

# Bootstrap Tutorial

We will be building a responsive web page using Twitter's Bootstrap framework.

## Resources:

- Bootstrap (Google it!)
- Code editor (TextWrangler, Brackets, Dw, NotePad++)

## Step 1 - Get Bootstrap

You will need to download the **Bootstrap** framework from the website - <http://getbootstrap.com/>

Go straight to the **Customise** tab, scroll right to the bottom and click **Compile and Download**.

Create a new folder for your project then **copy and paste** the Bootstrap folder content to your project folder.

If you look in the CSS folder you will find two types of CSS files. A "human readable" (bootstrap.css) version and a "robot min" (bootstrap.min.css) version. The min version is simply used to speed up the processing by removing all the blank space and lines.

## Step 2 - HTML & CSS

You will now create your two core files. Create an HTML file called **index.html** and a CSS file called **styles.css**.

Save **index.html** in your root folders and **styles.css** in your css folder.

Your **index** file should be HTML5 compatible so we must enter the following HTML into it unless you have create an HTML5 blank document in Dreamweaver:

```
1  <!doctype html>
2  <html>
3  <head>
4  <meta name="viewport" content="width=device-width, initial-scale=1.0" charset="UTF-8">
5      <link href="css/bootstrap.min.css" rel="stylesheet" type="text/css" />
6      <link href="css/style.css" rel="stylesheet" type="text/css" />
7  <title>Bootstrap 3 Tutorial</title>
8  </head>
9
10 <body>
11 </body>
12 </html>
```

**Meta:** Many responsive websites follow this meta information. This allows the site to work with the device and respond accordingly to the screen size, resolution, device and orientation.

**Link 1:** Bootstrap.min.css is added to the site for a fast processing of the framework.

**Link 2:** Our style sheet is added to allow for us to style the website specific to how we want it.

## Step 3 - jQuery Library

We will use the Google jQuery Library and this can be found at: <http://goo.gl/D4dWqx>

We want the following library and it is recommended you copy the top line of code to import into your **index.html** document.

**jQuery**

 **snippet:** `<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>`  
**snippet:** `<script src="//ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>`  
**site:** <http://jquery.com/>  
**versions:** 2.1.1, 2.1.0, 2.0.3, 2.0.2, 2.0.1, 2.0.0, 1.11.1, 1.11.0, 1.10.2, 1.1...  
**note:** 1.2.5 and 1.2.4 are not hosted due to their short and unstable lives in the wild.

By default the code begins with “//ajax.google...” we must add at the beginning of this **http**: so it reads “**http://ajax.google...**”.

The reason for this is because our website is not on the internet yet so we must define the hyper text transfer protocol (http) section at the beginning of the script domain.

Alternatively once live we could return the link back to the original “//ajax.google...” and this would simply check first to define if the site is **http** (unsecured) or **https** (secured).

## Bootstrap jQuery Library

We must also import our Bootstrap jQuery library too. We do this by linking our **html** document to the **bootstrap.js** file in the **js** folder. All code as follows.

```
13     <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
14     <script src="js/bootstrap.js"></script>
15 </body>
16 </html>
```

## Test

Test your website by creating a **<h1>Hello, World!</h1>** tag in the **<body>**. You will know it has worked if your font is NOT Times New Roman.

## Step 4 - Responsive Collapsing Nav Bar

Firstly remove the `<h1> Hello World` tag from your website, this was just to test it.

Secondly we will start to add the navigation via a few of our own elements of code but specifically linked directly to the responsive framework of Bootstrap.

We will use the frameworks **navbar** and **navbar-default** settings. Add these as follows in the same place the `<h1>` previously was. The navbar can also be inverted by changing **default** to **invertse**.

You will also notice the navbar may have curved corners, in order to correct this (if you wish) and to make the navbar static to the top of the screen we will also add to the class **navbar-static-top**. This can also be changed to **navbar-fixed-top** which will make it fixed to the top of the screen. Ideal for a one page long scrolling website.

This individual `<div>` has three separate **classes** associated to it by simply separating each class command with a space.

```
12     <div class="navbar navbar-inverse navbar-static-top">
13
14     </div>
```

### Title

Inside the previous `<div>` we will create another `<div>` and give this a **class="container"**.

This will also house our websites title of which we will add another class to it called **navbar-brand**. However our title needs to be clickable and thus we will add an **anchor** tag to it and give it a dummy link by adding `"#"` to the **href**.

```
12     <div class="navbar navbar-inverse navbar-static-top">
13         <div class="container">
14             <a href="#" class="navbar-brand">WebberBridge Sports Hall</a>
15         </div>
16     </div>
```

### Collapsible

We need to make sure the navbar collapses depending on the type of screen and device we are on. To do this simply add the following three classes (`collapse navbar-collapse navHeaderCollapse`) to a new `<div>` inside and directly under the title anchor.

```
12     <div class="navbar navbar-inverse navbar-static-top">
13         <div class="container">
14             <a href="#" class="navbar-brand">WebberBridge Sports Hall</a>
15             <div class="collapse navbar-collapse navHeaderCollapse">
16
17             </div>
18         </div>
19     </div>
```

## Unordered List Items

Inside the previous `<div>` we will create our unordered list that fundamentally will become our navigation. This unordered list will also require three classes adding to it.

