

# Developing an E-commerce Website for Sporty Shoes.

## SOURCE CODE

**Package name:** com.simplilearn.controller

**Class:** AdminController.java

```
package com.simplilearn.controller;
```

```
import java.util.List;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpSession;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.ModelAttribute;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
import com.simplilearn.dao.OrderDAO;
```

```
import com.simplilearn.dao.ProductDAO;
```

```
import com.simplilearn.dao.UserDAO;
```

```
import com.simplilearn.entity.Order;
```

```
import com.simplilearn.entity.Product;
```

```
import com.simplilearn.entity.User;

@Controller
@RequestMapping("/admin")
public class AdminController {

    @Autowired
    private ProductDAO productDAO;

    // inject

    @Autowired
    private UserDAO userDAO;

    // inject

    @Autowired
    private OrderDAO orderDAO;

    @GetMapping("/products")
    public String products(Model theModel, HttpServletRequest request) {

        HttpSession session = request.getSession(true);
        User currentUser = (User) session.getAttribute("currentUser");

        // get all products from DAO
        List<Product> products = productDAO.getProducts();

        // add the products to the model
        theModel.addAttribute("currentUser", currentUser);

        // add the products to the model
```

```
        theModel.addAttribute("products", products);

        return "mange-products";
    }
}
```

```
@GetMapping("/deleteProduct")
public String deleteProduct(@RequestParam("productId") int productId, Model
theModel) {
```

```
    productDAO.deleteProduct(productId);

    return "redirect:/admin/products";
}
```

```
@GetMapping("/deleteUser")
public String deleteUser(@RequestParam("userId") int userId, Model theModel) {
```

```
    userDAO.deleteUser(userId);

    return "redirect:/admin/manageUsers";
}
```

```
@GetMapping("/deleteOrder")
public String deleteOrder(@RequestParam("orderId") int orderId, Model theModel)
{

    orderDAO.deleteOrder(orderId);

    return "redirect:/admin/manageOrders";
}
```

```
@GetMapping("/manageUsers")
public String manageUsers(Model theModel) {

    // get all users from DAO
    List<User> users = userDAO getUsers();

    // add the users to the model
    theModel.addAttribute("users", users);

    return "manageUsers";
}
```

```
@GetMapping("/manageOrders")
public String manageOrders(Model theModel) {

    // get all users from DAO
    List<Order> orders = orderDAO.getOrder();

    int totalPrice = 0;

    for (Order order : orders) {
        totalPrice = totalPrice + order.getProduct().getPrice();
    }

    // add the users to the model
    theModel.addAttribute("orders", orders);

    // add the users to the model
    theModel.addAttribute("totalPrice", totalPrice);
}
```

```
        return "manageOrders";
    }
}
```

```
@GetMapping("/addProdcut")
public String addProdcut() {
```

```
    return "addproduct";
}
```

```
@PostMapping("/addProductProcess")
public String addProductProcess(HttpServletRequest request,
    @RequestParam("name") String name,
    @RequestParam("company") String company,
    @RequestParam("size") int size,
    @RequestParam("price") int price,
    @RequestParam("image") String image
) {
```

```
    Product product = new Product(name, company, size, price, image);
    productDAO.saveProduct(product);
```

```
    return "redirect:/admin/products";
}
```

```
@GetMapping("/updateProduct")

public String updateProduct(@RequestParam("productId") int productId , Model
theModel) {
```

```
// get all products from DAO
Product product = productDAO.getProduct(productId);

// add the users to the model
theModel.addAttribute("product", product);

return "update-product";
}
```

```
@PostMapping("/updateProductProcess")
public String updateProductProcess(HttpServletRequest request,
    @RequestParam("productId") int productId,
    @RequestParam("name") String name,
    @RequestParam("company") String company,
    @RequestParam("size") int size,
    @RequestParam("price") int price,
    @RequestParam("image") String image
) {
```

```
Product product = productDAO.getProduct(productId);
product.setCompany(company);
product.setSize(size);
product.setName(name);
product.setPrice(price);
product.setImage_link(image);

productDAO.saveProduct(product);
```

```
        return "redirect:/admin/products";
    }

}
```

**Class:** HomePageController.java

```
package com.simplilearn.controller;

import java.util.ArrayList;
import java.util.List;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import com.simplilearn.dao.OrderDAO;
import com.simplilearn.dao.ProductDAO;
import com.simplilearn.entity.Order;
import com.simplilearn.entity.Product;
import com.simplilearn.entity.User;

@Controller
@RequestMapping("/homepage")
public class HomePageController {
```

```

// need to inject

@Autowired

private ProductDAO productDAO;

@Autowired

private OrderDAO orderDAO;


@GetMapping("/products")

public String products(Model theModel, HttpServletRequest request) {

    HttpSession session = request.getSession(true);

    User currentUser = (User) session.getAttribute("currentUser");


    // get all products from DAO

    List<Product> products = productDAO.getProducts();


    List<Order> userOrders = orderDAO.getUserOrders(currentUser.getId());

    List<Integer> userProducts = new ArrayList<Integer>();


    for (Order order : userOrders) {

        userProducts.add(order.getProduct().getId());

    }


    // add the products to the model

    theModel.addAttribute("products", products);

    theModel.addAttribute("userProducts", userProducts);

    theModel.addAttribute("currentUser", currentUser);


    return "user-home";
}

```



```
}
```

```
@GetMapping("/orderProcess")
```

```
public String orderProcess(Model theModel, @RequestParam("productId") int  
productId, HttpServletRequest request) {
```

```
    HttpSession session = request.getSession(true);
```

```
    User currentUser = (User) session.getAttribute("currentUser");
```

```
    Product product = productDAO.getProduct(productId);
```

```
    Order order = new Order(currentUser, product);
```

```
    orderDAO.saveOrder(order);
```

```
    theModel.addAttribute("currentUser", currentUser);
```

```
    return "redirect:/homepage/products";
```

```
}
```

```
@GetMapping("/mycart")
```

```
public String showMyCart(Model theModel, HttpServletRequest request) {
```

```
    HttpSession session = request.getSession(true);
```

```
    User currentUser = (User) session.getAttribute("currentUser");
```

```
    // get all products from DAO
```

```
    List<Order> userOrders = orderDAO.getUserOrders(currentUser.getId());
```

```
    List<Product> userProducts = new ArrayList<Product>();
```

```
    for (Order order : userOrders) {
```

```
        userProducts.add(order.getProduct());
```

```

    }

    int total_price = 0;

    for (Product product : userProducts) {
        total_price = total_price + product.getPrice();
    }

    theModel.addAttribute("userProducts", userProducts);
    theModel.addAttribute("total_price", total_price);
    theModel.addAttribute("currentUser", currentUser);

    return "mycart";
}

@PostMapping("/searchProducts")
public String searchProducts(HttpServletRequest request, Model theModel,
@RequestParam("keySearch") String key) {

    HttpSession session = request.getSession(true);
    User currentUser = (User) session.getAttribute("currentUser");

    System.out.println(key);
    // get all products from DAO
    List<Product> products = productDAO.searchProducts(key);

    if (currentUser.getType() == 0) {
        // get all products from DAO
        List<Order> userOrders =
orderDAO.getUserOrders(currentUser.getId());

