**AuthController - ListUsers\_AddSecurity**

**Remove below lines in SecurityConfig.java**

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

.requestMatchers("/db-console/\*\*").permitAll()

**Update below lines in SecurityConfig.java**

 .requestMatchers("/token").permitAll()

**To**

.requestMatchers("/auth/token").permitAll()

**Add below line in SecurityConfig.java**

.requestMatchers("/auth/Users/add").permitAll()

.requestMatchers("/auth/users").hasRole("USER")

**Updated securityFilterChain method**

  @Bean

    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {

        http

                // CSRF configuration

                .csrf(csrf -> csrf

                        .disable()) // Disable CSRF for stateless JWT authentication

                // Frame options for H2 console

                .headers(headers -> headers

                        .frameOptions(frameOptions -> frameOptions.sameOrigin()))

                // Authorization configuration

                .authorizeHttpRequests(authorize -> authorize

                        .requestMatchers("/auth/token").permitAll()

                        .requestMatchers("/auth/users/add").permitAll()

                        .requestMatchers("/auth/users").hasRole("USER")

                        .requestMatchers("/swagger-ui/\*\*").permitAll()

                        .requestMatchers("/v3/api-docs/\*\*").permitAll()

                        .requestMatchers("/test").authenticated() // `/test` requires authentication

                )

                // JWT-based authentication

                .oauth2ResourceServer(oauth2 -> oauth2

                        .jwt(Customizer.withDefaults()))

                // Stateless session management

                .sessionManagement(session -> session

                        .sessionCreationPolicy(SessionCreationPolicy.STATELESS))

                .headers(headers -> headers

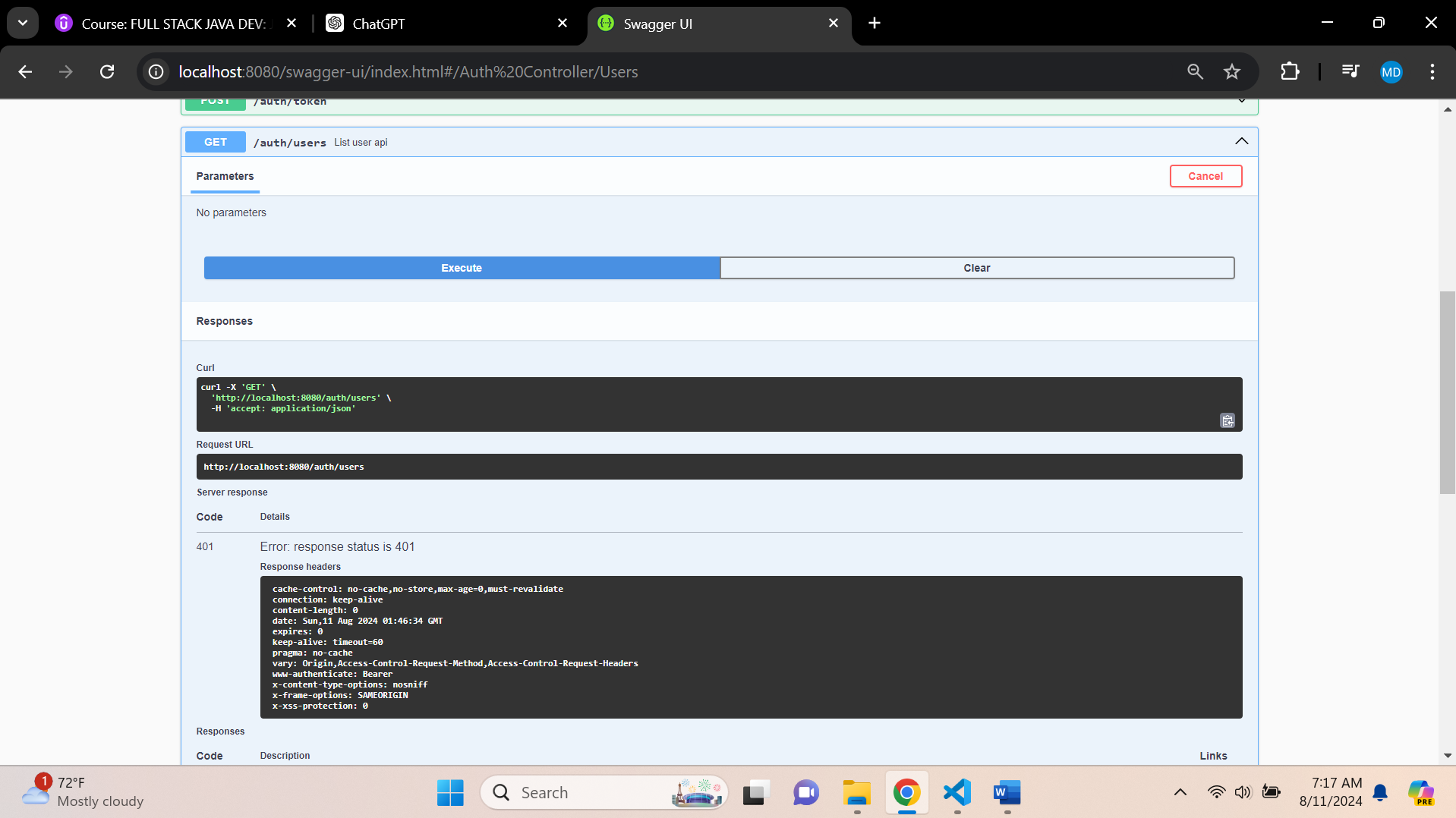
                        .frameOptions(frameOptions -> frameOptions.sameOrigin())); // Required for H2 console

