

Bootstrap Basics

In recent years, Bootstrap has become the most popular front-end framework. It's a clean, intuitive, and powerful mobile-first front-end framework that makes web development go faster and easier. HTML, CSS, and JavaScript are all used.

This training lesson will teach you the fundamentals of the Bootstrap Framework, which you can use to easily develop web projects.

Why to Learn Bootstrap?

Mobile first approach: Instead of having Mobile First styles in distinct files, the Bootstrap framework includes them across the entire library.

Browser Support: All major browsers are compatible with it.

Easy To Get Started: Anyone with a basic understanding of HTML and CSS can get started with Bootstrap. The official Bootstrap site also provides excellent documentation.

Responsive Design: The responsive CSS in Bootstrap adapts to Desktops, Tablets, and Mobiles..

- Provides a simple and consistent way for creating a developer interface.
- It comes with a number of attractive and effective built-in components that are simple to personalize.
- It also has a web-based customizing option.
- It's also open-source, which is a huge plus.

Applications Of Bootstrap:

Scaffolding: Bootstrap comes with a basic framework that includes a Grid System, link styles, and a background. The section Bootstrap Basic Structure goes over this in depth.

CSS: Global CSS settings, basic HTML components stylized and improved with extensible classes, and a complex grid system are all included in Bootstrap. The section Bootstrap with CSS goes over this in depth.

Components: Over a dozen reusable components are included in Bootstrap, including iconography, dropdowns, navigation, notifications, pop-overs, and more. The section Layout Components goes through this in depth.

JavaScript Plugins: Over a dozen bespoke jQuery plugins are included in Bootstrap. You may easily incorporate all of them or just one at a time. The section Bootstrap Plugins goes over this in depth.

Setup and Overview

In order to use Bootstrap, you first need to integrate it into your development environment, aka web page. For that, you have two different possibilities: load it remotely or download and use Bootstrap locally. However, for both, you first need something to load it into.

1. Create an HTML Page

As a first step in this Training, we will create a simple HTML template as a base where we will use Bootstrap. For that, the first thing you want to do is create a folder on your computer or server for the project files. In this case, we will simply call it bootstrap.

Here, create a new text file and call it index.html. Open it with a text editor of your choice (e.g. sublime, vscode) and then paste the code below into it.

```

!DOCTYPE html>
<html lang="en">
  <head>
    <title>Bootstrap Training Sample Page</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">

  </head>

  <body>
    .....
  </body>

</html>

```

Don't forget to save your file before moving on.

2a. Load Bootstrap via CDN

As already explained, Bootstrap consists mainly of style sheets and scripts. As such, they can be loaded in the header and footer of your web page like other assets such as custom fonts. The framework offers a CDN (content delivery network) access path for that. You can find it on the [Bootstrap download page](#) when you scroll down.

To get Bootstrap into your page, simply paste the code below into the <head> section of your template.

```

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/boot
strap.min.css" rel="stylesheet"
integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3y
D65VohhpuuCOMLASjC" crossorigin="anonymous">

```

Be sure to take the actual link from the download page in order to make sure you are using the latest version of Bootstrap.

When you now save the file, any browser that opens it will automatically load the Bootstrap assets.

Using the remote method is a good idea as many users will already have the file in the cache of their browser from loading other Bootstrap-based websites. If that is the case, they won't have to reload it when coming to your site, leading to faster page loading time. As a consequence, this is the recommended method for live sites.

However, for experimenting and development, or if you want to be independent of an Internet connection, you can also get your own copy of the files. This is what we're doing for this Bootstrap Training because it also results in less code we need to post

2b. Host Bootstrap Locally

An alternative way to set up Bootstrap is to download it to your hard drive and use the files locally. You find download options in the same place as the links to the remote version. Be sure to get the compiled CSS and JS files. You don't need the source files. Once you have finished downloading it, unzip the file and copy its contents into the same directory as index.html. After that, you can load the Bootstrap CSS into your project like this:

```
<link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">
```

You will notice that this includes the file path at which to find the Bootstrap file. In your case, make sure your path corresponds to your actual setup. For example, the names of the directories might differ if you downloaded a different version of Bootstrap.

4. Load Bootstrap JavaScript

The last step in setting up Bootstrap is to load the Bootstrap JavaScript library. It's included in the downloaded version of the framework and you also find links to remote sources in the same place as mentioned above. However, we will load it in a different place than the style sheet. Instead of the header, it goes into the page footer, right after the call for jQuery.

You can call it remotely like this:

```
<script  
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootst  
rap.bundle.min.js"
```

```
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>
```

Or locally like so:

```
<script src="bootstrap/js/bootstrap.min.js"></script>
```

5. Put it All Together

If you have followed the steps above correctly, you should end up with a file that looks like this for the remote solution:

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

    <title>Bootstrap Training Sample Page</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap
.min.css" rel="stylesheet"
integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3yD65Vo
hhpuuCOMLASjC" crossorigin="anonymous">

  </head>
  <body>

  <!-- rest of your code -->

    <script src="https://code.jquery.com/jquery-3.5.1.min.js"
integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0="
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.b
undle.min.js"
```

```
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+J
cXn/tWtIaxVXM" crossorigin="anonymous"></script>

</body>

</html>
```

And here is what you should have for the version that loads all assets locally:

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

    <title>Bootstrap Training Sample Page</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <link rel="stylesheet"
href="bootstrap/css/bootstrap.min.css">

  </head>
  <body>

    <script src="jquery-3.5.1.min.js"></script>
    <script src="bootstrap/js/bootstrap.min.js"></script>

  </body>

</html>
```

Design Your Landing Page

At the moment, when you open your sample site in a browser, you should simply see a blank page. Time to change that. In order to teach you Bootstrap, in this Training we will build a landing page with different elements so we can show you many use cases of the front-end framework.

1. Add a Navigation Bar

The first thing we want to do is add a navigation bar to the top of the page. Doing so allows your visitors to get around your site and discover the rest of your pages.

For that, we will use the **navbar** class. This is one of the [default elements](#) of Bootstrap, which you will see a lot of in the course of this Training. It creates a navigation bar that is responsive by default and will automatically collapse on smaller screens. It also offers built-in support for adding branding, color schemes, spacing, and other components.