```

```

        List<Integer> userProducts = new ArrayList<Integer>();

        for (Order order : userOrders) {
            userProducts.add(order.getProduct().getId());
        }

        theModel.addAttribute("userProducts", userProducts);
        theModel.addAttribute("currentUser", currentUser);
    }

    // add the products to the model
    theModel.addAttribute("products", products);

    if (currentUser.getType() == 0) {
        return "user-home";
    }

    else {
        return "mange-products";
    }

}

}

```

**Class:**LoginController.java

```
package com.simplilearn.controller;

import java.util.List;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import com.simplilearn.dao.ProductDAO;
import com.simplilearn.dao.UserDAO;
import com.simplilearn.entity.Product;
import com.simplilearn.entity.User;

@Controller
public class LoginController {

    // inject
    @Autowired
    private UserDAO userDAO;

    @RequestMapping("/login")
    public String login(Model theModel) {

        return "login";
    }
}
```

```

@RequestMapping("/register")

public String register(Model theModel) {

    return "register";

}

@PostMapping("/loginProcess")

public String loginProcess(HttpServletRequest request, @RequestParam("username")
String username,

    @RequestParam("password") String password) {

    HttpSession session = request.getSession(true);

    // get all users from DAO
    List<User> users = userDAO getUsers();

    boolean isusesr = false;
    User tempUser = null;

    for (User user : users) {

        if (user.getUsername().equals(username) &&
user.getPassword().equals(password)) {

            isusesr = true;
            tempUser = user;

            break;

        }

    }

}

```

```

        if (isusesr && tempUser.getType() == 0) {

            session.setAttribute("currentUser", tempUser);
            return "redirect:/homepage/products";
        }

        else if (isusesr && tempUser.getType() == 1) {

            session.setAttribute("currentUser", tempUser);
            return "redirect:/admin/products";
        }
        else
            return "login";
    }

    @PostMapping("/registerProcess")
    public String registerProcess(@RequestParam("username") String username,
                                  @RequestParam("age") String age,
                                  @RequestParam("password") String password) {

        User user = new User(username, password, Integer.parseInt(age));
        userDAO.saveUser(user);

        return "login";
    }
}

```

**Class:** UserController.java

```
package com.simplilearn.controller;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import com.simplilearn.dao.ProductDAO;
import com.simplilearn.dao.UserDAO;
import com.simplilearn.entity.Product;
import com.simplilearn.entity.User;
```

```
@Controller
```

```
@RequestMapping("/user")
```

```
public class UserController {
```

```
    @Autowired
```

```
    private ProductDAO productDAO;
```

```
    // inject
```

```
    @Autowired
```

```
    private UserDAO userDAO;
```

```
    @GetMapping("/myaccount")
```

```

public String myaccount(Model theModel, HttpServletRequest request) {

    HttpSession session = request.getSession(true);
    User currentUser = (User) session.getAttribute("currentUser");

    // add the products to the model
    theModel.addAttribute("currentUser", currentUser);

    return "myaccount";
}

@PostMapping("/updateAccount")
public String updateAccount(HttpServletRequest request,
    @RequestParam("username") String username,
    @RequestParam("age") String age,
    @RequestParam("password") String password , Model theModel
    ) {

    HttpSession session = request.getSession(true);
    User currentUser = (User) session.getAttribute("currentUser");
    System.out.println(currentUser.toString());

    currentUser.setUsername(username);
    currentUser.setAge(Integer.parseInt(age));
    currentUser.setPassword(password);

    theModel.addAttribute("currentUser", currentUser);
}

```



```
        userDao.saveUser(currentUser);

        return "myaccount";
    }

}
```

## **Package: com.simplilearm.dao**

**Class:** OrderDAO.java

```
package com.simplilearn.dao;

import java.util.List;

import com.simplilearn.entity.Order;
import com.simplilearn.entity.Product;
import com.simplilearn.entity.User;

public interface OrderDAO {

    public List<Order> getOrder();

    public Order getOrder(int orderId);

    public void saveOrder(Order order);

    public void deleteOrder(int orderId);

    public List<Order> getUserOrders(int userId);

}
```

**Class:** OrderDAOImpl.java

```
package com.simplilearn.dao;

import java.util.List;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.query.Query;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;
import org.springframework.transaction.annotation.Transactional;
import com.simplilearn.entity.Order;
import com.simplilearn.entity.User;
```

```
@Repository
```

```
public class OrderDAOImpl implements OrderDAO {
```

```
    // need to inject the session factory
```

```
    @Autowired
```

```
    private SessionFactory sessionFactory;
```

```
    @Override
```

```
    @Transactional
```

```
    public List<Order> getOrder() {
```

```
        // get the current hibernate session
```

```
        Session currentSession = sessionFactory.getCurrentSession();
```

```
        // create a query ... sort by last name
```

```
        Query<Order> theQuery = currentSession.createQuery("from Order",
Order.class);
```

```
        // execute query and get result list
```

```
        List<Order> orders = theQuery.getResultList();
```

```
        // return the results
```

```
        return orders;
```

```
    }
```

```
    @Override
```

@Transactional

```
public void saveOrder(Order order) {  
    // get current hibernate session  
    Session currentSession = sessionFactory.getCurrentSession();  
  
    // save/update the customer  
    currentSession.saveOrUpdate(order);  
}
```

@Override

@Transactional

```
public Order getOrder(int orderId) {  
    // get the current hibernate session  
    Session currentSession = sessionFactory.getCurrentSession();  
  
    // now retrieve/read from database using the primary key  
    Order order = currentSession.get(Order.class, orderId);  
  
    return order;  
}
```

@Override

@Transactional

```
public void deleteOrder(int orderId) {  
    // get the current hibernate session  
    Session currentSession = sessionFactory.getCurrentSession();  
  
    // delete object with primary key  
    Query theQuery = currentSession.createQuery("delete from Order where  
id=:orderId");  
    theQuery.setParameter("orderId", orderId);
```

```

        theQuery.executeUpdate();

    }

    @Override
    @Transactional
    public List<Order> getUserOrders(int userId) {
        // get the current hibernate session
        Session currentSession = sessionFactory.getCurrentSession();

        // create a query ... sort by last name
        Query<Order> theQuery = currentSession.createQuery("from Order where
user_id=: userId", Order.class);

        theQuery.setParameter("userId", userId);

        // execute query and get result list
        List<Order> orders = theQuery.getResultList();

        // return the results
        return orders;
    }
}

```

**Class:** ProductDAO.java

```

package com.simplilearn.dao;

import java.util.List;
import com.simplilearn.entity.Order;
import com.simplilearn.entity.Product;

