**🧠 اول از همه: ApplicationService چیه و چرا لازمه؟**

لایه‌ی ApplicationService در معماری چند‌لایه (Clean / Onion / DDD)  
پل بین **Domain (Entities + DAL)** و **Presentation (Web API / UI)** هست.

یعنی چی؟  
✅ ریپازیتوری فقط داده رو از دیتابیس می‌خونه یا ذخیره می‌کنه.  
✅ ولی ApplicationService تصمیم می‌گیره *چه زمانی، چطور و با چه منطقی* اون داده‌ها رو بخونه یا تغییر بده.

**⚙️ وظایف ApplicationService**

| **وظیفه** | **توضیح** |
| --- | --- |
| هماهنگی بین Repositoryها | اگر چند جدول درگیر باشن (مثل ثبت سفارش + پرداخت) |
| اعمال منطق تجاری | مثل “کاربر بدون اشتراک نمی‌تونه ثبت‌نام کلاس کنه” |
| مدیریت تراکنش‌ها | با استفاده از UnitOfWork |
| تبدیل داده‌ها | مثلاً DTO ↔ Entity برای تبادل با API |
| امنیت / مجوزها | بررسی نقش و دسترسی قبل از اجرای عملیات |

**🧩 نوع پروژه‌ای که باید اضافه کنی**

🔹 **Class Library (.NET Class Library)**  
اسمش رو بذار مثلاً:

GymSite.ApplicationService

🔸 دلیلش:

* ApplicationService خودش مستقیماً **UI یا API نداره**
* فقط شامل **منطق، سرویس‌ها، DTOها و Mapperها** هست
* به‌صورت **DLL** در پروژه Web API استفاده میشه

**🔗 Referenceها (وابستگی‌ها)**

بعد از ایجاد پروژه، باید ریفرنس‌های زیر رو اضافه کنی:

| **از** | **به** | **چرا** |
| --- | --- | --- |
| GymSite.ApplicationService | GymSite.DAL | چون سرویس‌ها از ریپازیتوری و UnitOfWork استفاده می‌کنن |
| GymSite.ApplicationService | GymSite.Entities | چون DTOها و Mapها با Entityها در ارتباط هستن |

**📦 پکیج‌هایی که باید نصب بشن (اختیاری اما پیشنهادی)**

| **پکیج** | **کاربرد** | **دستور نصب** |
| --- | --- | --- |
| **AutoMapper** | برای تبدیل Entity ↔ DTO | Install-Package AutoMapper  Install-Package AutoMapper.Extensions.Microsoft.DependencyInjection |
| **FluentValidation** | برای اعتبارسنجی داده‌ها (مثلاً فرم ثبت‌نام) | Install-Package FluentValidation |
| **Microsoft.Extensions.DependencyInjection.Abstractions** | برای تزریق وابستگی‌ها | Install-Package Microsoft.Extensions.DependencyInjection.Abstractions |

⚠️ EF Core نیازی به نصب مجدد نداره چون در DAL نصب شده و ApplicationService فقط از DAL استفاده می‌کنه.

**🧱 ساختار پیشنهادی پوشه‌ها در این پروژه**

GymSite.ApplicationService

│

├── Interfaces

│ ├── IPersonService.cs

│ ├── IOrderService.cs

│ └── ...

│

├── Services

│ ├── PersonService.cs

│ ├── OrderService.cs

│ └── ...

│

├── DTOs

│ ├── PersonDto.cs

│ ├── OrderDto.cs

│ └── ...

│

└── Mappings

└── MappingProfile.cs

**🚀 فرآیند اجرای یک عملیات در این لایه**

مثلاً وقتی کاربر در API ثبت‌نام می‌کنه:

Controller → ApplicationService(PersonService.Register) → UnitOfWork.PersonRepository.Save()

ApplicationService مسئول منطق ثبت‌نام هست، نه دیتابیس.  
Repository فقط عمل درج داده رو انجام می‌ده.

**✅ جمع‌بندی سریع**

| **بخش** | **توضیح** |
| --- | --- |
| نوع پروژه | Class Library (.NET) |
| ریفرنس‌ها | DAL و Entities |
| هدف | منطق تجاری + هماهنگی بین Repositoryها |
| پکیج‌ها | AutoMapper, FluentValidation |
| ساختار | Interfaces, Services, DTOs, Mappings |