CMPS 356 Enterprise Application Development
Lab 3 – CSS Fundamentals P-2

Objective

By the end of the lab you should be able to use:
- ✓ Grid layout, flexbox and media queries to develop responsive web page layouts
- ✓ Font Awesome toolkit to create different icons
- ✓ Use bootstrap framework to style your webs page layout and forms**.**

Overview

This lab has 4 parts:
- **Part A** Tutorial and In-Lab Exercise
  - o use font awesome to create awesome icons
  - o develop a responsive page layout using grid, flexbox and media queries system.
  - o use the Bootstrap framework to style your page icons and forms
- **PART B,** develop a complete website using the concepts that you practiced in part **A.**

First sync your cmps356 repository using **GitHub Desktop** Client. Create two subfolder named "**Lab3-CSS-Tutorial and Lab3-QBookDonation-Solution**" inside the local working folder of your repository. At the end of the Lab, complete the testing document then sync your repo to push your work to GitHub.

Part A-Tutorial
Grid, Flexbox, Bootstrap and FontAwsome

1. Get the **Lab3-CSS-2** repo from https://github.com/cmps356s19/cmps356-content/tree/master/Labs repo and copy it to your private repository
2. Open the "**Lab3-CSS-Tutorial folder** under the **Lab3-CSS-2** in **Webstorm**.
3. The project has two HTML pages. One called **index.html** and another one called "**registration.html**". You will style the two files using grid, bootstrap, font-awesome and flexbox.
4. Do the following four exercises covering each of the four concepts mentioned above.

Font Awesome 5 Icons

Exercise 1. Add the following icons to the footer of the **index.html** page

To do this

    a. First add the font awesome CDN to your web-page

```html
<link rel="stylesheet" href="https://use.fontawesome.com/releases
/v5.7.0/css/all.css"integrity="sha384-
lZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+A56Ebc1zFSJ"
crossorigin="anonymous">
```

    b. Check the following link to find the classes for the above icons
https://www.w3schools.com/icons/fontawesome_icons_text.asp

Grid Layout Cont..

In this part we will continue exploring the grid layout and discover the following concepts in depth.

- Container level
  - Display
    - Inline-grid
    - grid
  - Template
    - grid-template-columns
    - grid-template-rows
    - grid-template-areas
  - Gap, Aligning item and align-content(column), justifying content (row)
  - And many more concepts mentioned under http://grid.malven.co
- Children level
  - Grid-row
  - Grid-column

1. First visit this website https://www.w3schools.com/css/css_grid.asp .
2. Let us go through the tutorials and understand how the grid system works
3. Try to recreate the following layouts using grid layout . You should use the index.html provided in the lab folder.
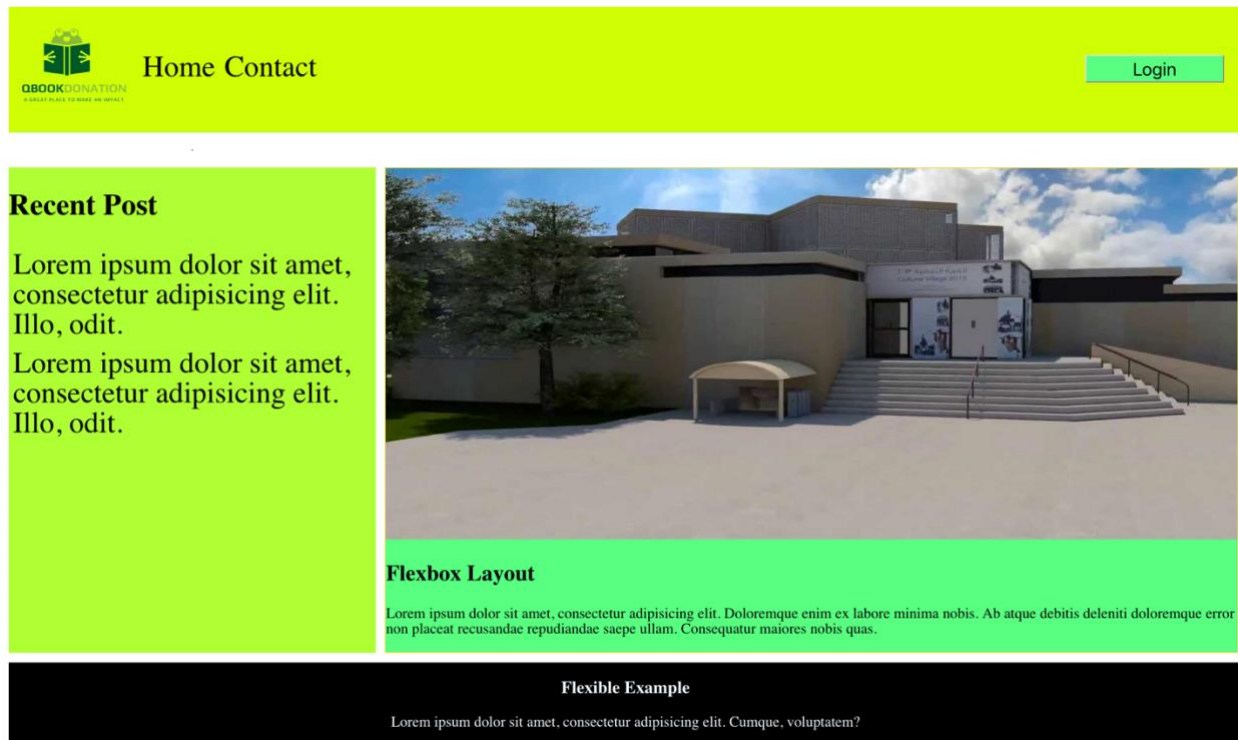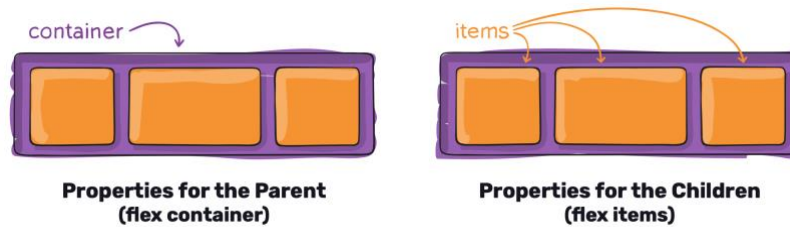

