



صورت مسئله

عنوان پروژه

سیستم مدیریت سفارشات (Order Management System)

شرح کلی

شما باید یک **ASP.NET Core Web API** طراحی و پیاده سازی کنید که امکان مدیریت مشتریان، سفارشات و آیتم های سفارش را فراهم کند.

۱. مدل داده

پروژه باید شامل موجودیت های زیر باشد:

- **Customer**
- **Order**
- **OrderItem**



۲. امکانات API

Customer APIs

- `GET /api/customers` لیست مشتریان
- `GET /api/customers/{id}` جزئیات یک مشتری + سفارشاتش
- `POST /api/customers` ایجاد مشتری جدید
- `PUT /api/customers/{id}` ویرایش مشتری
- `DELETE /api/customers/{id}` حذف مشتری (اگر سفارشی داشته باشد، حذف نشود)

Order APIs

- `GET /api/orders` لیست سفارش‌ها
- `GET /api/orders/{id}` جزئیات سفارش + آیتم‌ها
- `POST /api/orders` ایجاد سفارش (به همراه آیتم‌ها)
- `PUT /api/orders/{id}` ویرایش سفارش
- `DELETE /api/orders/{id}` حذف سفارش

۳. الزامات فنی

۱. استفاده از `ASP.NET Core Web API` نسخه ۶ یا بالاتر.
۲. استفاده از `Entity Framework Core` برای مدیریت دیتابیس.
۳. استفاده از `Migration` برای ساخت دیتابیس.
۴. پیاده‌سازی `DTOs` و جلوگیری از بازگشت مستقیم `Entity`.
۵. مدیریت `Validation` مثلاً ایمیل یونیک، تعداد و قیمت مثبت.
۶. مدیریت `Transaction` در ثبت سفارش و آیتم‌ها.
۷. برگرداندن `Status Code` مناسب (۲۰۰)، (۲۰۱)، (۴۰۰)، (۴۰۴) و...).
۸. پیاده‌سازی معماری تمیز (حداقل در سطح `Repository / Service Layer`).
۹. مستندسازی و امکان تست با `Swagger` یا `Postman`.



۴. معیارهای ارزیابی

- طراحی صحیح مدل داده و روابط.
- توانایی کار با **EF Core** و **Migration**.
- رعایت اصول کدنویسی تمیز و لایه‌بندی پروژه.
- رعایت اصول **REST API**.
- مدیریت خطاها و **Validation**.
- خروجی تمیز و قابل تست بودن پروژه.

۵. مراحل اختیاری (امتیاز اضافه)

- اضافه کردن **Search** و **Pagination** در لیست سفارش‌ها.
- گزارش‌گیری $\rightarrow$ `/api/reports/top-customers`: نمایش مشتریانی که بیشترین مبلغ خرید داشته‌اند.