

Sporty Shoes

(Project Source Code)

Version History:

Author	Nikhil Jain
Purpose	Source Code of the application
Date	11 th November 2021
Version	1.0


























Table of Contents

• Project Link	3
• Folder Structure	4
SecurityConfiguration.java	5
1. AdminRegistrationController.java	7
2. CartController.java	8
3. HomeController.java	9
4. ProductController.java	10
5. PurchaseController.java	11
6. RegistrationController.java	12

- Project Link

Repository Name	Sporty Shoes
GitHub Link	https://github.com/Niks4u2/SportyShoes
Deployed On	http://sportyshoesapplication-env.eba-pk5qivsc.ap-south-1.elasticbeanstalk.com/

- Folder Structure

- ▼  sporty-shoes [boot]
 - ▼  src/main/java
 - ▼  com.nikhil.sportyshoes
 - >  SportyShoesApplication.java
 - ▼  com.nikhil.sportyshoes.config
 - >  SecurityConfiguration.java
 - >  com.nikhil.sportyshoes.controller
 - >  com.nikhil.sportyshoes.model
 - ▼  com.nikhil.sportyshoes.repository
 - >  ProductRepository.java
 - >  PurchaseRepository.java
 - >  UserRepository.java
 - >  com.nikhil.sportyshoes.service
 - >  src/main/resources
 - >  src/test/java
 - >  target/generated-sources/annotations
 - >  JRE System Library [JavaSE-1.8]
 - >  Maven Dependencies
 - >  target/generated-test-sources/test-annotations
 - >  src
 - >  target
 - >  HELP.md
 - >  mvnw
 - >  mvnw.cmd
 - >  pom.xml
-

- [Source Code](#)

Configuration

SecurityConfiguration.java

```
package com.nikhil.sportyshoes.config;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.authentication.dao.DaoAuthenticationProvider;
import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.crypto.password.NoOpPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.web.util.matcher.AntPathRequestMatcher;

@SuppressWarnings("deprecation")
@Configuration
@EnableWebSecurity
public class SecurityConfiguration extends WebSecurityConfigurerAdapter {
    @Autowired
    UserDetailsService userDetailsService;

    @Bean
    public PasswordEncoder passwordEncoder() {
        return NoOpPasswordEncoder.getInstance();
    }

    @Bean
    public DaoAuthenticationProvider authenticationProvider()
    {
        DaoAuthenticationProvider auth = new DaoAuthenticationProvider();
        auth.setUserDetailsService(userDetailsService);
        auth.setPasswordEncoder(passwordEncoder());
        return auth;
    }

    // authentication
    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws
Exception {
```

```

        auth.authenticationProvider(authenticationProvider());
    }

    // authorization
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests()
            .antMatchers("/admindashboard**").hasAuthority("ADMIN")
            .antMatchers("/product**",
"/registeredusers**").hasAuthority("ADMIN")
            .antMatchers("/checkout").hasAnyAuthority("ADMIN", "USER")
            .antMatchers("/").permitAll()
            .and()
            .formLogin()
            .loginPage("/login")
            .permitAll()
            .and()
            .logout()
            .invalidateHttpSession(true)
            .clearAuthentication(true)
            .logoutRequestMatcher(new AntPathRequestMatcher("/logout"))
            .logoutSuccessUrl("/login?logout")
            .permitAll();

        http.cors().and().csrf().disable();
    }
}

```

Controllers

1. AdminRegistrationController.java

```
package com.nikhil.sportyshoes.controller;

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 com.nikhil.sportyshoes.model.User;
import com.nikhil.sportyshoes.service.UserService;

@Controller
@RequestMapping("/adminregistration")
public class AdminRegistrationController {

    @Autowired
    private UserService service;

    @GetMapping
    public String adminRegistrationPage(Model model)
    {
        model.addAttribute("admin", new User());
        return "adminregistration";
    }

    @PostMapping
    public String registerAdminAccount(@ModelAttribute("admin") User user)
    {
        user.setRoles("ADMIN");
        service.registerUser(user);
        return "redirect:/adminregistration?success";
    }
}
```

2. CartController.java

```
package com.nikhil.sportyshoes.controller;

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

import com.nikhil.sportyshoes.model.Cart;
import com.nikhil.sportyshoes.model.Product;
import com.nikhil.sportyshoes.service.ProductService;

@Controller
@RequestMapping("/cart")
public class CartController {

    @Autowired
    private ProductService service;

    @GetMapping
    public String getCartPage(Model model)
    {
        model.addAttribute("cartCount", Cart.cart.size());
        model.addAttribute("total",
            Cart.cart.stream().mapToDouble(Product::getPrice).sum());
        model.addAttribute("cart", Cart.cart);
        return "cart";
    }

    @GetMapping("/{prodId}")
    public String addToCart(@PathVariable Long prodId, Model model)
    {
        Cart.cart.add(service.getProductById(prodId));
        model.addAttribute("cart", Cart.cart);
        model.addAttribute("total",
            Cart.cart.stream().mapToDouble(Product::getPrice).sum());
        model.addAttribute("cartCount", Cart.cart.size());
        return "redirect:/cart";
    }

    @GetMapping("/delete/{index}")
    public String removeFromCart(@PathVariable int index, Model model)
    {
        Cart.cart.remove(index);
        model.addAttribute("cart", Cart.cart);
        return "redirect:/cart";
    }
}
```


