حتما! اینجا یک ساختار پوشه‌ای کامل و استاندارد برای پروژه ASP.NET Core احراز هویت (Authorization\\_Login\\_Asp.Net) همراه با سرویس لاگینگ (Serilog) و لایه‌بندی دقیق Domain، Application، Infrastructure، API و Tests آوردم.

---

# ساختار پوشه‌بندی کامل پروژه Authorization\\_Login\\_Asp.Net

```

Authorization\_Login\_Asp.Net

│

├── Authorization\_Login\_Asp.Net.Domain # لایه دامنه (Domain Layer)

│ ├── Entities # مدل‌های اصلی دامنه (Entities)

│ │ ├── User.cs

│ │ ├── Role.cs

│ │ ├── Permission.cs

│ │ ├── RolePermission.cs

│ │ └── RefreshToken.cs

│ │

│ ├── Enums # Enumهای دامنه

│ │ ├── UserRole.cs

│ │ ├── PermissionType.cs

│ │ └── RefreshTokenStatus.cs

│ │

│ ├── ValueObjects # Value Objectها (مانند Email)

│ │ └── Email.cs

│ │

│ └── Exceptions # Exceptionهای دامنه (DomainException)

│ └── DomainException.cs

│

├── Authorization\_Login\_Asp.Net.Application # لایه کاربردی (Application Layer)

│ ├── DTOs # مدل‌های داده ورودی/خروجی (Data Transfer Objects)

│ │ ├── RegisterRequest.cs

│ │ ├── LoginRequest.cs

│ │ ├── RefreshTokenRequest.cs

│ │ └── AuthResponse.cs

│ │

│ ├── Interfaces # اینترفیس‌ها برای سرویس‌ها، ریپازیتوری‌ها و UnitOfWork

│ │ ├── IUserService.cs

│ │ ├── IJwtService.cs

│ │ ├── IRefreshTokenService.cs

│ │ ├── IRepository.cs # اینترفیس پایه ریپازیتوری

│ │ ├── IUserRepository.cs

│ │ ├── IRoleRepository.cs

│ │ ├── IPermissionRepository.cs

│ │ ├── IRolePermissionRepository.cs

│ │ ├── IRefreshTokenRepository.cs

│ │ └── IUnitOfWork.cs

│ │

│ ├── Validators # ولیدیتورهای FluentValidation

│ │ ├── RegisterRequestValidator.cs

│ │ ├── LoginRequestValidator.cs

│ │ └── RefreshTokenRequestValidator.cs

│ │

│ ├── Exceptions # استثناهای اعتبارسنجی (ValidationException)

│ │ └── ValidationException.cs

│ │

│ └── Mappings # پروفایل‌های AutoMapper

│ └── MappingProfile.cs

│

├── Authorization\_Login\_Asp.Net.Infrastructure # لایه زیرساخت (Infrastructure Layer)

│ ├── Data # EF Core DbContext و پیکربندی‌های مدل

│ │ ├── AppDbContext.cs

│ │ └── Configurations

│ │ ├── UserConfiguration.cs

│ │ ├── RoleConfiguration.cs

│ │ ├── PermissionConfiguration.cs

│ │ ├── RolePermissionConfiguration.cs

│ │ └── RefreshTokenConfiguration.cs

│ │

│ ├── Services # پیاده‌سازی سرویس‌های لایه Application

│ │ ├── UserService.cs

│ │ ├── JwtService.cs

│ │ ├── RefreshTokenService.cs

│ │ ├── PermissionService.cs

│ │ └── LoggingService.cs # سرویس اختصاصی لاگینگ (اختیاری)

│ │

│ ├── Repositories # پیاده‌سازی ریپازیتوری‌ها (در صورت جدا کردن)

│ │ ├── UserRepository.cs

│ │ ├── RoleRepository.cs

│ │ ├── PermissionRepository.cs

│ │ ├── RolePermissionRepository.cs

│ │ └── RefreshTokenRepository.cs

│ │

│ ├── Extensions # اکستنشن‌های DI و سرویس‌ها

│ │ └── ServiceCollectionExtensions.cs

│ │

│ └── Configurations # تنظیمات برنامه (مثلاً JWT)

│ └── JwtSettings.cs

│

├── Authorization\_Login\_Asp.Net.Api # لایه API (Web API)

│ ├── Controllers # کنترلرهای WebAPI

│ │ ├── AuthController.cs

│ │ └── AdminController.cs

│ │

│ ├── Middleware # میان‌افزارها (Middleware)

│ │ └── ExceptionMiddleware.cs

│ │

│ ├── Filters # فیلترها (مثلاً فیلتر اعتبارسنجی مدل)

│ │ └── ValidationFilter.cs

│ │

│ ├── Authorization # پیاده‌سازی Authorization بر پایه Claims

│ │ ├── PermissionRequirement.cs

│ │ ├── PermissionHandler.cs

│ │ └── AuthorizationPolicies.cs

│ │

│ ├── Program.cs # نقطه شروع برنامه، DI، Middleware، JWT و Serilog

│ ├── appsettings.json # تنظیمات عمومی برنامه (اتصال دیتابیس، JWT و Serilog)

│ └── appsettings.Development.json # تنظیمات محیط توسعه

│

├── Authorization\_Login\_Asp.Net.Tests # پوشه تست‌ها

│ ├── UnitTests # تست‌های واحد (Unit Tests)

│ │ ├── AuthControllerTests.cs

│ │ ├── UserServiceTests.cs

│ │ └── RefreshTokenServiceTests.cs

│ │

│ └── TestHelpers # ابزارها و کلاس‌های کمکی برای تست

│ └── FakeAppDbContext.cs

│

├── docker-compose.yml # کانفیگ Docker جهت راه‌اندازی دیتابیس SQL Server محلی

└── Authorization\_Login\_Asp.Net.sln # فایل Solution

