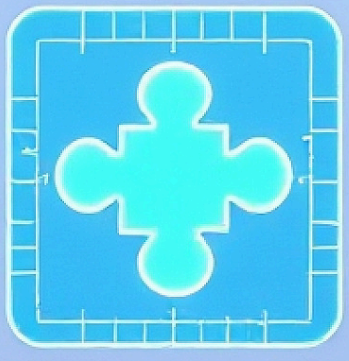
Bir uygulama fikri nasıl çalışan bir programa dönüşür?





www.yzokul.com.tr

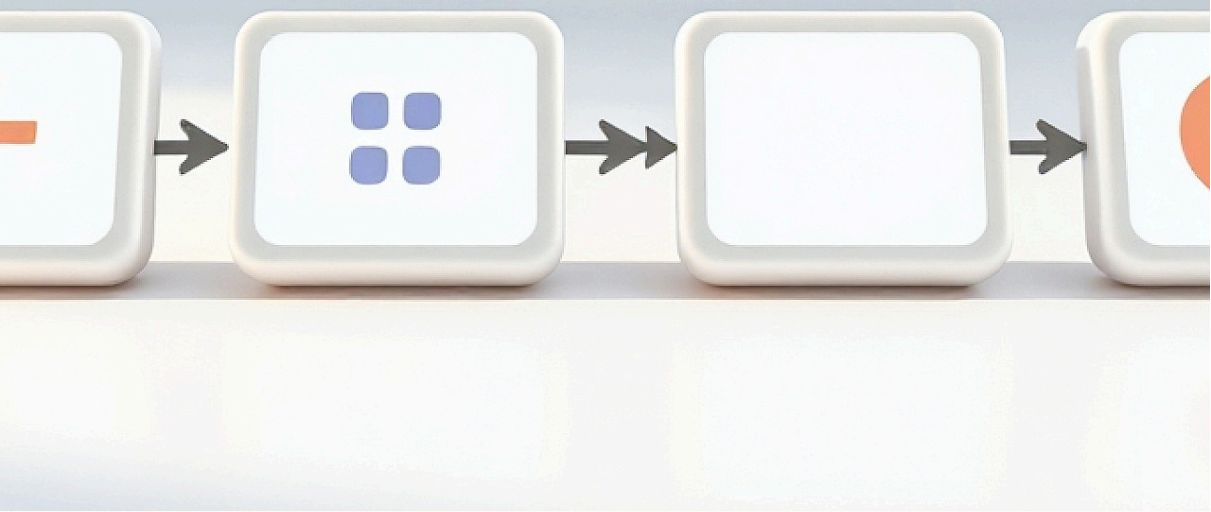
Bir uygulama fikri nasıl çalışan bir programa dönüşür?



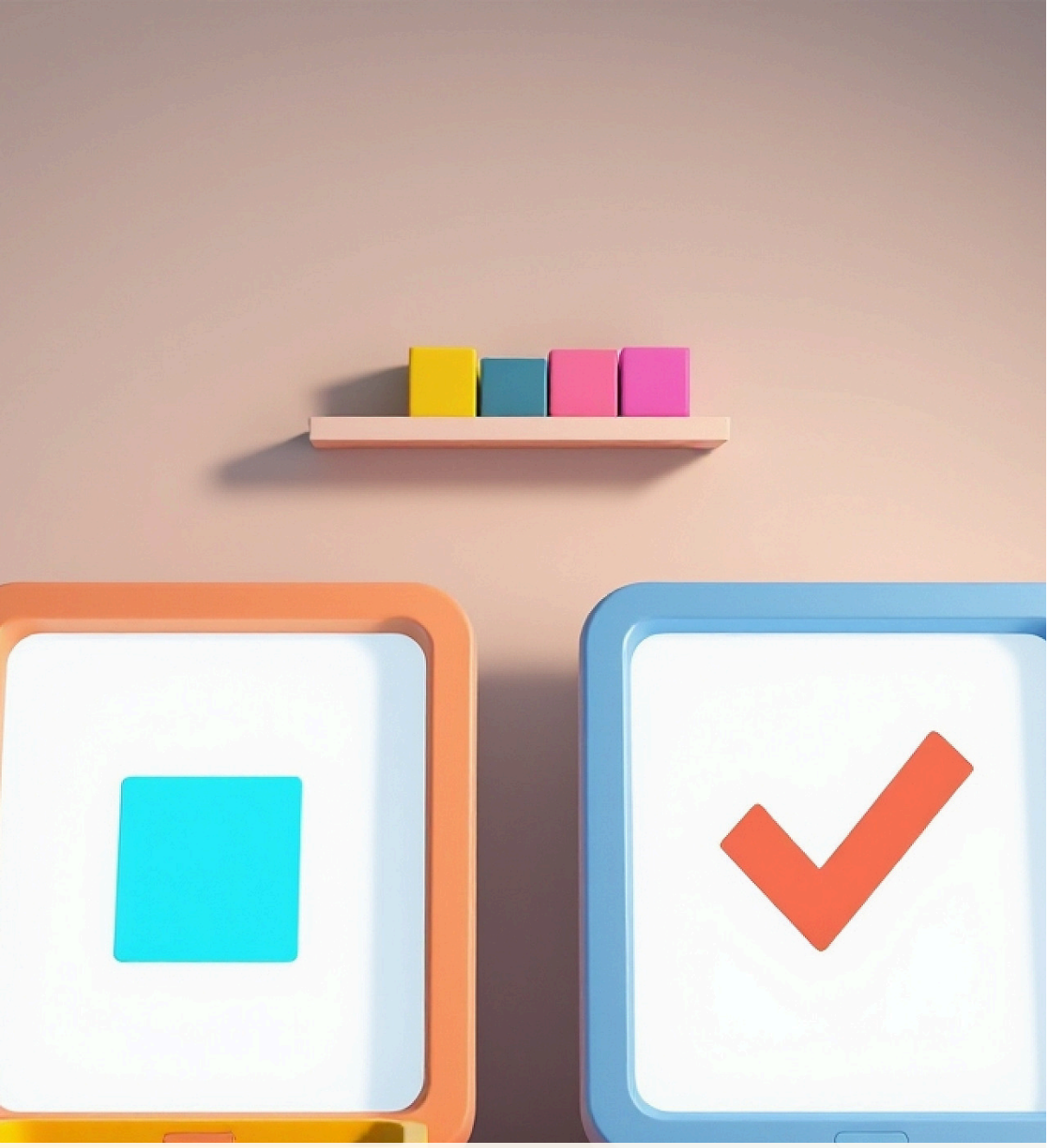
Bir uygulama fikrin var ve bunun nasıl çalışan bir programa dönüştüğünü