**Spring Security Assignment**

1)

Security1Application.java

package com.example.assignment1;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class Security1Application {

    public static void main(String[] args) {

        SpringApplication.run(Security1Application.class, args);

    }

}

MyConfig

package com.example.assignment1.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

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

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

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

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

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

@Configuration

@EnableWebSecurity

public class MyConfig extends WebSecurityConfigurerAdapter {

    @Override

    protected void configure(AuthenticationManagerBuilder auth) throws Exception {

        auth

                .inMemoryAuthentication()

                .withUser("Himanshu Wakade")

                .password(this.passwordEncoder().encode("Himanshu@12345"))

                .roles("ADMIN");

    }

    @Bean

    public PasswordEncoder passwordEncoder() {

        return new BCryptPasswordEncoder(10);

    }

}

HomeController

package com.example.assignment1.controller;

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

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

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

@RestController

@RequestMapping("/api")

public class HomeController {

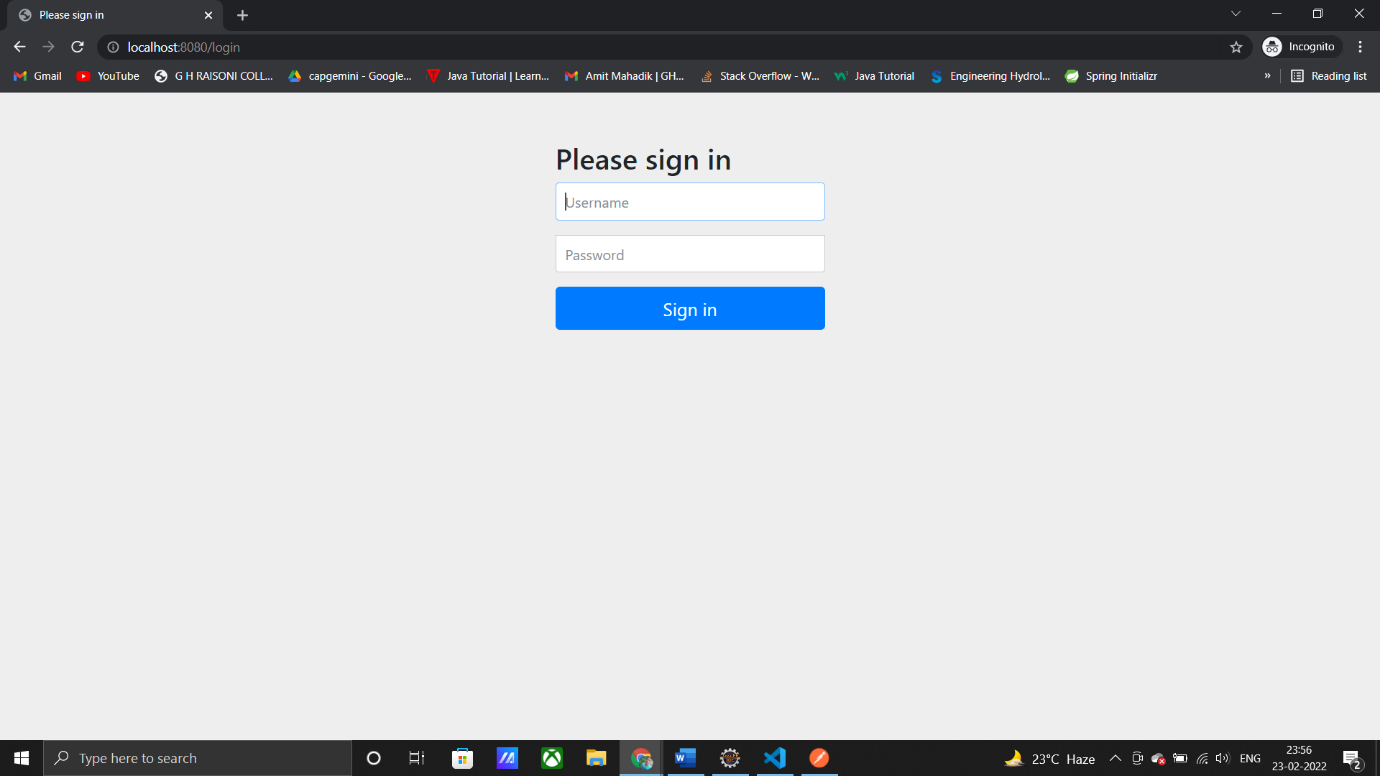
    @GetMapping("/")

    public String helloWorld() {

        return "Hello World!!";

