Creating a Product page using Materialize

So far, we have learned about various aspects of Materialize from the Grid Layout, CSS and JavaScript components, and attributes that make web designing a breeze. Of course, there is much more to Materialize and the learning curve is a process that never ceases. However, in this chapter, we will give a glimpse of how to create a simple product page using Materialize. In real-time scenarios, we use templates for these product pages on an e-commerce site; however, you will learn how to create an aesthetic page that can also be used as a prototype for web pages on similar project.

# Ready Steady Go

We will create a product page for Apple iPhone 6. Initially, we will create a navbar using the <nav> tag.

We create a <div> tag and assign the *container* and *nav-wrapper* classes to it. Then we define the anchor link for Daily Giveaways and assign the *brand-logo* class to it. We create an unordered list (<ul>) containing just one list item. We also define the nav-mobile id for the same <ul> tag in addition to specifying the *right* class to it so that the text is positioned at the right of the page. For the list item (<li>) tag, we enter the content text, i.e. About us. Also to the same <li> tag, we have defined an anchor tag and assigned the aboutus.html as the value for the href attribute meaning clicking on the About us text will take you to the About Us page (we will come to that later in the chapter).

Listing 6.1 shows the code snippet for the preceding explained for the navbar:

**Listing 6.1**

<nav>

<div class="nav-wrapper container">

<a href="#" class="brand-logo">Daily Giveaways</a>

<!-- Links hidden on medium and small screen device -->

<ul id="nav-mobile" class="right">

<li><a href="aboutus.html">About us</a></li>

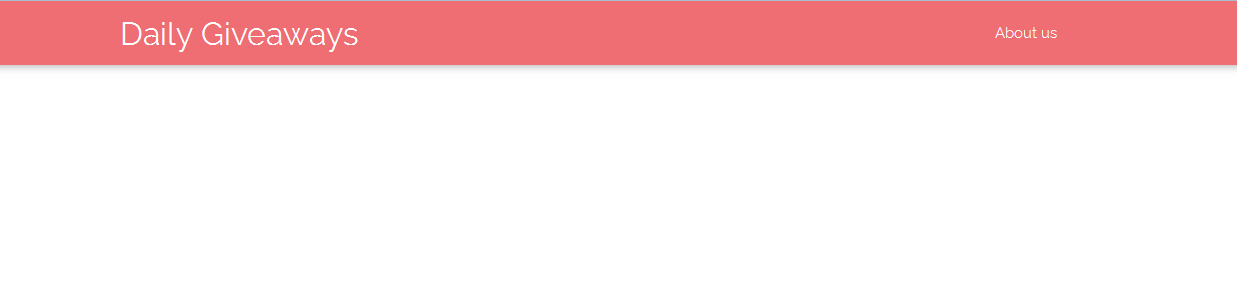
</ul>

</div>

</nav>

Figure 6-1 depicts the output of the code where you can see the navbar:

**Figure 6-1**



Now we design the section where we define the product details. We use the Apple iPhone image as the product image. We create a <section> tag and assign a headerSection class to it. We assign the light grey color to it. Then we define a <div> class to which we assign the container and row classes. We create another <div> inside that <div> to which we assign the productImage class and define a width of 4 columns meaning it will span 4 columns on medium -sized screens. We use the <img> tag to define the source of that image in addition to assigning the responsive-img class to it. Then we close that <div> and open a new <div> where we assign a width of 6 columns on medium-sized screens. We then define the text and content such as the release date, operating system, and memory to it apart from the price. If you see the code in the following Listing 6.2, you can see the usage of the inline material icon class using which we leverage the benefits of the immersive built-in icons for various elements. You can find more about icons at <http://materializecss.com/icons.html>. You need to use the inline <i> tag for the icons just like the way and specify the text for a particular icon within the <i> tags. Then close the respective <div> and <section> tags.

Listing 6.2 shows the code snippet for the same.

