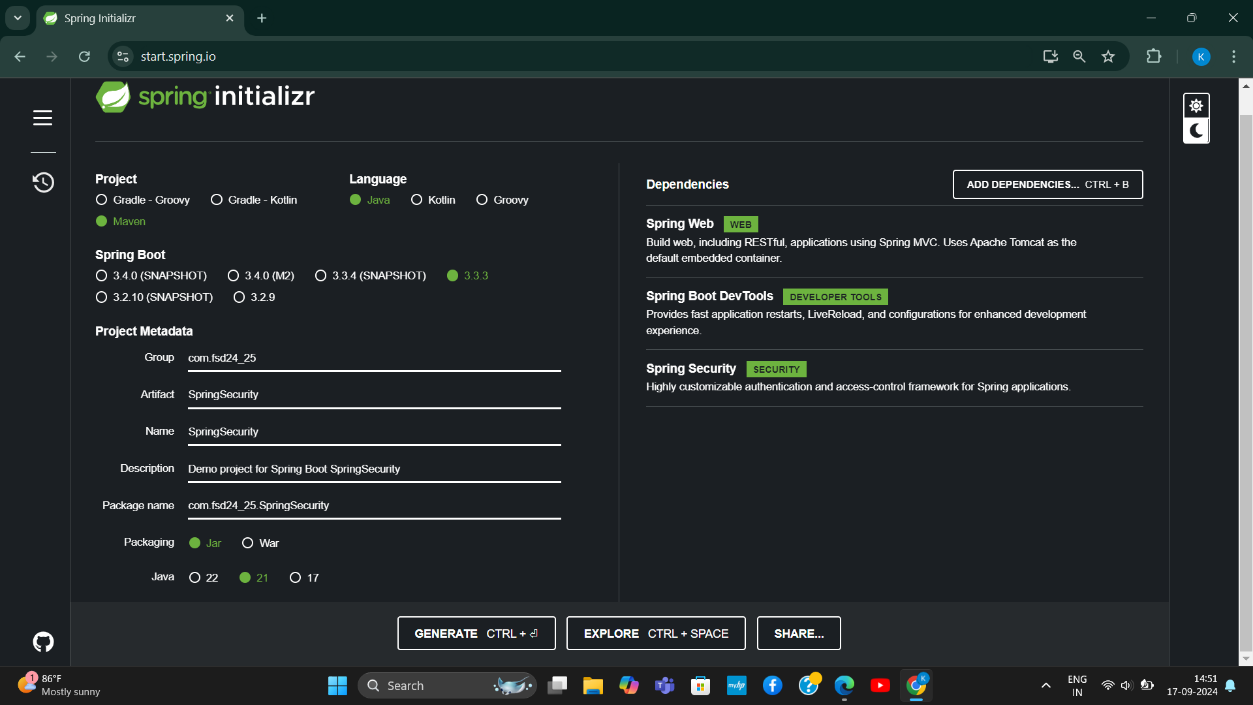
**5. Demonstrate securing REST APIs with spring security.**

Step1: To set project configuration🡪Go to browser🡪Search **start.spring.io**

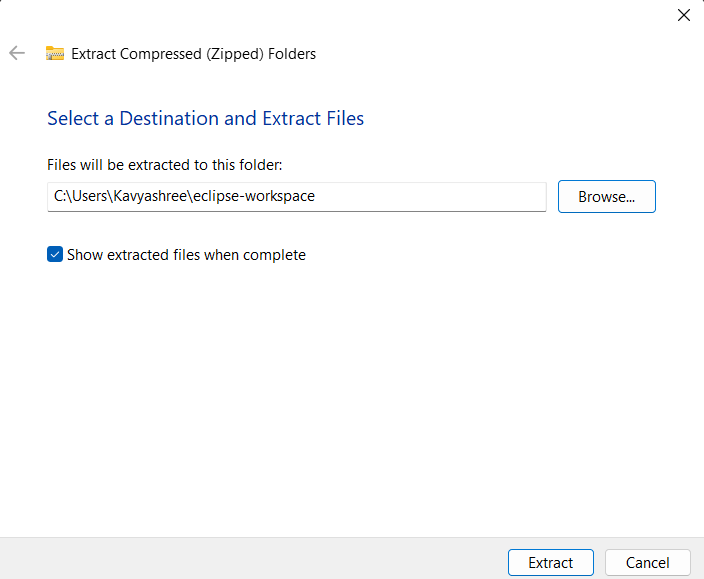
Step2: In spring initializr🡪Select Project-Maven🡪Language-Java🡪Spring Boot-3.3.3🡪 Group-com.fsd24\_25🡪Artifact-SpringSecurity🡪Description-Demo project for spring boot Securitydemo🡪Packaging-Jar🡪Java-21.

