Q1.

User.java

**package** com.springsecurity.model;

**public** **class** User {

String username;

String password;

String email;

**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** User(String username, String password, String email) {

**super**();

**this**.username = username;

**this**.password = password;

**this**.email = email;

}

**public** User() {

**super**();

// **TODO** Auto-generated constructor stub

}

}

UserController.java

**package** com.springsecurity.controller;

**import** java.util.List;

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

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

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

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

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

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

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

**import** com.springsecurity.UserService;

**import** com.springsecurity.model.User;

@RestController

@RequestMapping("/user")

**public** **class** UserController {

@Autowired

**private** UserService service;

@GetMapping("/")

**public** List<User> getAllUsers() {

**return** **this**.service.getAllUsers();

}

@GetMapping("/{username}")

**public** User getUser(@PathVariable("username") String username) {

**return** **this**.service.getUser(username);

}

@PostMapping("/")

**public** User addUser(@RequestBody User user) {

**return** **this**.service.addUser(user);

}

}

UserService.java

**package** com.springsecurity;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.springframework.stereotype.Service;

**import** com.springsecurity.model.User;

@Service

**public** **class** UserService {

List<User> list = **new** ArrayList<User>();

**public** UserService() {

list.add(**new** User("abc", "abc", "abc@gmail.com"));

list.add(**new** User("xyz", "xyz", "xyz@gmail.com"));

}

**public** List<User> getAllUsers(){

**return** **this**.list;

}

**public** User getUser(String username) {

**return** **this**.list.stream().filter((user)->user.getUsername().equals(username)).findAny().orElse(**null**);

}

**public** User addUser(User user) {

**this**.list.add(user);

**return** user;

}

}

MySecurityConfig.java

**package** com.springsecurity;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.springframework.stereotype.Service;

**import** com.springsecurity.model.User;

@Service

**public** **class** UserService {

List<User> list = **new** ArrayList<User>();

**public** UserService() {

list.add(**new** User("abc", "abc", "abc@gmail.com"));

list.add(**new** User("xyz", "xyz", "xyz@gmail.com"));

}

**public** List<User> getAllUsers(){

**return** **this**.list;

}

**public** User getUser(String username) {

**return** **this**.list.stream().filter((user)->user.getUsername().equals(username)).findAny().orElse(**null**);

}

**public** User addUser(User user) {

**this**.list.add(user);

**return** user;

}

}

HomeController.java

**package** com.springsecurity.controller;

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

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

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

@RestController

@RequestMapping("/public")

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

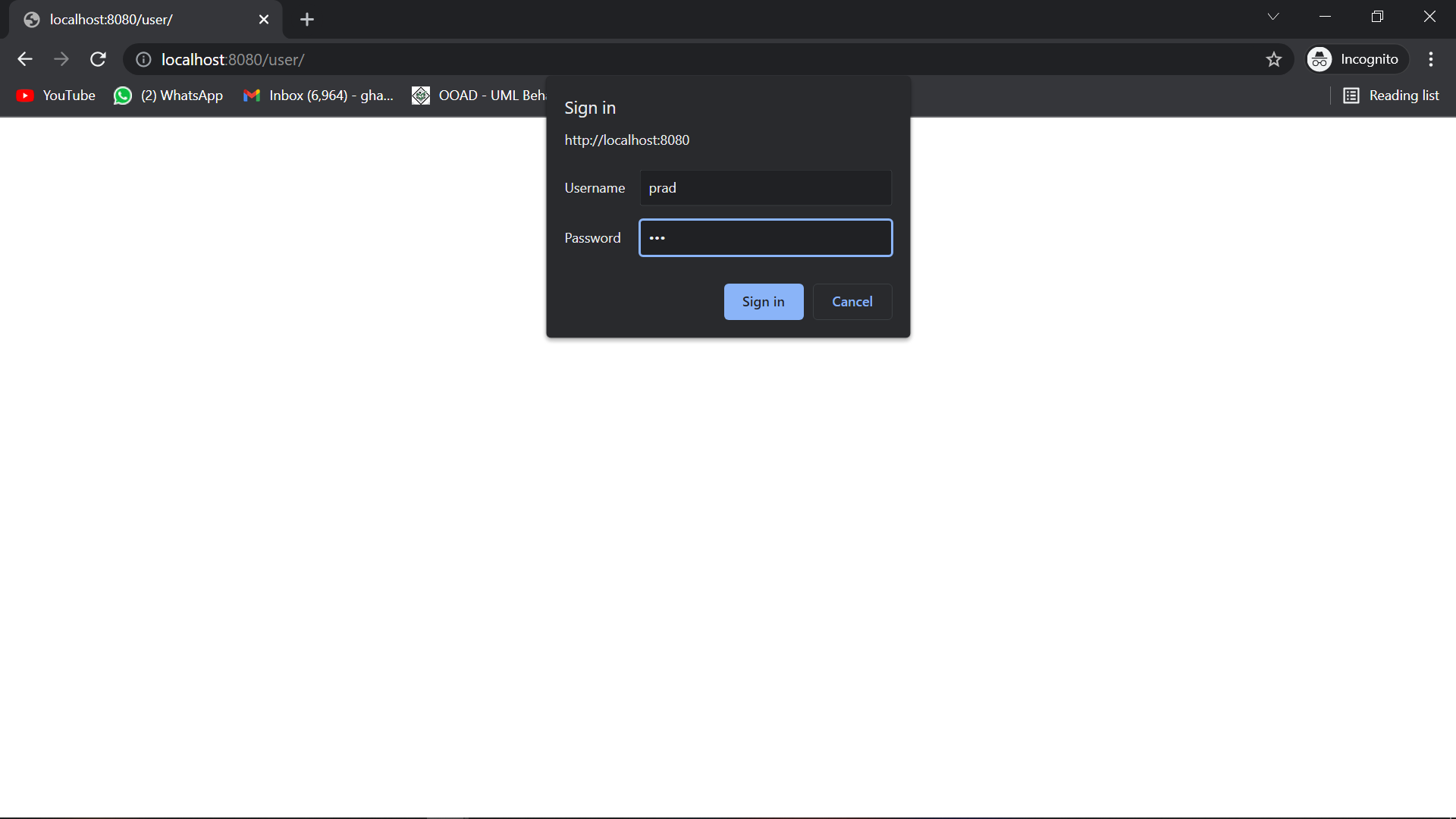
@GetMapping("/home")

