

COMS/SE 319: Construction of User Interface Fall2022

LAB Activity 2 – Constructing web UI with Bootstrap

In this lab activity, you will learn to use Bootstrap for designing web pages. Bootstrap is a front-end web UI framework that comes with css and js files. In this tutorial, you will learn how to add bootstrap to your html files and use the various components provided within.

- Adding Bootstrap
- Understanding 12 column layout
- Using components
 - Navbar
 - Sidebar
 - Form
- Working with icons
- Ripping implicit features

Prerequisites:

- Chrome Browser
- Text editor (eg. **VSCode**, Sublime, Notepad...)
 - We recommend using VSCode

Additional tutorial:

- <https://getbootstrap.com/docs/>

Task 1: Adding bootstrap to your project

- Create an “index.html” file with <head> and <body> tags

```
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

- Copy “bootstrap.min.js” and “bootstrap.min.css” from the [Bootstrap docs](#), to your html file.
- Copy the two lines (<link>, <script>), into the <head> tag of your

```
<html>
  <head>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-gayn36969wDm3J5pT91Cw11uq3W2y2z736h483913Ww6Xgbs7496/wyFY/qYT1wP+X67Cg" crossorigin="anonymous">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-3p4TBWnCdh44WjZB3v2a9G5d1e5Kee6dZ1vCwUwK278EEaB6gQv+VLDgI85N" crossorigin="anonymous"></script>
  </head>
  <body>
  </body>
</html>
```

- There are several ways to install or add Bootstrap, but this is the quickest!
- Note you are linking your file to remote “.css” and “.js” scripts. But you can also download the two scripts and change the “href” and “src” properties to point locally.

Task 2: Understanding the Bootstrap 12-column layout

The first thing you can notice is the change in font. Add Header `<h1>` tags and checkout the change in font. That's because bootstrap uses its own fonts such as "Roboto". But fonts are just the beginning, Bootstrap has an entire layout system, components, and scripts ready made to use.

```
<html>
  <head>

  </head>
  <body>
    <h1> Hello World! </h1>
  </body>
</html>
```

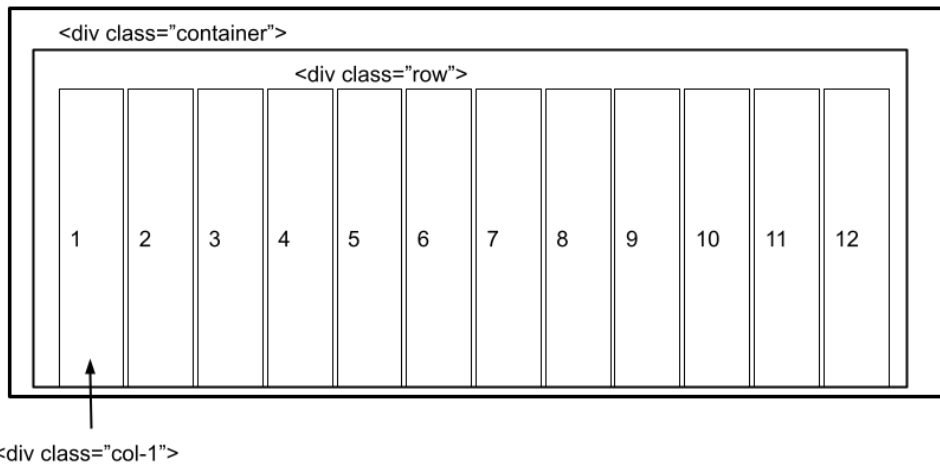
Hello World!

