Bootstrap should be downloaded through a source file (SASS Files) in order to modify the classes.

html tag should always have language set to “en”

head should have 3 meta tags:

1. <meta charset = “UTF-8”> - Recommended version of Unicode
2. <meta name = “viewport” content = “width = device-width, initial-scale=1.0”> - name and content as viewport and width in which width depends on device width and has an initial scale of 1.0
3. <meta http-equiv = “X-UA-Compatible” content = “IE=Edge”> - So that it will render properly in Internet Explorer based browsers, specifically edge.

Then, link the bootstrap cdn inside the head tag through a link tag (Copy-paste)

Also link the scripts for the body tag (Copy-paste)

Note: Make sure you’re using bootstrap 5 and Bundle codes

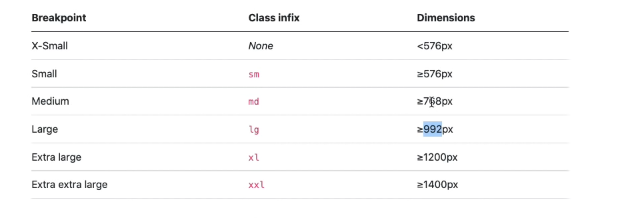
**Then always start with the navigation bar**

Code is : nav.navbar.navbar-expand.lg.bg-dark.navbar-dark

Which turns:

<nav class="navbar navbar-expand lg bg-dark navbar-dark">

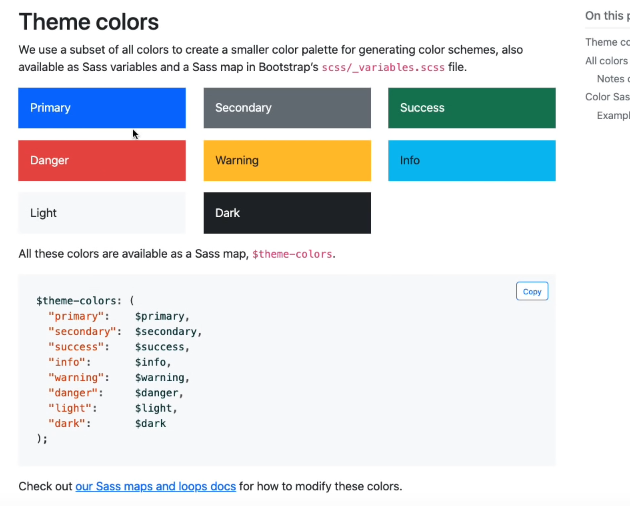
(This is also responsive for showing a hamburger menu lower than the given list and codes given):



The recommended is lg

Then you can also set the background through bg

Here are the list of backgrounds:



Redarking the navbar makes it lighter for readability since the background is already dark

Then putting the logo:

Code for putting logo – a.navbar-brand

Which turns into: <a href="" class="navbar-brand">

Adding a div container makes the website more organized and the navbar-brand will have space, spaced at the right side a bit

It should be inside the navbar created but under it is the navbar-brand

Inside the div container and the Front End Design Link, there should be **a .collapse.navbar-collapse** div in order for the links to disappear in smaller screens (and just display the hamburger icon). And those links will only appear for larger screens

Next you’ll need the ul.navbar-nav inside the collapse div (might be the list under the hamburger icon

Under the ul, there will be li.nav-item – where every list has a nav item

Under or for each of the li., there should be a navigation link: a.nav-link

Then name each of the link along with the list (copy paste)

Then make the hamburger icon, after the link after the container div and create a button for the hamburger icon using button.navbar-toggler

It has a button type, a bs-data-toggle = “collapse”, then it’s target is: bs-data-target = “#navmenu” (I think this is for the action, specifically for the collapse class)

Then inside the button tag, put: span.navbar-toggler-icon (For the hamburger menu’s icon/visual)

NOTE: The whole button itself targets the navmenu or the div, that is why it shows the div that holds all the lists that we created when clicking the hamburger icon

**NEXT IS THE SHOWCASE SECTION**

We start off with the format and a section: (pt-x), with x being 1-5 and is known as padding top, or (pb-x), with x being 1-5 and is known as padding bot, or (px/y-x), with x being 1-5 and is known as padding x/y axis, or (p-x), with x being 1-5 and is known as padding on ALL SIDES

section.bg-dark.text-light.p-5.text-center

Then as usual we put a container and type: .container

Divs will be inside the container to put all the elements (One for the image, one for the other content)

Adding a class to a div that contains both div (H1 and image) and if that class is named d-flex, it will turn those elements into a single row (The one after the container div)

Adding a class (img-fluid) to an image inside the container so that it will properly be bound to the container

As for the size, you can use css or w-50 inside the class along with the img-fluid

Flexbox looks bad on small screens, so we gotta have a solution in which we do:

Class = “d-flex” means “use flex on all sizes”

But we can add a breakpoints or breakpoint size classes in which we add:

d-sm-flex, where it will change the format to up and down on small sizes or once it reaches that point

You can use spans in order to change the color using classes with the color themes:

span.text-warning

span.text-info

The paragraph and the button is in the same div as the h1 tag

The “lead” class in paragraphs allows texts to be bigger

Then a button is created with blue color with large size, inputting:

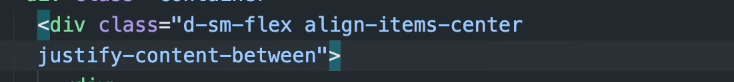
button.btn.btn-primary.btn-lg

adding my-4 to the lead class will allow that paragraph to have a margin in the y axis with a level of 4 for proper margin position purposes, it is called a margin class

Seeing the button or any element at the center at mobile screens or small screens in the center, but left at pc or larger screens, you’ll need to add a breakpoint in the class’ section which is text-sm-start. So at the start of the pc/in the left side of the screen the button will appear, but in the small screens or breakpoint it will not be in the start anymore (Also includes other texts on the div with the h1)

d-none will not display the image d-sm-block will only not show the image at small screens only, but will show at larger screens (should be inputted in the class of the image)

Now, to put the position of the y axis of the div with the h1 to the center maintaining we’ll need to use FLEXBOX, we can either use css or follow this:



Justify content simply justifies the whole div and aligin items center aligned the div to the center in the remaining y axis

**NEWSLETTER**

Start off with a section with a primary background color and a light text

section.bg-primary.text-light-p-5

NOTE: Padding is important since it is what creates the background and make it bigger

Next is to create a medium flexbox with and adjust the css justify content between as well as align items center

.d-md-flex (then css)

Next is the h3 tag with a margin bottom padding of 3 and for medium screens, padding will be disabled and be typed as:

h3.mb-3.mb-md-0

NOTE: The padding on top of the section can be adjusted in specific sizes using pt-lg-1 to 5 or other sections as well

(This can be seen in the “Forms” in the “input group” in getbootstrap.com [Bootstrap5])

<div class="input-group mb-3">

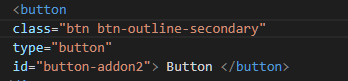
<input type="text" class="form-control" placeholder="Recipient's username" aria-label="Recipient's username" aria-describedby="button-addon2">

<button class="btn btn-outline-secondary" type="button" id="button-addon2">Button</button>

</div>

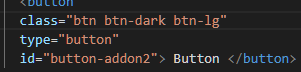
Copy paste this for the newsletter after the h3

Then we will modify the visuals of this button



Change the class = outline-secondary to dark to make it dark, as well as the size of the btn which will also affect the size of the input because it is a group

End Result:



As for the input

Change the placeholder text

Remove the addon and the aria

**Changing the input field size**

It can only be done through css, so add “news-input” or any name beside the “input-group” class

**CSS Input group**

At this point, you’ll need to only make it change at the medium breakpoint, so we type:

@media (min-width: 768px)

Where 768 pixels is for medium screens

And inside it we put news input or the div that holds the input group with a class called news-input or before the input tag

And inside the news input class inside that medium breakpoint, put a width of 50%

This looks better because once you go to medium screen, it doesn’t look good with a 100% of the input group being displayed

**Boxes (AKA Grid in CSS)**

Create a section as well for this one, with the padding everywhere at level 5 for space between the previous section:

section.p-5

then create the container

.container

TO USE THE GRID, we need to use a class of “row”:

.row

Then it’s now possible to create the columns in a single row

.col

.col

.col

Then put some dummy texts on the columns

To make the columns aligned at the center, add a css property of text aligning the class of row to the center or just do it in the tag itself, as it is bootstrap

As for mobile screens or smaller screens, you’ll just need a breakpoint to add after the column:

col-md (put it in the tag instead in the CSS)

What makes the boxes visuals itself is called cards, in which we would check the getbootstrap.com for references. It is located inside every div column-md.

The card has a dark background, with light text:

.card.bg-dark.text-light

Along with its body with formatting:

.card-body.text-center

You should also text-center the class row, just right after the container is created

We can also add bootstrap icons by downloading it or using a cdn (I downloaded using the css files inside the folder it has)

An example that we’ll use for creating the first icon which is inside the card:

i.bi.bi-laptop

Having a bigger size of the icon, since it is a font – you’ll need to use h1 or the other heading tags since it is for texts, SO make a div outside the icon tag and wrap it with the resizing h1 div class, you can use div class instead of using an h1:

.h1.mb-3 (This is the only h1 that uses the div class for now)

All of these Elements inside the card should have a margin in the bottom, recommended is level 3:

This includes the icon, the card title (h3 tag), and the card text(p tag)

