**SSE3308 Lab Assignment 2**

Semester 2 2022/2023

Duration: 1 Week

**Building a simple Bootstrap**

Bootstrap, which is the topic of this tutorial, is a front-end framework that helps you build mobile responsive websites more quickly and easily. First developed by Twitter, it now powers anything from web applications to WordPress themes. Famous users include Spotify, LinkedIn, among others. The framework is also completely free, versatile, and intuitive.

With Bootstrap, you can conjure complex web pages from standard HTML and customize them to your needs. It also comes with additional functionalities such as carousels, buttons, popups, and more.

Last, but not least, Bootstrap gives you a lot of shortcuts for creating web pages that will save you time and energy. All you need is a basic understanding of [HTML and CSS](https://websitesetup.org/website-coding-html-css/) to create web pages that are responsive, mobile-first, and compatible with all modern browsers.

# Bootstrap Tutorial Step 1: 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.

## Create an HTML Page

As a first step in this tutorial, 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. Atom, [Notepad++](https://notepad-plus-plus.org/)) and then paste the code below into it.

!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Tutorial 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.

## 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)](https://websitesetup.org/best-cdn-providers/) access path for that. You can find it on the [Bootstrap download page](https://getbootstrap.com/docs/4.5/getting-started/download/) when you scroll down.

To get Bootstrap into your page, simply paste the code below into the <head> section of your template or find any newest Bootstrap code online.

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" 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.

## Include jQuery

In order to get the full functionality of Bootstrap, you also need to load the [jQuery library](https://websitesetup.org/jquery-cheat-sheet/). Here, too, you have the possibility to load it remotely or host it locally.

You find the link to the latest version of the jQuery library [here](https://code.jquery.com/) (click on any of the links to get the remote URL). You can use it to load the library into your page by putting the line of code below right before where it says </body> on your page.

<script src="https://code.jquery.com/jquery-3.5.1.min.js" integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0=" crossorigin="anonymous"></script>

## 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@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-w1Q4orYjBQndcko6MimVbzY0tgp4pWB4lZ7lr30WKz0vr/aWKhXdBNmNb5D92v7s" crossorigin="anonymous"></script>

## 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 Tutorial Sample Page</title>

<meta charset="utf-8">

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

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

</head>

<body>

<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@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-w1Q4orYjBQndcko6MimVbzY0tgp4pWB4lZ7lr30WKz0vr/aWKhXdBNmNb5D92v7s" crossorigin="anonymous"></script>

</body>

</html>

If that looks like what’s in your file and you saved your work, you are now ready to move on to the next step.

# Bootstrap Tutorial Step 2: Design Your Landing Page

Alright, that was, admittedly, a lot of preparation work. However, it wasn’t very hard, was it? Plus, now the fun begins.

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 tutorial we will build a landing page with different elements so we can show you many use cases of the front-end framework.

## 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](https://getbootstrap.com/docs/4.5/components/navbar/) of Bootstrap, which you will see a lot of in the course of this tutorial. 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, 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 <ul> 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.



That’s because it doesn’t have a lot of styling attached to it. While you are able to add [default colors](https://getbootstrap.com/docs/4.5/utilities/colors/) (for example, by giving the navbar a class like bg-dark navbar-dark), we instead want to add our own.

## 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, just like with a WordPress child theme, you are able to add your own CSS files which you can use to overwrite existing styling.

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

<link rel="stylesheet" type="text/css" href="main.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;