Dependencies🡪ADD DEPENDENCIES🡪Select-Spring Boot Dev Tools ,Spring Web and Spring Security🡪 Click on Generate.



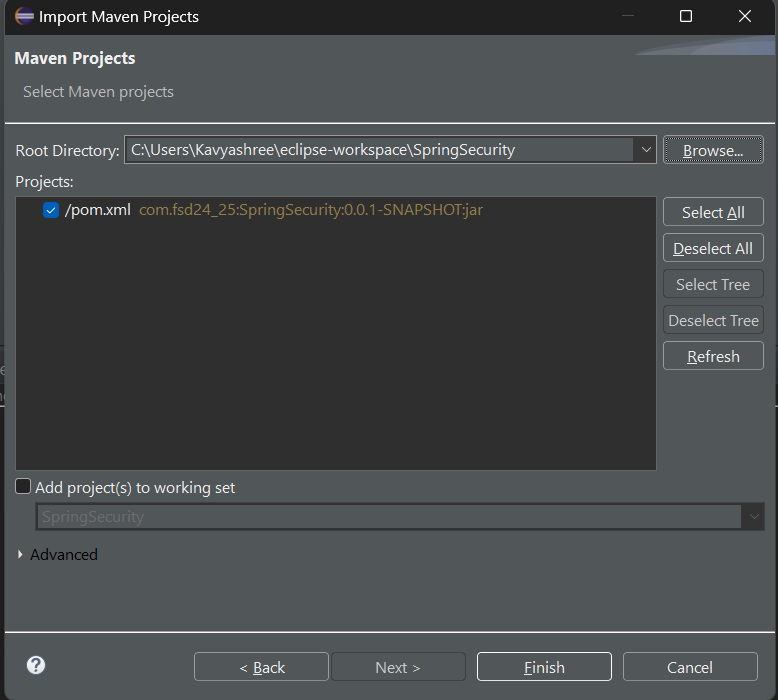
Step2: Next extract your zip file to workspace.

Step3: Select a destination.



Step4: Go to Eclipse🡪File🡪Import🡪Maven🡪Existing maven🡪Click on Browse and select the folder from where you extracted🡪Tick mark on pom.xml🡪Click on Finish.

Save the pom.xml



Step5: Now right click to your project🡪New🡪Packages🡪Name it as-com.fsd24\_25.SpringSecurity.Controller🡪Click on finish.

Step6: Inside com.fsd24\_25. SpringSecurity.Controller create a controller class.

Step7: Write the as follows

* **Using default credentials:**

**controller.java:**

package com.fsd24\_25. SpringSecurity.Controller;

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

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

@RestController

public class controller {

@GetMapping("/")

public String disp() {

return("welcome to spring security");

