1)

package org.Project;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.autoconfigure.security.servlet.UserDetailsServiceAutoConfiguration;

@SpringBootApplication(exclude= {UserDetailsServiceAutoConfiguration.class})

public class AdminShoePortal12Application {

public static void main(String[] args) {

SpringApplication.run(AdminShoePortal12Application.class, args);

}

}

2) package org.Project.configuration;

import org.Project.service.CustomUserDetailService;

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.builders.WebSecurity;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

@Configuration

@EnableWebSecurity

public class SecurityConfig {

@Autowired

private CustomUserDetailService customUserDetailService;

@SuppressWarnings({ "deprecation", "removal" })

public void configure(HttpSecurity http) throws Exception {

http .authorizeRequests().requestMatchers("/admin/\*\*").hasRole("ROLE\_ADMIN")

.requestMatchers("/user/\*\*").hasRole("ROLE\_USER")

.requestMatchers("/welcome/\*").permitAll()

.and().formLogin().and().csrf().disable();

// http.csrf().disable()

// .authorizeHttpRequests((authorize) -> authorize.requestMatchers("/register/\*\*", "/shop/\*\*").permitAll()

// .requestMatchers("/login").permitAll().requestMatchers("/admin").hasRole("ADMIN"))

// .formLogin(form -> form.loginPage("/login").permitAll().failureUrl("/login?error=true")

// .defaultSuccessUrl("/").usernameParameter("username").passwordParameter("password"))

// .logout(logout -> logout.logoutRequestMatcher(new AntPathRequestMatcher("/logout"))

// .logoutSuccessUrl("/login").permitAll());

// http.build();

}

@Bean

public static BCryptPasswordEncoder bCryptPasswordEncoder() {

return new BCryptPasswordEncoder();

}

@Bean

public DaoAuthenticationProvider authenticationProvider() {

DaoAuthenticationProvider daoAuthenticationProvider = new DaoAuthenticationProvider();

daoAuthenticationProvider.setUserDetailsService(customUserDetailService);

daoAuthenticationProvider.setPasswordEncoder(bCryptPasswordEncoder());

return daoAuthenticationProvider;

}

protected void configure(AuthenticationManagerBuilder auth) throws Exception {

auth.authenticationProvider(authenticationProvider());

}

protected void configure(WebSecurity web) throws Exception {

web.ignoring().requestMatchers("/resourses/\*\*", "/static/", "/productImage/\*\*");

}

}

3)

package org.Project.controller;

import java.io.IOException;

import org.Project.entities.Category;

import org.Project.entities.Product;

import org.Project.service.CategoryService;

import org.Project.service.ProductService;

import org.Project.service.UserService;

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.RequestParam;

import org.springframework.web.multipart.MultipartFile;

@Controller

public class AdminController {

public static String uploadDir = System.getProperty("user.dir") + "/src/main/resources/static/productImage";

private ProductService productService;

private UserService userService;

private CategoryService categoryService;

public AdminController(ProductService productService, UserService userService, CategoryService categoryService) {

super();

this.productService = productService;

this.userService = userService;

this.categoryService = categoryService;

}

@GetMapping("/admin")

public String showHome() {

return "index";

}

// categoryOperation

@GetMapping("/admin/categories")

public String getCategory(Model model) {

model.addAttribute("categories", categoryService.getAllCategory());

return "categories";

}

@GetMapping("/admin/categories/new")

public String createCategotyForm(Model model) {

model.addAttribute("category", new Category());

return "createCategory";

}

@PostMapping("/admin/categories")

public String saveCategory(@ModelAttribute("category") Category category) {

categoryService.saveCategory(category);

return "redirect:/admin/categories";

}

@GetMapping("/admin/categories/edit/{id}")

public String editCategoryForm(@PathVariable Long id, Model model) {

model.addAttribute("category", categoryService.getCategoryById(id));

return "editCategory";

}

@PostMapping("/admin/categories/update/{id}")

public String updateCategory(@PathVariable("id") Long id, @ModelAttribute("category") Category category,

Model model) {

Category ex1 = categoryService.getCategoryById(id);

ex1.setId(id);

ex1.setName(ex1.getName());

categoryService.updateCategory(ex1);

return "redirect:/admin/categories";

}

@GetMapping("/admin/categories/delete/{id}")

public String deleteCategory(@PathVariable Long id) {

categoryService.deleteCategoryById(id);

return "redirect:/admin/categories";

}

// Product Operation

@GetMapping("/admin/products")

public String listProduct(Model model) {

model.addAttribute("products", productService.getAllProduct());

return "products";

}

@GetMapping("/admin/products/new")

public String createProductForm(Model model) {

model.addAttribute("product", new Product());

model.addAttribute("categories", categoryService.getAllCategory());

return "createProduct";

}

@PostMapping("/admin/products")

public String saveProduct(@ModelAttribute("product") Product product,

@RequestParam("productImage") MultipartFile file, Model model) throws IOException {

Product products = new Product();

products.setId(product.getId());

products.setName(product.getName());

products.setCategory(categoryService.getCategoryById(product.getCategory().getId()));

products.setPrice(product.getPrice());

String imageUID = org.springframework.util.StringUtils.cleanPath(file.getOriginalFilename());

if (imageUID.contains("..")) {

System.out.print(" Not valid available");

}

try {

// products.setImageName(Base64.getEncoder().encodeToString(file.getBytes()));

products.setImageName(imageUID);

} catch (Exception e) {

e.printStackTrace();

}

model.addAttribute("product",productService.saveProduct(products));

//productService.saveProduct(product);

return "redirect:/admin/products";

}

@GetMapping("/admin/products/edit/{id}")

public String editProductForm(@PathVariable Long id, Model model) {

model.addAttribute("product", productService.getProductById(id));

model.addAttribute("categories", categoryService.getAllCategory());

return "editProduct";

}

@PostMapping("/admin/products/update/{id}")

public String updateProduct(@PathVariable("id") Long id, @ModelAttribute("product") Product product, Model model) {

Product ex = productService.getProductById(id).get();

ex.setId(id);

ex.setName(product.getName());

ex.setCategory(product.getCategory());

ex.setPrice(product.getPrice());

ex.setImageName(product.getImageName());

productService.updateProduct(ex);

return "redirect:/admin/products";

}

@GetMapping("/admin/products/delete/{id}")

public String deleteProduct(@PathVariable("id") Long id) {

productService.deleteProductById(id);

return "redirect:/admin/products";

}

// user Data

@GetMapping("/admin/users")

public String users(Model model) {

model.addAttribute("users", userService.FindAllUser());

return "users";

}

@GetMapping("/admin/users/delete/{id}")

public String deleteUSer(@PathVariable Long id) {

userService.deteteUserById(id);

return "redirect:/admin/users";

}

}

4)

package org.Project.controller;

import java.time.LocalDate;

import org.Project.GlobalData.GlobalData;

import org.Project.entities.Product;

import org.Project.service.ProductService;

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.PostMapping;

@Controller

public class CartController {

private ProductService productService;

public CartController(ProductService productService) {

super();

this.productService = productService;

}

@GetMapping("/addToCart/{id}")

public String addToCart(@PathVariable Long id)

