

# Fall Website Final Project

The final project is a series of exercises that will guide you through the development of the pages for a Fall website. There will be four html pages at the end of the project, one CSS file, and the usage of files given to you at the beginning.

## Guidelines for the Fall Website

Unlike the exercises in the book, the exercises for this project do not give complete step-by-step instructions.

These specifications are detailed enough for you to complete each exercise. However, you'll need to use your best judgment on how to code some of the details. To do that, write the code in the way that you think is best, based on the skills that are presented in the book; however, please make sure to follow the specific requirements as outlined. There are online resources that may help – however, note: copying and pasting entire blocks of code from the internet is not allowed.

Download the starting files for the website. These files are stored in a folder named `final_project` and include the `index.html` and all supporting files. The exercises will provide instructions for when to use each of these files.

The pictures are provided to give a rough estimate of how your project should look. If the fonts, sizes of headings, and some layout items should differ slightly – that is fine.

**Do not use html id attributes in this project.**

Correct file and folder structure: 10 points.

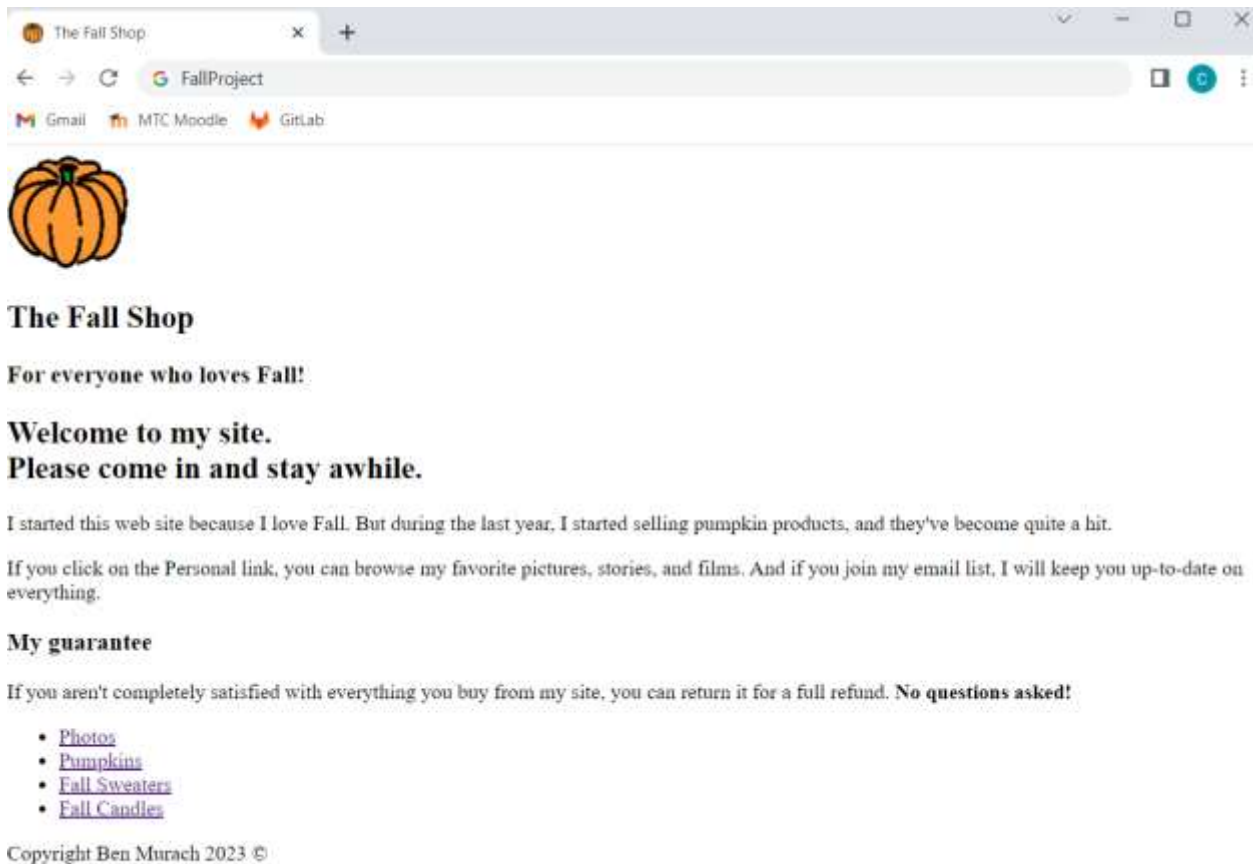
All files included in zip file: 10 points.