You can start by posting this just after the `<body>` tag:

```
<nav class="navbar navbar-expand-md">
  <a class="navbar-brand" href="#">Logo</a>
  <button class="navbar-toggler navbar-dark" type="button"
data-toggle="collapse" data-target="#main-navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="main-navigation">
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link" href="#">Home</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">About</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Contact</a>
      </li>
    </ul>
  </div>
</nav>
```

Some Explanation of the Code

The markup above is probably full of unknown CSS classes. Here is what they mean:

- **navbar-expand-md** — This denotes at which point the navigation bar expands from vertical or hamburger icon to a full-size horizontal bar. In this case, we have set it to medium screens (md), which, in Bootstrap, is anything greater than 768px.
- **navbar-brand** — This is used for your website branding. You can also include a logo image file here.
- **navbar-toggler** — Denotes the toggle button for the collapsed menu. The piece `data-toggle="collapse"` defines that this will turn to a hamburger menu, not to drop-down, which is the other option. It's important that you define a `data-target` with a CSS id (defined by the #) and wrap a `<div>` with the same name around the actual navbar element.
- **navbar-toggler-icon** — As you can probably guess, this creates the icon users click on to open the menu on smaller screens.
- **navbar-nav** — The class for the `` list element that holds the menu items. The latter are denoted with `nav-item` and `nav-link`.

Why are we explaining this so much?


Because that is the point of Bootstrap. It comes with all of these standards that allow you to quickly create elements with some HTML and CSS. You also don't have to worry about layout, as everything is already set up within Bootstrap. Plus, it's all mobile responsive out of the box! Are you beginning to see how helpful this is?

The above is enough to add a navigation bar to your site. However, at the moment, it still looks like very little.

[Logo](#) [Home](#) [About](#) [Contact](#)

That's because it doesn't have a lot of styling attached to it. While you are able to add default colors (for example, by giving the navbar a class like bg-dark navbar-dark), we instead want to add our own.

```
<nav class="navbar navbar-expand-md bg-dark navbar-dark">
```



Logo Home About Contact

2. Include Custom CSS

Fortunately, if you want to change the default styling, you don't have to wade through a large library of style sheets and make the changes by hand. Instead, you are able to add your own CSS files which you can use to overwrite existing styling.

For that, simply create a blank css file with your text editor and call it styles.css. Save it in a new folder called css, then add it to the head section of your Bootstrap site like this:

```
<link rel="stylesheet" type="text/css" href="css/styles.css">
```

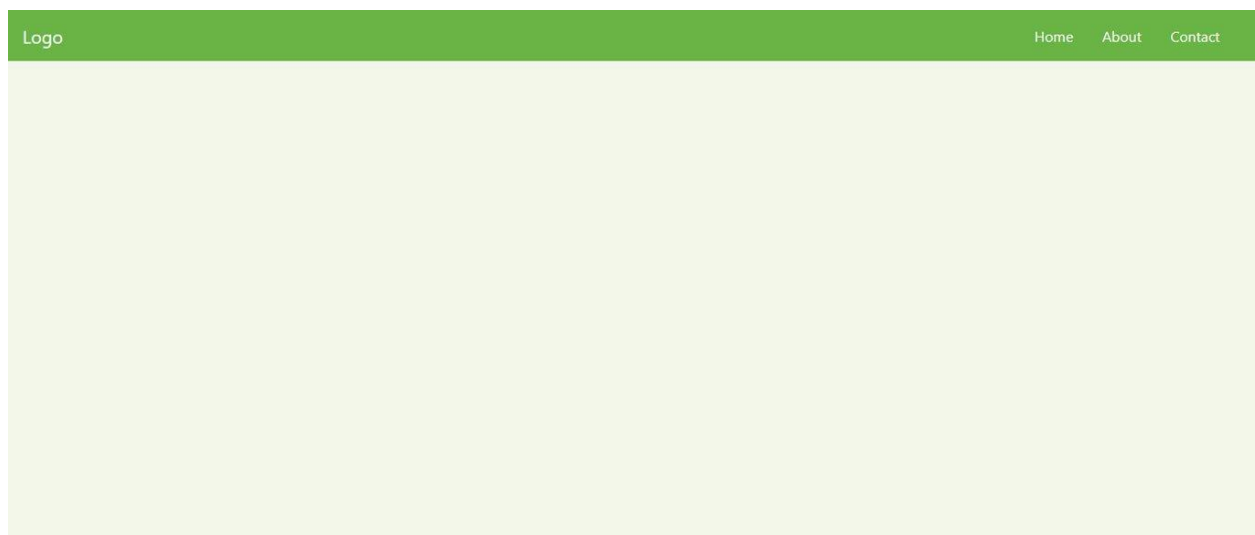
```
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
<link rel="stylesheet" type="text/css" href="css/styles.css">
```

This is the code for a style sheet that resides in the main directory. If you decide to place your inside the css folder, be sure to include the correct path in the link.

From here, you are able to add custom CSS to your site. For example, to style the page background as well as the navigation bar and its elements, you could use markup like this:

```
body {
    padding: 0;
    margin: 0;
    background: #f2f6e9;
}
.navbar {
    background: #6ab446;
}
.nav-link,
.navbar-brand {
    color: #fff;
    cursor: pointer;
}
.nav-link {
    margin-right: 1em !important;
}
.nav-link:hover {
    color: #000;
}
.navbar-collapse {
    justify-content: flex-end;
}
```

And here is the result:



3. Create a Page Content Container

After the navigation bar, the next thing you want is a container for the page content. This is really easy in Bootstrap as all you need for it is this underneath the navbar tag:

```
<header class="page-header header container-fluid">  
</header>
```

Notice the container-fluid class. This is another one of those default Bootstrap classes. Applying it to the div element automatically applies a bunch of CSS to it.

The -fluid part makes sure the container stretches across the entire width of the screen. There's also just **container**, which is a class that has fixed widths applied to it, so there will always be space on both sides of the screen.

However, if you now reload the page, you still won't see anything (unless you use the developer tools). That's because you only created an empty HTML element. This will start changing now

.

