Kullanıcı girdisi neden kontrol edilir?



www.yzokul.com.tr

Kullanıcı girdisi neden kontrol edilir?



Kullanıcı girdisi, bir kişinin programa yazdığı veya seçtiği her şeydir; örneğin bir ad, bir yaş ya da listeden bir seçim. Bu bilgi programın dışından gelir, bu yüzden içinde ne olacağını önceden bilemezsin. Girdi eksik olabilir, yanlış biçimde olabilir veya aralık dışında olabilir. Neden kontrol edildiğinin kısa yanıtı şudur: program, girdiye güvenmeden önce onun kullanılabilir olduğunu doğrulamalıdır; bu da programı doğru ve güvenli tutar.



Bir program sık sık bilgi ister: bir metin kutusunda ad, bir sayı alanında yaş ya da bir menüden seçim. Kişinin yazdığı veya seçtiği her şeye kullanıcı girdisi denir. Program bu girdiyi alır ve sonra onunla çalışır; örneğin onu saklar ya da onunla hesaplama yapar. Program bu bilgiye bağlı olduğu için önce bilginin mantıklı olduğundan emin olmalıdır.



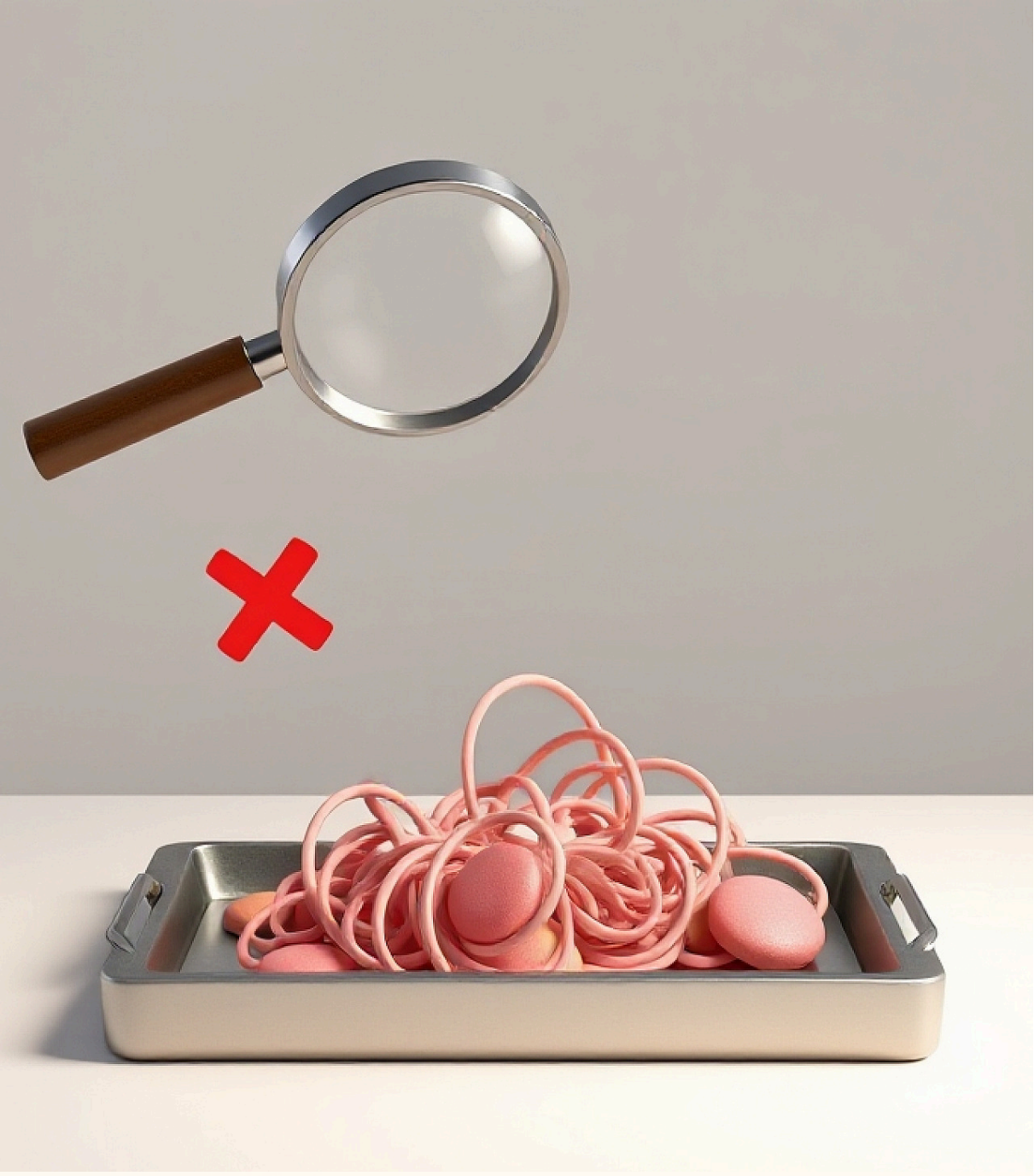
Bir form, bir alanı zorunlu olarak işaretleyebilir; yani program bu değer olmadan işini yapamaz. Kişi alanı boş bırakıp formu gönderirse program, bir şey beklediği yerde hiçbir şey alır. Boş değerle devam etmek yerine bu boşluğu fark etmelidir. Eksik girdi, kontrolün yakaladığı ilk yaygın sorundur.