    }

}



2)

package com.springsecurity.assignmentQ2;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class Application {

    public static void main(String[] args) {

        SpringApplication.run(Application.class, args);

    }

}

//

package com.springsecurity.assignmentQ2.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

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.crypto.bcrypt.BCryptPasswordEncoder;

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

@Configuration

@EnableWebSecurity

public class AppConfig extends WebSecurityConfigurerAdapter {

    @Override

    protected void configure(HttpSecurity http) throws Exception {

        http

                .authorizeRequests()

                .antMatchers("/signin").permitAll()

                .anyRequest()

                .authenticated()

                .and()

                .formLogin()

                .loginPage("/signin")

                .loginProcessingUrl("/dologin")

                .defaultSuccessUrl("/admin")

                .and().logout();

    }

    @Override

    protected void configure(AuthenticationManagerBuilder auth) throws Exception {

        auth

                .inMemoryAuthentication()

                .withUser("Himanshu")

                .password(this.passwordEncoder().encode("007"))

                .roles("ADMIN");

        auth

                .inMemoryAuthentication()

                .withUser("suraj")

                .password(this.passwordEncoder().encode("700"))

                .roles("NORMAL");

    }

    @Bean

    public PasswordEncoder passwordEncoder() {

        return new BCryptPasswordEncoder(10);

    }

}

//

package com.springsecurity.assignmentQ2.controller;

import org.springframework.stereotype.Controller;

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

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

import org.springframework.stereotype.Controller;

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

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

@Controller

public class HomeController {

    @GetMapping("/signin")

    public String login() {

        return "login.html";

    }

    @GetMapping("/admin")

    @ResponseBody

    public String hellWorld() {

        return ("<h1>Hello World</h1>");

}

}

//

<!doctype html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">

<head>

    <!-- Required meta tags -->

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->

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

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

    <title>Login Page</title>

</head>

<body>

    <div class="container">

        <div class="row">

            <div class="col-md-6 offset-md-3">

                <div class="card">

                    <div class="card-body">

                        <h3>Login Here</h3>

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

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

                                INVALID USERNAME OR PASSWORD

                            </div>

                            <div class="form-group">

                                <label for="exampleInputEmail1">Username</label>

                                <input name="username" type="text" class="form-control" id="exampleInputEmail1"

                                    aria-describedby="emailHelp" placeholder="Enter username">

                            </div>

                            <div class="form-group">

                                <label for="exampleInputPassword1">Password</label>

                                <input name="password" type="password" class="form-control" id="exampleInputPassword1"

                                    placeholder="Enter password">

                            </div>

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

                        </form>

                    </div>

                </div>

            </div>

        </div>

        <!-- Optional JavaScript -->

        <!-- jQuery first, then Popper.js, then Bootstrap JS -->

        <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"

            integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"

            crossorigin="anonymous"></script>

        <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"

            integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"

