# Lab 2 - Instructions

## CST8285

## LAB OBJECTIVE

The objective of this lab is to get familiar with the following:

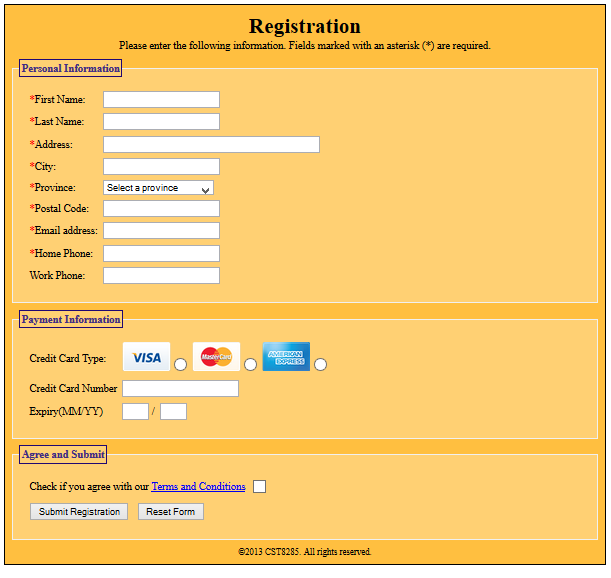
* Semantic HTML and Cascading Style Sheets (CSS)
* Experimenting with CSS library
* Experimenting with flex box
* Good documentation and indentation practice

## Earning

To earn your mark for this lab, each student should finish the lab’s requirements, submit your lab on the Brightspace and demonstrate the working code to the instructor.

You have two parts I and II

Part I: Sample page



In this lab, you will be modifying some website files to add some CSS styling, as well as add a web form. Once the website files have been modified, you will upload them to bright space **Please read the entire lab before beginning.**

### Step 1 - Download and Unzip the website files

There is a file called lab2.zip on bright space under Week 3 - Course Documents - Lab 2. Download this file and extract it to a folder called Lab\_2 somewhere on your hard drive. This will be your project folder.

### Step 2 - Review the file Lab2.html

The file Lab2.html contains some tags that we have not covered in class. Please familiarize yourself with these tags. Specifically, find some information on:

* <fieldset>
* <legend>

### Step 3 - Link your stylesheet to the lab2.html file

There is a stylesheet called style.css in the css folder, in the project folder. Using a <link> element, link the stylesheet to the HTML page.

### Step 4 - Change the <title> of the web page

The format of the title should be Lab 4 by Firstname Lastname

### Step 5 - Create the <form> element

Wrap all of the fieldset elements in a <form> element. The method of the form should be post, and the action should be javascript:void(0);

### Step 6 - Create the form elements

In each fieldset, there is a list of comments. Each comment will contain the name of a form field, and the type of form element for that field.

Here is an example: <!-- First Name | textbox, size of 40.-->

Create a two column table in **each** fieldset. For each comment in the fieldset, create a row in the table that has the name of the form field in the first column, and the form element in the second column.

Each form element should have a name and an id at minimum. There may be other attributes required, which will be described in the comment after the form element type.

For example, the above comment would look like this:  
<tr>  
    <td>First Name:</td>  
    <td><input type="text" name="firstName" id="firstName" size="40"></td>  
</tr>

If, in the comment, a form field name has an asterisk (\*) before the name, include it in the cell. Surround it with a <span> tag, with a class of required.

#### Notes on the Personal Information section

You will be required to create a dropdown box containing all the provinces. There should be an option for each province, and the value attribute of each option should be the province abbreviation. A list of provinces and their abbreviations is included in the comments.

#### Notes on the Payment Information section

The first form element is a group of radio buttons used to select a credit card type. There are three images in the images folder in the project folder. They are:

* visa.png
* mastercard.png
* amex.png

Using the <img> element, display each icon beside its corresponding radio button. Your result should look like this: 

#### Notes on the Agree and Submit section

The last row in the table does not contain a field name. Instead, place a submit button in one column, and a reset button in the other.

### Step 7 - Add some style with CSS

There is a comment in the CSS stylesheet that contains instructions for defining styles to apply to your web page. Please create the rules as described in the CSS file.

### Step 8 - responsiveness

Create a responsive layout using media queries.It needs to align to the screen size in response to resizing the screen size manually. Define a media query for screen widths less than or equal to 600 pixels.

### Step 9 – Documentations

### Structure the code , follow best practices, comment and add great spacing and indentation(Both par I and II).

End part I

### Part II: To understand and practice using CSS flexbox for creating flexible layouts

### Part II of lab2 will be implemented in lab3-css folder

This part will give you a hands-on experience in utilizing CSS flexbox to build flexible and responsive layouts.

Instructions:

1- Open the starter code file “flexbox.html”

2- Create a new CSS file and name it "flexbox.css" in lab3-css folder.

3- Open the "flexbox.css" file and add the necessary CSS code to complete the following tasks:

a) Add margin and padding 0 pixel to all the body elements.

b) Add any font family from your choice to all the body elements

c) Set the background color of the header to "#333" (Dark charcoal)and the text color to white

d) Add padding of 22 pixels to the header.

e) Center align the text in the header.

f) Create a flex container with the class "container".

g) Set the display property of the container to "flex".

h) Arrange the flex items horizontally and ensure equal space between them.

i) Allow the flex items to wrap onto the next line when there is not enou gh space.

j) Set the maximum width of the container to 800 pixels and center it horizontally on the page.

k) Add a margin of 40 pixels on the top and bottom of the container.

l) Style the boxes with the class "box" inside the container.

m) Set the background color to "#f1f1f1".

n) Add a border of 1 pixel solid color "#ddd".

o) Set a border-radius of 5 pixels.

p) Add padding of 20 pixels to the boxes.

q) Create equal-width boxes using flexbox properties.

r) Add margin 20 pixels to the paragraph located insides the footer

s) Ensure each box has a minimum width of 200 pixels.

t) Create a responsive layout using media queries.

u) Define a media query for screen widths less than or equal to 600 pixels.

v) Inside the media query, change the flex direction to vertical for the container.

w) Adjust any other properties as needed to create a visually appealing layout for smaller screens.

x) Open the "flexbox.html" file in a web browser to view the initial layout.

y) Test your layout by resizing the browser window to ensure it responds correctly.

z) Add any additional styles or enhancements to make the layout visually appealing.

### Part III: Submitting your Work

Using Bright Space, please submit a link to the lab2.html web page on your Algonquin College web space.

When your code is complete, make sure you zip up all the files (lab2.zip) created in this lab including a copy of any images that you have used in your growing website.

* Your modified version of the starter code for part 1.
* Your css stylesheet file (of Part I and Part 2)
* Screenshots of both HTML and CSS validation showing successful validation of code. ( Part I and Part 2)
* all images used

### Other Important Requirements

* Demo and justify your work and provide correct answer to professor’s questions.
* The work will be graded zero if you do not demo it on time, even if uploaded
* Part 1,2 are out of 6 points each.
* Indentation and documentations both HTML and CSS parts for 2 points out of 6

Note that 30% of your final grade comes from the grades you obtained from your labs and assignments.