**ASSIGNMENT-1**

1. **SpringsecuritydemoApplication.java**

package com.spring.security.springsecuritydemo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringsecuritydemoApplication {

public static void main(String[] args) {

SpringApplication.run(SpringsecuritydemoApplication.class, args);

}

}

**HelloResource.java**

package com.spring.security.springsecuritydemo;

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

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

@RestController

public class HelloResource {

@GetMapping("/")

public String home()

{

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

}

@GetMapping("/user")

public String user()

{

return("<h1>Welcome User</h1");

}

@GetMapping("/admin")

public String admin()

{

return("<h1>Welcome Admin</h1");

}

}

**securityConfiguration.java**

**package** com.spring.security.springsecuritydemo;

**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** securityConfiguration **extends** WebSecurityConfigurerAdapter {

@Override

**public** **void** configure(AuthenticationManagerBuilder auth) **throws** Exception

{

auth.inMemoryAuthentication()

.withUser("blah")

.password("blah")

.roles("USER")

.and()

.withUser("foo")

.password("foo")

.roles("ADMIN");

}

@Bean

**public** PasswordEncoder getPasswordEncoder() {

**return** ~~NoOpPasswordEncoder~~.~~getInstance~~();

}

@Override

**protected** **void** configure(HttpSecurity http) **throws** Exception {

http.authorizeRequests()

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

.antMatchers("/user").hasAnyRole("ADMIN", "USER")

.antMatchers("/").permitAll()