**nav navbar-nav:** these offer styling on all levels.

**navbar-right:** positioning.

Make sure to add for the **Home** button a direct like to **index.html** by using the **anchor** tag.

```

12     <div class="navbar navbar-inverse navbar-static-top">
13         <div class="container">
14             <a href="#" class="navbar-brand">WebberBridge Sports Hall</a>
15             <div class="collapse navbar-collapse navHeaderCollapse">
16                 <ul class="nav navbar-nav navbar-right">
17                     <li><a href="index.html">Home</a></li>
18                 </ul>
19             </div>
20         </div>
21     </div>

```

## Collapsible Button

Directly below the **anchor** tag for the **Home** button you will add a quick note to identify what is being added here. `<!--` - html notes - `-->`

The `<button>` tag will be used to make the element interactive. We add this along with the class **navbar-toggle**. We also define the **data-toggle** as “collapse” and link the **data-target** the class “**.navHeaderCollapse**”.

To test it simply add a single character in the within the button tag and resize the browser.

```

14     <a href="#" class="navbar-brand">WebberBridge Sports Hall</a>
15     <!--this is where the collapsable button goes-->
16     <button class="navbar-toggle" data-toggle="collapse" data-target=".navHeaderCollapse">
17         f
18     </button>
19     <div class="collapse navbar-collapse navHeaderCollapse">

```

We will add the universal **three bar menu button icon** within the button tag.

To do this we will add a **span** tag with the class “**icon-bar**”. Do this three times in order to get the three bars we are familiar with when on a mobile device.

```

16     <button class="navbar-toggle" data-toggle="collapse" data-target=".navHeaderCollapse">
17         <span class="icon-bar"></span>
18         <span class="icon-bar"></span>
19         <span class="icon-bar"></span>
20     </button>

```

## More List Items for the Navigation

Add the rest of your navigation in accordance to the **site map** you have produced.

We will also make the page we are developing an active link to show the user which page they are on. This aids UI/UX development. To do this simply add the **class="active"** to the **list item**.

```

21         <div class="collapse navbar-collapse navHeaderCollapse">
22             <ul class="nav navbar-nav navbar-right">
23                 <li class="active"><a href="index.html">Home</a></li>
24                 <li><a href="#">Sports</a></li>
25                 <li><a href="#">Bookings</a></li>
26                 <li><a href="#">Contact</a></li>
27             </ul>
28         </div>

```

## DropDown Menus

Our sports button needs to be a drop down menu. In order to create this we delete the **Sports list item** and replace it with a new list item:

```

<li class="dropdown">
  <a href="#" class="dropdown-toggle" data-toggle="dropdown">Sports</a>
</li>

```

We then need to create the content for the drop down menu. This will be housed within our list item for the drop down.

```

<li class="dropdown">
  <a href="#" class="dropdown-toggle" data-toggle="dropdown">Sports</a>
  <ul class="dropdown-menu">
    <li><a href="#">Football</a></li>
    <li><a href="#">Netball</a></li>
    <li><a href="#">Hockey</a></li>
    <li><a href="#">Badminton</a></li>
  </ul>
</li>

```

## Add the Caret

Drop down menus are obviously more apparent when they have an instruction (UI/UX). We will now add a little arrow known as a **caret** to identify the drop down menu.

This is added to the **Sports** button.

```

<a href="#" class="dropdown-toggle" data-toggle="dropdown">Sports <b class="caret"></b></a>

```

## Step 5 - Sticky Footer (viewport specific)

This will be created using the same process as the navigation but using a different colour and some slightly different instructions.

### The Setup

We will enter this code above our `<script>` code at the bottom of the website within the `<body>` tag.

Two `<div>` tags will be used. ONE to house the footer and TWO to house the content for the footer. The `<p>` tag will house our text that we want in the footer. For this tutorial place "Site Built by..." and your name after it.

We apply a selection of class's to the first `<div>` in order to get the nav bar to the bottom of the page. These can be viewed in the code below. Enter and refresh browser.

```
40 <div class="navbar navbar-default navbar-fixed-bottom">
41   <div class="container">
42     <p>Site Built by Shaun Reeves</p>
43   </div>
44 </div>
45
46 <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
```

### Your name, but better

To make your name more stylish in the footer we will add a simple **class** to the `<p>` tag.

```
<p class="navbar-text">Site Built by Shaun Reeves</p>
```

### Add a button

We add an `<a>` (anchor) tag below the `<p>` tag (your name) and this will be the button. Applying the class `"navbar-btn"` will help make this look stylish when we have finished.

```
<p class="navbar-text">Site Built by Shaun Reeves</p>
<a class="navbar-btn">Contact Us</a>
```

### Style your button

Now we have a button we need to style it. Head over to the Bootstrap website (<http://getbootstrap.com/css/#buttons>) and choose a default colour for your button.