            crossorigin="anonymous"></script>

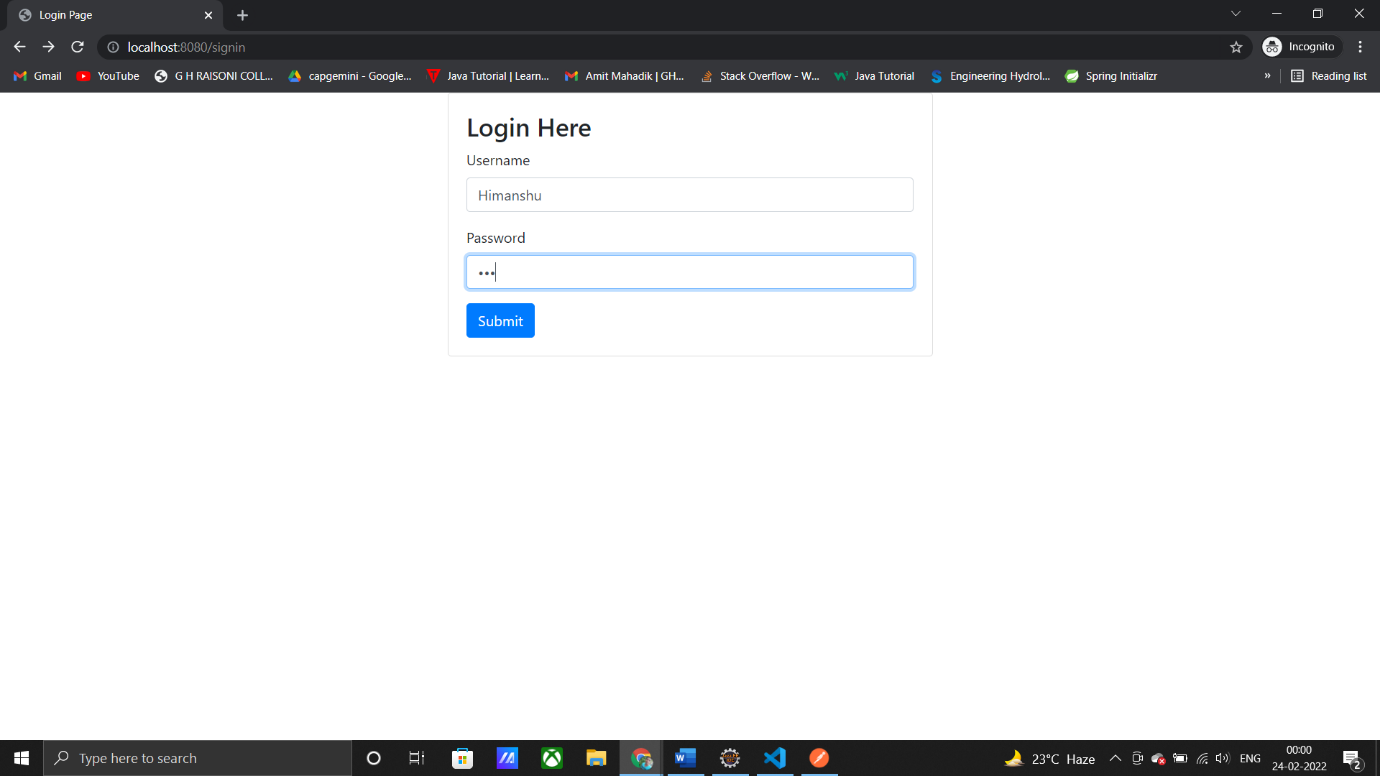
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"