4. Add a Background Image and Custom JavaScript

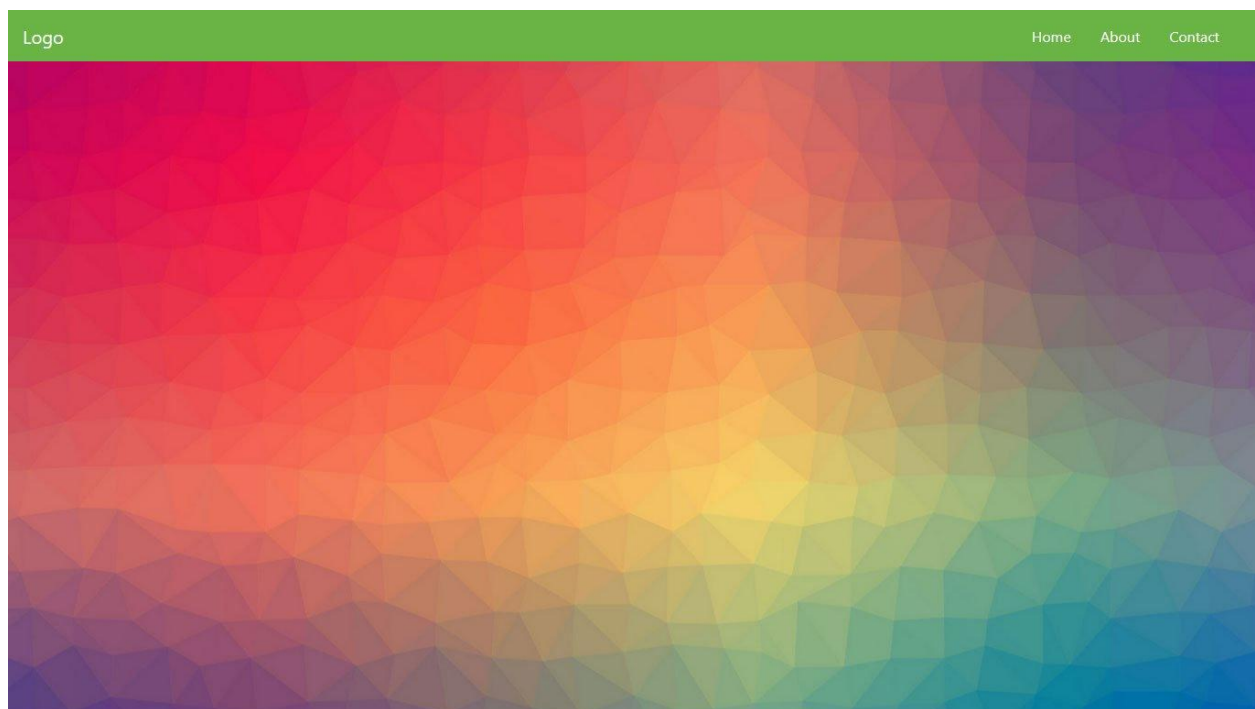
As the next step in this Bootstrap Training, we want to include a full-screen background image for our landing page header.

```
html, body {  
    height: 100%;  
    margin: 0;  
}
```

Then, the only thing that's left is to set a background image. You can do this like so inside main.css:

```
.header {  
    background-image: url('images/header-background.jpg');  
    background-size: cover;  
    background-position: center;  
    position: relative;  
    height: 100%;  
}
```

If you place an image of sufficient size at the location specified by the path above, you will achieve a result similar to this:



5. Add an Overlay

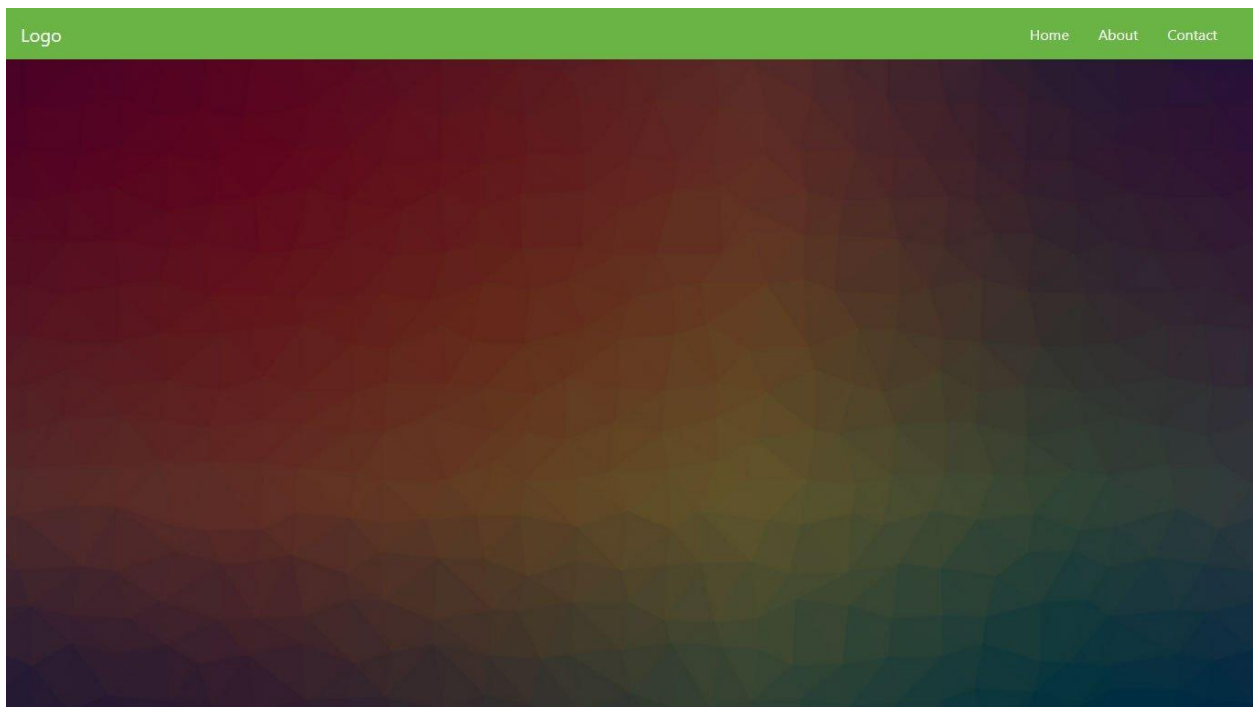
To make the background image extra stylish, we will also add an overlay. For that, create another div inside the `<header>` element you just created earlier.

```
<div class="overlay"></div>
```

Then, you can add the following in your custom CSS file:

```
.overlay {  
    position: absolute;  
    min-height: 100%;  
    min-width: 100%;  
    left: 0;  
    top: 0;  
    background: rgba(0, 0, 0, 0.6);  
}
```

This will create this nice overlay for the image you input earlier:



6. Include a Page Title and Body Text

As a next step, you probably want to add a page title in the form of a heading plus some body text. That way, your visitors will know immediately which site they are on and what they can expect from it.

To create those, simply add this snippet inside the container you set up in the last step, below the overlay:

```
<div class="description">
  <h1>Welcome to the Landing Page!</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Quisque interdum quam odio, quis placerat ante luctus eu.
Sed aliquet dolor id sapien rutrum, id vulputate quam iaculis.
Suspendisse consectetur mi id libero fringilla, in pharetra sem
ullamcorper.</p>
</div>
```

After that, add the following markup to main.css.

