# 5. CORS and JWT Authentication

## Task 5: CORS and JWT Authentication

### Objectives:  
- Explain and configure CORS  
- Setup JWT Authentication using Bearer tokens

### 1. What is CORS?  
Cross-Origin Resource Sharing lets frontend apps on different domains call APIs.

### 2. Enable CORS  
```csharp  
builder.Services.AddCors(options =>  
{  
 options.AddPolicy("AllowAll", builder => builder.AllowAnyOrigin().AllowAnyMethod().AllowAnyHeader());  
});  
app.UseCors("AllowAll");  
```

### 3. Add JWT Authentication  
Install: `Microsoft.AspNetCore.Authentication.JwtBearer`

### 4. In Program.cs or Startup.cs  
```csharp  
builder.Services.AddAuthentication(...).AddJwtBearer(...);  
app.UseAuthentication();  
```

### 5. Create AuthController  
```csharp  
[AllowAnonymous]  
[Route("auth")]  
public class AuthController : ControllerBase  
{  
 [HttpGet]  
 public string GetToken()  
 {  
 // returns JWT with user role  
 }  
}  
```

### 6. Protect Controller with `[Authorize(Roles = "Admin")]`  
Test in Postman using `Authorization` header: `Bearer <token>`

### 7. Handle Expiry and Role Mismatch  
- Use roles in `claims`  
- Token expires in 2 min (adjustable)