```

---

# نکات تکمیلی در ساختار:

\* \*\*LoggingService.cs\*\* داخل Infrastructure/Services:

پیاده‌سازی کلاس لاگینگ اختصاصی (در صورت نیاز) با استفاده از `ILogger<T>` که خودش توسط Serilog کانفیگ شده است.

\* \*\*ServiceCollectionExtensions.cs\*\*:

ثبت همه سرویس‌ها، ریپازیتوری‌ها، UnitOfWork، و Logging در DI کانتینر، به طوری که در Program.cs فقط کافی باشد یک خط فراخوانی شود.

\* \*\*Program.cs\*\*:

کانفیگ کامل Serilog (خواندن تنظیمات از appsettings.json)، راه‌اندازی JWT Authentication و Authorization و افزودن Middlewareهای سفارشی مثل ExceptionMiddleware.

Authorization Login Asp.Net/

├── Api/ # لایه API

│ ├── Controllers/ # کنترلرهای API

│ ├── Authorization/ # تنظیمات احراز هویت

│ ├── Filters/ # فیلترهای API

│ ├── Middleware/ # میدلورهای API

│ └── Startup.cs # تنظیمات راه‌اندازی API

│

├── Application/ # لایه منطق کسب و کار

│ ├── Interfaces/ # رابط‌های سرویس

│ ├── Validators/ # اعتبارسنجی‌ها

│ ├── DTOs/ # اشیاء انتقال داده

│ ├── Mappings/ # تنظیمات AutoMapper

│ └── Exceptions/ # استثناهای لایه برنامه

│

├── Domain/ # لایه دامنه

│ ├── Entities/ # موجودیت‌های دامنه

│ ├── ValueObjects/ # اشیاء ارزش

│ ├── Events/ # رویدادهای دامنه

│ ├── Common/ # کلاس‌های مشترک

│ ├── Specifications/ # مشخصات دامنه

│ ├── Enums/ # شمارش‌ها

│ └── Exceptions/ # استثناهای دامنه

│

├── Infrastructure/ # لایه زیرساخت

│ ├── Data/ # دسترسی به داده

│ ├── Services/ # سرویس‌های زیرساخت

│ ├── Middleware/ # میدلورها

│ ├── Extensions/ # متدهای توسعه

│ ├── Configurations/ # تنظیمات

│ ├── Caching/ # مدیریت کش

│ ├── Cache/ # پیاده‌سازی کش

│ ├── HealthChecks/ # بررسی سلامت

│ ├── Repositories/ # مخازن داده

│ ├── Security/ # امنیت

│ └── Logging/ # لاگینگ

│

├── Properties/ # تنظیمات پروژه

│ └── launchSettings.json # تنظیمات راه‌اندازی

│

├── .github/ # تنظیمات GitHub

│ └── workflows/ # تنظیمات CI/CD

│

├── Program.cs # نقطه ورود برنامه

├── appsettings.json # تنظیمات برنامه

├── appsettings.Development.json # تنظیمات محیط توسعه

└── Authorization Login Asp.Net.csproj # فایل پروژه

```

### توضیح پوشه‌های اصلی:

1. \*\*پوشه `Api/`\*\*:

- `Controllers/`: کنترلرهای REST API

- `Authorization/`: تنظیمات JWT و احراز هویت

- `Filters/`: فیلترهای API مانند اعتبارسنجی