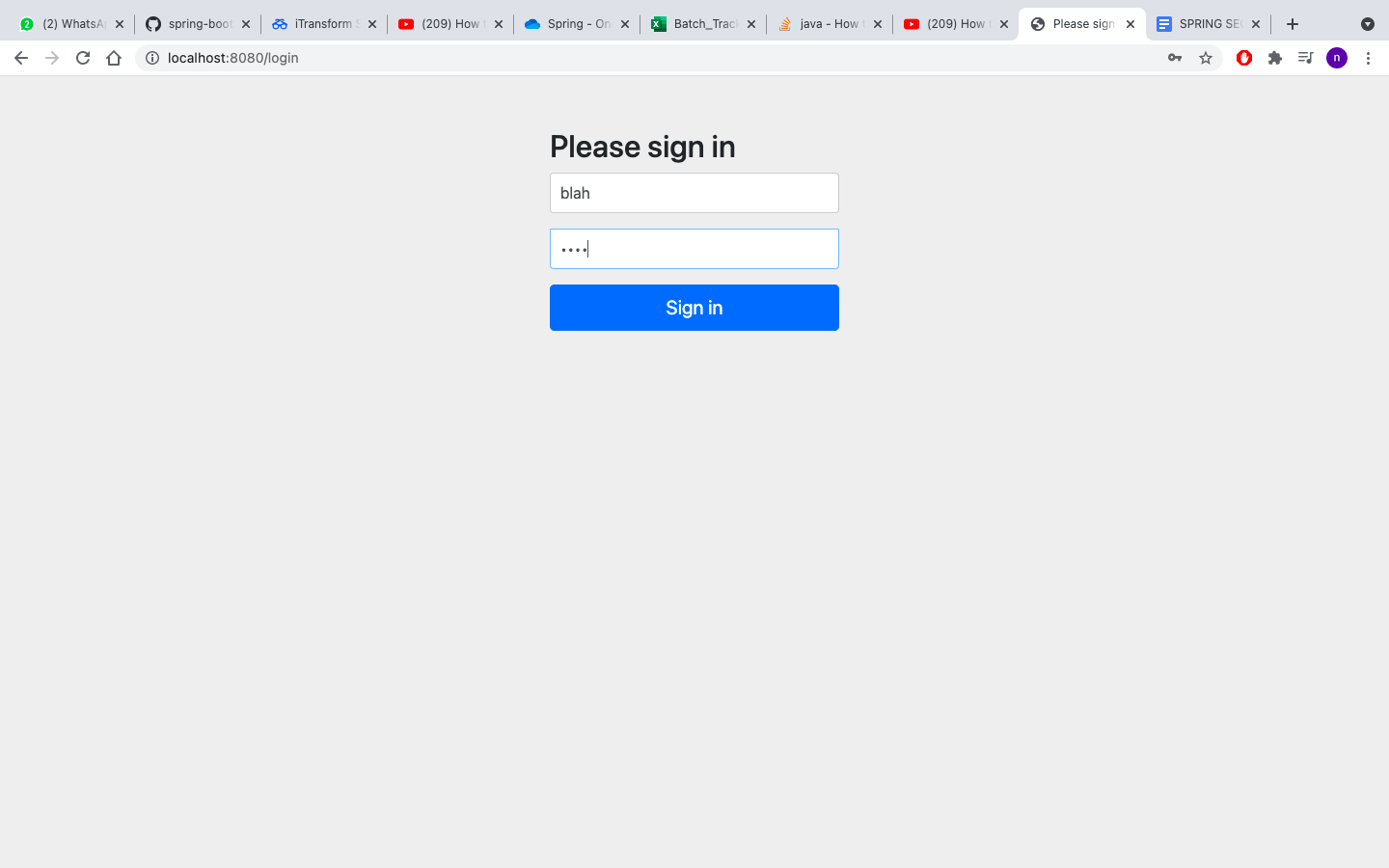
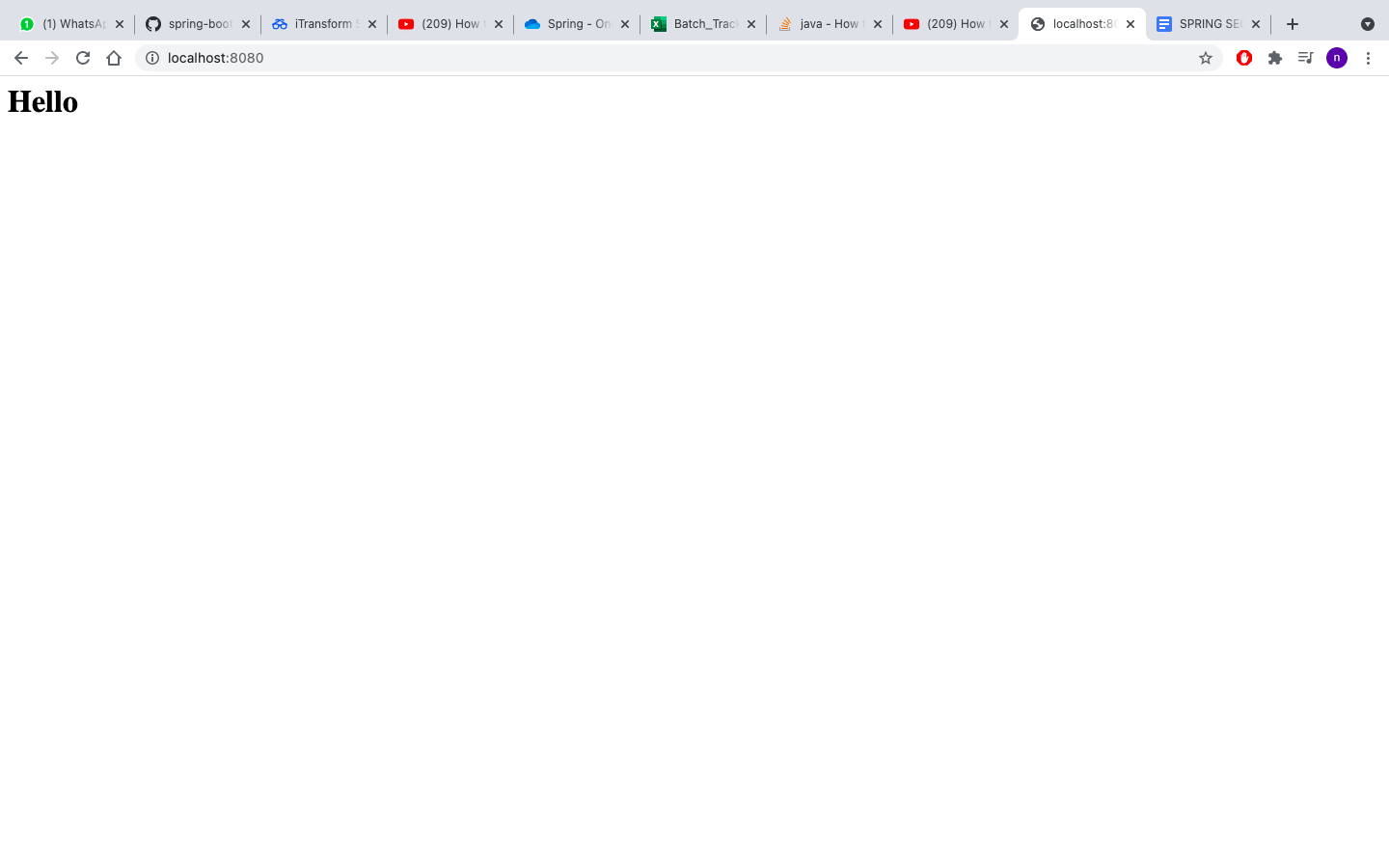
.and().formLogin();

