package com.project.DuAnTotNghiep.security;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.authentication.AuthenticationManager;

import org.springframework.security.config.annotation.authentication.configuration.AuthenticationConfiguration;

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.WebSecurityCustomizer;

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

import org.springframework.security.crypto.password.PasswordEncoder;

import org.springframework.security.web.SecurityFilterChain;

import org.springframework.security.web.authentication.AuthenticationSuccessHandler;

import org.springframework.security.web.authentication.SavedRequestAwareAuthenticationSuccessHandler;

*@Configuration*

*@EnableWebSecurity*

public class WebSecurityConfig {

*@Bean*

public PasswordEncoder passwordEncoder() {

return new BCryptPasswordEncoder();

}

*@Configuration*

public static class AppConfiguration {

*@Bean*

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http

.csrf().disable()

.authorizeRequests()

// ✅ 1. Mở toàn bộ tài nguyên tĩnh (frontend)

.antMatchers(

"/css/\*\*", "/js/\*\*", "/images/\*\*", "/vendor/\*\*",

"/fonts/\*\*", "/plugins/\*\*", "/static/\*\*",

"/webjars/\*\*", "/favicon.ico", "/error"

).permitAll()

// ✅ 2. Guest — xem giao diện cơ bản, không cần đăng nhập

.antMatchers(

"/", "/home/\*\*", "/getproduct/\*\*", "/getabout/\*\*",

"/getcontact/\*\*", "/user-login", "/register",

"/forgot-pass/\*\*", "/verify-email/\*\*"

).permitAll()

// ✅ 3. User — có quyền đặt hàng, thanh toán, đánh giá,...

.antMatchers(

"/shoping-cart/\*\*", "/checkout/\*\*", "/orders/\*\*",

"/profile/\*\*", "/payment/\*\*", "/comment/\*\*", "/discount/\*\*"

).hasAnyRole("USER", "VENDOR", "ADMIN")

// ✅ 4. Vendor — có thêm quyền quản lý Shop

.antMatchers(

"/vendor/\*\*", "/shop/\*\*", "/order-management/\*\*",

"/promotion/\*\*", "/revenue/\*\*"

).hasAnyRole("VENDOR", "ADMIN")

// ✅ 5. Admin — quản lý toàn hệ thống

.antMatchers(

"/admin/\*\*", "/management/\*\*", "/system/\*\*"

).hasRole("ADMIN")

// ✅ 6. Mặc định còn lại cho phép truy cập (để frontend load tự do)

.anyRequest().permitAll()

.and()

// ✅ Cấu hình trang login

.formLogin()

.loginPage("/user-login")

.loginProcessingUrl("/user\_login")

.usernameParameter("email")

.defaultSuccessUrl("/", true)

.successHandler(successHandler())

.permitAll()

.and()

// ✅ Logout

.logout()

.logoutUrl("/user\_logout")

.logoutSuccessUrl("/")

.permitAll()

.and()

// ✅ Remember me

.rememberMe()

.key("AbcDefgHijklmnOp\_123456789")

.rememberMeParameter("remember-me")

.tokenValiditySeconds(7 \* 24 \* 60 \* 60);

// ✅ Cho phép iframe (h2-console hoặc template nhúng)

http.headers().frameOptions().disable();

return http.build();

}

*@Bean*

public AuthenticationSuccessHandler successHandler() {

SavedRequestAwareAuthenticationSuccessHandler handler =

new SavedRequestAwareAuthenticationSuccessHandler();

handler.setDefaultTargetUrl("/");

return handler;

}

}

*@Bean*

public AuthenticationManager authenticationManager(

AuthenticationConfiguration authConfig) throws Exception {

return authConfig.getAuthenticationManager();

}

// ✅ Cho phép static resource load tự do (backup)

*@Bean*

public WebSecurityCustomizer webSecurityCustomizer() {

return (web) -> web.ignoring().antMatchers(

"/img/\*\*", "/js/\*\*", "/css/\*\*", "/fonts/\*\*", "/plugins/\*\*",

"/vendor/\*\*", "/static/\*\*", "/webjars/\*\*", "/images/\*\*",

"/favicon.ico", "/error"

);

}

}