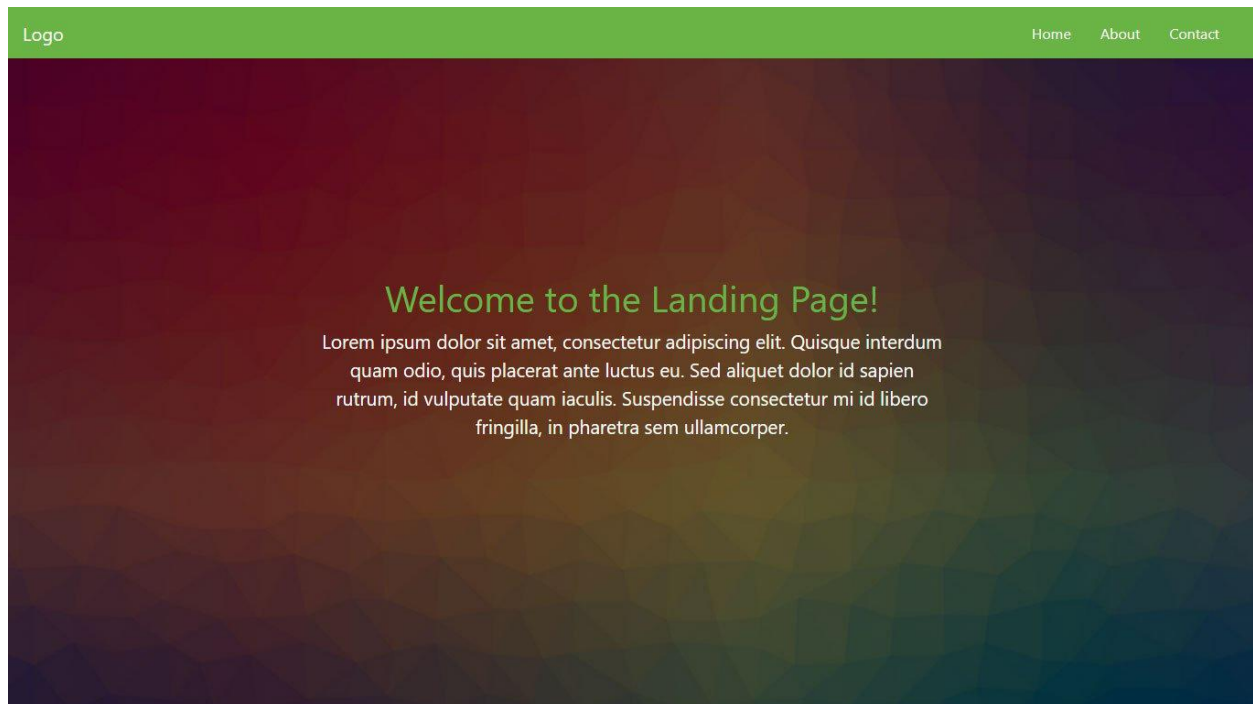
```
.description {
  left: 50%;
  position: absolute;
  top: 45%;
  transform: translate(-50%, -55%);
  text-align: center;
}
```

```

.description h1 {
    color: #6ab446;
}
.description p {
    color: #fff;
    font-size: 1.3rem;
    line-height: 1.5;
}

```

When you do, the landing page now looks like this:



It's really starting to come together, isn't it?

7. Create a CTA Button

No landing page is complete without a call to action, most often in the form of a button. For that reason, we would be remiss not to include how to create one in this Bootstrap Training.

Luckily, the framework offers plenty of tools to create buttons quickly and easily. You can find a lot of examples [here](#).

In our case, we add the following markup right below the page content inside the `<description>` container:

```

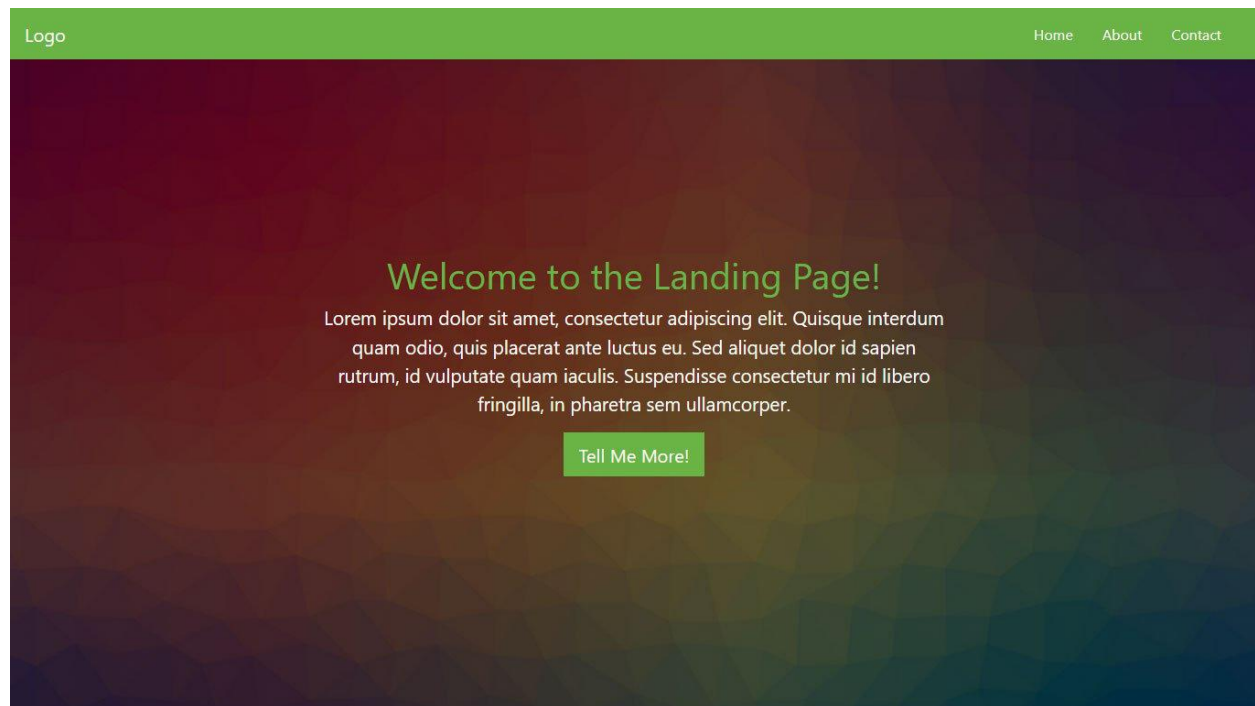
<button class="btn btn-outline-secondary btn-lg">Tell Me
More!</button>

```

In addition to that, we add this CSS to main.css:

```
.description button {
    border:1px solid #6ab446;
    background:#6ab446;
    border-radius: 0;
    color:#fff;
}
.description button:hover {
    border:1px solid #fff;
    background:#fff;
    color:#000;
}
```