{

GlobalData.cart.add(productService.getProductById(id).get());

return "redirect:/shop";

}

@GetMapping("/cart")

public String cartGet(Model model)

{

model.addAttribute("cartCount",GlobalData.cart.size());

model.addAttribute("total", GlobalData.cart.stream().mapToDouble(Product::getPrice).sum());

model.addAttribute("cart",GlobalData.cart);

return "cart";

}

@GetMapping("/cart/removeItem/{index}")

public String cartItemRemove(@PathVariable int index)

{

GlobalData.cart.remove(index);

return "redirect:/cart";

}

@GetMapping("/checkout")

public String checkout(Model model)

{

model.addAttribute("total", GlobalData.cart.stream().mapToDouble(org.Project.entities.Product::getPrice).sum());

return "checkout";

}

// payNow

@PostMapping("/payNow")

public String payNow(Model model) {

model.addAttribute("result","congratulations..!!,your order is now booked");

LocalDate localDate = LocalDate.now();

model.addAttribute("localDate",localDate);

model.addAttribute("parameters", GlobalData.cart);

model.addAttribute("total", GlobalData.cart.stream().mapToDouble(Product::getPrice).sum());

return "orderPlaced";

}

}

5)

**package** org.Project.controller;

**import** java.util.Optional;

**import** org.Project.GlobalData.GlobalData;

**import** org.Project.entities.Product;

**import** org.Project.service.CategoryService;

**import** org.Project.service.ProductService;

**import** org.Project.service.UserService;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

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

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

@Controller

**public** **class** HomeController {

**private** CategoryService categoryService;

**private** ProductService productService;

@SuppressWarnings("unused")

**private** UserService userService;

**public** HomeController(CategoryService categoryService, ProductService productService, UserService userService) {

**super**();

**this**.categoryService = categoryService;

**this**.productService = productService;

**this**.userService = userService;

}

@GetMapping({"/home","/"})

**public** String home(Model model) {

model.addAttribute("cartCount", GlobalData.*cart*.size());

**return** "welcome";

}

@GetMapping("/shop")

**public** String shop(Model model) {

model.addAttribute("categories", categoryService.getAllCategory());

model.addAttribute("products", productService.getAllProduct());

model.addAttribute("cartCount", GlobalData.*cart*.size());

**return** "shop";

}

@GetMapping("/shop/category/{id}")

**public** String shopByCategory(Model model, @PathVariable **long** id) {

model.addAttribute("categories", categoryService.getAllCategory());

model.addAttribute("products", productService.getAllProductByCategoryId(id));

model.addAttribute("cartCount", GlobalData.*cart*.size());

**return** "shop";

}

@GetMapping("/shop/viewproduct/{id}")

**public** String viewProduct(Model model, @PathVariable **long** id) {

Optional<Product> product=productService.getProductById(id);

model.addAttribute("product", productService.getProductById(id));

model.addAttribute("product", product.get());

model.addAttribute("cartCount", GlobalData.*cart*.size());

**return** "viewProduct";

}

}

6) package org.Project.controller;

import java.util.ArrayList;

import java.util.List;

import org.Project.GlobalData.GlobalData;

import org.Project.entities.Role;

import org.Project.entities.User;

import org.Project.repository.RoleRepository;

import org.Project.repository.UserRepository;

import org.Project.service.UserService;

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

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

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

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

import jakarta.servlet.ServletException;

import jakarta.servlet.http.HttpServletRequest;

@Controller

public class LoginController {

private BCryptPasswordEncoder bCryptPasswordEncoder;

private UserRepository userRepository;

@Autowired

UserService userService;

@Autowired

RoleRepository roleRepository;

public LoginController( UserRepository userRepository,

RoleRepository roleRepository) {

super();

this.bCryptPasswordEncoder = bCryptPasswordEncoder;

this.userRepository = userRepository;

}

@GetMapping("/login")

public String login() {

GlobalData.cart.clear();

return "index";

}

@GetMapping("/register")

public String registerGet(Model model) {

User user=new User();

model.addAttribute("user", "user");

return "register";

}

// @PostMapping("/register")

// public String registration(@ModelAttribute("user") User user,

// BindingResult result,

// Model model){

// User existingUser = userService.findUserByName(user.getusername());

//

// if(existingUser != null && existingUser.getusername() != null && !existingUser.getusername().isEmpty()){

// result.rejectValue("username", null,

// "There is already an account registered with the same username");

// }

//

// if(result.hasErrors()){

// model.addAttribute("user", user);

// return "/register";

// }

//

// userService.saveUser(user);

// return "redirect:/";

// }

@PostMapping("/register")

public String registerPost(@ModelAttribute("user") User user, HttpServletRequest request) throws ServletException {

String password = user.getPassword();

user.setPassword(bCryptPasswordEncoder.encode(password));

List<Role> roles = new ArrayList<>();

roles.add(roleRepository.findByName("ROLE\_ADMIN"));

user.setRoles(roles);

userRepository.save(user);

request.login(user.getusername(), password);

return "redirect:/";

}

}

7) package org.Project.entities;

import java.util.ArrayList;

import java.util.List;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.OneToMany;

@Entity

public class Category {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "category\_id")

private Long id;

private String name;

@OneToMany(mappedBy = "category")

private List<Product> products=new ArrayList<>();

public Category() {

// TODO Auto-generated constructor stub

}

public Category(Long id, String name) {

super();

this.id = id;

this.name = name;

}

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 List<Product> getProducts() {

return products;

}

public void setProducts(List<Product> products) {

this.products = products;

}

}

8) package org.Project.entities;

import java.util.ArrayList;

import java.util.Collection;

import java.util.List;

import org.springframework.security.core.GrantedAuthority;

import org.springframework.security.core.authority.SimpleGrantedAuthority;

import org.springframework.security.core.userdetails.UserDetails;

@SuppressWarnings("serial")

public class CustomUserDetail extends User implements UserDetails {

public CustomUserDetail(User user) {

super(user);

}

@Override

public Collection<? extends GrantedAuthority> getAuthorities() {

List<GrantedAuthority> authorityList = new ArrayList<>();

super.getRoles().forEach(role -> {

authorityList.add(new SimpleGrantedAuthority(role.getName()));

});

return authorityList;

}

@Override

public String getUsername() {

return super.getusername();

}

@Override

public String getPassword() {

return super.getPassword();

}

@Override

public boolean isAccountNonExpired() {

return true;

}

@Override

public boolean isAccountNonLocked() {

return true;

}

@Override

public boolean isCredentialsNonExpired() {

return true;

}

@Override

public boolean isEnabled() {

return true;

}

}

9) package org.Project.entities;

import jakarta.persistence.CascadeType;

import jakarta.persistence.Entity;

import jakarta.persistence.FetchType;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.ManyToOne;

@Entity

public class Product {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private double price;

private String imageName;

@ManyToOne(fetch = FetchType.LAZY)

@JoinColumn(name = "category\_id", referencedColumnName = "category\_id")

private Category category;

@ManyToOne(cascade = CascadeType.ALL)