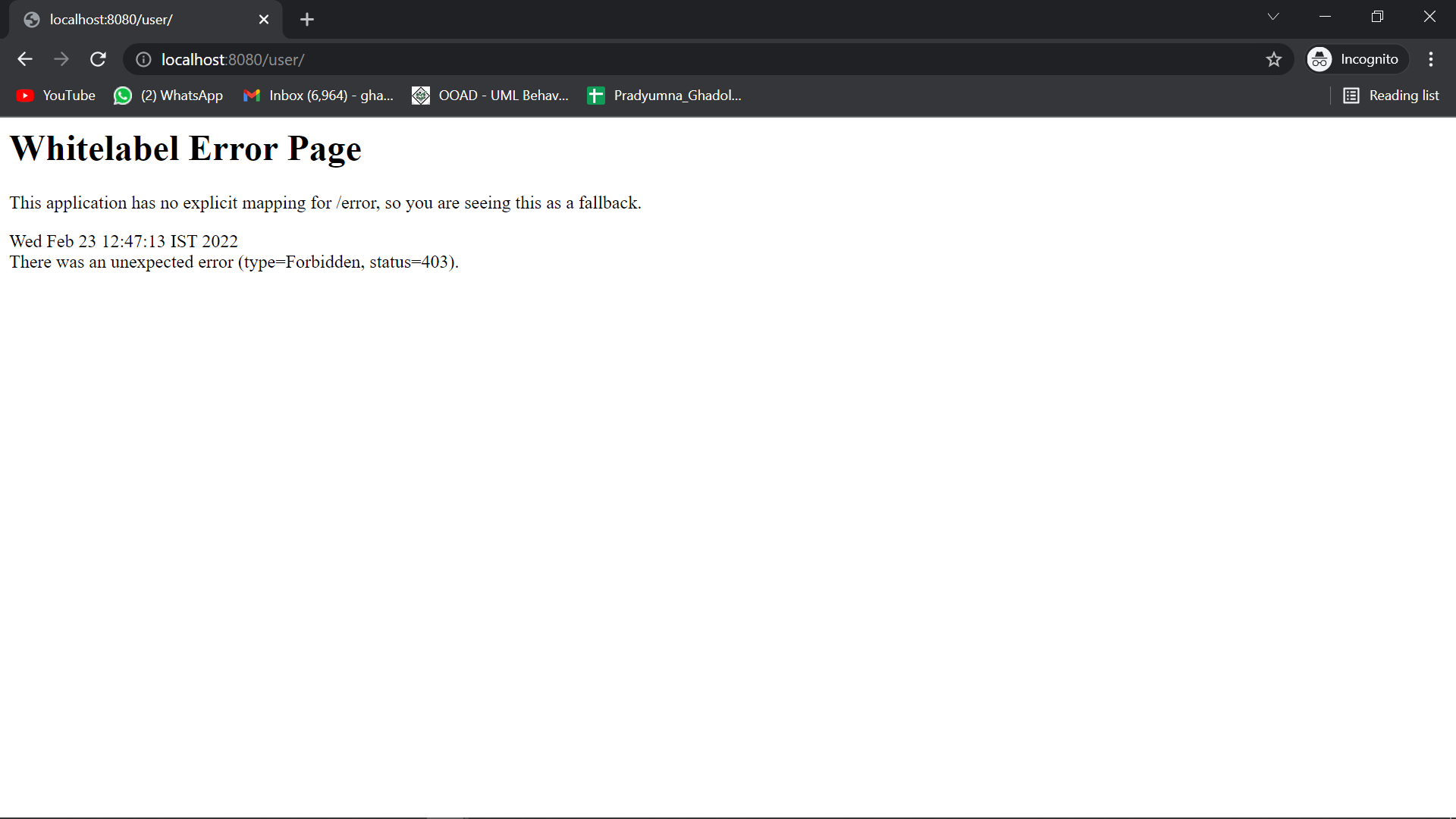
**public** String home() {