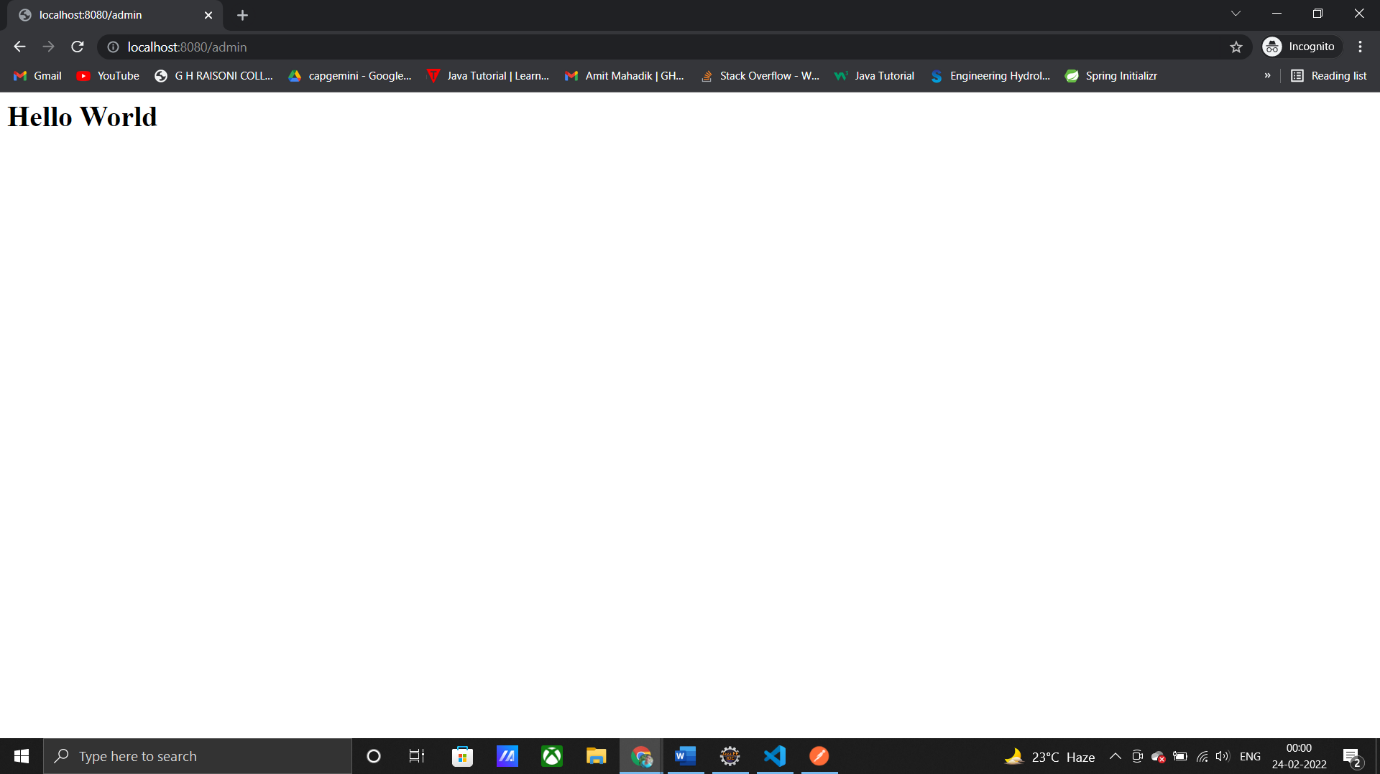
            integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"

            crossorigin="anonymous"></script>

</body>

</html>





3)

//

package com.example.assignment2;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class Security3Application {

    public static void main(String[] args) {

        SpringApplication.run(Security3Application.class, args);

    }

}

//

package com.example.assignment2.config;

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

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

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.crypto.bcrypt.BCryptPasswordEncoder;

import com.example.assignment2.service.CustomUserDetailService;

@Configuration

@EnableWebSecurity

public class MyConfig extends WebSecurityConfigurerAdapter {

    @Autowired

    private CustomUserDetailService customUserDetailService;

    @Override

    protected void configure(HttpSecurity http) throws Exception {

        http

                .authorizeRequests()

                .antMatchers("/signin").permitAll()

                .anyRequest()

                .authenticated()

                .and()

                .formLogin()

                .loginPage("/signin")

                .loginProcessingUrl("/dologin")

                .defaultSuccessUrl("/admin")

                .and().logout();

    }

    @Override

    protected void configure(AuthenticationManagerBuilder auth) throws Exception {

        auth

                .userDetailsService(customUserDetailService)

                .passwordEncoder(passwordEncoder());

    }

    @Bean

    public BCryptPasswordEncoder passwordEncoder() {

        return new BCryptPasswordEncoder(10);

    }

}

//

package com.example.assignment2.controller;

import org.springframework.stereotype.Controller;

