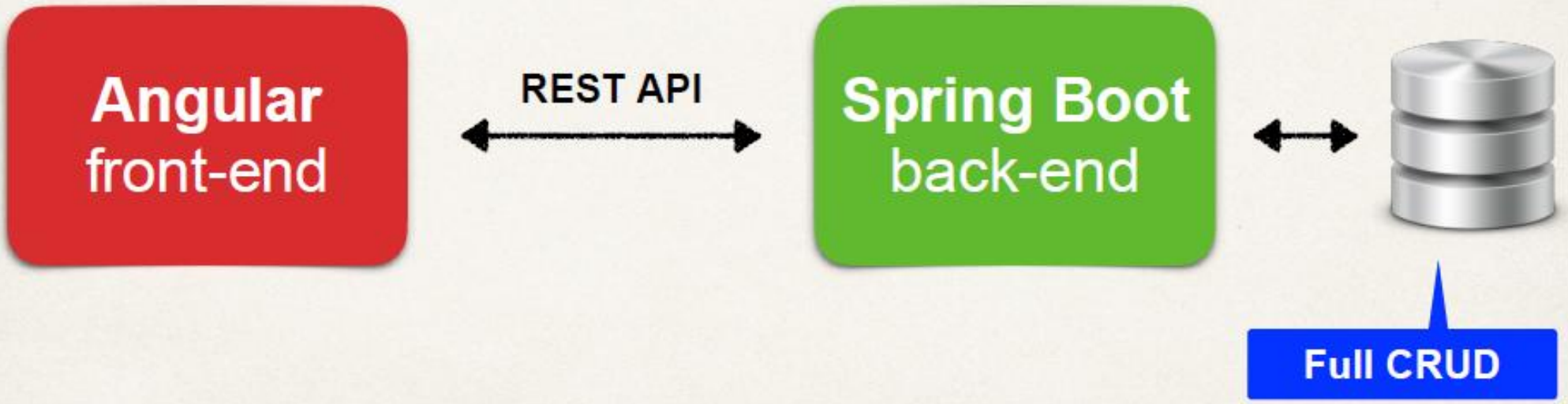
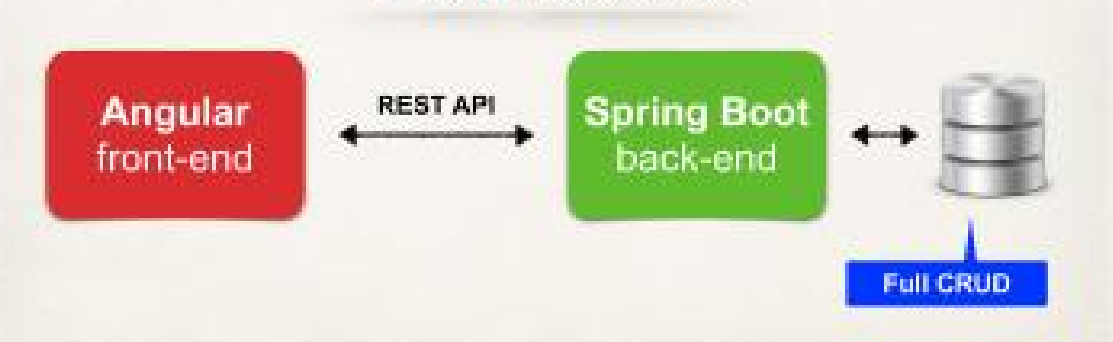
- Build a Full-Stack application with Angular and Spring Boot