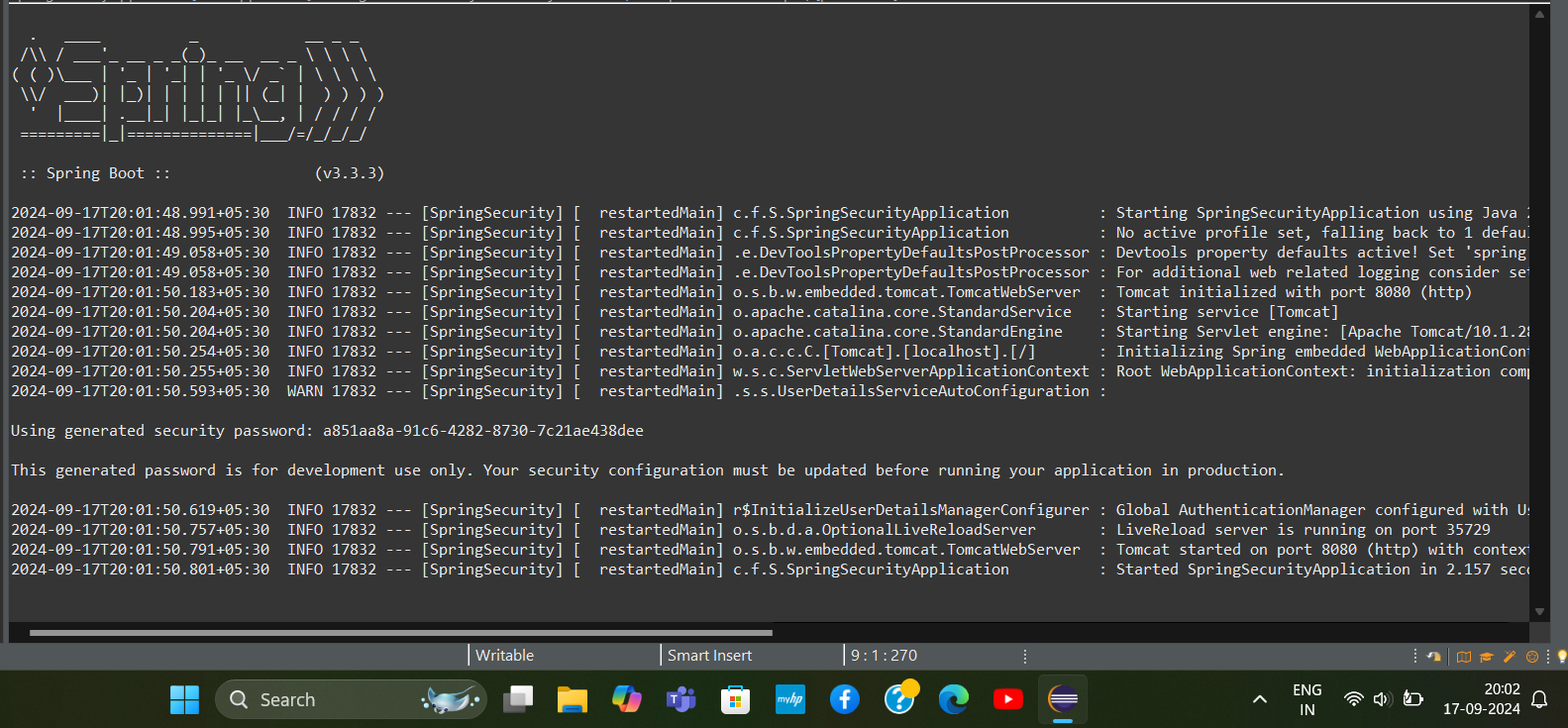
}

}

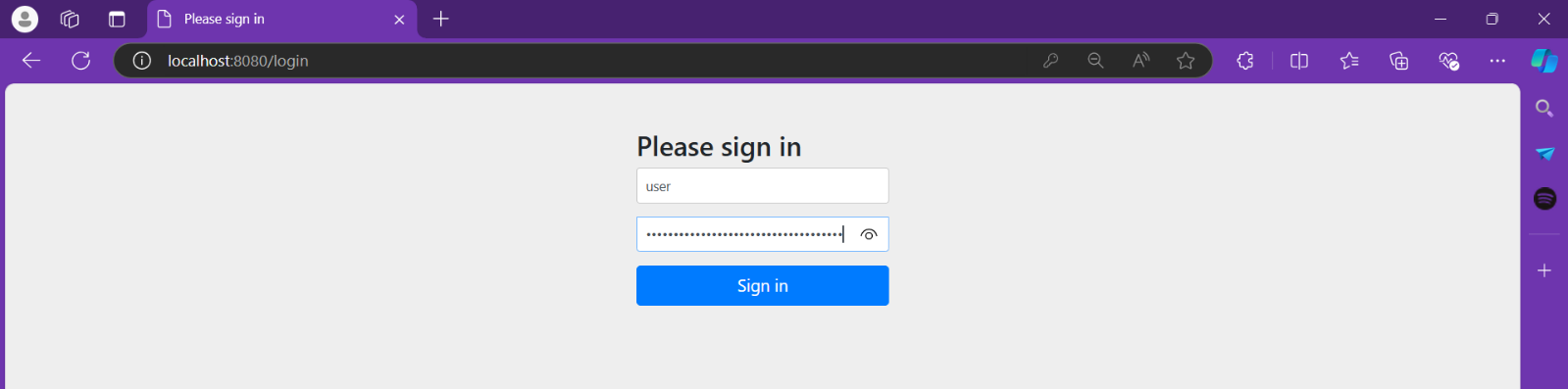
**Now run the code:**

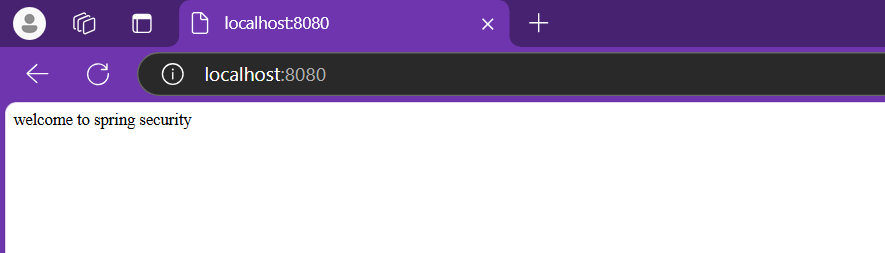
Right click to your project🡪Run as🡪Java Application🡪Search your project🡪Click on OK.