Figure 1 : Flexible Layout Design

For More tutorial on Grid Layour check out this link https://css-tricks.com/snippets/css/complete-guide-grid/
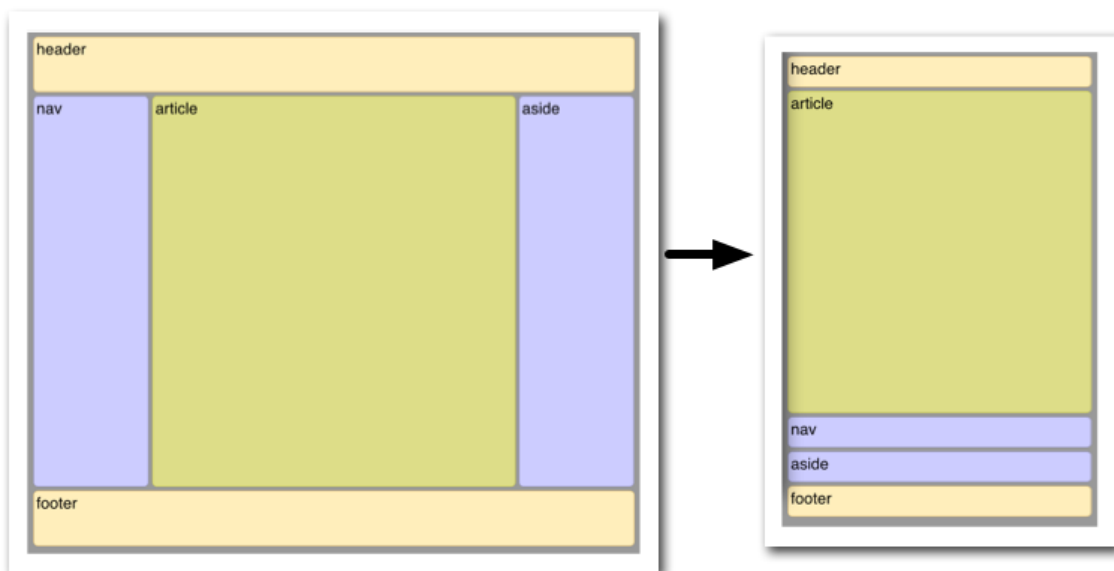
CSS Flexbox

The main idea behind the flex layout is to give the container the ability to alter its items' width/height (and order) to best fill the available space (mostly to accommodate to all kind of display devices and screen sizes). A flex container expands items to fill available free space, or shrinks them to prevent overflow.

Properties for the Parent
(flex container)

Properties for the Children
(flex items)

Difference between Flexbox and Grid

Flexbox layout is most appropriate to the components of an application, and small-scale layouts, while the Grid layout is intended for larger scale layouts.

1. Click on the following link to start the tutorial
   https://www.w3schools.com/css/css3_flexbox.asp
2. Re-Create the following layout using Flexbox and GridLayout. Also make sure you use media queries to handle the screen change and adapt your layout accordingly

<u>Bootstrap</u>

In this lab we will introduce the most popular HTML, CSS, and JavaScript framework for developing **responsive**, **mobile-first**

websites called bootstrap. In the lab we will only introduce the bootstrap components and style classes.

However, we will not cover the bootstrap grid system as we already have the built in CSS grid , flex-box and media queries to handle responsiveness.

1. Add the Bootstrap CDN to the registration.html page that is shared with you. **<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css">**

2. Visit the following website https://getbootstrap.com/docs/3.3/components/

3. Go to the input group and try the different form styling classes