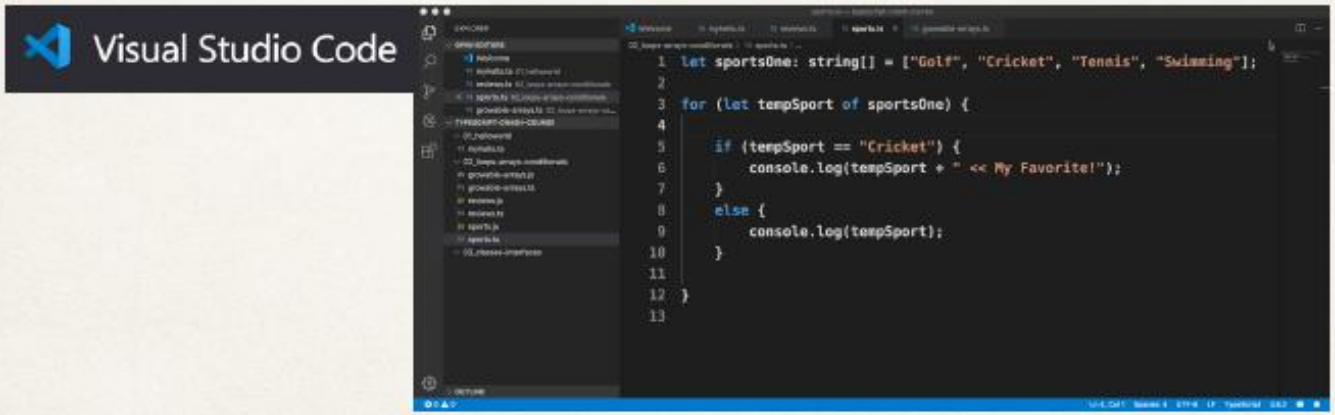


Full-Stack



Visual Studio Code

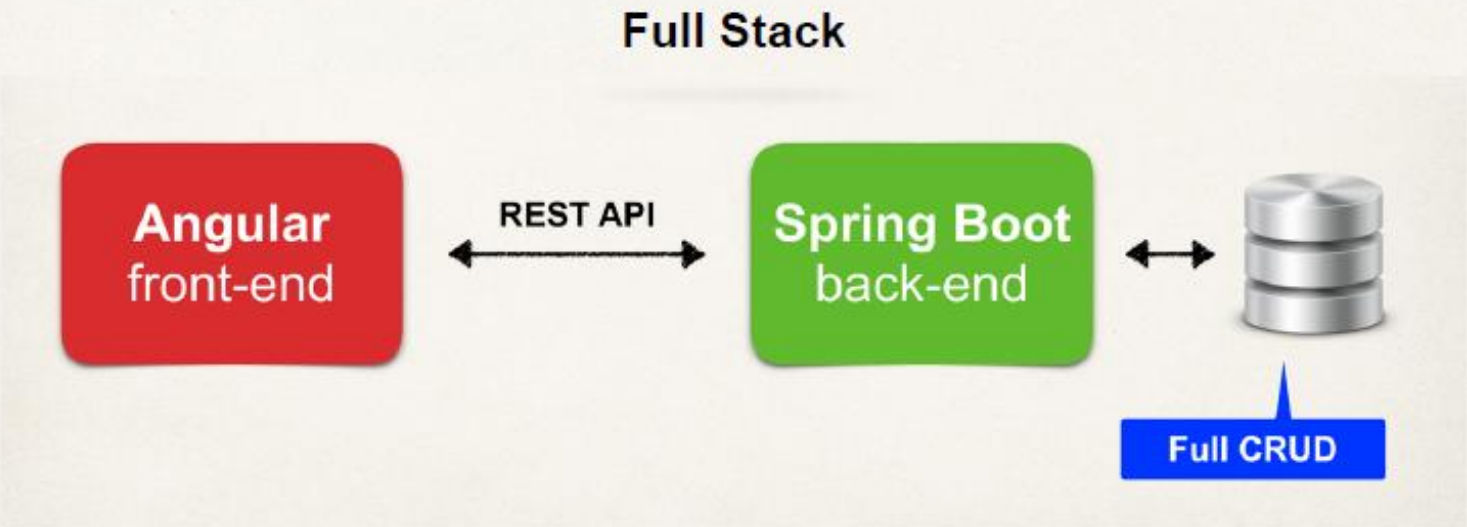
- Free IDE that supports multiple programming languages
- Has built-in support for **TypeScript**
- IDE features such as IntelliSense, Debugging, etc …