You will need the class that is specific to the button. I will be using **"Success"** which has the class `"btn-success"` and to make it look like a button we use the global `"btn"` class to style the edges. Try this with and without the `"btn"` class to see what it does.

```
<p class="navbar-text">Site Built by Shaun Reeves</p>
<a class="navbar-btn btn btn-success">Contact Us</a>
```

Finally to float the button right and the text left use the classes `"pull-right"` and `"pull-left"`.

```
<p class="navbar-text pull-left">Site Built by Shaun Reeves</p>
<a class="navbar-btn btn btn-success btn pull-right">Contact Us</a>
```

## Step 6 - Jumbotron

A Jumbotron is a large area that could be used for a variety of different things. Adverts, video or articles.

We start by adding a `<div>` outside our navigation div but above our footer div.

```

<!--Jumbotron goes here -->
<div class="container">

  <div class="jumbotron">

  </div>

</div>

```

Within the **Jumbotron** tag add a `<h1>` tag and for now just add the text “**Hello, World!**”.

```

<div class="jumbotron">
  <h1>Hello, World!</h1>
</div>

```

## Dummy Text

We will then add some dummy text also known as **Lorem Ipsum** to fill the Jumbotron. Head over to <http://www.lipsum.com/> and select **1 paragraph** to be generated. From here just select **2-3 sentences** of text. This will be placed within the `<p>` paragraph tags in our Jumbotron.

```

43     <div class="jumbotron">
44         <h1>Hello, World!</h1>
45         <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lorem orci, tristique tristique
        lorem ut, tincidunt iaculis libero. Aliquam elementum auctor neque ut placerat. Duis facilisis purus nec leo
        vestibulum, a imperdiet odio elementum. </p>
46     </div>
47
48 </div>

```

## Jumbotron Buttons

Buttons can also be added to the Jumbotron by simply adding an anchor tag `<a>` with a button class `<a class="btn btn-default">`

Remember to use the pre-defined buttons used on the Bootstrap > CSS > Buttons section of the website - <http://getbootstrap.com/css/#buttons>

We will add two buttons: 1. “*Watch Now*” and 2. “*Tweet It!*”

```

        lorem ut, tincidunt iaculis libero. Aliquam elementum
        vestibulum, a imperdiet odio elementum. </p>
46         <a class="btn btn-default">Watch Now</a>
47         <a class="btn btn-info">Tweet It!</a>
48     </div>

```

We want the Twitter button to be a lighter shade of blue to match the colour scheme of Twitter. So we will make the class for the “*Tweet It!*” button to be “*btn-info*”.

To finally centre the Jumbotron content by adding the class “**text-centre**” to the Jumbotron `<div>`.

## Step 7 - Grid System

This will allow the content of our website to respond to the browser window and device.

### Setting up the grid system

We need to enter our code into our website **below the JUMBOTRON and above the FOOTER NAV**. You should easily identify these via the notes that you have been entering throughout...

...I hope!

```
<!--Grid-->
<div class="container">

  <div class="row">

    <div class="col-md-3">

    </div>

  </div>

</div>
```

### Adding content

So here we will add content and create a summary of an article to entice the user to read. We do this by adding inside the **<div class="col-md-3">** a **header**, **paragraph** and a **button**.

```
<!--Grid-->
<div class="container">

  <div class="row">

    <div class="col-md-3">

      <h3><a href="#">Sports News</a></h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
      Etiam lorem orci, tristique tristique lorem ut, tincidunt iaculis libero.
      Aliquam elementum auctor neque ut placerat. Duis facilisis purus nec leo
      vestibulum, a imperdiet odio elementum.</p>
      <a href="#" class="btn btn-default">Read More</a>

    </div>

  </div>

</div>
```

Be sure to copy the **Lorem Ipsum** from the **Jumbotron** to save you time when filling out the summary section for the **paragraph <p>**.

You will also notice we are using the default button. If you want to change the colour of the button please feel free to do so referencing the code from the Bootstrap website - <http://getbootstrap.com/css/#buttons>



## More Content

Now copy your first article summary and paste it below **three** more times. As shown below.