- `Middleware/`: میدلورهای مخصوص API

- `Startup.cs`: تنظیمات راه‌اندازی API

2. \*\*پوشه `Application/`\*\*:

- `Interfaces/`: رابط‌های سرویس‌های برنامه

- `Validators/`: اعتبارسنجی‌های FluentValidation

- `DTOs/`: اشیاء انتقال داده

- `Mappings/`: تنظیمات AutoMapper

- `Exceptions/`: استثناهای مخصوص لایه برنامه

3. \*\*پوشه `Domain/`\*\*:

- `Entities/`: موجودیت‌های اصلی دامنه

- `ValueObjects/`: اشیاء ارزش غیرقابل تغییر

- `Events/`: رویدادهای دامنه

- `Common/`: کلاس‌های مشترک دامنه

- `Specifications/`: مشخصات و قوانین دامنه

- `Enums/`: شمارش‌های دامنه

- `Exceptions/`: استثناهای مخصوص دامنه

4. \*\*پوشه `Infrastructure/`\*\*:

- `Data/`: دسترسی به داده و Entity Framework

- `Services/`: پیاده‌سازی سرویس‌های زیرساخت

- `Middleware/`: میدلورهای سراسری

- `Extensions/`: متدهای توسعه

- `Configurations/`: تنظیمات برنامه

- `Caching/`: مدیریت کش

- `HealthChecks/`: بررسی سلامت سرویس‌ها

- `Repositories/`: مخازن داده

- `Security/`: امنیت و رمزنگاری

- `Logging/`: سیستم لاگینگ

5. \*\*فایل‌های اصلی\*\*:

- `Program.cs`: نقطه ورود و تنظیمات اصلی برنامه

- `appsettings.json`: تنظیمات برنامه

- `appsettings.Development.json`: تنظیمات محیط توسعه

- `Authorization Login Asp.Net.csproj`: فایل پروژه و وابستگی‌ها

این ساختار از معماری Clean Architecture پیروی می‌کند و لایه‌های مختلف را به خوبی از هم جدا کرده است. هر لایه مسئولیت‌های مشخصی دارد و وابستگی‌ها به سمت داخل (به سمت لایه دامنه) هستند.

Authorization Login Asp.Net/

├── Api/ # لایه API

│ ├── Controllers/ # کنترلرهای API

│ ├── Authorization/ # تنظیمات احراز هویت

│ ├── Filters/ # فیلترهای API

│ ├── Middleware/ # میدلورهای API

│ └── Startup.cs # تنظیمات راه‌اندازی API

│

├── Application/ # لایه منطق کسب و کار

│ ├── Interfaces/ # رابط‌های سرویس

│ │ └── IRepository.cs # رابط مخزن داده

│ ├── Validators/ # اعتبارسنجی‌ها

│ ├── DTOs/ # اشیاء انتقال داده

│ ├── Mappings/ # تنظیمات AutoMapper

│ │ └── MappingProfile.cs # پروفایل نگاشت

│ └── Exceptions/ # استثناهای لایه برنامه

│ └── NotFoundException.cs # استثناهای مختلف

│

├── Domain/ # لایه دامنه

│ ├── Entities/ # موجودیت‌های دامنه

│ │ ├── User.cs # مدل کاربر

│ │ ├── Role.cs # مدل نقش

│ │ └── Permission.cs # مدل دسترسی

│ ├── ValueObjects/ # اشیاء ارزش

│ │ └── Email.cs # شیء ارزش ایمیل