        return http.build();

    }

**Output**

**Trying to list of users without authorizing**

****

**Update UserLoginDTO.java**

package org.studyeasy.SpringRestdemo.payload.auth;

import io.swagger.v3.oas.annotations.media.Schema;

import io.swagger.v3.oas.annotations.media.Schema.RequiredMode;

import jakarta.validation.constraints.Email;

import jakarta.validation.constraints.Size;

import lombok.Getter;

import lombok.Setter;

@Getter

@Setter

public class UserLoginDTO {

    @Email

    @Schema(description = "Email Address", example = "user@user.com", requiredMode = RequiredMode.REQUIRED)

    private String email;

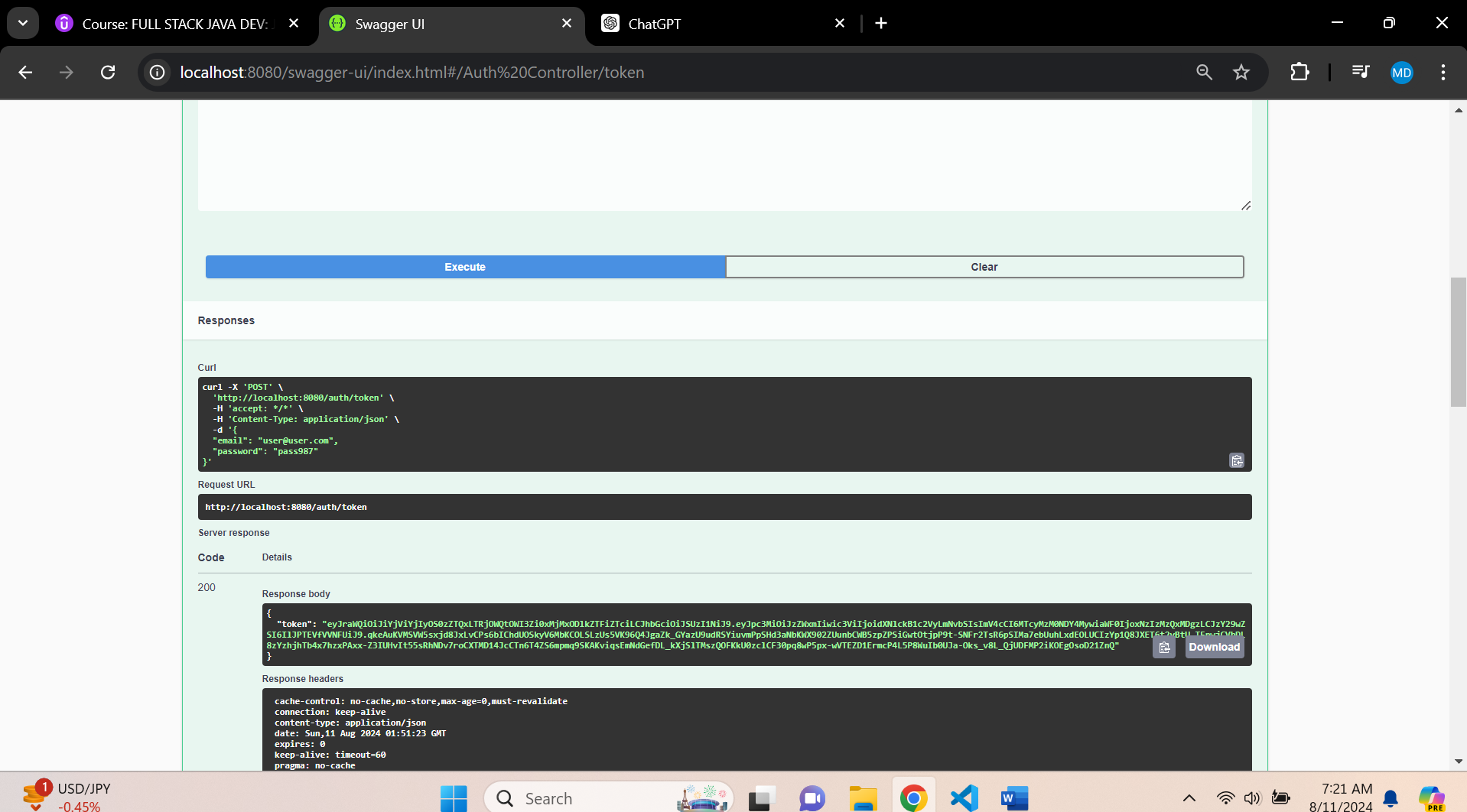
    @Size(min = 6, max = 20)

