Building an E-Commerce Website with Bootstrap

You took a look at the various attributes and features of Bootstrap in the earlier chapters. However, just learning the concepts is not enough, as you need to possess the skills to implement them practically. In this chapter, we will create an e-commerce website that will help you get to grips with web designing using Bootstrap.

Initially, we will build the parts step by step and increase the level of difficulty gradually so that you do not get overwhelmed with an information overload. We will first build the ecommerce.html page and then create the other pages such as category.html, account.html, and product.html, resulting in a responsive website.

With Notepad, you cannot determine whether each <div> element has been closed, resulting in an incorrect output. In this example, we will be using Notepad++ as it provides syntax highlighting, word completion, and syntax folding, which will help you code better. Notepad++ is open source and can be downloaded for free from its official website at http://notepad-plus-plus.org/. It is a good practice to use Notepad++ or any advanced editor for your projects, as it streamlines your web designing experience.

Designing the ecommerce.html page

Create four web pages, namely ecommerce.html, category.html, account.html, and product.html for the e-commerce, product categories, account, and the product pages respectively.

Initially, we will look at ecommerce.html.
For now, let's paste the following basic Bootstrap code in the `ecommerce.html` file:

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="">
  <meta name="author" content="">
  
  <title>Bootstrap Store</title>
  <!-- Bootstrap Core CSS -->
  <link href="css/bootstrap.css" rel="stylesheet">
  <!-- Custom CSS -->
  <link href="style.css" rel="stylesheet">
  
  <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
  <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
  <!--[if lt IE 9]>
  <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
  <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
  <![endif]-->
</head>

<body>
  <!-- jQuery Version 1.11.0 -->
  <script src="js/jquery-1.11.1.js"></script>
  
  <!-- Bootstrap Core JavaScript -->
  <script src="js/bootstrap.js"></script>
</body>
</html>
```

As you can see, we have also added `respond.min.js`, the jQuery file, and HTML shiv along with the Bootstrap files.
Some CSS3 properties and HTML5 elements are not fully supported by the Internet Explorer 8 browser. Therefore, Internet Explorer 8 requires the use of html5shiv to correctly display those elements and Respond js to enable media query support.

We will now create a navbar for the web page.

Similar to the process of creating a navbar in the preceding chapters, you define the navbar between the <body> tags. We define navbar-brand as Bootstrap Store and use the .navbar-inverse class, resulting in a black background and white text for the navbar:

```html
<body>
  <!-- Navigation -->
  <nav class="navbar-inverse" role="navigation">
    <div class="container-fluid">
      <!-- Brand and toggle get grouped for better mobile display -->
      <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">
          <span class="sr-only">Toggle navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="ecommerce.html">Bootstrap Store</a>
      </div>
      <!-- Collect the nav links, forms, and other content for toggling -->
      <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
      </div><!-- /.navbar-collapse -->
    </div><!-- /.container-fluid -->
  </nav>

  <!-- jQuery Version 1.11.0 -->
  <script src="js/jquery-1.11.1.js"></script>
  <!-- Bootstrap Core JavaScript -->
  <script src="js/bootstrap.js"></script>
</body>
```
The output of the code on execution will be as follows:

Further on, we will add the categories and navigation links to this navbar. The code has to be inserted after the <!-- Collect the nav links, forms, and other content for toggling --> comment and before the <div> element containing the <!-- /.navbar-collapse --> comment.

In the code, we define the Categories link in addition to the other navigation links. We create a dropdown for the Categories link, wherein we define the various types of products by their genre such as baby products, electronics, and shoes.

Take a look at the following code to understand it better:

```html
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
  <ul class="nav navbar-nav">
    <li class="dropdown">
      <a href="#" class="active dropdown-toggle" data-toggle="dropdown"
        >Categories <span class="caret"></span></a>
      <ul class="dropdown-menu" role="menu">
        <li><a href="category.html">Apparel & Accessories</a></li>
        <li><a href="category.html">Baby Products</a></li>
        <li><a href="category.html">Beauty & Health</a></li>
        <li><a href="category.html">Electronics</a></li>
        <li><a href="category.html">Furniture</a></li>
        <li><a href="category.html">Home & Garden</a></li>
        <li><a href="category.html">Luggage & Bags</a></li>
        <li><a href="category.html">Shoes</a></li>
        <li><a href="category.html">Sports & Entertainment</a></li>
        <li><a href="category.html">Watches</a></li>
        <li class="divider"></li>
        <li><a href="ecommerce.html">All Categories</a></li>
      </ul>
    </li>
    <li><a href="#">Link</a></li>
    <li><a href="#">Link</a></li>
  </ul>
</div>
```
The output of the code will be as follows:

<table>
<thead>
<tr>
<th>Bootstrap Store</th>
<th>Categories</th>
<th>Link</th>
<th>Link</th>
</tr>
</thead>
</table>

Thus, we have defined the **Categories** menu and the drop-down menu, which is visible on clicking the caret.

Now, we add the **Sign in** link, user account link, and the shopping cart with a badge to the right-hand side of the navbar and specify Glyphicons for them. Remember that this snippet needs to be added after the links defined prior to it and before the `<div>` element containing the <!-- /.navbar-collapse --> comment:

```html
<ul class="nav navbar-nav navbar-right">
    <li><a href="#" style="border: 0px; outline: 0px; padding: 0px; margin: 0px;">
        <span class="badge pull-right">4</span><i class="glyphicon glyphicon-shopping-cart"></i>
    </a></li>
    <li><a href="account.html"><i class="glyphicon glyphicon-user"></i></a></li>
    <li><a href="#" data-toggle="modal" data-target="#myModal">Sign in</a></li>
</ul>
```

The output of the added code on execution will result in the following screenshot:

<table>
<thead>
<tr>
<th>Bootstrap Store</th>
<th>Categories</th>
<th>Link</th>
<th>Link</th>
</tr>
</thead>
</table>

For the **Sign in** part, we will create a modal that will be displayed by clicking on the **Sign in** link.
The following code snippet needs to be added after the closed </nav> element:

```html
<!-- Modal -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal"><span aria-hidden="true">×</span><span class="sr-only">Close</span></button>
        <h2 class="modal-title" id="myModalLabel">Sign in</h2>
      </div>
      <div class="modal-body">
        <form class="form-signin" role="form">
          <h3 class="form-signin-heading">Sign in with your email address</h3>
          <div class="form-group">
            <input type="email" class="form-control" placeholder="Email address" required autofocus>
          </div>
          <div class="form-group">
            <input type="password" class="form-control" placeholder="Password" required>
          </div>
          <div class="checkbox">
            <label>
              <input type="checkbox" value="remember-me"> Remember me
            </label>
          </div>
          <button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>
        </form>
      </div>
    </div>
  </div>
</div>
```

```html
<div class="row">
  <div class="col-xs-3">
    <a href="#" class="btn btn-facebook btn-large btn-caps btn-block">Facebook <span class="icon-facebook"></span></a>
  </div>
  <div class="col-xs-3">
    <a href="#" class="btn btn-twitter btn-large btn-caps btn-block">Twitter <span class="icon-twitter"></span></a>
  </div>
</div>
```

[6]
If you click on the **Sign in** link, the following dialog box will be displayed:

![Sign in dialog box](image)

From the preceding code and the output, you can see that we have defined a form to get the user authentication input and have inserted it inside the code for the modal.

We will now create a carousel for the web page. After the modal code has been defined, we define the page content. We start with the `<!-- Page Content -->` comment for easy readability, which indicates that the entire content will be defined within the `<div>` element using the `.container` class and `content` as the ID for it.

Further on, we define the carousel within that container using the following code snippet:

