5.3 Class and Style Binding



This section will guide you to:

* Implement attribute, class, and style binding

This guide has four subsections, namely:

5.3.1 Disabling a button using attribute binding

5.3.2 Implementing class binding

5.3.3 Implementing style binding

5.3.4 Pushing the code to Github repositories

**Step 5.3.1:** Disabling a button using attribute binding

* Disable a button using the *disabled* attribute of the element. Set the *disabled* to *false* by binding value to *attr.disabled* attribute property.
* Using table data for image apply if with showImage variable from component.
* Add toggle() function inside a products.component.ts
* Add the following code to *products.component.html*

<div>

    <h1> Product List Page </h1>

    <table class="table">

        <tr>

<th><input type="button" class="btn btn-success" value="Show Image" (click)="toggleImage()"/></th>

            <th>Product</th>

            <th>Code</th>

            <th>Available</th>

            <th>Price</th>

            <th>Rating</th>

        </tr>

    <tr \*ngFor="*let* p of products">

        <td>

            <span \*ngIf="showImage">

            <img

                [src]='p.imageUrl'

                [title]='p.productName'

                [alt]='p.productName'

                [style.width.px]='imageWidth'

                [style.margin.px]='imageMargin'/>

            </span>

        </td>

        <td>{{p.productName}}</td>

        <td>{{p.code}}</td>

        <td>{{p.available}}</td>

        <td>{{p.price}}</td>

        <td>{{p.rating}}</td>

    </tr>

    </table>

    <div>

    </div>

Add the following code to *products.component.ts*

import { Component } from '@angular/core';

import { Product } from '../product';

@Component({

  selector: 'app-products',

  templateUrl: './products.component.html',

  styleUrls: ['./products.component.css']

})

export *class* ProductsComponents {

   pageTitle: *string* = "Product List Page";

   imageWidth:*number* = 80;

   imageMargin:*number* = 10;

   showImage:*boolean* = false;

    products:*Array*<Product>=[

    new Product("https://i.pinimg.com/236x/d9/b9/49/d9b9490af419f70a300d4552d6d737af.jpg","Leaf Rake","GDN-0011","March 19, 2016",19.95,3.5),

    new Product("https://i.pinimg.com/236x/1c/46/9d/1c469d4c102b6aa7f89a8c1b7c7cc797.jpg","Garden Cart","GDN-0023","March 18, 2016",32.99,4.2),

    new Product("https://i.pinimg.com/236x/38/13/43/38134365cf2f9a2e4e21ae3b1c7aa158.jpg","Hammer","TBX-0048","March 21, 2016",8.9,4.8),

    new Product("https://i.pinimg.com/236x/0d/ae/ed/0daeedbf7a3c6e22a35623a695a8c31b.jpg","Saw","TBX-0022","March 15, 2016",11.55,3.7),

    new Product("https://i.pinimg.com/236x/43/e7/82/43e78206fa3c28e1bfc8f92ba349052d.jpg","Video Game Controller","GMG-0042","October 15, 2015",35.95,4.6),

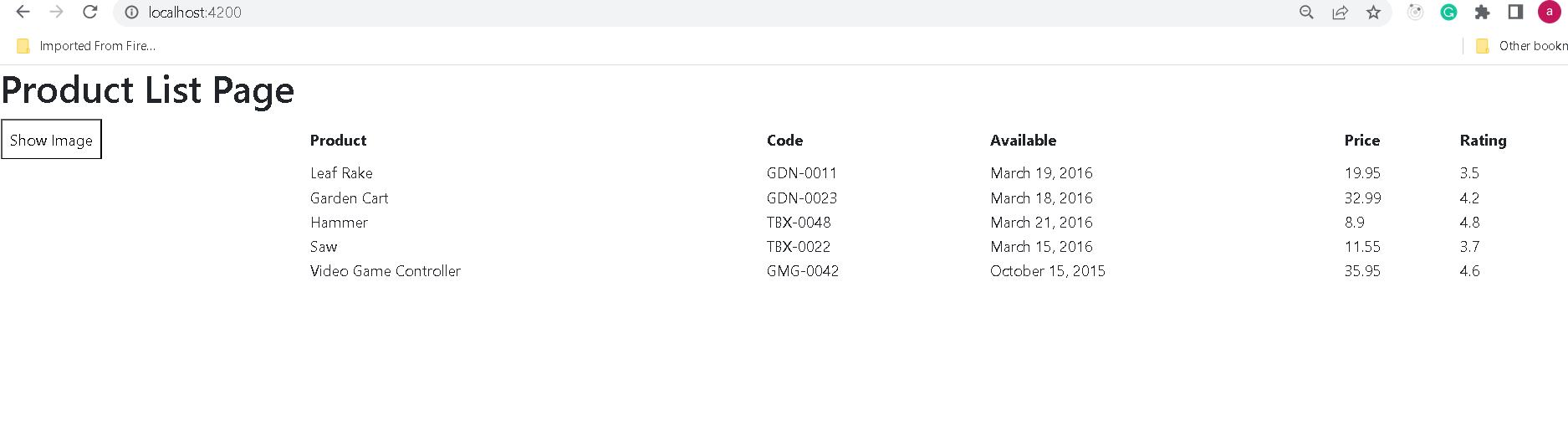
  ]

  toggleImage() {

  }

}

**Output:**



**Step 5.3.2:** Implementing class binding

* Create an external CSS file *product-list.component.css* inside *products* folder.
* Create a CSS class name *price*.