**ADVICE:** add notes to show where each article starts as I have done.

```
<!--Grid-->
<div class="container">

  <div class="row">
    <!--article 1-->
      <div class="col-md-3">

        <h3><a href="#">Sports News</a></h3>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam lorem orci, tristique tristique lorem ut, tincidunt iaculis libero.
Aliquam elementum auctor neque ut placerat. Duis facilisis purus nec leo
vestibulum, a imperdiet odio elementum.</p>
        <a href="#" class="btn btn-default">Read More</a>

      <!--article 2-->
    </div>

    <div class="col-md-3">

      <h3><a href="#">Sports News</a></h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam lorem orci, tristique tristique lorem ut, tincidunt iaculis libero.
Aliquam elementum auctor neque ut placerat. Duis facilisis purus nec leo
vestibulum, a imperdiet odio elementum.</p>
      <a href="#" class="btn btn-default">Read More</a>

    </div>
    <!--article 3-->
      <div class="col-md-3">

        <h3><a href="#">Sports News</a></h3>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam lorem orci, tristique tristique lorem ut, tincidunt iaculis libero.
Aliquam elementum auctor neque ut placerat. Duis facilisis purus nec leo
vestibulum, a imperdiet odio elementum.</p>
        <a href="#" class="btn btn-default">Read More</a>

      </div>
      <!--article 4-->
        <div class="col-md-3">

          <h3><a href="#">Sports News</a></h3>
          <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam lorem orci, tristique tristique lorem ut, tincidunt iaculis libero.
Aliquam elementum auctor neque ut placerat. Duis facilisis purus nec leo
vestibulum, a imperdiet odio elementum.</p>
          <a href="#" class="btn btn-default">Read More</a>

        </div>

      </div>

    </div>

  </div>
```

## 3 is NOT the magic number, 12 is!

The class **col-md-3** is specific to the number of columns. Here is how it works.

We have a grid system that is relative to **12 Columns**.

1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---

The **3** in our class (**col-md-3**) is relative to **3** cells of our **row** starting from the left.

3	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---

As we add more content in that row we must make sure our row adds up to 12 and **no more than 12!** Therefore if we have **four articles** set to a class of **col-md-3** then we have **12**.

3	3	3	3
---	---	---	---

We could have the following too, but as long as it doesn't come to more than 12 we are fine!

3	3	2	4
---	---	---	---

Test this by resizing your browser.

## Step 8 - Popup Box by Modals

Begin by adding the clickable event to the Contact button of the navigation. In a later section of this tutorial I will explain how to create a contact form but for now I will introduce the Modal function.

Add the following code to the **Contact** section of the navigation. The **href** will be specific to a **#contact** link which will be identified and called upon through the jQuery library. We also need to add the **data-toggle** which will be set to **modal**.

```
<li><a href="#">Bookings</a></li>
<li><a href="#contact" data-toggle="modal">Contact</a>
```

## Embedding the Modal

We are going to add the Modal in the bottom of our HTML page because we want it to load last. The user is not going to load the Modal form straight away and it will give the page time to load.

We will embed the Modal form at the bottom of our page, above the **<script>** and below the **<footer>**.

```
<!--modal-->
<div class="modal fade" id="contact" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h4>Contact WebberBridge Sports Hall</h4>
      </div>
    </div>
  </div>
</div>
```

Bootstrap uses lots of `<div>` tags to get the job done but in the end it does work very well and is ultra compatible. So apologies!

To the main `<div class="modal fade">` we need to add an `id="contact"` this will make the link between the navigation “**Contact**” button and the **Modal** form itself. And we need to have a role for the **Modal** form connecting it by `role="dialog"`.

Test your website to see if this has worked.

## Modal Body

Enter the Modal body into a `<div class="modal-body">` below the `<div class="modal-header">`. Just for now copy and paste some **Lorem Ipsum** from one of the news articles in Step 7 - Grid System.

I have also added some HTML notes to show where the content is. You could do this to help you.

```
<!--modal header-->
  <div class="modal-header">
    <h4>Contact WebberBridge Sports Hall</h4>
  </div>
<!--modal body-->
  <div class="modal-body">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing
    Etiam lorem orci, tristique tristique lorem ut, tincidunt iaculis
    Aliquam elementum auctor neque ut placerat. Duis facilisis purus nec
    vestibulum, a imperdiet odio elementum.</p>
  </div>
```

## Modal Footer

Enter the Modal footer into a `<div class="modal-footer">` below the `<div class="modal-body">`. We will also add to this a button.

```
<!--modal header-->
  <div class="modal-header">
    <h4>Contact WebberBridge Sports Hall</h4>
  </div>
<!--modal body-->
  <div class="modal-body">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing
    elit. Etiam lorem orci, tristique tristique lorem ut, tincidunt iaculis
    libero. Aliquam elementum auctor neque ut placerat. Duis facilisis purus nec
    leo vestibulum, a imperdiet odio elementum.</p>
  </div>
<!--modal footer-->
  <div class="modal-footer">
    <a class="btn btn-default">
      Close
    </a>
  </div>
```

To make the close button work we simply add the following code (I've also cleaned up the button code too hence why it looks different to the one above):

```
<div class="modal-footer">
  <a class="btn btn-default" data-dismiss="modal">Close</a>
</div>
```

If we want to add another button to the form all we do is copy the code on the previous page and make changes where necessary, such as:

```
<div class="modal-footer">
  <a class="btn btn-success">Submit</a>
  <a class="btn btn-default" data-dismiss="modal">Close</a>
</div>
```

Because our contact form is, well a contact form(!) I have added a Submit button to it. This at present doesn't submit but that could be done in future updates and is currently just an example.