After saving and reloading, it looks like this:



8. Set Up a Three-Column Section

We can already be quite satisfied with how things are shaping up. However, we are not done with the page yet. Next up, we want to create three columns below the main content for additional information. This is a specialty of Bootstrap since it plays to its strength, creating a grid, and no Training is complete without it. Here's how to do that in this case:

```
<div class="container features">
  <div class="row">
    <div class="col-lg-4 col-md-4 col-sm-12">
```

```

        <h3 class="feature-title">Lorem ipsum</h3>
        
        <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Quisque interdum quam odio, quis placerat ante
luctus eu. Sed aliquet dolor id sapien rutrum, id vulputate
quam iaculis.</p>
    </div>

    <div class="col-lg-4 col-md-4 col-sm-12">
        <h3 class="feature-title">Lorem ipsum</h3>
        
        <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Quisque interdum quam odio, quis placerat ante
luctus eu. Sed aliquet dolor id sapien rutrum, id vulputate
quam iaculis.</p>
    </div>

    <div class="col-lg-4 col-md-4 col-sm-12">

    </div>
</div>
</div>

```

The first thing you will notice is the row element. You need this whenever creating columns to act as a container for the grid.

As for the columns, they all have several classes: col-lg-4, col-md-4, and col-sm-12. These denote that we are dealing with columns and the size they will take on on different screens.

To understand this, you need to know that in a Bootstrap grid, all columns in one row always add up to the number twelve. So, giving them the classes above means that they will take up one-third of the screen on large and medium screens ($12/4=3$) but the entire screen on small devices (twelve out of twelve columns).

You will also notice that we included images and added the .img-fluid class to them. This is to make them responsive so that they scale along with the screen that the page is viewed on.

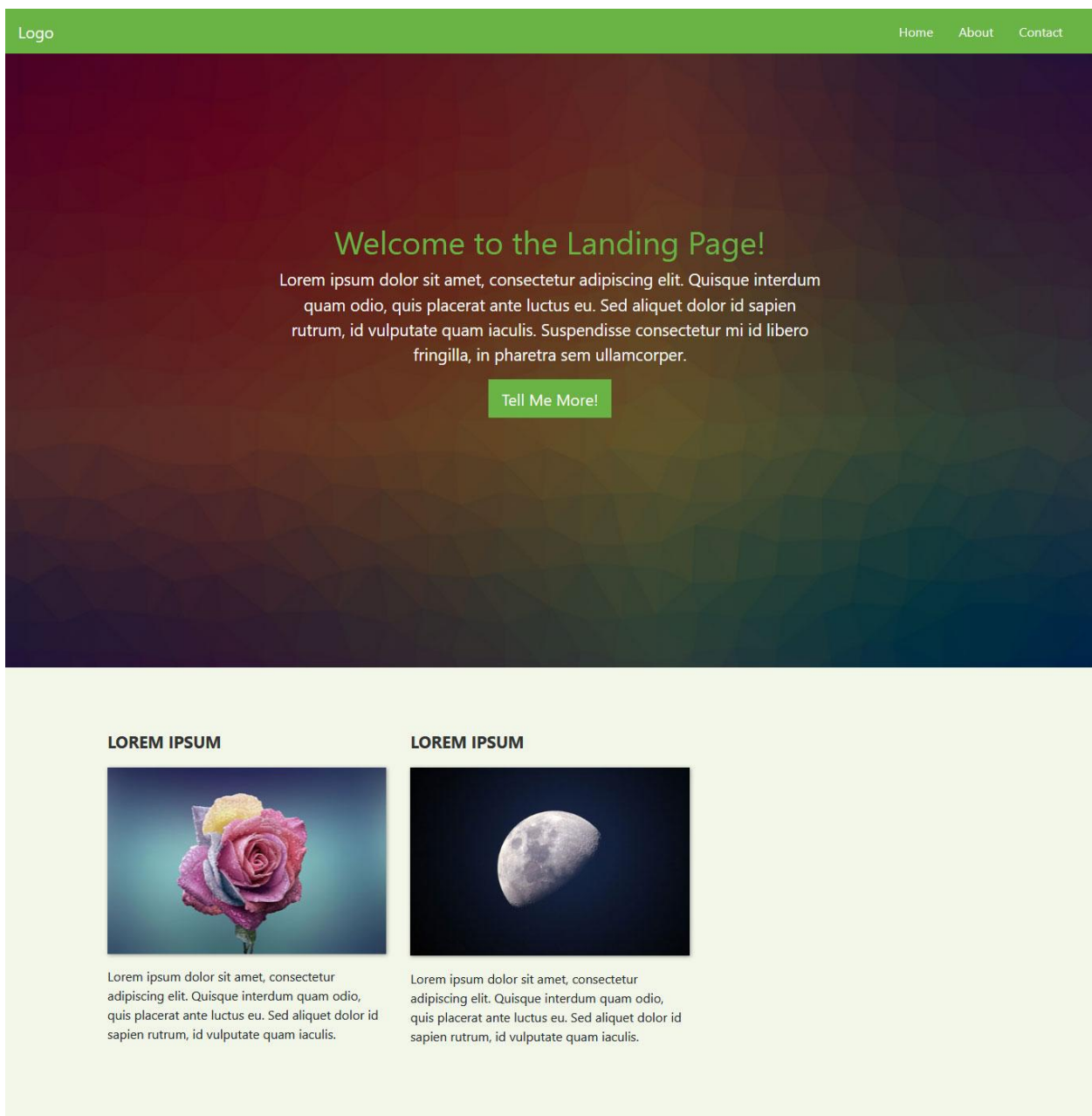
In addition to that, you can include the following styling in the styles.css file:

```
.features {
    margin: 4em auto;
    padding: 1em;
    position: relative;
}

.feature-title {
    color: #333;
    font-size: 1.3rem;
    font-weight: 700;
    margin-bottom: 20px;
    text-transform: uppercase;
}

.features img {
    -webkit-box-shadow: 1px 1px 4px rgba(0, 0, 0, 0.4);
    -moz-box-shadow: 1px 1px 4px rgba(0, 0, 0, 0.4);
    box-shadow: 1px 1px 4px rgba(0, 0, 0, 0.4);
    margin-bottom: 16px;
}
```

When added below the main content and saved, it looks like this:



By the way, if you don't want to manually set the box shadow on your elements via CSS, you can also assign Bootstrap's pre-set classes for that. These are `.shadow-sm`, `.shadow`, and `.shadow-lg`. More information [here](#).