.i.inStock{

*background*: #096d09;

*color*:#ffff;

*Font-size*:15px;

 }

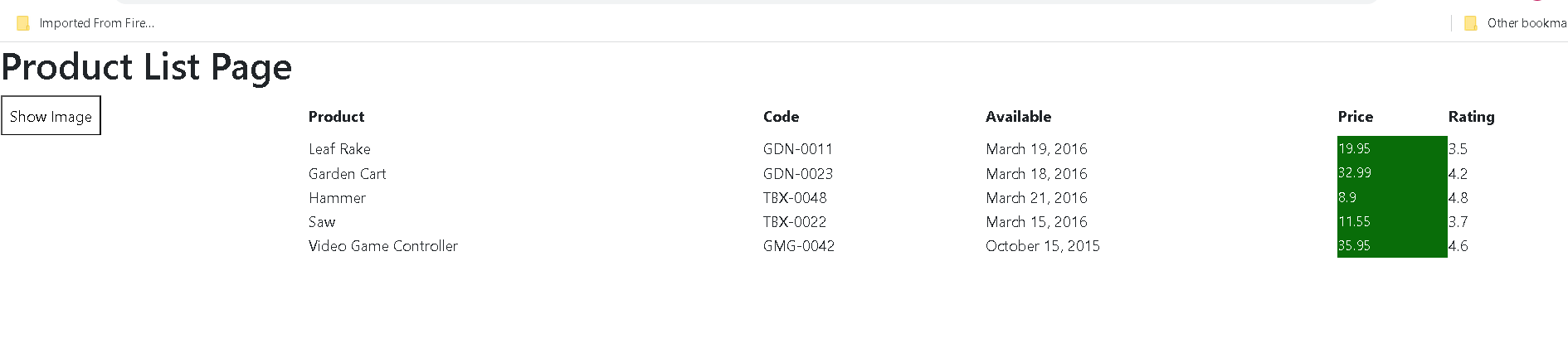
* Apply the CSS class in *Product-list.component.html* file and set its value to *true*.

<td [class.inStock]="true">

{{ product.price}}

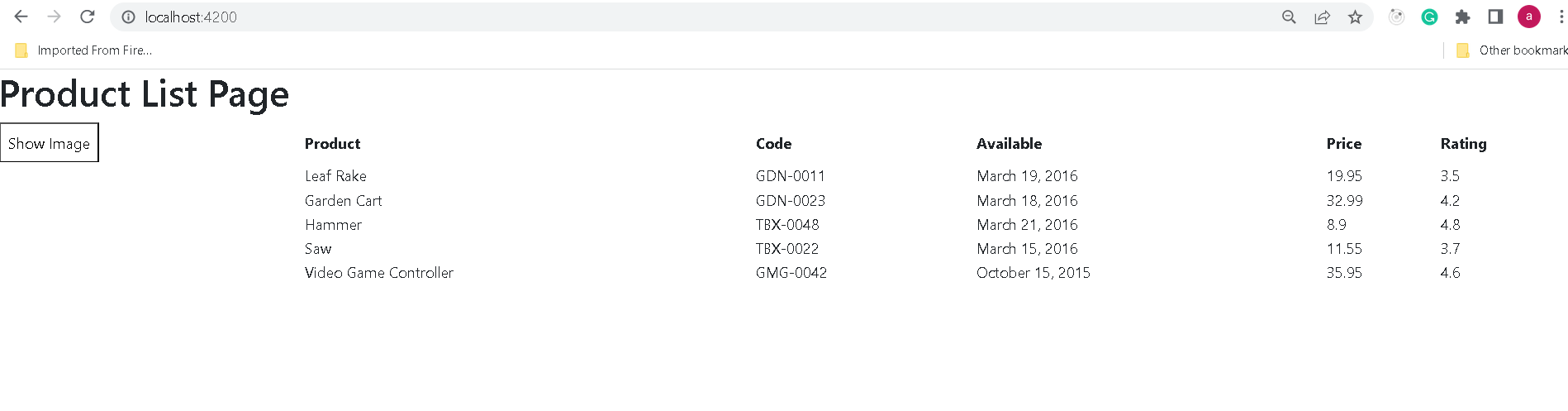
</td>

**Output:**



* Set class value to *false* in *Product-list.component.html*.