## Step 9 - A New Page

We will create a new page for our articles to be viewed in. Firstly we need to do the following:

1. Create a new **html** document.
2. Call this document "**read.html**" and save it in the **root folder**.
3. Copy and paste **ALL** the html from **index.html** to **read.html**.
4. Now we will begin removing some of the code from **read.html** as we do not need this in the article page. Remove the following section by highlighting everything from just above the **FOOTER NAVBAR** all the way up to and including the **JUMBOTRON**.

## The Setup

We are going to setup the article section ready for the content to be added by using the grid system previously used on the index page. We add the following html to do this over 9 columns, remembering 12 is our magic number.

```
<!--read articles-->
<div class="container">
  <div class="row">
    <div class="col-lg-9">
      <div class="panel panel-default">
        <div class="panel-body">
          Test
        </div>
      </div>
    </div>
  </div>
</div>
```

## Article/Page Header

We add the article/page header in the **panel-body** div as shown below:

```
<div class="panel-body">
  <!-- page header-->
  <div class="page-header">
    <h3>Article/Page Header <small>Posted on October 2014</small></h3>
  </div>
</div>
```

You will notice I have also added a new tag call **<small>** this has been used to show the date the article was created but it could be used for the author or anything else.

## Personally Style the Article/Page Header

In order to add our own personal styling to the page we will enter our own CSS into our separate style sheet. Open the CSS file we created called “**style.css**”. In here we will remove the gap at the **top of the header** and the **panel-body** by adding the following code:

```
1 .page-header {
2     margin-top: 0;
3 }
4 .panel-body {
5     padding-top: 0;
6 }
```

## Adding Our Image

We will add our own image into the page. The image can be found on my blog or you can create your own. The dimensions for the image are **1000 x 327 px**.

The image is called “**sportsHall.jpg**”, create a new folder in our root folder called “**img**” and place this image in there.



To import it into our website we add the following code into our **read.html** page, directly under the **<div>** for the Article/Page Header.

```
<!-- page header-->
<div class="page-header">
    <h3>Article/Page Header <small>Posted on October 2014</small></h3>
</div>

```

A problem we have is the image is overlapping the container for the article. So we add a **width="100%"** to the **<img>** so it fits 100% the width of the **parent**, in this case the parent is the **panel-body**.

```

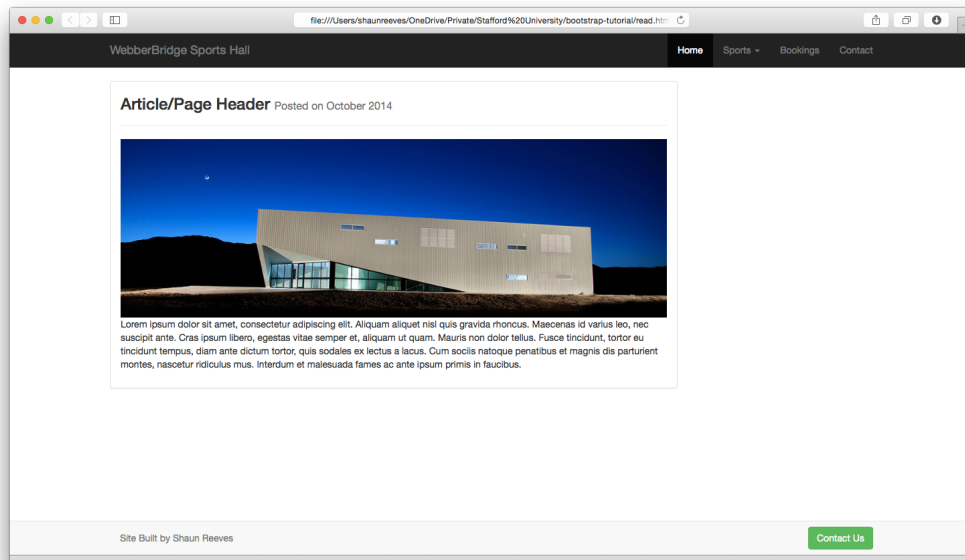
```

## Adding Our Article

You will need some more Lorem Ipsum for this section so get yourself over to the website (see Step 6 - Jumbotron) and copy a paragraph of Lorem Ipsum from the website. Add this directly under the **<img>** tag.

```
50         
51         <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam aliquet
    nisl quis gravida rhoncus. Maecenas id varius leo, nec suscipit ante. Cras ipsum libero, egestas
    vitae semper et, aliquam ut quam. Mauris non dolor tellus. Fusce tincidunt, tortor eu tincidunt
    tempus, diam ante dictum tortor, quis sodales ex lectus a lacus. Cum sociis natoque penatibus et
    magnis dis parturient montes, nascetur ridiculus mus. Interdum et malesuada fames ac ante ipsum
    primis in faucibus.</p>
```

At present our website should look similar to this.