http://code.visualstudio.com

Project

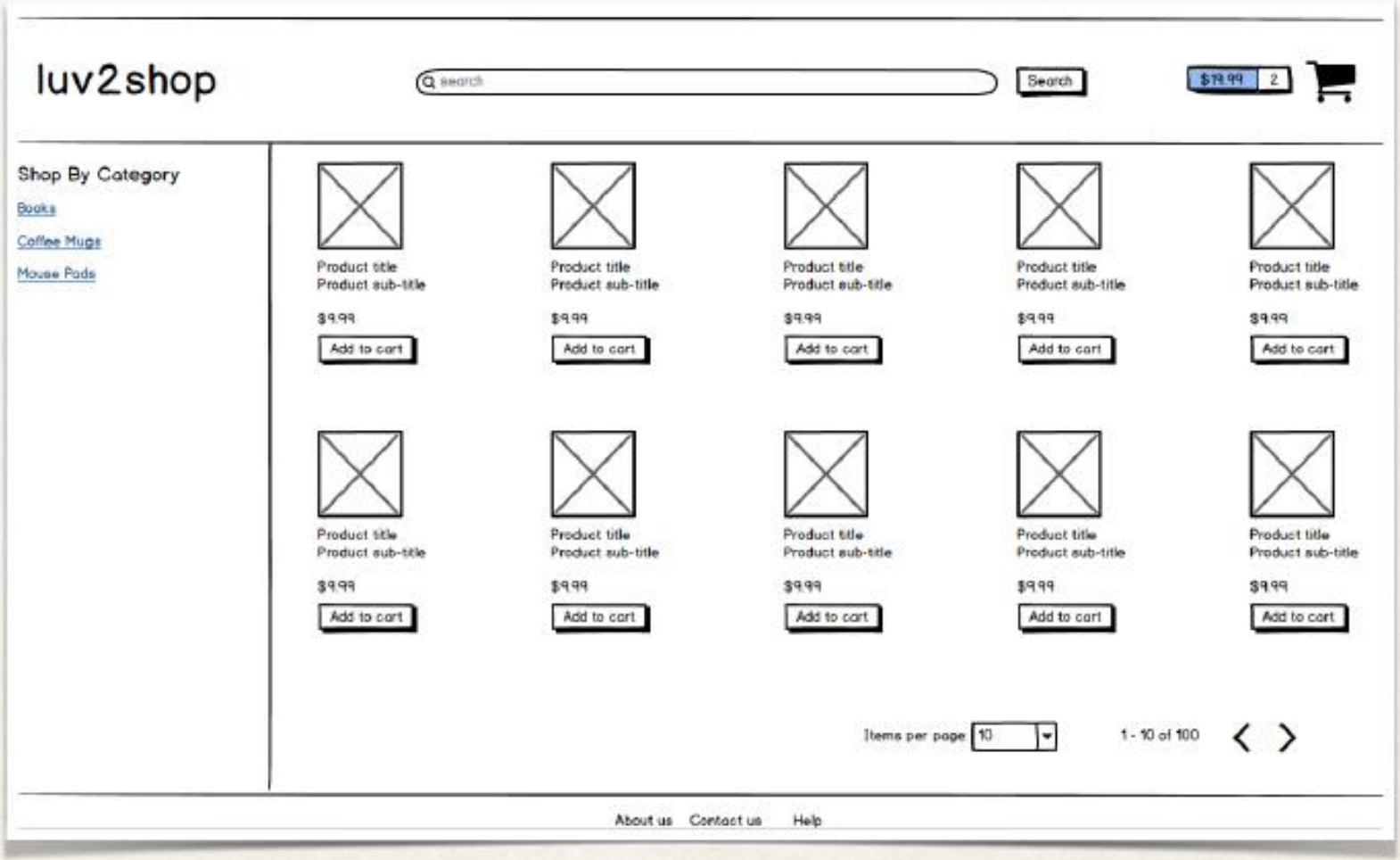
- Build Real-time eCommerce App



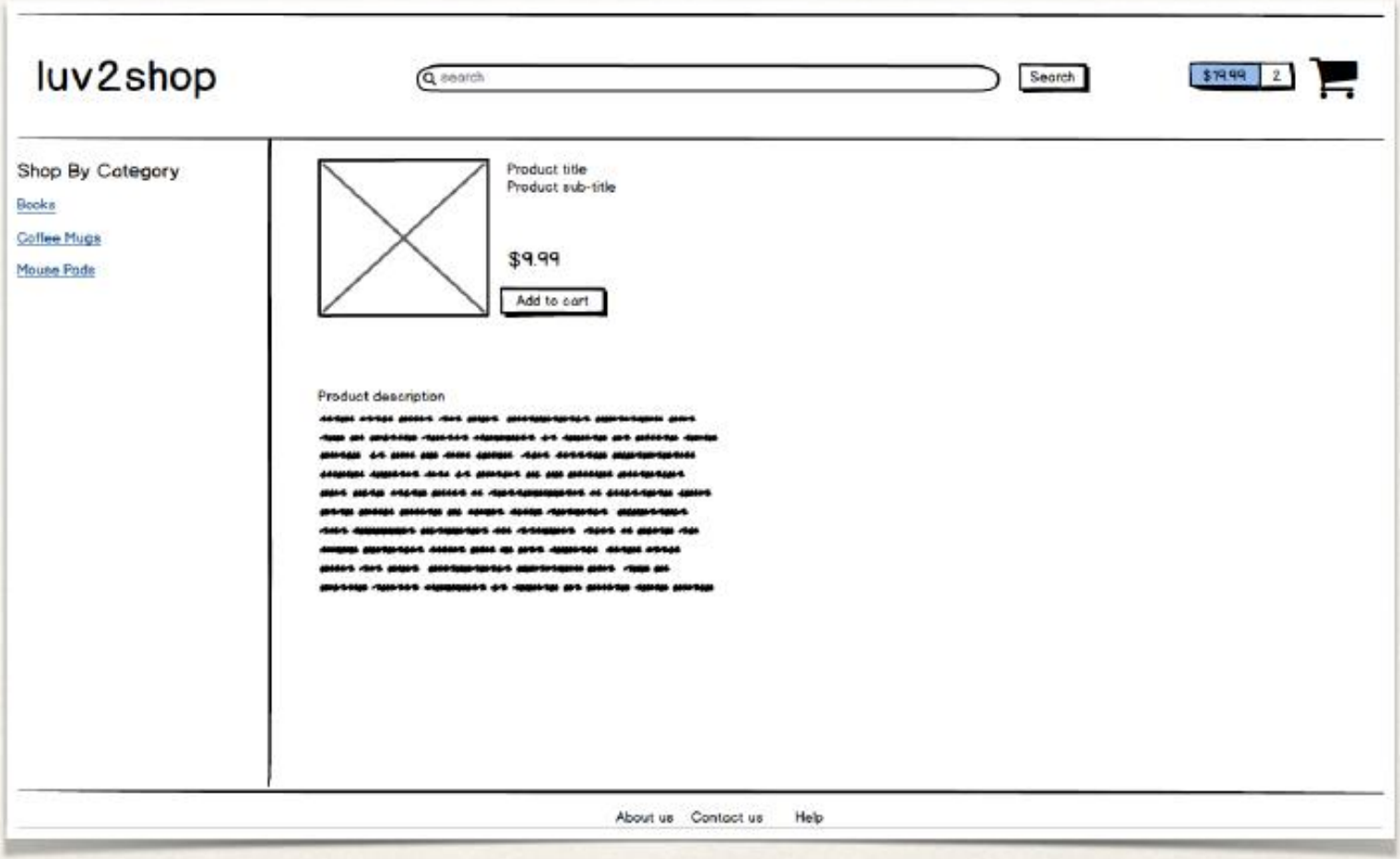
Requirements

- Show a list of products
- Add products to shopping cart (CRUD)
- Shopping cart check out
- User login/logout security
- Track previous orders for logged in users

