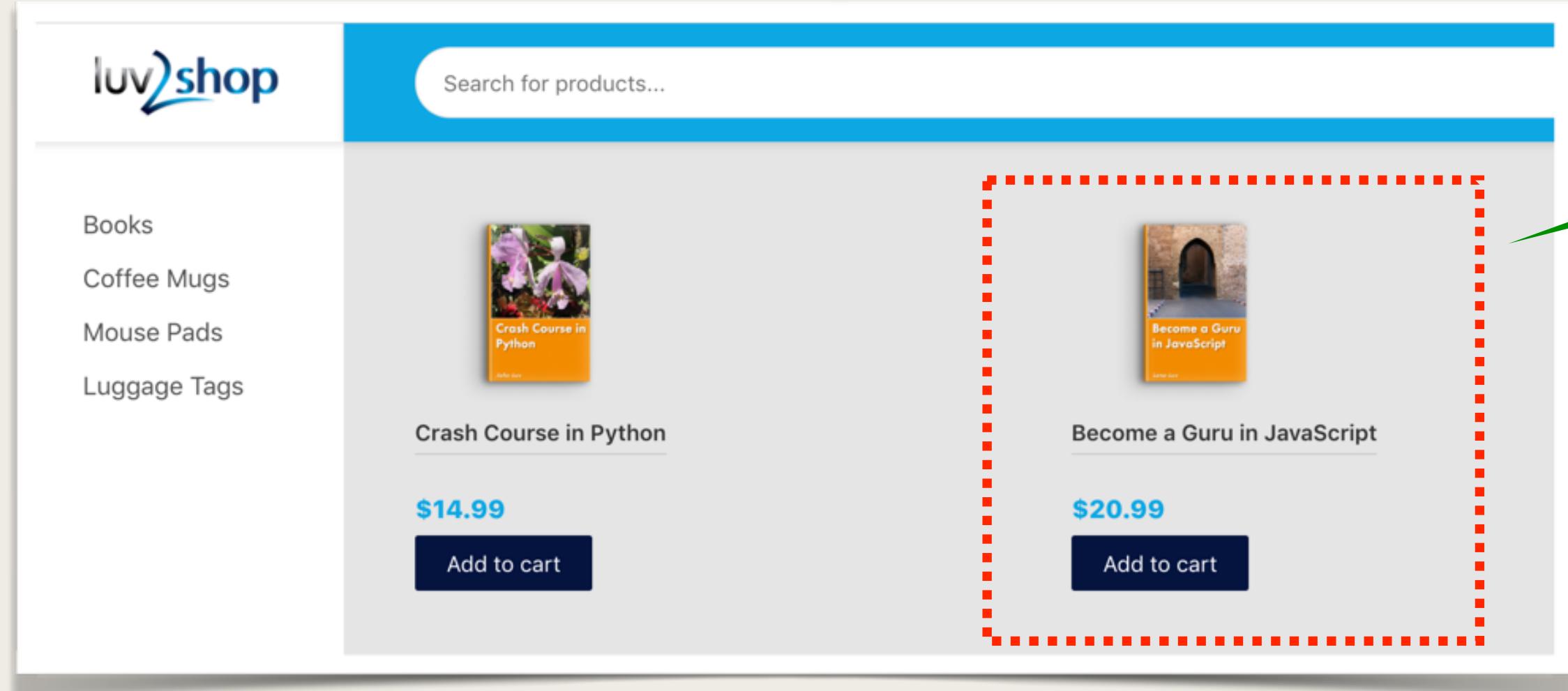


Product Master-Detail View

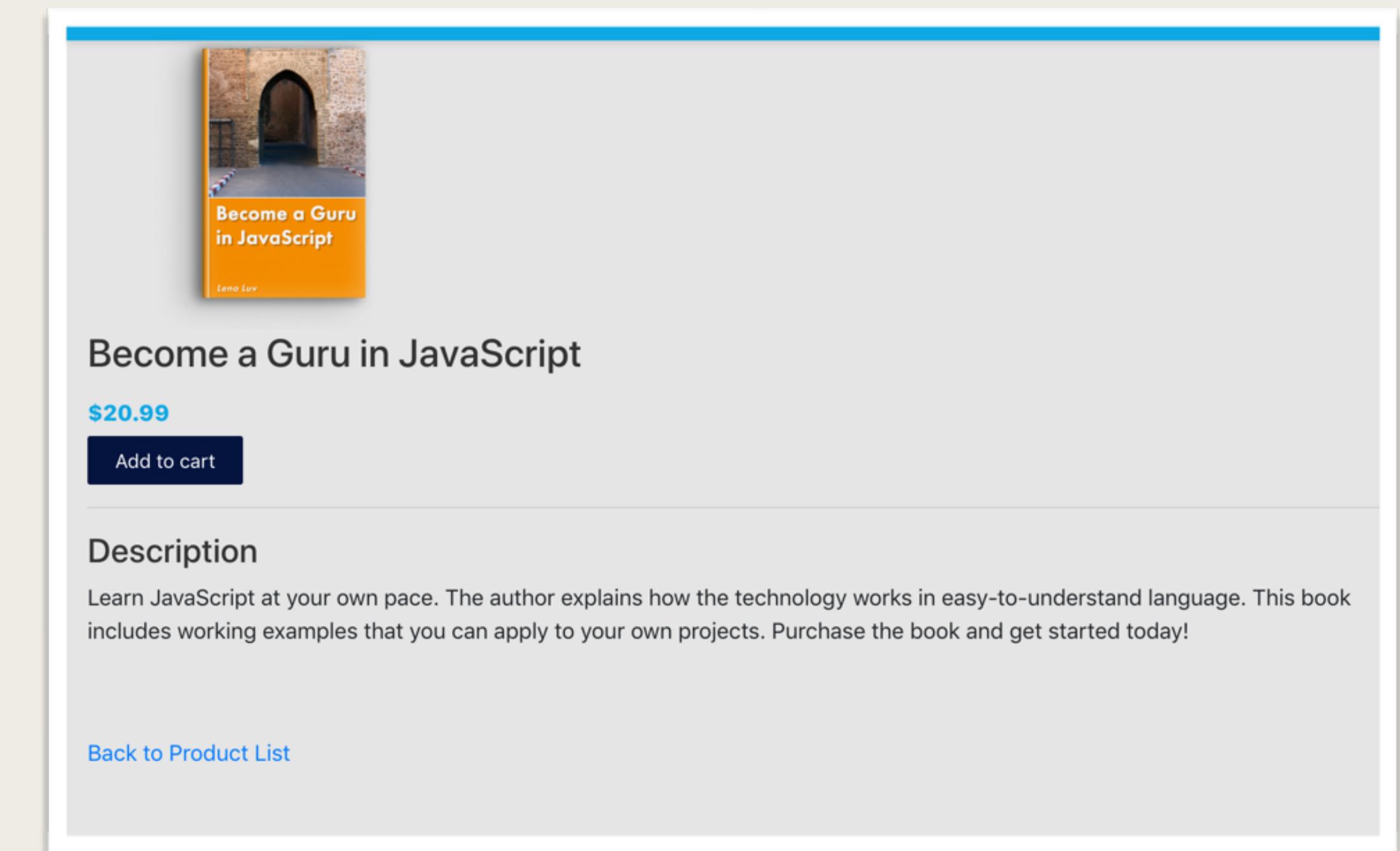


Master-Detail View



Master View

User clicks the product to see details



Detail View

Application shows the details for selected product

Development Process

Step-By-Step

1. Create new component for product details
2. Add new Angular route for product details
3. Add router links to the product-list-grid HTML page
4. Enhance ProductDetailsComponent to retrieve product from ProductService
5. Update ProductService to call URL on Spring Boot app
6. Update HTML page for ProductDetailsComponent to display product details

