

School *of* Coding

Pre Bootcamp Module for Absolute beginners.

This material is property of The Hacking School. Cannot be copied in part or full, reproduced in print or digital or any other form. The Hacking School deserves the right to track fair usage and prosecute misuse, if any.

Available on: **Bootcamp**
code.in

Introduction to Bootstrap 4+

Documentation : <https://getbootstrap.com/docs/4.5/getting-started/introduction/>

What is Bootstrap?

- Bootstrap is a free front-end framework for faster and easier web development
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
- Bootstrap also gives you the ability to easily **create responsive designs**
- Easy to use: Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
- Responsive features: Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- **Mobile-first approach:** In Bootstrap, mobile-first styles are part of the core framework
- Browser compatibility: Bootstrap 4 is compatible with all modern browsers (Chrome, Firefox, Internet Explorer 10+, Edge, Safari, and Opera)

How to Install Bootstrap

Method 1 Download from <https://getbootstrap.com/docs/4.5/getting-started/download/>

Method 2 CDN

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYYxFfc+NcPb1dKGj7Sk"
crossorigin="anonymous">

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
integrity="sha384-0gVRvuATP1z7JjHLku0U7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
crossorigin="anonymous"></script>
```

Exercise 1

Create a Bootstrap based HTML web page demonstrating **container** and **container-fluid**

Solution

Step 1 Prepare HTML5 layout with viewport meta tag

Step 2 Implement bootstrap container

Step 1 Code

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>
</head>
</html>

```

Bootstrap 4 is mobile-first

Bootstrap 4 is designed to be responsive to mobile devices. Mobile-first styles are part of the core framework.

To ensure proper rendering and touch zooming, add

```

<meta name="viewport" content="width=device-width, initial-scale=1">

```

Step 2 Code

```

<body>

<div class="container">
  <h1>My First Bootstrap Page</h1>
  <p>This part is inside a .container class.</p>
  <p>The .container class provides a responsive fixed width container.</p>
</div>

</body>

```

Note : Change the class container to container-fluid to observe the difference

container vs container-fluid

There are two container classes to choose from:

- The **.container** class provides a responsive fixed width container
- The **.container-fluid** class provides a full width container, spanning the entire width of the viewport

Container Padding

By default, containers have 15px left and right padding, with no top or bottom padding. Therefore, we often use spacing utilities, such as extra padding and margins to make them look even better. For example, **.pt-3 means "add a top padding of 16px"**:

```
<body>

<div class="container pt-3">
  <h1>My First Bootstrap Page</h1>
  <p>This part is inside a .container class.</p>
  <p>The .container class provides a responsive fixed width container.</p>
</div>

</body>
```

Container Border and Color

```
<div class="container p-3 my-3 border">
  <h1>My First Bootstrap Page</h1>
  <p>This container has a border and some extra padding and margins.</p>
</div>

<div class="container p-3 my-3 bg-dark text-white">
  <h1>My First Bootstrap Page</h1>
  <p>This container has a dark background color and a white text, and some extra padding and margins.</p>
</div>

<div class="container p-3 my-3 bg-primary text-white">
  <h1>My First Bootstrap Page</h1>
  <p>This container has a blue background color and a white text, and some extra padding and margins.</p>
</div>
```

Responsive Containers

- You can also use the **.container-sm|md|lg|xl** classes to create responsive containers.
- The max-width of the container will change on different screen sizes/viewports:

Class	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
.container-sm	100%	540px	720px	960px	1140px
.container-md	100%	100%	720px	960px	1140px

.container-lg	100%	100%	100%	960px	1140px
.container-xl	100%	100%	100%	100%	1140px

Code Example

```
<body>

  <div class="container">
    <h1>Responsive Containers</h1>
    <p>Resize the browser window to see the effect.</p>
  </div>

  <div class="container-sm border">.container-sm</div>
  <div class="container-md mt-3 border">.container-md</div>
  <div class="container-lg mt-3 border">.container-lg</div>
  <div class="container-xl mt-3 border">.container-xl</div>

</body>
```

Bootstrap Grid System

- Bootstrap's grid system is built with flexbox and allows up to 12 columns across the page.
- If you do not want to use all 12 columns individually, you can group the columns together to create wider columns:

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

- The grid system is responsive, and the columns will re-arrange automatically depending on the screen size.
- Make sure that the sum adds up to 12 or fewer (it is not required that you use all 12 available columns).

Grid Classes

The Bootstrap 4 grid system has five classes:

- **.col-** (extra small devices - screen width less than 576px)
- **.col-sm-** (small devices - screen width equal to or greater than 576px)

- **.col-md-** (medium devices - screen width equal to or greater than 768px)
- **.col-lg-** (large devices - screen width equal to or greater than 992px)
- **.col-xl-** (xlarge devices - screen width equal to or greater than 1200px)

The classes above can be combined to create more dynamic and flexible layouts.

Syntax of Bootstrap4 Grid

Example 1 : Syntax of Bootstrap4 Grid

```
<body>
<!-- Control the column width, and how they should appear on different devices -->
  <div class="row">
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
  </div>
  <div class="row">
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
  </div>
</body>
```

Explanation of above code snippet

- create a row (<div class="row">).
- Then, add the desired number of columns (tags with appropriate .col-*-* classes).
- The first star (*) represents the responsiveness: **sm, md, lg or xl**,
- while the second star represents a number, which should add up to 12 for each row.