@JoinColumn(name ="user\_id" ,referencedColumnName = "user\_id")

private User user;

public Product() {

// TODO Auto-generated constructor stub

}

public Product(Long id, String name, double price, String imageName) {

super();

this.id = id;

this.name = name;

this.price = price;

this.imageName = imageName;

}

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 getImageName() {

return imageName;

}

public void setImageName(String imageName) {

this.imageName = imageName;

}

public Category getCategory() {

return category;

}

public void setCategory(Category category) {

this.category = category;

}

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

}

10)

package org.Project.entities;

import java.util.List;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.ManyToMany;

@Entity

public class Role {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name="role\_id")

private Long id;

@Column(nullable = false,unique = true)

private String name;

@ManyToMany(mappedBy = "roles")

private List<User> users;

public Role() {

// TODO Auto-generated constructor stub

}

public Role(Long id, String name, List<User> users) {

super();

this.id = id;

this.name = name;

this.users = users;

}

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 List<User> getUsers() {

return users;

}

public void setUsers(List<User> users) {

this.users = users;

}

}

11)

package org.Project.entities;

import java.util.ArrayList;

import java.util.Collection;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import jakarta.persistence.CascadeType;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.FetchType;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.JoinTable;

import jakarta.persistence.ManyToMany;

import jakarta.persistence.OneToMany;

@Entity

public class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "user\_id")

private Long id;

@Column(nullable = false, unique = true)

private String username;

private String password;

private String address;

private String email;

@ManyToMany(cascade = CascadeType.MERGE, fetch = FetchType.EAGER)

@JoinTable(name = "user\_role", joinColumns =@JoinColumn(name = "user\_id"),

inverseJoinColumns = @JoinColumn(name = "role\_id"))

public List<Role> roles =new ArrayList<>();

@OneToMany(mappedBy = "user")

private Set<Product> products = new HashSet<Product>();

public User() {

// TODO Auto-generated constructor stub

}

public User(User user) {

super();

this.id = user.id;

this.username = user.username;

this.password = user.password;

this.address = user.address;

this.email = user.email;

this.roles = user.roles;

}

public User(String username, String password,List<Role> roles) {

super();

this.username = username;

this.password = password;

this.roles = roles;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

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 String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public List<Role> getRoles() {

return roles;

}

public void setRoles(List<Role> roles) {

this.roles = roles;

}

public Set<Product> getProducts() {

return products;

}

public void setProducts(Set<Product> products) {

this.products = products;

}

}

12) package org.Project.GlobalData;

import java.util.ArrayList;

import java.util.List;

import org.Project.entities.Product;

public class GlobalData {

public static List<Product> cart;

static

{

cart=new ArrayList<Product>();

}

}

13)

package org.Project.repository;

import org.Project.entities.Category;

import org.springframework.data.jpa.repository.JpaRepository;

public interface CategoryRepository extends JpaRepository<Category, Long> {

}

14) package org.Project.repository;

import java.util.List;

import org.Project.entities.Product;

import org.springframework.data.jpa.repository.JpaRepository;

public interface ProductRepository extends JpaRepository<Product,Long>{

List<Product> findAllByCategory\_Id(long id);

}

15) package org.Project.repository;

import org.Project.entities.Role;

import org.springframework.data.jpa.repository.JpaRepository;

public interface RoleRepository extends JpaRepository<Role, Long>{

Role findByName(String string);

//Role findeById(Long id);

}

16)

package org.Project.repository;

import org.Project.entities.User;

import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User, Long> {

User findUserByUsername(String username);

}

17)

package org.Project.service;

import java.util.List;

import org.Project.entities.Category;

import org.Project.repository.CategoryRepository;

import org.springframework.stereotype.Service;

@Service

public class CategoryService

{

private CategoryRepository categoryRepository;

public CategoryService(CategoryRepository categoryRepository) {

super();

this.categoryRepository = categoryRepository;

}

public List<Category> getAllCategory() {

return categoryRepository.findAll();

}

public Category saveCategory(Category category) {

return categoryRepository.save(category);

}

public Category getCategoryById(Long id) {

return categoryRepository.findById(id).get();

}

public Category updateCategory(Category category)

{

return categoryRepository.save(category);

}

public void deleteCategoryById(Long id) {

categoryRepository.deleteById(id);

}

}

18) package org.Project.service;

import java.util.Collection;

import java.util.stream.Collectors;

import org.Project.entities.Role;

import org.Project.entities.User;

import org.Project.repository.UserRepository;

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

import org.springframework.security.core.GrantedAuthority;

import org.springframework.security.core.authority.SimpleGrantedAuthority;

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.security.core.userdetails.UserDetailsService;

import org.springframework.security.core.userdetails.UsernameNotFoundException;

import org.springframework.stereotype.Service;

@Service

public class CustomUserDetailService implements UserDetailsService {

@Autowired

private UserRepository userRepository;

@Override

public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

User user = userRepository.findUserByUsername(username);

if(user!=null)

{

return new org.springframework.security.core.userdetails.User(user.getusername(), user.getPassword(),mapRolesToAuthorities(user.getRoles()));

}else {

throw new UsernameNotFoundException("Invalid User or password");

}

}

private Collection<? extends GrantedAuthority> mapRolesToAuthorities(Collection<Role> roles) {

Collection < ? extends GrantedAuthority> mapRoles = roles.stream().map(role -> new SimpleGrantedAuthority(role.getName()))

.collect(Collectors.toList());

return mapRoles;

}

}

19)

package org.Project.service;

import java.util.List;

import java.util.Optional;

import org.Project.entities.Product;

import org.Project.repository.CategoryRepository;

import org.Project.repository.ProductRepository;

import org.springframework.stereotype.Service;

@Service

public class ProductService {

private ProductRepository productRepository;

public ProductService(ProductRepository productRepository, CategoryRepository categoryRepository) {

super();

this.productRepository = productRepository;

}

// getAllProduct

public List<Product> getAllProduct() {

return productRepository.findAll();

}

// public Product addProduct(ProductModel )

// //Add New Product

public Product saveProduct(Product product) {

return productRepository.save(product);

}

// Get Product By ID

public Optional<Product> getProductById(Long id) {

return productRepository.findById(id);

}

// upadte Product data

public Product updateProduct(Product product) {

return productRepository.save(product);

}

// delete Product By id

public void deleteProductById(Long id) {

productRepository.deleteById(id);

}

public List<Product> getAllProductByCategoryId(long id)

{

return productRepository.findAllByCategory\_Id(id);

}

}

20)

package org.Project.service;

import java.util.Arrays;

import java.util.List;

import java.util.stream.Collectors;

import org.Project.entities.Role;

import org.Project.entities.User;

import org.Project.repository.RoleRepository;

import org.Project.repository.UserRepository;

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

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import org.springframework.stereotype.Service;

@Service

public class UserService {

@Autowired

private UserRepository userRepository;

@Autowired

private RoleRepository roleRepository;

@Autowired

private BCryptPasswordEncoder bCryptPasswordEncoder;

// public void saveUser(User user) {

// User users = new User();

// user.setusername(users.getusername());

// user.setPassword(bCryptPasswordEncoder.encode(users.getPassword()));

// user.setAddress(users.getAddress());

// user.setEmail(users.getEmail());