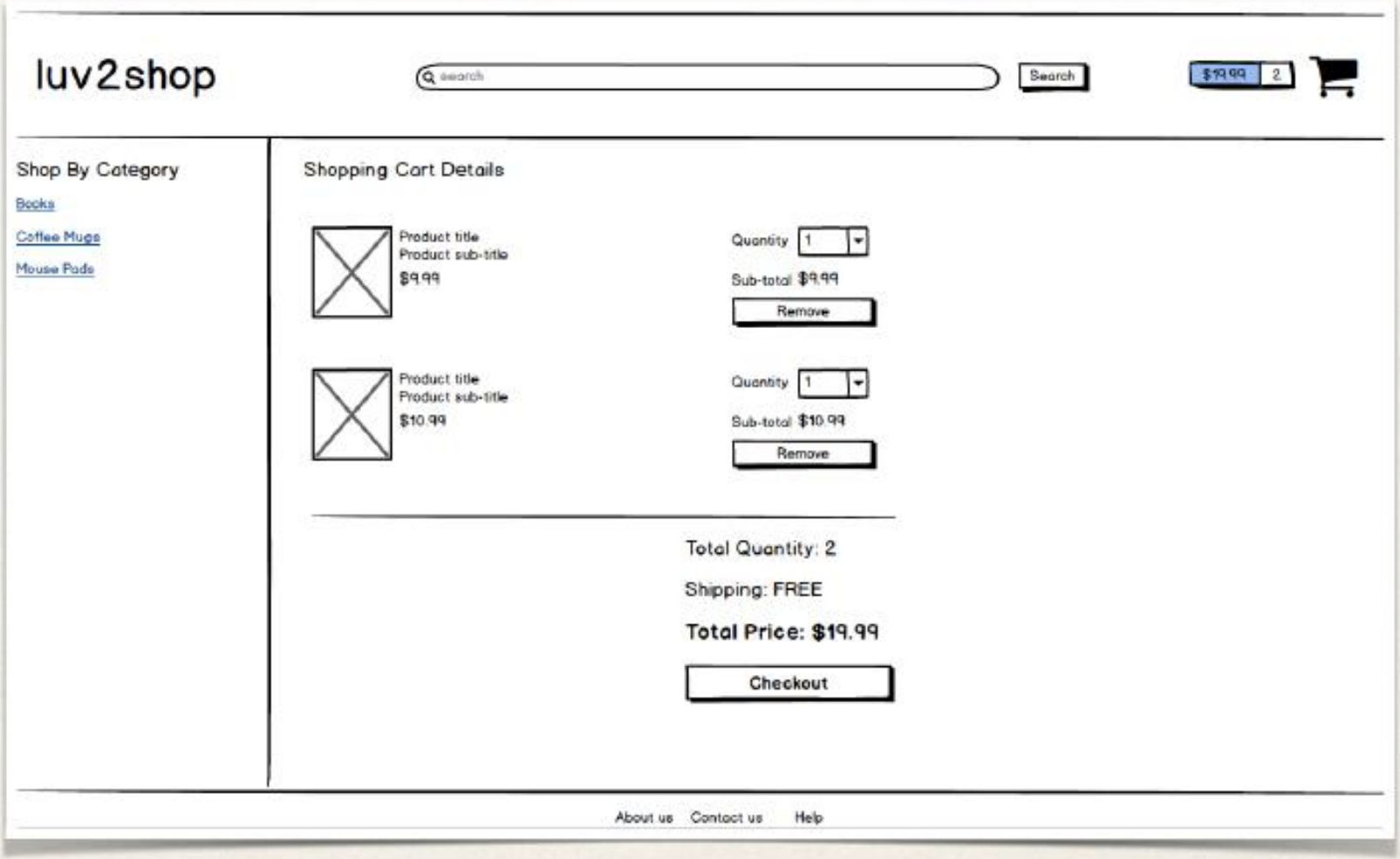
Wireframes - Home Page



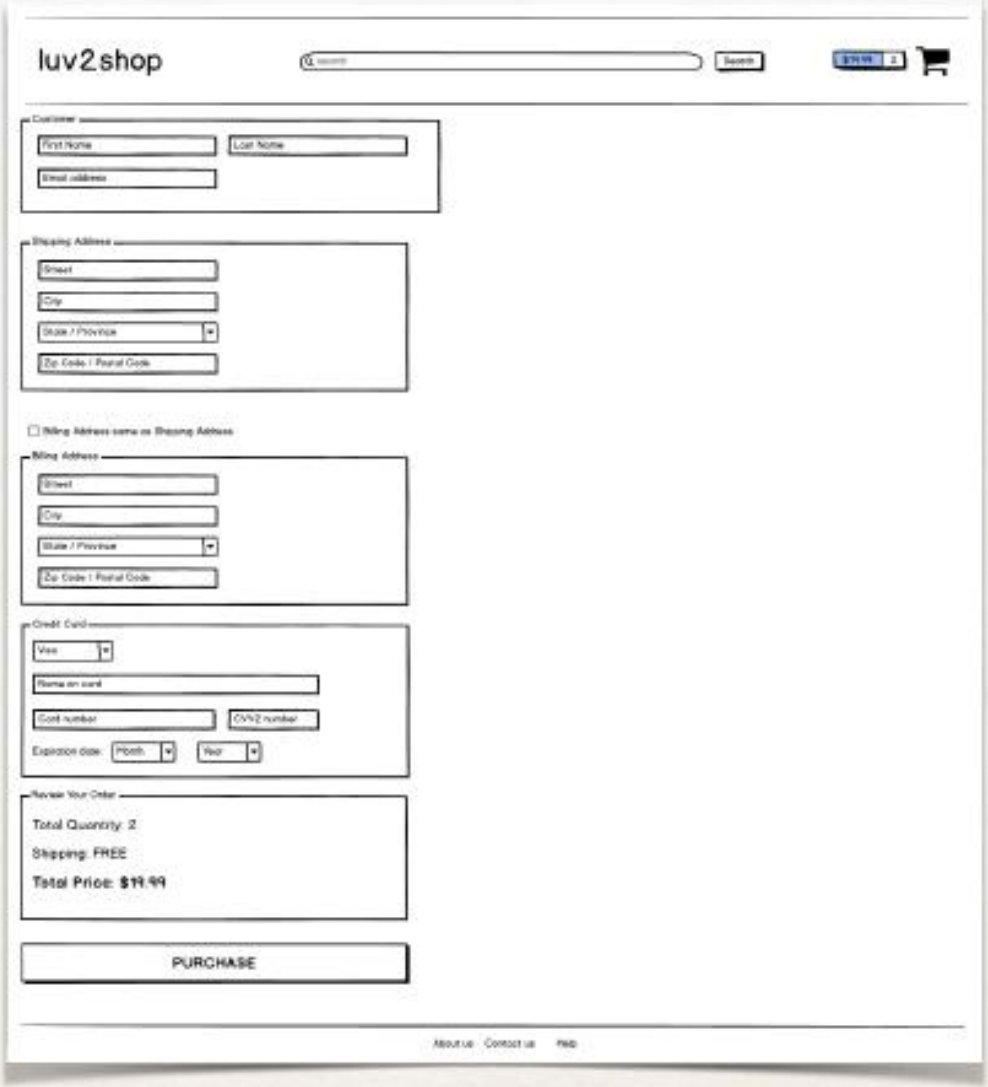
Wireframes - Product Details



Wireframes - Shopping Cart Details



Wireframes - Check Out

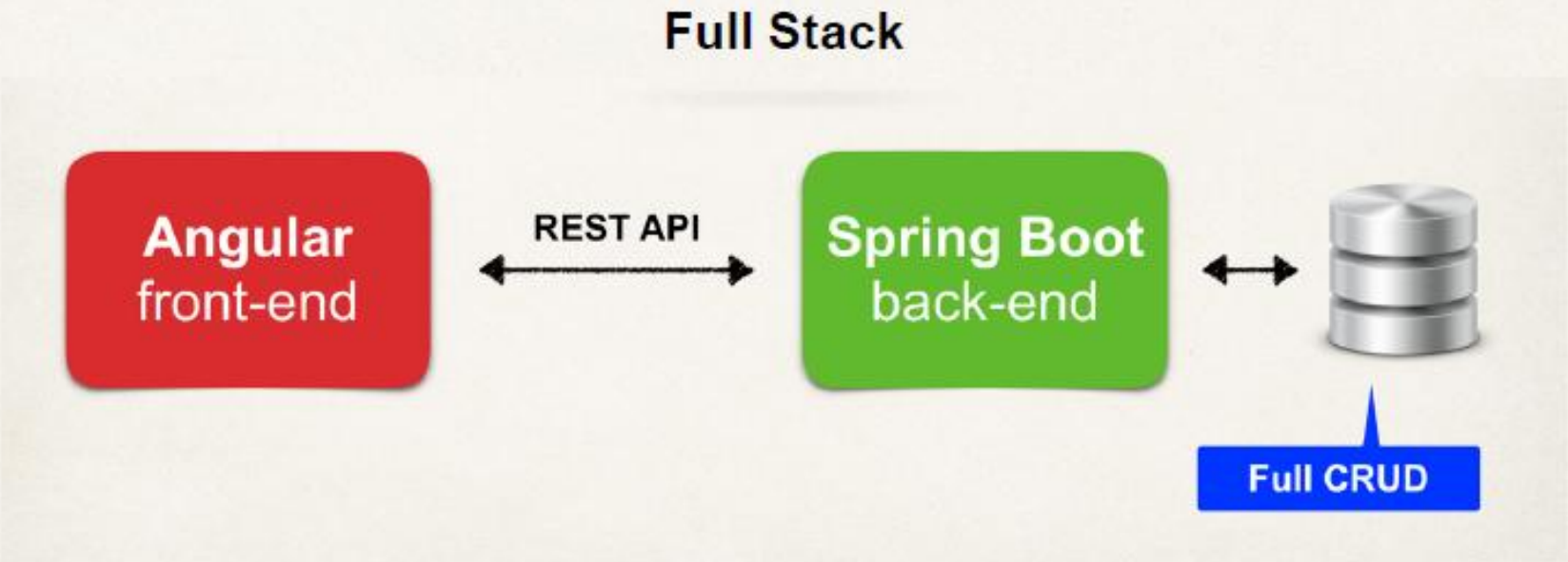


Spring Boot Back End

- You should have the following items already installed
  - Java Development Kit (JDK)
  - Java IDE (we'll use IntelliJ in the videos, but any Java IDE will work)
  - Maven
  - MySQL Database and MySQL Workbench