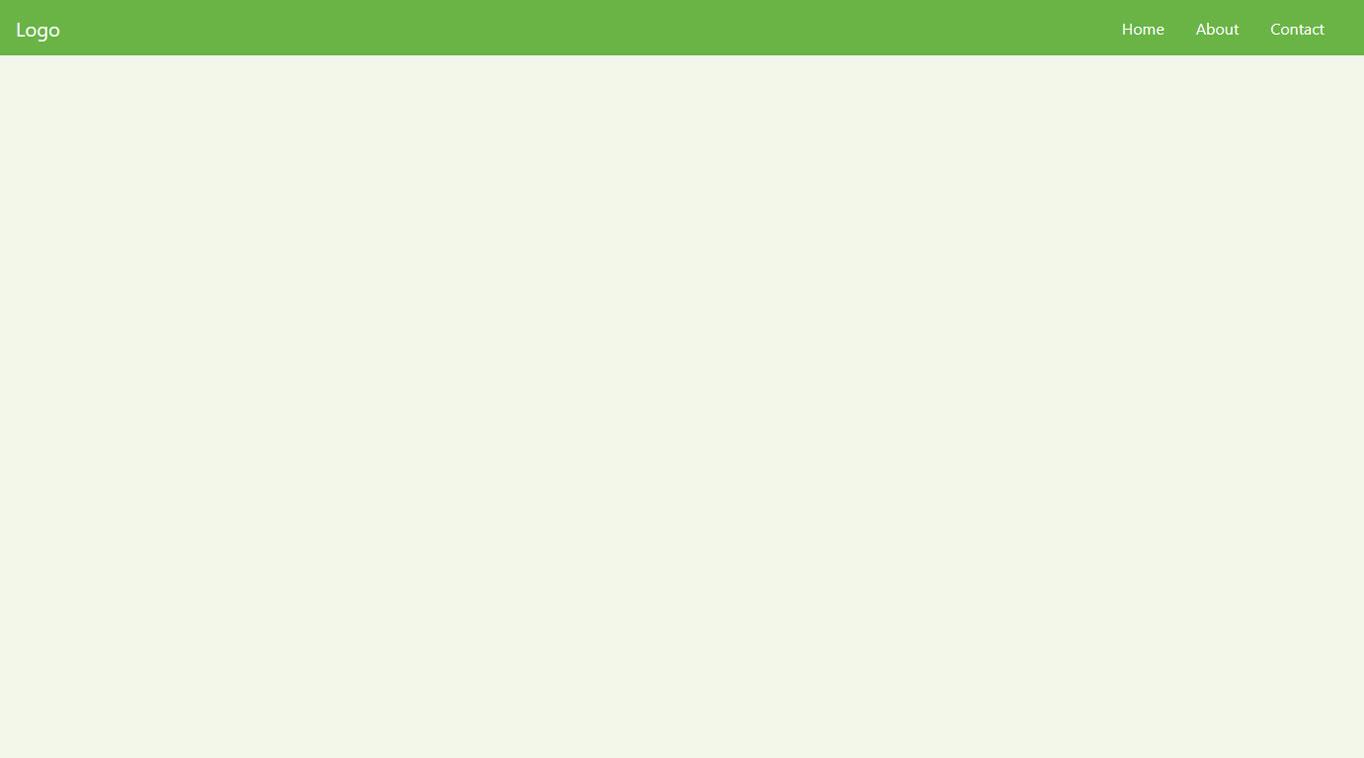
}

.navbar-collapse {

justify-content: flex-end;

}

And here is the result:



Looks better than before, doesn’t it?

## 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.

## Add a Background Image and Custom JavaScript

As the next step in this Bootstrap tutorial, we want to include a full-screen background image for our landing page header. To that end, we will have to use some jQuery or JavaScript to make the image stretch all the way across the screen.

You do that the same way you include custom CSS. First, create a text file of the name ***main.js*** and place it inside your site folder. Then, call it before the closing </body> tag inside ***index.html***.

<script src="main.js"></script>

After that, you can copy and paste this piece of jQuery code to make the <header> element stretch across the entire screen:

$(document).ready(function(){

$('.header').height($(window).height());

})

However, as mentioned, Bootstrap 5 will no longer come with the jQuery library. Therefore, as an alternative and to future proof your site, you can achieve the same with plain JavaScript like so:

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;