**Listing 6.2**

<section class="headerSection grey lighten-1">

<div class="container row">

<div class="productImage col m4">

<img src="images/product.jpg" class="responsive-img">

</div>

<div class="productDetails col m6">

<h3>Apple iPhone 6</h3>

<p><i class="material-icons">date\_range</i> Released 2014, September</p>

<p><i class="material-icons">developer\_mode</i>iOS 8, up to iOS 10.0.2</p>

<p><i class="material-icons">memory</i>16/64/128GB storage, no card slot</p>

<h4>$450.00</h4>

</div>

</div>

</section>

**Figure 6-2**



If you see the preceding Figure 6-2, you can see the product details with the iPhone image. However, you can also see the background image behind the product. This is done with the help of the CSS3 background-blend-mode property within the <style> tags in the <head> tags before the <body> of the code.

More info on background-blend-mode can be found at <https://developer.mozilla.org/en/docs/Web/CSS/background-blend-mode>.

By the way, the code snippet for the background-blend-mode property is defined as follows in the <head> tags.

.headerSection{

margin: 0;

padding: 25px 0;

background: url(images/bg.jpg);

background-blend-mode: screen;

}

Moving forward, we use another new < section> tag where we define the specifications for the iPhone. We create four specifications, namely- the pixels, Camera, Memory, and Charge facets. For this, we create 4 <div>s within the <div> tag within the new <section> tag. We allocate a width of 12 columns on a small screen and 4 columns on a medium-sized screen. We assign a different color for each of the facets, and use tooltips, and assigned the data-position attribute and data-tooltip in addition to defining the built-in material icons for each facet.

Listing 6.3 shows the code snippet for the same:

**Listing 6.3**

<section class="featureSection">

<div class="container row">

<div class="col s12 m3 center-align purple-text accent-1 tooltipped" data-position="bottom" data-delay="50" data-tooltip="Display size and resolution">

<h1><i class="material-icons largeIcon">phone\_iphone</i></h1>

<h3>4.7"</h3>

<p>750x1334 pixels</p>

</div>

<div class="col s12 m3 center-align blue-text accent-1 tooltipped" data-position="bottom" data-delay="50" data-tooltip="Camera: Photo and Video">

<h1><i class="material-icons largeIcon">photo\_camera</i></h1>

<h3>8MP</h3>

<p>1080p</p>

</div>

<div class="col s12 m3 center-align amber-text accent-2 tooltipped" data-position="bottom" data-delay="50" data-tooltip="RAM and chipset">

<h1><i class="material-icons largeIcon">memory</i></h1>

<h3>1GB RAM</h3>

<p>Apple A8</p>

</div>

<div class="col s12 m3 center-align light-green-text accent-2 tooltipped" data-position="bottom" data-delay="50" data-tooltip="Battery Capacity and Technology">