bilmek istiyorsun. Kısa yanıt şu: problemi tanımlarsın, parçaları tasarlarsın, sonra küçük parçaları tek tek oluşturup test edersin. Bu kitap bu yolda sana adım adım rehberlik eder. Sonunda, büyük bir fikrin küçük, test edilebilir parçalarla nasıl büyüdüğünü anlayacaksın.



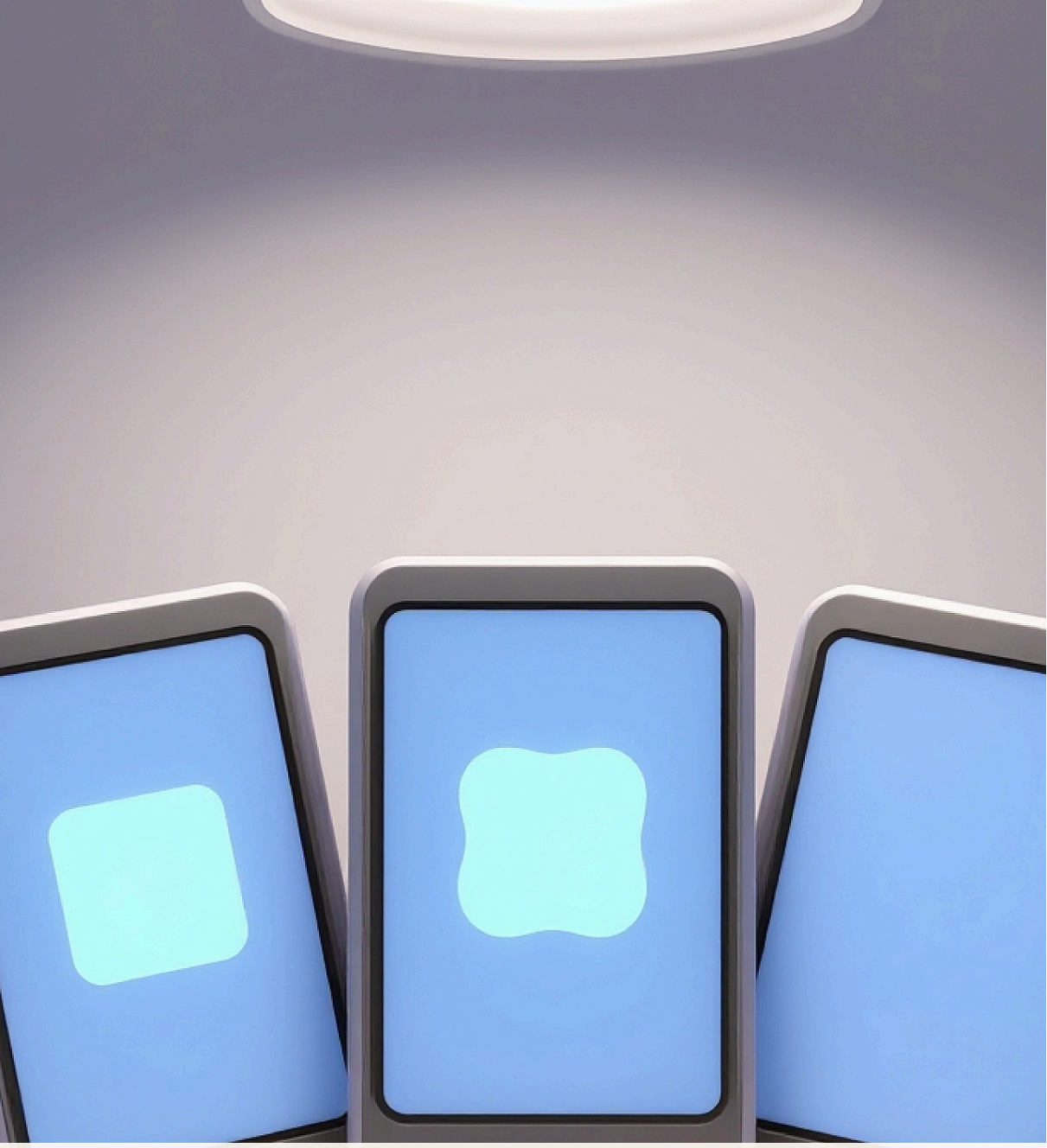