│ ├── Events/ # رویدادهای دامنه

│ │ ├── DomainEvent.cs # رویداد پایه

│ │ └── UserEvents/ # رویدادهای کاربر

│ │ └── UserCreatedEvent.cs

│ ├── Common/ # کلاس‌های مشترک

│ │ ├── BaseEntity.cs # موجودیت پایه

│ │ └── Result.cs # کلاس نتیجه

│ ├── Specifications/ # مشخصات دامنه

│ │ └── BaseSpecification.cs # مشخصات پایه

│ ├── Enums/ # شمارش‌ها

│ │ ├── UserRole.cs # نقش‌های کاربری

│ │ └── PermissionType.cs # انواع دسترسی

│ └── Exceptions/ # استثناهای دامنه

│ └── DomainException.cs # استثنای دامنه

│

├── Infrastructure/ # لایه زیرساخت

│ ├── Data/ # دسترسی به داده

│ │ ├── AppDbContext.cs # کانتکست دیتابیس

│ │ └── Configurations/ # تنظیمات EF Core

│ │ └── UserConfiguration.cs

│ ├── Services/ # سرویس‌های زیرساخت

│ │ └── LoggingService.cs # سرویس لاگینگ

│ ├── Security/ # امنیت

│ │ ├── JwtTokenGenerator.cs # تولید توکن JWT

│ │ ├── PasswordHasher.cs # هش رمز عبور

│ │ └── RateLimiter.cs # محدودیت نرخ درخواست

│ ├── Middleware/ # میدلورها

│ ├── Extensions/ # متدهای توسعه

│ ├── Configurations/ # تنظیمات

│ ├── Caching/ # مدیریت کش

│ ├── Cache/ # پیاده‌سازی کش

│ ├── HealthChecks/ # بررسی سلامت

│ ├── Repositories/ # مخازن داده

│ └── Logging/ # لاگینگ

│

├── Properties/ # تنظیمات پروژه

│ └── launchSettings.json # تنظیمات راه‌اندازی

│

├── .github/ # تنظیمات GitHub

│ └── workflows/ # تنظیمات CI/CD

│

├── Program.cs # نقطه ورود برنامه

├── appsettings.json # تنظیمات برنامه

├── appsettings.Development.json # تنظیمات محیط توسعه

└── Authorization Login Asp.Net.csproj # فایل پروژه

```

### توضیح کلاس‌های اصلی:

1. \*\*لایه Domain\*\*:

- `User.cs`: مدل اصلی کاربر با تمام ویژگی‌ها

- `Role.cs`: مدل نقش‌های سیستم

- `Permission.cs`: مدل دسترسی‌ها

- `Email.cs`: شیء ارزش برای ایمیل

- `BaseEntity.cs`: کلاس پایه برای موجودیت‌ها

2. \*\*لایه Application\*\*:

- `IRepository.cs`: رابط مخزن داده

- `MappingProfile.cs`: تنظیمات AutoMapper

- `NotFoundException.cs`: استثناهای برنامه

3. \*\*لایه Infrastructure\*\*:

- `AppDbContext.cs`: کانتکست Entity Framework

- `JwtTokenGenerator.cs`: تولید توکن JWT

- `PasswordHasher.cs`: هش رمز عبور

- `RateLimiter.cs`: محدودیت نرخ درخواست

- `LoggingService.cs`: سرویس لاگینگ

4. \*\*لایه API\*\*:

- `Startup.cs`: تنظیمات راه‌اندازی API