<h1><i class="material-icons largeIcon">battery\_charging\_full</i></h1>

<h3>1810mAH</h3>

<p>Li-Po</p>

</div>

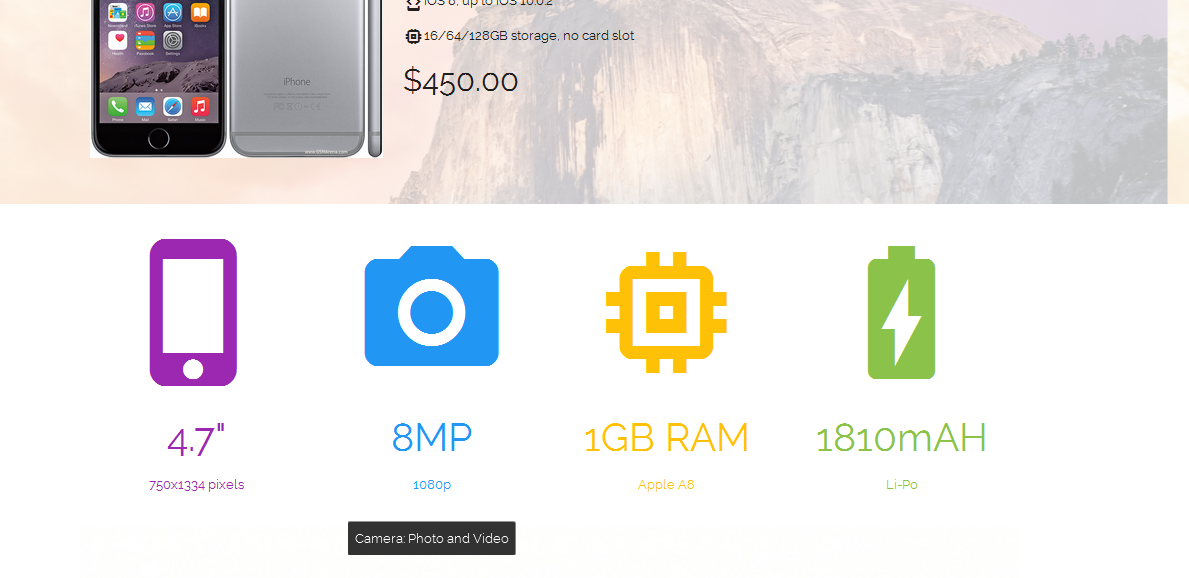
</div>

</section>

The output of the code on execution will display the 4 different specifications as defined and on hovering on each feature, you can see the respective tooltips. In this example, we have hovered over the camera icon.

Figure 6-3 displays the output and the subsequent tooltip on hovering over the camera icon.

**Figure 6-3**



Moving forward, we create another <section> tag where we define the carousel slider. We define the <div> tags within the <section> tags and another <div> within that <div> where we assign the carousel and carousel-slider classes to that <div> . we create 3 anchor tags (<a> ) and assign the carousel-item class to each anchor tag and define the image source for these 3 anchor tags. Close all the <div> and <section> tags.

**Listing 6.4**

<section class="productCarousel">

<div class="container row">

<div class="carousel carousel-slider">

<a class="carousel-item" href="#one!"><img src="images/apple-iphone-6-2.jpg"></a>

<a class="carousel-item" href="#two!"><img src="images/apple-iphone-6-3.jpg"></a>

<a class="carousel-item" href="#three!"><img src="images/apple-iphone-6-and-iphone-6-plus.jpg"></a>

</div>

</div>

</section>

Figure 6-4 depicts the output of the code which results in an image slider carousel below the specifications content:

**Figure 6-4**



Please note that we have defined the styling for the carousel within the <style> tags in the <head> tags. Refer to the code bundle to check out the explicit styling within the <style> tags,; in real time scenarios, you would need to follow the best practice of maintaining separate CSS sheets in an organized way.

Next, we create a small form to notify about the availability of the iPhone as it is out of stock. We create a new section using the <section> tags. We create a <div> element and assign the container and row classes to it just like the other examples. Then, we define the heading and enter the Out of Stock Message as the content. We proceed to create a form wherein we create 3 rows using the <div> tags. Each <div> within the parent <div> for that section is assigned a width of 12 columns on a small screen whereas only 4 columns on a medium screen. We create the Name and Email fields and create a Notify button and add the waves effect to the button. We have also added the validation to the Name and Email fields.

Listing 6.5 shows the code snippet for the form section:

**Listing 6.5**

<section class="inquiry">

<div class="container row">

<h2>Out of Stock</h2>

<p class="red-text">iPhone 6 is presently unavailable. Please provide us with your details so that we can notify you when other deals are available.</p>

<div class="row">

<div class="col s12 m4">

<label for="name">Name</label>

<input type="text" name="name" id="name" placeholder="Enter your Name"/>

</div>

<div class="col s12 m4">

<label for="email">Email</label>

<input type="email" name="email" id="email" class="validate" placeholder="Enter your E-mail address" />

</div>

</div>

<div class="row">

<div class="col s12 m4">

<input type="submit" name="submit" value="Notify me" id="submit" class="btn red waves-effect waves-light" />