Example 2 : Syntax of Bootstrap4 Grid

```
<body>
<!-- Let Bootstrap automatically handle the layout -->
  <div class="row">
    <div class="col"></div>
    <div class="col"></div>
    <div class="col"></div>
  </div>
</body>
```

Explanation of above code snippet

- Instead of adding a number to each col, let bootstrap handle the layout, to create equal width columns: two "col" elements = 50% width to each col. three cols = 33.33% width to each col. four cols = 25% width, etc.
- You can also use **.col-sm|md|lg|xl** to make the columns responsive.

Grid Exercise 1

Create three equal-width columns, on all devices and screen widths.

Solution :

```
<body>
  <div class="container-fluid">
    <h1>Three equal width columns</h1>
    <p>Add a new div with class="col" inside the row class - this will create three
equal-width columns.
    </p>
    <div class="row">
      <div class="col" style="background-color:lavender;">.col</div>
      <div class="col" style="background-color:orange;">.col</div>
      <div class="col" style="background-color:lavender;">.col</div>
    </div>
  </div>
</body>
```

Grid Exercise 2

Create four equal-width columns starting at tablets and scaling to extra large desktops.

Note : On mobile phones or screens that are less than 576px wide, the columns should automatically stack on top of each other

Solution :

```
<body>
  <div class="container-fluid">
    <h1>Responsive Columns</h1>
    <p>Resize the browser window to see the effect.</p>
    <p>The columns will automatically stack on top of each other when the screen is less
than 576px wide.</p>
    <div class="row">
      <div class="col-sm-3" style="background-color:lavender;">.col-sm-3</div>
      <div class="col-sm-3" style="background-color:lavenderblush;">.col-sm-3</div>
      <div class="col-sm-3" style="background-color:lavender;">.col-sm-3</div>
      <div class="col-sm-3" style="background-color:lavenderblush;">.col-sm-3</div>
    </div>
  </div>
</body>
```

Grid Exercise 3

Create two any Unequal Responsive Columns

Solution :

```
<body>
  <div class="container-fluid">
    <h1>Two Unequal Responsive Columns</h1>
    <p>Resize the browser window to see the effect.</p>
    <p>The columns will automatically stack on top of each other when the screen is less
than 576px wide.</p>
    <div class="row">
      <div class="col-sm-4" style="background-color:lavender;">.col-sm-4</div>
      <div class="col-sm-8" style="background-color:lavenderblush;">.col-sm-8</div>
    </div>
  </div>
</body>
```

Grid System Rules

- Rows must be placed within a **.container** (fixed-width) or **.container-fluid** (full-width) for proper alignment and padding
- Use rows to create horizontal groups of columns
- Content should be placed within columns, and only columns may be immediate children of rows
- Predefined classes like **.row** and **.col-sm-4** are available for quickly making grid layouts
- Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on **.rows**
- Grid columns are created by specifying the number of **12** available columns you wish to span. For example, three equal columns would use three **.col-sm-4**
- Column widths are in **percentage**, so they are always **fluid** and sized **relative** to their parent element
- Three elements with **.col-sm** will each automatically be 33.33% wide from the small breakpoint and up.

Example Code Snippet demonstrating the responsive columns

```
<body>
  <div class="container-fluid">
    <h1>Basic Grid Structure</h1>
    <p>Resize the browser window to see the effect.</p>
    <p>The first, second and third row will automatically stack on top of each other when
the screen is less than 576px wide.</p>
    <div class="container-fluid">
      <!-- Control the column width, and how they should appear on different devices -->
      <div class="row">
        <div class="col-sm-6" style="background-color:yellow;">50%</div>
        <div class="col-sm-6" style="background-color:orange;">50%</div>
      </div>
      <br>
      <div class="row">
        <div class="col-sm-4" style="background-color:yellow;">33.33%</div>
        <div class="col-sm-4" style="background-color:orange;">33.33%</div>
        <div class="col-sm-4" style="background-color:yellow;">33.33%</div>
      </div>
      <br>
      <!-- Bootstrap automatically handle the layout -->
      <div class="row">
        <div class="col-sm" style="background-color:yellow;">25%</div>
        <div class="col-sm" style="background-color:orange;">25%</div>
        <div class="col-sm" style="background-color:yellow;">25%</div>
        <div class="col-sm" style="background-color:orange;">25%</div>
      </div>
      <br>
      <div class="row">
        <div class="col" style="background-color:yellow;">25%</div>
        <div class="col" style="background-color:orange;">25%</div>
        <div class="col" style="background-color:yellow;">25%</div>
        <div class="col" style="background-color:orange;">25%</div>
      </div>
    </div>
  </div>
</body>
```

External Resources

<https://getbootstrap.com/docs/4.0/layout/grid/>

https://www.w3schools.com/bootstrap4/bootstrap_grid_examples.asp

Typography

Bootstrap 4 Default Settings

- Bootstrap 4 uses a default font-size of 16px, and its line-height is 1.5.
- The default font-family is "Helvetica Neue", Helvetica, Arial, sans-serif.
- In addition, all <p> elements have **margin-top: 0** and **margin-bottom: 1rem** (16px by default).