When we the run the password will be automatically generated we can see in console.

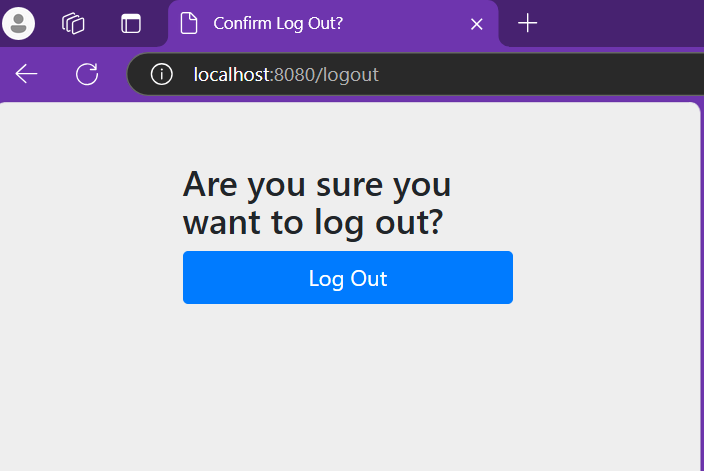


Go to the browser and enter: http://localhost:8080/





Then logout🡪http://localhost:8080/logout



Step8: **Using static predefined properties🡪**Write the code as follows in Application.properties.

**Src/main/resources🡪Application.properties:**

**spring.application.name=SpringSecurity**

**spring.security.user.name=kavyashree**

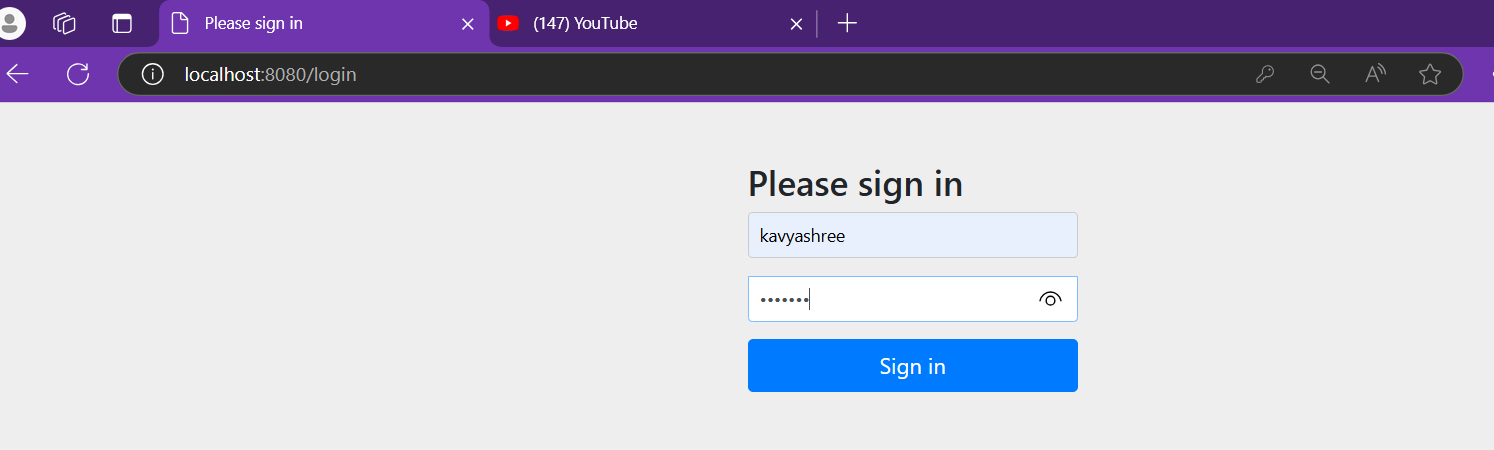
**spring.security.user.password=!abc23!**