```

```
public interface ProductDAO {  
    public List<Product> getProducts();  
  
    public Product getProduct(int productId);  
    public void saveProduct(Product product);  
    public void deleteProduct(int productId);  
    public List<Product> searchProducts(String key);  
  
}
```

**Class:** ProductDAOImpl.java

```
package com.simplilearn.dao;  
import java.util.List;  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.query.Query;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Repository;  
import org.springframework.transaction.annotation.Transactional;  
import com.simplilearn.entity.Order;  
import com.simplilearn.entity.Product;
```

@Repository

```
public class ProductDAOImpl implements ProductDAO {
```

```
    // need to inject the session factory
```

```
    @Autowired
```

```

private SessionFactory sessionFactory;

@Override
@Transactional
public List<Product> getProducts() {
    // get the current hibernate session
    Session currentSession = sessionFactory.getCurrentSession();

    // create a query
    Query<Product> theQuery = currentSession.createQuery("from Product",
Product.class);

    // execute query and get result list
    List<Product> products = theQuery.getResultList();

    // return the results
    return products;
}

@Override
@Transactional
public Product getProduct(int productId) {
    // get the current hibernate session
    Session currentSession = sessionFactory.getCurrentSession();

    // now retrieve/read from database using the primary key
    Product prod = currentSession.get(Product.class, productId);

    return prod;
}

```

@Override

@Transactional

```
public void saveProduct(Product product) {  
    // get current hibernate session  
    Session currentSession = sessionFactory.getCurrentSession();  
  
    // save/update  
    currentSession.saveOrUpdate(product);  
  
}
```

@Override

@Transactional

```
public void deleteProduct(int productId) {  
    // get the current hibernate session  
    Session currentSession = sessionFactory.getCurrentSession();  
  
    // delete object with primary key  
    Query theQuery = currentSession.createQuery("delete from  
Product where id=:productId");  
    theQuery.setParameter("productId", productId);  
  
    theQuery.executeUpdate();  
  
}
```

@Override

@Transactional

```
public List<Product> searchProducts(String key) {  
    // get the current hibernate session  
    Session currentSession = sessionFactory.getCurrentSession();
```

```

        // create a query
        Query<Product> theQuery = currentSession.createQuery("from
Product where name like: key", Product.class);

        theQuery.setParameter("key", "%" + key + "%");

        // execute query and get result list
        List<Product> products = theQuery.getResultList();

        System.out.println();

        // return the results
        return products;
    }
}

```

```

}

```

**Class:** UserDao.java

```

package com.simplilearn.dao;

import java.util.List;
import com.simplilearn.entity.User;

public interface UserDao {

    public List<User> getUsers();
    public User getUser(int user);
    public void saveUser(User user);
    public void deleteUser(int user);

}

```

**Class:** UserDaoImpl.java

```

package com.simplilearn.dao;

import java.util.List;

```



```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.query.Query;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;
import org.springframework.transaction.annotation.Transactional;
import com.simplilearn.entity.User;
```

```
@Repository
```

```
public class UserDAOImpl implements UserDAO {
```

```
    // need to inject the session factory
```

```
    @Autowired
```

```
    private SessionFactory sessionFactory;
```

```
    @Override
```

```
    @Transactional
```

```
    public List<User> getUsers() {
```

```
        // get the current hibernate session
```

```
        Session currentSession = sessionFactory.getCurrentSession();
```

```
        // create a query ... sort by last name
```

```
        Query<User> theQuery = currentSession.createQuery("from User",
User.class);
```

```
        // execute query and get result list
```

```
        List<User> users = theQuery.getResultList();
```

```
        // return the results
```

```
        return users;
    }
}
```

```
@Override
```

```
@Transactional
```

```
public User getUser(int userId) {
    // get the current hibernate session
    Session currentSession = sessionFactory.getCurrentSession();

    // now retrieve/read from database using the primary key
    User user = currentSession.get(User.class, userId);

    return user;
}
```

```
@Override
```

```
@Transactional
```

```
public void saveUser(User user) {

    // get current hibernate session
    Session currentSession = sessionFactory.getCurrentSession();

    // save/update the customer
    currentSession.saveOrUpdate(user);

}
```

```
@Override
```

```
@Transactional
```

```
public void deleteUser(int userId) {
```

```

        // get the current hibernate session
        Session currentSession = sessionFactory.getCurrentSession();

        // delete object with primary key
        Query theQuery =
            currentSession.createQuery("delete from User where
id=:userId");

        theQuery.setParameter("userId", userId);

        theQuery.executeUpdate();

    }

}

```

**Package: com.simplilearn.entity**

**Class: Order.java**

```

package com.simplilearn.entity;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

@Entity
@Table(name="orders")

public class Order {

```

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

@Column(name="id")

private int id;

@ManyToOne

@JoinColumn(name="user\_id")

private User user;

@ManyToOne

@JoinColumn(name="product\_id")

private Product product;

public Order() {

}

public Order(User user, Product product) {

    super();

    this.user = user;

    this.product = product;

}

public int getId() {

    return id;

}

public void setId(int id) {

    this.id = id;

}

```

    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    public Product getProduct() {
        return product;
    }

    public void setProduct(Product product) {
        this.product = product;
    }

    @Override
    public String toString() {
        return "Order [id=" + id + ", user=" + user + ", product=" + product + "]";
    }
}

```

**Class:** Product.java

```

package com.simplilearn.entity;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

```

```
import javax.persistence.Table;
```

```
@Entity
```

```
@Table(name="products")
```

```
public class Product {
```

```
    @Id
```

```
    @GeneratedValue(strategy=GenerationType.IDENTITY)
```

```
    @Column(name="id")
```

```
    private int id;
```

```
    @Column(name="name")
```

```
    private String name;
```

```
    @Column(name="company")
```

```
    private String company;
```

```
    @Column(name="size")
```

```
    private int size;
```

```
    @Column(name="price")
```

```
    private int price;
```

```
    @Column(name="image_link")
```

```
    private String image_link;
```

```
    public Product() {
```

```
}
```

```
public Product(String name, String company, int size, int price, String image_link) {  
    super();  
    this.name = name;  
    this.company = company;  
    this.size = size;  
    this.price = price;  
    this.image_link = image_link;  
}
```

```
public int getId() {  
    return id;  
}
```

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

```
public String getName() {  
    return name;  
}
```

```
public void setName(String name) {
```

```
        this.name = name;
    }
```

```
    public String getCompany() {
        return company;
    }
```

```
    public void setCompany(String company) {
        this.company = company;
    }
```

```
    public int getSize() {
        return size;
    }
```

```
    public void setSize(int size) {
        this.size = size;
    }
```

```
    public int getPrice() {
        return price;
    }
```

```
    public void setPrice(int price) {
```



```
        this.price = price;
    }
```

```
    public String getImage_link() {
        return image_link;
    }
```

```
    public void setImage_link(String image_link) {
        this.image_link = image_link;
    }
```

```
    @Override
    public String toString() {
        return "Product [id=" + id + ", name=" + name + ", company=" + company +
            ", size=" + size + ", price=" + price
                + ", image_link=" + image_link + "]";
    }
```

```
}
```