Card Title Creation:

h3.card-title.mb-3

Card Text Creation:

p.card-text.mb-3

If you want to create a paragraph that is shortened, use lorem10/up then enter

Card images also use the same format

Then create the link for “READ MORE” below the paragraph:

a.btn.btn-primary

Then do the other column (just copy paste starting from card bg)

Change the TEXT of the OTHER TWO as well as ICONS, as well as the button color, and background color

**LEARN THE FUNDAMENTALS (USES 2 COLUMNS, TEXT RIGHT SIDE, IMAGE LEFT SIDE**)

We will have a section with a usual padding of 5 on all sides for spacing, and an ID when the user clicks learn, so it has the same ID as “learn”

section.p-5, and add id = “learn”

Add a container as usual:

.container

Then add a row:

.row

Also add an ID for the row for CSS since

align items and justify content between will be used for aligning the items inside the row and justifying it as the format

For the boxes, or the ones that uses cards, in small screens it will look like the y position has not been adjusted, but the solution for the is to add gap which can be g-4, or gy-4 for only the vertical spacing/gap

**BOXES NOTE:**

Adding a gap in the row class will adjust the smaller screens to have a vertical gap, better if “gy-x” is used

**Line Hati**

First, in setting a dotted line in css, we first have to set it’s border to none so we can set the top border – dotted, red. Then set the color to white or #fff in order to see the dots

And also a background-color of #fff to make it more cleaner and clearer

Then set its height and width

**Back to “LEARN THE FUNDAMENTALS”**

Put inside the row some columns, we’re purely going to make use of rows and columns here

When you put first column, make sure to put an image with a class of img-fluid so that it’s formatted nicely where it stays in the container

Then make the second column with a padding of 5 for spacing purposes in medium screen

.col-md.p-5

Make the h2 tag and content

Make the paragraph with content (with a lead class for better large recommended font size)

Add a button using a tag with a light background and a top margin of level 3

a.btn.btn-light.mt-3

Then put the content “Read More” with a symbol of chevron

How to make a chevron icon: i.bi.bi-chevron-right

**React Section**

Same as last section, just need to modify the section’s background to dark for styling

Where p-5 bg-dark exists

Put the image at the right so put it after the text div

To see the text, it needs to be light colored so add text-light to the section right after bg-dark

**Questions / Accordion**

As usual, create a section with a padding of 5

You’re going to create an id for the section called “questions” in order to redirect this when you click on the questions in the navbar

Create a container as usual

Create an h2 text with centered text and margin bottom 4 for it to create a space on the bottom based on level which gives a gap to the next accordion:

h2.text-center.mb-4

Make the accordion – A sample can be seen in getbootstrap.com, flush is a good example of the accordion since it’s the most basic one

Then change the content

First is the id = “accordionFlushExample” to id = “questions”

Then organize the items inside the accordion by looking at the item tags

Get rid of the accessibility tags to avoid confusion (Aria)

Remove the flush heading ids

Match the data bs target to the div where the bs parent is, exactly match the bs data target to the id of where the bs parent is:

(Found in button) data-bs-target = “#question-one”

(Found in the div with the data-bs-parent) id = “question-one”

Same with other accordion buttons and divs

**Note for Navbar:**

In order to make the navbar position fixed on the top, you just have to go for the first nav created at the top and add something to the class which is typed as: fixed-top

In order to fix the navbar on the top that’s not displaying the original top space, we’ll need to go to css and type:

body::before{

display:block;

content:’’;

height:60px;

}

What this does is add a ghost element from the top and pushes it down to the bottom in order to retain the original look

**INSTRUCTORS**

First thing that you need to do is to create a section with an instructor id for the navigation, with a class padding of 5 and a primary background:

section#instructor.p-5.bg-primary

As usual, a class container:

.container

Then create an h2 tag with a text-center and white text (white text is more clearer in primary than light:

h2.text-center.text-white

Then add a paragraph under the h2 tag, with a class of lead to make it bigger, as usual.

Also add text-center and text-white for formatting:

p.lead.text-center.text-white

Make a row class for the alignment of the instructor cards with columns inside

Starting with the row class:

.row.g-4

g-4 is for the level 4 gap for the row

For the columns, make the class columns in the medium screens to take half the space only so if you put another column and has level 6 as well in medium screens, only 2 columns can be seen there:

.col-md-6

As for large screens, it should take 3 of the spacing, since we want 4 columns in large screens:

.col-md-6.col-lg-3

What we want here is to add the card just like in the previous cards we used:

Since the background is primary, we need a div with a class of card with a light background.