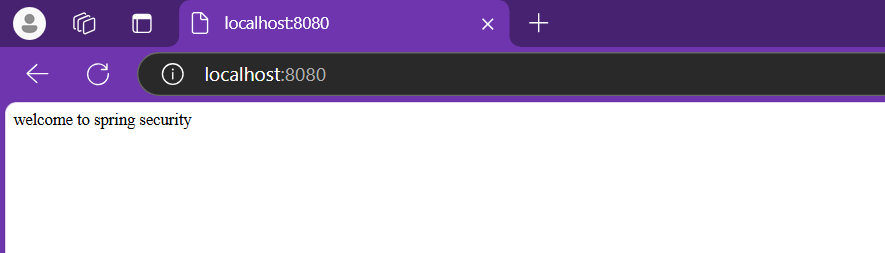
**Now run the code:**

Right click to your project🡪Run as🡪Java Application🡪Search your project🡪Click on OK.

Go to the browser🡪http://localhost:8080/

Enter the username and password as you written in application.properties.





Step9: **With authentication flow and authorization🡪**Write the code as follows in controller class.

**controller.java:**

**package com.fsd24\_25.SpringSecurity.Controller;**

import org.springframework.http.ResponseEntity;

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

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

@RestController

public class controller {

@GetMapping("/")

public ResponseEntity<String>Welcome() {

return ResponseEntity.ok("welcome to spring security ");

}

@GetMapping("/user")

public ResponseEntity<String> UserResponse(){

return ResponseEntity.ok("Welcome user...");

}

@GetMapping("/admin")

public ResponseEntity<String> AdminResponse(){

return ResponseEntity.ok("Welcome Admin...");

}

@GetMapping("/error/access-denied")

public ResponseEntity<?> AccessDeniedResponse(){

return ResponseEntity.ok("Invalid Credentials. Access Denied.");

}

}

Step10: Create another package🡪com.fsd24\_25.SpringSecurity.Config

Step11: Create a configuration class inside com.fsd24\_25.SpringSecurity.Config.

Write the code as follows:

**Configuration.java:**

**package com.fsd24\_25.SpringSecurity.Config;**

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.Customizer;

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

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

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

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

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

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

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

import org.springframework.security.provisioning.InMemoryUserDetailsManager;

import org.springframework.security.web.SecurityFilterChain;

@Configuration

@EnableWebSecurity

public class Config {

@Bean

public PasswordEncoder selectEncoder() {

return new BCryptPasswordEncoder();

}

@Bean

public UserDetailsService createUsers() {

UserDetails user1 = User.builder()

.username("varshitha")

.password(selectEncoder().encode("abc28"))

.roles("USER")

.build();

UserDetails user2 = User.builder()

.username("kavyashree")

.password(selectEncoder().encode("abc21"))

.roles("USER")

.build();

UserDetails admin = User.builder()

.username("kavyshree")

.password(selectEncoder().encode("!abc23!"))

.roles("ADMIN", "USER")

.build();

return new InMemoryUserDetailsManager(user1, user2, admin);

}

@Bean

public SecurityFilterChain createRoles(HttpSecurity http) throws Exception {

http.authorizeHttpRequests((authorizeHttpRequests) ->

authorizeHttpRequests

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

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

.requestMatchers("/error/\*\*").permitAll()

.requestMatchers("/").permitAll()

)

.formLogin(Customizer.withDefaults())

.exceptionHandling((exception) ->

exception.accessDeniedPage("/error/access-denied"));

return http.build();

}

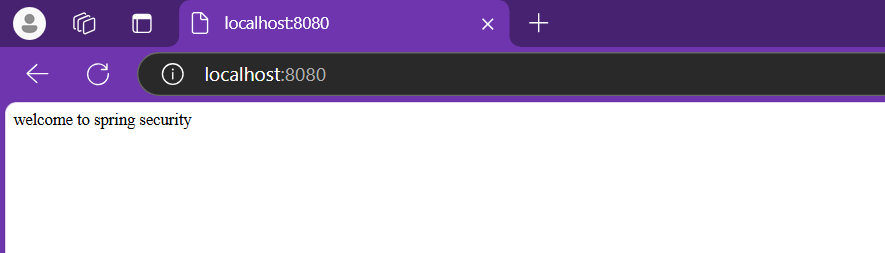
}

**Now run the code:**

Right click to your project🡪Run as🡪Java Application🡪Search your project🡪Click on OK.