Step 1: Create new component for product details

```
> ng generate component components/ProductDetails
```

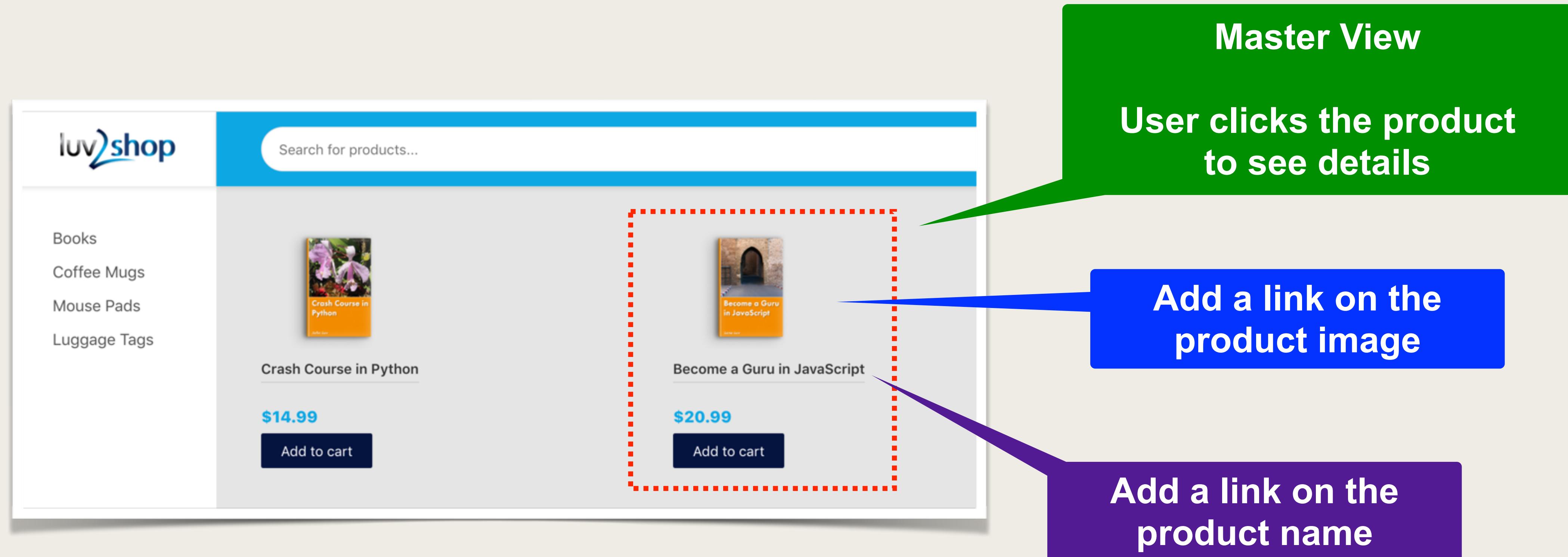
Step 2: Add new Angular route for product details

File: app.module.ts

```
...  
  {path: 'products/:id', component: ProductDetailsComponent},  
...
```

Parameter

Step 3: Add router links to product-list-grid HTML



Step 3: Add router links to product-list-grid HTML

File: product-list-grid.component.html

```
<!-- loop over the collection of products -->
<div *ngFor="let tempProduct of products" class="col-md-3">

  <div class="product-box">

    <a routerLink="/products/{{ tempProduct.id }}">
      
    </a>

    <a routerLink="/products/{{ tempProduct.id }}">
      <h1>{{ tempProduct.name }}</h1>
    </a>
    ...
  </div>

</div>
```

Add a link on the
product image

Add a link on the
product name

Step 3: Add router links to product-list-grid HTML

File: product-list-grid.component.html

```
<!-- loop over the collection of products -->
<div *ngFor="let tempProduct of products" class="col-md-3">

  <div class="product-box">

    <a routerLink="/products/{{ tempProduct.id }}">
      
    </a>

    <a routerLink="/products/{{ tempProduct.id }}">
      <h1>{{ tempProduct.name }}</h1>
    </a>
    ...
  </div>
</div>
```