}

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



## 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:



## 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 {

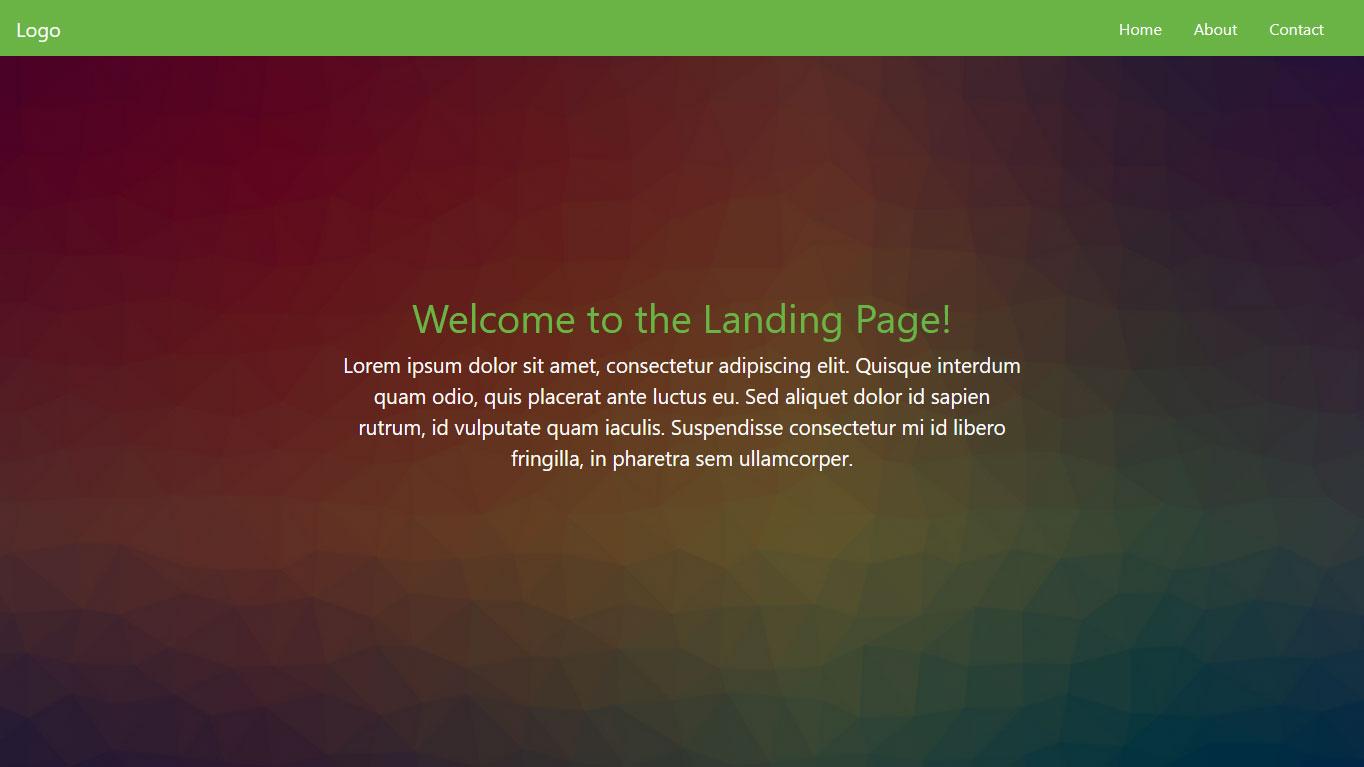
color: #fff;

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?

## 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 tutorial.

Luckily, the framework offers plenty of tools to create buttons quickly and easily. You can find a lot of examples [here](https://getbootstrap.com/docs/4.5/components/buttons/). 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 {

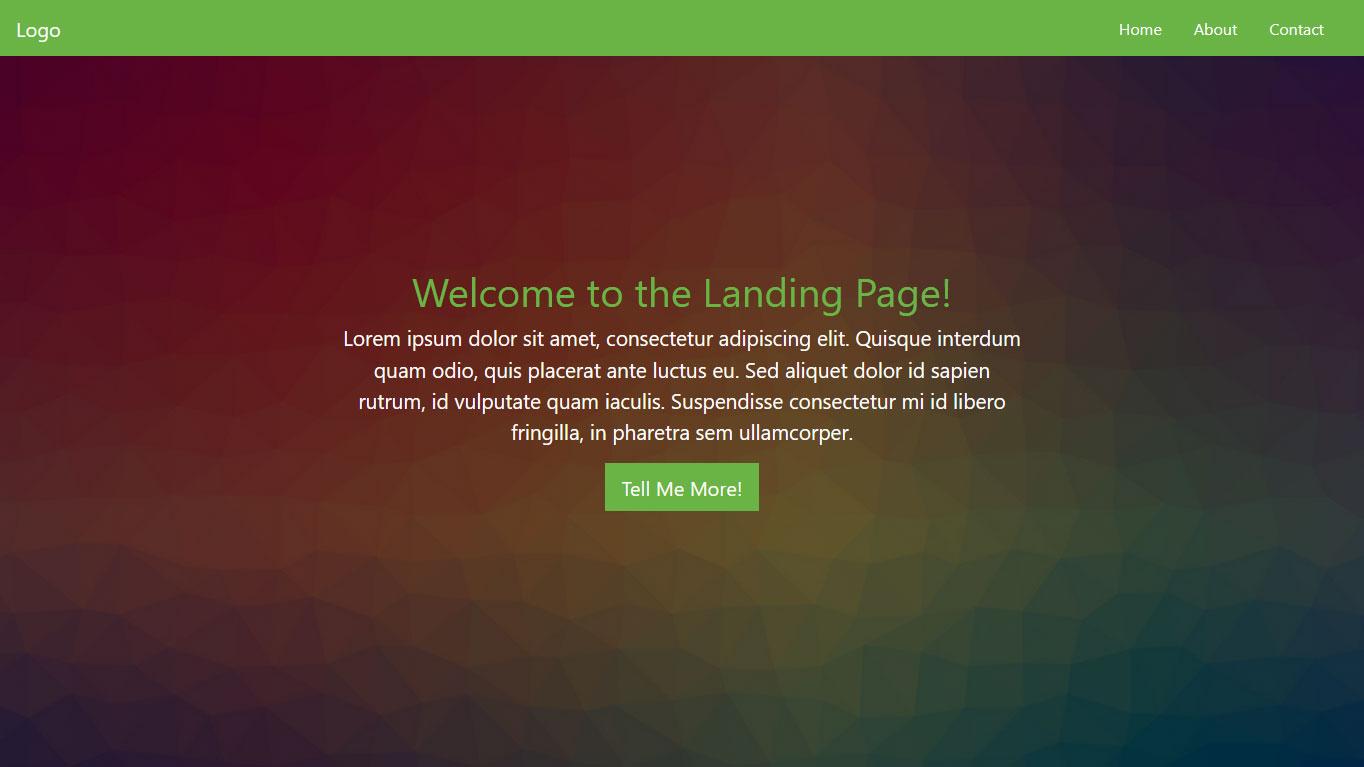
border:1px solid #fff;