**Class:**User.java

```
package com.simplilearn.entity;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
import javax.persistence.Table;
import javax.validation.constraints.Max;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Pattern;
import org.hibernate.validator.constraints.Range;
```

```
@Entity
```

```
@Table(name="users")
```

```
public class User {
```

```
    @Id
```

```
    @GeneratedValue(strategy=GenerationType.IDENTITY)
```

```
    @Column(name="id")
```

```
    private int id;
```

```
    @Column(name="type")
```

```
    private int type;
```

```
    @Column(name="username")
```

```
    private String username;
```

```
    @Column(name="password")
```

```
    private String password;
```

```
    @Range(min=1, max=120,message = "Invalid Age")
```

```
    @Column(name="age")
```

```
    private int age;
```

```
public User() {
```

```
}
```

```
public User(String username, String password, int age) {
```

```
    super();
```

```
    this.username = username;
```

```
    this.password = password;
```

```
    this.age = age;
```

```
}
```

```
public int getId() {
```

```
    return id;
```

```
}
```

```
public void setId(int id) {
```

```
    this.id = id;
```

```
}
```

```
public int getType() {
```

```
    return type;
```

```
}
```

```
public void setType(int type) {
```

```
    this.type = type;
```

```
}
```

```
public String getUsername() {
```

```
    return username;
```

```
}
```

```
public void setUsername(String username) {  
    this.username = username;  
}
```

```
public String getPassword() {  
    return password;  
}
```

```
public void setPassword(String password) {  
    this.password = password;  
}
```

```
public int getAge() {  
    return age;  
}
```

```
public void setAge(int age) {  
    this.age = age;  
}
```

```
@Override
```

```
public String toString() {  
    return "User [id=" + id + ", type=" + type + ", username=" + username + ",  
password=" + password + ", age=" + age + "];"  
}
```

}

**Src/main/webapp:**

**Src/main/webapp/META-INF/MANIFEST-MF:**

Manifest-Version: 1.0

Class-Path:

**Src/main/webapp/WEB-INF/view:**

[src/main/webapp/WEB-INF/view/addproduct.jsp](#):

<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>List of Classes</title>

<link type="text/css" rel="stylesheet" href="../resources/css/style.css">

<link type="text/css" rel="stylesheet" href="../resources/css/login.css">

<link rel="stylesheet"

href="../resources/bootstrap/css/bootstrap.min.css">

<script src="../resources/bootstrap/js/bootstrap.min.js"></script>

<link rel="stylesheet"

href="https://fonts.googleapis.com/css?family=Nunito+Sans">

<link rel="stylesheet" href="../resources/fonts/font-awesome.min.css">

<link rel="stylesheet" href="../resources/fonts/ionicons.min.css">

<link rel="stylesheet" href="../resources/fonts/font-awesome.min.css">

<link rel="stylesheet" href="../resources/fonts/ionicons.min.css">

<link rel="stylesheet" href="../resources/fonts/material-icons.min.css">

```
</head>

<body
style="background-image: url('../resources/images/background.jpg');">

<div class="container-fluid h-100">
<div class="row h-100 ">

<jsp:include page="left-list.jsp" />


<div class="col-lg-10 float-left">

<br>

<center>

<h3>Add a New Product</h3>

</center>

<br>

<form action="addProductProcess" method="POST">
<div class="container">
<label>Product Name : </label> <br /> <input " type="text"
placeholder="Enter name" name="name" required>
<br>

<label>Company : </label> <br /> <input " type="text"
placeholder="Enter company" name="company" required>
<br>

<label>Size : </label> <br /> <input " type="text"
placeholder="Enter size" name="size" required>
<br>
```

```
<label>Price : </label> <br /> <input " type="text"
placeholder="Enter price" name="price" required>
<br>
```

```
<label>Image Link : </label> <br /> <input " type="text"
placeholder="Enter image" name="image" required>
<br>
<br>
<button type="submit">Add</button>
<br>
```

```
</div>
</form>
<br><br>
</div>
</div>
</div>
</body>
</html>
```

[src/main/webapp/WEB-INF/view/left-list.jsp](http://main/webapp/WEB-INF/view/left-list.jsp):

```
<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

```
<div class="sidenav col-lg-2 d-flex flex-column h-100">
```

```
<h4 id="logo">
```

```
Sparty<br />Shoes Store
```

</h4>

<c:choose>

<c:when test="\${currentUser.type == 0 }">

<nav class="nav flex-column">

<a class="nav-link"

href="\${pageContext.request.contextPath}/homepage/products"><i

class="fa fa-home fa-lg"></i> Home</a> <a class="nav-link active"

href="\${pageContext.request.contextPath}/homepage/mycart"> <i

class="fa fa-shopping-cart fa-lg"></i> My Cart

</a> <a class="nav-link"

href="\${pageContext.request.contextPath}/user/myaccount"><i

class="fa fa-user fa-lg"></i> My Account</a> <a class="nav-link"

href="\${pageContext.request.contextPath}/login"><i

class="fa fa-sign-out fa-lg"></i> Log Out</a>

</nav>

</c:when>

<c:otherwise>

<nav class="nav flex-column">

<a class="nav-link active"

href="\${pageContext.request.contextPath}/admin/products"> <i

class="fa fa-shopping-cart fa-lg"></i> <h style="font-size: 22px;">Manage Products</h>

</a>



```

<a class="nav-link"
href="${pageContext.request.contextPath}/admin/manageUsers"><i
class="fa fa-user fa-lg"></i> Manage Users</a>

<a class="nav-link"
href="${pageContext.request.contextPath}/admin/manageOrders"><i
class="fa fa-shopping-bag fa-lg"></i> Manage Orders</a>

<a class="nav-link"
href="${pageContext.request.contextPath}/admin/addProdcut"><i
class="fa fa-plus fa-lg"></i> Add Product</a>

<a class="nav-link"
href="${pageContext.request.contextPath}/login"><i
class="fa fa-sign-out fa-lg"></i> Log Out</a>