Bazı alanlar belirli bir biçim bekler; örneğin yaş için bir sayı ya da et işareti içeren bir e-posta adresi. Sayı bekleyen bir alan harf alırsa biçim, programın ihtiyacıyla uyuşmaz. Girdi vardır ama olduğu gibi kullanılamaz. Biçimi kontrol etmek, girdinin programın beklediği şekle sahip olduğunu kullanmadan önce doğrulamak demektir.



Bir değer doğru biçimde olup yine de yanlış olabilir. 500 yaşı bir sayıdır, yani biçim sorunsuzdur, ama hiçbir kişi o yaşta değildir, dolayısıyla değer izin verilen aralığın dışına düşer. Aralık, programın kabul ettiği değerler kümesidir; örneğin 0 ile 120 arasındaki yaşlar. Aralığı kontrol etmek, değerin bu sınırların içinde olduğunu doğrulamak demektir.



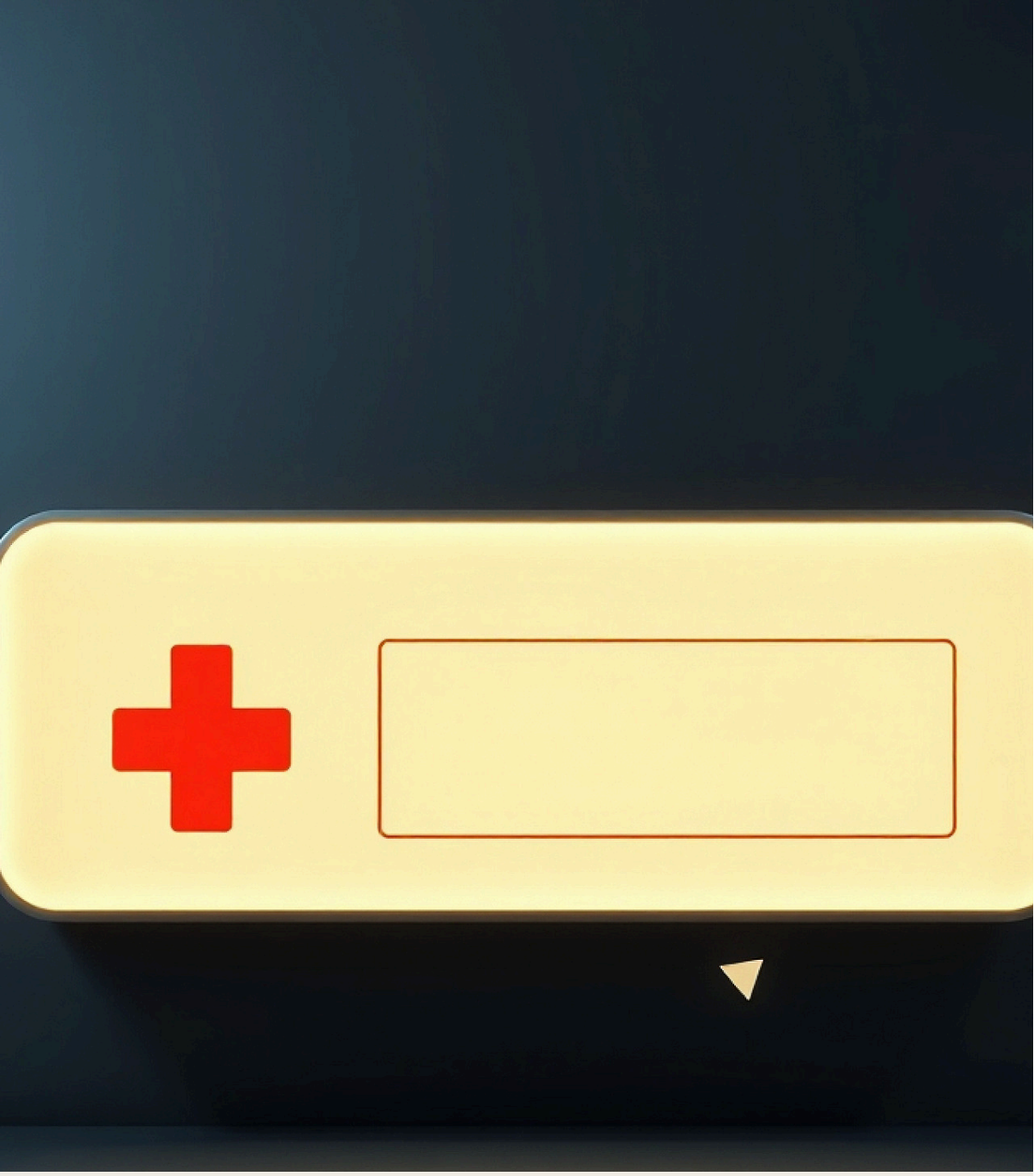
Yaşını soran bir form düşün ve birisi sayı yerine abc harflerini yazsın. Bu küçük örnek sorunları açıkça gösterir: alan artık boş değildir ama biçim yanlıştır ve harflerden geçerli bir aralık okunamaz. Program bunu hemen yakalamalı ve açık bir mesajla yanıt vermelidir. Kullanıcı girdisinin neden kontrol edildiğinin basit örneği budur.