// Role role = roleRepository.findByName("ROLE\_ADMIN");

// if (role == null) {

// role = checkRoleExist();

// }

// user.setRoles(Arrays.asList(role));

// userRepository.save(user);

// }

//

public User findUserByName(String username)

{

return userRepository.findUserByUsername(username);

}

public List<User> FindAllUser() {

return userRepository.findAll();

}

// public List<User> findAllUser() {

// List<User> users=userRepository.findAll();

// return users.stream().map((user)-> mapToUser(user)).collect(Collectors.toList());

//

// }

//

//

// private User mapToUser(User user) {

// User user1 = new User();

// String st=user1.getusername();

// user1.setAddress(user1.getAddress());;

// user1.setEmail(user1.getEmail());

// return user1;

// }

//

//

public void deteteUserById(Long id) {

userRepository.deleteById(id);

}

// private Role checkRoleExist() {

// Role role = new Role();

// role.setName("ROLE\_ADMIN");

// return roleRepository.save(role);

// }

}

21) <!doctype html>

<html lang="en" xmlns:th="http://www.thymeleaf.org"

xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"

integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.11.2/css/all.css">

<title>Cart</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<div class="container-fluid">

<a class="navbar-brand" href="#">

<img th:src="@{/images/logo.png}" src="../static/images/logo.png" width="auto" height="40"

class="d-inline-block align-top" alt="" />

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"

aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav mr-auto"></ul>

<ul class="navbar-nav">

<li class="nav-item active">

<a class="nav-link" th:href="@{/}" href="#">home</a>

</li>

<li class="nav-item active">

<a class="nav-link" th:href="@{/shop}" href="#">shop</a>

</li>

<li class="nav-item active">

<a class="nav-link" sec:authorize="isAnonymous()" th:href="@{/login}" href="#">login</a>

</li>

<li class="nav-item active">

<a class="nav-link" sec:authorize="isAuthenticated()" th:href="@{/logout}" href="#">logout</a>

</li>

<li class="nav-item active">

<a class="nav-link" th:href="@{/cart}" href="#">cart</a>

</li>

<li class="nav-item active">

<span class="nav-link" th:text="${cartCount}">0</span>

</li>

</ul>

</div>

</div>

</nav>

<main>

<div class="container">

<!--Section: Block Content-->

<section class="mt-5 mb-4">

<!--Grid row-->

<div class="row">

<!--Grid column-->

<div class="col-lg-8">

<!-- Card -->

<div class="card wish-list mb-4">

<div class="card-body">

<h5 class="mb-4">Cart (<span th:text="${cartCount}">2</span> items)</h5>

<div class="row mb-4" th:each="item, iStat : ${cart}">

<div class="col-md-5 col-lg-3 col-xl-3">

<div class="mb-3 mb-md-0">

<img class="img-fluid w-100"

th:src="@{${'/productImage/' + item.imageName}}"

src="https://mdbootstrap.com/img/Photos/Horizontal/E-commerce/Vertical/12a.jpg"

alt="Sample">

</div>

</div>

<div class="col-md-7 col-lg-9 col-xl-9">

<div>

<div class="d-flex justify-content-between">

<div>

<h5 th:text="${item.name}">Item:</h5>

<!--<p th:text="${product.category.name}"></p>-->

</div>

</div>

<div class="mt-5 d-flex justify-content-between align-items-center">

<div>

<a href="#!"

th:href="@{/cart/removeItem/{index}(index=${iStat.index})}"

type="button"

class="card-link-secondary small text-uppercase mr-3"><i

class="fas fa-trash-alt mr-1"></i> Remove item </a>

</div>

<p class="mb-0"><span><strong>₹<span

th:text="${item.price}"></span></strong></span></p>

</div>

</div>

</div>

</div>

<hr class="mb-4">

<p class="text-primary mb-0"><i class="fas fa-info-circle mr-1"></i> Do not delay the

purchase, adding

items to your cart does not mean booking them.</p>

</div>

</div>

</div>

<!--Grid column-->

<!--Grid column-->

<div class="col-lg-4">

<!-- Card -->

<div class="card mb-4">

<div class="card-body">

<h5 class="mb-3">The total amount </h5>

<ul class="list-group list-group-flush">

<li

class="list-group-item d-flex justify-content-between align-items-center border-0 px-0 pb-0">

Amount Payable

<span>₹<span th:text="${total}"></span></span>

</li>

<li class="list-group-item d-flex justify-content-between align-items-center px-0">

Shipping

<span>Gratis</span>

</li>

<li

class="list-group-item d-flex justify-content-between align-items-center border-0 px-0 mb-3">

<div>

<strong>The total amount of</strong>

<strong>

<p class="mb-0">(including VAT)</p>

</strong>

</div>

<span><strong>₹<span th:text="${total}"></span></strong></span>

</li>

</ul>

<a th:href="@{/checkout}" h

class="btn btn-primary btn-block waves-effect waves-light">go to

checkout</a>

</div>

</div>

<!-- Card -->

<!-- Card -->

<div class="card mb-4">

<div class="card-body">

<a class="dark-grey-text d-flex justify-content-between" data-toggle="collapse"

href="#collapseExample" aria-expanded="false" aria-controls="collapseExample">

Add a discount code (optional)

<span><i class="fas fa-chevron-down pt-1"></i></span>

</a>

<div class="collapse" id="collapseExample">

<div class="mt-3">

<div class="md-form md-outline mb-0">

<input type="text" id="discount-code" class="form-control font-weight-light"

placeholder="Enter discount code">

</div>

</div>

</div>

</div>

</div>

<!-- Card -->

</div>

<!--Grid column-->

</div>

<!--Grid row-->

</section>

<!--Section: Block Content-->

</div>

</main>

<!--Main layout-->

</body>

</html>

22) <!DOCTYPE html>

<html lang="en" xmlns:th="http://www.thymeleaf.org">

<head>

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

<title>Admin Shoes Portal</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"

integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

</head>

<body>

<nav class="navbar navbar-expand-md bg-dark navbar-dark">

<!-- Brand -->

<a class="navbar-brand" href="#">Admin Shoes Portal</a>

<!-- Toggler/collapsibe Button -->

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">

<span class="navbar-toggler-icon"></span>

</button>

<!-- Navbar links -->

<div class="collapse navbar-collapse" id="collapsibleNavbar">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" th:href="@{/admin}">Welcome Admin</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/products}">Product</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/users}">user</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/categories}">category</a>

</li>

</ul>

</div>

</nav>

<div class="container">

<div class="row">

<h1> List Categories </h1>

</div>

<div class="row">

<div class="col-lg-3">

<a th:href="@{/admin/categories/new}" class="btn btn-primary btn-sm mb-3"> Add Category</a>

</div>

</div>

<table class="table table-striped table-bordered">

<thead class="table-dark">

<tr>

<th>SN</th>

<th> Category Name</th>

<th> Action </th>

</tr>

</thead>

<tbody>

<tr th:each="category :${categories}">

<th scope="row" th:text="${category.id}"></th>

<td th:text="${category.name}"></td>

<td>

<a th:href="@{/admin/categories/edit/{id}(id=${category.id})}"

class="btn btn-primary">Update</a>

<a th:href="@{/admin/categories/delete/{id}(id=${category.id})}"

class="btn btn-danger">Delete</a>

</td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

23) <!doctype html>

<html lang="en" xmlns:th="http://www.thymeleaf.org"

xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"

integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.11.2/css/all.css">

<title>Document</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-light bg-light" >