}

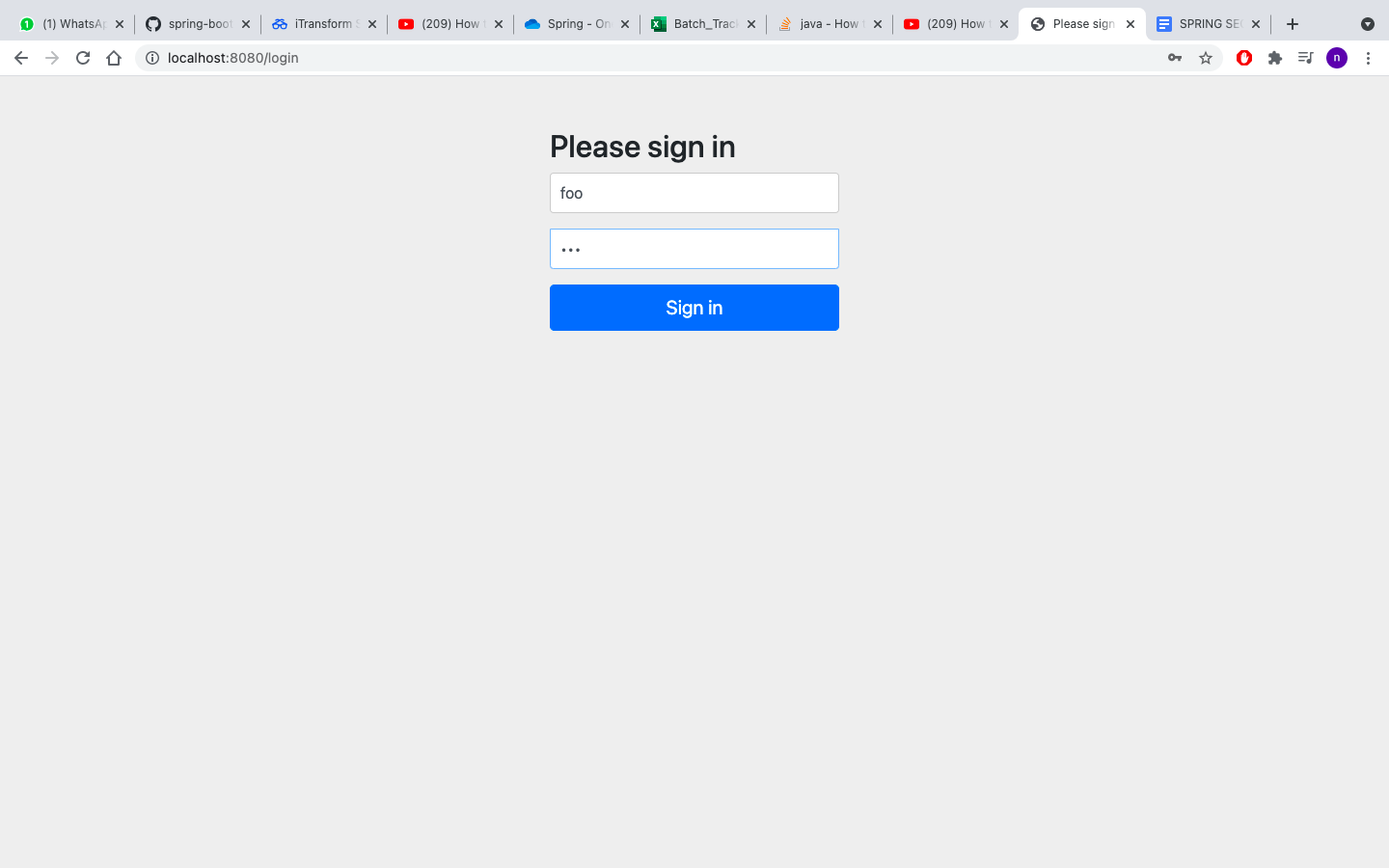
}

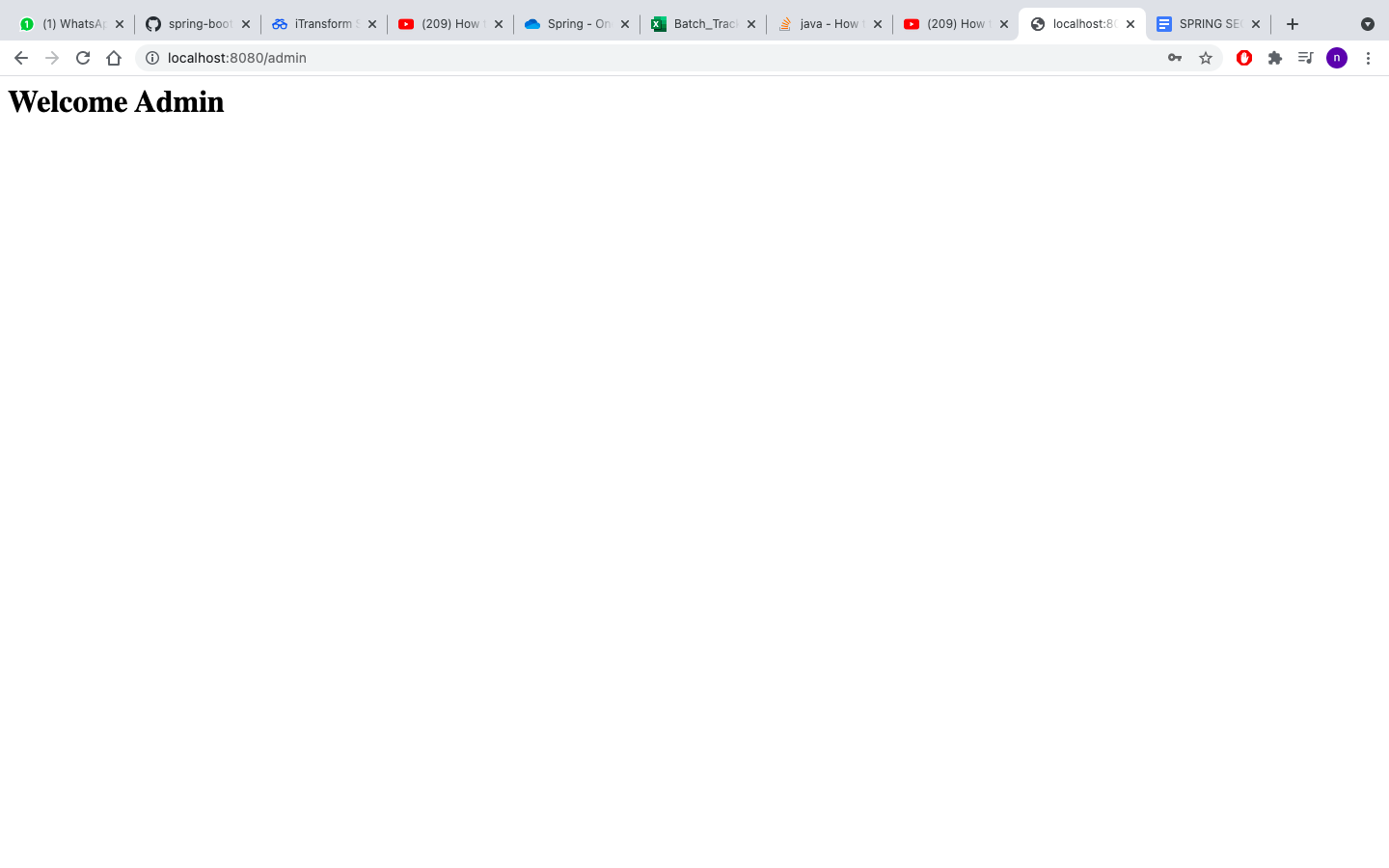
**Ouput:**

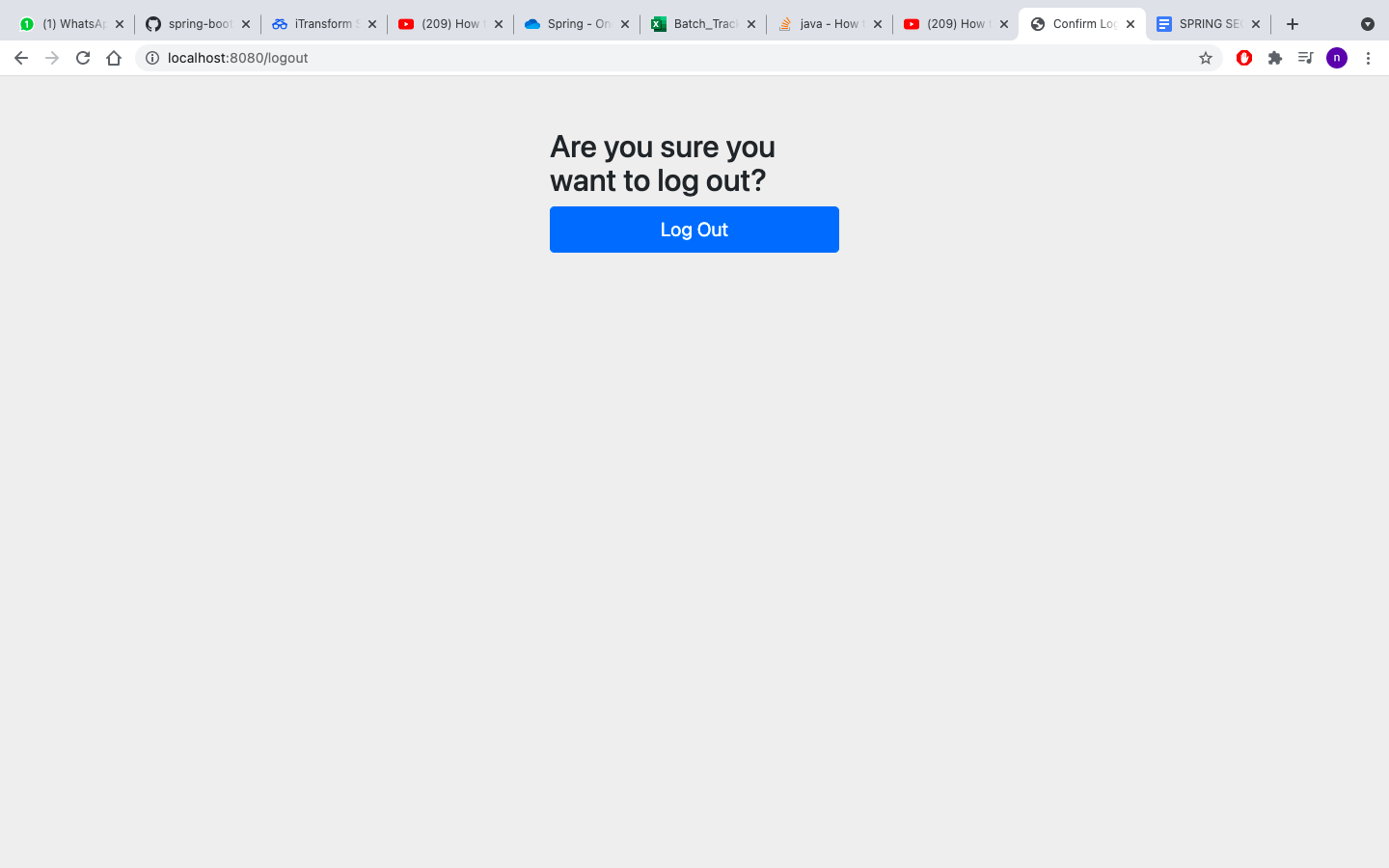
**localhost:8080/login**

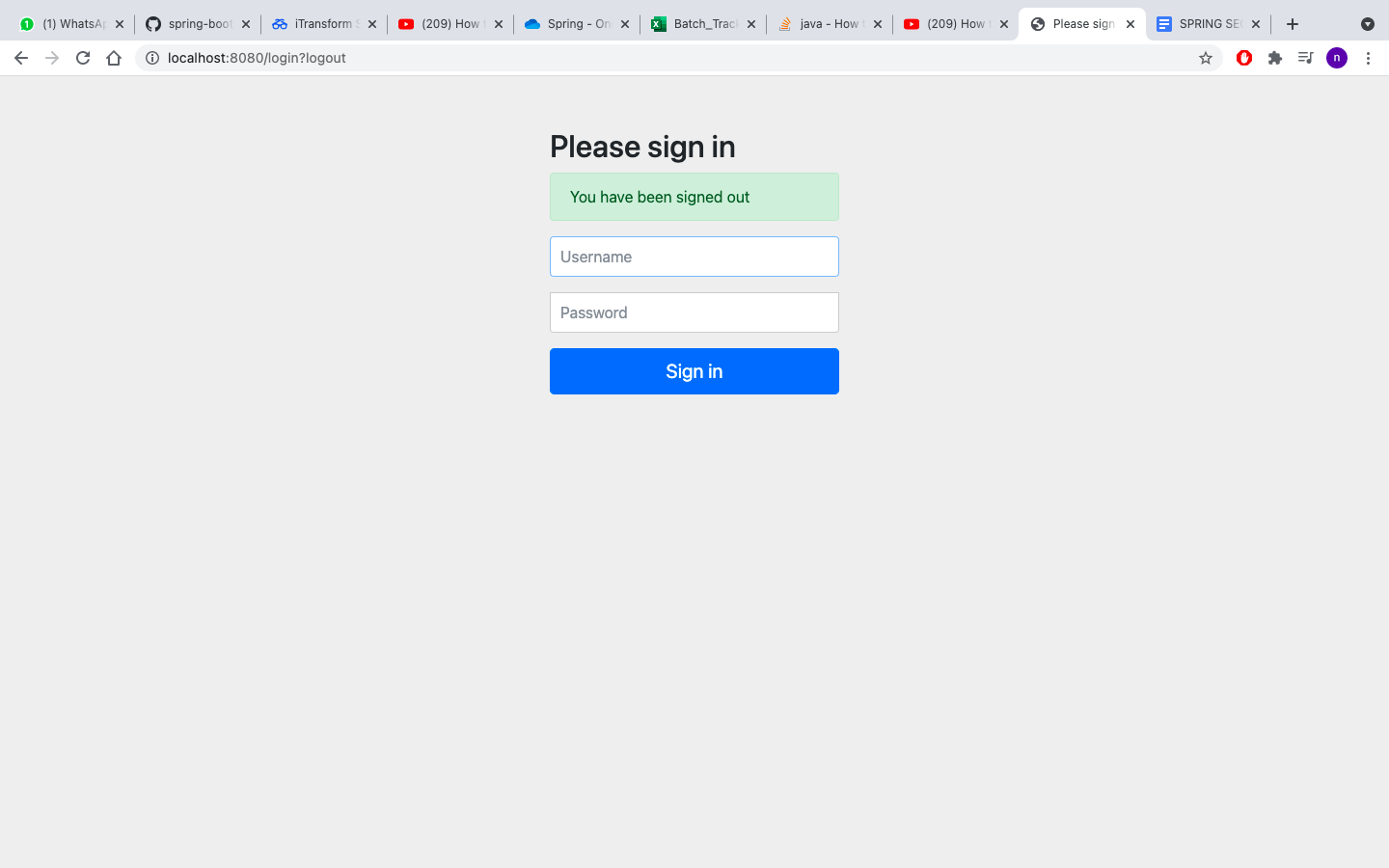


**localhost:8080/admin**



**localhost:8080/logout**





**2) USING JDBC**

**SpringSecurityJdbcApplication.java**

package com.springsecurity.springsecurityjdbc;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringSecurityJdbcApplication {

public static void main(String[] args) {

SpringApplication.run(SpringSecurityJdbcApplication.class, args);