All file paths are relative and work on any operating system: 40 points.

No use of id attributes: 10 points.

## Create the Home Page

In this exercise, you'll code the HTML for the home page. When you're through, the page should look like this:



## Specifications

Open the home page: index.html located in the root folder for the project. This file includes just the basic HTML elements and one comment.

Use the HTML semantic elements to structure the page so it includes a header. The header includes child elements of the image and two headings shown above: h2 is the element for “The Fall Shop” and h3 is the element for “For everyone who loves Fall.” The main content belongs in a main element that is directly under (a sibling of) the header. The main element includes a section and an aside. The aside element has a child navigation element, and the child elements that create the secondary menu. Below the main element is the footer.

Add an image and two levels of headings to the header (use h2 and h3). The image, named header-pumpkin.gif, is in the images folder. Add a width attribute to the image of 95 pixels.

The text for the header and other areas of the page are located in the index.txt file in the text folder.

Add a primary heading and the required text for the main content (...Welcome to my ... stay awhile. --see above). Make sure each of the set of sentences below the primary heading are in paragraph tags. Using just HTML, boldface the last sentence of the guarantee. Place "My guarantee" in an h3 element.

Place the following portion of the website in a section element:

**Welcome to my site.**

**Please come in and stay awhile.**

I started this web site because I love Fall. But during the last year, I started selling pumpkin products, and they've become quite a hit.

If you click on the Personal link, you can browse my favorite pictures, stories, and films. And if you join my email list, I will keep you up-to-date on everything.

**My guarantee**

If you aren't completely satisfied with everything you buy from my site, you can return it for a full refund. **No questions asked!**

The nav should be a child of an aside and contain an unordered list, with list items and links. Code the product categories as an unordered list with links. Use the # in the href attribute – the links will "go nowhere" and remain on the page when clicked.

- [Photos](#)
- [Pumpkins](#)
- [Fall Sweaters](#)
- [Fall Candles](#)

Create a footer element – child of the body - under the main element, uncomment out the HTML and JavaScript provided in index.html and placed inside the footer. The JavaScript will keep the copyright year current without having to update the site in the future.

Add in the favicon in the head. Update the title element in the head – The Fall Shop. Add a unique meta description element. Add unique meta description elements on every page.

Header: 15 points.

Main and children: 40 points.

Side navigation: 10 points.

Footer: 5 points.

