بله، دقیقاً! 🚀 برای پیاده‌سازی **CQRS** در پروژه‌ی **PhoneBook** با **جداسازی لایه‌ها**، بهتر است یک **پوشه‌ی جدید به نام Application** اضافه کنی. این پوشه مسئول **مدیریت لاجیک پردازش درخواست‌ها** خواهد بود و به‌عنوان **لایه‌ی واسط بین Core و Infrastructure** عمل می‌کند.

🔹 **ساختار پیشنهادی پروژه‌ی تو با CQRS و معماری لایه‌ای** 👇

📂 **PhoneBook** *(پروژه اصلی)*  
│── 📁 **Core** *(ساختار داده‌ها و مدل‌های اصلی)*  
│ ├── 📁 **Entities** *(مدل‌های User و Contact)*  
│ ├── 📁 **Interfaces** *(رابط‌های مرتبط با ریپازیتوری‌ها)*  
│── 📁 **Application** *(مدیریت عملیات CQRS)*  
│ ├── 📁 **Commands** *(مدیریت عملیات نوشتن)*  
│ ├── 📁 **Queries** *(مدیریت عملیات خواندن)*  
│── 📁 **Infrastructure** *(دسترسی به دیتابیس و سرویس‌ها)*  
│ ├── 📁 **Persistence** *(ارتباط با DbContext و ذخیره اطلاعات)*  
│ ├── 📁 **Repositories** *(مدیریت ریپازیتوری‌ها)*  
│── 📁 **Web** *(UI و API که به Infrastructure متصل می‌شود)*  
│ ├── 📁 **Controllers** *(ارتباط با Application)*

✅ **چرا Application را جدا کنیم؟**  
🔹 Application **مسئول پردازش درخواست‌های خواندن (Query) و نوشتن (Command) است.**  
🔹 Web **فقط API و UI را مدیریت می‌کند و مستقیماً با Application ارتباط دارد.**  
🔹 Infrastructure **ارتباط مستقیم با دیتابیس و ریپازیتوری‌ها را مدیریت می‌کند.**  
🔹 Core **فقط ساختار اصلی داده‌ها را نگه می‌دارد و هیچ وابستگی‌ای به Web ندارد.**

[ Web Layer ] → ارتباط فقط با → [ Application Layer ] → ارتباط فقط با → [ Core Layer ]

[ Infrastructure Layer ] → ارتباط فقط با → [ Core Layer و Application Layer ]

**ارتباط لایه‌ها طبق Clean Architecture:**  
✔️ **Web فقط با Application ارتباط دارد و هیچ وابستگی‌ای به Core یا Infrastructure ندارد.**  
✔️ **Application فقط با Core ارتباط دارد، و Queries و Commands را مدیریت می‌کند.**  
✔️ **Infrastructure فقط با Core و Application ارتباط دارد و مسئول دیتابیس و Repositories است.**

📌 **این مدل معماری در تمام پروژه‌هایی که از CQRS و Clean Architecture استفاده می‌کنند، یکسان است.**