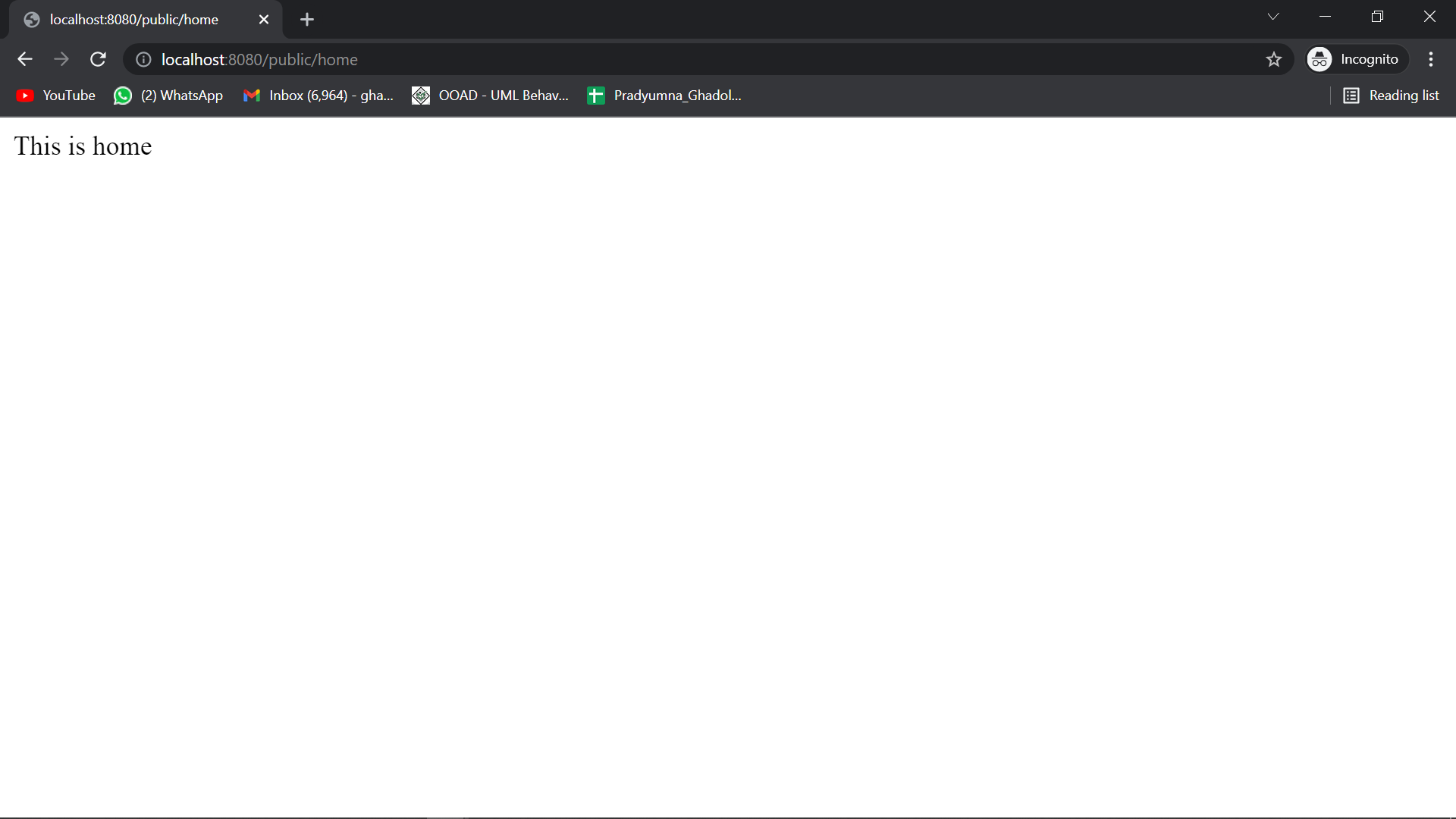
**return** "This is home";