background:#fff;

color:#000;

}

After saving and reloading, it looks like this:



## 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](https://getbootstrap.com/docs/4.5/layout/grid/), and no tutorial 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>

<img src="images/column-1.jpg" class="img-fluid">

<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>

<img src="images/column-2.jpg" class="img-fluid">

<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).

It makes sense, doesn’t it?

You will also notice that we included images and added the .image-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 usual place:

.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);

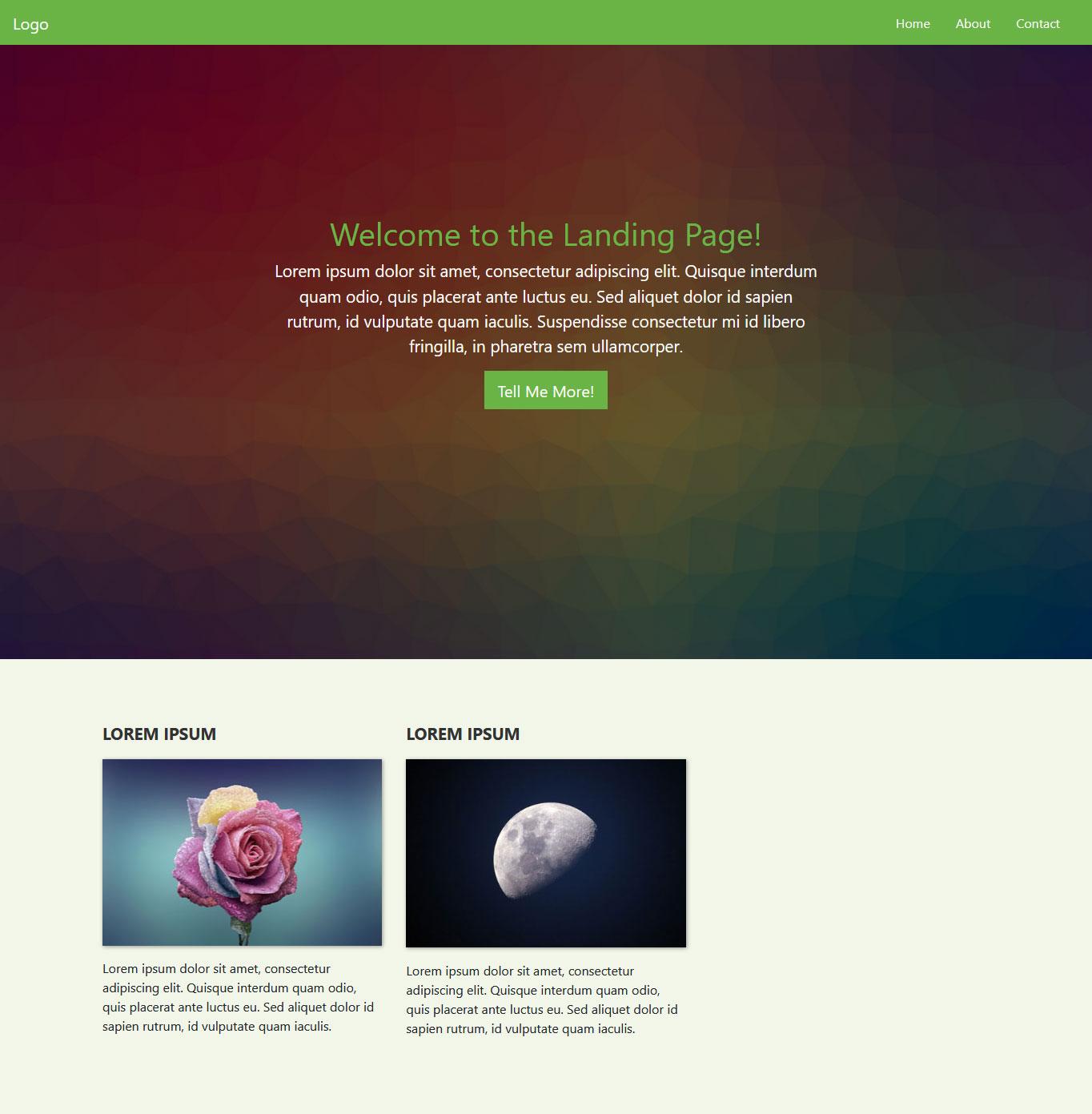
-moz-box-shadow: 1px 1px 4px rgba(0, 0, 0, 0.4);

