

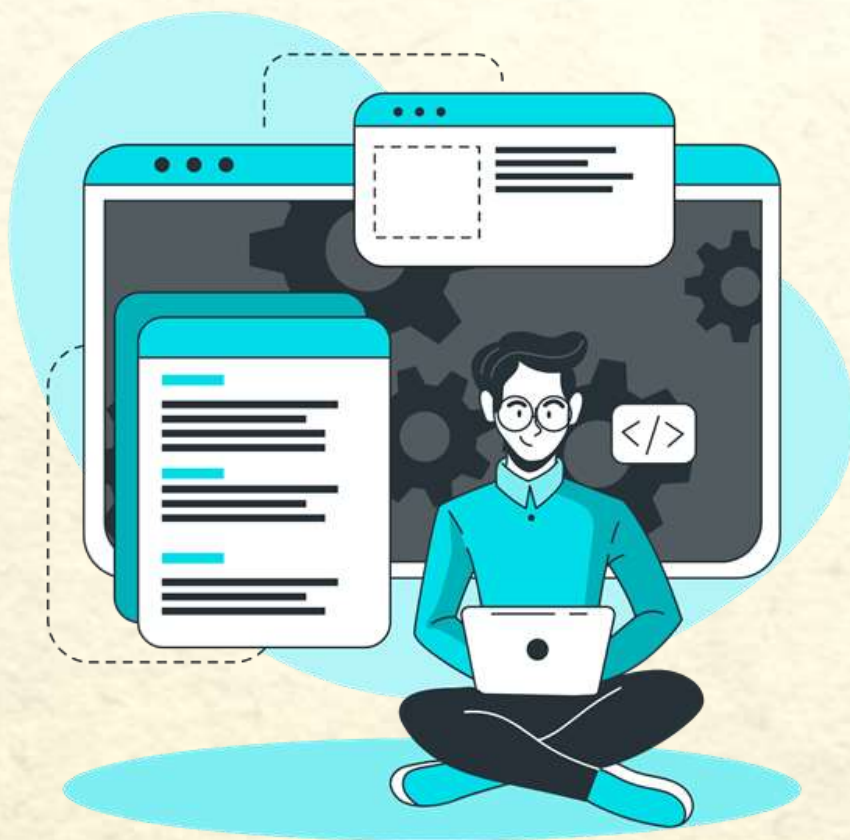
مکتب شریف

اولین بوتکمپ آموزشی - استخدامی ایران



C#/.Net

مکتب ۱۴۰





Cw-10

تمرین به صورت انفرادی می باشد. و اگر سوالی براتون پیش امد اول از هم گروهی هاتون سپس از مربی میتوانید کمک بگیرید و حتما باید امروز تمرین تمام شود .

در این پروژه باید یک سیستم یکپارچه مدیریت کاربر و فروشگاه را در قالب یک پروژه Console App پیاده سازی کنید.

فاز ۱ – لایه Repository و ذخیره سازی JSON

برای هر موجودیت Repository جدا (بدون Generic) بسازید:

IUserRepository / JsonUserRepository

JsonProductRepository – IProductRepository

JsonOrderRepository / IOrderRepository

اطلاعات باید در فایل های زیر ذخیره شوند:

users.json - products.json - orders.json (وظیفه Repository فقط خواندن و نوشتن فایل باشد).

فاز ۲ – Service Layer + Dependency Injection

کلاس های زیر را ایجاد کنید:

UserService -

ProductService -

OrderService -

شرایط:

- Repository ها باید از طریق سازنده تزریق شوند.

- داخل Service ها new کردن ممنوع است.

- منطق تجاری (Business Logic) داخل Service ها باشد.

فاز ۳ - اعتبارسنجی حرفه‌ای رمز عبور + Custom Exception

در این فاز باید قوانین رمز عبور را از کلاس UserService جدا کنید تا ساختار پروژه تمیزتر، قابل توسعه‌تر و مطابق با اصول SOLID باشد.

مراحل انجام کار:

(۱) یک Interface به نام IPasswordPolicy طراحی کنید که متدی مانند زیر

داشته باشد: `Validate(string username, string password)`

یک کلاس StrongPasswordPolicy بسازید که IPasswordPolicy را پیاده‌سازی کند.

(۲) قوانین رمز عبور:

- حداقل ۸ کاراکتر داشته باشد.
- حداقل شامل ۱ حرف بزرگ (A-Z) باشد.
- حداقل شامل ۱ حرف کوچک (a-z) باشد.
- حداقل شامل ۱ عدد (۰-۹) باشد.
- حداقل شامل ۱ کاراکتر خاص مانند ! @ # \$ % باشد.
- رمز عبور نباید شامل Username کاربر باشد.

(۴) برای هر قانونی که رعایت نشد، یک Custom Exception جداگانه ایجاد کنید (PasswordTooShortException, PasswordMissingDigitException, ...).

و در صورت نقض قانون، همان Exception را throw کنید + (به همراه خطای مخصوص خود).

(۵) در کلاس UserService هنگام ثبت نام کاربر، به جای نوشتن if های متعدد، فقط متد Validate را صدا بزنید.

(۶) کلاس UserService باید IPasswordPolicy را از طریق Constructor Injection دریافت کند.

(داخل Service مجاز به new کردن StrongPasswordPolicy نیستید.)

فاز ۴ - سیستم پرداخت

کلاس abstract به نام Payment ایجاد کنید.

کلاس‌های مشتق:

CashPayment - CardPayment - OnlinePayment -

فاز ۵ - مدیریت خطا و Logger

در Repository:

- اگر فایل وجود نداشت، ساخته شود. اگر فایل خالی بود، لیست خالی برگردانده شود. , اگر JSON خراب بود، خطا ثبت شود و برنامه ادامه پیدا کند .
 موارد زیر را ایجاد کنید:

- ILogger - ConsoleLogger - FileLogger

Logger نیز باید از طریق DI تزریق شود.

منوی نهایی برنامه (Console)

(۱) ثبت نام / ورود

(۲) مدیریت کاربران (CRUD)

(۳) مدیریت محصولات (CRUD)

(۴) ثبت سفارش

(۵) پرداخت سفارش

(۶) مشاهده سفارشات من

(۰) خروج