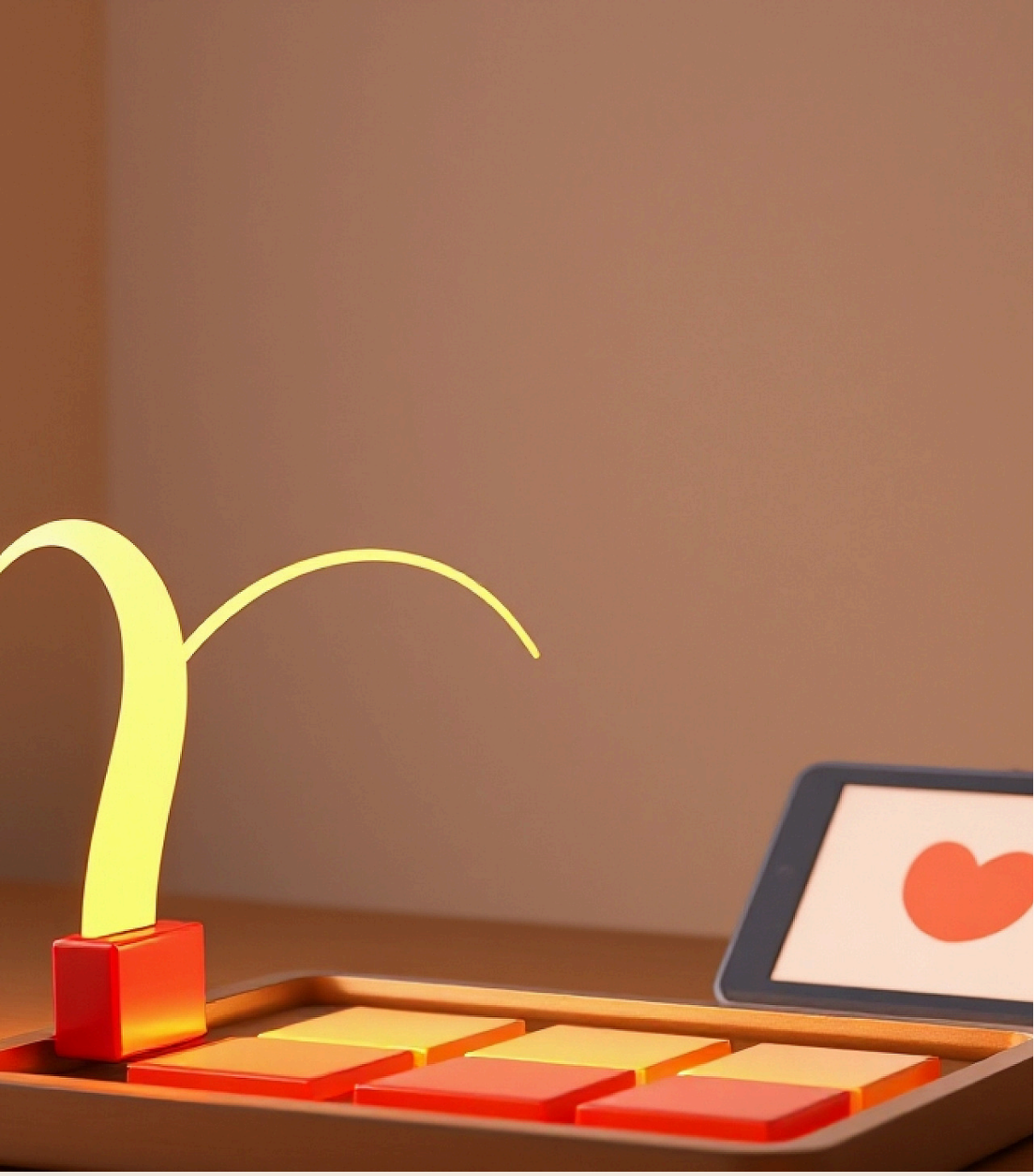
Bir program kötü girdiye güvendiğinde sorunlar gelir. Çökebilir ve çalışmayı durdurabilir, verisine saçmalık kaydedebilir ya da kullanıcıyı şaşırtan yanlış bir sonuç hesaplayabilir. Çökme, programın beklenmedik biçimde, çoğu zaman kullanıcı dostu olmayan bir hatayla durması demektir. Bunların hiçbiri kullanıcının suçu değildir; program yalnızca aldığı şeyi kullanmadan önce kontrol etmeyi başaramamıştır.



Girdiyi kontrol etmenin doğru anı, o geldiği anda, program onu herhangi bir şey için kullanmadan öncedir. Kötü veri girişten geçerse hesaplamalara, kaydedilen kayıtlara ve sonraki ekranlara yayılabilir; orada bulunması daha zor olur. Erken kontrol etmek kötü verinin yayılmasını önler. Kontrolü, her değerin geçmesi gereken bir kapı olarak düşün.