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

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

@Controller

public class HomeController {

    @GetMapping("/signin")

    public String login() {

        return "login.html";

    }

    @GetMapping("/admin")

    @ResponseBody

    public String hellWorld() {

        return "Hello World!!";

    }

}

//

package com.example.assignment2.model;

import java.util.Collection;

import java.util.HashSet;

import org.springframework.security.core.GrantedAuthority;

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

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

public class CustomUserDetail implements UserDetails {

    private User user;

    public CustomUserDetail(User user) {

        this.user = user;

    }

    @Override

    public Collection<? extends GrantedAuthority> getAuthorities() {

        HashSet<SimpleGrantedAuthority> set = new HashSet<>();

        set.add(new SimpleGrantedAuthority(this.user.getRole()));

        return set;

    }

    @Override

    public String getPassword() {

        return this.user.getPassword();

    }

    @Override

    public String getUsername() {

        return this.user.getUsername();

    }

    @Override

    public boolean isAccountNonExpired() {

        return true;

    }

    @Override

    public boolean isAccountNonLocked() {

        return true;

    }

    @Override

    public boolean isCredentialsNonExpired() {

        return true;

    }

    @Override

    public boolean isEnabled() {

        return true;

    }

}

//

package com.example.assignment2.model;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class User {

    @Id

    String username;

    String password;

    String email;

    String role;

    public User() {

    }

    public User(String username, String password, String email, String role) {

        this.username = username;

        this.password = password;

        this.email = email;

        this.role = role;

    }

    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 getEmail() {

        return email;

    }

    public void setEmail(String email) {

        this.email = email;

    }

    public String getRole() {

        return role;

    }

    public void setRole(String role) {

        this.role = role;

    }

    @Override

    public String toString() {

        return "User [email=" + email + ", password=" + password + ", role=" + role + ", username=" + username + "]";

    }

}

//

package com.example.assignment2.repository;

import com.example.assignment2.model.User;

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

import org.springframework.stereotype.Repository;

@Repository

public interface UserRepo extends JpaRepository<User, String> {

    public User findByUsername(String username);

}

//

package com.example.assignment2.service;

import com.example.assignment2.model.CustomUserDetail;

import com.example.assignment2.model.User;

import com.example.assignment2.repository.UserRepo;

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

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 UserRepo userRepo;

    @Override

    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

        User user = this.userRepo.findByUsername(username);

        if (user == null) {

            throw new UsernameNotFoundException("INVALID USERNAME!!");

        }

        return new CustomUserDetail(user);

    }

}

//

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

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

spring.jpa.show-sql=true

spring.jpa.hibernate.ddl-auto=update

5)

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DemoApplication {

    public static void main(String[] args) {

        SpringApplication.run(DemoApplication.class, args);

    }

}

//

package com.example.demo.config;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.example.demo.service.CustomerUserDetailService;

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

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

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.AuthenticationException;

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

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

@Configuration

@EnableWebSecurity

public class MyConfig extends WebSecurityConfigurerAdapter {

    static int count = 0;

    @Autowired

    private CustomerUserDetailService customUserDetailService;

    @Override

    protected void configure(HttpSecurity http) throws Exception {

        http

                .authorizeRequests()

                .antMatchers("/signin").permitAll()

                .anyRequest()

                .authenticated()

                .and()

                .formLogin()

                .loginPage("/signin")

                .loginProcessingUrl("/dologin")

                .defaultSuccessUrl("/admin")

                .failureHandler(new AuthenticationFailureHandler() {

                    @Override

                    public void onAuthenticationFailure(HttpServletRequest request, HttpServletResponse response,

                            AuthenticationException exception) throws IOException, ServletException {

                        System.out.println("A user has failed to login. Error: " + exception.getMessage());

                        if (count == 3) {

                        }

                        response.sendRedirect("signin?error");

                    }

                })

                .and()

                .rememberMe().userDetailsService(customUserDetailService)

                .tokenValiditySeconds(3 \* 24 \* 60 \* 60);