<div class="container-fluid">

<a class="navbar-brand" href="#">

<img th:src="@{/images/logo.png}" src="../static/images/logo.png" width="auto" height="40" class="d-inline-block align-top" alt=""/>

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav mr-auto"></ul>

<ul class="navbar-nav">

<li class="nav-item active">

<a class="nav-link" th:href="@{/}" href="#">home</a>

</li>

<li class="nav-item active">

<a class="nav-link" th:href="@{/shop}" href="#">shop</a>

</li>

<li class="nav-item active">

<a class="nav-link" sec:authorize="isAnonymous()" th:href="@{/login}" href="#">login</a>

</li>

<li class="nav-item active">

<a class="nav-link" sec:authorize="isAuthenticated()" th:href="@{/logout}" href="#">logout</a>

</li>

<li class="nav-item active">

<a class="nav-link" th:href="@{/cart}" href="#">cart</a>

</li>

<li class="nav-item active">

<span class="nav-link" th:text="${cartCount}">0</span>

</li>

</ul>

</div>

</div>

</nav>

<div class="container">

<!--Section: Block Content-->

<section class="mt-5 mb-4">

<!--Grid row-->

<div class="row">

<!--Grid column-->

<div class="col-lg-8 mb-4">

<!-- Card -->

<div class="card wish-list pb-1">

<div class="card-body">

<h5 class="mb-2">Billing details</h5>

<!-- Grid row -->

<div class="row">

<!-- Grid column -->

<div class="col-lg-6">

<!-- First name -->

<div class="md-form md-outline mb-0 mb-lg-4">

<label for="firstName">First name</label>

<input type="text" id="firstName" required class="form-control mb-0 mb-lg-2">

</div>

</div>

<!-- Grid column -->

<!-- Grid column -->

<div class="col-lg-6">

<!-- Last name -->

<div class="md-form md-outline">

<label for="lastName">Last name</label>

<input type="text" id="lastName" required class="form-control">

</div>

</div>

<!-- Grid column -->

</div>

<!-- Grid row -->

<!-- Country -->

<div class="d-flex flex-wrap">

<div class="select-outline position-relative w-100">

<label>Country - <strong>INDIA</strong></label>

</div>

</div>

<!-- Address Part 1 -->

<div class="md-form md-outline mt-2">

<label for="form14">Address (Line 1)</label>

<input type="text" required id="form14" placeholder="House number and street name" class="form-control">

</div>

<!-- Address Part 2 -->

<div class="md-form md-outline mt-2">

<label for="form15">Address (Line 2)</label>

<input type="text" id="form15" placeholder="Apartment, suite, unit etc. (optional)"

class="form-control">

</div>

<!-- Postcode / ZIP -->

<div class="md-form md-outline mt-3">

<label for="form16">Postcode / ZIP</label>

<input type="text" maxlength="6" minlength="6" required id="form16" class="form-control">

</div>

<!-- Town / City -->

<div class="md-form md-outline mt-2">

<label for="form17">Town / City</label>

<input type="text" required id="form17" class="form-control">

</div>

<!-- Phone -->

<div class="md-form md-outline mt-2">

<label for="form18">Phone</label>

<input type="number" minlength="10" maxlength="10" required id="form18" class="form-control">

</div>

<!-- Email address -->

<div class="md-form md-outline mt-2">

<label for="form19">Email address</label>

<input type="email" required id="form19" class="form-control">

</div>

<!-- Additional information -->

<div class="md-form md-outline mt-2">

<label for="form76">Additional information</label>

<textarea id="form76" class="md-textarea form-control" rows="4"></textarea>

</div>

</div>

</div>

<!-- Card -->

</div>

<!--Grid column-->

<!--Grid column-->

<div class="col-lg-4">

<!-- Card -->

<div class="card mb-4">

<div class="card-body">

<h5 class="mb-3">The total amount </h5>

<ul class="list-group list-group-flush">

<li class="list-group-item d-flex justify-content-between align-items-center border-0 px-0 pb-0">

Amount Payable

<span>₹<span th:text="${total}"></span></span>

</li>

<li class="list-group-item d-flex justify-content-between align-items-center px-0">

Shipping

<span>Gratis</span>

</li>

<li class="list-group-item d-flex justify-content-between align-items-center border-0 px-0 mb-3">

<div>

<strong>The total amount of</strong>

<strong>

<p class="mb-0">(including VAT)</p>

</strong>

</div>

<span><strong>₹<span th:text="${total}"></span></strong></span>

</li>

</ul>

<form action="" method="post" th:action="@{/payNow}" >

<button type="submit" class="btn btn-primary btn-block waves-effect waves-light">Pay Now</button>

</form>

</div>

</div>

<!-- Card -->

<!-- Card -->

<div class="card mb-4">

<div class="card-body">

<a class="dark-grey-text d-flex justify-content-between" data-toggle="collapse" href="#collapseExample"

aria-expanded="false" aria-controls="collapseExample">

Add a discount code (optional)

<span><i class="fas fa-chevron-down pt-1"></i></span>

</a>

<div class="collapse" id="collapseExample">

<div class="mt-3">

<div class="md-form md-outline mb-0">

<input type="text" id="discount-code" class="form-control font-weight-light"

placeholder="Enter discount code">

</div>

</div>

</div>

</div>

</div>

<!-- Card -->

</div>

<!--Grid column-->

</div>

<!--Grid row-->

</section>

<!--Section: Block Content-->

</div>

</body>

</html>

24)

<!DOCTYPE html>

<html lang="en" xmlns:th="http://www.thymeleaf.org">

<head>

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

<title>Admin Shoes Portal</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"

integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

</head>

<body>

<nav class="navbar navbar-expand-md bg-dark navbar-dark">

<!-- Brand -->

<a class="navbar-brand" href="#">Admin Shoes Portal</a>

<!-- Toggler/collapsibe Button -->

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">

<span class="navbar-toggler-icon"></span>

</button>

<!-- Navbar links -->

<div class="collapse navbar-collapse" id="collapsibleNavbar">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" th:href="@{/admin}">Welcome Admin</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/products}">Product</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/users}">user</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/categories}">category</a>

</li>

</ul>

</div>

</nav>

<div class="contaioner">

<div class="row">

<div class="col-md-6 container justify-center card">

<h1>Add New Category</h1>

<div class="cart-body">

<form th:action="@{/admin/categories}" th:object="${category}" method="post">

<input type="hidden" name="id" th:field="\*{id}">

<div class="form-group-row">

<label>category Name</label>

<input type="text" name="name" id="name" th:field="\*{name}" placeholder="enter category name"><br>