```
<html>
  <head>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" rel="script">
  </head>
  <body>
    <h1> Hello World! </h1>
  </body>
</html>
```

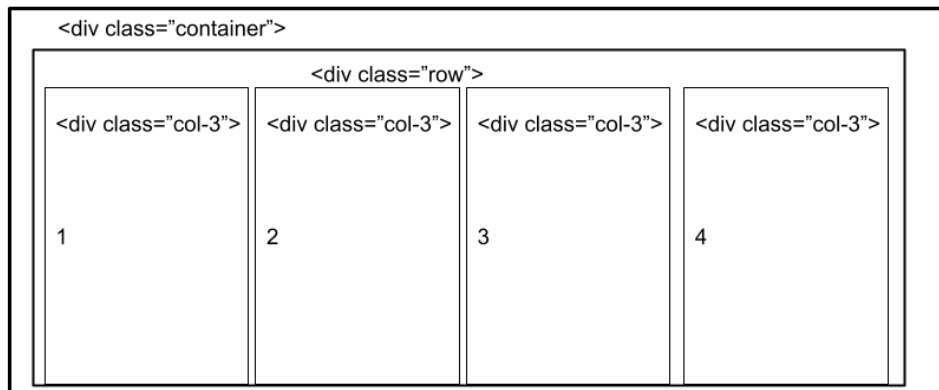
Hello World!

- Bootstrap cascades three classes to create its layout.
 - **Container** class: One per page
 - **Row** class: Defines an entire row
 - **Col** class: Defines columns
- The best way to design a page is to have a div with class container, `<div class="container">`, and one or more divs with class row, `<div class="row">`. Then each "row" div can be divided by up to 12 "col-1" divs, `<div class="col-1">`.
- A key point to remember here is that each "row" div can be divided by "col" divs that sum up to 12.

- For instance, the picture below shows a row divided by 12 “col-1” divs. That is, each column will have a width of size 1.



- Similarly, we could also layout the page such that the row is divided into 4 equally sized divs of size 3. I.e. four “col-3” divs.



- In both cases notice that the “col” div sizes sum up to 12. Hence the 12 column design.
- It is important to note that the col sizes need not be similar, we could design a page with one “col-3” column and a wider “col-5” column to its right. We can then sandwich them between two “col-2” divs.
- Lets go ahead and do that. You can add custom styles to see the borders.

```

<html>
  <head>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js" rel="script">
  </head>
  <body>
    <div class="container" style="border: 2px solid black;">
      <div class="row">
        <div class="col-2" style="background-color: #bisque;"></div>
        <div class="col-3">
          <h2>Sidebar area</h2>
        </div>
        <div class="col-5">
          <h1>Main content</h1>
        </div>
        <div class="col-2" style="background-color: #bisque;"></div>
      </div>
    </div>
  </body>
</html>

```

- Output should look like this.

	Sidebar area	Main content	
--	--------------	--------------	--

- Checkout the [Bootstrap layouts documentation](https://getbootstrap.com/docs/5.2/layout/containers/) for more information.

Task 3: Utilizing readymade components

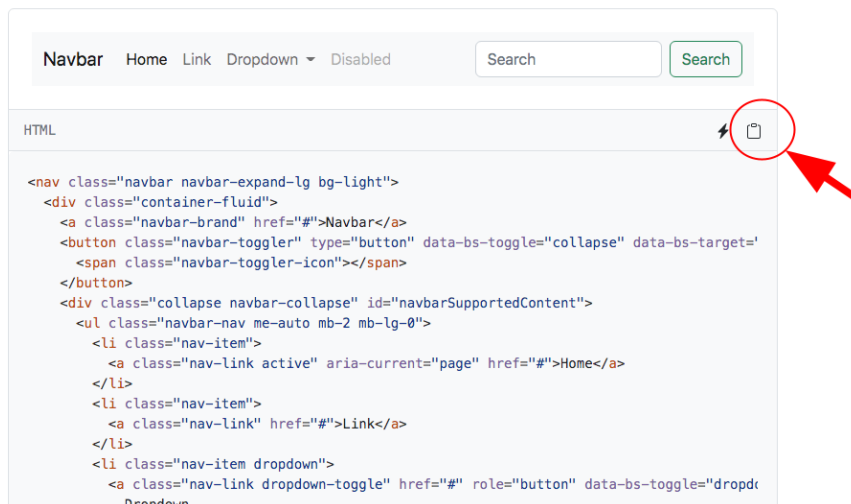
Bootstrap offers a catalogue of readymade components. These components include navbars, forms, buttons, inputs, carousals and much more. This is what makes Bootstrap very suitable for rapidly developing web-based UI applications. It comes in handy especially when designing back-end web applications as opposed to front-end ad websites that might require lots of custom graphics designing. It is also important to note that we can use other styles alongside bootstrap. But when it comes to designing friendly and consistent web application UI, Bootstrap provides sufficient components for the most part.

Navbar

- Navbars allow us to put important links in the top section of our page
 - These links usually lead elsewhere to another page.
- We could also add search bars and buttons.
- Bootstrap provides several navbars, so lets go ahead and copy one into our “main content” div. I.e. `<div class="col-5">`
- Bootstrap makes it easy to yank components of its page. [Go to Navbar](#), just click on the copy icon in the underlying code snippet under the component. Simply paste it where you want and voila!

- `.navbar-text` for adding vertically centered strings of text.
- `.collapse.navbar-collapse` for grouping and hiding navbar contents by a parent breakpoint.
- Add an optional `.navbar-scroll` to set a `max-height` and [scroll expanded navbar content](#).

Here's an example of all the sub-components included in a responsive light-themed navbar that automatically collapses at the `lg` (large) breakpoint.



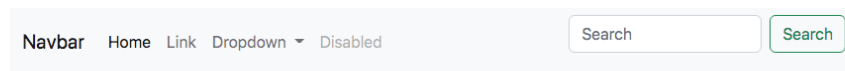
The screenshot shows a Bootstrap navbar component with the following HTML code snippet:

```
<nav class="navbar navbar-expand-lg bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropd
            Dropdown
          </a>
          <div class="dropdown-menu">
            <a href="#">Dropdown item 1</a>
            <a href="#">Dropdown item 2</a>
          </div>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

- After removing the custom styles, resizing col sizes in the previous layout, and pasting the navbar in the “main section”, you should get this output.
- **Note: the Navbar is wrapped/minimized on my editor.** It’s about 36 lines of code.

```
<html>
  <head>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js" rel="script">
  </head>
  <body>
    <div class="container" >
      <div class="row">
        <div class="col-1" ></div>
        <div class="col-2">
          </div>
        <div class="col-8">
          <nav class="navbar navbar-expand-lg bg-light">...
          </nav>
        </div>
        <div class="col-1"></div>
      </div>
    </div>
  </body>
</html>
```

- Remember to adjust the “col” sizes, and you should see the output below.



Sidebar

- Lets use a “list group” as a sidebar in our page
- Grab a “list group” from [here](#)

```
<html>
  <head>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/boo
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/boo
  </head>
  <body>

    <div class="container" >

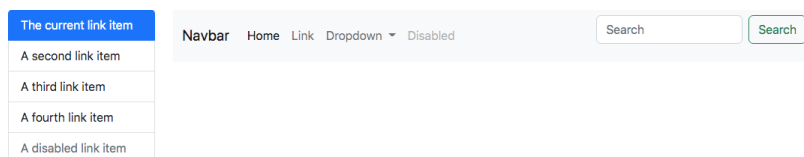
      <div class="row">
        <div class="col-1" ></div>

        <div class="col-2">
          <div class="list-group">...
        </div>

        <div class="col-8">
          <nav class="navbar navbar-expand-lg bg-light">...
        </div>

        <div class="col-1"></div>
      </div>
    </div>
  </body>
</html>
```

- You should see this output



Forms

- Grab a form from [here](#), and paste it in the “main section”. With minor edits you should see something like this.

The image shows a Bootstrap form layout. On the left is a sidebar with a list of link items: 'The current link item' (highlighted in blue), 'A second link item', 'A third link item', 'A fourth link item', and 'A disabled link item'. The main content area features a light gray header with a 'Navbar' containing 'Home', 'Link', 'Dropdown' (with a dropdown arrow), and 'Disabled'. To the right of the navbar is a search bar with a 'Search' button. Below the header, the form includes fields for 'Email' and 'Password', an 'Address' field with the value '1234 Main St', an 'Address 2' field with the placeholder 'Apartment, studio, or floor', and fields for 'City', 'State' (a dropdown menu with 'Choose...' selected), and 'Zip'. At the bottom left of the form is a checkbox labeled 'Check me out' and a blue 'Order' button.

