[Thực hành] Thêm sản phẩm vào giỏ hàng

Mục tiêu

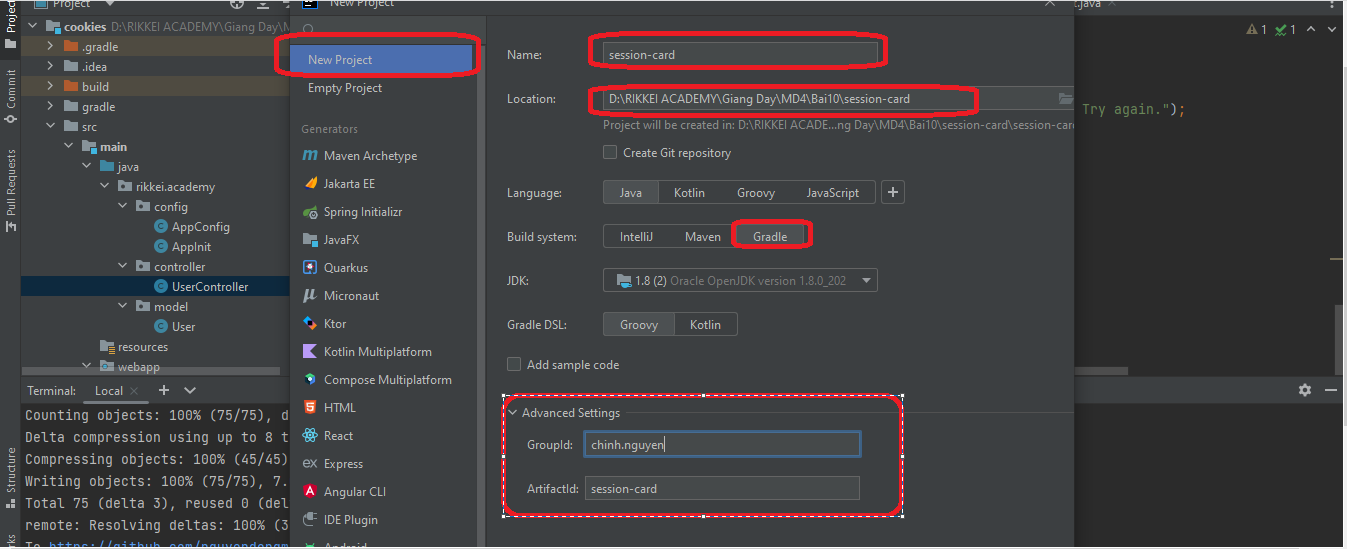
Tạo được giỏ hàng và thêm được sản phẩm vào trong giỏ hàng.

Mô tả

Trong phần này, chúng ta sẽ phát triển một tính năng cho phép người dùng có thể thêm một sản phẩm vào trong giỏ hàng của mình.

Hướng dẫn:

* Tạo project với gradle.



* Thêm thư viện vào **build.gradle**:

compileOnly('javax.servlet:javax.servlet-api:4.0.1')  
implementation group: 'org.springframework', name: 'spring-core', version: '5.3.2'  
implementation group: 'org.springframework', name: 'spring-context', version: '5.3.2'  
implementation group: 'org.springframework', name: 'spring-beans', version: '5.3.2'  
implementation group: 'org.springframework', name: 'spring-web', version: '5.3.2'  
implementation group: 'org.springframework', name: 'spring-webmvc', version: '5.3.2'  
implementation group: 'org.thymeleaf', name: 'thymeleaf-spring5', version: '3.0.11.RELEASE'  
implementation group: 'nz.net.ultraq.thymeleaf', name: 'thymeleaf-layout-dialect', version: '2.5.2'  
implementation group: 'org.hibernate', name: 'hibernate-core', version: '5.3.0.Final'  
implementation group: 'org.hibernate', name: 'hibernate-entitymanager', version: '5.3.0.Final'  
implementation group: 'org.springframework', name: 'spring-orm', version: '5.3.2'  
implementation group: 'mysql', name: 'mysql-connector-java', version: '8.0.22'  
implementation group: 'org.springframework.data', name: 'spring-data-jpa', version: '2.4.2'