    }

    @Override

    protected void configure(AuthenticationManagerBuilder auth) throws Exception {

        auth

                .userDetailsService(customUserDetailService)

                .passwordEncoder(passwordEncoder());

    }

    @Bean

    public BCryptPasswordEncoder passwordEncoder() {

        return new BCryptPasswordEncoder(10);

    }

}

//

package com.example.demo.controller;

import org.springframework.stereotype.Controller;

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

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

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

@Controller

public class HomeController {

    @GetMapping("/signin")

    public String login() {

        return "login.html";

    }

    @GetMapping("/admin")

    @ResponseBody

    public String hellWorld() {

        return "Hello World!!";

    }

    @PostMapping("login-fail")

    public String handleFailedLogin() {

        System.out.println("A User has failed to login");

        return "redirect:/signin?error";

    }

}

//

package com.example.demo.model;

import java.util.Collection;

import java.util.HashSet;

import org.springframework.security.core.GrantedAuthority;

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

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

public class CustomerUserDetail implements UserDetails {

    private User user;

    public CustomerUserDetail(User user) {

        this.user = user;

    }

    public CustomerUserDetail() {

    }

    @Override

    public Collection<? extends GrantedAuthority> getAuthorities() {

        HashSet<SimpleGrantedAuthority> set = new HashSet<>();

        set.add(new SimpleGrantedAuthority(this.user.getRole()));

        return set;

    }

    @Override

    public String getPassword() {

        return this.user.getPassword();

    }

    @Override

    public String getUsername() {

        return this.user.getUsername();

    }

    @Override

    public boolean isAccountNonExpired() {

        return true;

    }

    @Override

    public boolean isAccountNonLocked() {

        return true;

    }

    @Override

    public boolean isCredentialsNonExpired() {

        return true;

    }

    @Override

    public boolean isEnabled() {

        return true;

    }

}

//

package com.example.demo.model;

import javax.persistence.Id;

import org.springframework.boot.autoconfigure.domain.EntityScan;

@EntityScan

public class User {

    @Id

    String username;

    String password;

    String email;

    String role;

    public User() {

    }

    public User(String username, String password, String email, String role) {

        this.username = username;

        this.password = password;

        this.email = email;

        this.role = role;

    }

    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 getEmail() {

        return email;

    }

    public void setEmail(String email) {

        this.email = email;

    }

    public String getRole() {

        return role;

    }

    public void setRole(String role) {

        this.role = role;

    }

    @Override

    public String toString() {

        return "User [email=" + email + ", password=" + password + ", role=" + role + ", username=" + username + "]";

    }

}

//

package com.example.demo.repository;

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

import org.springframework.stereotype.Repository;

import com.example.demo.model.User;

@Repository

public interface UserRepo extends JpaRepository<User, String> {

    public User findByUsername(String username);

}

//

package com.example.demo.service;

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

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;

import com.example.demo.model.CustomerUserDetail;

import com.example.demo.model.User;

import com.example.demo.repository.UserRepo;

@Service

public class CustomerUserDetailService implements UserDetailsService {

    @Autowired

    private UserRepo userRepo;

    @Override

    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

        User user = this.userRepo.findByUsername(username);

        if (user == null) {

            throw new UsernameNotFoundException("INVALID USERNAME!!");

        }

        return new CustomerUserDetail(user);