Girdi bir kontrolden geçemezse program bunu kısa ve sade bir mesajla söylemelidir. İyi bir hata mesajı alanı ve sorunu adlandırır; örneğin: lütfen yaşını bir sayı olarak gir. Kullanıcı böylece tahmin etmeden tam olarak neyin yanlış gittiğini bilir. hata gibi belirsiz bir mesaj, kullanıcının nedeni aramasına yol açar ve çoğu zaman vazgeçmesine neden olur.



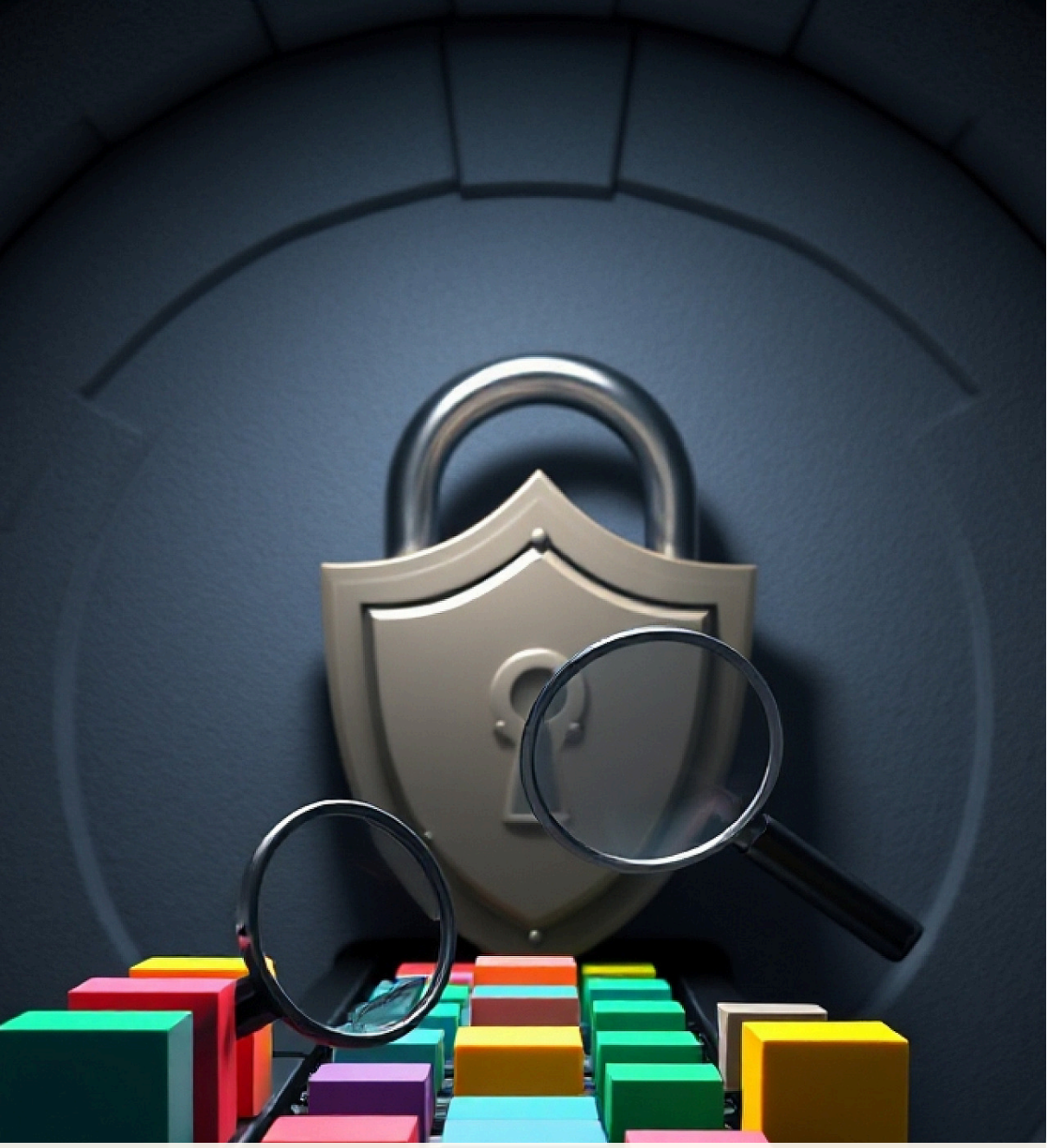
Hatayı gösterdikten sonra program düzeltme istemeli ve kullanıcının yeniden denemesine izin vermelidir. Formun geri kalanını doldurulmuş tut ki kişi baştan başlamasın. Kullanıcı kontrolü elinde tutar ve görevi tamamlayabilir. Girdiyi kontrol etmek ve düzeltme istemek, çıkmaz bir yolu küçük, düzeltilebilir bir ana dönüştürür.