3. HomeController.java

```
package com.nikhil.sportyshoes.controller;

import java.util.List;

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 com.nikhil.sportyshoes.model.Cart;
import com.nikhil.sportyshoes.model.Product;
import com.nikhil.sportyshoes.service.ProductService;

@Controller
public class HomeController {

    @Autowired
    private ProductService service;

    @GetMapping("/")
    public String home(Model model)
    {
        model.addAttribute("cartCount", Cart.cart.size());
        List<Product> productlist = service.getAllProducts();
        model.addAttribute("prodlist", productlist);
        return "index";
    }

    @GetMapping("/login")
    public String loginPage()
    {
        return "login";
    }

    @GetMapping("/admindashboard")
    public String admin(Model model)
    {
        List<Product> productlist = service.getAllProducts();
        model.addAttribute("prods", productlist);
        return "admindashboard";
    }
}
```

4. ProductController.java

```
package com.nikhil.sportyshoes.controller;

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.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;

import com.nikhil.sportyshoes.model.Product;
import com.nikhil.sportyshoes.service.ProductService;

@Controller
@RequestMapping("/product")
public class ProductController
{
    @Autowired
    private ProductService service;

    @GetMapping
    public String addProductPage(Model model)
    {
        model.addAttribute("product", new Product());
        return "addproduct";
    }

    @PostMapping
    public String addProduct(@ModelAttribute("product") Product product)
    {
        service.addProduct(product);
        return "redirect:/product?success";
    }

    @GetMapping("/{prodId}")
    public String editProductById(@PathVariable Long prodId, Model model)
    {
        Product product = service.getProductById(prodId);
        model.addAttribute("prod", product);
        return "editproduct";
    }

    @GetMapping("/delete/{prodId}")
    public String deleteProduct(@PathVariable("prodId") Long prodId, Model
model) {
        service.deleteProductById(prodId);
        return "redirect:/admindashboard?success";
    }
}
```

5. PurchaseController.java

```
package com.nikhil.sportyshoes.controller;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
import java.util.stream.Collectors;

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 com.nikhil.sportyshoes.model.Cart;
import com.nikhil.sportyshoes.model.Product;
import com.nikhil.sportyshoes.model.Purchase;
import com.nikhil.sportyshoes.service.MyUserDetailsService;
import com.nikhil.sportyshoes.service.PurchaseService;

@Controller
public class PurchaseController {
    @Autowired
    private MyUserDetailsService detailsService;

    @Autowired
    private PurchaseService service;

    @GetMapping("/checkout")
    public String checkout(Model model)
    {
        String productName =
        Cart.cart.stream().map(Product::getProductName)

        .collect(Collectors.joining(", "));
        String purchaseDate = new SimpleDateFormat("dd-MM-
        yyyy").format(new Date());
        double totalCost =
        Cart.cart.stream().mapToDouble(Product::getPrice).sum();
        String user = detailsService.getLoggedInUsername();
        Purchase purchase = new Purchase(productName, purchaseDate,
        totalCost , user);
        service.addPurchaseDetails(purchase);
        model.addAttribute("cart", Cart.cart);
        model.addAttribute("total",
        Cart.cart.stream().mapToDouble(Product::getPrice).sum());
        return "checkout";
    }

    @GetMapping("/purchasereport")
    public String getPurchaseReport(Model model)
    {
        List<Purchase> purchases = service.getAllPurchaseReport();
        model.addAttribute("purchases", purchases);
        return "purchasereport";
    }
}
```

6. RegistrationController.java

```
package com.nikhil.sportyshoes.controller;

import java.util.List;

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 com.nikhil.sportyshoes.model.User;
import com.nikhil.sportyshoes.service.UserService;

@Controller
public class RegistrationController {

    @Autowired
    private UserService service;

    @GetMapping("/registration")
    public String userRegistrationPage(Model model)
    {
        model.addAttribute("user", new User());
        return "registration";
    }

    @PostMapping("/registration")
    public String registerUserAccount(@ModelAttribute("user") User user)
    {
        user.setRoles("USER");
        service.registerUser(user);
        return "redirect:/registration?success";
    }

    @GetMapping("/registeredusers")
    public String getRegisteredUsers(Model model)
    {
        List<User> users = service.getAllRegisteredUser();
        model.addAttribute("users", users);
        return "registeredusers";
    }
}
```