**5. WebApi\_Handson**

1. Explain CORS enablement for Web API access for local application

CORS (Cross-Origin Resource Sharing) allows web applications hosted on one domain to access resources from another domain. In Web API, CORS must be enabled to allow requests from external front-end clients such as React or Angular apps. To enable CORS, install the Microsoft.AspNetCore.Cors NuGet package. In Startup.cs, add services.AddCors() in ConfigureServices and configure allowed origins, headers, and methods. Then, in the Configure method, use app.UseCors() to apply it globally or selectively.

2. Demonstrate security in WebAPI

Security in Web API can be implemented using JWT (JSON Web Token) authentication with Bearer tokens. In Startup.cs, configure services.AddAuthentication() and services.AddJwtBearer() with token validation parameters like issuer, audience, and signing key. Use the [Authorize] attribute on controllers or actions to restrict access and include role-based security. In the AuthController, use [AllowAnonymous] to allow token generation for users. Claims can be added to the JWT to include user identity and roles.

**Code:**

**Program.cs**

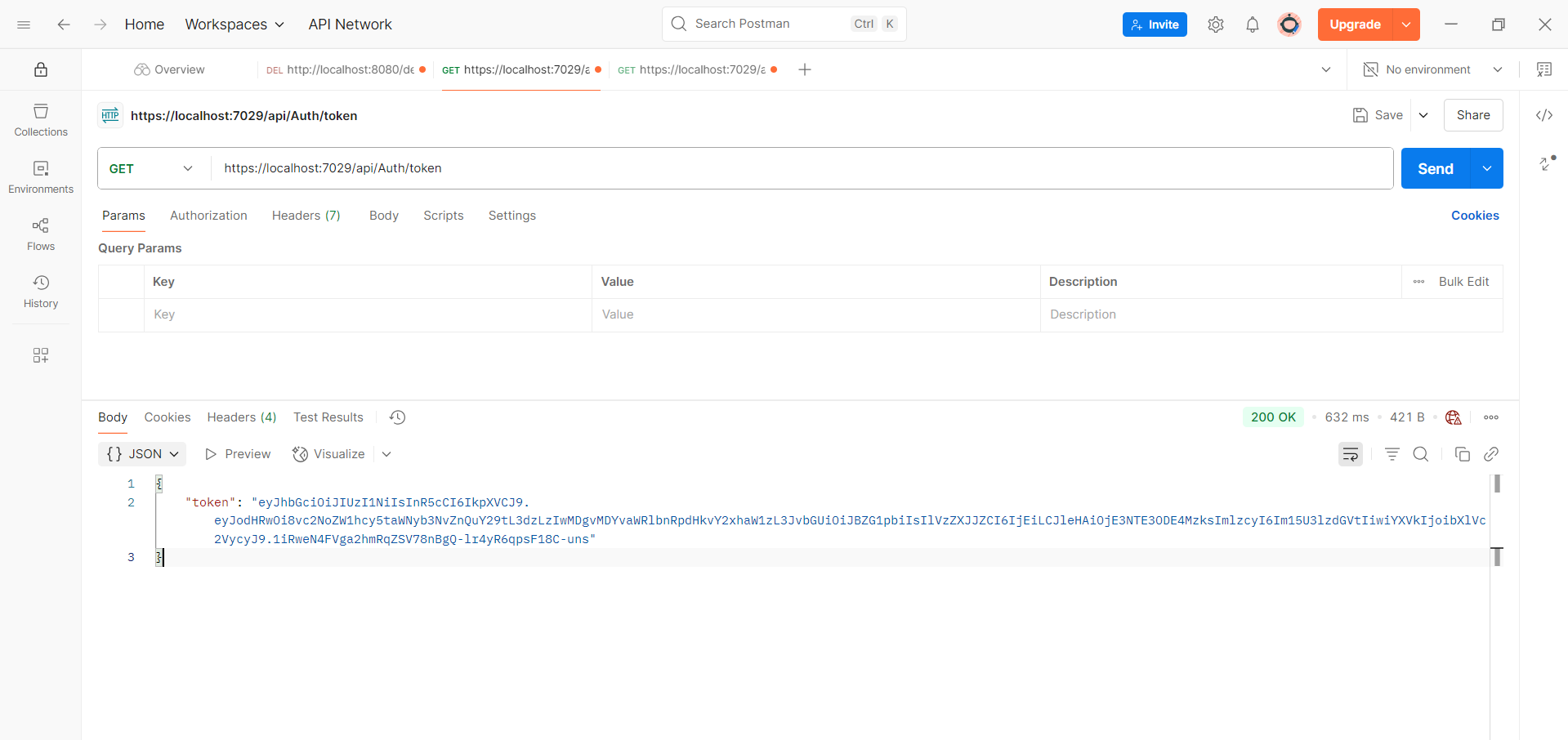
using Microsoft.AspNetCore.Authentication.JwtBearer;  
using Microsoft.IdentityModel.Tokens;  
using System.Text;  
  
