**Primary Class Project**

Establish an e-commerce website using WordPress and the WooCommerce “Storefront” theme.

**Basic Configuration**

* WooCommerce “Storefront” theme used as parent-theme (... *wp-content/themes/storefront*)
* A child-theme was added to readily make “look & feel” changes via a custom css file (... *wp-content/themes/storefront-child/****style.css***)
* Custom “child” CSS was activated using a local functions.php file (... *wp-content/themes/storefront-child/****functions.php***)

<?php

function local\_enqueue\_styles() {

wp\_enqueue\_style('parent-theme', get\_template\_directory\_uri() .'/style.css');

wp\_enqueue\_style('child-theme', get\_stylesheet\_directory\_uri() .'/style.css', array('parent-theme'));

}

add\_action('wp\_enqueue\_scripts', 'example\_enqueue\_styles');

?>

**Extra Credit Assignment**

**Assignment as presented by instructor**

Create a custom WP contact form plugin by making minor cosmetic changes to a pre-existing plugin that was provided.

**Assignment on steroids**

I opted to re-design the plugin from the ground up using CSS Grid Layout.

Intro

CSS Grid Layout is a method designed for the two-dimensional layout of items on a page. You can find ALL the technical details for CSS Grid Layout on the W3C website (<https://www.w3.org/TR/css-grid-1/>).

The basic concept of a grid implemented in CSS provides a great deal of flexibility in the visuals that can be produced …

* Grids can readily adapt to changes on viewport
* Elements can be constrained to reside in one or more grid cells & elements can overlap
* Moving elements to a different location on the grid is simple

Using CSS Grid Layout in the customization of the WP contact page plugin turned out to be unexpectedly positive experience.

My approach

I determined number of columns and rows, position of grid in viewport, position of each element on the grid, and styling needed for each element.

The elements of the contact form (i.e. name, email, message, etc.) are defined in ... *wp-content/plugins/**BB-contact-the-dragon.php*. The styling scripts for these elements are in ... *wp-content/themes/storefront-child/****style.css***. The .php file contains the hooks needed for element styling and element grid position. The .css file contains the styling scripts and the grid positioning scripts.

A styling example …

***BB-contact-the-dragon.php***

echo '<div id="item-12"><textarea class="cfTextareaDD" … </textarea></div>';

In the statement above, the message box is implemented as a textarea input control and a class named cfTextareaDD is associated with it.

***styles.css***

.cfTextareaDD {

width: 100%;

text-align: left;

line-height: 1em;

color: black;

background: white;

border-style: solid;

border-width: 2px;

border-color: black;

}

cfTextareaDD is the class/hook defined in ***BB-contact-the-dragon.php*** that is associated with the textarea input control. textarea can now be styled via .cfTextareaDD (see above).

A grid positioning example …

***BB-contact-the-dragon.php***

echo '<div id="item-12"><textarea class="cfTextareaDD"…</textarea></div>'; <<< message box

In the statement above, the message box is uniquely identified as item-12 via id=. You use “id” instead of “class” in this circumstance because “id” is unique and “class” is not.

***styles.css***

#item-12 {

grid-column: 1/span 4; <<< message box starts in col 4 & spans (goes across) 4 columns; resides in col 1-4

grid-row: 5/span 2; <<< message box starts in row 5 & spans (goes across) 2 rows; resides in rows 5 & 6

}

In ***BB-contact-the-dragon.php***, we defined item-12 to be associated with the message box. In the .css file, we can use that id/hook to position #item-12 (aka the message box) on the grid. (see above)

Associated files:

files found in *Bill Burns-custom\_WP\_plugin* folder:

*BB- Custom WP plugin Creation-Use of CSS Grid Layout.docx* (this file)

Synopsis of the creation of a custom WP contact form plugin.

*BB-CSS grid layout design.xlsx*

Design tool I created in Excel to help me position the grid elements.

files found in *'wp-content\_themes\_storefront-child'\_* folder:

*functions.php*

Source code used to activate (enqueue) CSS at the “child theme” level.

*style.css*

Scripts to style the page elements at the “child theme” level and the custom plugin *BB-contact-the-dragon.php*.

files found in *'wp-content\_plugins'\_BB-contact-the-dragon\_* folder:

*BB-contact-the-dragon.php*

Source code responsible for rendering the contact form, collecting user input, and sending the collected data to the hosting server