Uygulamanın çözeceği problemi bul. Net bir problem, uygulamanın neden var olduğunu ve kimin için olduğunu sana söyler. Örneğin, bir görev listesi uygulaması, yapman gerekenleri unutma problemini çözer. Problemi tek bir cümleyle yaz. Problem net olduğunda, sonraki her karar kolaylaşır.



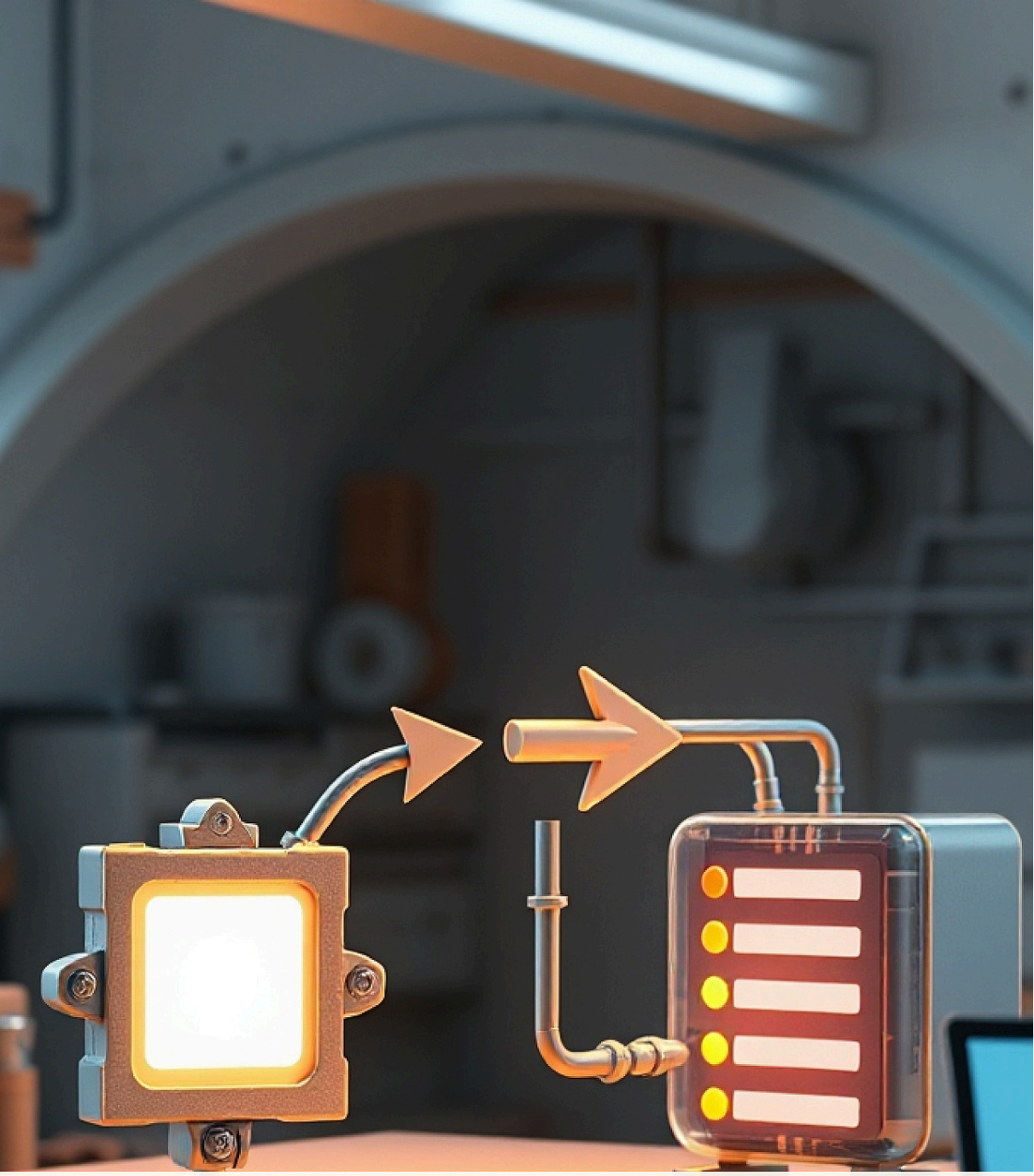
Kullanıcının uygulamayla ne yaptığını adım adım tanımla. Her eylemi yaz; örneğin bir görev yazıp eklemek için bir düğmeye dokunmak gibi. Bu kullanıcı eylemleri, programının desteklemesi gereken hedefler olur. Bir görev listesi uygulamasında kullanıcı görev ekler, listeyi görüntüler, görev siler ve görevleri kaydeder. Listeyi kısa ve somut tut.