    }

}

//

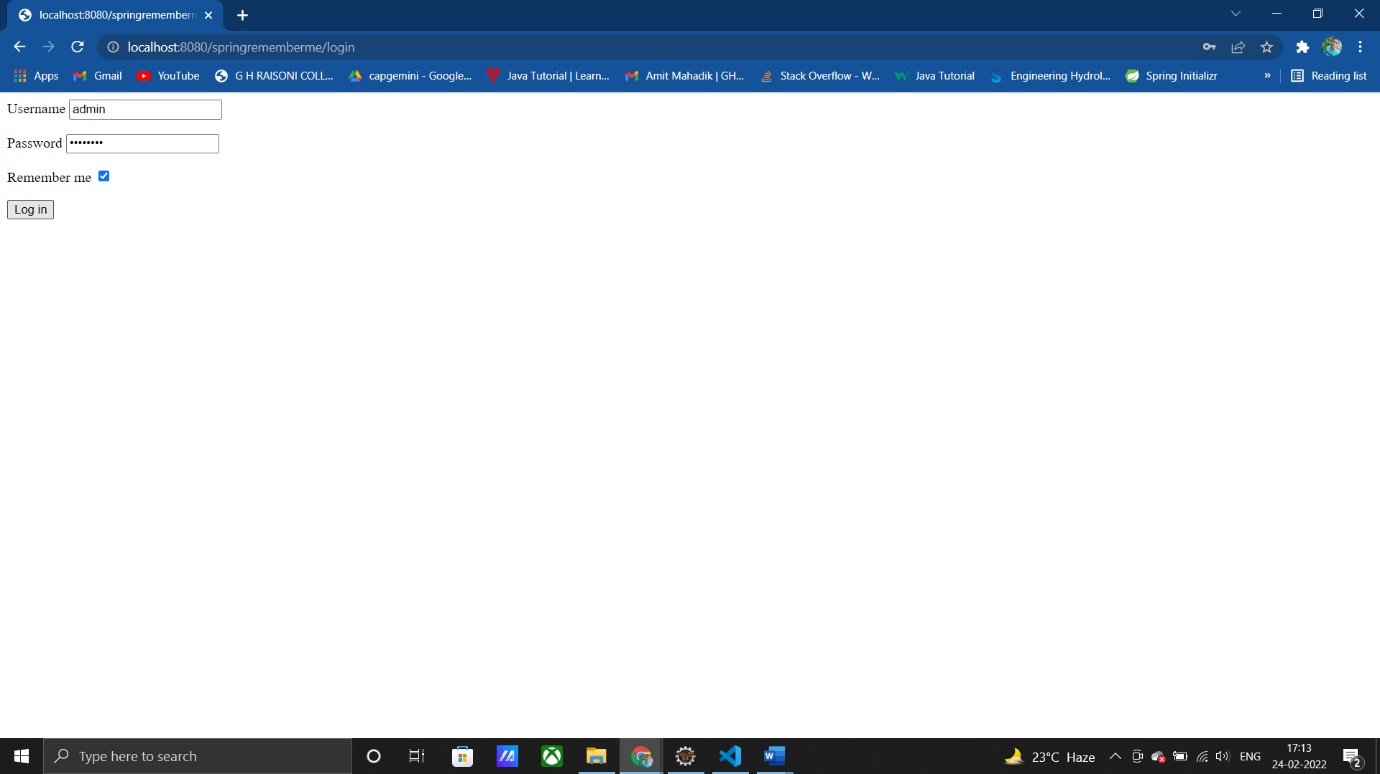
spring.datasource.url=jdbc:mysql://localhost:3306/security5

spring.datasource.username= root

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

spring.jpa.hibernate.ddl-auto=update



6)

package com.example.security6;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class Security6Application {

    public static void main(String[] args) {

        SpringApplication.run(Security6Application.class, args);

    }

}

//

package com.example.security6.config;

import java.io.IOException;

import javax.naming.AuthenticationException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

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

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

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.crypto.bcrypt.BCryptPasswordEncoder;

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

import com.example.security6.service.CustomUserDetailService;

@Configuration

@EnableWebSecurity

public class MyConfig extends WebSecurityConfigurerAdapter {

    static int count = 0;

    @Autowired

    private CustomUserDetailService customUserDetailService;

    @Override

    protected void configure(HttpSecurity http) throws Exception {

        http

                .authorizeRequests()

                .antMatchers("/signin").permitAll()

                .anyRequest()

                .authenticated()

                .and()

                .formLogin()

                .loginPage("/signin")

                .loginProcessingUrl("/dologin")

                .defaultSuccessUrl("/admin")