```html
<div id="content" class="container">
  <div class="row carousel-holder">
    <div class="col-md-12">
      <div id="carousel-example-generic" class="carousel slide" data-ride="carousel">
```

---

[7]
Now that we have defined the carousel, the output of the code upon execution will be as follows:
Let's now define the product categories after the carousel code using the following code snippet:

```html
<hr />
<div class="row">
  <div class="col-sm-4 col-md-3">
    <h3>Categories</h3>
    <div class="list-group">
      <a href="category.html" class="list-group-item">Apparel & Accessories</a>
      <a href="category.html" class="list-group-item">Baby Products</a>
      <a href="category.html" class="list-group-item">Beauty & Health</a>
      <a href="category.html" class="list-group-item">Electronics</a>
      <a href="category.html" class="list-group-item">Furniture</a>
      <a href="category.html" class="list-group-item">Home & Garden</a>
      <a href="category.html" class="list-group-item">Luggage & Bags</a>
      <a href="category.html" class="list-group-item">Shoes</a>
      <a href="category.html" class="list-group-item">Sports & Entertainment</a>
      <a href="category.html" class="list-group-item">Watches</a>
    </div>
  </div>
</div>
</div><!-- /.container class with content as the id-->
```

After all the code we have written so far, let's create a footer for the web page by defining the <div> element using the .container class only.
Take a look at the following code snippet to understand it better:

```html
<div class="container">
<hr />
<!-- Footer -->
<footer>
  <div class="row">
    <div class="col-lg-12">
      <p>Copyright &copy; <a href="ecommerce.html">Packt Publishing</a> 2014</p>
    </div>
  </div>
</footer>
</div>
```

Check out each `<div>` element and see that it is appropriately closed as there are multitudes of the `<div>` element in the code. Using Notepad++, you will be able to understand whether the syntax is appropriate and also whether every corresponding `<div>` is closed correctly.

The output of the code on adding the product categories and the footer will be as follows:
Now we will proceed further and add the products along with a brief description so that they are reflected on your web page. The `<div>` element with the categories was defined with the `.col-sm-4 col-md-3` class, and it occupies three columns of the web page on the left-hand side on a medium-sized display screen and four columns on a small-screen device. Since it is a 12-column grid, the remaining space will be used to display individual products.

Therefore, after the categories have been defined, let's add the following code to it:

```html
<div class="col-sm-8 col-md-9">
  <div class="row">

  </div>
</div>
```

Insert the following code in the preceding nested `<div>` element with the `.row` class to add a product to the right-hand side of the **Categories** menu:

```html
<div class="col-sm-6 col-md-4">
  <div class="thumbnail">
    <img src="http://placehold.it/750x600" alt="">
    <div class="add-to-cart">
      <a href="#" class="glyphicon glyphicon-plus-sign" data-toggle="tooltip" data-placement="top" title="Add to cart"></a>
    </div>
    <div class="caption">
      <h4 class="pull-right">$24.99</h4>
      <h4><a href="product.html">1<sup>st</sup> Product</a></h4>
      <p>This is a short description. Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
      <div class="ratings">
        <p class="pull-right"><a href="product.html#comments">15 reviews</a></p>
        <p><span class="glyphicon glyphicon-star"></span><span class="glyphicon glyphicon-star"></span><span class="glyphicon glyphicon-star"></span><span class="glyphicon glyphicon-star"></span><span class="glyphicon glyphicon-star"></span></p>
      </div>
    </div>
  </div>
</div>
```
The output after defining the product is as follows:

Copy the code several times and paste it to get a grid of replicated products. On execution of the code, you can see the following screen below the carousel:
Therefore, we have succeeded in building a web page with the navbar with links, categories to the left-hand side, and products ranging from the center to the right-hand side of the page. We defined a carousel and also defined a modal, wherein a dialog box comes into the picture on clicking on the **Sign in** option.

Now, let's look at the next steps wherein we create the remaining three pages, that is, `account.html`, `category.html`, and `product.html`, so that you have a real-time scenario.
Copy the navigation and the footer code and add it to the account.html, category.html, and product.html pages. Then, all of these three remaining pages will reflect the same output. For clarity, we have created a custom common.html file that contains the code for navigation links, the modal, and the footer.

Note that a custom common.html file has been created so that we realize that these features will be the same in all four web pages, including ecommerce.html. The common.html file is not used, but creating such a file is quite useful if you want to copy and paste the default code for various web pages in your projects. Also, in real-time scenarios, we use tailored templates for such purposes or to minimize the repetition of code.

### Designing the account.html web page

The account.html web page looks like this:

```html
<!-- Page Content -->
<div id="content" class="container">
</div><!-- div element with the content as id and a .container class -->
```

Similar to the procedure followed in the ecommerce.html page, we define the page content. Note that you need to add the page content by inserting the code snippet after the modal code and before the `<div>` element with the `.container` class that encloses the footer of the web page. For code clarity and readability, we add the `<!-- Page Content -->` comment and then define a `<div>` element with the `.container` class and `#content` as the ID, and add a `</div>` element for it. Now our entire code needs to be inserted between these specific `<div>` elements.

Take a look at the following code snippet to understand it better:

```html
<!-- Page Content -->
<div id="content" class="container">

</div><!-- div element with the content as id and a .container class -->
```
We then create a `<div>` element with the `.row` class, wherein we will include the account menu. Let's create a Manage Order section and enclose it within three columns for a medium-sized device. Then, we define a list group of items and define the different options, namely, All Order, Manage Feedback, My Coupons, and My Shipping Address:

```html
<div class="row">
  <div class="col-md-3">
    <h3 class="">
      Manage Orders
    </h3>
    <div class="list-group">
      <a href="account.html" class="list-group-item">All Order</a>
      <a href="account.html" class="list-group-item">Manage Feedback</a>
      <a href="account.html" class="list-group-item">My Coupons</a>
      <a href="account.html" class="list-group-item">My Shipping Address</a>
    </div>
  </div>
</div>
```

The output of the code upon execution will be:

![Manage Orders](image)

Let's define a `<div>` element with the `.thumbnail` class and create another `<div>` element within it with a `.row` class. Thereafter, we add the product names, the number of orders, prices, the track buttons, and the messages for the latest status of the products.
We will use the following code snippet to understand this better:

```html
<div class="thumbnail">
    <div class="row">
        <div class="col-sm-1">
            <img class="img-responsive" src="http://placehold.it/750x600" alt="" />
        </div>
        <div class="col-sm-4">
            <a href="product.html">Product Name</a>
        </div>
        <div class="col-sm-1">
            1
        </div>
        <div class="col-sm-2">
            $49.99
        </div>
        <div class="col-sm-2">
            <button class="btn btn-sm btn-default">Track</button>
        </div>
        <div class="col-sm-2">
            <a href="#">1 Message</a>
        </div>
    </div>
</div>
```

On executing the code, you can see the following output:
Now that we have a single product defined and the order status for that product, let's replicate the products five times. We customize the status of the products by specifying different amounts of messages, prices, and the latest updates of some of the products so as to determine whether the status is **Cancelled** or **Completed** (or a **Track** button to determine whether it is on its way).

Thus, to create a wide array of order status' for different products, we use the following code snippet:

```html
<div class="thumbnail">
  <div class="row">
    <div class="col-sm-1">
      <img class="img-responsive" src="http://placehold.it/750x600" alt="" />
    </div>
    <div class="col-sm-4">
      <a href="product.html">Product Name</a>
    </div>
    <div class="col-sm-1">
      1
    </div>
    <div class="col-sm-2">
      $19.99
    </div>
    <div class="col-sm-2">
      <button class="btn btn-sm btn-success">Completed</button>
    </div>
    <div class="col-sm-2">
      <a href="#">1 Message</a>
    </div>
  </div>
</div>