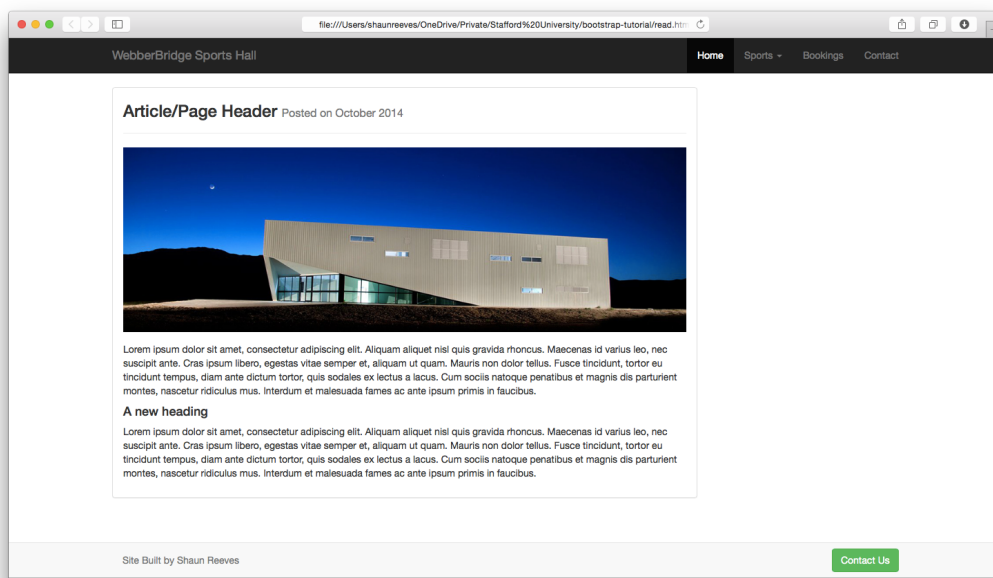
## Space the text and add more content

The text below the article picture is too close so we need to add a **padding-bottom** in our stylesheet to the image do this by adding the following in **style.css**:

```
7  .featureImg{
8      padding-bottom: 15px;
9  }
```

More content can also be added by adding a **<h4>** header and entering the title into this tag and copy and paste the **<p>** tag for Lorem Ipsum as an example below.

```
53      <h4>A new heading</h4>
54      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam aliquet
nisl quis gravida rhoncus. Maecenas id varius leo, nec suscipit ante. Cras ipsum libero, egestas
vitae semper et, aliquam ut quam. Mauris non dolor tellus. Fusce tincidunt, tortor eu tincidunt
tempus, diam ante dictum tortor, quis sodales ex lectus a lacus. Cum sociis natoque penatibus et
magnis dis parturient montes, nascetur ridiculus mus. Interdum et malesuada fames ac ante ipsum
primis in faucibus.</p>
```



## Step 10 Article Side Bar

To add the side bar to our Article page we will add another **3 column div** to the right of the main article. This will house summaries of other articles the user could read.

This 3 column div must be placed outside the previous **9 column div** we created for the main article. **This is the magic number 12.**

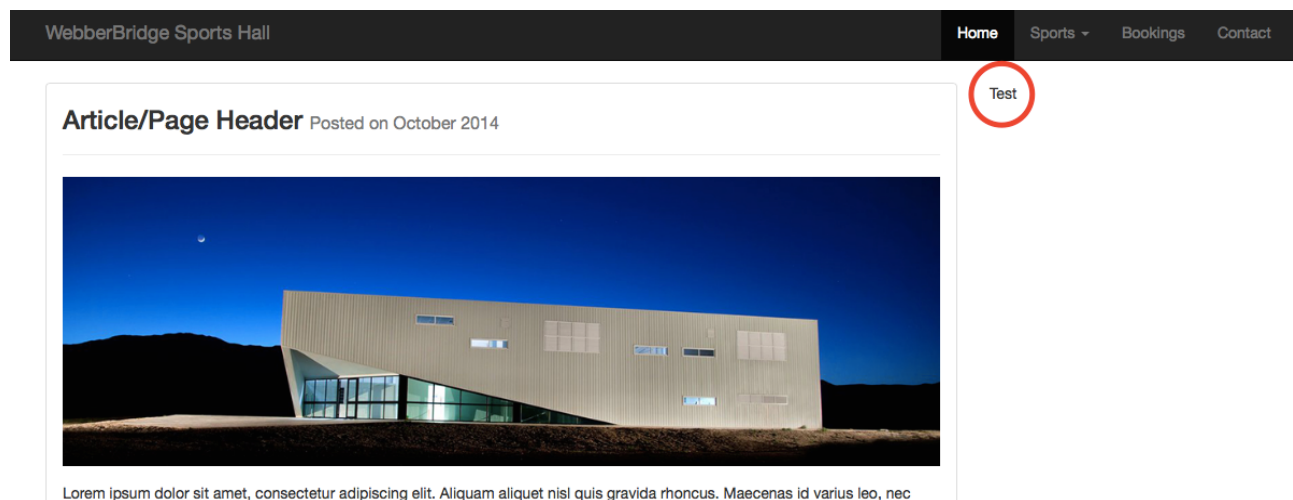
Find the closed `</div>` for `<div class="col-lg-9">`. This is usually three closed `</div>` tags up from where the **footer** starts in our example. Place "test" in here and previous website to make sure it is in the correct place.

```

59         <div class="col-lg-3">
60             Test
61         </div>
62     </div>
63 </div>
64
65 <!-- footer -->
66 <div class="navbar navbar-default navbar-fixed-bottom">