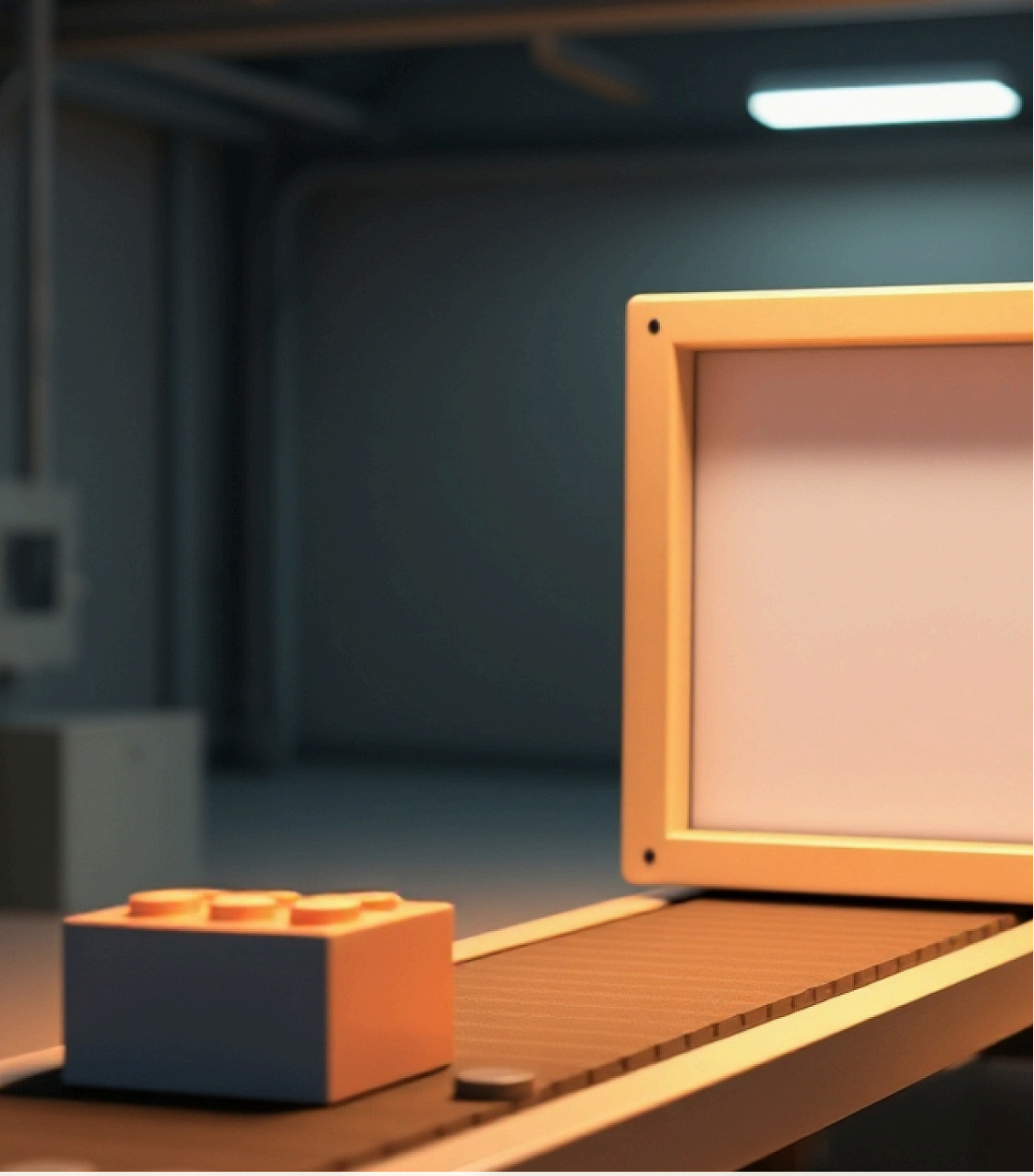
Uygulamanın saklaması gereken bilgi parçalarını listele. Bir görev listesi uygulamasında her görevin bir adı ve tamamlanıp tamamlanmadığı bilgisi gerekir. Programın sakladığı ve üzerinde çalıştığı şey bu veridir. Her parçayı basit bir etiket olarak yaz; örneğin görev adı ve tamamlandı. Verini erken bilmek, uygun ekranları ve işlem adımlarını tasarlamana yardımcı olur.



Kullanıcının göreceği ekranları kabaca çiz. Her ekran için kâğıda basit kutular çiz; neyin görüneceğini ve kullanıcının neye dokunabileceğini ya da ne yazabileceğini göster. Bir görev listesi uygulamasında bir ekran bir metin alanı ve bir ekle düğmesi gösterir, bir diğeri görev listesini gösterir. Çizimler hızlıdır ve değiştirmesi ucuzdur. Herhangi bir kod yazmadan önce eksik parçaları fark etmene yardımcı olurlar.



Her ekranın arkasındaki işlem adımlarını planla. Programın ne yapması gerektiğini tanımla; örneğin yazılan bir görevi alıp listeye eklemek gibi. Bu işlem adımları veriyi ekranlara bağlar. Örneğin, kullanıcı ekle düğmesine dokunduğunda program metni okur, bir görev oluşturur ve gösterir. Kodlamadan önce her adımı sade sözcüklerle yaz.