Parameter

File: app.module.ts

```
...
  {path: 'products/:id', component: ProductDetailsComponent},
...
```

Step 3: Add router links to product-list-grid HTML

File: product-list-grid.component.html

```
<!-- loop over the collection of products -->
<div *ngFor="let tempProduct of products" class="col-md-3">

  <div class="product-box">

    <a routerLink="/products/{{ tempProduct.id }}">
      
    </a>

    <a routerLink="/products/{{ tempProduct.id }}">
      <h1>{{ tempProduct.name }}</h1>
    </a>
    ...
  </div>
</div>
```

Parameter

File: app.module.ts

```
...
  {path: 'products/:id', component: ProductDetailsComponent},
...
```

Step 4: Enhance ProductDetailsComponent to retrieve product from ProductService

File: product-details.component.ts

```
...
// get the "id" param string. convert string to a number using the "+" symbol
const theProductId: number = +this.route.snapshot.paramMap.get('id');

this.productService.getProduct(theProductId).subscribe(
  data => {
    this.product = data;
  }
)
...
```

File: app.module.ts

```
...
{path: 'products/:id', component: ProductDetailsComponent},
...
```

Parameter

Step 5: Update product service to call URL on Spring Boot app

File: product.service.ts

```
getProduct(theProductId: number): Observable<Product> {  
    // need to build URL based on product id  
    const productUrl = `${this.baseUrl}/${theProductId}`;  
  
    return this.httpClient.get<Product>(productUrl);  
}
```

URL for retrieving a product

[http://localhost:8080/api/products/\\${theProductId}](http://localhost:8080/api/products/${theProductId})