<h1> - <h6>

Bootstrap 4 styles HTML headings (<h1> to <h6>) with a bolder **font-weight** and an increased **font-size**

Example

```
<body>
  <div class="container">
    <h1>h1 Bootstrap heading (2.5rem = 40px)</h1>
    <h2>h2 Bootstrap heading (2rem = 32px)</h2>
    <h3>h3 Bootstrap heading (1.75rem = 28px)</h3>
    <h4>h4 Bootstrap heading (1.5rem = 24px)</h4>
    <h5>h5 Bootstrap heading (1.25rem = 20px)</h5>
    <h6>h6 Bootstrap heading (1rem = 16px)</h6>
  </div>
</body>
```

Display Headings

Display headings are used to stand out more than normal headings (larger font-size and lighter font-weight), and there are four classes to choose from: **.display-1**, **.display-2**, **.display-3**, **.display-4**

Example

```
<body>

  <div class="container">
    <h1>Display Headings</h1>
    <p>Display headings are used to stand out more than normal headings (larger font-size and lighter font-weight):</p>
    <h1 class="display-1">Display 1</h1>
    <h1 class="display-2">Display 2</h1>
    <h1 class="display-3">Display 3</h1>
    <h1 class="display-4">Display 4</h1>
  </div>

</body>
```

<small>

In Bootstrap 4 the HTML <small> element is used to create a lighter, secondary text in any heading

Example

```
<body>

  <div class="container">
    <h1>Lighter, Secondary Text</h1>
    <p>The small element is used to create a lighter, secondary text in any heading:</p>
    <h1>h1 heading <small>secondary text</small></h1>
    <h2>h2 heading <small>secondary text</small></h2>
    <h3>h3 heading <small>secondary text</small></h3>
    <h4>h4 heading <small>secondary text</small></h4>
    <h5>h5 heading <small>secondary text</small></h5>
    <h6>h6 heading <small>secondary text</small></h6>
  </div>

</body>
```

<mark>

Bootstrap 4 will style the HTML <mark> element with a yellow background color and some padding:

Example

```
<body>

  <div class="container">
    <h1>Highlight Text</h1>
    <p>Use the mark element to <mark>highlight</mark> text.</p>
  </div>

</body>
```

External Resources

https://www.w3schools.com/bootstrap4/bootstrap_ref_all_classes.asp

Bootstrap 4 Colors

Text Colors

- Bootstrap 4 has some contextual classes that can be used to provide "meaning through colors".
- The classes for text colors are:
 - **.text-muted**,
 - **.text-primary**
 - **.text-success**
 - **.text-info**
 - **.text-warning**
 - **.text-danger**
 - **.text-secondary**
 - **.text-white**
 - **.text-dark**
 - **.text-body** (default body color/often black) and
 - **.text-light**

Example

```
<body>

  <div class="container">
    <p class="text-muted">This text is muted.</p>
    <p class="text-primary">This text is important.</p>
    <p class="text-success">This text indicates success.</p>
    <p class="text-info">This text represents some information.</p>
    <p class="text-warning">This text represents a warning.</p>
    <p class="text-danger">This text represents danger.</p>
    <p class="text-secondary">Secondary text.</p>
    <p class="text-dark">This text is dark grey.</p>
    <p class="text-body">Default body color (often black).</p>
    <p class="text-light">This text is light grey (on white background).</p>
    <p class="text-white">This text is white (on white background).</p>
  </div>

</body>
```

Background Colors

- The classes for background colors are:
 - **.bg-primary**
 - **.bg-success**
 - **.bg-info**
 - **.bg-warning**
 - **.bg-danger**
 - **.bg-secondary**
 - **.bg-dark** and
 - **.bg-light**.

Note that background colors do not set the text color, so in some cases you'll want to use them together with a **.text-*** class.

Example

```
<body>

  <div class="container">
    <p>Note that you can also add a .text-* class if you want a different text color:</p>
    <p class="bg-primary text-white">This text is important.</p>
    <p class="bg-success text-white">This text indicates success.</p>
    <p class="bg-info text-white">This text represents some information.</p>
    <p class="bg-warning text-white">This text represents a warning.</p>
    <p class="bg-danger text-white">This text represents danger.</p>
    <p class="bg-secondary text-white">Secondary background color.</p>
    <p class="bg-dark text-white">Dark grey background color.</p>
    <p class="bg-light text-dark">Light grey background color.</p>
  </div>

</body>
```

Bootstrap 4 Images

Image Shapes

Rounded Corners

The **.rounded** class adds rounded corners to an image

Example

```
<body>
  <div class="container">
    <h2>Rounded Corners</h2>
    <p>The .rounded class adds rounded corners to an image:</p>
    
  </div>
</body>
```

Circle

The .rounded-circle class shapes the image to a circle

Example

```
<body>
  <div class="container">
    <h2>Circle</h2>
    <p>The .rounded-circle class shapes the image to a circle:</p>
    
  </div>
</body>
```