Spring Boot Back End

- Leverage Spring Data REST for REST API
- Minimizes the coding for Spring Boot back end



Create Repository

- Spring Data REST will scan your project for **JpaRepository**
- Expose REST APIs for each entity type for your **JpaRepository**

```
public interface ProductRepository extends JpaRepository<Product, Long> {

}
```

REST Endpoints

- By default, Spring Data REST will create endpoints based on entity type
- Simple pluralized form
  - First character of Entity type is lowercase
  - Then just adds an "s" to the entity

```
public interface ProductRepository extends JpaRepository<Product, Long> {

}
```

/products

REST API

- Spring Data REST will expose these endpoints for free!

HTTP Method		CRUD Action
POST	/products	**C**reate a new product
GET	/products	**R**ead a list of products
GET	/products/{id}	**R**ead a single product
PUT	/products/{id}	**U**pdate an existing product
DELETE	/products/{id}	**D**elete an existing product


@ManyToOne
Many products belong to one category
product
id BIGINT(20)
sku VARCHAR(255)
name VARCHAR(255)
description VARCHAR(255)
unit_price DECIMAL(13,2)
image_url VARCHAR(255)
active BIT(1)
units_in_stock INT(11)
date_created DATETIME(6)
last_updated DATETIME(6)
category_id BIGINT(20)
Indexes
product_category
id BIGINT(20)
category_name VARCHAR(255)
Indexes
@OneToMany
One category has many products

Two Database Scripts

- 01-create-user.sql
- 02-create-products.sql

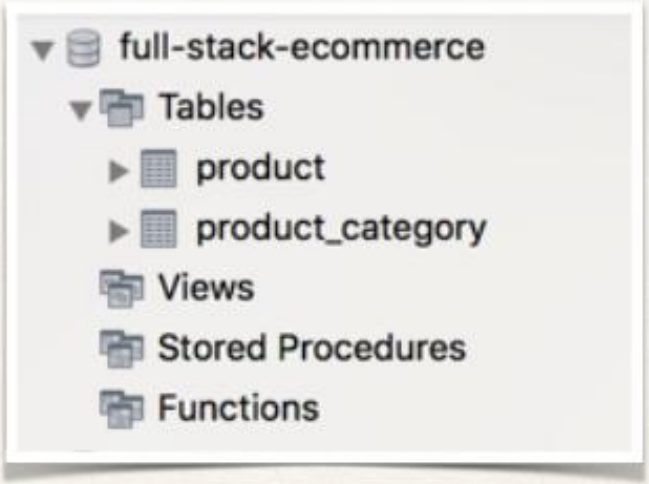
About: 01-create-user.sql

1. Create a new MySQL user for our application
   - user id: **ecommerceapp**
   - password: **ecommerceapp**

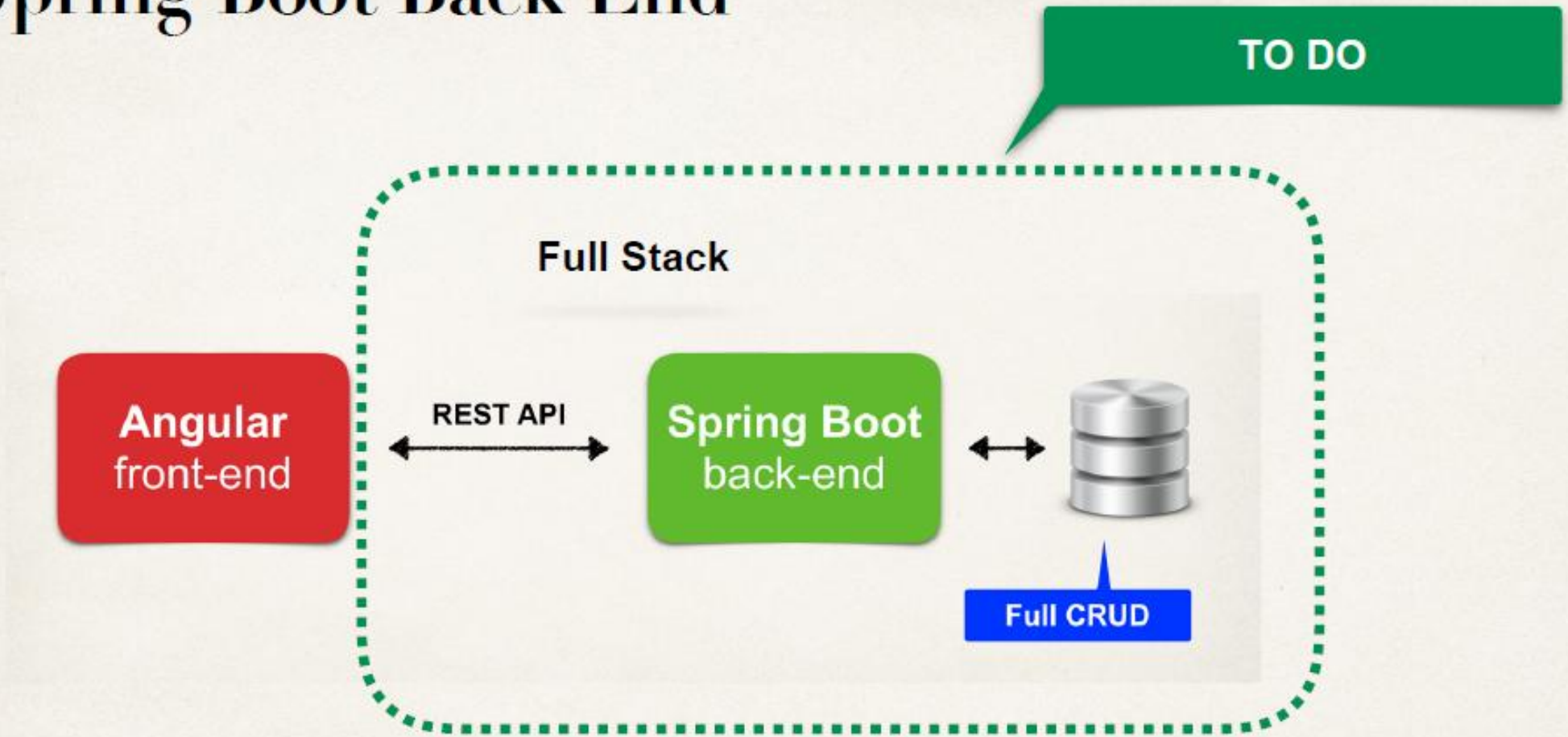
About: 02-create-products.sql

1. Create new database tables: **product, product_category**

2. Load tables with sample data



Spring Boot Back End



Development Process

Step-By-Step

1. Set up the database tables
2. Create a Spring Boot starter project (start.spring.io)

```
spring-boot-starter-data-jpa
spring-boot-starter-data-rest
mysql-connector-java
lombok
```

