

## WDD 231

[Home](#)[W1](#)[W2](#)[W3](#)[W4](#)[W5](#)[W6](#)[W7](#)

# W01 Assignment: Course Home Page

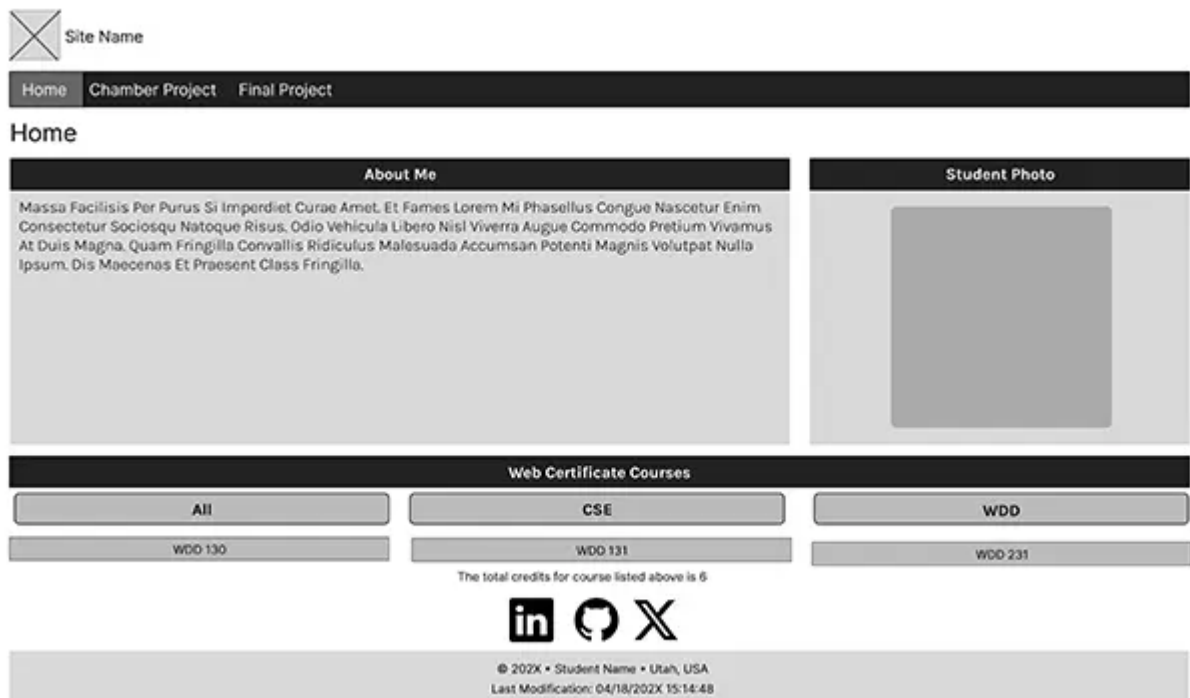
## Overview

This assignment demonstrates your prerequisite knowledge by applying HTML, CSS, and JavaScript to the design and development your course home page.

## Task

Design and develop your page using the following example as a guide for the layout and content. The page must be meet course **standards**, be **responsive**, and pass **accessibility**, **best practice**, and **SEO** tests from DevTools Lighthouse audits.

### ▼ Screenshots



Example Home Page – Desktop



Site Name

**Home****Chamber Project****Final Project**

## Home

### About Me

Massa Facilisis Per Purus Si Imperdiet  
Curae Amet. Et Fames Lorem Mi  
Phasellus Congue Nascetur Enim  
Consectetur Sociosqu Natoque Risus.  
Odio Vehicula Libero Nisl Viverra Augue  
Commodo Pretium Vivamus At Duis  
Magna. Quam Fringilla Convallis  
Ridiculus Malesuada Accumsan  
Potenti Magnis Volutpat Nulla Ipsum.  
Dis Maecenas Et Praesent Class  
Fringilla.

### Student Photo



### Web Certificate Courses

All

CSE

WDD

WDD 130

WDD 131

WDD 231

The total credits for course listed above is 6



© 202X • Student Name • Utah, USA  
Last Modification: 04/18/202X 15:14:48

Example Home Page – Mobile

## Instructions

### Step 1: File and Folder Setup

1. In VS Code, open your **wdd231** folder that you cloned from your GitHub account during the [Setup: Hosting on GitHub](#) activity.
2. Create a default home page for this repository so that when published to GitHub, will render at <https://yourgithubusername.github.io/wdd231>. What must this file be named?