Thumbnail

The .img-thumbnail class shapes the image to a thumbnail (bordered)

Example

```
<body>
  <div class="container">
    <h2>Thumbnail</h2>
    <p>The .img-thumbnail class creates a thumbnail of the image:</p>
    
  </div>

</body>
```

Aligning Images

Float an image to the right with the **.float-right** class or to the left with **.float-left**

Example

```
<body>

  <div class="container">
    <h2>Aligning images</h2>
```

```
<p>Use the float classes to float the image to the left or to the right:</p>


</div>

</body>
```

Centered Image

Center an image by adding the utility classes **.mx-auto** (margin:auto) and **.d-block** (display:block) to the image

Example

```
<body>

  <div class="container">
    <h2>Centered Image</h2>
    <p>Center an image by adding the utility classes .mx-auto (margin:auto) and .d-block
(display:block) to the image</p>
    
  </div>

</body>
```

Responsive Images

- Images come in all sizes. So do screens. Responsive images automatically adjust to fit the size of the screen.
- Create responsive images by adding an **.img-fluid** class to the **** tag.
- The image will then scale nicely to the parent element.
- The **.img-fluid** class applies **max-width: 100%**; and **height: auto**; to the image

Example

```
<body>
  <div class="container">
    <h2>Image</h2>
    <p>The .img-fluid class makes the image scale nicely to the parent element (resize the
browser window to see the effect)</p>
    
  </div>
</body>
```

Bootstrap 4 Alerts

Bootstrap 4 provides an easy way to create predefined alert messages

Example

```
<body>

  <div class="container">
    <h2>Alert Links</h2>
    <p>Add the alert-link class to any links inside the alert box to create "matching colored links".</p>
    <div class="alert alert-success">
      <strong>Success!</strong> You should <a href="#" class="alert-link">read this message</a>.
    </div>
    <div class="alert alert-info">
      <strong>Info!</strong> You should <a href="#" class="alert-link">read this message</a>.
    </div>
    <div class="alert alert-warning">
      <strong>Warning!</strong> You should <a href="#" class="alert-link">read this message</a>.
    </div>
    <div class="alert alert-danger">
      <strong>Danger!</strong> You should <a href="#" class="alert-link">read this message</a>.
    </div>
    <div class="alert alert-primary">
      <strong>Primary!</strong> You should <a href="#" class="alert-link">read this message</a>.
    </div>
    <div class="alert alert-secondary">
      <strong>Secondary!</strong> You should <a href="#" class="alert-link">read this message</a>.
    </div>
    <div class="alert alert-dark">
      <strong>Dark!</strong> You should <a href="#" class="alert-link">read this message</a>.
    </div>
    <div class="alert alert-light">
      <strong>Light!</strong> You should <a href="#" class="alert-link">read this message</a>.
    </div>
  </div>

</body>
```


Closing Alerts

- To close the alert message, add a **.alert-dismissible** class to the alert container.
- Then add **class="close"** and **data-dismiss="alert"** to a link or a button element (when you click on this the alert box will disappear)

Example

```
<body>
  <div class="container">
    <h2>Alerts</h2>
    <p>The button with class="close" and data-dismiss="alert" is used to close the alert
box.</p>
    <p>The alert-dismissible class adds some extra padding to the close button.</p>
    <div class="alert alert-success alert-dismissible">
      <button type="button" class="close" data-dismiss="alert">&times;</button>
      <strong>Success!</strong> This alert box could indicate a successful or positive
action.
    </div>
    <div class="alert alert-info alert-dismissible">
      <button type="button" class="close" data-dismiss="alert">&times;</button>
      <strong>Info!</strong> This alert box could indicate a neutral informative change
or action.
    </div>
    <div class="alert alert-warning alert-dismissible">
      <button type="button" class="close" data-dismiss="alert">&times;</button>
      <strong>Warning!</strong> This alert box could indicate a warning that might need
attention.
    </div>
    <div class="alert alert-danger alert-dismissible">
      <button type="button" class="close" data-dismiss="alert">&times;</button>
      <strong>Danger!</strong> This alert box could indicate a dangerous or potentially
negative action.
    </div>
    <div class="alert alert-primary alert-dismissible">
      <button type="button" class="close" data-dismiss="alert">&times;</button>
      <strong>Primary!</strong> Indicates an important action.
    </div>
    <div class="alert alert-secondary alert-dismissible">
      <button type="button" class="close" data-dismiss="alert">&times;</button>
      <strong>Secondary!</strong> Indicates a slightly less important action.
    </div>
    <div class="alert alert-dark alert-dismissible">
      <button type="button" class="close" data-dismiss="alert">&times;</button>
      <strong>Dark!</strong> Dark grey alert.
    </div>
    <div class="alert alert-light alert-dismissible">
      <button type="button" class="close" data-dismiss="alert">&times;</button>
      <strong>Light!</strong> Light grey alert.
    </div>
  </div>
</body>
```

External Resources

https://www.w3schools.com/bootstrap4/bootstrap_ref_js_alert.asp

Bootstrap 4 Buttons

Button Styles

Bootstrap 4 provides different styles of buttons.