Kontrol etmek yalnızca hataları durdurmakla ilgili değildir; kullanıcıya da yol gösterir. Bir alanın yanındaki kısa bir ipucu, örneğin yalnızca sayılar, kişilere yazmadan önce ne beklendiğini söyler. Açık geri bildirim programı kullanımı kolaylaştırır ve hayal kırıklığını azaltır. İyi kontrol kuralı açıklar, hatayı yakalar ve düzeltmeye işaret eder.



Geçerli girdi, doğru çıktının temelidir. Program aralık içinde bir sayı olarak gerçek bir yaş aldığında doğru sonucu hesaplayıp saklayabilir ve sonraki ekranlar doğru bilgiyi gösterir. Yanlış girdi, programın geri kalanı ne kadar iyi olursa olsun yanlış sonuç üretir. Önce kontrol etmek programı baştan itibaren doğru tutar.



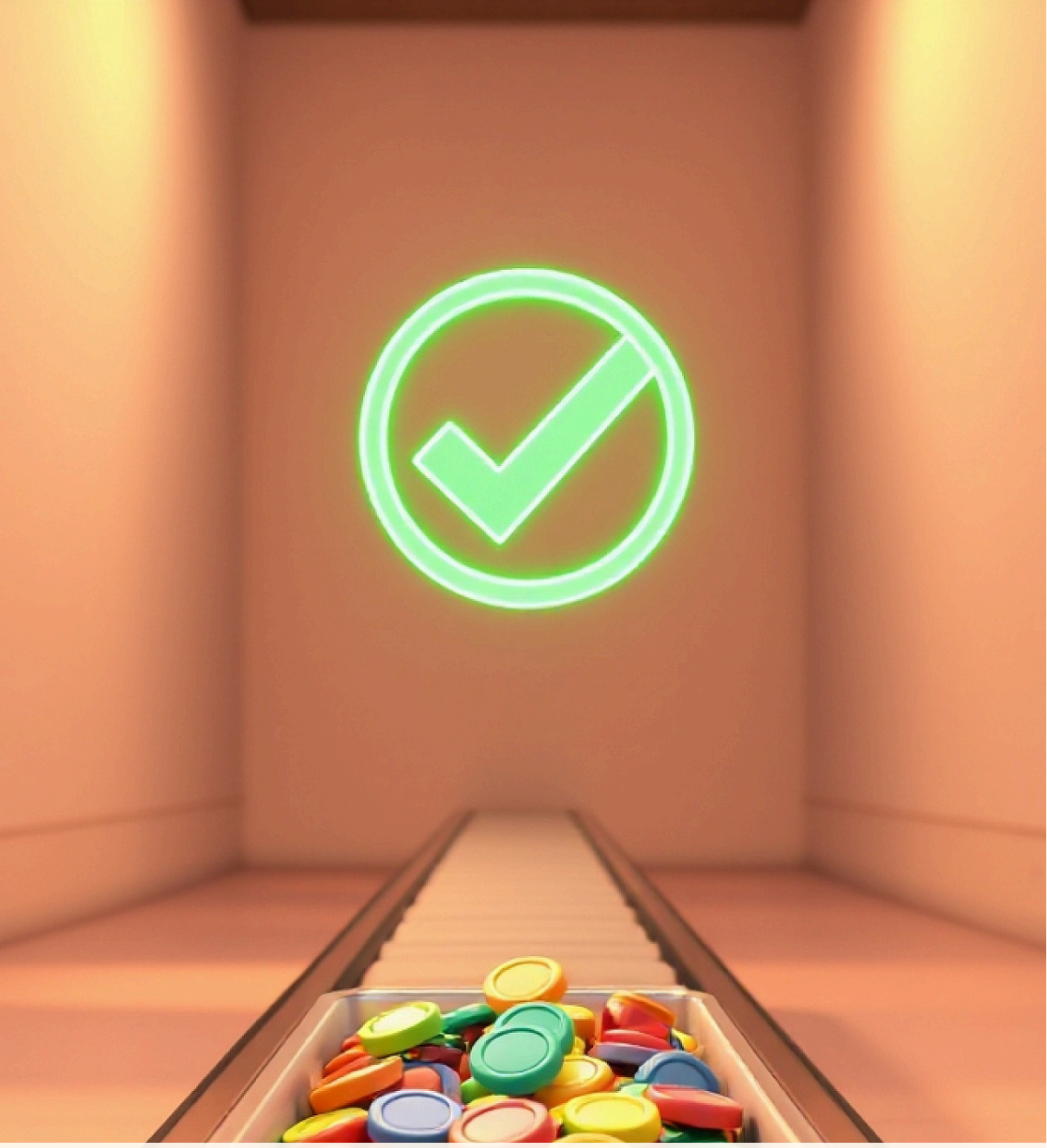
Kontrol edilmeyen girdi gerçek zarara yol açabilir: çökmeler, sonrası için kaydedilen kötü veriler ve kimsenin güvenemeyeceği sonuçlar. Kontrol etmek, her programın, küçük bir programın bile kullanması gereken basit bir güvenlik önlemidir. Az maliyetlidir ve bu sorunların çoğunu önler. Programın dışından gelen girdiyi doğrulanacak bir şey olarak gör, asla varsayılan olarak güvenilecek bir şey olarak görme.