.card.bg-light

Under the card, we add the formatting of the card class div through another div:

.card-body.text-center

Then we add the image, we can generate pictures of random people using an API ( through this link):

<https://randomuser.me/api/portraits/men/(number.jpg)>

OR

https://randomuser.me/api/portraits/women/(number.jpg)

Then add the class for the image format with a bottom margin of level 3:

class = “rounded-circle mb-3”

After that, add a tag of h3 with a class of card-title and also a level 3 bottom margin – this exactly puts a name or caption at the bottom at the h3 font size:

h3.card-title.mb-3

As for the paragraph, it doesn’t need any space formats, just a card-text class:

p.card-text

Social media links are the ones below the paragraph and will be constructed by first having an a tag and an href inside:

a.

This will be done with an icon inside the a tag

The icon is a twitter logo with the x margin having 1 level and dark text since the primary color is already blue (the background)

<a href=”#”> i.bi.bi-twitter.mx-1 </a>

COPY THE OTHER A TAGS AND MAKE IT OTHER SOCIAL MEDIA

Like facebook, Instagram, and tiktok

COPY AND PASTE THE WHOLE COLUMN AND MODIFY THE OTHERS’ PICTURES AS WELL AS NAMES

**CONTACT INFO AND MAPBOX**

Start with the usual section with padding 5:

section.p-5

Then next is a container div class:

.container

The usual row creation with a gutter of 4:

.row.g-4

Then create 2 columns for medium screens (without any value):

.col-md

Then inside the column, put an h2 tag with a class of the usual text-center and a new margin bottom level of 4 (Header for Contact Info):

h2.text-center.mb-4

Then create a list group without any symbol on the left side, it is created by the ul tag with a class of list group and a specific list group flush and lead for bigger font:

ul.list-group.list-group-flush.lead

Then create the li’s (Each li’s created are going to have a class of list-group-item):

li.list-group-item

Make a separate span inside the first li created with a font weight of bold inside the span class (also can be done through css) THIS IS FOR THE LEFT SIDE or LABEL FOR THE INFO:

span.fw-bold

THEN COPY PASTE AND MODIFY THE LI

THEN Create the 2nd column

In the second Column, you will create the mapbox, in which you will have to go **to maplibre.com and download the files and follow the cdn**

**It didn’t work for me, so I had to embed using this format:**

  <div id="map-container-google-1" class="z-depth-1-half map-container" style="height: 500px">

    <iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3860.418550672067!2d121.05468770091439!3d14.632166530173755!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x3397b79842be8df3%3A0x67997699eb77e0da!2sEast%20Kamias%2C%20Quezon%20City%2C%20Metro%20Manila!5e0!3m2!1sen!2sph!4v1678763170365!5m2!1sen!2sph" width="600" height="450" style="border:0;" allowfullscreen="" loading="lazy" referrerpolicy="no-referrer-when-downgrade"></iframe>

  </div>

**FOOTER**

A footer tag will be used instead of a section, with a padding of 5 as usual, and a dark background since it is a good mix of colors, as for the formatting, add text-center and text-white for visibility and positioning, and lastly POSITION RELATIVE, so that it will follow the “Go back up” inside the footer that is absolute so that it will absolutely go there:

footer.p-5.bg-dark.text-center.text-white.position-relative

next is the container class as usual:

.container

Next is to get a normal lead text through a paragraph tag to have the copyright text output inside the footer:

p.lead

Content: Copyright &copy; 2021 Front End Design

Take note that the &copy is a symbol

Now we will create the go back up link previously mentioned through an A TAG:

a.position-absolute.bottom-0.end-0.p-5 (It has a padding of 5, bottom-0 and end-0 will bring it back to the top along with absolute position because the footer is relative)

Next is to create the icon for the tag:

i.bi.bi-arrow-up-circle.h1 (h1 makes the icon bigger)

**SHOWCASE BUTTON**

Add attribute to the button for enroll button:

data-bs-toggle = “modal” –

data-bs-target = “#enroll”

data-bs-toggle - A modal will appear if it’s toggled/clicked ; type of UI that will appear when the button is clicked

data-bs-target = “” – Enroll is the specific id used for the modal (This will appear when you click the button)

You’ll have to choose a modal in getbootstrap.com, only the modal since you already set the toggle and the target

**PUT IT UNDER THE FOOTER FOR UNIFORMITY (IT CAN BE PLACED ANYWHERE TBH)**

Change all the ids’ from exampleModal to enroll, as well as the ones with labels

After that, change the body or content – the one with “…” three dots by putting first a paragraph with a class of lead for better size.

p.lead

Also modify the title

Then below the paragraph is a form tag only:

form

Inside the form, we will have a div with a class of mb-3, or level 3 bottom margin so it’ll have a gap between or space

We will create a label using a **label for** tag, with a class for forms

Then inside the div, there will be a **label for** tag, then it will have:

for = “first-name”

class = “col-form-label”

Under the label, there will be an input for forms:

input.form-control#first-name

Just copy the first one for the other inputs and labels/3 more

Don’t forget to modify as well as the button

THEN WE ARE DONE