Example

```
<body>
  <div class="container">
    <h2>Button Styles</h2>
    <button type="button" class="btn">Basic</button>
    <button type="button" class="btn btn-primary">Primary</button>
    <button type="button" class="btn btn-secondary">Secondary</button>
    <button type="button" class="btn btn-success">Success</button>
    <button type="button" class="btn btn-info">Info</button>
    <button type="button" class="btn btn-warning">Warning</button>
    <button type="button" class="btn btn-danger">Danger</button>
    <button type="button" class="btn btn-dark">Dark</button>
    <button type="button" class="btn btn-light">Light</button>
    <button type="button" class="btn btn-link">Link</button>
  </div>
</body>
```

Active/Disabled Buttons

- A button can be set to an active (appear pressed) or a disabled (unclickable) state.
- The class **.active** makes a button appear pressed, and the disabled attribute makes a button unclickable.

Note that `<a>` elements do not support the disabled attribute and must therefore use the **.disabled** class to make it visually appear disabled.

Example

```
<body>
  <div class="container">
    <h2>Button States</h2>
    <button type="button" class="btn btn-primary">Primary Button</button>
    <button type="button" class="btn btn-primary active">Active Primary</button>
    <button type="button" class="btn btn-primary disabled">Disabled Primary</button>
    <a href="#" class="btn btn-primary disabled">Disabled Link</a>
  </div>
</body>
```

Spinner Buttons (Loading)

You can also add "spinners" to a button that can be used while showing loading progress.

Example

```
<body>
  <div class="container">
    <h2>Spinner Buttons</h2>
    <p>Add spinners to buttons:</p>

    <button class="btn btn-primary">
      <span class="spinner-border spinner-border-sm"></span>
    </button>

    <button class="btn btn-primary">
      <span class="spinner-border spinner-border-sm"></span>
      Loading..
    </button>

    <button class="btn btn-primary disabled">
      <span class="spinner-border spinner-border-sm"></span>
      Loading..
    </button>

    <button class="btn btn-primary disabled">
      <span class="spinner-grow spinner-grow-sm"></span>
      Loading..
    </button>
  </div>
</body>
```

Bootstrap Layout Exercises

<https://getbootstrap.com/docs/4.0/examples/>

Project 1

Build Personal Portfolio Website with Bootstrap 4

Step1

Clone the project setup from https://github.com/thehackingschool/SOC_Project1_BoilerPlate and initialize the div container or div container-fluid

Step 2 Implement Header Section into the above imported setup

```
<body>
  <div class="container">
    <header id="main-header">
      <div class="row no-gutters">
        <div class="col-lg-4 col-md-5">
          
        </div>
        <div class="col-lg-8 col-md-7">
          <div class="d-flex flex-column">
            <div class="p-5 bg-dark text-white">
              <div class="name d-flex flex-row justify-content-between align-items-center">
                <h1 class="display-4">Elon Musk</h1>
                <div><i class="fa fa-twitter"></i></div>
                <div><i class="fa fa-facebook"></i></div>
                <div><i class="fa fa-instagram"></i></div>
                <div><i class="fa fa-github"></i></div>
              </div>
            </div>
            <div class="p-4 bg-black">
              Experienced Full Stack Web Developer
            </div>
          </div>
          <div class="d-flex flex-row text-white align-items-stretch text-center">
            <div class="port-item p-4 bg-primary" data-toggle="collapse" data-target="#home">
              <i class="fa fa-home d-block"></i> Home
            </div>
            <div class="port-item p-4 bg-success" data-toggle="collapse" data-target="#resume">
              <i class="fa fa-graduation-cap d-block"></i> Resume
            </div>
            <div class="port-item p-4 bg-warning" data-toggle="collapse" data-target="#work">
              <i class="fa fa-folder-open d-block"></i> Work
            </div>
            <div class="port-item p-4 bg-danger" data-toggle="collapse" data-target="#contact">
              <i class="fa fa-envelope d-block"></i> Contact
            </div>
          </div>
        </div>
      </div>
    </header>
  </body>
```

Step 3 Implement Home Page Section

```
<!-- HOME -->
  <div id="home" class="collapse show">
    <div class="card card-body bg-primary text-white py-5">
      <h2>Welcome to my page</h2>
      <p class="lead">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Perferendis,
quia.</p>
    </div>
```

```

<div class="card card-body py-5">
  <h3>My Skills</h3>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Dolorum nulla et, modi
harum hic deserunt.</p>
  <hr>
  <h4>HTML</h4>
  <div class="progress mb-3">
    <div class="progress-bar bg-success" style="width:100%"></div>
  </div>
  <h4>CSS</h4>
  <div class="progress mb-3">
    <div class="progress-bar bg-success" style="width:100%"></div>
  </div>
  <h4>JavaScript</h4>
  <div class="progress mb-3">
    <div class="progress-bar bg-success" style="width:90%"></div>
  </div>
  <h4>PHP</h4>
  <div class="progress mb-3">
    <div class="progress-bar bg-success" style="width:80%"></div>
  </div>
  <h4>Python</h4>
  <div class="progress mb-3">
    <div class="progress-bar bg-success" style="width:70%"></div>
  </div>
</div>
</div>