Go to the browser🡪http://localhost:8080/

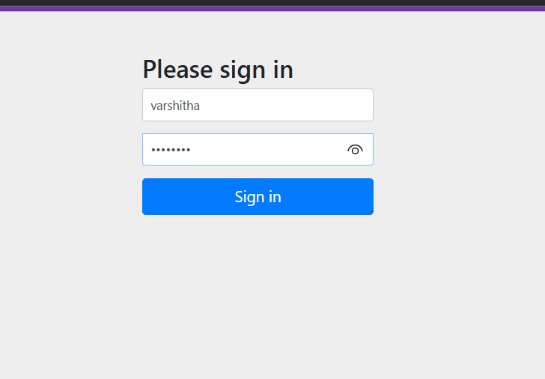
You will receive the welcome message page without logging into it.

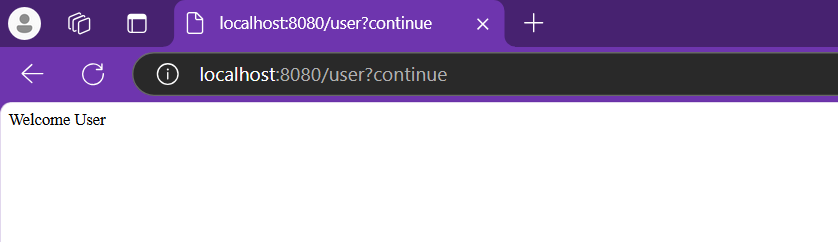


<http://localhost:8080/user>

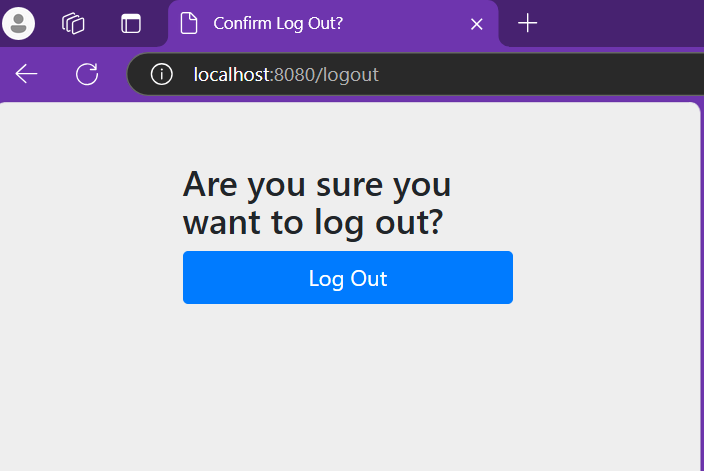
Now login with the username and password as you written in the code.

http://localhost:8080/user

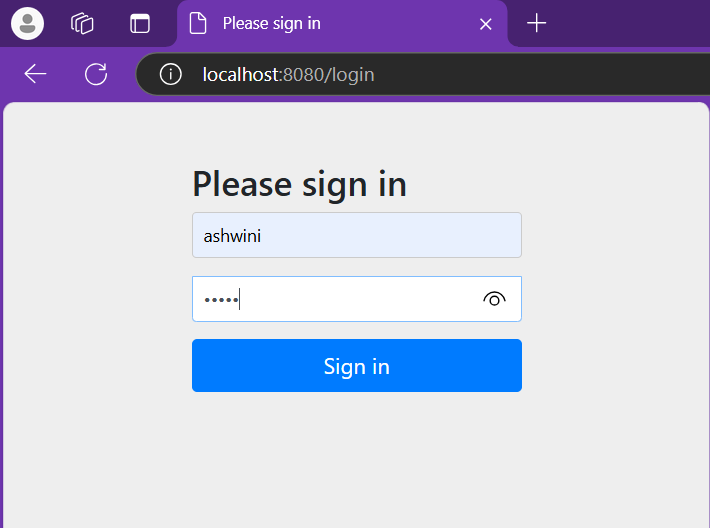


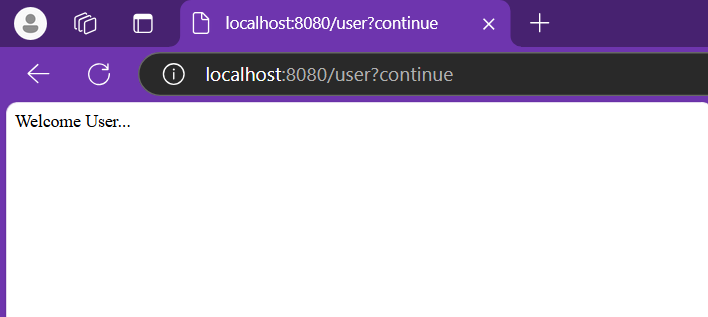


Logout🡪http://localhost:8080/logout.