* Tạo package: **chinh.nguyen.config** (từ thư mục java):
* Tạo class: **AppConfig** trong package: **config**

package chinh.nguyen.config;  
  
import org.springframework.beans.BeansException;  
import org.springframework.beans.factory.annotation.Qualifier;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.ApplicationContextAware;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.ComponentScan;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;  
import org.springframework.data.web.config.EnableSpringDataWebSupport;  
import org.springframework.jdbc.datasource.DriverManagerDataSource;  
import org.springframework.orm.jpa.JpaTransactionManager;  
import org.springframework.orm.jpa.JpaVendorAdapter;  
import org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean;  
import org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter;  
import org.springframework.transaction.PlatformTransactionManager;  
import org.springframework.transaction.annotation.EnableTransactionManagement;  
import org.springframework.web.servlet.config.annotation.EnableWebMvc;  
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;  
import org.thymeleaf.spring5.SpringTemplateEngine;  
import org.thymeleaf.spring5.templateresolver.SpringResourceTemplateResolver;  
import org.thymeleaf.spring5.view.ThymeleafViewResolver;  
import org.thymeleaf.templatemode.TemplateMode;  
  
import javax.persistence.EntityManager;  
import javax.persistence.EntityManagerFactory;  
import javax.sql.DataSource;  
import java.util.Properties;  
  
@Configuration  
@EnableWebMvc  
@EnableTransactionManagement  
@EnableSpringDataWebSupport  
public class AppConfig implements WebMvcConfigurer, ApplicationContextAware {  
 private ApplicationContext applicationContext;  
  
 @Override  
 public void setApplicationContext(ApplicationContext applicationContext) throws BeansException {  
 this.applicationContext = applicationContext;  
 }  
  
 //Cấu hình Thymleaf  
 @Bean  
 public SpringResourceTemplateResolver templateResolver() {  
 SpringResourceTemplateResolver templateResolver = new SpringResourceTemplateResolver();  
 templateResolver.setApplicationContext(applicationContext);  
 templateResolver.setPrefix("/WEB-INF/views");  
 templateResolver.setSuffix(".html");  
 templateResolver.setTemplateMode(TemplateMode.*HTML*);  
 templateResolver.setCharacterEncoding("UTF-8");  
 return templateResolver;  
 }  
  
 @Bean  
 public SpringTemplateEngine templateEngine() {  
 SpringTemplateEngine templateEngine = new SpringTemplateEngine();  
 templateEngine.setTemplateResolver(templateResolver());  
 return templateEngine;  
 }  
  
 @Bean  
 public ThymeleafViewResolver viewResolver() {  
 ThymeleafViewResolver viewResolver = new ThymeleafViewResolver();  
 viewResolver.setTemplateEngine(templateEngine());  
 viewResolver.setCharacterEncoding("UTF-8");  
 viewResolver.setContentType("UTF-8");  
 return viewResolver;  
 }  
  
 //Cấu hình JPA  
 @Bean  
 @Qualifier(value = "entityManager")  
 public EntityManager entityManager(EntityManagerFactory entityManagerFactory) {  
 return entityManagerFactory.createEntityManager();  
 }  
  
 @Bean  
 public LocalContainerEntityManagerFactoryBean entityManagerFactory() {  
 LocalContainerEntityManagerFactoryBean em = new LocalContainerEntityManagerFactoryBean();  
 em.setDataSource(dataSource());  
 em.setPackagesToScan("chinh.nguyen.model");  
 JpaVendorAdapter vendorAdapter = new HibernateJpaVendorAdapter();  
 em.setJpaVendorAdapter(vendorAdapter);  
 em.setJpaProperties(additionalProperties());  
 return em;  
 }  
  
 @Bean  
 public DataSource dataSource() {  
 DriverManagerDataSource dataSource = new DriverManagerDataSource();  
 dataSource.setDriverClassName("com.mysql.cj.jdbc.Driver");  
 dataSource.setUrl("jdbc:mysql://localhost:3306/session\_card");  
 dataSource.setUsername("root");  
 dataSource.setPassword("Minhtri29092014");  
 return dataSource;  
 }  
  
 @Bean  
 public PlatformTransactionManager transactionManager(EntityManagerFactory emf) {  
 JpaTransactionManager transactionManager = new JpaTransactionManager();  
 transactionManager.setEntityManagerFactory(emf);  
 return transactionManager;  
 }  
  
 public Properties additionalProperties() {  
 Properties properties = new Properties();  
 properties.setProperty("hibernate.hbm2ddl.auto", "update");  
 properties.setProperty("hibernate.dialect", "org.hibernate.dialect.MySQL5Dialect");  
 return properties;  
 }  
}