```

Step 4 Implement Resume Section

```

<div id="resume" class="collapse">
  <div class="card card-body bg-success text-white py-5">
    <h2>My Resume</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Illo nobis ut labore iure
tempore qui!</p>
  </div>

  <div class="card card-body py-5">
    <h3>Where have I worked?</h3>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Nobis eligendi, ex officia
itaque tempora natus.
    </p>
    <div class="card-deck">
      <div class="card">
        <div class="card-body">
          <h4 class="card-title">Devmasters</h4>
          <p class="card-text">Lorem ipsum dolor sit amet, consectetur adipisicing elit.
Quod, nostrum.</p>
          <p class="p-2 mb-3 bg-dark text-white">
            Position: Full Stack Developer
          </p>
          <p class="p-2 mb-3 bg-dark text-white">
            Phone: (444) 444-4444
          </p>
        </div>
      </div>
    </div>
  </div>

```

```

    </div>
    <div class="card-footer">
      <h6 class="text-muted">Dates: 2015 - 2017</h6>
    </div>
  </div>

  <div class="card">
    <div class="card-body">
      <h4 class="card-title">Websites Pro</h4>
      <p class="card-text">Lorem ipsum dolor sit amet, consectetur adipisicing elit.
Quod, nostrum.</p>
      <p class="p-2 mb-3 bg-dark text-white">
        Position: Front End Developer
      </p>
      <p class="p-2 mb-3 bg-dark text-white">
        Phone: (333) 333-3333
      </p>
    </div>
    <div class="card-footer">
      <h6 class="text-muted">Dates: 2013 - 2015</h6>
    </div>
  </div>

  <div class="card">
    <div class="card-body">
      <h4 class="card-title">123 Designs</h4>
      <p class="card-text">Lorem ipsum dolor sit amet, consectetur adipisicing elit.
Quod, nostrum.</p>
      <p class="p-2 mb-3 bg-dark text-white">
        Position: Designer
      </p>
      <p class="p-2 mb-3 bg-dark text-white">
        Phone: (222) 222-2222
      </p>
    </div>
    <div class="card-footer">
      <h6 class="text-muted">Dates: 2010 - 2013</h6>
    </div>
  </div>
</div>
</div>
</div>

```

Step 5 Implement Work Section

```

<!-- WORK -->
<div id="work" class="collapse">
  <div class="card card-body bg-warning py-5">
    <h3>My Portfolio</h3>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Saepe, adipisci?</p>
  </div>

  <div class="card card-body py-5">
    <h3>What have I built?</h3>

```

```

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Dicta, eum.</p>
<div class="row no-gutters">
  <div class="col-md-3">
    <a href="https://unsplash.it/1200/768.jpg?image=251" data-toggle="lightbox">
      
    </a>
  </div>
  <div class="col-md-3">
    <a href="https://unsplash.it/1200/768.jpg?image=252" data-toggle="lightbox">
      
    </a>
  </div>
  <div class="col-md-3">
    <a href="https://unsplash.it/1200/768.jpg?image=253" data-toggle="lightbox">
      
    </a>
  </div>
  <div class="col-md-3">
    <a href="https://unsplash.it/1200/768.jpg?image=254" data-toggle="lightbox">
      
    </a>
  </div>
</div>

<div class="row no-gutters">
  <div class="col-md-3">
    <a href="https://unsplash.it/1200/768.jpg?image=255" data-toggle="lightbox">
      
    </a>
  </div>
  <div class="col-md-3">
    <a href="https://unsplash.it/1200/768.jpg?image=256" data-toggle="lightbox">
      
    </a>
  </div>
  <div class="col-md-3">
    <a href="https://unsplash.it/1200/768.jpg?image=257" data-toggle="lightbox">
      
    </a>
  </div>
  <div class="col-md-3">
    <a href="https://unsplash.it/1200/768.jpg?image=258" data-toggle="lightbox">
      
    </a>
  </div>
</div>
</div>
</div>

```

Step 6 Implement Contact Section

```

<!-- CONTACT -->
<div id="contact" class="collapse">

```

```

<div class="card card-body bg-danger text-white py-5">
  <h2>Contact</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Exercitationem,
dignissimos?</p>
</div>

<div class="card card-body py-5">
  <h3>Get in touch</h3>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Laborum, autem.</p>
  <form>
    <div class="form-group">
      <div class="input-group input-group-lg">
        <span class="input-group-addon bg-danger text-white">
          <i class="fa fa-user"></i>
        </span>
        <input type="text" class="form-control bg-dark text-white" placeholder="Name">
      </div>
    </div>
    <div class="form-group">
      <div class="input-group input-group-lg">
        <span class="input-group-addon bg-danger text-white">
          <i class="fa fa-envelope"></i>
        </span>
        <input type="email" class="form-control bg-dark text-white" placeholder="Email">
      </div>
    </div>
    <div class="form-group">
      <div class="input-group input-group-lg">
        <span class="input-group-addon bg-danger text-white">
          <i class="fa fa-pencil"></i>
        </span>
        <textarea rows="5" class="form-control bg-dark text-white"
placeholder="Message"></textarea>
      </div>
    </div>
    <input type="submit" value="Submit" class="btn btn-danger btn-block btn-lg">
  </form>
</div>
</div>

```

Step 7 Implement Footer Section

```

<!-- FOOTER -->
<footer id="main-footer" class="p-5 bg-dark text-white">
  <div class="row">
    <div class="col-md-6">
      <a href="#" class="btn btn-outline-light"><i class="fa fa-cloud"></i>
Download Resume</a>
    </div>
  </div>
</footer>

```


Javascript Project 1 - Battery API Project

Web APIs

When writing code for the Web, there are a large number of Web APIs available. Below is a list of all the APIs and interfaces (object types) that you may be able to use while developing your Web app or site.