}

}

**HelloResource.java**

package com.spring.security.springsecuritydemo;

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

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

@RestController

public class HelloResource {

@GetMapping("/")

public String home()

{

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

}

@GetMapping("/user")

public String user()

{

return("<h1>Welcome User</h1");

}

@GetMapping("/admin")

public String admin()

{

return("<h1>Welcome Admin</h1");

}

}

**securityConfiguration.java**

**package** com.springsecurity.springsecurityjdbc;

**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.core.userdetails.User;

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

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

@EnableWebSecurity

**public** **class** securityConfiguration **extends** WebSecurityConfigurerAdapter {

@Autowired

DataSource dataSource;

@Override

**public** **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").hasAnyRole("ADMIN", "USER")

.antMatchers("/").permitAll()

.and().formLogin();

}

@Bean

**public** PasswordEncoder getPasswordEncoder() {

**return** ~~NoOpPasswordEncoder~~.~~getInstance~~();

}

}

**schema.sql**

create table users(

username varchar\_ignorecase(50) not null primary key,

password varchar\_ignorecase(50) not null,

enabled boolean not null

);

create table authorities (

username varchar\_ignorecase(50) not null,

authority varchar\_ignorecase(50) not null,

constraint fk\_authorities\_users foreign key(username) references users(username)

);

create unique index ix\_auth\_username on authorities (username,authority);

**data.sql**

INSERT INTO users(username, password, enabled)

values('user'

,'pass',

true);

INSERT INTO users(username, password, enabled)

values('admin'

,'pass',

true);

INSERT INTO authorities(username, authority)

values('user'