```

It should appear here (highlighted by a red circle):



To create our first article summary we add a new `<div>` with a `class="list-group-item"`. Within here we add a clickable area or an **anchor tag** and within that we add our content, as below:

```

60         <div class="list-group">
61             <a href="#" class="list-group-item">
62                 <h4 class="list-group-item-heading">Lorem Ipsum</h4>
63                 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam aliquet
nisl quis gravida rhoncus.</p>
64             </a>
65         </div>
66     </div>

```



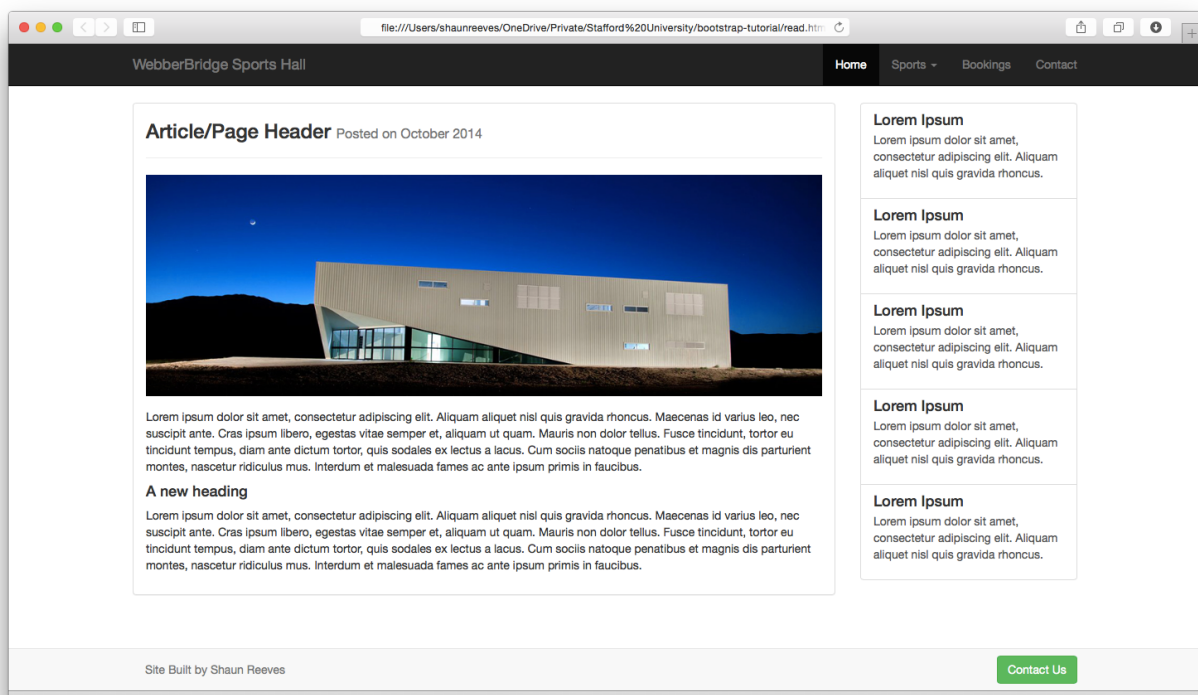
## Duplicate Article Summaries

To create more article summaries simply copy and past, starting from the `<a>` (line 61 in my example on previous page) to the closed `</a>` (line 64).

```

60         <div class="list-group">
61             <a href="#" class="list-group-item">
62                 <h4 class="list-group-item-heading">Lorem Ipsum</h4>
63                 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam aliquet
nisl quis gravida rhoncus.</p>
64             </a>
65             <a href="#" class="list-group-item">
66                 <h4 class="list-group-item-heading">Lorem Ipsum</h4>
67                 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam aliquet
nisl quis gravida rhoncus.</p>
68             </a>
69             <a href="#" class="list-group-item">
70                 <h4 class="list-group-item-heading">Lorem Ipsum</h4>
71                 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam aliquet
nisl quis gravida rhoncus.</p>
72             </a>
73             <a href="#" class="list-group-item">
74                 <h4 class="list-group-item-heading">Lorem Ipsum</h4>
75                 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam aliquet
nisl quis gravida rhoncus.</p>
76             </a>
77             <a href="#" class="list-group-item">
78                 <h4 class="list-group-item-heading">Lorem Ipsum</h4>
79                 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam aliquet
nisl quis gravida rhoncus.</p>
80             </a>
81         </div>

```



Save and preview website.