9. Add a Contact Form

You will notice that one of the new column fields is still empty. This was on purpose because we want to add a contact form to it. This is a very normal practice for landing pages to allow visitors to get in touch.

Creating a contact form in Bootstrap is quite easy:

```
<h3 class="feature-title">Get in Touch!</h3>
```

```

<div class="form-group">
  <input type="text" class="form-control" placeholder="Name" name="">
</div>
<div class="form-group">
  <input type="email" class="form-control" placeholder="Email Address"
name="email">
</div>
<div class="form-group">
  <textarea class="form-control" rows="4"></textarea>
</div>
<input type="submit" class="btn btn-secondary btn-block" value="Send" name="">

```

Pay attention to these CSS classes:

- form-group — Used to wrap around form fields for formatting.
- form-control — Denotes form fields such as inputs, text areas, etc.

There is a lot more you can do with forms, which you can find out in the [documentation](#).

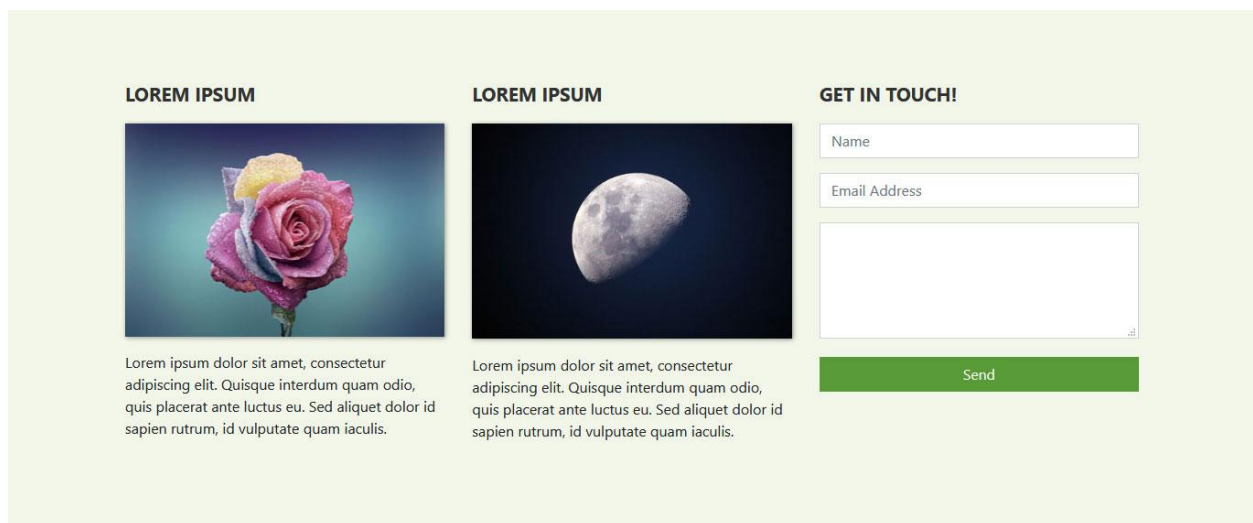
However, for demonstration purposes, the above is enough. Place it inside the remaining empty column and then add this styling in main.css:

```

.features .form-control,
.features input {
  border-radius: 0;
}
.features .btn {
  background-color: #589b37;
  border: 1px solid #589b37;
  color: #fff;
  margin-top: 20px;
}
.features .btn:hover {
  background-color: #333;
  border: 1px solid #333;
}
.form-group {
  margin-bottom: 5px;;
}

```

When you do, you get a form like this:



10. Include a Team Section

Card layouts are another thing Bootstrap excels at. Its custom CSS classes make creating a Pinterest-like layout a cinch.

In the following, we will use this to set up a team section, where we display team members and their positions in the company.

However, we want to make this section a different color. For that reason, we first include another `<div>` element as its background. Inside, there is another container with a custom class for more customizability, followed by a row.

```
<div class="background">
  <div class="container team">
    <div class="row">

      </div>
    </div>
  </div>
```

Creating cards is easy, simply give an element the class `card`. You can also use grid classes for sizing it and determining how many cards appear in one row. Plus, since we want all the content in the card to be centered, we will also add the Bootstrap class `text-center` to it.

```
<div class="card col-lg-3 col-md-3 col-sm-4 text-center">
</div>
```

Cards can contain different elements.

In our case,

- we want an image at the top,
- followed by a title (the name of the person) and
- a description that says what they do.

As is probably no surprise, Bootstrap offers custom classes for all of that.

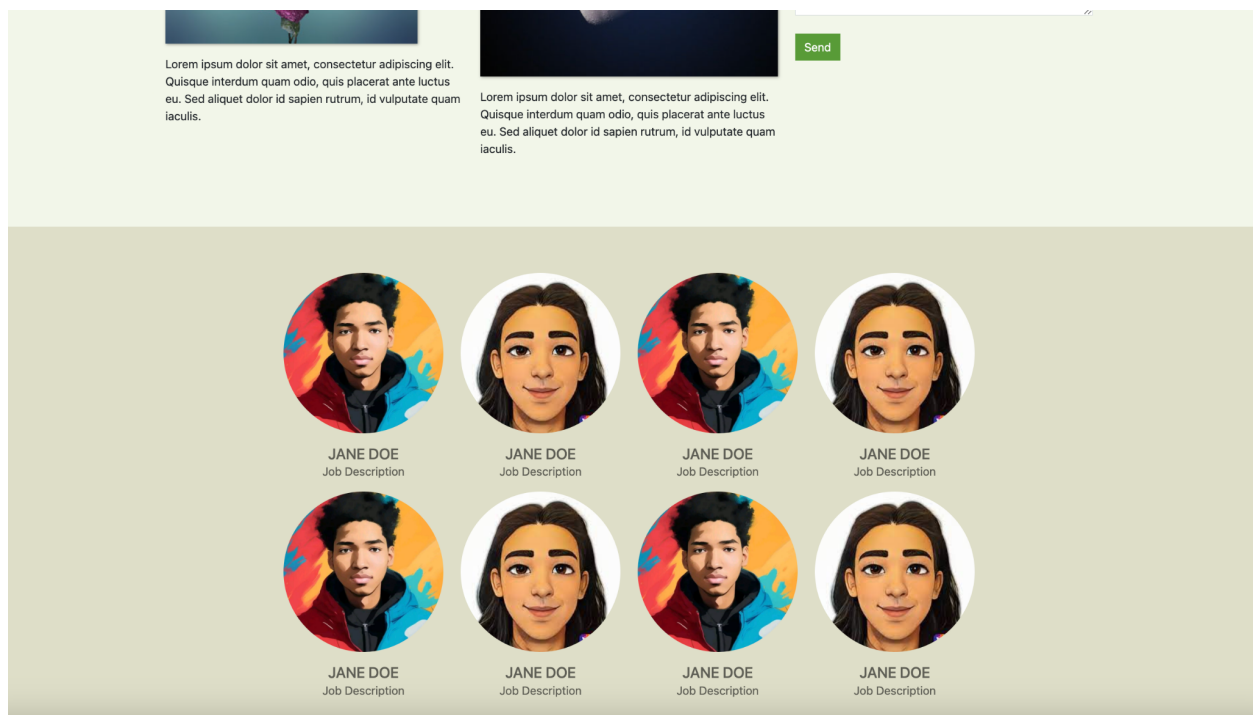
For images, we will use the `card-img-top` class to position an image at the roof of the card.