3. Develop the Entities: **Product** and **ProductCategory**
4. Create REST APIs with Spring Data JPA Repositories and Spring Data REST

Project Lombok

- Modern Java project
- Lombok automagically generates the getters/setters (behind the scenes)
- No need for the developer to manually define getters/setters, etc ...
- Easy-to-use Annotations to eliminate boilerplate code

http://www.projectlombok.org

Project Lombok

Before Lombok

```java
@Entity
@Table(name = "product")
public class Product {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name="id")
    private Long id;

    @Column(name = "name")
    @NotBlank
    private String name;

    public Product() {
    }

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass())
          return false;
        Product product = (Product) o;
        return Objects.equals(id, product.id) &&
                Objects.equals(name, product.name);
    }

    @Override
    public int hashCode() {
        return Objects.hash(id, name);
    }
}
```

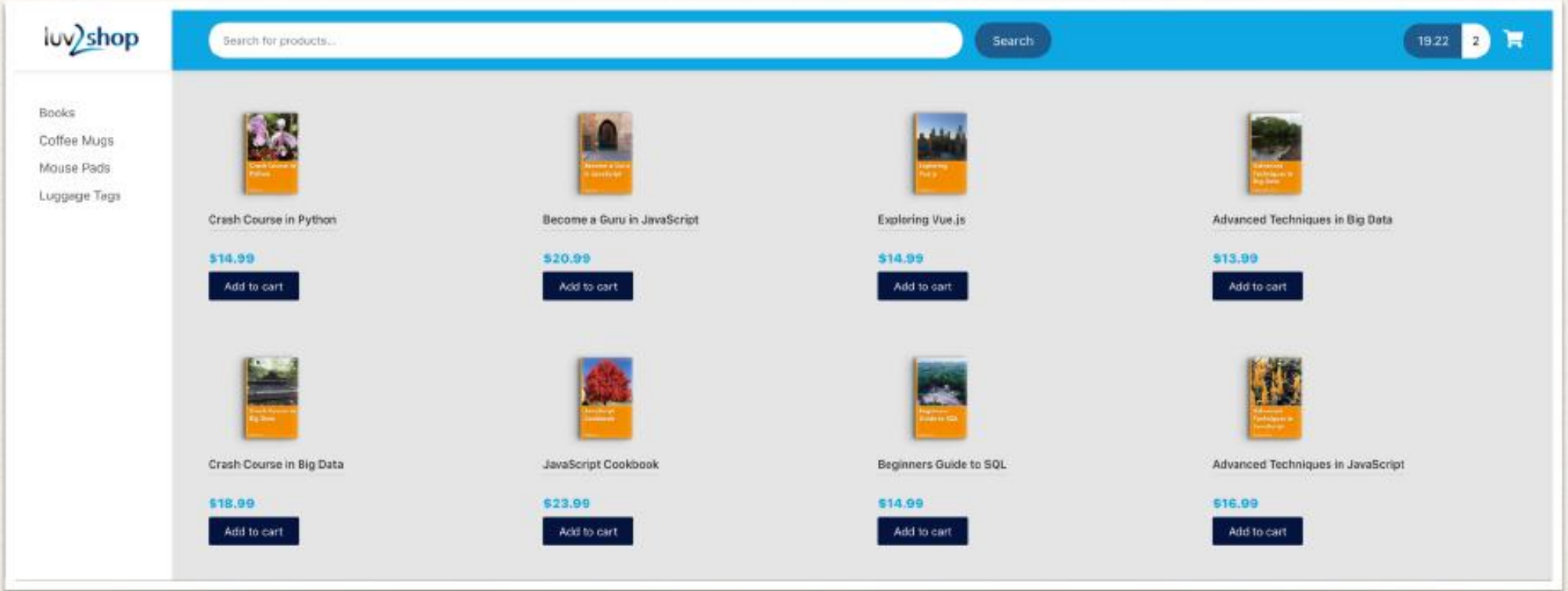
After Lombok

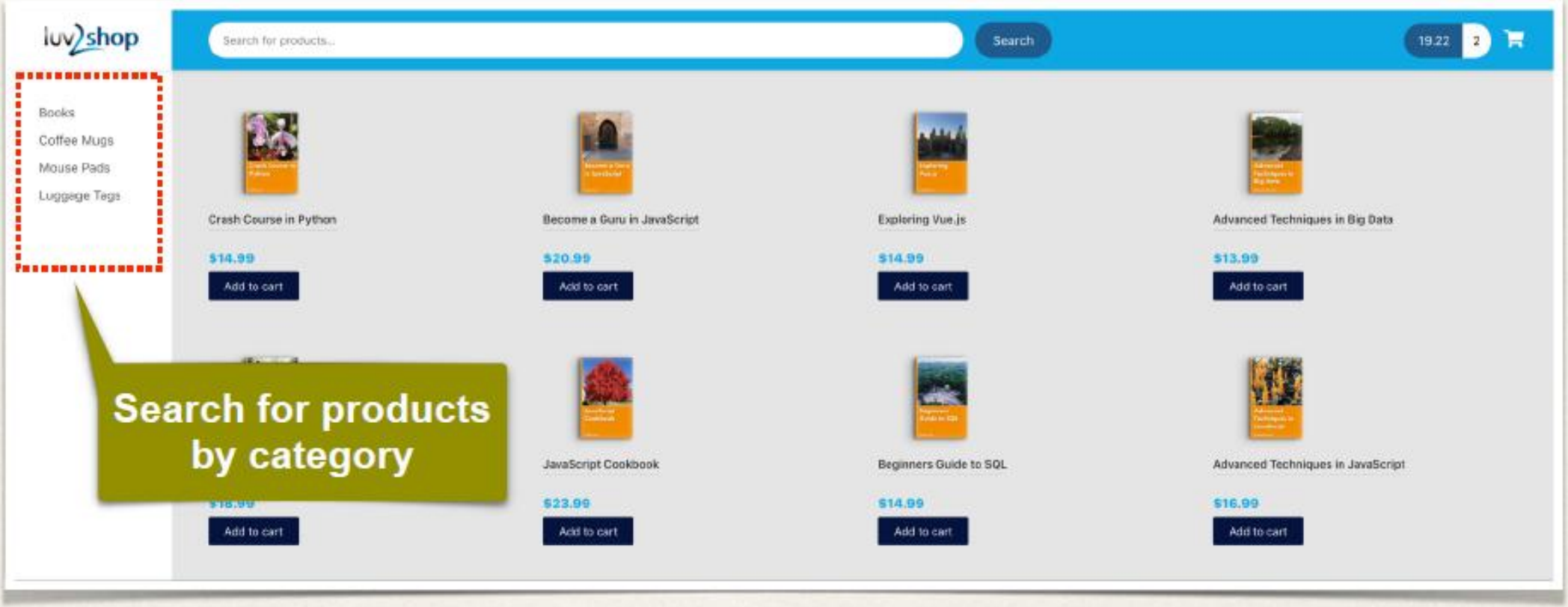
Lombok annotation

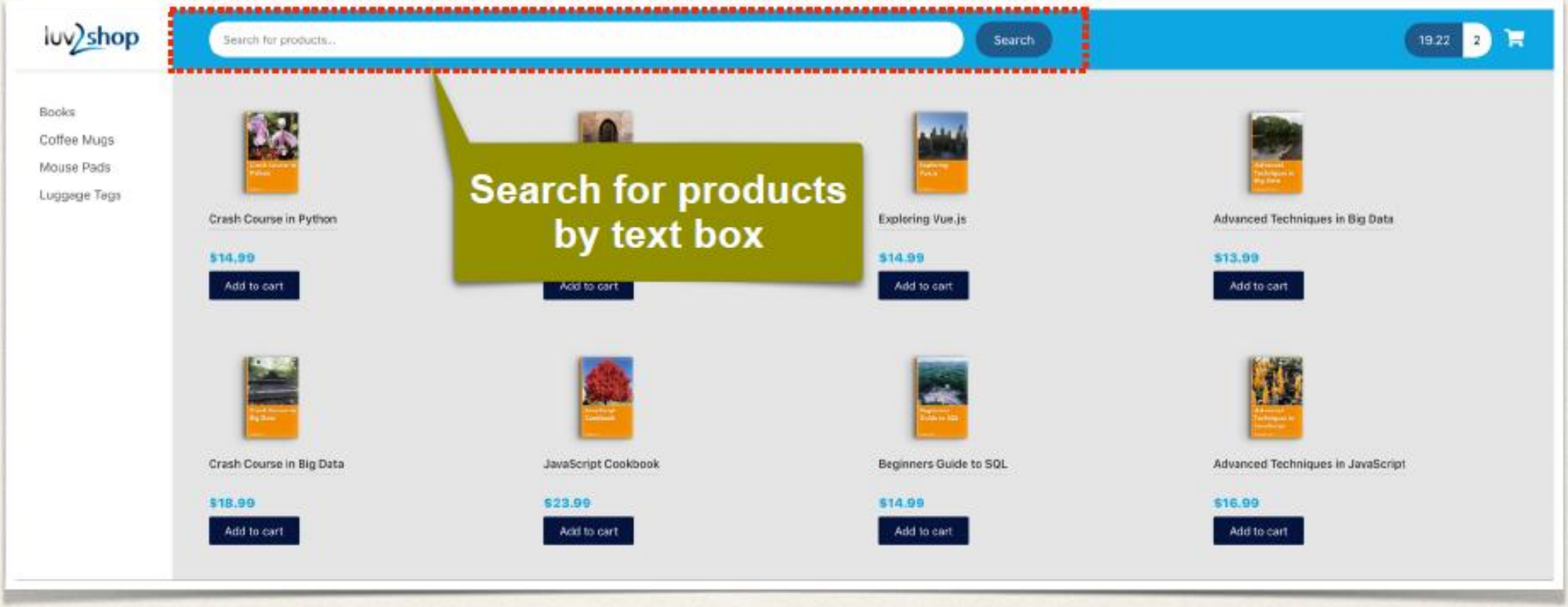
```java
@Entity
@Table(name = "product")
@Data
public class Product {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name="id")
    private Long id;

    @Column(name = "name")
    @NotBlank
    private String name;
}
```

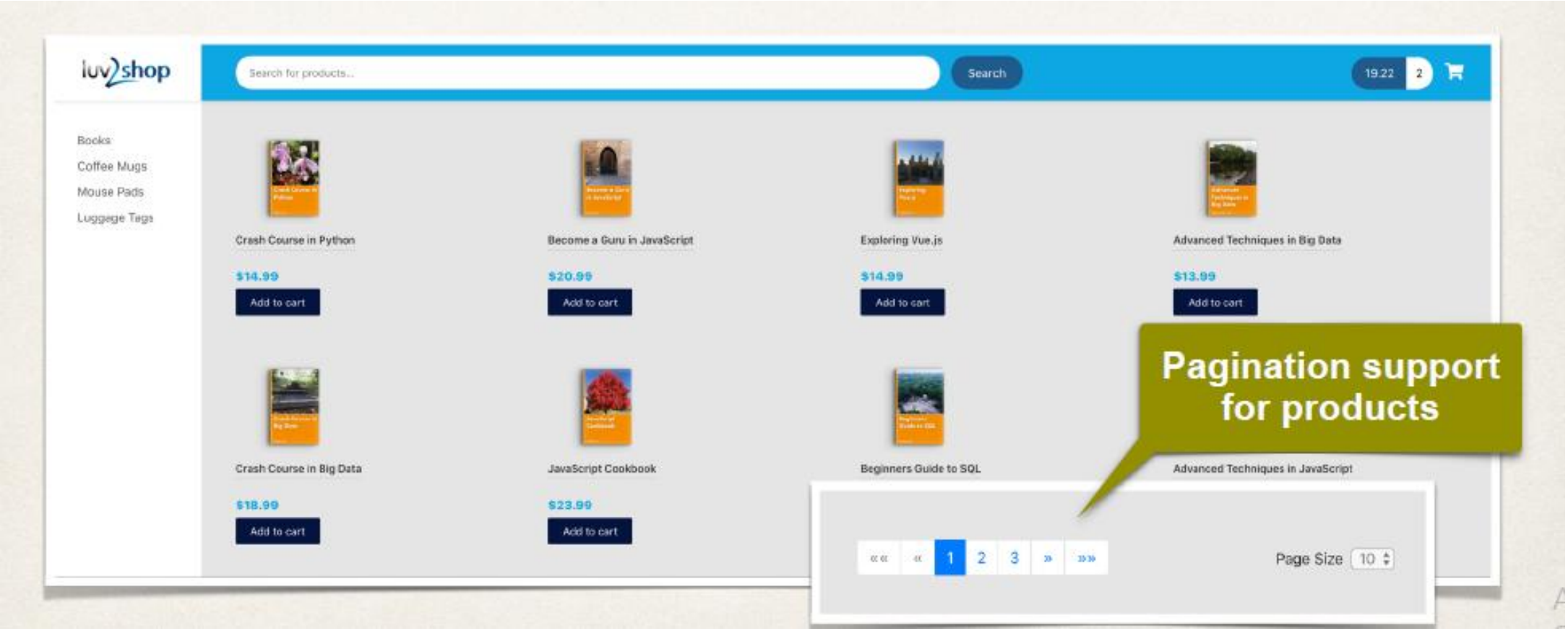
That's It!!!
Absolutely no need to generate getters and setters