</nav>

</c:otherwise>

</c:choose>

</div>

```

**src/main/webapp/WEB-INF/view/login.jsp**

```

<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

```

<title>Login</title>

<link type="text/css" rel="stylesheet" href="resources/css/login.css">

</head>

<body style="">

<center> <h1> Login </h1> </center>

<form action="loginProcess" method="POST">

<div class="container">

<input type="hidden" name="command" value="LOGIN" />

<label>Username : </label>

<br/>

<input type="text" placeholder="Enter Username" name="username" required>

<br/>

<label>Password : </label>

<br/>

<input type="password" placeholder="Enter Password" name="password" required>

<br/>

<button type="submit">Login</button>

<br/>

<a href="{pageContext.request.contextPath}/register">Don't have an account?  
Register now!</a>

</div>

</form>

</body>

</html>

**src/main/webapp/WEB-INF/view/manageOrders.jsp:**

<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>List of Classes</title>

<link type="text/css" rel="stylesheet" href="../resources/css/style.css">

<link type="text/css" rel="stylesheet" href="../resources/css/login.css">

<link rel="stylesheet"

href="../resources/bootstrap/css/bootstrap.min.css">

<script src="../resources/bootstrap/js/bootstrap.min.js"></script>

<link rel="stylesheet"

href="https://fonts.googleapis.com/css?family=Nunito+Sans">

<link rel="stylesheet" href="../resources/fonts/font-awesome.min.css">

<link rel="stylesheet" href="../resources/fonts/ionicons.min.css">

<link rel="stylesheet" href="../resources/fonts/font-awesome.min.css">

<link rel="stylesheet" href="../resources/fonts/ionicons.min.css">

<link rel="stylesheet" href="../resources/fonts/material-icons.min.css">

</head>

```
<body
style="background-image: url('../resources/images/background.jpg');">

<div class="container-fluid h-100">

<div class="row h-100 ">


<jsp:include page="left-list.jsp" />


<div class="col-lg-10 float-left">


<br>

<center>

<h3>Manage Orders</h3>


</center>

<br>

<div id="container">

<center>

<div id="content">


<table style="width: 65% ">
```

<tr>

<th>User Name</th>

<th>Product Name</th>

<th>Price</th>

<th>Action</th>

</tr>

<c:forEach var="order" items="\${ orders }">

<!-- construct an "delete" link with user id -->

<c:url var="deleteLink" value="/admin/deleteOrder">

<c:param name="orderId" value="\${ order.id} " />

</c:url>

<tr>

<td>\${ order.user.username }</td>

<td>\${ order.product.name }</td>

<td>\${ order.product.price }\$</td>

<td><a href="{deleteLink}" class="btn btn-danger">

Delete</a></td>

</tr>

</c:forEach>

<tr>

<td colspan="4">Total Price = <strong>\${totalPrice}\$</strong></td>

</tr>

</table>

</div>

<br>

</center>

</div>

</div>

</div>

</div>

</body>

</html>

[src/main/webapp/WEB-INF/view/manageUsers.jsp](#):

<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>List of Classes</title>

<link type="text/css" rel="stylesheet" href="../resources/css/style.css">

<link type="text/css" rel="stylesheet" href="../resources/css/login.css">

<link rel="stylesheet"

href="../resources/bootstrap/css/bootstrap.min.css">

<script src="../resources/bootstrap/js/bootstrap.min.js"></script>

<link rel="stylesheet"

href="https://fonts.googleapis.com/css?family=Nunito+Sans">

<link rel="stylesheet" href="../resources/fonts/font-awesome.min.css">

<link rel="stylesheet" href="../resources/fonts/ionicons.min.css">

<link rel="stylesheet" href="../resources/fonts/font-awesome.min.css">

<link rel="stylesheet" href="../resources/fonts/ionicons.min.css">

<link rel="stylesheet" href="../resources/fonts/material-icons.min.css">

</head>

<body

style="background-image: url('../resources/images/background.jpg');">

<div class="container-fluid h-100">

<div class="row h-100 ">

<jsp:include page="left-list.jsp" />

<div class="col-lg-10 float-left">

<br>

<center>

<h3>Manage Users</h3>

</center>

<br>

<div id="container">

<center>

<div id="content">

<table style="width: 45%">

<tr>



```
<th>User Name</th>
```

```
<th>Age</th>
```

```
<th>Action</th>
```

```
</tr>
```

```
<c:forEach var="user" items="${users}">
```

```
<c:if test="${user.type == 0}">
```

```
<!-- construct an "delete" link with user id -->
```

```
<c:url var="deleteLink" value="/admin/deleteUser">
```

```
<c:param name="userId" value="${user.id}" />
```

```
</c:url>
```

```
<tr>
```

```
<td>${user.username}</td>
```

```
<td>${user.age}</td>
```

```
<td><a href="${deleteLink}" class="btn btn-danger">
```

```
Delete</a></td>
```

</tr>

</c:if>

</c:forEach>

</table>

</div>

<br>

</center>

</div>

</div>

</div>

</div>

</body>

</html>

**[src/main/webapp/WEB-INF/view/mange-products.jsp](#):**

<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>

<html>

```
<head>

<meta charset="ISO-8859-1">

<title>List of Classes</title>

<link type="text/css" rel="stylesheet" href="../resources/css/style.css">


<link rel="stylesheet"

href="../resources/bootstrap/css/bootstrap.min.css">

<script src="../resources/bootstrap/js/bootstrap.min.js"></script>

<link rel="stylesheet"

href="https://fonts.googleapis.com/css?family=Nunito+Sans">

<link rel="stylesheet" href="../resources/fonts/font-awesome.min.css">

<link rel="stylesheet" href="../resources/fonts/ionicons.min.css">

<link rel="stylesheet" href="../resources/fonts/font-awesome.min.css">

<link rel="stylesheet" href="../resources/fonts/ionicons.min.css">

<link rel="stylesheet" href="../resources/fonts/material-icons.min.css">


</head>

<body

style="background-image: url('../resources/images/background.jpg');">

<div class="container-fluid h-100">

<div class="row  h-100 ">


<jsp:include page="left-list.jsp" />
```

```
<div style="margin-top: 7px;" class="col-lg-10 float-left">
```

```
<form action="../../homepage/searchProducts" method="POST">
```

```
<div class="input-group md-form form-sm form-2 pl-0 ">
```

```
<input name="keySearch" class="form-control my-0 py-1 red-border"
```

```
type="text" placeholder="Search Product" aria-label="Search">
```

```
<button style="border-width: 0" type="submit">
```

```
<span class="input-group-text red lighten-3" id="basic-text1"><i
```

```
class="fa fa-search text-grey" aria-hidden="true"></i></span>
```

```
</button>
```

```
</div>
```

```
</form>
```

```
<div id="cards_container" class="container">
```

```
<div class="row">
```

```
<!-- loop over and print our customers -->
```

```
<c:forEach var="product" items="${products}">
```

```
<c:url var="updateLink" value="/admin/updateProduct">
```

```
<c:param name="productId" value="${product.id}" />
```

```
</c:url>
```

```
<c:url var="deleteLink" value="/admin/deleteProduct">
```

```
<c:param name="productId" value="${product.id}" />
```

```
</c:url>
```

```
<div class="card col-xs-3"
```

```
style="width: 14rem; margin: 21px 38px;">
```

```

```

```
<div style="margin-left: auto; margin-right: auto;"
```

```
class="card-body">
```

```
<h5 class="card-title">${product.name}</h5>
```

```
<p class="card-text">
```

```
${product.company}<br>${product.size} inches<br>${product.price}$
```

```
</p>
```

```
<div class="container">

<div class="row">

<div class="col">

<a href="{updateLink}" class="btn btn-success"> Update</a>

</div>

<div class="col">

<a href="{deleteLink}" class="btn btn-danger">Remove</a>

</div>

</div>

</div>

</div>

</c:forEach>

</div>

</div>

</div>

</body>

</html>
```

**src/main/webapp/WEB-INF/view/myaccount.jsp:**

```
<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="ISO-8859-1">
```

```
<title>List of Classes</title>
```

```
<link type="text/css" rel="stylesheet" href="../resources/css/style.css">
```