- Edit the names and links at will.

Its quite important to thoroughly checkout the [docs](#), before starting a bootstrap based project. Knowing the components and understanding the various CSS classes at use will save you plenty of time when designing your applications.

Task 4: Icons

[Bootstrap icons](#) require separate installation or linking. Since we’re still not using NodeJS, lets go ahead and link the file just like we did with “bootstrap.min.css”. Go to the [install section](#) on the icons page, and grab the css link. Then include it in your <head> tag.

```
<html>
<head>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity=
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-A3r
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.1/font/bootstrap-icons.css">
</head>
<body>
```

Once, we've linked the css file, we can now use the icons through several means. An easy way could be using icon fonts. Simply add the `<i>` tag with the correct class property. The surround the `<i>` tag by the element we want the icon to appear in. Eg. A button, or a label.

- In the previous page, lets add a small '\$' icon on the "Order" button to signify a transaction.

```
</div>
<div class="col-12">
  <button type="submit" class="btn btn-primary"> <i class="bi-currency-dollar"></i> Order</button>
</div>
</form>
```

- Notice the '\$' icon inside the "Order" button.

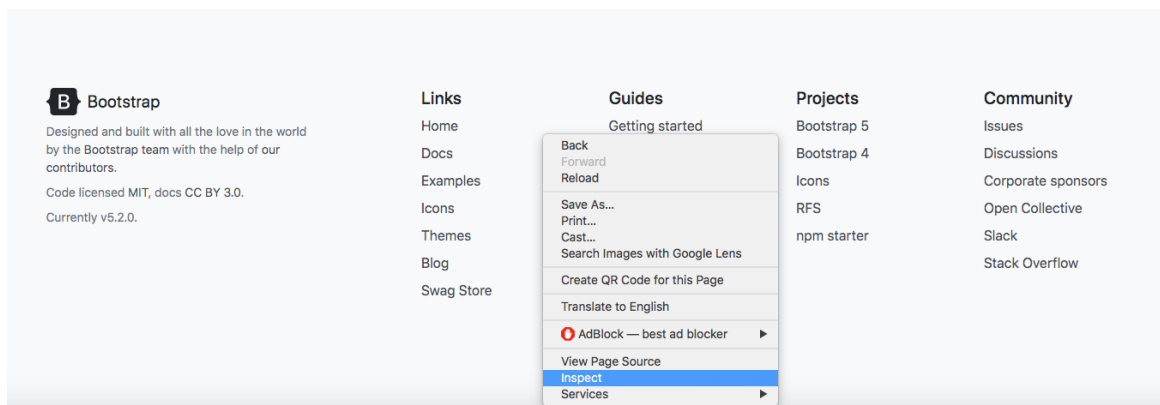
The current link item	Navbar Home Link Dropdown ▾ Disabled			Search	Search
A second link item	Email		Password		
A third link item	<input type="text"/>		<input type="password"/>		
A fourth link item	Address				
A disabled link item	<input type="text" value="1234 Main St"/>				
	Address 2				
	<input type="text" value="Apartment, studio, or floor"/>				
	City	State	Zip		
	<input type="text"/>	Choose... ▾	<input type="text"/>		
	<input type="checkbox"/> Check me out				
	<input type="button" value="\$ Order"/>				

- Avoid overusing or misusing icons.