    @Schema(description = "Password", example = "pass987", requiredMode = RequiredMode.REQUIRED, maxLength = 20, minLength = 6)

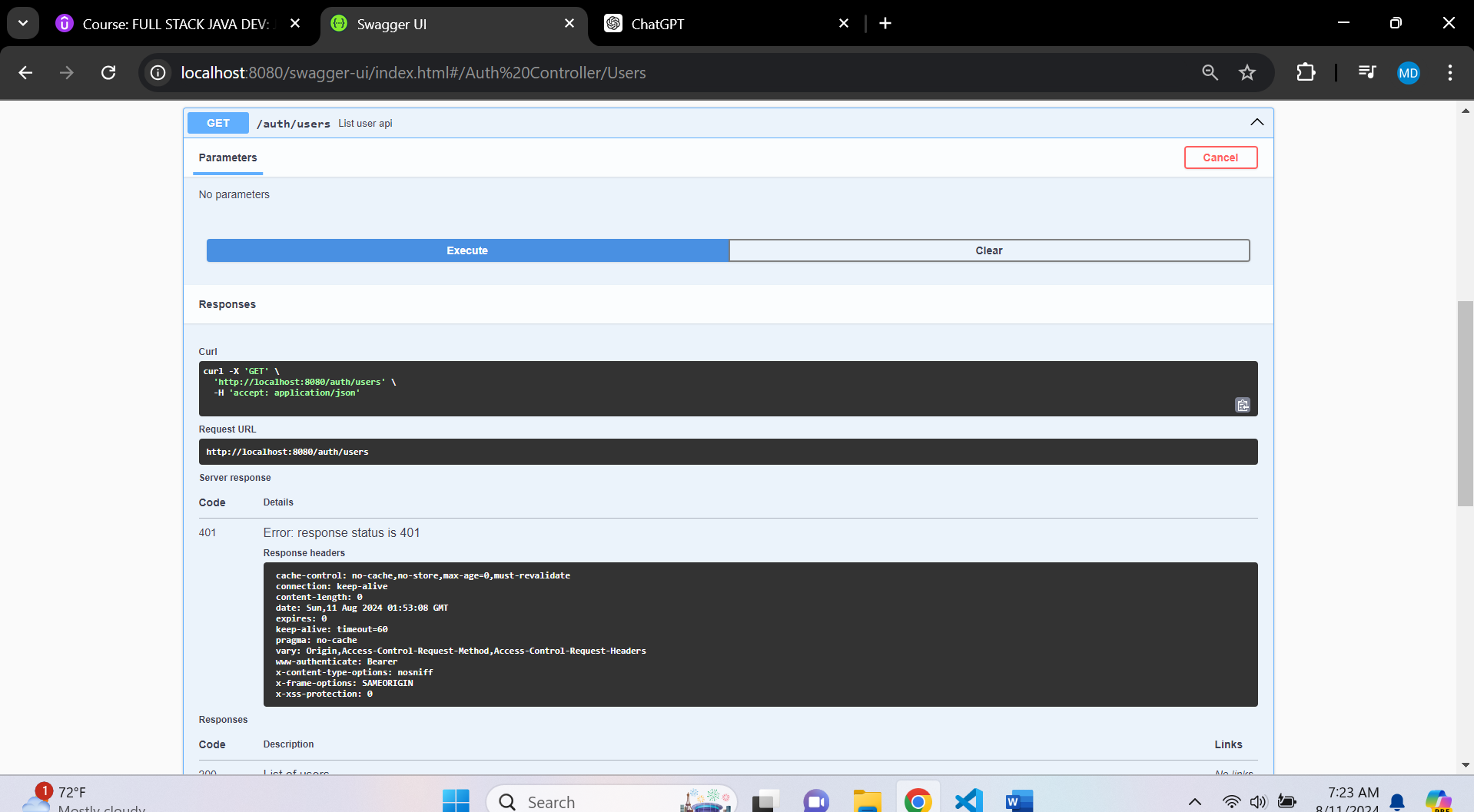
    private String password;