**Output:**



**Step 5.3.3:** Implementing style binding

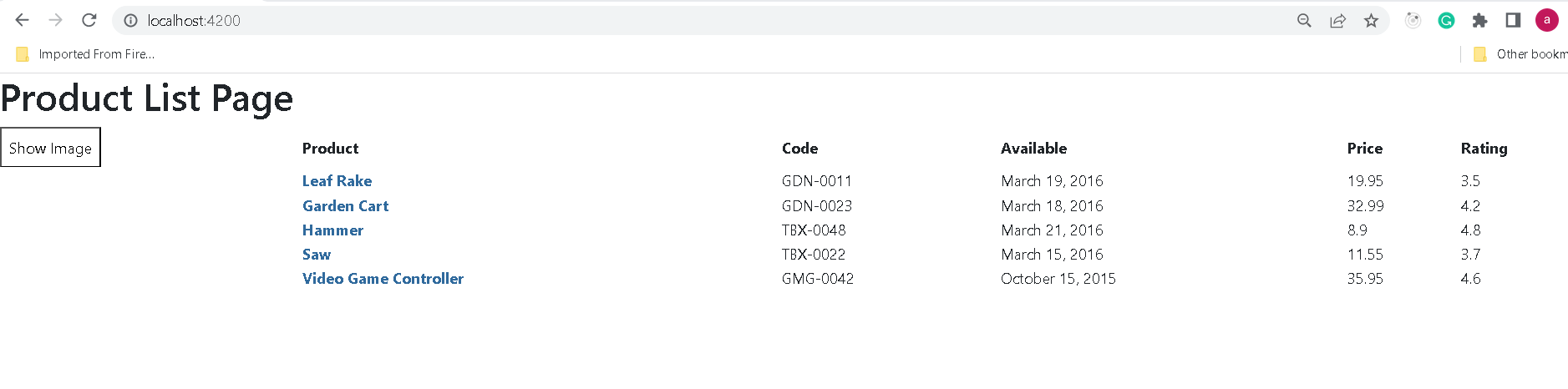
* Bind a color and font-weight style to the product element in *Product-list.component.html* file.

 <td [style.color]="'#306A9D'" [style.font-weight]="700">

            {{ p.productName }}

 </td>

**Output:**



* You can also apply conditional CSS using style binding.

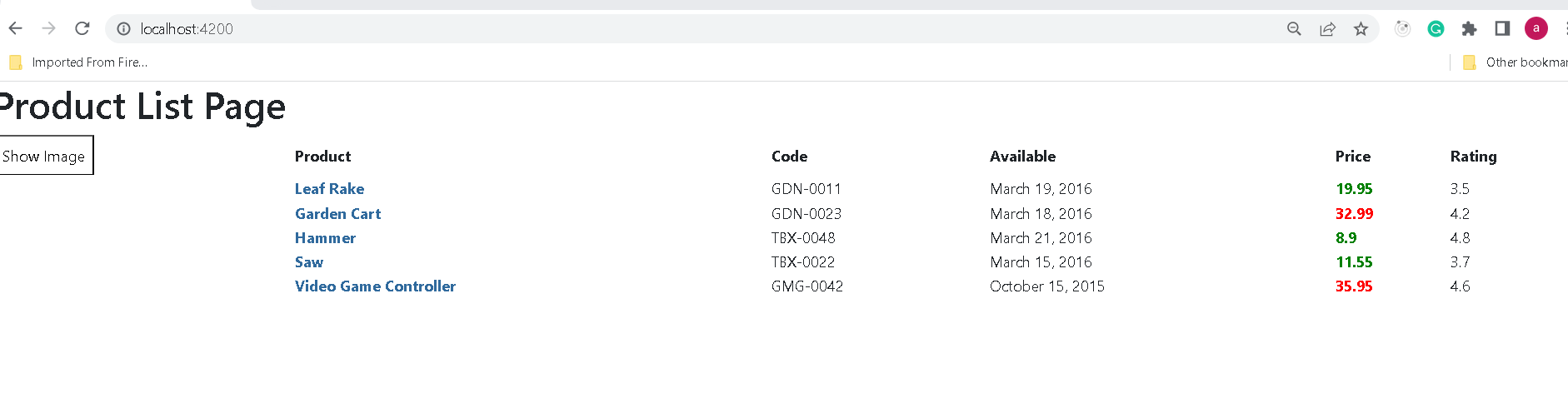
<td [style.color]="p.price > 20 ? 'red': 'green'"

        [style.font-size.px]="16"  [style.font-weight]="700">

        {{ p.price}}

  </td>

**Output:**



**Step 5.3.4:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master