                .failureHandler(new AuthenticationFailureHandler() {

                     @Override

                    public void onAuthenticationFailure(HttpServletRequest request, HttpServletResponse response,

                            AuthenticationException exception) throws IOException, ServletException {

                        System.out.println("A user has failed to login. Error: " + exception.getMessage());

                        if (count == 3) {

                        }

                        response.sendRedirect("signin?error");

                    }

                    @Override

                    public void onAuthenticationFailure(HttpServletRequest request, HttpServletResponse response,

                            org.springframework.security.core.AuthenticationException exception)

                            throws IOException, ServletException {

                        // TODO Auto-generated method stub

                    }

                })

                .and()

                .rememberMe().userDetailsService(customUserDetailService)

                .tokenValiditySeconds(3 \* 24 \* 60 \* 60);

    }

    @Override

    protected void configure(AuthenticationManagerBuilder auth) throws Exception {

        auth

                .userDetailsService(customUserDetailService)

                .passwordEncoder(passwordEncoder());

    }

    @Bean

    public BCryptPasswordEncoder passwordEncoder() {

        return new BCryptPasswordEncoder(10);

    }

}

//

package com.example.security6.controller;

import org.springframework.stereotype.Controller;

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

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

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

@Controller

public class HomeController {

    @GetMapping("/signin")

    public String login() {

        return "login.html";

    }

    @GetMapping("/admin")

    @ResponseBody

    public String hellWorld() {

        return "Hello World!!";

    }

    @PostMapping("login-fail")

    public String handleFailedLogin() {

        System.out.println("A User has failed to login");

        return "redirect:/signin?error";

    }

}

//

package com.example.security6.model;

import java.util.Collection;

import java.util.HashSet;

import org.springframework.security.core.GrantedAuthority;

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

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

public class CustomUserDetail implements UserDetails {

    private User user;

    public CustomUserDetail(User user) {

        this.user = user;

    }

    public CustomUserDetail() {

    }

    @Override

    public Collection<? extends GrantedAuthority> getAuthorities() {

        HashSet<SimpleGrantedAuthority> set = new HashSet<>();

        set.add(new SimpleGrantedAuthority(this.user.getRole()));

        return set;

    }

    @Override

    public String getPassword() {

        return this.user.getPassword();

    }

    @Override

    public String getUsername() {

        return this.user.getUsername();

    }

    @Override

    public boolean isAccountNonExpired() {

        return true;

    }

    @Override

    public boolean isAccountNonLocked() {

        return true;

    }

    @Override

    public boolean isCredentialsNonExpired() {

        return true;

    }

    @Override

    public boolean isEnabled() {

        return true;

    }

}

//

package com.example.security6.model;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class User {

    @Id

    String username;

    String password;

    String email;

    String role;

    public User() {

    }

    public User(String username, String password, String email, String role) {

        this.username = username;

        this.password = password;

        this.email = email;

        this.role = role;

    }

    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 getEmail() {

        return email;

    }

    public void setEmail(String email) {

        this.email = email;

    }

    public String getRole() {

        return role;

    }

    public void setRole(String role) {

        this.role = role;

    }

    @Override

    public String toString() {

        return "User [email=" + email + ", password=" + password + ", role=" + role + ", username=" + username + "]";

    }

}

//

package com.example.security6.repository;

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

import org.springframework.stereotype.Repository;

import com.example.security6.model.User;

@Repository

public interface UserRepo extends JpaRepository<User, String> {

    public User findByUsername(String username);

}

//

package com.example.security6.service;

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

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;

import com.example.security6.model.CustomUserDetail;

import com.example.security6.model.User;

import com.example.security6.repository.UserRepo;

@Service

public class CustomUserDetailService implements UserDetailsService {

    @Autowired

    private UserRepo userRepo;

    @Override

    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

        User user = this.userRepo.findByUsername(username);

        if (user == null) {

            throw new UsernameNotFoundException("INVALID USERNAME!!");

        }

        return new CustomUserDetail(user);

    }

}