#### ▼ Check Your Understanding

**index.html**


3. Create appropriately named, supporting folders for images, CSS, and JavaScript files.

These subfolders must be placed in the root directory, **wdd231**.

#### ▼ Check Your Understanding

- Images will be stored in a folder named **images**
- CSS files will be stored in a folder named **styles**
- JavaScript files will be stored in a folder named **scripts**

Review the [Naming Conventions](#) learning activity.

Note that the words *directory* and *folder* have essentially the same meaning. Directory is the more accurate term for file systems while "folder"  refers to the graphical metaphor that is generally accepted because it is highly related to the term "file" in the organized world.

4. Add CSS files as needed. You are required to write your own custom CSS in this course. Frameworks and libraries are **not** allowed. The assignment requires you to create a responsive layout for all screen widths from (320px) and up. For this class, you may use a normalize CSS file. The class requires you to use a small.css followed by a larger.css file. The larger css file must contain a media query.

5. Add multiple JavaScript files as needed. For example, create files to support the responsive navigation (navigation.js), dates (date.js), and course information cards (course.js). Reference these in the **head** using the **defer** attribute method. As you move through this course, you will need to add additional JS files. Only load the files for each page that are needed for that page.

## Step 2: HTML

1. Add the standard **HTML document structure** to the document.
2. Add the required elements to the **<head>** including the **title**, **meta description**, and the **meta author**.
3. Use **semantic HTML** to create the basic layout structure of the page. Use a **header**, a **nav**, a **main**, and a **footer** element.
4. Add the **content** as shown in the example screenshots above.
5. Note that all images used on your pages must be optimized.

In this class, the image memory size threshold is **125 kB** or less per image.

6. The **<header>** tag should contain a site logo (using an svg file) and your name rendered using a **<span>** tag, and a hamburger icon on small screens.
7. The responsive navigation **<nav>** menu contains four links:
  - Home** – this current page.
  - Chamber** – will link to the Chamber of Commerce website project (week 2).
  - Final** – will link to the Final project (week 6).

8. The **<main>** element contains the following:

Home rendered using an **<h1>** tag.

Three **<section>** tags with headings **<h2>** as shown in the example screenshots above.

9. The **<footer>** has three items.

1. The first item is a series of social media links including GitHub and Linked In and one or two others that you choose. These icons should link to your GitHub repository and your Linked In account.
2. The next line contains the following:

the copyright symbol and current year that are written dynamically using JavaScript

### ▼ Check Your Understanding

In HTML, provide a placeholder that can easily be selected using JavaScript.

Example: `<span id="currentyear"></span>`

your name

your state or country

3. The second paragraph has an **id** of "**lastModified**" and will be populated dynamically, with JavaScript.

## Step 3: CSS

1. As you style and layout your page, ensure that the following, required **web design principles** and best design practices are met.

Appropriate **proximity, alignment, repetition**, and **contrast** design principles are used.

There is no horizontal scrolling in views between 320px to 1200px wide.

Text does not touch the edge of the screen causing a visual tangent error.

Any text overlays on images, other elements, or other text/content is legible and does not overflow in smaller views.

That there is no twitching in the navigation on hover.

Images are not squished nor distorted as the page is resized.

Images are not oversized causing pixelation.

2. Use your own color schema and typography choices.
3. Use the [Google Fonts API](#) to select one font family to use on this page. We covered this in [week 1 of WDD131](#) if you need to review Google Fonts
4. Responsive navigation must support the following features:

Responsive design includes providing a positive user experience of **readability, usability, accessibility, visual appeal** that adapts to the user's device by size, orientation, and resolution.

small-screen hamburger like navigation

large-screen horizontal navigation using [CSS Flex](#)

wayfinding

hover effect for menu items

5. Your design cannot just be a simple one column layout (default flow layout) in all views in order to warrant full credit consideration.

## Step 4: JavaScript

1. Reference JavaScript files by using a `<script>` reference in the **head** of the HTML file and using the attribute **defer**.

Why is the [defer](#) boolean attribute important?