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](https://getbootstrap.com/docs/4.5/utilities/shadows/).

## Add a Contact Form

You will notice that one of the new 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](https://getbootstrap.com/docs/4.5/components/forms/). 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;

}

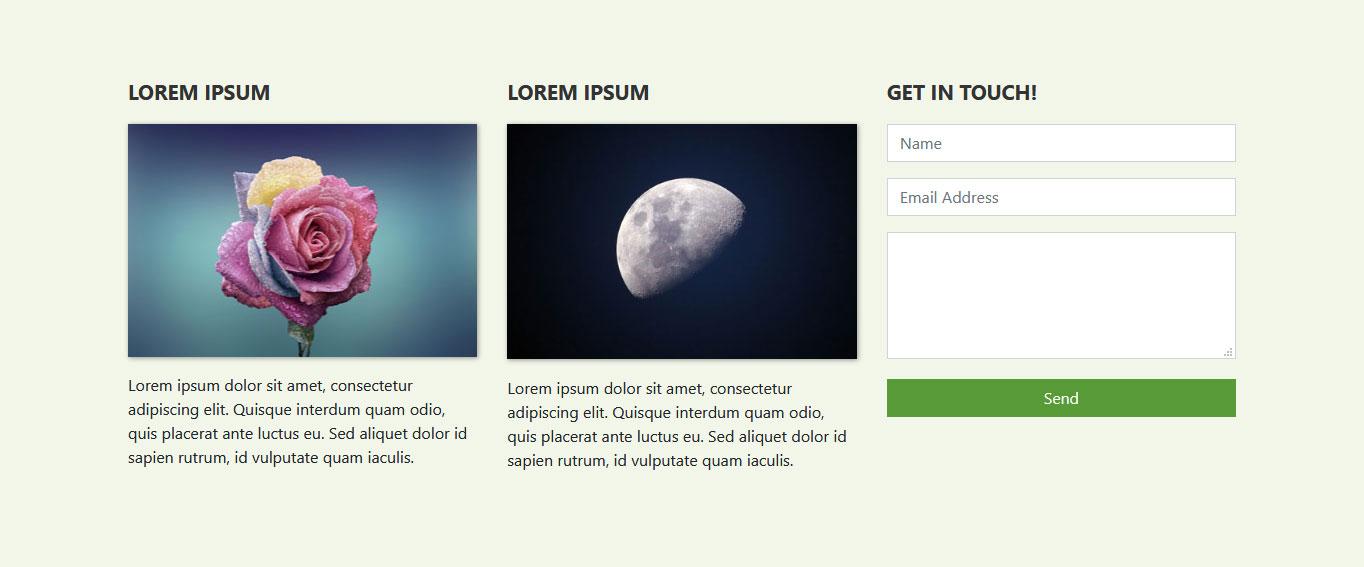
.features .btn:hover {

background-color: #333;

border: 1px solid #333;