var builder = WebApplication.CreateBuilder(args);  
  
// Add services to the container.  
builder.Services.AddControllers();  
builder.Services.AddEndpointsApiExplorer();  
  
// Add CORS  
builder.Services.AddCors(options =>  
{  
 options.AddPolicy("AllowLocalhost",  
 policy => policy.AllowAnyOrigin().AllowAnyMethod().AllowAnyHeader());  
});  
  
// JWT Setup  
var securityKey = "mysuperdupersecretkey1234567890abcdef";  
var symmetricSecurityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(securityKey));  
  
builder.Services.AddAuthentication(options =>  
{  
 options.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;  
 options.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;  
})  
.AddJwtBearer(options =>  
{  
 options.TokenValidationParameters = new TokenValidationParameters  
 {  
 ValidateIssuer = true,  
 ValidateAudience = true,  
 ValidateLifetime = true,  
 ValidateIssuerSigningKey = true,  
 ValidIssuer = "mySystem",  
 ValidAudience = "myUsers",  
 IssuerSigningKey = symmetricSecurityKey  
 };  
});  
  
var app = builder.Build();  
app.UseCors("AllowLocalhost");  
app.UseAuthentication();  
app.UseAuthorization();  
app.MapControllers();  
app.Run();

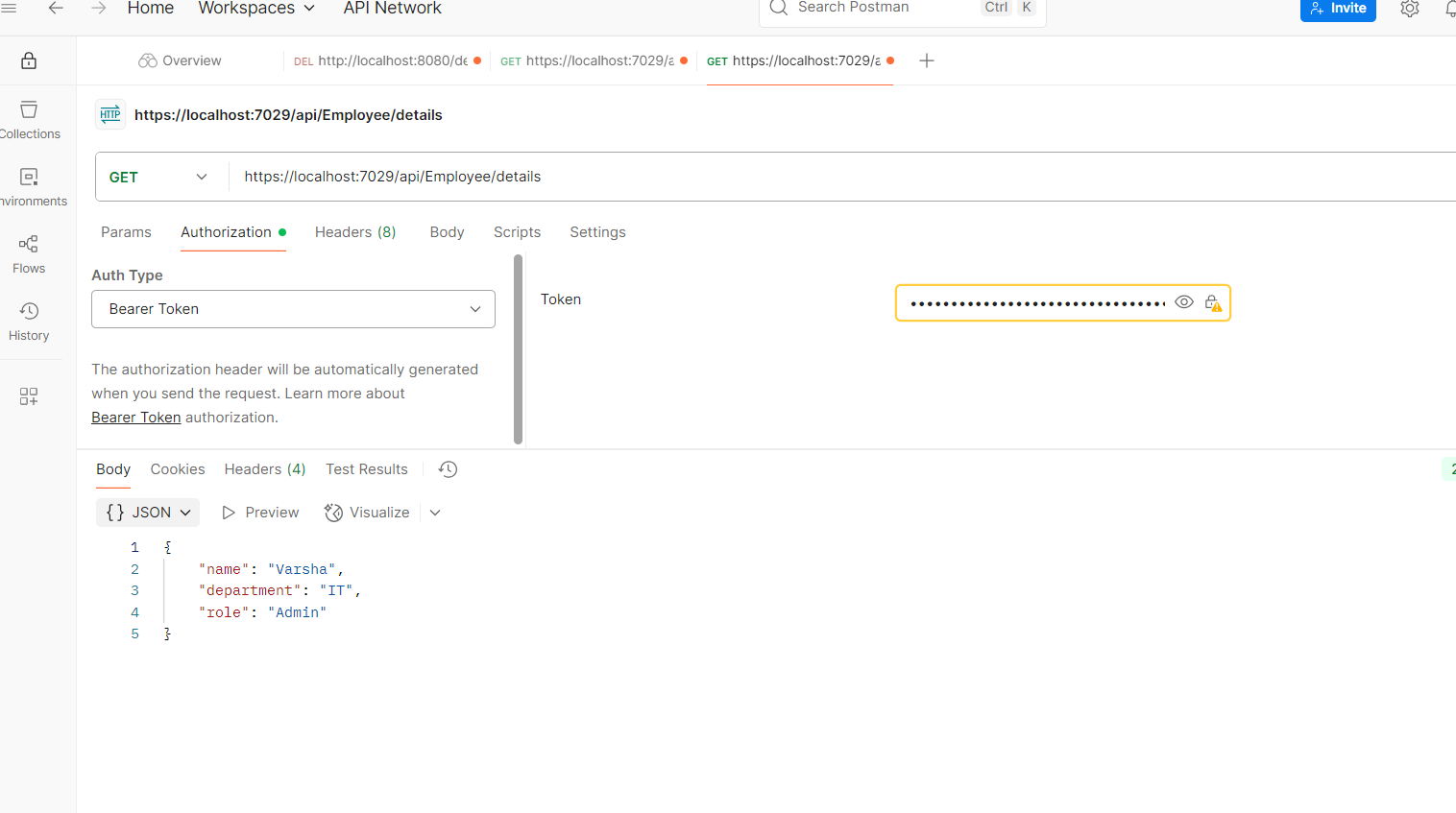
**AuthController.cs**

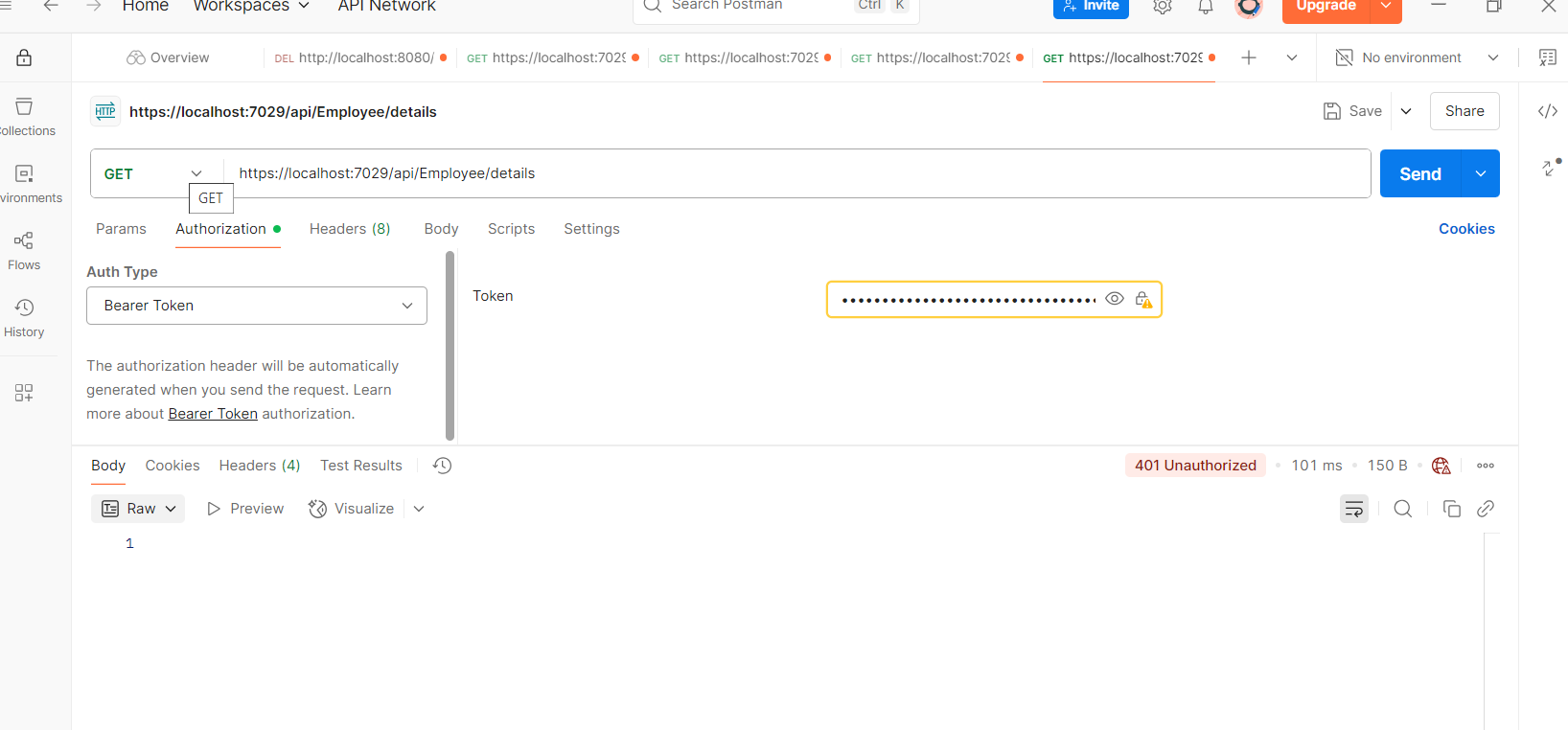
using Microsoft.AspNetCore.Authorization;  
using Microsoft.AspNetCore.Mvc;  
using Microsoft.IdentityModel.Tokens;  
using System.IdentityModel.Tokens.Jwt;  
using System.Security.Claims;  
using System.Text;  
  
namespace JwtAuthDemoApi.Controllers  
{  
 [AllowAnonymous]  
 [Route("api/[controller]")]  
 [ApiController]  
 public class AuthController : ControllerBase  
 {  
 [HttpGet("token")]  
 public IActionResult GetToken()  
 {  
 var token = GenerateJSONWebToken(1, "Admin");  
 return Ok(new { token });  
 }  
  
 private string GenerateJSONWebToken(int userId, string userRole)  
 {  
 var securityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes("mysuperdupersecretkey1234567890abcdef"));  
 var credentials = new SigningCredentials(securityKey, SecurityAlgorithms.HmacSha256);  
  
 var claims = new List<Claim>  
 {  
 new Claim(ClaimTypes.Role, userRole),  
 new Claim("UserId", userId.ToString())  
 };  
  
 var token = new JwtSecurityToken(  
 issuer: "mySystem",  
 audience: "myUsers",  
 claims: claims,  
 expires: DateTime.Now.AddMinutes(2),  
 signingCredentials: credentials);  
  
 return new JwtSecurityTokenHandler().WriteToken(token);  
 }  
 }  
}

**EmployeeController.cs**

using Microsoft.AspNetCore.Authorization;  
using Microsoft.AspNetCore.Mvc;  
  
namespace JwtAuthDemoApi.Controllers  
{  
 [Authorize(Roles = "Admin,POC")]  
 [Route("api/[controller]")]  
 [ApiController]  
 public class EmployeeController : ControllerBase  
 {  
 [HttpGet("details")]  
 public IActionResult GetDetails()  
 {  
 return Ok(new  
 {  
 Name = "Varsha",  
 Department = "IT",  
 Role = "Admin"  
 });  
 }  
 }  
}

**Output:  
**

****

****