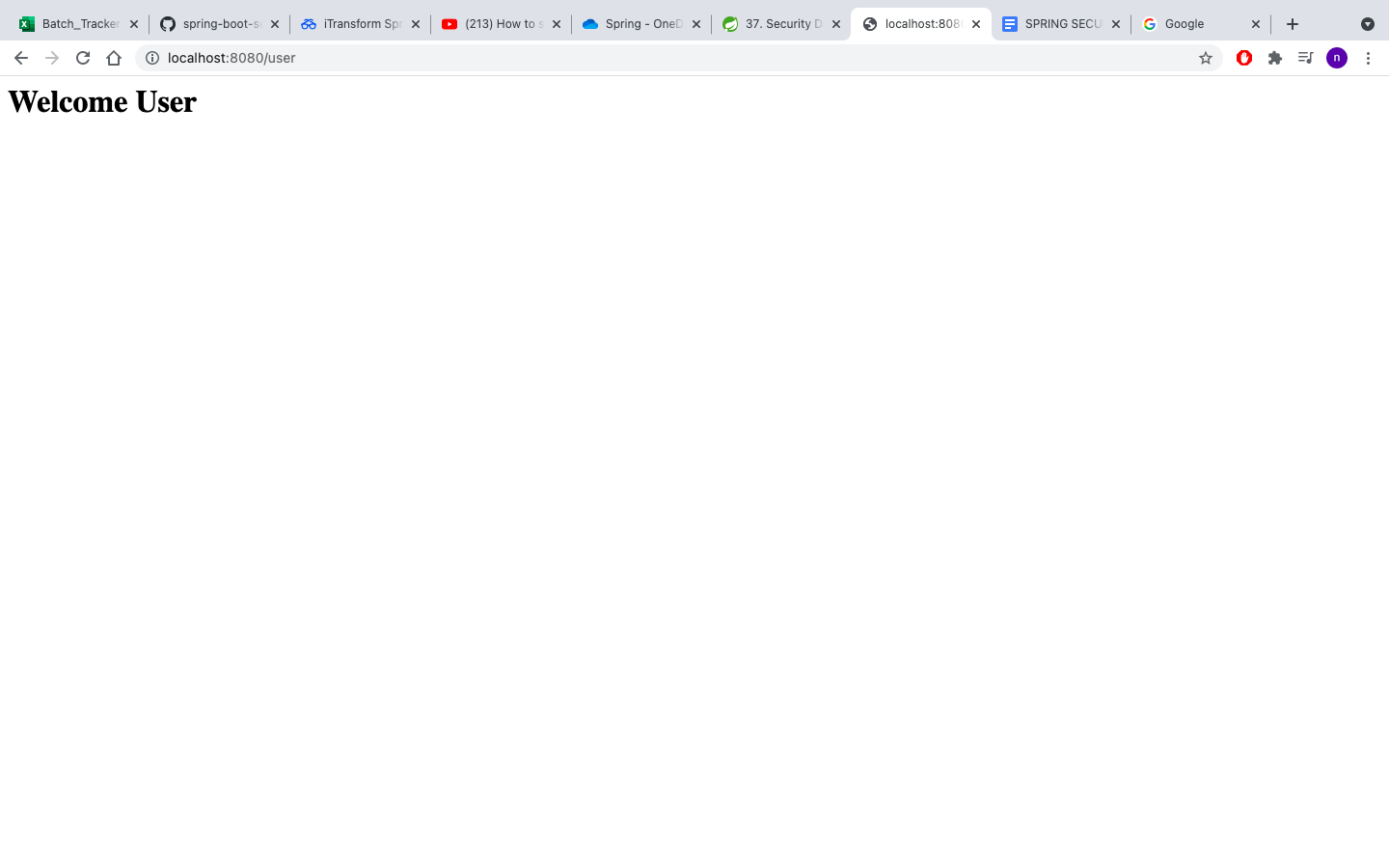
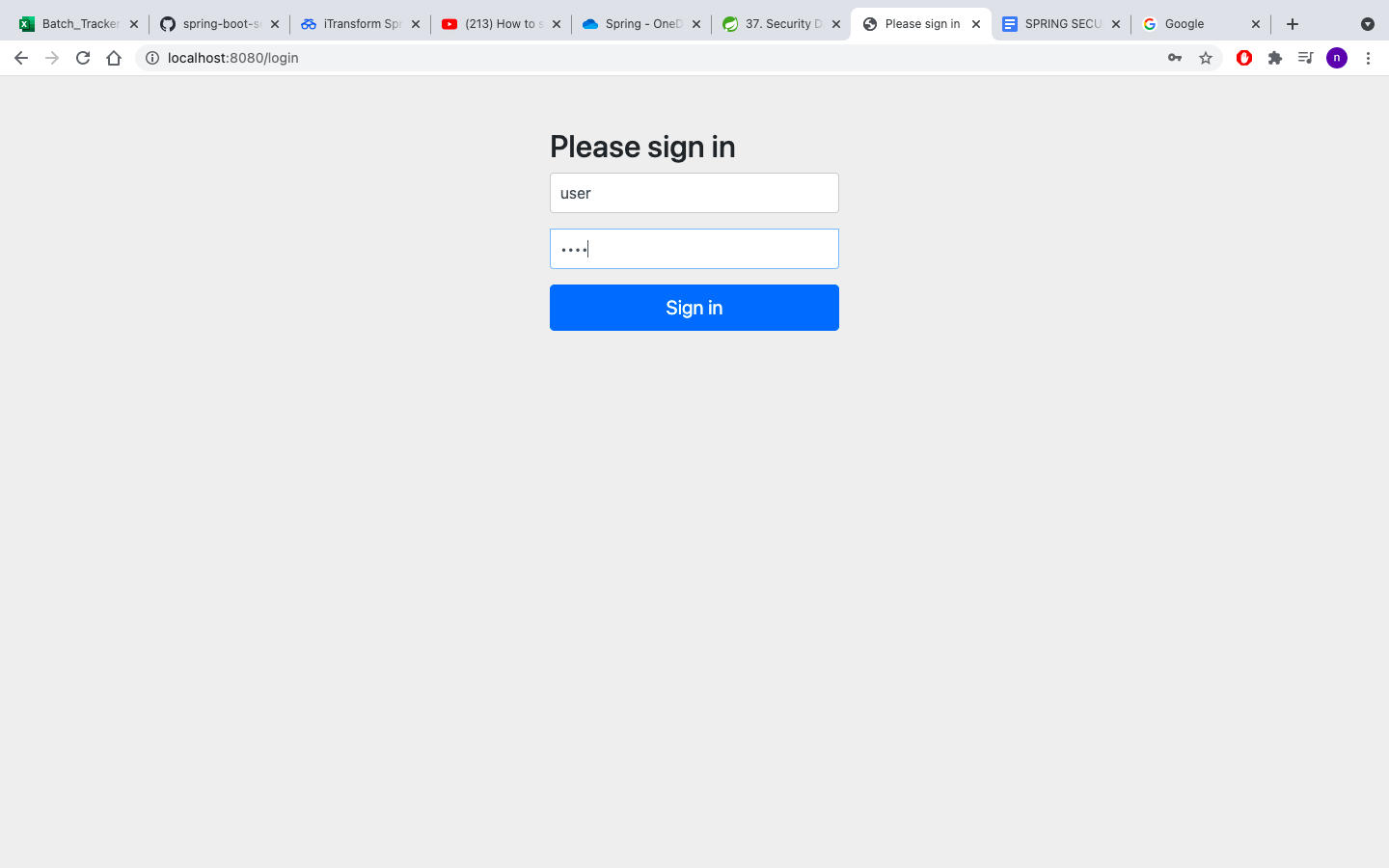
,'ROLE\_USER');

INSERT INTO authorities(username, authority)

values('admin'

,'ROLE\_USER');

**OUTPUT:**



**ASSIGNMENT - 2**

1. **Role based application**
2. **SpringsecuritydemoApplication.java**

package com.spring.security.springsecuritydemo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringsecuritydemoApplication {

public static void main(String[] args) {

SpringApplication.run(SpringsecuritydemoApplication.class, args);

}

}

**HelloResource.java**

package com.spring.security.springsecuritydemo;

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

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

@RestController

public class HelloResource {

@GetMapping("/")

public String home()