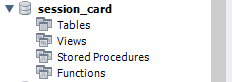
* Tạo class: **AppInit** trong package: **config**:

package chinh.nguyen.config;  
  
import org.springframework.web.filter.CharacterEncodingFilter;  
import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;  
  
import javax.servlet.Filter;  
  
public class AppInit extends AbstractAnnotationConfigDispatcherServletInitializer {  
 @Override  
 protected Class<?>[] getRootConfigClasses() {  
 return new Class[]{AppConfig.class};  
 }  
  
 @Override  
 protected Class<?>[] getServletConfigClasses() {  
 return new Class[0];  
 }  
  
 @Override  
 protected String[] getServletMappings() {  
 return new String[]{"/"};  
 }  
 @Override  
 protected Filter[] getServletFilters() {  
 CharacterEncodingFilter filter = new CharacterEncodingFilter();  
 filter.setForceEncoding(true);  
 filter.setEncoding("UTF-8");  
 return new Filter[]{filter};  
 }  
}

* Tạo package: **model** => Tạo class: **Product** trong package này

package chinh.nguyen.model;  
  
  
import javax.persistence.\*;  
  
@Entity  
@Table(name = "product")  
public class Product {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id;  
 private String name;  
 private double price;  
 private String description;  
  
 public Product() {  
 }  
  
 public Product(String name, double price, String description) {  
 this.name = name;  
 this.price = price;  
 this.description = description;  
 }  
  
 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;  
 }  
  
 public double getPrice() {  
 return price;  
 }  
  
 public void setPrice(double price) {  
 this.price = price;  
 }  
  
 public String getDescription() {  
 return description;  
 }  
  
 public void setDescription(String description) {  
 this.description = description;  
 }  
}

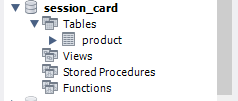
* Tạo database: **session\_card** trong MySQL Workbench



* Thêm id ‘war’ trong **build.gradle:**

plugins **{** id 'java'  
 id 'war'  
**}**

* Add Tomcat và chạy chương trình => Quan sát database



* Tạo class: **Cart** trong **model:**
* package chinh.nguyen.model;  
    
  import java.util.HashMap;  
  import java.util.Map;  
    
  public class Cart {  
   private Map<Product,Integer> products = new HashMap<>();  
    
   public Cart() {  
   }  
    
   public Cart(Map<Product,Integer> products) {  
   this.products = products;  
   }  
    
   public Map<Product,Integer> getProducts() {  
   return products;  
   }  
    
   private boolean checkItemInCart(Product product){  
   for (Map.Entry<Product, Integer> entry : products.entrySet()) {  
   if(entry.getKey().getId().equals(product.getId())){  
   return true;  
   }  
   }  
   return false;  
   }  
    
   private Map.Entry<Product, Integer> selectItemInCart(Product product){  
   for (Map.Entry<Product, Integer> entry : products.entrySet()) {  
   if(entry.getKey().getId().equals(product.getId())){  
   return entry;  
   }  
   }  
   return null;  
   }  
    
   public void addProduct(Product product){  
   if (!checkItemInCart(product)){  
   products.put(product,1);  
   } else {  
   Map.Entry<Product, Integer> itemEntry = selectItemInCart(product);  
   Integer newQuantity = itemEntry.getValue() + 1;  
   products.replace(itemEntry.getKey(),newQuantity);  
   }  
   }  
   public void removeProduct(Product product){  
   if (!checkItemInCart(product)){  
   products.put(product,1);  
   } else {  
   Map.Entry<Product, Integer> itemEntry = selectItemInCart(product);  
   Integer newQuantity = itemEntry.getValue() - 1;  
   if(itemEntry.getValue()==0){  
   newQuantity=0;  
   }  
   products.replace(itemEntry.getKey(),newQuantity);  
   }  
   }  
    
   public Integer countProductQuantity(){  
   Integer productQuantity = 0;  
   for (Map.Entry<Product, Integer> entry : products.entrySet()) {  
   productQuantity += entry.getValue();  
   }  
   return productQuantity;  
   }  
    
   public Integer countItemQuantity(){  
   return products.size();  
   }  
    
   public Float countTotalPayment(){  
   float payment = 0;  
   for (Map.Entry<Product, Integer> entry : products.entrySet()) {  
   payment += entry.getKey().getPrice() \* (float) entry.getValue();  
   }  
   return payment;  
   }  
    
  }

Phương thức checkIntemInCart() để kiểm tra xem sản phẩm đó đã có trong giỏ hàng hay chưa

Phương thức addProduct() được sử dụng để thêm sản phẩm vào trong giỏ hàng.