```
<link type="text/css" rel="stylesheet" href="../resources/css/login.css">
```

```
<link rel="stylesheet"
```

```
    href="../resources/bootstrap/css/bootstrap.min.css">
```

```
<script src="../resources/bootstrap/js/bootstrap.min.js"></script>
```

```
<link rel="stylesheet"
```

```
    href="https://fonts.googleapis.com/css?family=Nunito+Sans">
```

```
<link rel="stylesheet" href="../resources/fonts/font-awesome.min.css">
```

```
<link rel="stylesheet" href="../resources/fonts/ionicons.min.css">
```

```
<link rel="stylesheet" href="../resources/fonts/font-awesome.min.css">
```

```
<link rel="stylesheet" href="../resources/fonts/ionicons.min.css">
```

```
<link rel="stylesheet" href="../resources/fonts/material-icons.min.css">
```

```
</head>
```

```
<body
```

```
    style="background-image: url('../resources/images/background.jpg');">
```

```
    <div class="container-fluid h-100">
```

```
<div class="row h-100 ">
```

```
<jsp:include page="left-list.jsp" />
```

```
<div class="col-lg-10 float-left">
```

```
<br>
```

```
<center>
```

```
<h3>My Account</h3>
```

```
</center>
```

```
<form action="updateAccount" method="POST">
```

```
<div class="container">
```

```
<label>Username
```

```
: </label> <br /> <input
```

```
value="${currentUser.username}" type="text" placeholder="Enter Username"
```

```
name="username" required> <br />
```

```
<label>Age : </label> <br />
```

```
<input value="${currentUser.age}" type="text"
```

```
placeholder="Enter Age" name="age" required>
```

```
<br /> <label>Password : </label> <br /> <input
```

```
value="${currentUser.password}" type="text"
```

```
placeholder="Enter Password"
```

```
name="password" required> <br />
```

```
<br>
```

```
<button type="submit">Update</button>
```

```
<br> <br />
```



</div>

</form>

</div>

</div>

</div>

</body>

</html>

[src/main/webapp/WEB-INF/view/mycart.jsp](#):

<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>List of Classes</title>

<link type="text/css" rel="stylesheet" href="../resources/css/style.css">

<link type="text/css" rel="stylesheet" href="../resources/css/login.css">

<link rel="stylesheet"

href="../resources/bootstrap/css/bootstrap.min.css">

<script src="../resources/bootstrap/js/bootstrap.min.js"></script>

<link rel="stylesheet"

href="https://fonts.googleapis.com/css?family=Nunito+Sans">

<link rel="stylesheet" href="../resources/fonts/font-awesome.min.css">

<link rel="stylesheet" href="../resources/fonts/ionicons.min.css">

<link rel="stylesheet" href="../resources/fonts/font-awesome.min.css">

<link rel="stylesheet" href="../resources/fonts/ionicons.min.css">

<link rel="stylesheet" href="../resources/fonts/material-icons.min.css">

</head>

<body

style="background-image: url('../resources/images/background.jpg');">

<div class="container-fluid h-100">

<div class="row h-100 ">

<jsp:include page="left-list.jsp" />

<div class="col-lg-10 float-left">

<br>

<center>

<h3>My Cart</h3>

</center>

<br>

<div id="container">

<center>

<div id="content">

<table>

<tr>

<th>Name</th>

<th>Company</th>

<th>Size</th>

<th>Price</th>

</tr>

<c:forEach var="userProducts" items="\${userProducts}">

<tr>

<td>\${userProducts.name}</td>

<td>\${userProducts.company}</td>

<td>\${userProducts.size} inches</td>

<td>\${userProducts.price}\$</td>

</tr>

</c:forEach>

<tr>

<td colspan="4">Total Price = <strong>\${total\_price}</strong></td>

</tr>

</table>

</div>

<br>

<button style="width: 10%" class="btn btn-success" type="button">Purchase</button>

</center>

</div>

</div>

</div>

</div>

</body>

</html>

**[src/main/webapp/WEB-INF/view/register.jsp](#):**

<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="ISO-8859-1">
```

```
<title>Login</title>
```

```
<link type="text/css" rel="stylesheet" href="resources/css/login.css">
```

```
</head>
```

```
<body style="">
```

```
<center> <h1> Register </h1> </center>
```

```
<form action="registerProcess" method="POST">
```

```
<div class="container">
```

```
<input type="hidden" name="command" value="LOGIN" />
```

```
<label>Username : </label>
```

```
<br/>
```

```
<input type="text" placeholder="Enter Username" name="username" required>
```

```
<br/>
```

```
<label>Age : </label>
```

```
<br/>
```

```
<input type="text" placeholder="Enter Age" name="age" required>
```

```
<br/>
```

```
<label>Password : </label>
```

```
<br/>
```

<input type="password" placeholder="Enter Password" name="password" required>

<br/>

<button type="submit">Register</button>

<br/>

</div>

</form>

</body>

</html>

**src/main/webapp/WEB-INF/view/update-product.jsp:**

<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>List of Classes</title>

<link type="text/css" rel="stylesheet" href="../resources/css/style.css">

<link type="text/css" rel="stylesheet" href="../resources/css/login.css">

<link rel="stylesheet"

href="../resources/bootstrap/css/bootstrap.min.css">

<script src="../resources/bootstrap/js/bootstrap.min.js"></script>

```
<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Nunito+Sans">
<link rel="stylesheet" href="../resources/fonts/font-awesome.min.css">
<link rel="stylesheet" href="../resources/fonts/ionicons.min.css">
<link rel="stylesheet" href="../resources/fonts/font-awesome.min.css">
<link rel="stylesheet" href="../resources/fonts/ionicons.min.css">
<link rel="stylesheet" href="../resources/fonts/material-icons.min.css">
```

```
</head>
```

```
<body
```

```
style="background-image: url('../resources/images/background.jpg');">
```

```
<div class="container-fluid h-100">
```

```
<div class="row h-100 ">
```

```
<jsp:include page="left-list.jsp" />
```

```
<div class="col-lg-10 float-left">
```

```
<br>
```

```
<center>
```

```
<h3>Add a New Product</h3>
```

```
</center>
```

```
<br>
```

```
<form action="updateProductProcess" method="POST">
```

```
<div class="container">
```

```
<input type="hidden"
```

```
name="productId" value="{product.id}">
```

```
<label>Product Name : </label> <br /> <input type="text"
```

```
placeholder="Enter name" name="name" value="{product.name}" required>
```

```
<br>
```

```
<label>Company : </label> <br /> <input type="text"
```

```
placeholder="Enter company" name="company" value="{product.company}" required>
```

```
<br>
```

```
<label>Size : </label> <br /> <input type="text"
```

```
placeholder="Enter size" name="size" value="{product.size}" required>
```

```
<br>
```

```
<label>Price : </label> <br /> <input type="text"
```

```
placeholder="Enter price" name="price" value="{product.size}" required>
```

```
<br>
```

```
<label>Image Link : </label> <br /> <input type="text"
```

```
placeholder="Enter image" name="image" value="{product.image_link}" required>
```



