as a column.
2. Rewrite the [width](https://www.w3schools.com/cssref/pr_dim_width.asp) of your flex children to make them take up 100% of the width of their container.
3. Rewrite the [grid-template-areas](https://www.w3schools.com/cssref/pr_grid-template-areas.asp), [grid-template-rows](https://www.w3schools.com/cssref/pr_grid-template-rows.asp) and [grid-template-columns](https://www.w3schools.com/cssref/pr_grid-template-columns.asp) of your grid container to display for mobile views.   
   *Note: See below for an example from Unit 18.2 for an example of how you might accomplish this.*  
     
   Here is a sample grid container set to display on desktop:  
     
     
   And, here it is rewritten for mobile browsers. You will have to edit the grid-template-rows and grid-template columns, as well as the grid-template-areas for your grid to display correctly.  
   

**PART SEVEN: Upload your coded Website’s index.html, style.css, and images folder to GitHub to update GitHub Pages (2 hours)**

1. Upload your modified webpage to GitHub using GitHub Desktop.
2. Publish your site using Github Pages.
3. Open your published GitHub Pages url to see your changes applied.
4. Make sure you copy the URL. You will need it to submit your homework via Bootcampspot.

## **Tips**

The following general developer workflow tips 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, and a great resource to help answer all your development questions.

## **Resources**

* [Best practices for mobile design](https://xd.adobe.com/ideas/principles/web-design/best-practices-tips-responsive-design-multiple-screens/)
* [Media Query Tutorial By MDN](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Media_queries)
* [MDN Flexbox basics](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox)
* [MDN basic concepts of Grid layouts](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout)

### **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” when using Google Drive to submit your homework.

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 website.
  + The redlined “My Ux Skills” and “My Work” Sections.
* A link to your hosted GitHub Pages.
* A link to your GitHub repo.