package com.example.demo;

import javax.sql.DataSource;

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

import org.springframework.context.annotation.Bean;

import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;

import org.springframework.security.config.annotation.web.builders.WebSecurity;

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

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

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

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

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

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

@EnableWebSecurity

public class SpringSecurityUsingJDBC extends WebSecurityConfigurerAdapter{

@Autowired

DataSource ds;

@Override

protected void configure(AuthenticationManagerBuilder auth) throws Exception {

auth.jdbcAuthentication().dataSource(ds).withDefaultSchema()

//.withUser(User.withUsername("Mahesh").password("Mahesh@123").roles("SALESPERSON"))

//.withUser(User.withUsername("Kanika").password("Kanika@123").roles("Clerk"));

.withUser(User.withUsername("Mahesh").password(getPasswordEncoder().encode("Mahesh@123")).roles("SALESPERSON"));

}

@Bean

PasswordEncoder getPasswordEncoder() {

//return NoOpPasswordEncoder.getInstance();

return new BCryptPasswordEncoder();

}

@Override

public void configure(WebSecurity web) throws Exception {

web.ignoring().antMatchers("/h2-console/\*\*");

}

}