 Logout (پاک‌کردن RefreshToken و Session)

 RefreshToken (گرفتن توکن جدید با RefreshToken)

 Register (در صورت نیاز ثبت‌نام)

 شاید ChangePassword یا ForgotPassword

1. **Captcha Validation**: الان متد ValidateCaptchaAsync همیشه فقط captchaToken رو چک می‌کنه. اگه واقعاً Captcha سرویسی مثل Google reCAPTCHA یا hCaptcha دارید، باید درخواست HTTP به اون سرویس بفرسته.
2. **Lockout Handling**:
   * الان Lockout فقط در سطح نرم‌افزاریه (چک با FailedLoginAttempts).
   * بد نیست کلیدهای LockUntil یا FailedAttempts رو هم تو جدول User نگه داریم → چک سریع‌تر میشه.
3. **Exception vs. Custom Result**:
   * الان اکثر خطاها throw new Exception("...") می‌زنن. بهتره به جای Exception، یک **Result Object** با Status/Code برگردونیم. اینطوری Controller می‌تونه پاسخ مناسب (401, 403, 409) بده.
4. **Claims در JWT**:
   * الان فقط UserId و UserName هست. بهتره Roleها هم در Claim اضافه بشن (برای Authorization راحت‌تر).
5. **Logout**:
   * LogoutAsync همه‌ی RefreshTokenها رو revoke می‌کنه. شاید بخوایم فقط توکن مربوط به همون Session رو باطل کنیم.
6. **RefreshToken Rotation**:
   * روش شما درسته (هر بار یک RefreshToken جدید ساخته میشه).
   * پیشنهاد: قدیمی رو Revoked کنید (انجام شده 👍).

 **Hash & Salt**

* استفاده از SHA256(password + salt) امنیت پایینی داره.
* پیشنهاد: برید روی PBKDF2, bcrypt یا Argon2 (در .NET می‌شه با Rfc2898DeriveBytes).

 **JWT Claims**

* اضافه کردن Roleها و شاید SessionId به Claimها.
* الان فقط UserId و UserName هستن.

 **خطاها**

* ValidateUserCredentialsAsync فقط true/false برمی‌گردونه. بهتره **جزئیات دلیل رد شدن** رو برگردونه (کاربر غیرفعال، پسورد اشتباه، کاربر وجود ندارد).

 **Session Management**

* بهتره RefreshTokenها بر اساس SessionId revoke بشن، نه همه‌ی توکن‌های کاربر.
* اینطوری Logout یک دستگاه، بقیه دستگاه‌ها رو بیرون نمی‌اندازه.

 **Audit Logging**

* فعلاً تو Repository لاگی ثبت نمی‌شه. همه‌ی لاگ‌ها تو Service هست.
* می‌شه سطح Repository هم Log مختصر بزنه (مثلاً ایجاد Session، revoke Token).