Kontroller sunucuda da yapılabilir; sunucu, verileri saklayan ve kullanıcının ekranından gelen istekleri yanıtlayan bilgisayardır. Arayüz kontrolleri tarayıcıda, yani kullanıcının gördüğü programda yapılır ve hızlı ve kullanıcı dostudur. Sunucu doğrulaması önemli kontrolleri sunucuda tekrarlar, çünkü bir istek arayüzden geçmeden gelebilir. İki katmanlı kontrol programı daha çok korur.



Yeni başlayanlar sık sık girdiye körü körüne güvenir, kontrolü atlar ve sonra programın neden bozulduğunu merak eder. Bir başka yaygın hata, hangi alanın başarısız olduğunu veya ne yazılacağını söylemeyen belirsiz bir hata mesajıdır. Üçüncüsü, yalnızca tarayıcıda kontrol edip sunucuda kontrol etmemektir. Düzeltmeler basittir: her değeri kontrol et, sorunu açıkça anlat ve önemli kontrolleri sunucuda tekrarla.



Kullanıcı girdisi eksik olabilir, yanlış biçimde olabilir veya aralık dışında olabilir; bu yüzden program onu kullanmadan önce kontrol et. Alanı ve sorunu adlandıran açık bir hata mesajı göster, sonra düzeltme iste ve kullanıcının yeniden denemesine izin ver. Ek koruma için önemli kontrolleri sunucu tarafında tekrarla. Bu, programı doğru, güvenli ve kullanımı kolay tutar.