Here is the list of MDN Web APIs that comes built in with most of the web browsers

<https://developer.mozilla.org/en-US/docs/Web/API>

We are going to build an app that will notify the laptop current battery status using Web API of Battery Status and JS DOM.

Battery Status API :

https://developer.mozilla.org/en-US/docs/Web/API/Battery_Status_API

<https://www.w3.org/TR/battery-status/#examples>

Source Code

```
<!DOCTYPE html>
<html>

<head>
  <title>Battery Status API Project</title>
</head>

<body>
  <div id="charging"></div>
  <div id="level"></div>
  <progress id="progresslevel" max="100"> </progress>
  <div id="dischargingTime"></div>
</body>
<script>
  function updateBatteryStatus(battery) {
    document.querySelector('#charging').textContent = battery.charging ? 'The Current
Current status : Charging' : 'The Current Charging status : Not Charging';
    document.querySelector('#level').textContent = 'The Current Battery Level : ' +
battery.level * 100 + '%';
    document.querySelector('#progresslevel').value = battery.level * 100;

    document.querySelector('#dischargingTime').textContent = 'The Remaining Battery in
Minutes : ' + battery.dischargingTime / 60;
  }

  navigator.getBattery().then(function(battery) {
    // Update the battery status initially when the promise resolves ...
    updateBatteryStatus(battery);
  });
</script>
</html>
```

```

    // .. and for any subsequent updates.
    battery.onchargingchange = function() {
        updateBatteryStatus(battery);
    };

    battery.onlevelchange = function() {
        updateBatteryStatus(battery);
    };

    battery.ondischargingtimechange = function() {
        updateBatteryStatus(battery);
    };
});
</script>

</html>

```

Note : You don't need to understand the behaviour of navigator.getBattery() api at this stage. All we need to know how to integrate the available web API into our HTML code and JS DOM. We need to fetch the values and insert dynamically in the HTML tags. You can refer to the documentation for more example code snippets. You can apply stylings if you need to beautify the layout.

Javascript Project 2 - GitHub Search Engine

Axios

Axios is a Javascript library used to make HTTP requests from node.js or XMLHttpRequests from the browser that also supports the ES6 Promise API.

Axios CDN

<https://cdnjs.com/libraries/axios>

HTML Layout of GitHub Search Engine

index.html

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>GitHub Profile Finder</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/axios/0.19.2/axios.js"
integrity="sha256-bd8XIKzrtyJ105Sh3Xp3GiuMIzWC42ZekvrMMD4GxRg="
crossorigin="anonymous"></script>
</head>

<body>
    <nav>

```

```

<div class="nav-wrapper">
  <a href="#" class="brand-logo center ">GitHub Profile Search</a>
</div>
</nav>

<div class="container">
  <div class="row">
    <div class="col s5 push-s7">
      <div class="row">
        <div class="col s12 m7">
          <div id="profile-card" class="card">
            <div class="card-image">
              <img id="profile-image" src="">
              <span id="profile-title" class="card-title"></span>
            </div>
            <div class="card-content">
              <div class="row">
                <div id="profile-followers" class="col s6 badge
valign-wrapper"></div>
                <div id="profile-following" class="col s6 badge
valign-wrapper"></div>
              </div>
              <div class="row">
                <div id="profile-location" class="col s12
center-align"></div>
              </div>
            </div>
            <div class="card-action center-align">
              <a id="profile-page" href="#" target="_blank">Profile Page</a>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div class="col s7 pull-s5">
      <div class="row">
        <div class="input-field col s6">
          <input id="profile-name" type="text" class="validate">
          <label for="profile-name">Search Profile</label>
        </div>
      </div>
    </div>
  </div>
  <div id="not-found">
    <div id="row">
      <div class="center-align">
        <h5>Profile Not Found</h5>
      </div>
    </div>
  </div>
  <div id="other-details" class="row center-align">
    <div class="row red accent-1">
      <div class="flow-text">Other Details</div>
    </div>
  </div>
</div>

```

```

</div>

<div class="row z-depth-1">
  <div class="col s4 left-align">
    <p>Organization</p>
  </div>
  <div class="col s4 left-align">
    <p id="org"></p>
  </div>
</div>
<div class="row z-depth-1">
  <div class="col s4 left-align">
    <p>Bio</p>
  </div>
  <div class="col s4 left-align">
    <p id="bio"></p>
  </div>
</div>
<div class="row z-depth-1">
  <div class="col s4 left-align">
    <p>Repos</p>
  </div>
  <div class="col s4 left-align">
    <p id="repos"></p>
  </div>
</div>
<div class="row z-depth-1">
  <div class="col s4 left-align">
    <p>Gists</p>
  </div>
  <div class="col s4 left-align">
    <p id="gists"></p>
  </div>
</div>
<div class="row z-depth-1">
  <div class="col s4 left-align">
    <p>Blog</p>
  </div>
  <div class="col s4 left-align">
    <p id="blog"></p>
  </div>
</div>
</div>
</div>
</body>
<script src="./js/github-profile.js"></script>

