

همانطور که می‌دانید Identity، فریمورک نسبتاً جدیدی هست که مایکروسافت برای مدیریت کاربران و احراز هویت آن‌ها معرفی کرده و پیشرفت چشمگیری داشته است. در قسمت IdentityConfig (قسمتی که برای کانفیگ کردن Identity استفاده می‌شود) بخشی قابل تنظیم برای کانفیگ کردن سیاست‌های تعیین پسورد وجود دارد. به‌طور مثال: تعیین حداقل تعداد حروف برای کلمه‌ی عبور، ضرورت کوچک و بزرگ بودن حروف، الزام وجود کاراکتر ویژه.

این نیاز وجود دارد که PasswordValidator موجود در Identity را برای پروژه‌های مختلف سفارشی‌سازی کرد و این امکان فراهم شود که بتوان سیاست‌های کاری شرکت و پروژه را در قالب PasswordValidator اعمال کرد. به عنوان مثال بررسی کنیم اگر پسورد کاربر عدد 12345 وارد شده است، خطا صادر کنیم و اجازه انتساب آن را برای کاربر ندهیم یا منطق‌های دیگری که نیاز داریم. پس در اینجا به وجود یک CustomPasswordValidator نیاز هست است.

```
public class CustomPasswordValidator : PasswordValidator
{
    public override async Task<IdentityResult> ValidateAsync(string pass)
    {
        IdentityResult result = await base.ValidateAsync(pass);
        if (pass.Contains("12345"))
        {
            var errors = result.Errors.ToList();
            errors.Add("Passwords cannot contain numeric sequences");
            result = new IdentityResult(errors);
        }
        return result;
    }
}
```

در قطعه کد بالا یک کلاس ایجاد شد با نام CustomPasswordValidator و از PasswordValidator موجود در فضای نام Microsoft.AspNet.Identity ارث‌بری شد تا ویژگی‌های اصلی این کلاس را به‌صورت ذاتی داشته باشیم و بتوانیم سفارشی‌سازی‌های خودمان را بر روی آن اعمال می‌کنیم. متد ValidateAsync موجود override شده و بر روی ورودی آن شرط‌های پروژه و سیاست‌ها بررسی شده‌اند و در صورت تخلف از قوانین خطا صادر شد.

نقطه‌ی پایان کار اینجاست که در داخل کلاس کانفیگ موجود برای Identity که درون فایل web.config مشخص شده است (در این مثال کلاس IdentityConfig) برای قسمت PasswordValidator، حالا باید از کلاس CustomPasswordValidator، یک شیء جدید ساخته شود:

```
// Configure custom validation logic for passwords
manager.PasswordValidator = new CustomPasswordValidator
{
    RequiredLength = 6,
    RequireNonLetterOrDigit = false,
    RequireDigit = false,
    RequireLowercase = false,
    RequireUppercase = false,
};
```