}

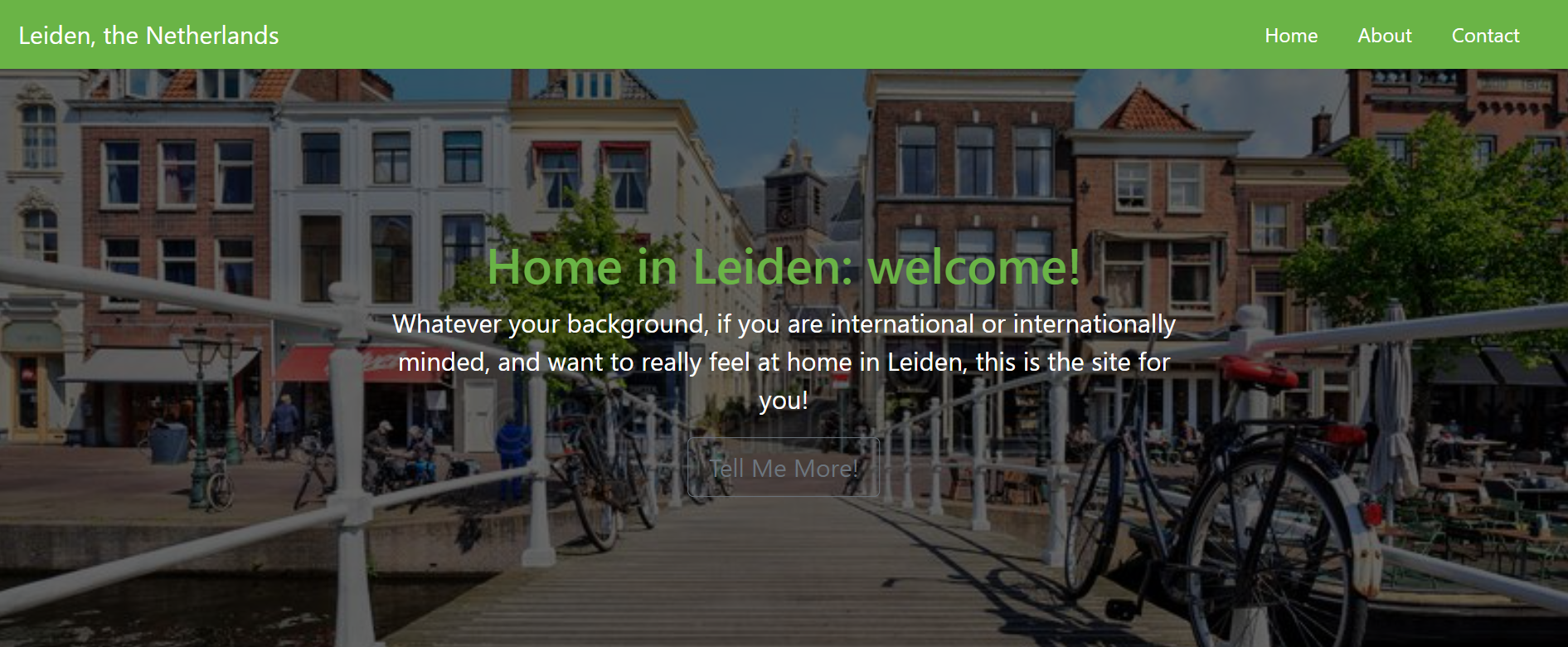
When you do, you get a form like this:



## A website of your favourite place

Turn or modify this bootstrap template to be a website of your favourite place.

Example:





* When the **tell me more button** is clicked, the website will link to a youtube video of your favourite place.
* Change the **Logo** (navigation bar), to the **name of the place** and **link** to your favourite place website.
* When **contact** (navigation bar) is clicked, the page will display or focus on the “**Get in Touch!**”.
* Link the **about** (navigation bar) to **the three-column** section.

## Upload Your Website to a Web Host

If you have been following along, you should now be set with a finished website (plus optional modal) that is also completely responsive.

However, so far, nobody but you can see it. To change that, you need a [web host and domain](https://websitesetup.org/web-hosting-and-domain-explained/). That way, people can input your website address into their browser and then access your newly minted Bootstrap website online.

**Hand In (Putrablast):**

1. Link of your website
2. All files