}

}







Q2.

User.java

**package** com.springsecurity.model;

**import** javax.persistence.Entity;

**import** javax.persistence.Id;

@Entity

**public** **class** User {

@Id

String username;

String password;

String email;

String role;

**public** String getRole() {

**return** role;

}

**public** **void** setRole(String role) {

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

**super**();

**this**.username = username;

**this**.password = password;

**this**.email = email;

**this**.role = role;

}

**public** User() {

**super**();

// **TODO** Auto-generated constructor stub

}

}

UserService.java

package com.springsecurity;

import java.util.ArrayList;

import java.util.List;

import org.springframework.stereotype.Service;

import com.springsecurity.model.User;

@Service

public class UserService {

List<User> list = new ArrayList<User>();

public UserService() {

list.add(new User("abc", "abc", "abc@gmail.com","NORMAL"));

list.add(new User("xyz", "xyz", "xyz@gmail.com","ADMIN"));

}

public List<User> getAllUsers(){

return this.list;

}

public User getUser(String username) {

return this.list.stream().filter((user)->user.getUsername().equals(username)).findAny().orElse(null);

}

public User addUser(User user) {

this.list.add(user);

return user;

}

}

UserRepository.java

**package** com.springsecurity.repo;

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

**import** org.springframework.stereotype.Repository;

**import** com.springsecurity.model.User;

@Repository

**public** **interface** UserRepository **extends** JpaRepository<User, String>{

}

MySecurityConfig.java

package com.springsecurity.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;

@Configuration

@EnableWebSecurity

public class MySecurityConfig extends WebSecurityConfigurerAdapter{

@Override

protected void configure(HttpSecurity http) throws Exception {

http.authorizeRequests()

.antMatchers("/public/\*\*").hasRole("NORMAL")

.antMatchers("/user/\*\*").hasRole("ADMIN")

.anyRequest()

.authenticated()

.and()

.httpBasic();

}

@Override

protected void configure(AuthenticationManagerBuilder auth) throws Exception {

auth.inMemoryAuthentication()

.withUser("prad")

.password(this.encoder().encode("123"))

.roles("NORMAL")

.and()

.withUser("rish")

.password(this.encoder().encode("123"))

.roles("ADMIN");

}

@Bean

public BCryptPasswordEncoder encoder() {

return new BCryptPasswordEncoder(10);

}

}

Assignment2SpringSecurityQ1Application.java

**package** com.springsecurity;

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

**import** org.springframework.boot.CommandLineRunner;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

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

**import** com.springsecurity.model.User;

**import** com.springsecurity.repo.UserRepository;

@SpringBootApplication

**public** **class** Assignment2SpringSecurityQ1Application **implements** CommandLineRunner {

@Autowired

**private** UserRepository userRepository;

@Autowired

**private** BCryptPasswordEncoder bCryptPasswordEncoder;

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(Assignment2SpringSecurityQ1Application.**class**, args);

}

@Override

**public** **void** run(String... args) **throws** Exception {

**this**.userRepository.save(**new** User("abc", **this**.bCryptPasswordEncoder.encode("123"), "abc@gmail.com","NORMAL"));

**this**.userRepository.save(**new** User("xyz", **this**.bCryptPasswordEncoder.encode("123"), "xyz@gmail.com","ADMIN"));