</div>

<div class="box-footer">

<button type="submit" class="btn btn-primary">Submit</button>

</div>

</form>

</div>

</div>

</div>

</div>

</html>

25) <!DOCTYPE html>

<html xmlns:th="http://www.thymeLeaf.org">

<head>

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

<title>Admin Shoes Portal</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css"

integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

</head>

<body>

<nav class="navbar navbar-expand-md bg-dark navbar-dark">

<!-- Brand -->

<a class="navbar-brand" href="#">Admin Shoes Portal</a>

<!-- Toggler/collapsibe Button -->

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">

<span class="navbar-toggler-icon"></span>

</button>

<!-- Navbar links -->

<div class="collapse navbar-collapse" id="collapsibleNavbar">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" th:href="@{/admin}">Welcome Admin</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/products}">Product</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/users}">user</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/categories}">category</a>

</li>

</ul>

</div>

</nav>

<div class="container">

<div class="row">

<div class="col-lg-6 col-md-6 col-sm-6 container justify-content-center card">

<h1>Add New Product</h1>

<div class="card-body">

<form th:action="@{/admin/products}" th:object="${product}" method="post"

enctype="multipart/form-data">

<input type="hidden" name="id" th:field="\*{id}">

<div class="form-group row">

<label class="col-form-label col-sm-4">Product Name</label>

<div class="col-sm-8">

<input type="text" name="name" th:field="\*{name}" placeholder="enter product name"><br>

</div>

</div>

<div class="form-group row">

<label class="col-form-label col-sm-4" for="category"> Category</label>

<div class="col-sm-8">

<select th:field="\*{category}" class="form-control" id="category" name="category"

required>Select

<option th:each="category :${categories}" th:value="${category.id}"

th:text="${category.name}"></option>

</select>

</div>

</div>

<div class="form-group row">

<label class="col-form-label col-sm-4">Price </label>

<div class="col-sm-8">

<input type="number" name="price" th:field="\*{price}"

placeholder="enter product price"><br>

</div>

</div>

<div class="form-group row ">

<label class="col-form-label col-sm-4" for="productImage"> Product Image</label>

<div class="col-sm-8 custom-file">

<input type="file" class="custom-file-input" name="productImage" accept="image/jpeg, image/png" id="productImage" />

<label class="custom-file-label" for="productImage">Choose file</label>

</div>

</div>

<div class="text-center p-3">

<button type="submit" class="btn btn-primary">Submit</button>

</div>

</form>

</div>

</div>

</div>

</div>

</body>

</html>

26)

<!DOCTYPE html>

<html lang="en" xmlns:th="http://www.thymeleaf.org">

<head>

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

<title>Admin Shoes Portal</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"

integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

</head>

<body>

<nav class="navbar navbar-expand-md bg-dark navbar-dark">

<!-- Brand -->

<a class="navbar-brand" href="#">Admin Shoes Portal</a>

<!-- Toggler/collapsibe Button -->

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">

<span class="navbar-toggler-icon"></span>

</button>

<!-- Navbar links -->

<div class="collapse navbar-collapse" id="collapsibleNavbar">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" th:href="@{/admin}">Welcome Admin</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/products}">Product</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/users}">user</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/categories}">category</a>

</li>

</ul>

</div>

</nav>

<div class="contaioner">

<div class="row">

<div class="col-lg-6 col-md-6 col-sm-6 container justify-content-center card">

<h1 class="text-center">Update Category</h1>

<div class="cart-body">

<form th:action="@{/admin/categories/update/{id} {id=${category.id}}}" th:object="${category}"

method="post">

<input type="hidden" name="id" th:field="\*{id}" class="form-control">

<div class="form-group">

<label>category Name</label>

<input type="text" name="name" th:field="\*{name}" class="form-control"

placeholder="enter category name" />

</div>

<div class="box-footer">

<button type="submit" class="btn btn-primary">Submit</button>

</div>

</form>

</div>

</div>

</div>

</div>

</html>

27)

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

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

<title>Admin Shoe Portal</title>

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"

integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"

crossorigin="anonymous">

</head>

<body>

<nav class="navbar navbar-expand-md bg-dark navbar-dark">

<!-- Brand -->

<a class="navbar-brand" href="#">Admin Shoes Portal</a>

<!-- Toggler/collapsibe Button -->

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">

<span class="navbar-toggler-icon"></span>

</button>

<!-- Navbar links -->

<div class="collapse navbar-collapse" id="collapsibleNavbar">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" th:href="@{/admin}">Welcome Admin</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/products}">Product</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/users}">user</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/categories}">category</a>

</li>

</ul>

</div>

</nav>

<br>

<br>

<div class = "container">

<div class = "row">

<div class ="col-lg-6 col-md-6 col-sm-6 container justify-content-center card">

<h1 class = "text-center"> Update Product </h1>

<div class = "card-body">

<form th:action="@{/admin/products/update/{id} (id=${product.id})}" th:object = "${product}" method="POST">

<div class ="form-group">

<label> Product Name </label>

<input type = "text"

name = "name"

th:field = "\*{name}"

class = "form-control"

placeholder="Enter product Name"/>

</div>

<div class ="form-group">

<label> Categoty </label>

<input

type = "text"

name = "category"

th:field = "\*{category}"

class = "form-control"

placeholder="Enter category"

/>

</div>

<div class ="form-group">

<label> Price </label>

<input

type = "number"

name = "priice"

th:field = "\*{price}"

class = "form-control"

placeholder="Enter price"

/>

</div>

<div class ="form-group">

<label> Preview </label>

<input

type = "file"

name = "image"

th:field = "\*{imageName}"

class = "form-control"

placeholder="Enter image"

/>

</div>

<div class = "box-footer">

<button type="submit" class = "btn btn-primary">

Submit

</button>

</div>

</form>

</div>

</div>

</div>

</div>

</body>

</html>

28) <!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

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

<title>Welcome to Admin Home Page</title>

<link rel="stylesheet" type="text/css" href="/webjars/bootstrap/css/bootstrap.min.css" />

<script type="text/javascript" src="/webjars/jquery/jquery.min.js"></script>

<script type="text/javascript" src="/webjars/bootstrap/js/bootstrap.min.js"></script>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"

integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.0/css/all.css"

integrity="sha384-lZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+A56Ebc1zFSJ" crossorigin="anonymous">

</head>

<nav class="navbar navbar-expand-md bg-dark navbar-dark">

<!-- Brand -->

<a class="navbar-brand" href="#">Admin Shoes Portal</a>

<!-- Toggler/collapsibe Button -->

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">

<span class="navbar-toggler-icon"></span>

</button>

<!-- Navbar links -->

<div class="collapse navbar-collapse" id="collapsibleNavbar">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" th:href="@{/admin}">Welcome Admin</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/products}">Product</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/users}">user</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/categories}">category</a>

</li>

</ul>

</div>

</nav>

<body>

<div class="Jumbotron text-center">

<h1 class="display-5">Welcome to Admin Shoe portal </h1>

<h5><a th:href="@{/login}">Login</a></h5>

<h5><a th:href="@{/shop}">ShopHome</a></h5>

</div>

<div class="container-fluid">

<div class="row row 6 text-center">

<div class="col-sm-4 pt-3">

<div class="cart" style="background-color: blue;">

<div class="cart-body">

<h4>Categories</h4>

<p class="cart-text">Manage The Categories Section Here.</p>

<a href="#" th:href="@{/admin/categories}" class="cart-link btn btn-primary">Manage</a>