</div>

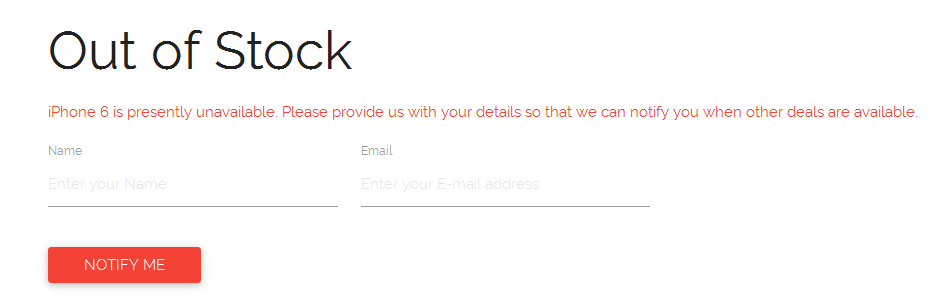
</div>

</div>

</section>

Figure 6-5 depicts the small form we have created below the Carrousel:

**Figure 6-5**



Finally we add the footer to the page. We assign the page-footer class to the opening <footer>. Then we define the copyright by assigning the footer-copyright class to a <div> tag within the <footer> tags. We proceed to create another <div> and assign a container class to it. Close all the <div>s and the <footer> tags.

Listing 6.6 depicts the code snippet for the footer area:

**Listing 6.6**

<footer class="page-footer">

<div class="footer-copyright">

<div class="container">

&copy; Daily Giveaway

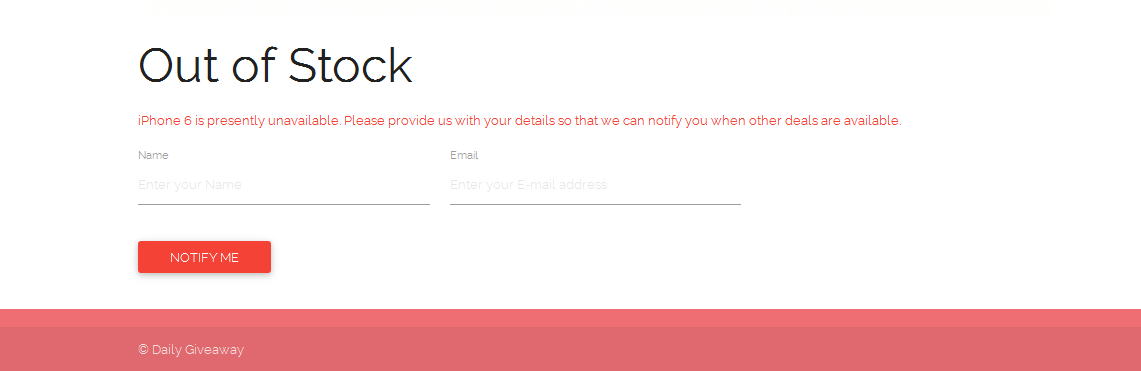
</div>

</div>

</footer>

Now that all the code is done, we execute the code and you can see the footer section on the webpage.

**Figure 6-6**



If you remember in the very first code section where we defined the navbar on top of the page, we had linked the About Us section in the top of the page in the navbar with the About us page using the following line of code (Refer Listing 6.1 for the entire navbar snippet):

**<li><a href="aboutus.html">About us</a></li>**

We have created another page called About Us wherein we have created a navbar followed by an About Us section. We create a form below the About Us content and below the submit form, we define the footer.

**Listing 6.7**

<body>

<!-- Nav Bar -->

<nav>

<div class="nav-wrapper container">

<a href="#" class="brand-logo">Daily Giveaways</a>

<!-- Links hidden on medium and small screen device -->

</div>

</nav>

<!-- Header section which would occupy full screen space on small screens and be two column layout on medium and higher device -->

<section class="aboutContent">

<div class="container">

<h1>About Us</h1>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam dignissim ligula eget accumsan elementum. Suspendisse potenti. Ut aliquet nunc in elit hendrerit finibus. Donec tincidunt consequat tempus. In egestas urna et ligula tristique, feugiat feugiat dolor consectetur. Nullam facilisis feugiat tempus. Donec consequat neque aliquet, accumsan. </p>

</div>

</section>

<!-- Inquiry section will contain form for inquiry -->

<section class="inquiry">

<div class="container row">

<h2>Get in touch</h2>

<div class="row">

<div class="col s12 m4">

<label for="name"> Name</label>

<input type="text" name="name" id="name" required class="validate" placeholder="Enter your Name"/>

