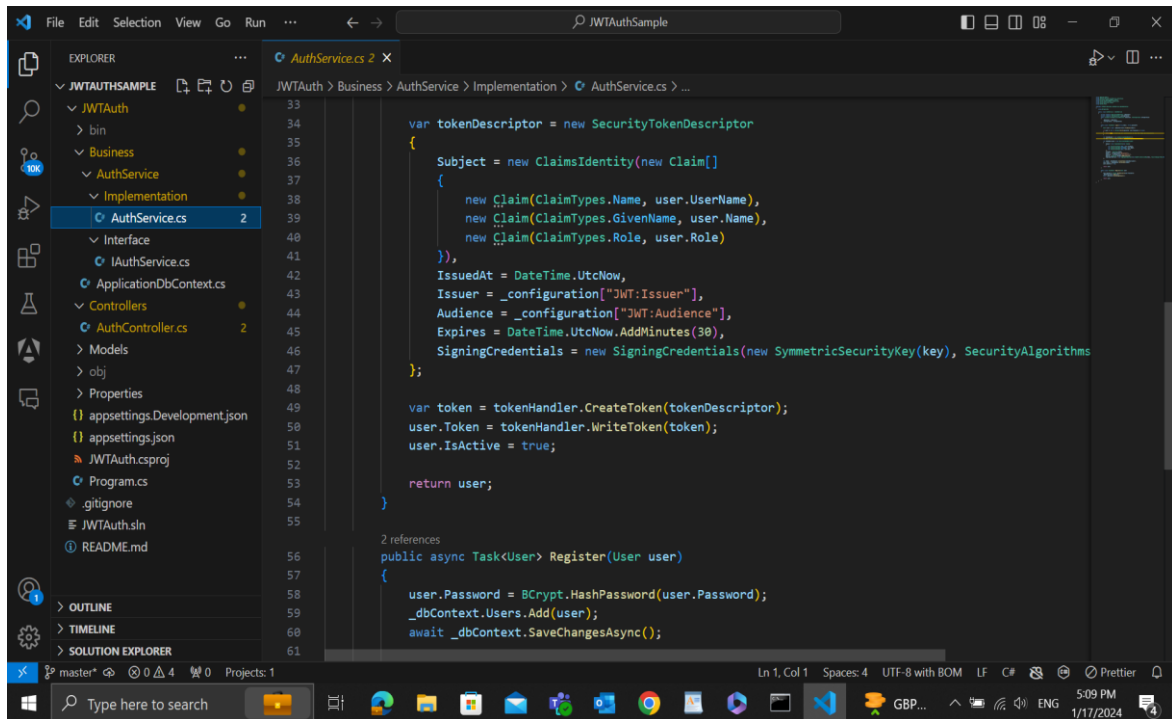


WEBAPI WITH JWT AUTHENTICATION

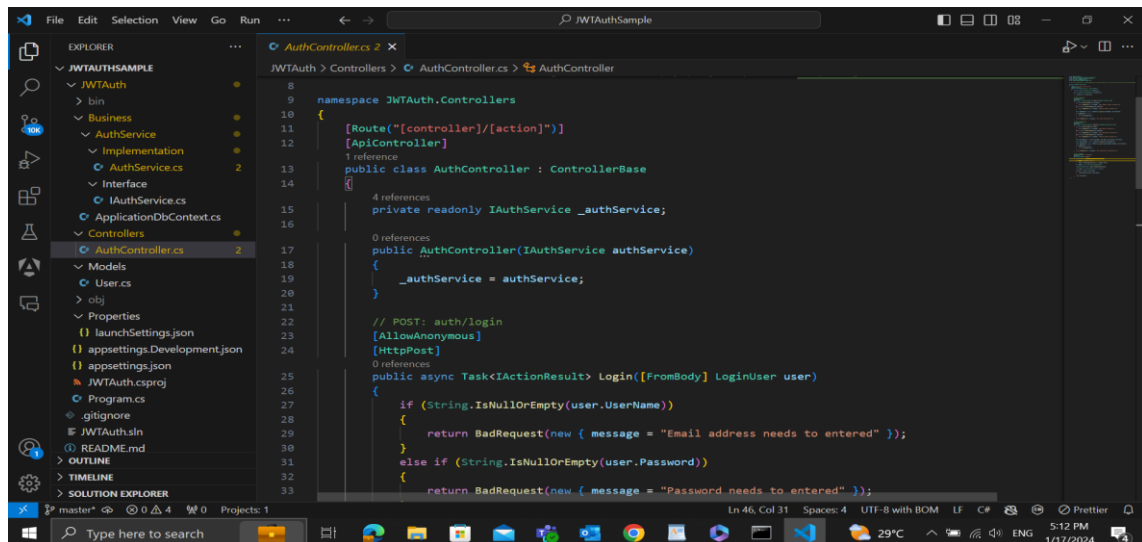


```
var tokenDescriptor = new SecurityTokenDescriptor
{
    Subject = new ClaimsIdentity(new Claim[]
    {
        new Claim(ClaimTypes.Name, user.UserName),
        new Claim(ClaimTypes.GivenName, user.Name),
        new Claim(ClaimTypes.Role, user.Role)
    }),
    IssuedAt = DateTime.UtcNow,
    Issuer = _configuration["JWT:Issuer"],
    Audience = _configuration["JWT:Audience"],
    Expires = DateTime.UtcNow.AddMinutes(30),
    SigningCredentials = new SigningCredentials(new SymmetricSecurityKey(key), SecurityAlgorithms
};

var token = tokenHandler.CreateToken(tokenDescriptor);
user.Token = tokenHandler.WriteToken(token);
user.IsActive = true;

return user;

2 references
public async Task<User> Register(User user)
{
    user.Password = BCrypt.HashPassword(user.Password);
    _dbContext.Users.Add(user);
    await _dbContext.SaveChangesAsync();
}
```



```
namespace JWTAuth.Controllers
{
    [Route("[controller]/[action]")]
    [ApiController]
    public class AuthController : ControllerBase
    {
        private readonly IAuthService _authService;

        public AuthController(IAuthService authService)
        {
            _authService = authService;
        }

        // POST: auth/login
        [AllowAnonymous]
        [HttpPost]
        public async Task<ActionResult> Login([FromBody] LoginUser user)
        {
            if (String.IsNullOrEmpty(user.UserName))
            {
                return BadRequest(new { message = "Email address needs to entered" });
            }
            else if (String.IsNullOrEmpty(user.Password))
            {
                return BadRequest(new { message = "Password needs to entered" });
            }
        }
    }
}
```