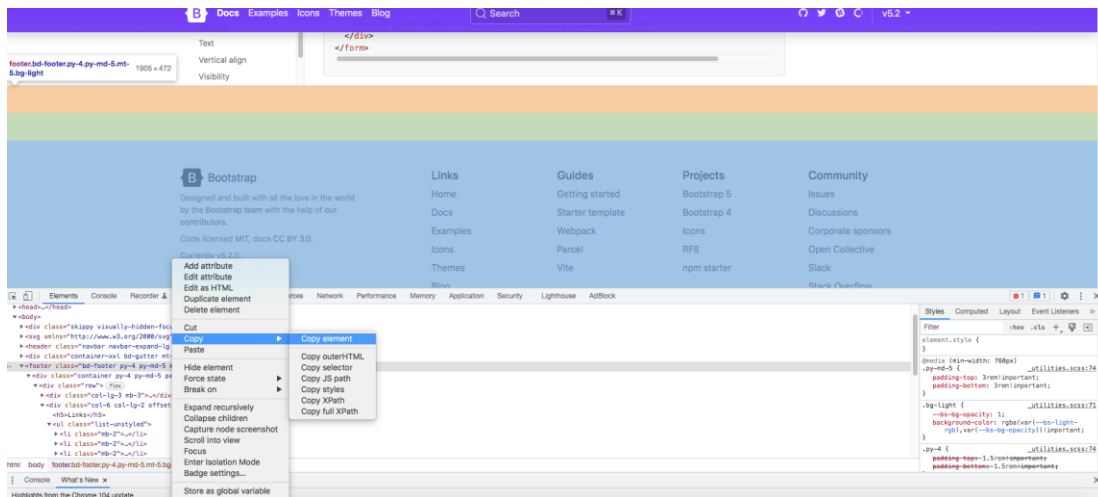
Task 4: Ripping components

Sometimes, the component we want might not be presented in the docs. Ironically however, it might be used in the docs page itself. A good example might be the Footer of the Bootstrap page. If we want to use a similar Footer in our app, we will have to rip it off the docs page. Here is where Chrome and its cool inspection tool comes in handy. By inspecting the page source itself, we can grab/copy the footer and paste it into our project.

- Go to the Bootstrap [docs page](#).
- Scroll down all the way to the bottom
- Assuming you're on Chrome
 - Right click on the Footer and click on "Inspect"



- In the inspection window, you should be on the "element" tab.
- Whenever you hover over any element, you should see your Chrome browser highlighting the selected elements in real-time.



- Right click the “Footer” element, hover over to “copy” and click the “copy element” option.
- You can now paste it inside your “main section”, or inside “row” or even outside “container”. However, I’m pasting it in the main section. This gives the page a narrower, seemingly resizable design.

The current link item

A second link item

A third link item

A fourth link item

A disabled link item

Navbar Home Link Dropdown Disabled Search

Email Password

Address 1234 Main St

Address 2 Apartment, studio, or floor

City State Zip Choose...

☐ Check me out

\$ Order

Bootstrap

Designed and built with all the love in the world by the [Bootstrap team](#) with the help of [our contributors](#).

Code licensed [MIT](#), docs [CC BY 3.0](#).

Currently v5.2.0.

Links

[Home](#)
[Docs](#)
[Examples](#)
[Icons](#)
[Themes](#)
[Blog](#)
[Swag Store](#)

Guides

[Getting started](#)
[Starter template](#)
[Webpack](#)
[Parcel](#)
[Vite](#)

Projects

[Bootstrap 5](#)
[Bootstrap 4](#)
[Icons](#)
[RFS](#)
[npm starter](#)

Community

[Issues](#)
[Discussions](#)
[Corporate sponsors](#)
[Open Collective](#)
[Slack](#)
[Stack Overflow](#)

Summary

This was a brief tutorial on how to incorporate Bootstrap components to enhance your web app's UI. There are several ways to install Bootstrap. In this tutorial, we simply linked remote css and js scripts. We also linked the Bootstrap Icons script, which is a separate package. We learned about the 12-column layout, and adding Bootstrap components to our app. We also learned how to use the Chrome inspection tool to rip elements.

It is important to note that these lab exercises are by no means comprehensive. They only serve as an introduction by providing tips and snippets on how to develop your apps. It is up to you to read and experiment further in order to master UI development. In light of this fact, here are some things worth exploring for your upcoming homework and your portfolio project.

- Keep experimenting with copying different Bootstrap components to see how they look and work in your page.
- Checkout the Icons page, it has nice images to enhance your UI
- When it comes to web development, the Chrome inspection tool is your best friend.
 - Checkout the different tabs and windows provided in the inspection tool
 - We will revisit it later when working with Javascript

Now make sure to attend Friday Lab since Quiz 2 will be posted in class!