</html>

```

Note : You can import the above html layout or you can make your own customised bootstrap/css/material ui layout. Remember, all the values for the div tags will be coming from GitHub API called '<https://api.github.com/users/>'; We use axios to pull the data and insert into the above div tags dynamically.

github-profile.js

```
const profileCard = document.getElementById('profile-card');
const profileOtherDetails = document.getElementById('other-details');
const profileNotFound = document.getElementById('not-found');
const profileNameInput = document.getElementById('profile-name');
const apiUrl = 'https://api.github.com/users/';

function init() {
  profileCard.style.display = 'none';
  profileNotFound.style.display = 'none';
  profileOtherDetails.style.display = 'none';
  profileNameInput.addEventListener('change', onProfileNameChange);
}

function renderProfileCard(profile) {
  const { blog, public_repos, public_gists, bio, avatar_url, name, followers, following,
html_url, location, company } = profile;
  profileCard.style.display = 'block';
  profileOtherDetails.style.display = 'block';
  const profileImage = document.getElementById('profile-image');
  const profileTitle = document.getElementById('profile-title');
  const profileFollower = document.getElementById('profile-followers');
  const profileFollowing = document.getElementById('profile-following');
  const profilePage = document.getElementById('profile-page');
  const profileLocation = document.getElementById('profile-location');
  const organization = document.getElementById('org');
  const repos = document.getElementById('repos');
  const gists = document.getElementById('gists');
  const biod = document.getElementById('bio');
  const blg = document.getElementById('blog');

  // Card Details
  profileImage.setAttribute('src', avatar_url);
  profileTitle.innerHTML = name;
  profileFollower.innerHTML = `Follower ${followers || 0}`;
  profileFollowing.innerHTML = `Following ${following || 0}`;
  profilePage.setAttribute('href', html_url);
  profileLocation.innerHTML = location;

  // Other Details

  organization.innerHTML = company || 'No Organization Data';
  biod.innerHTML = bio || 'No Bio data';
  repos.innerHTML = public_repos || 0;
  gists.innerHTML = public_gists || 0;
  blg.innerHTML = blog || 'No blog data'
}

function onProfileNameChange(event) {
  const profileName = event.target.value;
  if (profileName.length > 0) {
    axios.get(`${apiUrl}${profileName}`)
      .then(({ data }) => renderProfileCard(data))
      .catch((err) => {
        if (err) {
          console.log('Error: ', err);
        }
      })
  }
}
```

```

        profileCard.style.display = 'none';
        profileOtherDetails.style.display = 'none';
        profileNotFound.style.display = 'block';
    });
} else {
    profileCard.style.display = 'none';
    profileOtherDetails.style.display = 'none';
    profileNotFound.style.display = 'none';
}
}

init();

```

Javascript Project 3 - Weather App

Register at <https://openweathermap.org/> to get the API keys.

Source Code

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Weather App</title>
    <link rel="stylesheet" href="css/styles.css">
    <script src="https://cdn.jsdelivr.net/npm/axios@0.19.2/dist/axios.js"
        integrity="sha256-bd8XIKzrtyJ105Sh3Xp3GiuMIzWC42ZekvrMMD4GxRg="
        crossorigin="anonymous"></script>
</head>

<body>
    <div>
        <h1>Weather App</h1>
        <label for="city">Enter City Name: </label>
        <input type="text" name="city" id="city" placeholder="Enter City Name">
        <input type="submit" value="Submit" onclick="init()">
        <div>
            <p id="cord"></p>
            <p id="weather"></p>
            <p id="temp"></p>
        </div>
    </div>
</body>
<script>
    function init() {
        const url = 'http://api.openweathermap.org/data/2.5/weather?q=';
        const appid = '8e5c5d89aa0321b039a906842ea53fed'; //register to get app id
        const cityName = document.getElementById('city').value;
    }

```

```
const apiCall = `${url}${cityName}&appid=${appid}`;
axios.get(apiCall)
  .then((res) => {
    document.getElementById('cord').innerHTML = `The Latitude of City :
<b>${res.data.coord.lat}<sup>0</sup></b> and The Longitude of City :
<b>${res.data.coord.lon}<sup>0</sup></b>`;

    document.getElementById('weather').innerHTML = `Weather Cloudiness :
<b>${res.data.weather[0].description}</b>`;

    document.getElementById('temp').innerHTML = `Humidity :
<b>${res.data.main.humidity}%</b> Pressure : <b>${res.data.main.pressure}</b> hpa</b> Temperature
: <b>${res.data.main.temp}K</b>`;

  })
  .catch((err) => {
    document.write("Invalid City. Please refresh and Enter valid city name");
  })
}
</script>

</html>
```