<br>

<br>

<button type="submit">Update</button>

<br>

</div>

</form>

<br><br>

</div>

</div>

</div>

</body>

</html>

**src/main/webapp/WEB-INF/view/user-home.jsp:**

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>List of Classes</title>

<link type="text/css" rel="stylesheet" href="../resources/css/style.css">

<link rel="stylesheet"

href="../resources/bootstrap/css/bootstrap.min.css">

```
<script src="../../resources/bootstrap/js/bootstrap.min.js"></script>
```

```
<link rel="stylesheet"
```

```
href="https://fonts.googleapis.com/css?family=Nunito+Sans">
```

```
<link rel="stylesheet" href="../../resources/fonts/font-awesome.min.css">
```

```
<link rel="stylesheet" href="../../resources/fonts/ionicons.min.css">
```

```
<link rel="stylesheet" href="../../resources/fonts/font-awesome.min.css">
```

```
<link rel="stylesheet" href="../../resources/fonts/ionicons.min.css">
```

```
<link rel="stylesheet" href="../../resources/fonts/material-icons.min.css">
```

```
</head>
```

```
<body
```

```
style="background-image: url('../../resources/images/background.jpg');">
```

```
<div class="container-fluid h-100">
```

```
<div class="row h-100 ">
```

```
<jsp:include page="left-list.jsp" />
```

```
<div style="margin-top: 7px;" class="col-lg-10 float-left">
```

```
<form action="searchProducts" method="POST">
```

```
<div class="input-group md-form form-sm form-2 pl-0 ">
```

```
<input name="keySearch" class="form-control my-0 py-1 red-border" type="text"
placeholder="Search Product" aria-label="Search">
```

```
<button style="border-width: 0" type="submit">
<span class="input-group-text red lighten-3" id="basic-text1"><i
class="fa fa-search text-grey" aria-hidden="true"></i></span>
</button>
```

```
</div>
</form>
```

```
<div id="cards_container" class="container">
<div class="row">
```

```
<!-- loop over and print our customers -->
<c:forEach var="product" items="{products}">
```

```
<c:url var="orderLink" value="/homepage/orderProcess">
<c:param name="productId" value="{product.id}" />
</c:url>
```

```
<div class="card col-xs-3"
```

```
style="width: 14rem; margin: 21px 38px;">

<img class="card-img-top" src=${product.image_link}>

<div style="margin-left: auto; margin-right: auto;"

class="card-body">

<h5 class="card-title">${product.name}</h5>

<p class="card-text">

${product.company}<br>${product.size} inches<br>${product.price}$

</p>


<c:choose>

<c:when test="${userProducts.contains(product.id)}">

<a href="${orderLink}" class="btn btn-danger"><i

class="fa fa-shopping-cart fa-lg"></i> Remove from Cart</a>

</c:when>


<c:otherwise>

<a href="${orderLink}" class="btn btn-primary"><i

class="fa fa-shopping-cart fa-lg"></i> Add to My Cart</a>

</c:otherwise>

</c:choose>

</div>

</div>
```

</c:forEach>

</div>

</div>

</div>

</div>

</div>

</body>

</html>

**[src/main/webapp/WEB-INF/sporty-shoes-servlet.xml](#)**:

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:tx="http://www.springframework.org/schema/tx"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/mvc

http://www.springframework.org/schema/mvc/spring-mvc.xsd

http://www.springframework.org/schema/tx

```
http://www.springframework.org/schema/tx/spring-tx.xsd">
```

```
<!-- Add support for component scanning -->
```

```
<context:component-scan base-package="com.simplilearn" />
```

```
<!-- Add support for conversion, formatting and validation support -->
```

```
<mvc:annotation-driven/>
```

```
<!-- Define Spring MVC view resolver -->
```

```
<bean
```

```
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
```

```
<property name="prefix" value="/WEB-INF/view/" />
```

```
<property name="suffix" value=".jsp" />
```

```
</bean>
```

```
<!-- Step 1: Define Database DataSource / connection pool -->
```

```
<bean id="myDataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"
```

```
destroy-method="close">
```

```
<property name="driverClass" value="com.mysql.cj.jdbc.Driver" />
```

```
<property name="jdbcUrl" value="jdbc:mysql://localhost:3306/shoesdata" />
```

```
<property name="user" value="root" />
```

```
<property name="password" value="Bhavs@333" />
```

```
<!-- these are connection pool properties for C3P0 -->
```

```
<property name="minPoolSize" value="5" />
```

```
<property name="maxPoolSize" value="20" />
<property name="maxIdleTime" value="30000" />
</bean>
```

<!-- Step 2: Setup Hibernate session factory -->

```
<bean id="sessionFactory"
class="org.springframework.orm.hibernate5.LocalSessionFactoryBean">
<property name="dataSource" ref="myDataSource" />
<property name="packagesToScan" value="com.simplilearn.entity" />
<property name="hibernateProperties">
<props>
<prop key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>
<prop key="hibernate.show_sql">true</prop>
</props>
</property>
</bean>
```

<!-- Step 3: Setup Hibernate transaction manager -->

```
<bean id="myTransactionManager"
class="org.springframework.orm.hibernate5.HibernateTransactionManager">
<property name="sessionFactory" ref="sessionFactory"/>
</bean>
```

<!-- Step 4: Enable configuration of transactional behavior based on annotations -->

```
<tx:annotation-driven transaction-manager="myTransactionManager" />
```

<!-- Add support for reading web resources: css, images, js, etc ... -->

<mvc:resources location="/resources/" mapping="/resources/\*\*"></mvc:resources>

</beans>

[src/main/webapp/WEB-INF/web.xml:](#)

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns="http://xmlns.jcp.org/xml/ns/javaee"  
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee  
http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd" id="WebApp\_ID" version="3.1">

<display-name>Sporty-Shoes-Store</display-name>

<absolute-ordering />

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

<welcome-file>index.html</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>dispatcher</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>



```
<param-value>/WEB-INF/sporty-shoes-servlet.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>


<servlet-mapping>

  <servlet-name>dispatcher</servlet-name>

  <url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>
```