</div>

</div>

</div>

<div class="col-sm-4 pt-3">

<div class="cart" style="background-color: blue;">

<div class="cart-body">

<h4>Products</h4>

<p class="cart-text">Manage The Products Section Here.</p>

<a href="#" th:href="@{/admin/products}" class="cart-link btn btn-primary">Manage</a>

</div>

</div>

</div>

<div class="col-sm-4 pt-3">

<div class="cart" style="background-color: blue;">

<div class="cart-body">

<h4>Users</h4>

<p class="cart-text">Manage The Users Section Here.</p>

<a href="#" th:href="@{/admin/users}" class="cart-link btn btn-primary">Manage</a>

</div>

</div>

</div>

</div>

<div class="Jumbotron text-center">

<img th:src="@{/productImage/WelcomeLogo.jpg}" src="../static/images/logo.png" width="auto" height="500px"

class="d-inline-block align-top" alt="" />

</div>

</div>

</body>

</html>

29) <!doctype html>

<html lang="en" xmlns:th="http://www.thymeleaf.org">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"

integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.0/css/all.css"

integrity="sha384-lZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+A56Ebc1zFSJ" crossorigin="anonymous">

<title>Login</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-light bg-light" >

<div class="container-fluid">

<a class="navbar-brand" href="#">

<img th:src="@{/images/logo.png}" src="../static/images/logo.png" width="auto" height="40" class="d-inline-block align-top" alt="" style="width: auto;height: 500%;"/>

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav mr-auto"></ul>

<ul class="navbar-nav">

<li class="nav-item active">

<a class="nav-link" th:href="@{/}" href="#">home</a>

</li>

<li class="nav-item active">

<a class="nav-link" th:href="@{/login}" href="#">login</a>

</li>

<li class="nav-item active">

<a class="nav-link" th:href="@{/shop}" href="#">shop</a>

</li>

<li class="nav-item active">

<a class="nav-link" href="#">Sign Up</a>

</li>

</ul>

</div>

</div>

</nav>

<div class="container">

<div class="row">

<div class="col-sm-6">

<h2>Login Page</h2>

<p>Please fill out login Details</p>

<form th:action="@{/shop}" method="post">

<div class="form-group">

<label for="username">Username</label>

<input type="username" name="username" id="username" placeholder="Your username" class="form-control form-control-lg">

</div>

<div class="form-group">

<label for="password">Password</label>

<input type="password" class="form-control form-control-lg" placeholder="Password" name="password" id="password">

</div>

<div th:if="${param.error}" class="alert alert-error text-danger">

Invalid username or password.

</div>

<input type="submit" value="Login" class="btn btn-primary btn-block">

<br>

<span >Don't have an account <a class="linkControl" href="./register.html"

th:href="@{/register}">Register here</a></span>

<br>

<a href="" th:href="@{/forgotpassword}">Forgot password ?</a>

</form>

</div>

</div>

</div>

</body>

</html>

30)

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

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

<title>Admin Shoes Portal</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"

integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

</head>

<body>

<nav class="navbar navbar-expand-md bg-dark navbar-dark">

<!-- Brand -->

<a class="navbar-brand" href="#">Admin Shoes Portal</a>

<!-- Toggler/collapsibe Button -->

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">

<span class="navbar-toggler-icon"></span>

</button>

<!-- Navbar links -->

<div class="collapse navbar-collapse" id="collapsibleNavbar">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" th:href="@{/admin}">Welcome Admin</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/products}">Product</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/users}">user</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/categories}">category</a>

</li>

</ul>

</div>

</nav>

<div class="container">

<div class="row">

<h1> List Products </h1>

</div>

<div class="row">

<div class="col-lg-3">

<a th:href="@{/admin/products/new}" class="btn btn-primary btn-sm mb-3"> Add Product</a>

</div>

</div>

<table class="table table-striped table-bordered">

<thead class="table-dark">

<tr>

<th> SN</th>

<th> Product Name</th>

<th> Category Name</th>

<th> Preview </th>

<th> Actions </th>

</tr>

</thead>

<tbody>

<tr th:each="product: ${products}">

<th scope="row" th:text="${product.id}"></th>

<td th:text="${product.name}"></td>

<td th:text="${product.category.name}"></td>

<td><img src="" height="100px" width="100px" th:src="@{${'/productImage/' + product.imageName}}"></td>

<td>

<a th:href="@{/admin/products/edit/{id}(id=${product.id})}" class="btn btn-primary">Update</a>

<a th:href="@{/admin/products/delete/{id}(id=${product.id})}" class="btn btn-danger">Delete</a>

</td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

31) <!doctype html>

<html lang="en" xmlns:th="http://www.thymeleaf.org">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"

integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.0/css/all.css"

integrity="sha384-lZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+A56Ebc1zFSJ" crossorigin="anonymous">

<title>Document</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<div class="container-fluid">

<a class="navbar-brand" href="#">

<img th:src="@{/images/logo.png}" src="../static/images/logo.png" width="auto" height="40"

class="d-inline-block align-top" alt="" />

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"

aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav mr-auto"></ul>

<ul class="navbar-nav">

<li class="nav-item active">

<a class="nav-link" th:href="@{/}" href="#">home</a>

</li>

<li class="nav-item active">

<a class="nav-link" th:href="@{/shop}" href="#">shop</a>

</li>

<li class="nav-item active">

<a class="nav-link" href="#">Sign Up</a>

</li>

</ul>

</div>

</div>

</nav>

<div class="container">

<div class="col-sm-6">

<h3 style="margin-top: 10px">Sign Up Now</h3>

<p>Please fill out this to register</p>

<form th:action="@{/shop}" th:object="${user}" method="get">

<div class="form-group">

<label for="username">Username</label>

<input type="text" name="username" id="username" required placeholder="Your username"

class="form-control form-control-lg">

</div>

<div class="form-group">

<label for="password">Password</label>

<input type="password" class="form-control form-control-lg" required placeholder="Password"

name="password" id="password">

</div>

<div class="form-group">

<label for="email">Email address</label>

<input type="email" class="form-control form-control-lg" required minlength="6" placeholder="Email"

name="email" id="email" aria-describedby="emailHelp">

</div>

<div class="form-group">

<label for="address">Adress</label>

<input type="text" name="address" id="address" required placeholder="Your address"

class="form-control form-control-lg">

</div>

<input type="submit" value="Register" class="btn btn-primary btn-block">

</form>

<span style="margin-top: 10px">Already have an account <a class="linkControl" href="./login.html"

th:href="@{/login1}">Login here</a></span>

</div>

</div>

</body>

</html>

32)

<!doctype html>

<html lang="en" xmlns:th="http://www.thymeleaf.org"

xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"

integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.0/css/all.css"

integrity="sha384-lZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+A56Ebc1zFSJ" crossorigin="anonymous">

<title>ShopHome</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav mr-auto"></ul>