If you want to make an article active simply add the **class "active"** to the `<a>` of the article, such as:

```
<a href="#" class="list-group-item active">
```



## Step 11 Contact Form

Here we will add the contact content to the Modal contact form. This tutorial is based on front end view and not back end functionality. A PHP script could be added back end but we will not worry about that at this stage.

Please note we will be working from **index.html in the MODAL** section of our website near the **footer**. Once you have done this here you will need to copy and past the new MODAL code to each of your other pages and test them before launch.

### Clean Up

We need to start by removing the `<p>Lorem Ipsum...</p>` tag housed within the `<div class="modal-body">`

Our **modal-body** should now be empty.

```
<!--modal body-->
    <div class="modal-body">

    </div>
```

### Start the form implementation

The form needs to be inserted above the `<div class="modal-header">` tag. I have created another HTML note to highlight where I will begin the form implementation.

The **form** tag needs to be closed around the following div classes: **modal-header**, **modal-body** and **modal-footer**.

```
<div class="modal-content">
<!--modal form start-->
    <form class="form-horizontal">

        <!--modal header-->
            <div class="modal-header">
                <h4>Contact WebberBridge Sports Hall</h4>
            </div>
        <!--modal body-->
            <div class="modal-body">

            </div>
        <!--modal footer-->
            <div class="modal-footer">
                <a class="btn btn-success">Submit</a>
                <a class="btn btn-default" data-dismiss="modal">Close</a>
            </div>

    </form>
</div>
```

## Form body

The form body is implemented into the `<div class="modal-body">` tag. First we create the Form group, within here we add a form label, another div to house the input methods and then the input method itself. As follows:

### NAME

```
<!--modal body-->
<div class="modal-body">

  <div class="form-group">
    <label for="contact-name" class="col-lg-2 control-label">Name</label>
    <div class="col-lg-10">
      <input type="text" class="form-control" id="contact-name" placeholder="Full Name">
    </div>
  </div>

</div>
```

You should have a contact form that is beginning to look like this...

### EMAIL

The email section is basically a copy and pasted version of the NAME form group. So lets do this now, **copy** the `<div class="form-group">...</div>` and **paste** it below.

```
<div class="form-group">
  <label for="contact-email" class="col-lg-2 control-label">Email</label>
  <div class="col-lg-10">
    <input type="email" class="form-control" id="contact-email" placeholder="you@example.com">
  </div>
</div>
```

A few tips: we change the **label for="contact-email"** so when the website is viewed on a mobile device the keyboard with the "@" symbol pops up to make it easier for the user to enter their email address. Remember, this is Bootstrap by Twitter which is designed with a "mobile first" philosophy.

**MESSAGE**

Here we will add our **message** section, so copy and paste again like we did for **EMAIL** and make the relevant adjustments (“for” and “label”). After here we need to make some slight adjustments to the **<label>** and then **delete** the entire **<input>** tag.

We will then add a new tag called **<textarea>**. We add a class to this called **class=“form-control”** to make the style match the rest of the form. We can also control the height of the textarea by adding the number of rows after the class. See below for the example:

```
<div class="form-group">
  <label for="contact-msg" class="col-lg-2 control-label">Message</label>
  <div class="col-lg-10">
    <textarea class="form-control" rows="8"></textarea>
  </div>
</div>
```

**Adding your submit button**

To add the “send” button simply delete the previous “Submit” button we created and below the “Close” button add the following code:

```
<!--modal footer-->
  <div class="modal-footer">
    <a class="btn btn-default" data-dismiss="modal">Close</a>
    <button class="btn btn-success" type="submit">Send</button>
  </div>
```

The **type=“submit”** is required so the form can be submitted to the PHP form for processing on the server and an email can be generated with the content of the form.

**Adding your contact form to other pages**

Simply copy your entire **MODAL** code and paste it over the entire **MODAL** code in your other pages. For our example this would be the **read.html** page.

**Step 12 - Problem Footer Quick Fix**

The footer can hang over your content. If this is the case go to your **style.css** and add the following code to the **body** tag:

```
body {
  padding-bottom: 80px;
}
```

Now you should have a **two page, responsive** website built using **Bootstrap (the Twitter framework)**.