Phương thức removeProduct() được sử dụng để bớt sản phẩm trong giỏ hàng

Phương thức countProductQuantity() dùng để đếm số lượng sản phẩm đó hiện có trong giỏ hàng.

Phương thức countItemQuantity() để đếm số lượng sản phẩm có trong giỏ hàng.

Phương thức countTotalPayment() dùng để tính tổng số tiền cần phải thanh toán.

* Tạo package: **repository =>** Tạo interface: **IProductRepository** trong package này:

package chinh.nguyen.repository;  
  
import chinh.nguyen.model.Product;  
import org.springframework.data.repository.PagingAndSortingRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface IProductRepository extends PagingAndSortingRepository<Product, Long> {  
   
}

* Tạo package: **service:**
* Tạo interface **IProductService** trong package: **service**

package chinh.nguyen.service;  
  
import chinh.nguyen.model.Product;  
  
import java.util.Optional;  
  
public interface IProductService {  
 Iterable<Product> findAll();  
 Optional<Product> findById(Long id);  
  
}

* Tạo class: **ProductServiceIMPL** triển khai interface trên

package chinh.nguyen.service;  
  
import chinh.nguyen.model.Product;  
import chinh.nguyen.repository.IProductRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import java.util.Optional;  
@Service  
public class ProductServiceIMPL implements IProductService{  
 @Autowired  
 private IProductRepository productRepository;  
  
 @Override  
 public Iterable<Product> findAll() {  
 return productRepository.findAll();  
 }  
  
 @Override  
 public Optional<Product> findById(Long id) {  
 return productRepository.findById(id);  
 }  
  
}

* Tạo package: **controller (***chú ý***:** Nhớ scan luồng chạy xuống package này thông qua AppConfig, ngoài ra ta scan luồng chạy xuống cả tầng repository)

@EnableJpaRepositories("chinh.nguyen.repository")

@ComponentScan("chinh.nguyen.controller")  
public class AppConfig implements WebMvcConfigurer, ApplicationContextAware {

* Xây dựng **ProductController** để hiển thị danh sách sản phẩm đang có

package chinh.nguyen.controller;  
  
import chinh.nguyen.model.Cart;  
import chinh.nguyen.model.Product;  
import chinh.nguyen.service.IProductService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.\*;  
import org.springframework.web.servlet.ModelAndView;  
  
import java.util.Optional;  
  
@Controller  
@SessionAttributes("cart")  
public class ProductController {  
 @Autowired  
 private IProductService productService;  
  
 @ModelAttribute("cart")  
 public Cart setupCart() {  
 return new Cart();  
 }  
  
 @GetMapping("/shop")  
 public ModelAndView showShop() {  
 ModelAndView modelAndView = new ModelAndView("/shop");  
 modelAndView.addObject("products", productService.findAll());  
 return modelAndView;  
 }  
  
 @GetMapping("/add/{id}")  
 public String addToCart(@PathVariable Long id, @ModelAttribute Cart cart, @RequestParam("action") String action) {  
 Optional<Product> productOptional = productService.findById(id);  
 if (!productOptional.isPresent()) {  
 return "/error.404";  
 }  
 if (action.equals("increase")) {  
 cart.addProduct(productOptional.get());  
 return "redirect:/shopping-cart";  
 } else if (action.equals("decrease")) {  
 cart.removeProduct(productOptional.get());  
 return "redirect:/shopping-cart";  
 }  
  
 cart.addProduct(productOptional.get());  
 return "redirect:/shop";  
 }  
  
}

* Thêm đoạn code sau vào: AppConfig để có thể khởi tạo @Bean lớp service sang lớp controller

@Bean  
public IProductService productService(){  
 return new ProductServiceIMPL();  
}

Phương thương addToCart được sử dụng để thêm sản phẩm vào giỏ hàng trong đó action được sử dụng để chuyển hướng tới giao diện hiển thị tất cả các sản phẩm đang có trong giỏ hàng nếu action = “show” và ngược lại sẽ hiển thị tất cả sản phẩm của shop.