We will also give it the `rounded-circle` class to make the image round.

After that, we create a new element with the class `card-body` that contains a heading and paragraph with `card-title` and `card-text` classes respectively. Here's what it looks like all together:

```
<div class="card col-lg-3 col-md-3 col-sm-4 text-center">
  
  <div class="card-body">
    <h4 class="card-title">Jane Doe</h4>
    <p class="card-text">Job Description</p>
  </div>
</div>
```

When we now place it inside the row element, replicate it eight times, and place image files in the respective locations, we get this result.



Of course, to make it look like this, we also included some custom CSS. Here is the markup you can add to your style sheet:

```
.background {
    background: #dedec8;
    padding: 4em 0;
}

.team {
    color: #5e5e55;
    padding: 0 180px;
}

.team .card-columns {
    -webkit-column-count: 4;
    -moz-column-count: 4;
    column-count: 4;
}

.team .card {
    background: none;
    border: none;
}

.team .card-title {
    font-size: 1.3rem;
    margin-bottom: 0;
    text-transform: uppercase;
}
```

11. Create a Two-Column Footer

Alright, we are now getting towards the end of the Bootstrap Training mini Project. The last thing we want to add to our landing page is a footer section with two columns. By now, this shouldn't pose much of a problem for you anymore.

```
<footer class="page-footer">
  <div class="container">
    <div class="row">
      <div class="col-lg-8 col-md-8 col-sm-12">
        <h6 class="text-uppercase font-weight-bold">Additional
Information</h6>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Quisque interdum quam odio, quis placerat ante luctus eu. Sed
aliquet dolor id sapien rutrum, id vulputate quam iaculis.</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Quisque interdum quam odio, quis placerat ante luctus eu. Sed
aliquet dolor id sapien rutrum, id vulputate quam iaculis.</p>
      </div>
      <div class="col-lg-4 col-md-4 col-sm-12">
        <h6 class="text-uppercase font-weight-bold">Contact</h6>
        <p>1640 Riverside Drive, Hill Valley, California
        <br/>info@mywebsite.com
        <br/>+ 01 234 567 88
        <br/>+ 01 234 567 89</p>
      </div>
    </div>
    <div class="footer-copyright text-center">© 2023 Copyright:
MyWebsite.com</div>
</footer>
```

Besides the usual grid markup, this section highlights a few possibilities to modify typography with Bootstrap:

- text-uppercase
- font-weight-bold
- text-center

It should be pretty clear from the names of the classes that they do. You can find more information about Bootstrap and typography [here](#).

In addition to the above, you can use styling like this:

```
/* footer */
```

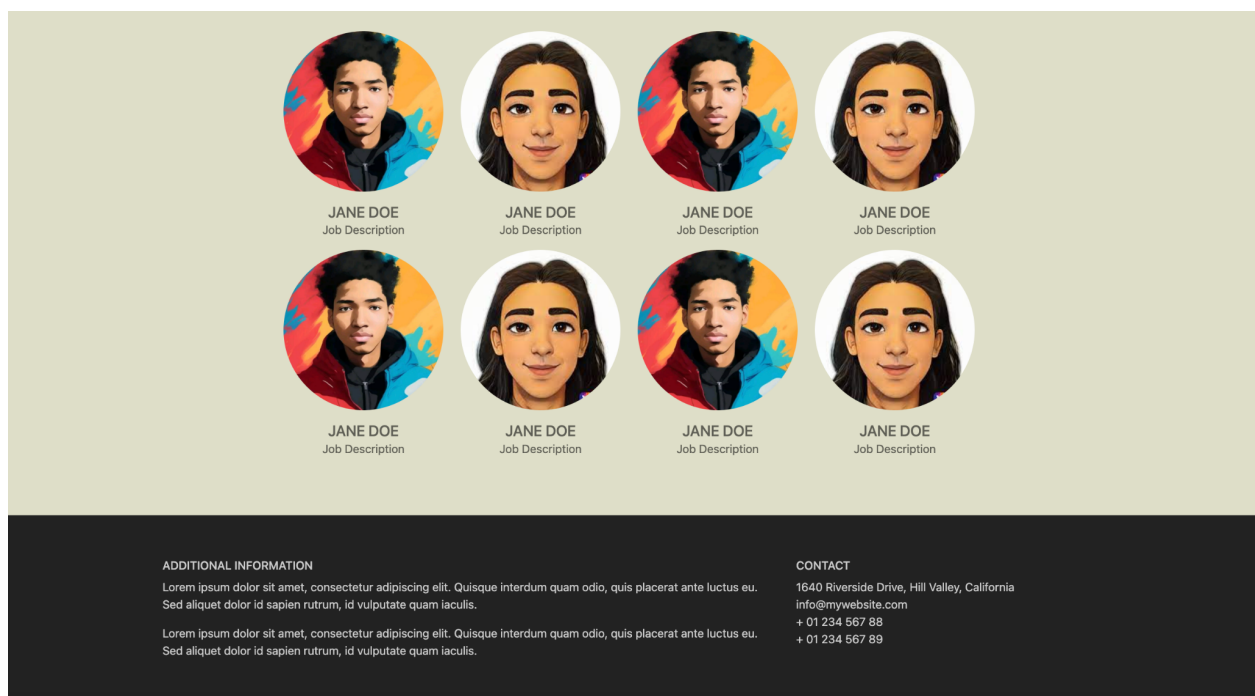
```

.page-footer {
  background-color: #222;
  color: #ccc;
  padding: 60px 0 30px;
}

.footer-copyright {
  color: #666;
  padding: 40px 0;
}

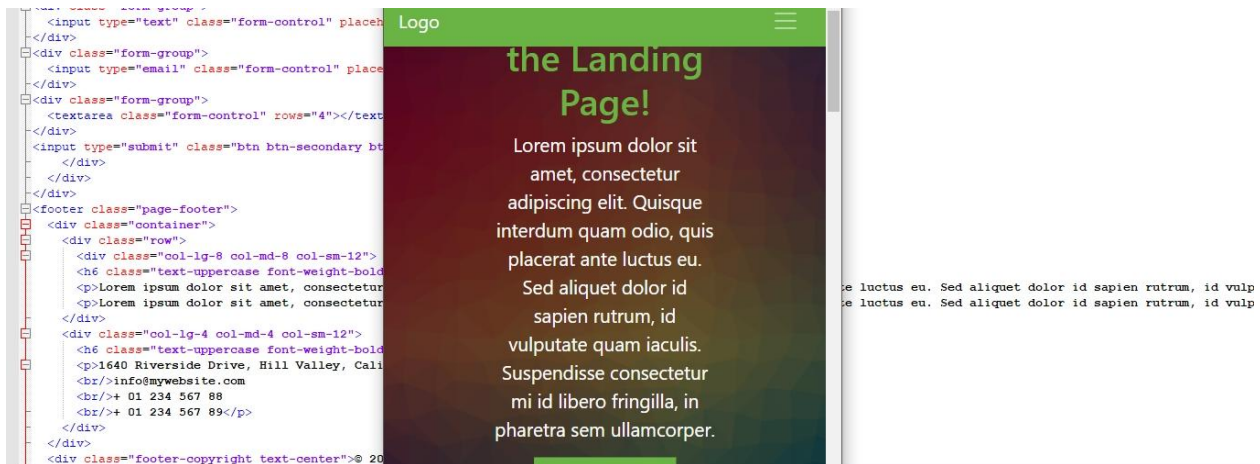
```

