

# CSC372 Web Development

## Term Project Milestone 1: Product Page Prototypes

Due Thursday, Feb 6, 2025.

---

### Introduction

In this milestone, you will create the starting page prototypes for your web application. A prototype shows the visual elements of the page, but the page itself is not functional other than to link to the other page prototypes you have created. You will create prototypes for the home page and the product pages for your online store.

A website's home page is typically designed to engage potential users and get them excited about what the application can do. It highlights the key features and provides convenient ways to navigate and continue using the application. Each of these components is discussed in detail in the specifications below. Be creative and have fun!

### Specifications

#### Home Page - index.html

##### The "Splash" Section

This is the marketing material designed to attract new users to the application. At a minimum, it should include a title, tagline, brief description of your application and featured products, along with a clear call to action for potential users. However, for full credit, you will need to use your imagination to come up with content that draws potential users in. (trending products/newest etc.) Here are some questions to help you get started.

- What is a catchy name and tagline for the application, and what is an engaging way to display them?
- What color schemes, images, or even videos could be created and tastefully incorporated into the home page to make it appealing to potential users?
- What is the best way to briefly and convincingly describe the main features of the application (i.e. font families, sizes, colors, bulleted lists, text positioning, etc.) so that a new user will naturally respond to the page's call to action (e.g. sign up for an account)?

##### Search Field

There should be a search field to allow for the user to search for specific products. Eventually the search will lead to a product listing page with products that match the search criteria. But for now, create a place for the search term to be entered and a "submit" or "go" button to be clicked.

#### Product List page - products.html

This page lists all the products available for the category the user selected. This should give minimal information about the product but enough to interest the visitor. Each item listed on the page should have:

- A picture/thumbnail of the item
- The product name.
- The price
- A link to view the details (use href="#" for now)

The desktop view should accommodate three or four items across per row. Think about the kinds of controls that might enhance the user experience on your listing pages (i.e. Sorting options, filters, paging controls, etc.). Include any in your prototypes that you feel are appropriate.

#### Product Details page – details.html

This page provides all the details for one of the products you are selling. The page should have:

- A button to add the item to the shopping cart (right now it will not do anything)
- The picture of the product (later there may be more pictures)
- The name
- A description
- The price
- At least 2 other pieces of information that would be of interest to the visitor.

## Navigation Controls

Most applications include controls that allow users to easily access the main features of the system. In a web application, such navigation controls are typically located at the top, left, or right side of a page and are usually accompanied by the application title and/or logo. The controls remain consistent across each page of the application (with rare exceptions). Your web application will need navigation controls (i.e. hyperlinks or buttons) to allow users to quickly access the following key features.

- View items in various categories that you define.
- Get to the shopping cart.
- Signing in or out of the application.
- Getting to the application's home page (via a linked title or logo).

Implement a set of navigation controls such that they appear in the same place and format on every page of your web application.

## Footer

End your pages with a brief footer that includes copyright information and anything else you want users to know about the site. This footer should be set up so that it is reproduced at the bottom of every page of the application.

## Additional Guidelines

- Think carefully about styling your pages – that is the color scheme, background, fonts, images, etc. you use in it. Consider how these might make your application's user experience more appealing to your target audience. Bear in mind that the styles you use for your application's home page should be replicated across other pages of your site. This will foster a consistent experience for your users.
- Again, have fun! Express yourself throughout this project in a manner consistent with who you are as a person. (For instance, if you are analytical and logical, your application's user interface might look different than if you have a goofy sense of humor.) This is art as well as science.

## Technical Requirements

- Follow the coding style guidelines.
- All pages should use the same header, footer, and styles (thematic look and feel).
- You must completely separate your HTML from your CSS, putting each into its appropriate file. (i.e. avoid inline styling).
- All CSS rules should reside in a single file named *styles.css* and be applied to HTML elements via appropriate class or id attributes. Place *styles.css* in a "styles" subfolder.
- **Your home page should employ at least 5 distinct block level HTML elements and at least 2 inline HTML elements. Favor using semantic elements instead of divs where it makes sense.**
- Any other resources needed by your pages (i.e. image files) should reside in the sub-directory "images". Make sure your relative links to the files work as desired.
- Your pages must look and work correctly in the latest versions of Firefox, Chrome, and Edge.
- You can use "href=#" for links to resources you have not created yet, such as those pointed to by navigation links.

**Submission:** In your GitHub Pages repo, make a subfolder for the project and add your work in there. Add links to the project on your profile page, just like you did with Homework 1. Later when we have a backend, we will make a repo for deploying at a hosting service!