</div>

<div class="col s12 m4">

<label for="email" data-error="Invalid email address" data-success="Thank you!">Email</label>

<input type="email" name="email" id="email" class="validate" placeholder="Enter your Email "/>

</div>

<div class="col m12">

<label for="email">Message</label>

<textarea class="message materialize-textarea" placeholder="Fill in your Remarks/Suggestions/Queries"></textarea>

</div>

</div>

<div class="row">

<div class="col s12 m4">

<input type="submit" name="submit" value="Submit" id="submit" class="btn red waves-effect waves-light" />

</div>

</div>

</div>

</section>

<!-- Footer -->

<footer class="page-footer">

<div class="footer-copyright">

<div class="container">

&copy; Daily Giveaway

</div>

</div>

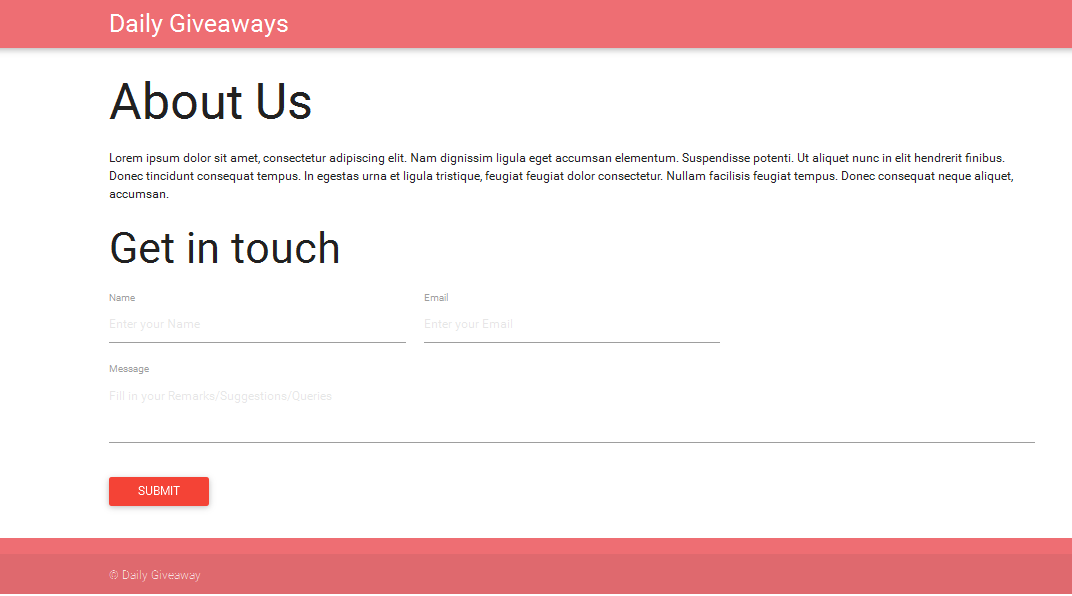
</footer>

</body>

On the index.html file when we click on the About us link, it will take you to the About Us page as defined in the code.

Figure 6-7 shows the output due to clicking the About Us link on the top of the Project page (index.html) which directs you to the About Us page:

**Figure 6-7**



**Summary**

Material Design helps impart continuity over its web and enables a consistent experience in a productive and powerful manner. It creates a uniformity in a clear and concise way without much ambiguity. With responsiveness baked-in along with an immersive experience, it enhances the UX/UI significantly. Natural motions, Animations, and Transitions are streamlined to create an awesome real-world experience. Materialize, as a framework, has HTML, CSS, and JavaScript as its building blocks and helps you deliver an engaging user experience which was what the concept of the World Wide Web was all about, though in a Google –oriented way. Designers can get started in a second as Material Design specifications mix and match good design with innovation. Materialize enables you to deliver all these aspects as the rules are set in an enhanced and simplistic manner.

In this book, we touched base with Materialize and how you can use it for developing supersonic sites though at a basic level. The learning curve is really steep as the web is evolving faster than expected bringing in a new tomorrow. With effectiveness and well-implemented methodologies, Materialize is the go-to framework to implement the intuitiveness of Material Design.