Call REST API

Returns an observable

JSON Data return can be converted
directly to Product object

NO NEED to unwrap the JSON
from Spring Data REST

There is no _embedded entry

Mapping JSON to Product class

URL for retrieving product id: 2

<http://localhost:8080/api/products/2>

JSON Data can be converted directly to Product object

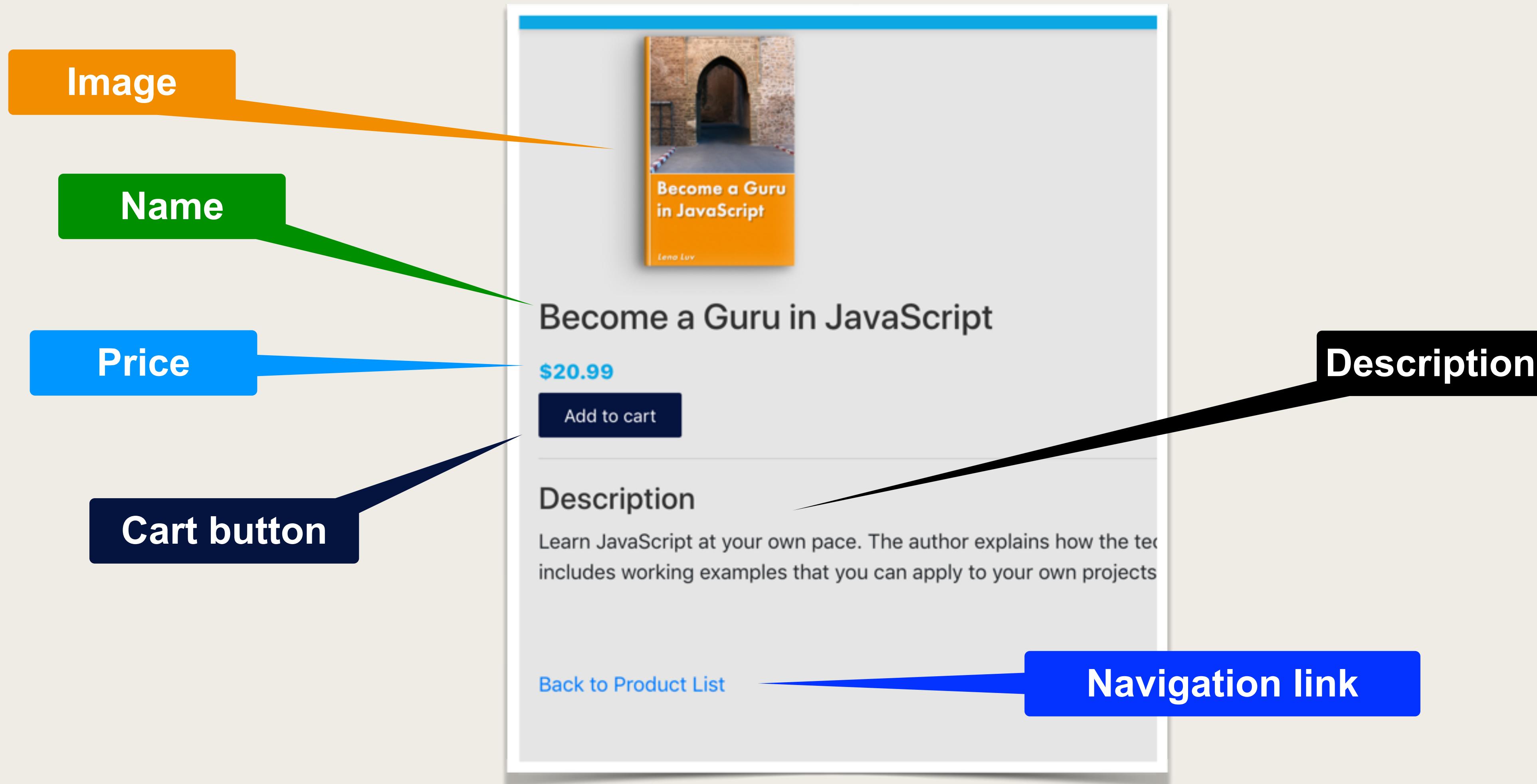
JSON properties map directly
to
properties in Product class

```
{  
    "id" : 2,  
    "sku" : "BOOK-TECH-1001",  
    "name" : "Become a Guru in JavaScript",  
    "description" : "Learn JavaScript at your own pace. The author explains how the  
technology works in easy-to-understand language. This book includes working examples  
that you can apply to your own projects. Purchase the book and get started today!",  
    "unitPrice" : 20.99,  
    "imageUrl" : "assets/images/products/books/book-luv2code-1001.png",  
    "active" : true,  
    "unitsInStock" : 100,  
    "dateCreated" : "2020-01-11T16:58:24.000+0000",  
    "lastUpdated" : null,  
    "links" : {  
        "self" : {  
            "href" : "http://localhost:8080/api/products/2"  
        },  
        "product" : {  
            "href" : "http://localhost:8080/api/products/2"  
        },  
        "category" : {  
            "href" : "http://localhost:8080/api/products/2/category"  
        }  
    }  
}
```

File: product.ts

```
export class Product {  
    sku: string;  
    name: string;  
    description: string;  
    unitPrice: number;  
    imageUrl: string;  
    active: boolean;  
    unitsInStock: number;  
    dateCreated: Date;  
    lastUpdated: Date;  
}
```

Step 6: Update HTML page for ProductDetailsComponent to display product details



Step 6: Update HTML page for ProductDetailsComponent to display product details

File: product-details.component.html

```
<div class="detail-section">
  <div class="container-fluid">

    
    <h3>{{ product.name }}</h3>
    <div class="price">{{ product.unitPrice | currency:'USD' }}</div>
    <a href="#" class="primary-btn">Add to cart</a>

    <hr>
    <h4>Description</h4>
    <p>{{ product.description }}</p>

    <a routerLink="/products" class="mt-5">Back to Product List</a>

  </div>
</div>
```