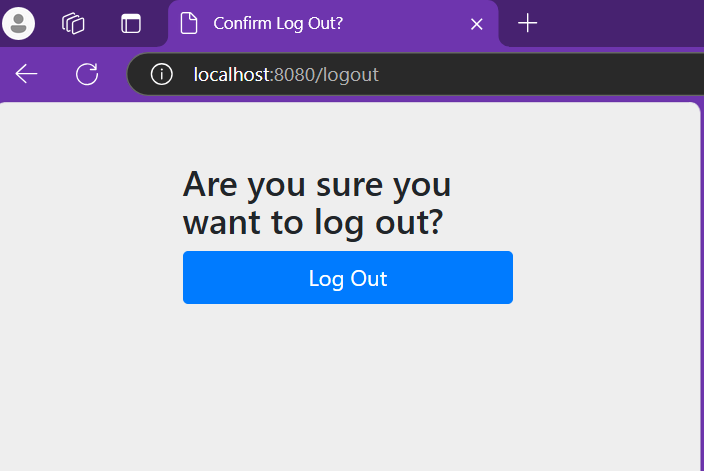


Login again using user2 credentials.

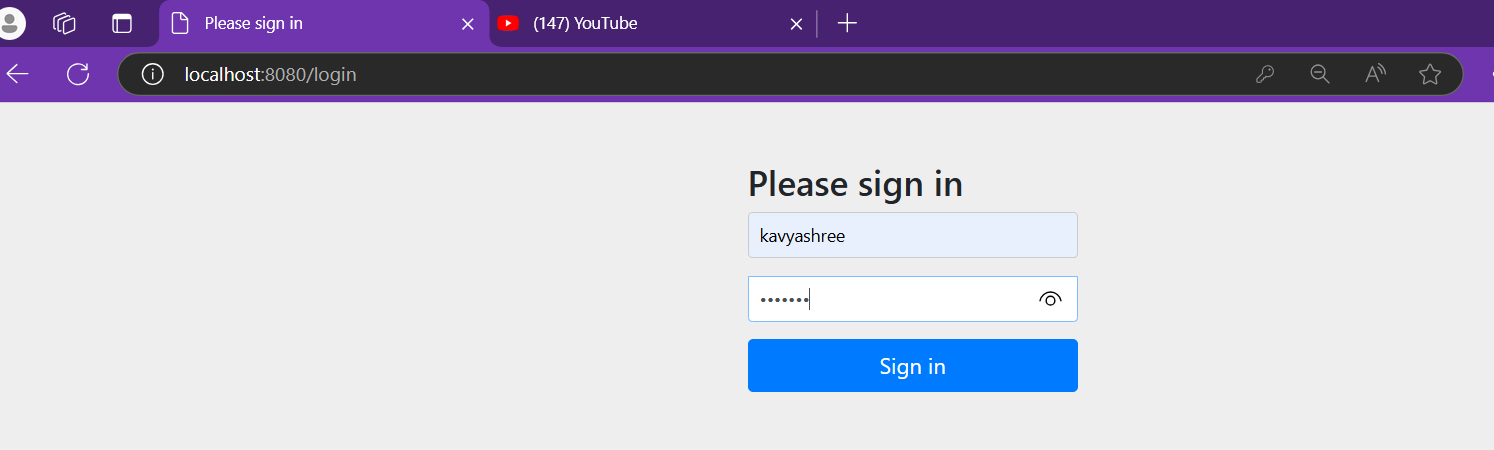


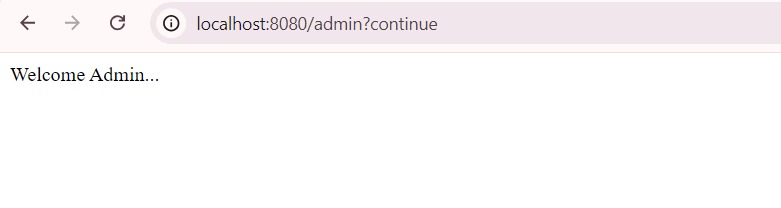


Logout🡪http://localhost:8080/logout

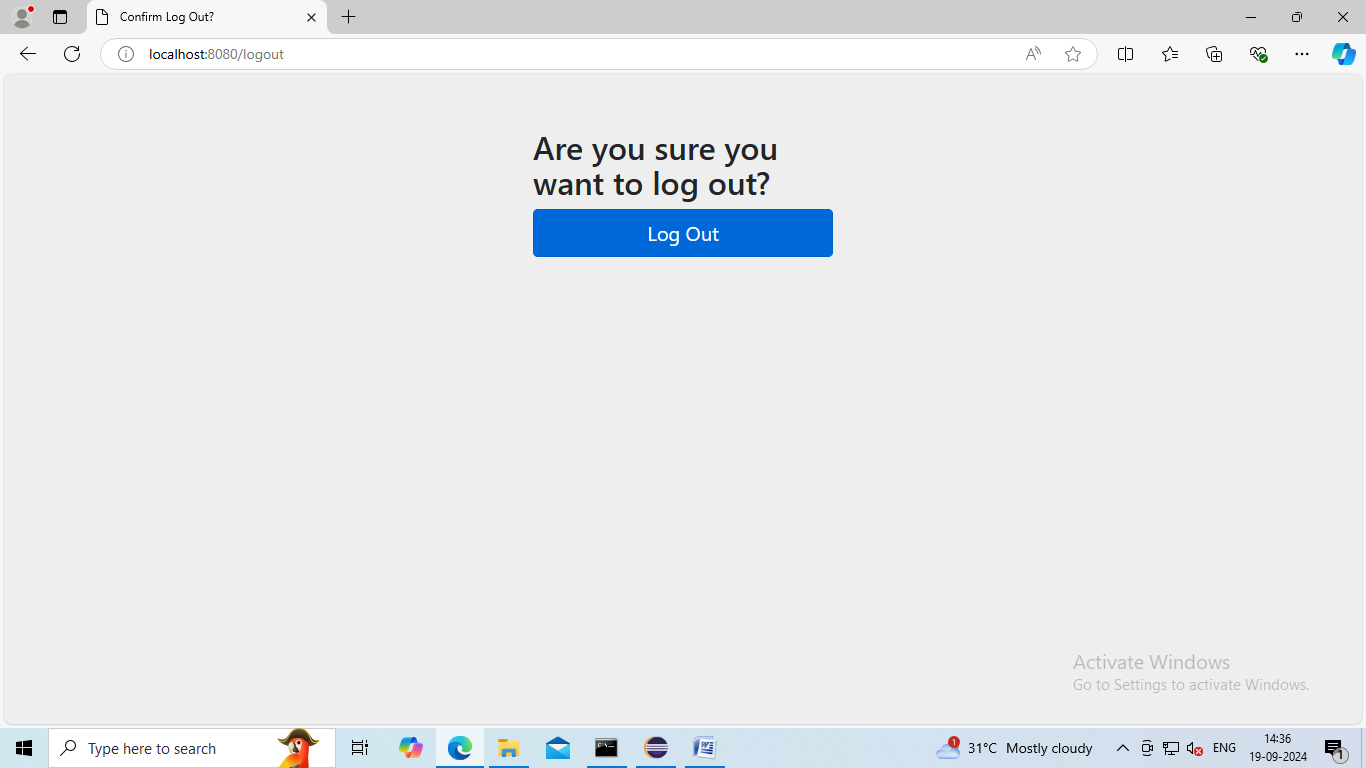