}

}

Application.properties

spring.datasource.url=jdbc:mysql://localhost:3306/auth?useSSL=false

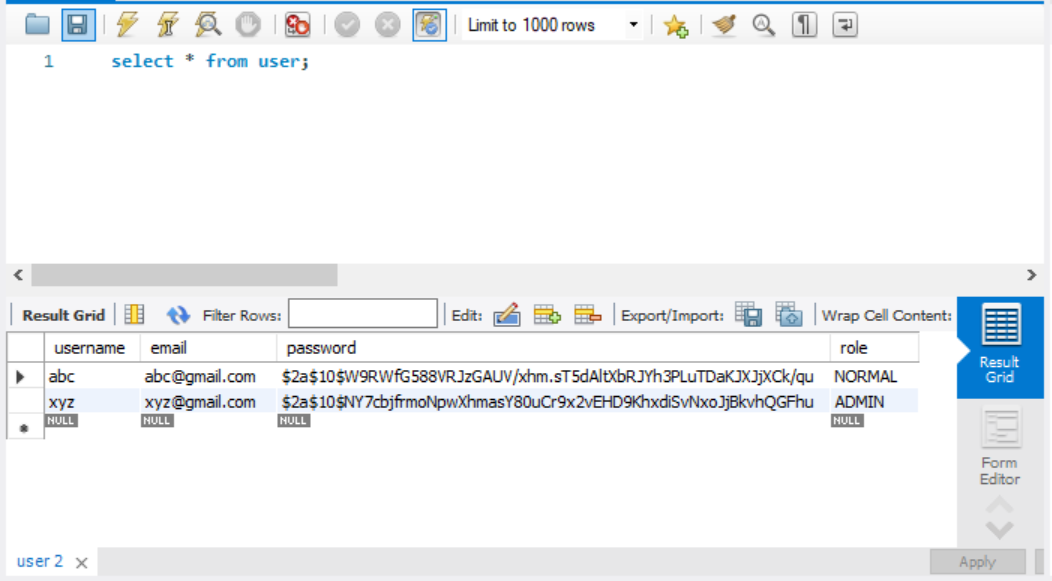
spring.datasource.username=root

spring.datasource.password=Pradumnya@9

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect

spring.jpa.show-sql= true

spring.jpa.hibernate.ddl-auto=update



Q3.

Helloconfig.java

package com.springsecurity.config;

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.HttpSecurity;

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

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

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

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

@EnableWebSecurity

public class HelloConfig extends WebSecurityConfigurerAdapter {

@Autowired

DataSource dataSource;

@Override

protected void configure(AuthenticationManagerBuilder auth) throws Exception {

auth.jdbcAuthentication()

.dataSource(dataSource);

}

@Override

protected void configure(HttpSecurity http) throws Exception {

http.authorizeRequests()

.antMatchers("/admin").hasRole("ADMIN")

.antMatchers("/user").hasRole("USER")

.anyRequest().authenticated()

.and()

.formLogin()

.loginPage("/login")

.permitAll()

.and()

.rememberMe().key("PRADRISHGHAD123456789")

.tokenValiditySeconds(3 \* 20 \* 60 \* 60);

}

@Bean

PasswordEncoder encoder() {

return NoOpPasswordEncoder.getInstance();

}

}

helloResource.java

package com.springsecurity.resource;

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

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

@RestController

public class HelloResource {

@GetMapping("/admin")

public String hello() {

return "Hello World!!";

}

@GetMapping("/user")

public String user() {

return "You cannot access helloworld";

}

}

Logincontroller.java

package com.springsecurity.resource;

import org.springframework.stereotype.Controller;

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

@Controller

public class LoginController {

@GetMapping("/login")

public String login() {

return "login";

}

}

Login.html

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

<head>

<title>Please Login</title>

</head>

<body>

<div>

<form name=*"f"* th:action=*"@{/login}"* method=*"post"*>

<fieldset>

<legend>Please login</legend>

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

username and password

</div>

<div th:if=*"${param.logout}"* class=*"alert alert-success"*>You

have been logged out!

</div>

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

<input type=*"text"* id=*"username"* name=*"username"*>

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

<input type=*"password"* id=*"password"* name=*"password"*>

<p>

<input type = *"checkbox"* name= *"remember-me"* /> Remember me

</p>

<div class=*"form-actions"*>

<button type=*"submit"* class=*"btn"*>Log in</button>

</div>

</fieldset>

</form>

</div>

</body>

</html>

Application.properties

spring.h2.console.enabled=true

spring.sql.init.platform=h2

spring.datasource.url=jdbc:h2:mem:pradumnya

pom.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project xmlns=*"http://maven.apache.org/POM/4.0.0"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.6.3</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<groupId>com.springboot</groupId>

<artifactId>Assignment-SpringSecurity-Q1</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>Assignment-SpringSecurity-Q1</name>

<description>spring security jwt</description>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-jdbc</artifactId>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