2. Use JavaScript to support the responsive menu.
3. Use JavaScript to dynamically output the following:

the copyright year (the current year) in the footer's first paragraph

Note this [CodePen](#) summary of the **Date** object.

the date the document was last modified in the second paragraph

You can use the **lastModified** property of the document object to get this date/time dynamically. An example of this is shown below:

```
document.getElementById("lastModified").innerHTML = document.
```



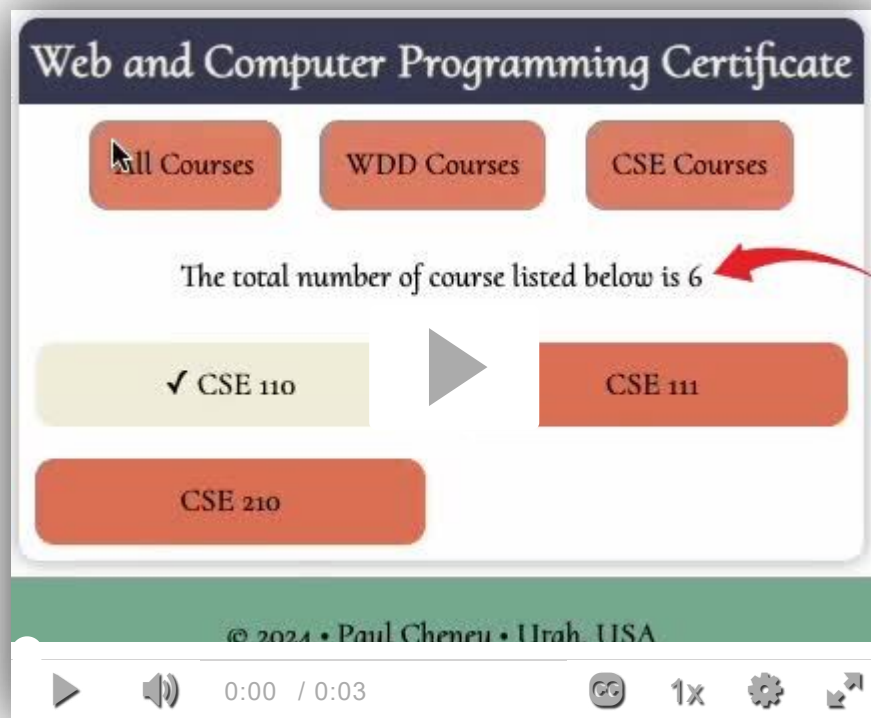
Note that **document.lastModified** returns a simple string in JavaScript. Therefore, you do not need to manipulate its output for this assignment.

4. Copy this array of course objects into a JavaScript file: [Course List Array](#)

This array contains the course information for the required courses that are in the first certificate called [Web and Computer Programming](#) of the [Software Development degree](#).

5. Modify the **courses** array content in your script file by changing the **completed** property to **true** if you have completed a course.
6. Dynamically display all the courses in the certificate section as shown in the example above. The courses that you have completed must be marked in a different way versus those that you have not completed. Use your page color scheme. The page should adjust automatically if the data source changes.

7. Using buttons that listen for the click event, allow the user to select to display **All Courses**, **WDD Courses**, or **CSE Courses**. Hint: Use the array **filter** method.

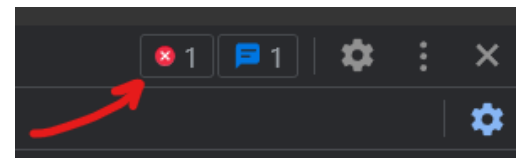


Demonstration of User Interface Filtering Course Output

8. Design the course cards to indicate those courses that **you have completed personally** in a complimentary, but different style than the rest, indicating course completion.
9. Provide a total number of **credits** required dynamically by using a **reduce** function (not shown on the screenshots). The number of credits shown should reflect just the courses currently being displayed.

## Step 5: Testing

1. Continuously check your work by having it loaded in your browser using Local Server.
2. Use the browser's DevTools to check for JavaScript runtime errors in the console or click the red, error icon in the upper right corner of DevTools.




"DevTools" is an abbreviation for the browser's "Developer Tools." It refers to a set of tools or utilities provided by web browsers to help developers debug, profile, and analyze web pages during the development process. The tools are typically accessed by pressing the F12 function key or selecting the menu option for the browser's developer tools.

3. Use DevTools [CSS Overview](#) to check your color contrast.

4. Generate the DevTools **Lighthouse** report and run diagnostics for **Accessibility**, **Best Practices**, and **SEO** in both the mobile and Desktop views.

Use the **Private** or **Incognito** mode to test your page when using Lighthouse. The standard is to reach a score of 95+ in each of these categories.

## Submission and Audit

1. Commit and sync your local work to your a GitHub Pages enabled **wdd231** repository
2. Use this  [audit tool](#) to self-check your work for some of the required HTML elements and CSS content. This audit tool is also used by the assessment team.
3. Return to Canvas and submit your GitHub Pages enabled URL for wdd231, e.g.,

`https://your-github-username.github.io/wdd231`

Do not need to include **index.html** in the URL reference because index.html is the default file retrieved with a folder request on GitHub Pages.