* Tạo class: **ShoppingCartController**

package chinh.nguyen.controller;  
  
import chinh.nguyen.model.Cart;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.ModelAttribute;  
import org.springframework.web.bind.annotation.SessionAttribute;  
import org.springframework.web.servlet.ModelAndView;  
@Controller  
public class ShoppingCartController {  
 @ModelAttribute("cart")  
 public Cart setupCart(){  
 return new Cart();  
 }  
  
 @GetMapping("/shopping-cart")  
 public ModelAndView showCart (@SessionAttribute("cart") Cart cart){  
 ModelAndView modelAndView = new ModelAndView("/cart");  
 modelAndView.addObject("cart",cart);  
 return modelAndView;  
 }  
  
}

* Tạo thư mục: **webapp/WEB-INF/views** (từ mục **main**)
* Xây dựng file: **shop.html**

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Shop</title>  
</head>  
<body>  
<p>  
 <a href="/shopping-cart">Your cart</a>  
</p>  
<table border="1">  
 <tr>  
 <td>Name</td>  
 <td>Price</td>  
 <td>Add To Cart</td>  
 </tr>  
 <th:block th:each="product:${products}">  
 <tr>  
 <td th:text="${product.name}"></td>  
 <td th:text="${product.price}"></td>  
 <td><a th:href="@{/add/\_\_${product.id}\_\_(action='list')}">Add To Cart</a></td>  
 </tr>  
 </th:block>  
</table>  
</body>  
</html>

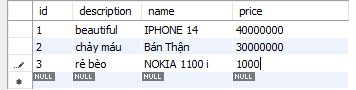
* Xây dựng **cart.html**

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Shopping Cart</title>  
</head>  
<body>  
<p>  
 <a href="/shop">Shop</a>  
</p>  
<h1>Your Cart</h1>  
<form th:action="@{/shop}" th:object="${cart}" method="post">  
 <h3>Total Items: <span th:text="${cart.countItemQuantity()}"></span></h3>  
 <h3>Total Products: <span th:text="${cart.countProductQuantity()}"></span></h3>  
 <table border="1">  
 <tr>  
 <th>Item</th>  
 <th>Quantity</th>  
 <th>Price</th>  
 </tr>  
 <tr th:each="product: ${cart.products}">  
 <td th:text="${product.getKey().name}"></td>  
 <td>  
 <a id="decrease\_button" th:href="@{/add/\_\_${product.getKey().id}\_\_(action='decrease')}" th:text="${'-'}"></a>  
 <span th:text="${product.getValue()}"></span>  
 <a id="increase\_button" th:href="@{/add/\_\_${product.getKey().id}\_\_(action='increase')}"  
 th:text="${'+'}"></a>  
 </td>  
 <td th:text="${product.getKey().price}"></td>  
 </tr>  
 <tr>  
 <td>Total</td>  
 <td></td>  
 <td th:text="${cart.countTotalPayment()}"></td>  
 </tr>  
 </table>  
</form>  
</body>  
</html>

* Xây dựng view **error.404.html**

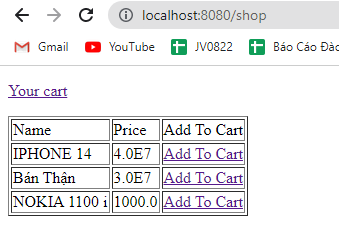
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Error</title>  
</head>  
<body>  
<h1>NOT FOUND</h1>  
</body>  
</html>

* Fix một vài trường dữ liệu trong bảng **product** dưới MySQL

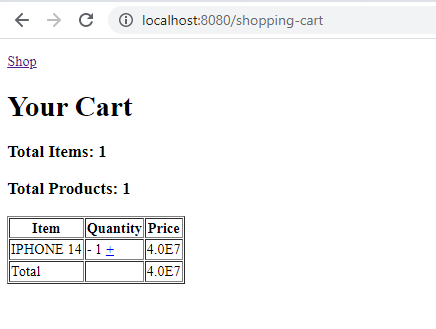


* Chạy ứng dụng, nhập vào thanh địa chỉ như sau: <http://localhost:8080/shop>

và click addToCart để thêm sản phẩm vào giỏ hàng



Sau đó truy cập vào: <http://localhost:8080/shopping-cart> để xem sản phẩm hiện có trong giỏ hàng



### **Có thể click vào + hoặc – để thêm bớt sản phẩm trong giỏ hàng**

https://github.com/nguyendongminhtri/MD4-Bai10-TH3-Shoping-Cart.git

### **Hướng dẫn nộp bài:**

Up code lên github.

Paste link github vào phần nộp bài.