{

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

}

@GetMapping("/user")

public String user()

{

return("<h1>Welcome User</h1");

}

@GetMapping("/admin")

public String admin()

{

return("<h1>Welcome Admin</h1");

}

}

**securityConfiguration.java**

**package** com.spring.security.springsecuritydemo;

**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** securityConfiguration **extends** WebSecurityConfigurerAdapter {

@Override

**public** **void** configure(AuthenticationManagerBuilder auth) **throws** Exception

{

auth.inMemoryAuthentication()

.withUser("blah")

.password("blah")

.roles("USER")

.and()

.withUser("foo")

.password("foo")

.roles("ADMIN");

}

@Bean

**public** PasswordEncoder getPasswordEncoder() {

**return** ~~NoOpPasswordEncoder~~.~~getInstance~~();

}

@Override

**protected** **void** configure(HttpSecurity http) **throws** Exception {

http.authorizeRequests()

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

.antMatchers("/user").hasAnyRole("ADMIN", "USER")

.antMatchers("/").permitAll()

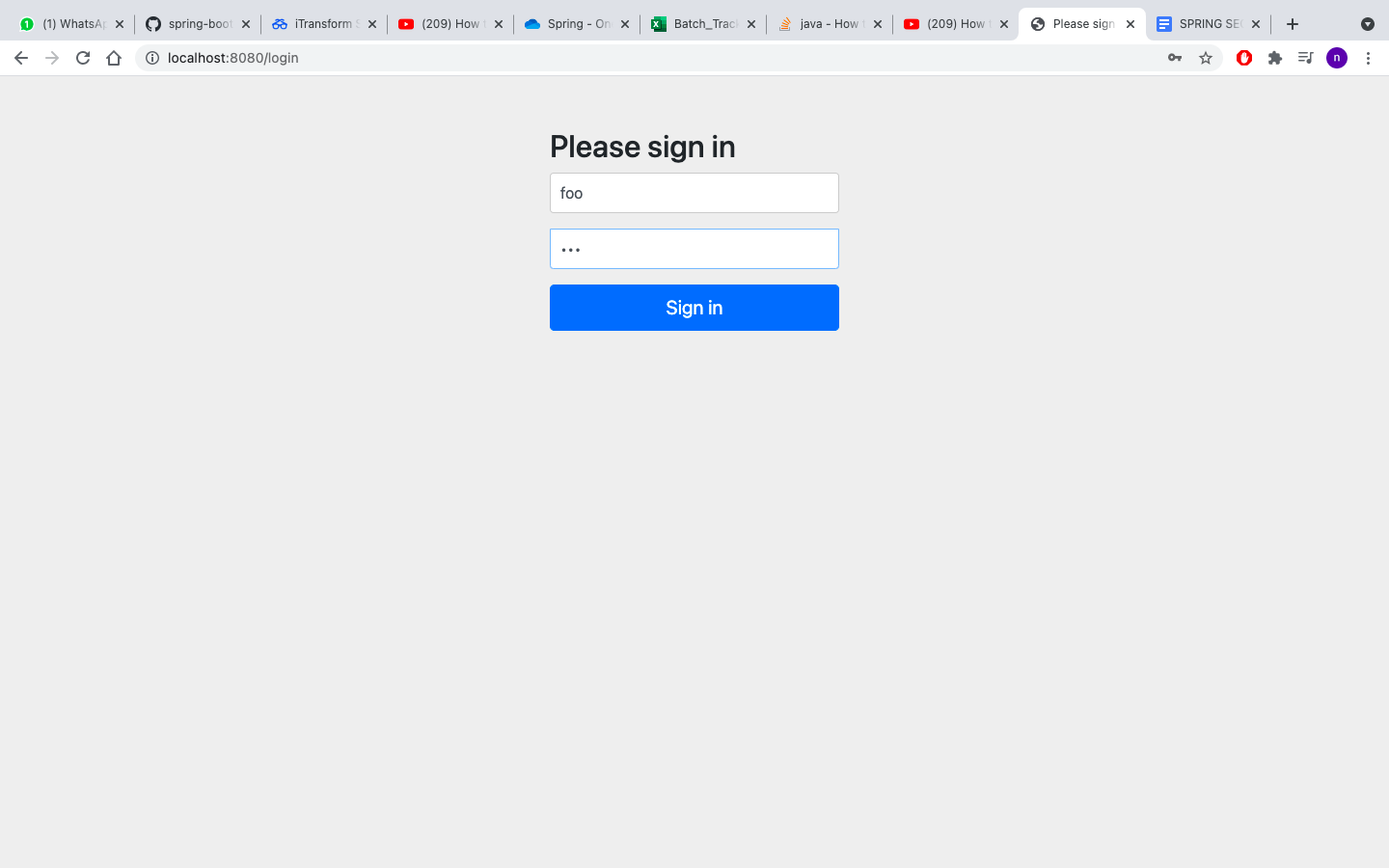
.and().formLogin();

}

}

**Output:**

**localhost:8080/admin**



**localhost:8080/user**