**[src/main/webapp/resources/css/login.css](#):**

```
Body {

  font-family: Calibri, Helvetica, sans-serif;

  background-image: url('../images/background.jpg');

}

button {

  justify-content: center;

  background-color: #4CAF50;

  width: 100%;

  color: white;

  padding: 15px;

  margin: 10px 0px;

  border: none;

  cursor: pointer;
```

```
}
```

```
form {
```

```
    border: 1.4px solid black;
```

```
        width: 45%;
```

```
        margin: 0 auto;
```

```
}
```

```
input[type=text], input[type=password] {
```

```
    justify-content: center;
```

```
    width: 100%;
```

```
    margin: 8px 0;
```

```
    padding: 12px 20px;
```

```
    display: inline-block;
```

```
    border: 2px solid green;
```

```
    box-sizing: border-box;
```

```
}
```

```
button:hover {
```

```
    opacity: 0.7;
```

```
}
```

```
.container {
```

```
    justify-content: center;
```

```
padding: 15px;

background-color: #FFF8DC;

}
```

<src/main/webapp/resources/css/style.css>:

```
html, body{

    padding:0px;

    font-family:Verdana, Arial, Helvetica, sans-serif;

    min-height: 100%;

    height: 100%;

}
```

```
.flex-fill {

    flex:1;

}
```

```
.sidenav {

    border-color: #FFFFFFF;

    position: fixed;

    z-index: 1;

    top: 0;

    left: 0;
```

```
background-color: #000080;

overflow-x: hidden;

padding-top: 20px;

min-height: 100%;

height: 100%;

}
```

```
.sidenav a {

padding: 6px 6px 6px 32px;

text-decoration: none;

font-size: 25px;

color: white;

display: block;

}
```

```
.sidenav a:hover {

color: blue;

}
```

```
#page{

height: 100%;

margin-left: 13%;
```

```
}
```

```
#logo{
```

```
    font-family: 'Trebuchet MS', sans-serif;
```

```
    text-align: center;
```

```
    color: white;
```

```
}
```

```
.bar-item{
```

```
    border-color: #FFFFFFF;
```

```
    border-width: 3px;
```

```
    border-bottom: 1.5px solid rgba(255, 255, 255, 0.247);
```

```
}
```

```
table {
```

```
    border-collapse: collapse;
```

```
    border: 1px solid gray;
```

font-family: Tahoma,Verdana,Segoe,sans-serif;

width:63%;

table-layout: fixed;

}

th {

border-bottom:1px solid gray;

background:none repeat scroll 0 0 #0775d3;

padding:10px;

color: #FFFFFF;

}

tr {

border-top:1px solid gray;

text-align:center;

}

tr:nth-child(even) {background: #FFFFFF}

tr:nth-child(odd) {background: #BBBBBB}

**[src/main/webapp/resources/database/sporty-shoes.sql](#):**

-- phpMyAdmin SQL Dump

-- version 5.1.0

-- <https://www.phpmyadmin.net/>

--

-- Host: 127.0.0.1:3307

-- Generation Time: April 13, 2022 at 03:43 PM

-- Server version: 10.4.18-MariaDB

-- PHP Version: 8.0.3

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET

@OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET

@OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET

@OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `sporty-shoes`

--

-----

--

-- Table structure for table `orders`

--

```
CREATE TABLE `orders` (  
  `id` int(11) NOT NULL,  
  `user_id` int(11) NOT NULL,  
  `product_id` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

--

-- Dumping data for table `orders`

--

```
INSERT INTO `orders` (`id`, `user_id`, `product_id`) VALUES  
(7, 3, 1),  
(10, 20, 3),  
(11, 20, 1),  
(12, 20, 4);  
  
drop tables orders;
```

-----

--

-- Table structure for table `products`

--



```

CREATE TABLE `products` (

  `id` int(11) NOT NULL,

  `name` varchar(50) NOT NULL,

  `company` varchar(50) NOT NULL,

  `size` int(11) DEFAULT NULL,

  `price` double NOT NULL,

  `image_link` varchar(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `products`

--

INSERT INTO `products` (`id`, `name`, `company`, `size`, `price`, `image_link`)
VALUES

(1, 'Reebok Originals', 'Reebok', 34, 654, 'https://i.imgur.com/NYeJGJw.jpg'),

(2, 'The Puma 750', 'Puma', 24, 543, 'https://i.imgur.com/JzC8UwF.jpg'),

(3, 'Adidas original', 'Adidas', 45, 678, 'https://imgur.com/0GXitLh.jpg'),

(4, 'Bata Sk8-Hi Flame', 'Bata', 28, 897, 'https://imgur.com/xa1NmjZ.jpg');

drop tables products;

-----

--

-- Table structure for table `users`

--

```

```
CREATE TABLE `users` (  
  `id` int(11) NOT NULL,  
  `type` int(10) NOT NULL DEFAULT 0,  
  `username` varchar(50) NOT NULL,  
  `password` varchar(50) NOT NULL,  
  `age` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
--  
-- Dumping data for table `users`  
--  
  
INSERT INTO `users` (`id`, `type`, `username`, `password`, `age`) VALUES  
(2, 0, 'Anu', '1234', 24),  
(4, 0, 'anju', '1234', 34),  
(6, 0, 'karthi', '1234', 1),  
(8, 0, 'Sam', '1234', 23),  
(15, 1, 'admin', 'admin', 0),  
(16, 0, 'Hasin', '4242', 24),  
(13, 0, 'Manasa', '12345', 33);  
  
drop tables users;  
  
--
```

```
-- Indexes for dumped tables

--

--

-- Indexes for table `orders`

--

ALTER TABLE `orders`

  ADD PRIMARY KEY (`id`),

  ADD KEY `product_id` (`product_id`),

  ADD KEY `user_id` (`user_id`);

commit;

--

-- Indexes for table `products`

--

ALTER TABLE `products`

  ADD PRIMARY KEY (`id`);

commit;

--

-- Indexes for table `users`

--

ALTER TABLE `users`

  ADD PRIMARY KEY (`id`);
```

```
--  
  
-- AUTO_INCREMENT for dumped tables  
  
--  
  
--  
  
-- AUTO_INCREMENT for table `orders`  
  
--  
  
ALTER TABLE `orders`  
  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=13;  
  
commit;  
  
--  
  
-- AUTO_INCREMENT for table `products`  
  
--  
  
ALTER TABLE `products`  
  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;  
  
commit;  
  
--  
  
-- AUTO_INCREMENT for table `users`  
  
--  
  
ALTER TABLE `users`  
  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;  
  
--
```

```
-- Constraints for dumped tables
```

```
--
```

```
--
```

```
-- Constraints for table `orders`
```

```
--
```

```
ALTER TABLE `orders`
```

```
  ADD CONSTRAINT `product_id` FOREIGN KEY (`product_id`) REFERENCES  
`products` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
```

```
  ADD CONSTRAINT `user_id` FOREIGN KEY (`user_id`) REFERENCES `users`  
(`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
COMMIT;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT  
*/;
```

```
/*!40101 SET  
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```

```
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION  
*/;
```

**[src/main/webapp/index.jsp](#):**

```
<% response.sendRedirect("login"); %>
```