Now Login as admin 🡪http://localhost:8080/admin

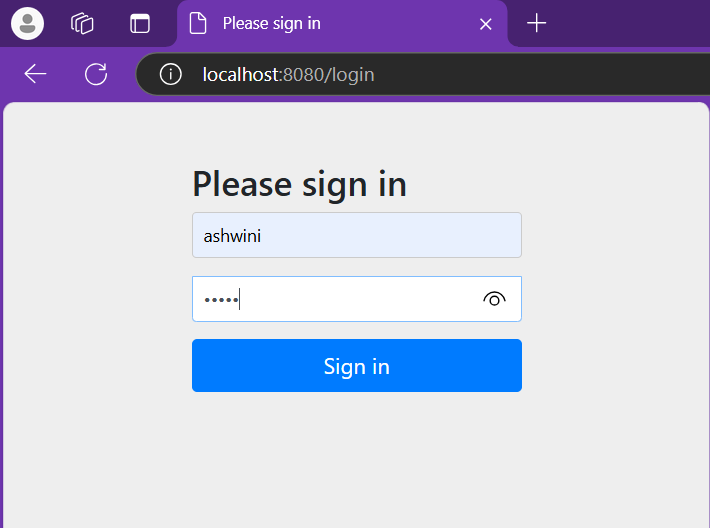




Logout🡪http://localhost:8080/logout



When you login as a user in http://localhost:808/admin



You will receive the error message.