}

**Output:- Token Generation**

****

**And you still get error**

****

**Make changes in HomeController.java - commenting**

package org.studyeasy.SpringRestdemo.controller;

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

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

// import io.swagger.v3.oas.annotations.security.SecurityRequirement;

// import io.swagger.v3.oas.annotations.tags.Tag;

@RestController

public class HomeController {

    @GetMapping("/")

    public String demo(){

        return "Hello World";

    }

    // @GetMapping("/test")

    // @Tag(name = "Test", description = "The Test API.")

    // @SecurityRequirement(name = "studyeasy-demo-api")

    // public String test(){

    //     return "Test Api";

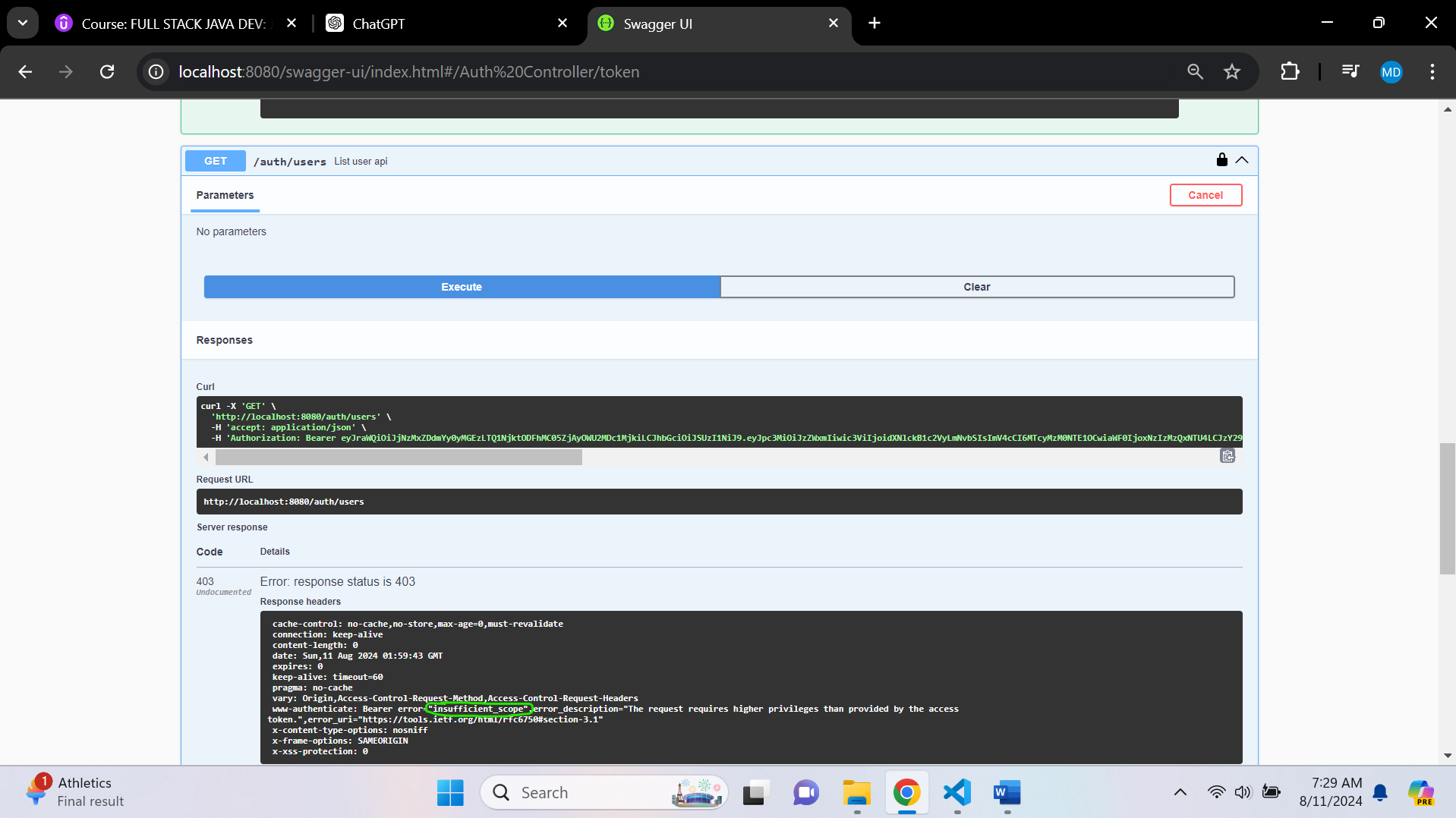
    // }

}

**In AuthController -> method -> Users -> add annotation -> SecurityRequirement**

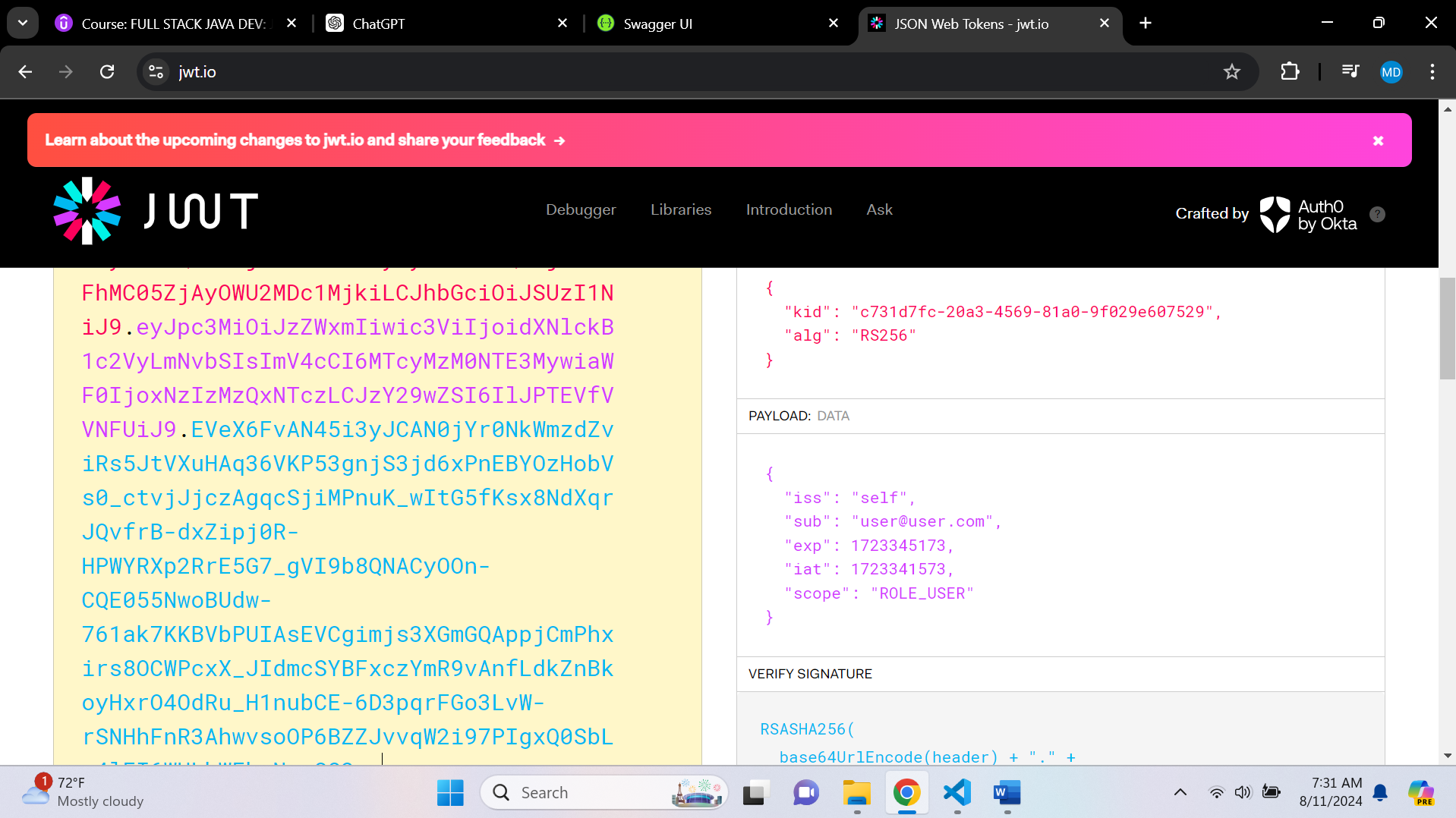
@SecurityRequirement(name = "studyeasy-demo-api")

**Output – Insufficient scope**

****

**Check it through -> jwt.io**

**It is valid scope and token; then why it is giving like this?**

****

**.hasRole() -> Used for monolithic application. So use has Authority()**