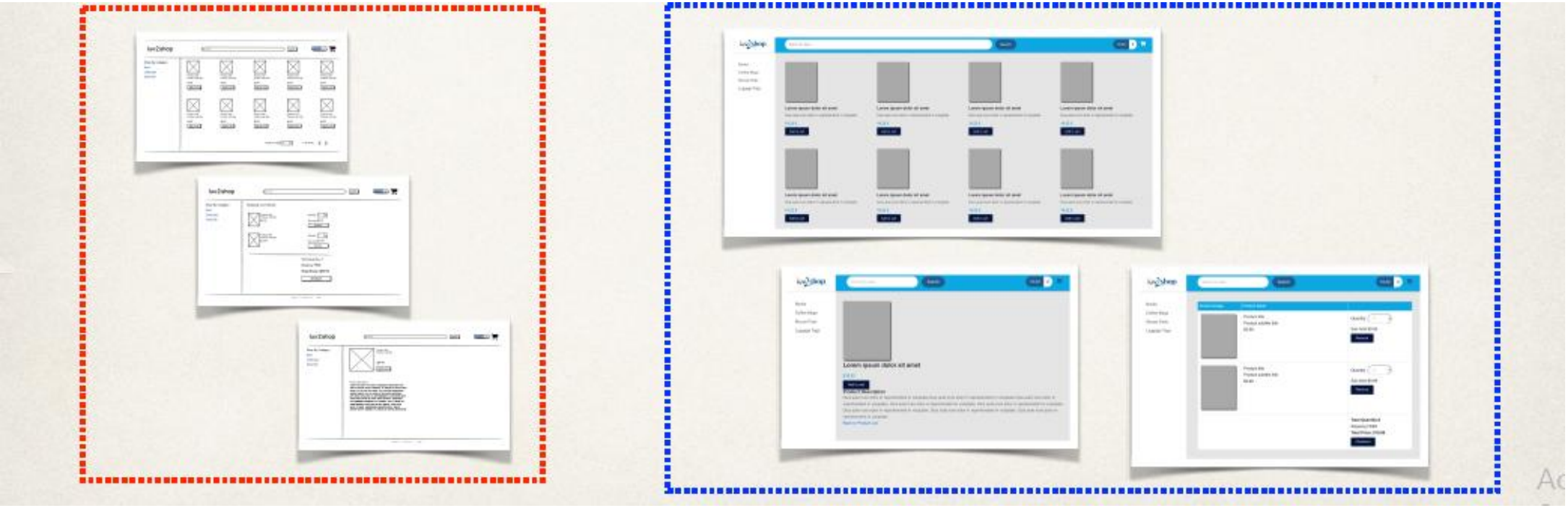
Lombok will do this work for you automagically behind the scenes


luv2shop
Search for products...
Search
19.22
2
Books
Coffee Mugs
Mouse Pads
Luggage Tags
Crash Course in Python
$14.99
Add to cart
Become a Guru in JavaScript
$20.99
Add to cart
Exploring Vue.js
$14.99
Add to cart
Advanced Techniques in Big Data
$13.99
Add to cart
Crash Course in Big Data
$18.99
Add to cart
JavaScript Cookbook
$23.99
Add to cart
Beginners Guide to SQL
$14.99
Add to cart
Advanced Techniques in JavaScript
$16.99
Add to cart


luv2shop
Search for products...
Search
19.22
2
Books
Coffee Mugs
Mouse Pads
Luggage Tags
Crash Course in Python
$14.99
Add to cart
Become a Guru in JavaScript
$20.99
Add to cart
Exploring Vue.js
$14.99
Add to cart
Advanced Techniques in Big Data
$13.99
Add to cart
Add to cart
JavaScript Cookbook
$23.99
Add to cart
Beginners Guide to SQL
$14.99
Add to cart
Advanced Techniques in JavaScript
$16.99
Add to cart
Search for products by category


luv2shop
Search for products...
Search
19.22
2
Books
Coffee Mugs
Mouse Pads
Luggege Tags
Search for products by text box
Crash Course in Python
$14.99
Add to cart
Exploring Vue.js
$14.99
Add to cart
Advanced Techniques in Big Data
$13.99
Add to cart
Crash Course in Big Data
$18.99
Add to cart
JavaScript Cookbook
$23.99
Add to cart
Beginners Guide to SQL
$14.99
Add to cart
Advanced Techniques in JavaScript
$16.99
Add to cart


luv2shop
Search for products...
Search
19.22
2
Books
Coffee Mugs
Mouse Pads
Luggage Tags
Crash Course in Python
$14.99
Add to cart
Become a Guru in JavaScript
$20.99
Add to cart
Exploring Vue.js
$14.99
Add to cart
Advanced Techniques in Big Data
$13.99
Add to cart
Crash Course in Big Data
$18.99
Add to cart
JavaScript Cookbook
$23.99
Add to cart
Beginners Guide to SQL
Advanced Techniques in JavaScript
Pagination support for products
1
2
3
Page Size
10



www.w3schools.com/css

www.w3schools.com/bootstrap4

www.luv2code.com