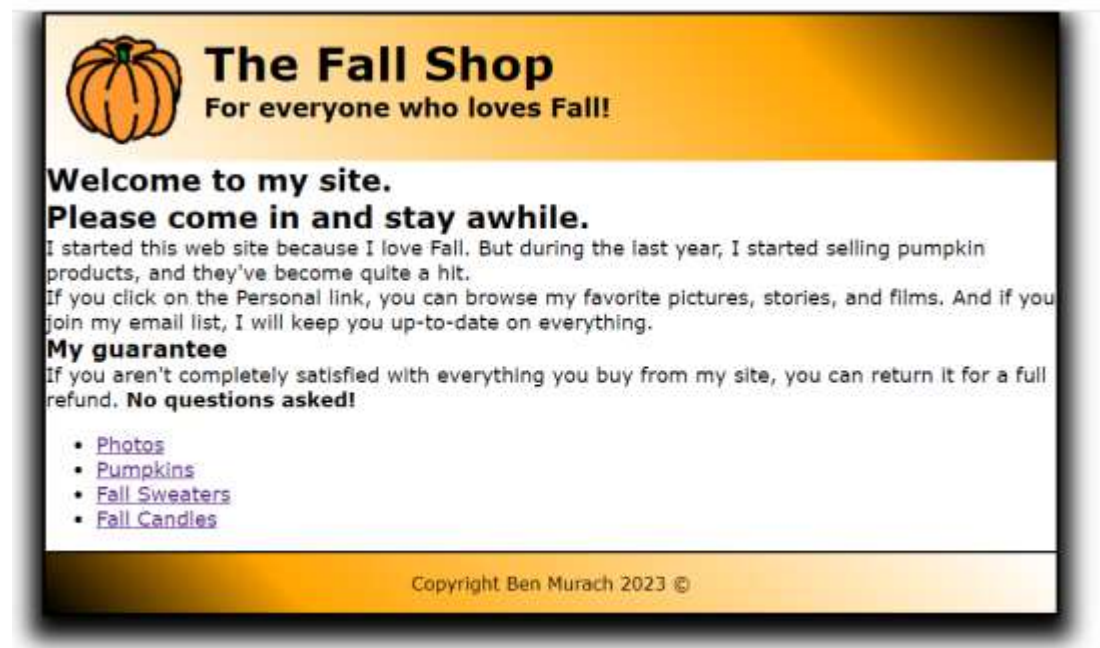
Head: 10 points.

## Format the home page

In this exercise, you'll create an external style sheet that you can use to format the Fall Shop home page. You'll also use the CSS box model to add spacing, borders, and backgrounds. When you're through, the page should look similar to this:



(Zoomed in center)



## Specifications

Create a folder named `css` and create an external stylesheet named `style.css` for the home page. Make sure to link it in the `index.html`. All styles must be either type or class selectors, they may not be id selectors.

Format the body of the page so it's 850 pixels wide, has a white background, has a black border with a shadow, and is centered in the browser window.

Add a background image behind the body: use `foliage.png` in the `images` folder. This image should repeat both horizontally and vertically.

Set the default font for the document to a sans-serif font; make sure there are two other fonts (of your choosing) in front of that default.

Create the font sizes for all of the headings. Although it's not a normal progression – h2 is the biggest, then h1, then h3 (see above).

In the header, float the image to the left and indent (using padding is fine) both headings per the screen shot above. Use descendant selectors, header being the parent, h2, h3 being children.

Add a black border below the header and above the footer.

Reduce the font size for the footer and center it.

Add a 45-degree gradient to the header and footer. The gradient in the header should range from white at the left to orange  $\frac{3}{4}$  of the way to the right to black at the right. The gradient in the footer should be the reverse.

Link to stylesheet: 10 points.

Body CSS: 10 points.

Header CSS: 10 points.

Borders and gradients: 10 points.

## Use a 2-column layout

In this exercise, you'll enhance the formatting of the home page so it uses one way of creating a 2-column layout (there are other ways to do this). You'll also add some additional content to the page. When you're through, the page should look similar to this:



## Specifications

Adjust the side navigation with the unordered list to look like a sidebar that's floated to the left of the remaining content, (which is coded in a section). The width of the side navigation should be 160 pixels. Adjust the section so it is on the right of the side navigation. Be sure to clear the footer so it remains at the bottom of the page. CSS rules for the side navigation may not use class or id selectors.

Adjust the margins and padding for the side nav and section as necessary. (Note: Because the unordered list is given some left padding by default, you might not need to apply any additional margin or padding to it.)

Modify the HTML for the first heading in the section so it's displayed on two lines if it hasn't been already.

Add spacing between the headings and the paragraphs as shown above.

Format the first letter of the primary heading in the main content so it's larger than the other letters in the heading.

Create a secondary heading as shown (“Product of the Week”). Add the image (cornucopia.png) and the link below the heading (“Cornucopia Table Decoration”). The link should display just the image in the browser when clicked. Open the image in a new browser window. The link should be orange initially and green when hovered over.