This results in a beautiful footer that looks like so:



12. Add Media Queries

The page is basically ready by now. It is also fully responsive. However, in the mobile view of the browser, some sections don't come out quite right yet. For example, here's what the header image looks like on mobile devices:



However, no worries, you can correct that quite easily with a simple media query.

Unless you are using SASS for compiling your Bootstrap site, these work the same way as in other instances. You just need to keep in mind the preset breakpoints included in Bootstrap.

As a consequence, to correct the above problem, you can simply include a piece of code like this:

```
@media (max-width: 991.98px) {

    .team {
        padding: 0 15px;
    }

}

@media (max-width: 575.98px) {

    .description {
        left: 0;
        padding: 0 15px;
        position: absolute;
        top: 10%;
        transform: none;
        text-align: center;
    }

    .description h1 {
        font-size: 2em;
    }

    .description p {
        font-size: 1.2rem;
    }

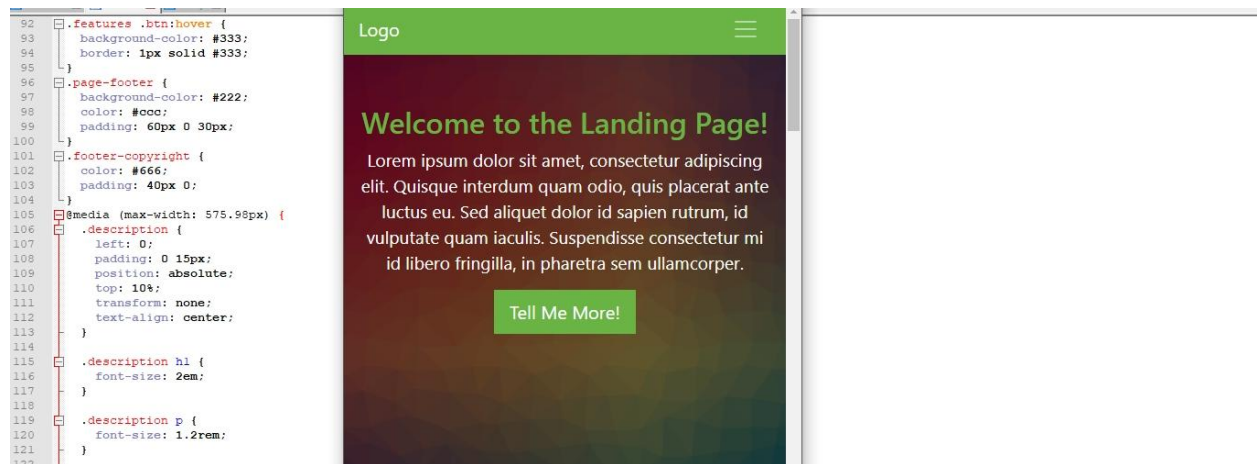
    .features {
        margin: 0;
    }

    .team {
        max-width: 60%;
    }

}
```

```
}
```

After that, everything is as it should be:



If you download the files below, you will also see some other markup in the media query section to style some of the other parts of the landing page on smaller screens.

13. Create Additional Pages

So far, we have built only a single page for our Bootstrap website. However, you might remember that, when setting up the navigation menu, we included several menu items. Yet, currently, these links don't lead anywhere when you click them.

If you want to change that and add additional pages to your Bootstrap website, there's nothing easier than that. The most straightforward way is to simply go to make a copy of `index.html` and give it a different name, typically that of the other page you want to create.

TASK

So Go ahead and use the bootstrap components from the site to

- create a similar page to what we have done on our portfolio project for the previous training courses on HTML, CSS, and JS , and link that to every team member card so when the card is clicked it goes to the Portfolio page;
- An about us page



FIRSTNAME LASTNAME

Web Developer At XYZCompany

{{ Pause and ponder in the silence }}

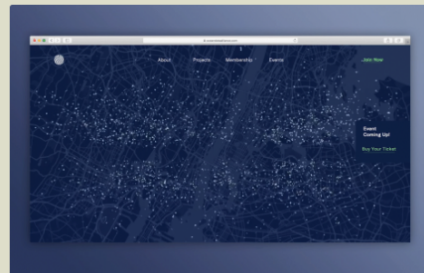
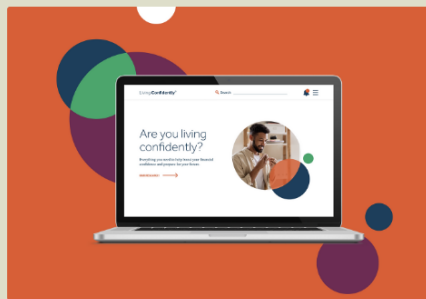
I am

An engineer
A reader
A hobby cook

I like to

Meet new people
Play guitar
Eat Icecream

My Projects



ADDITIONAL INFORMATION

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque interdum quam odio, quis placerat ante luctus eu. Sed aliquet dolor id sapien rutrum, id vulputate quam iaculis.

CONTACT

1640 Riverside Drive, Hill Valley, California
info@mywebsite.com