<div class="thumbnail">
  <div class="row">
    <div class="col-sm-1">
      <img class="img-responsive" src="http://placehold.it/750x600" alt="" />
    </div>
    <div class="col-sm-4">
      <a href="product.html">Product Name</a>
    </div>
    <div class="col-sm-1">
      1
    </div>
    <div class="col-sm-2">
      $19.99
    </div>
    <div class="col-sm-2">
      <button class="btn btn-sm btn-success">Completed</button>
    </div>
    <div class="col-sm-2">
      <a href="#">1 Message</a>
    </div>
  </div>
</div>
```
Building an E-Commerce Website with Bootstrap

```html
<div class="col-sm-1">
  1
</div>
<div class="col-sm-2">
  $39.99
</div>
<div class="col-sm-2">
  <button class="btn btn-sm btn-danger">Cancelled</button>
</div>
<div class="col-sm-2">
  <a href="#">0 Messages</a>
</div>
</div>

<div class="thumbnail">
  <img class="img-responsive" src="http://placehold.it/750x600* alt=" />
  <div class="row">
    <div class="col-sm-1">
      <img class="img-responsive" src="http://placehold.it/750x600* alt=" />
    </div>
    <div class="col-sm-4">
      <a href="product.html">Product Name</a>
    </div>
    <div class="col-sm-1">
      1
    </div>
    <div class="col-sm-2">
      $49.99
    </div>
    <div class="col-sm-2">
      <button class="btn btn-sm btn-default">Track</button>
    </div>
    <div class="col-sm-2">
      <a href="#">1 Message</a>
    </div>
  </div>
</div>

```
</div>
<div class="col-sm-4">
<a href="product.html">Product Name</a>
</div>
</div>
</div>
</div>
</div>
<div class="row">
<div class="col-sm-1">
<img class="img-responsive" src="http://placehold.it/750x600" alt="" />
</div>
<div class="col-sm-4">
<a href="product.html">Product Name</a>
</div>
<div class="col-sm-1">
1
</div>
<div class="col-sm-2">
$19.99
</div>
<div class="col-sm-2">
<button class="btn btn-sm btn-success">Completed</button>
</div>
<div class="col-sm-2">
<a href="#">1 Message</a>
</div>
</div>

</div>

</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
The output of the code upon execution will be as follows:

Let's further enhance the web page by adding a .pagination class so that your website users can scroll for different products on different web pages.

Take a look at the following code snippet to understand it better:

```html
<div class="col-sm-12 center">
  <ul class="pagination">
    <li class="disabled"><a href="#">&laquo;</a></li>
    <li class="active"><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li><a href="#">5</a></li>
    <li><a href="#">&raquo;</a></li>
  </ul>
</div>
```
The output of the executed code will have pagination at the bottom of the page displayed as follows:

![Pagination Example](image)

**Designing the category.html web page**

Just like the account.html web page, the category.html web page looks like the following screenshot due to the common code containing the navigation links, the modal, and the various Glyphicons:

![Category Page Example](image)
Similar to the e-commerce web page, we then create a page content section after the modal. We add a `<div>` element with the `.container` class and `#content` as the ID and close it with its corresponding `</div>` element. Within these specific `<div>` elements, we create a category menu using the `list-group` items attribute. It is the same code snippet used to create a category menu on the left-hand side of the web page. We do not use a carousel here as is the case with the `ecommerce.html` web page. Then, we add the products in the same way we defined the products in the `ecommerce.html` web page.

The `category.html` web page will look like this:

![Category Page](image)

Add the following `.pagination` class to the code for it to reflect on different pages. It is similar to the code we used for pagination for the `account.html` web page.

Take a look at the following code to understand it better:

```html
<div class="col-sm-12 center">
  <ul class="pagination">
    <li class="disabled"><a href="#">&laquo;</a></li>
    <li class="active"><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li><a href="#">5</a></li>
    <li><a href="#">&raquo;</a></li>
  </ul>
</div>
```
Now, the lower half of your `category.html` page will look like this:

![Image of category HTML page](image)

We have defined the `account.html` and `category.html` pages so far. Next, we proceed to the final part, the `product.html` web page.

### Designing the `product.html` web page

Similar to `category.html`, we create a **Categories** menu on the left-hand side of the screen below the navbar.
Suppose we want to add a particular product to the web page. Initially, we define the column width for the product. In this case, we have defined the `.col-md-9` class as the space to be particularly assigned to the product, meaning that the product details can be viewed on the right-hand side of the **Categories** menu. Then, we define the graphic image for the product by defining the `.thumbnail` class. Next, we nest four small different images in the same `.thumbnail` class.

Take a look at the following code to understand it better:

```html
<div class="col-md-9">
  <div class="thumbnail">
    <div class="row">
      <div class="col-sm-6">
        <img class="img-responsive" src="http://placehold.it/750x600" alt="" />
        <div class="thumbnails row">
          <div class="col-xs-3">
            <a href="#"><img src="http://placehold.it/750x600" alt="" class="img-thumbnail img-responsive" /></a>
          </div>
          <div class="col-xs-3"></div>
          <div class="col-xs-3">
            <a href="#"><img src="http://placehold.it/750x600" alt="" class="img-thumbnail img-responsive" /></a>
          </div>
          <div class="col-xs-3">
            <a href="#"><img src="http://placehold.it/750x600" alt="" class="img-thumbnail img-responsive" /></a>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```
The output of the code upon execution will be as follows:

```

```

Next, we define the product name and a short description that will be displayed to the right-hand side of the main image. In the code, we have defined the .col-sm-6 class to determine the space assigned for the product name and the short description.

Therefore, we add the following code for this purpose:

```

```

The output of the code will be as follows:

```html
<form role="form">
  <div class="form-group">
    <label class="control-label" for="number">Number</label>
    <input type="number" class="form-control input-sm" id="number" />
  </div>

  <div class="form-group">
    <label>Color</label>
    <select id="color">
      <option name="color">Blue</option>
      <option name="color">Green</option>
      <option name="color">Red</option>
      <option name="color">Yellow</option>
    </select>
  </div>

  <div class="form-group">
    <label>Size</label>
  </div>
</form>
```

Now, we write the code for a form, wherein we specify the color, number of products to be ordered, and size, while also adding the Contact Seller and Add to Cart buttons.

Take a look at the following code to understand it better:

```html
<form role="form">
  <div class="form-group">
    <label class="control-label" for="number">Number</label>
    <input type="number" class="form-control input-sm" id="number" />
  </div>

  <div class="form-group">
    <label>Color</label>
    <select id="color">
      <option name="color">Blue</option>
      <option name="color">Green</option>
      <option name="color">Red</option>
      <option name="color">Yellow</option>
    </select>
  </div>

  <div class="form-group">
    <label>Size</label>
  </div>
</form>
```
The output of the code on execution will be as follows:

Let's also add the product's detailed description while also defining the reviews section. We specify the number of ratings and use Glyphicons to create the stars to rate the review. Using a .wells class, we create a shaded space to house the **Leave a Review** text.
Take a look at the following code snippet to understand this:

```html
<div class="description">
  <p>
  </p>
</div>

<div class="ratings">
  <p class="pull-right">3 reviews</p>
  <p>
    <span class="glyphicon glyphicon-star"></span>
    <span class="glyphicon glyphicon-star"></span>
    <span class="glyphicon glyphicon-star"></span>
    <span class="glyphicon glyphicon-star"></span>
    <span class="glyphicon glyphicon-star-empty"></span>
  4.0 stars
  </p>
</div>

<div class="well">
  <div class="text-right">
    <a class="btn btn-success">Leave a Review</a>
  </div>
</div>
```
The lower half of the `product.html` web page will be as follows:

```
now, let's just add some reviews to enhance the layout and make it look like an authentic web page. Do this by adding the following code snippet:

```html
<div class="row">
    <div class="col-md-12">
        ★★★★☆ 4.0 stars
        <div class="row">
            <div class="col-md-12">
                <span class="glyphicon glyphicon-star"></span>
                <span class="glyphicon glyphicon-star"></span>
                <span class="glyphicon glyphicon-star"></span>
                <span class="glyphicon glyphicon-star"></span>
                <span class="glyphicon glyphicon-star-empty"></span>
                David More
                <span class="pull-right">1 day ago</span>
                <p>From <em>United States</em></p>
                <p>This product was great in terms of quality. I would definitely buy another!</p>
        </div>
    </div>
</div>
```
The addition of reviews results in the following output that is displayed on the lower-half of the page as we scroll down the screen:
In real-time scenarios, on an e-commerce website, we can sometimes view the **Similar Products** section or **Other things to Buy** suggestions, which are kind of related to the showcased product. So in our following code snippet, we will take a look at the procedure to showcase similar products so that you can implement it in your web designing projects.

Take a look at the following code to understand it better:

```html
<div class="row">
    <div class="col-sm-12">
        <h3>Other products you may want to buy</h3>
    </div>
    <div class="col-sm-6 col-md-4">
        <div class="thumbnail">
            <img src="http://placehold.it/750x600" alt="">
            <div class="add-to-cart">
                <a href="#" class="glyphicon glyphicon-plus-sign" data-toggle="tooltip" data-placement="top" title="Add to cart"></a>
            </div>
            <div class="caption">
                <h4 class="pull-right">$84.99</h4>
                <h4><a href="product.html">1<sup>st</sup> Product</a></h4>
            </div>
        </div>
    </div>
</div>
```
Building an E-Commerce Website with Bootstrap

<p>This is a short description. Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>

<div class="ratings">
  <p class="pull-right"><a href="product.html#comments">6 reviews</a></p>
  <p>
    <span class="glyphicon glyphicon-star"></span>
    <span class="glyphicon glyphicon-star"></span>
    <span class="glyphicon glyphicon-star"></span>
    <span class="glyphicon glyphicon-star-empty"></span>
    <span class="glyphicon glyphicon-star-empty"></span>
  </p>
</div>

<div class="col-sm-6 col-md-4">
  <div class="thumbnail">
    <img src="http://placehold.it/750x600" alt="">
    <div class="add-to-cart">
      <a href="#" class="glyphicon glyphicon-plus-sign" data-toggle="tooltip" data-placement="top" title="Add to cart"></a>
    </div>
    <div class="caption">
      <h4 class="pull-right">$94.99</h4>
      <h4><a href="product.html">2nd Product</a></h4>
      <p>This is a short description. Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
      <div class="ratings">
        <p class="pull-right"><a href="product.html#comments">18 reviews</a></p>
        <p>
          <span class="glyphicon glyphicon-star"></span>
          <span class="glyphicon glyphicon-star"></span>
          <span class="glyphicon glyphicon-star"></span>
        </p>
      </div>
    </div>
  </div>
</div>
Chapter 9
On execution of the code, the following page is displayed when you scroll down the web page:

From the preceding code and the output, you can see that we have defined the suggested products and assigned the same space for all of them. We defined the `.col-sm-6 col-md-4` class for 1st Product, 2nd Product, and 3rd Product respectively, due to which you see the products displayed when you scroll down the product page.

In the preceding code example, we have linked all the products to the `product.html` page. Thus, by clicking on any product in the web page, you will be directed to the same `product.html` page. In real-time scenarios, you can link the products to their respective web pages to see more information and order the products if need be.
Summary

It has been quite an interesting journey with Bootstrap and you can now leverage the knowledge gained from this book for real-time implementation in your web designing projects. Practice makes perfect, and this applies to coding too. It also helps if you develop an analytical bent of mind as computers understand only logic. With this book, you will get to grips with Bootstrap but this is just the start of the journey; the sea of knowledge is far beyond. It's all learning. The more you learn, you realize that there is much more to learn. From my experience, learning never ends and the learning curve is always steep as it is an ongoing process.

Happy Bootstrapping!