Adjust the margins and padding for the elements on the page so it looks as shown above.

Side navigation: 20 points.

Primary letter: 2 points.

Image and heading: 15 points.

## Two navigation menus

In this exercise, you'll enhance the home page so it includes a vertical navigation menu, a two-tier horizontal navigation menu. When you're through, the page should look similar to this:



## Specifications

Add a 2-tier (2 level) navigation menu that includes the links shown above, use the semantic element for navigation. Place the new nav under (below) the header but above the main element. It should not be included in the header or the main; it is a child of the body. Only "Decorating Ideas" has a sub-menu as shown above. For the styles for the top menu, use the class name `top_2_tier_menu`.

Format the navigation menu so the link for the page that's currently displayed (in this case, Home) has orange text, and so the link that the mouse is hovering over in the horizontal navigation (in this case, Decorating Ideas) has a gray background.

All links except the Home and two additional links should have an href attribute that "goes nowhere" as referenced earlier in the project. Two links should be updated – Join Email should link to a page called `email.html` that will be in the root directory. The link for "Personal" should be linked to `favorites/personal.html`. **These pages have not yet been created and will be created in the following exercises.**

Format the links in the sidebar as a vertical navigation menu as shown above. Be sure to make the entire box for each link clickable – not just the words. This is done with a few properties in CSS. Remove the bullets, change colors.



Top 2 tier navigation: 50 points.

Update side navigation: 20 points.

## Create a page that uses a form

In this exercise, you'll create a page that includes a form for joining an email list. In addition, you'll create a page that's displayed when the form is submitted. When you're through, the pages should look similar to the next two images:



## Specifications for the email form

Create a new page named email.html in the root folder of the project by copying the index.html. You will be only working in the section element – clear out the index specific information.

Modify the horizontal navigation menu so it indicates that the Join Email page is the current page.

Add the heading and the form elements to the section. Code the form element so a page named subscribe.html in the root folder is displayed when the Subscribe button is clicked after validation. The Subscribe page is shown below.

Add the label and input elements for the form; all labels for the fields must be in label elements. The focus should be on the first text field when the form is first displayed. Add the button element to the form.

Align the controls as shown above, and format the button/submit so it's orange with a black border. Add these styles: label, input, and button to style.css. Add a comment in the CSS file that explains those styles are only for the email page.

Use the data validation attributes to require entries in only the first three text fields. In addition, use a regular expression to validate an entry in the Zip Code field, and display an appropriate message in the tooltip if the entry is invalid.

Use a placeholder to display a message in the Zip Code field as shown above.

Email page template: 10 points.

Form HTML: 50 points.

Form CSS: 15 points.



### Specifications for the subscribe (thank you) page

Create a new page named subscribe.html from the index.html in the root folder for the project. You will be only working in the section element – clear out the index specific information.

Modify the horizontal navigation menu so it indicates that no page is current; the subscribe page is not a link in either navigation menu.

Add the heading and paragraph shown above, and format it as necessary.

Subscribe page: 10 points.

## Create a page that uses video

When you're through, the page should look similar to this:



### Specifications

To create the personal page, copy the index.html to the provided favorites folder and rename to personal.html. Remove all of the information from within the section element. Only code the new HTML inside the section element. Make sure the CSS file still works – the path will need to be adjusted. So will the paths for the links and the images. Use only relative pathing.

Modify the horizontal navigation menu so it indicates that the personal page is the current page.

Add a heading and the video to the section. You'll find the video file you need in the media folder. The video shouldn't start playing until you click on the play button.

CSS: Add a transition to the sentence "Wow, I just love mountains in the fall!" that is 1 or 2 seconds, transition the color from the original black to orange, and make the transition work when the user hovers over the words. Make sure it is only on that sentence and doesn't affect the other pages. The end of the transition is shown below:



Remember to add a unique meta description element to all four html pages.

Personal page: 10 points.  
Correct pathing: 20 points.  
Video: 5 points.  
CSS transition: 10 points.  
Unique meta: 3 points.