<ul class="navbar-nav">

<li class="nav-item active">

<a class="nav-link" th:href="@{/}" href="#">home</a>

</li>

<li class="nav-item active">

<a class="nav-link" th:href="@{/shop}" href="#">shop</a>

</li>

<li class="nav-item active">

<a class="nav-link" sec:authorize="isAnonymous()" th:href="@{/login}" href="#">login</a>

</li>

<li class="nav-item active">

<a class="nav-link" sec:authorize="isAuthenticated()" th:href="@{/logout}" href="#">logout</a>

</li>

<li class="nav-item active">

<a class="nav-link" th:href="@{/cart}" href="#">cart</a>

</li>

<li class="nav-item active">

<span class="nav-link" th:text="${cartCount}">0</span>

</li>

</ul>

</div>

</div>

</nav>

<div class="container" style="margin-top: 30px">

<div class="row">

<div class="col-sm-3">

<h4>Categories</h4>

<ul class="list-group">

<a th:href="@{/shop}">

<li class="list-group-item">All Products</li>

</a>

<a href="" th:each="category, iStat : ${categories}"

th:href="@{/shop/category/{id}(id=${category.id})}">

<li class="list-group-item" th:text="${category.name}"></li>

</a>

</ul>

</div>

<div class="col-sm-9">

<div class="card" style="margin-top: 20px" th:each="product: ${products}">

<div class="row no-gutters">

<div class="col-sm-5 d-flex justify-content-center">

<img class="" height="150px" width="150px"

th:src="@{${'/productImage/' + product.imageName}}" th:alt="${product.name}">

</div>

<div class="col-sm-7 d-flex justify-content-center">

<div class="card-body">

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

<h4>₹ <span th:text="${product.price}"></span></h4>

<a href="#" th:href="@{/shop/viewproduct/{id}(id=${product.id})}"

class="btn btn-primary">View Product</a>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

</html>

33) <!DOCTYPE html>

<html lang="en" xmlns:th="http://www.thymeleaf.org">

<head>

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

<title>Admin Shoes Portal</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"

integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

</head>

<body>

<nav class="navbar navbar-expand-md bg-dark navbar-dark">

<!-- Brand -->

<a class="navbar-brand" href="#">Admin Shoes Portal</a>

<!-- Toggler/collapsibe Button -->

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">

<span class="navbar-toggler-icon"></span>

</button>

<!-- Navbar links -->

<div class="collapse navbar-collapse" id="collapsibleNavbar">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" th:href="@{/admin}">Welcome Admin</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/products}">Product</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/users}">user</a>

</li>

<li class="nav-item">

<a class="nav-link" th:href="@{/admin/categories}">category</a>

</li>

</ul>

</div>

</nav>

<div class="container">

<div class="row">

<h1> List Users </h1>

</div>

<table class="table table-striped table-bordered">

<thead class="table-dark">

<tr>

<th>SN</th>

<th> UserName</th>

<th> Email</th>

<th> Addess</th>

<th> Action </th>

</tr>

</thead>

<tbody>

<tr th:each="user :${users}">

//<th scope="row" th:text="${user.id}"></th>

<td th:text="${user.name}"></td>

<td th:text="${user.email}"></td>

<td th:text="${user.address}"></td>

<td>

<a th:href="@{/admin/users/delete/{id}(id=${user.id})}" class="btn btn-danger">Delete</a>

</td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

34)

<!doctype html>

<html lang="en" xmlns:th="http://www.thymeleaf.org"

xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"

integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.0/css/all.css"

integrity="sha384-lZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+A56Ebc1zFSJ" crossorigin="anonymous">

<title>Document</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<div class="container-fluid">

<a class="navbar-brand" href="#">

<img th:src="@{/images/logo.png}" src="../static/images/logo.png" width="auto" height="40"

class="d-inline-block align-top" alt="" />

</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"

aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav mr-auto"></ul>

<ul class="navbar-nav">

<li class="nav-item active">

<a class="nav-link" th:href="@{/}" href="#">home</a>

</li>

<li class="nav-item active">

<a class="nav-link" th:href="@{/shop}" href="#">shop</a>

</li>

<li class="nav-item active">

<a class="nav-link" sec:authorize="isAnonymous()" th:href="@{/login}" href="#">login</a>

</li>

<li class="nav-item active">

<a class="nav-link" sec:authorize="isAuthenticated()" th:href="@{/logout}" href="#">logout</a>

</li>

<li class="nav-item active">

<a class="nav-link" th:href="@{/cart}" href="#">cart</a>

</li>

<li class="nav-item active">

<span class="nav-link" th:text="${cartCount}"></span>

</li>

</ul>

</div>

</div>

</nav>

<div class="container mt-5">

<div class="row">

<div class="col-sm-5 d-flex justify-content-center">

<img class="mt-5" height="300px" width="300px" src=""

th:src="@{${'/productImage/' + product.imageName}}" alt="">

</div>

<div class="col-sm-7 mt-5">

<h2 th:text="${product.name}"></h2>

<p th:text="${product.category.name}"></p>

<h2>₹ <span th:text="${product.price}"></span></h2>

<a href="" th:href="@{/addToCart/{id}(id = ${product.id})}" class="btn btn-primary">Add to cart</a>

</div>

</div>

</div>

</body>

</html>

35) <!DOCTYPE html>

<html lang="en" xmlns:th="http://www.thymeleaf.org">

<head>

<meta charset="UTF-8">

<title>Registration and Login System</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" th:href="@{/home}">Registration and Login System</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse"

data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"

aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav mr-auto"></ul>

<ul class="navbar-nav">

<li class="nav-item active">

<a class="nav-link" th:href="@{/}" href="#">home</a>

</li>

<li class="nav-item active">

<a class="nav-link" th:href="@{/admin}" href="#">Admin</a>

</li>

<li class="nav-item active">

<a class="nav-link" th:href="@{/shop}" href="#">shop</a>

</li>

<li class="nav-item active">

<a class="nav-link" href="#">Sign Up</a>

</li>

</ul>

</div>

</div>

</nav>

<br />

<div class="container">

<div class="row 10 Jumbotron text-center" style="background-color:lightgreen;">

<h1 class="text-center"> Welcome to Home Page</h1>

<h5><a th:href="@{/login}">Login</a></h5>

<h5><a th:href="@{/register}">Register</a></h5>

</div>

<div class=" row 6 text-center">

<img th:src="@{/productImage/WelcomeLogo.jpg}" src="../static/images/logo.png" width="auto" height="500px"

class="d-inline-block align-top" alt="" />

</div>

</div>

</body>

</html>

36) server.port=1053

spring.datasource.url=jdbc:mysql://localhost:3306/myshoe

spring.datasource.username=root

spring.datasource.password=manager

spring.jpa.properties.hibernate.format\_sql=true

server.error.include-binding-errors=always

spring.jpa.show-sql=true

spring.jpa.hibernate.ddl-auto=update

spring.servlet.multipart.enabled=true

spring.servlet.multipart.file-size-threshold=200MB

spring.servlet.multipart.max-file-size=6000MB

spring.servlet.multipart.max-request-size=215MB

server.error.include-exception=true

server.error.include-stacktrace=always

spring.main.allow-bean-definition-overriding=true

spring.main.allow-circular-references=true

spring.security.user.name=admin

spring.security.user.password=chhaya