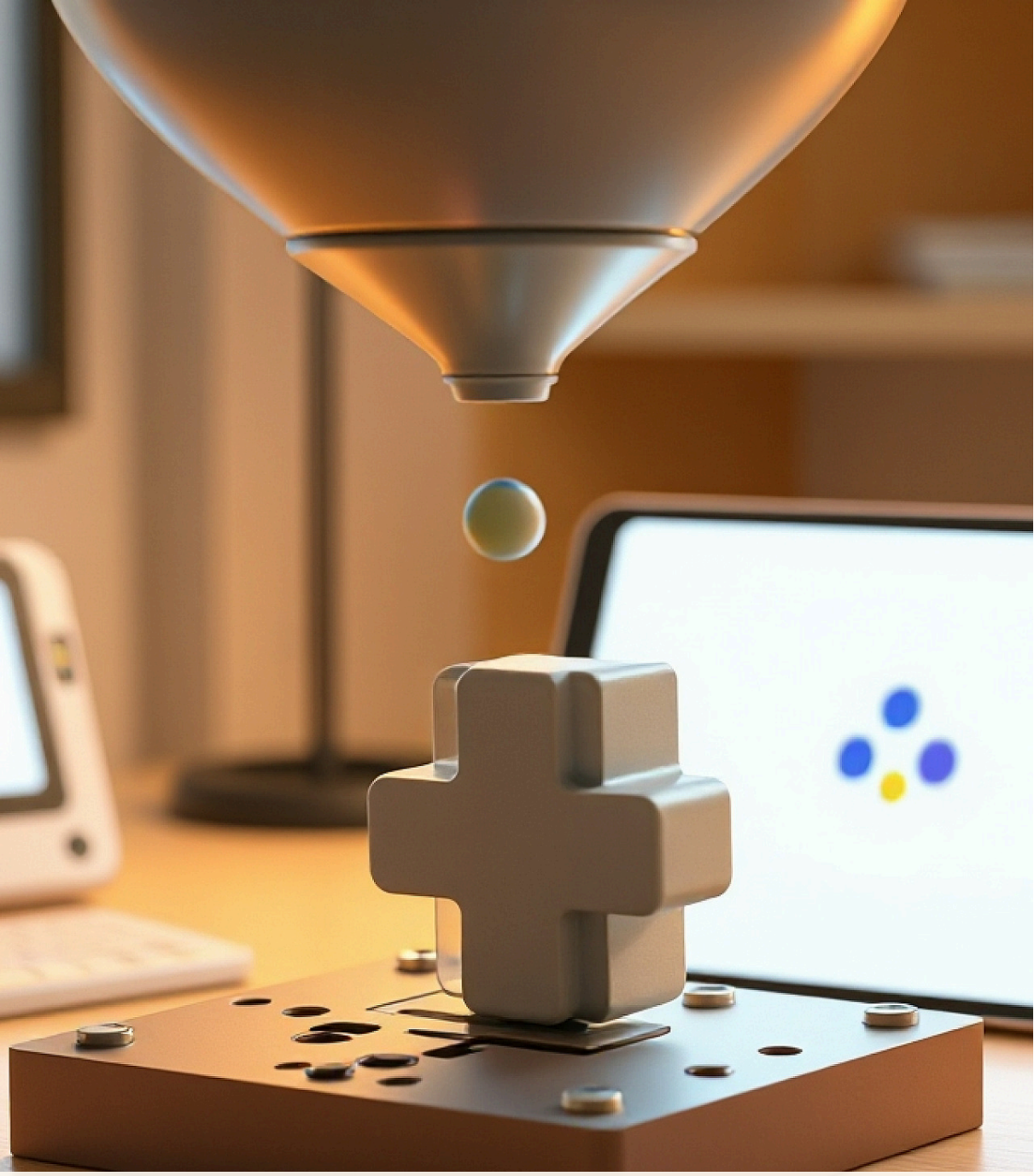
Yalnızca en temel eylemi kapsayan, gerçekten çalışan küçük bir sürüm oluştur. Henüz her özelliğe ihtiyacı yok; sadece çalışması yeter. Bir görev listesi uygulamasında, bir görev ekleyip onu ekranda göstermekle başla. Kodu basit ve odaklı tut. Küçük, çalışan bir sürüm, fikrinin gerçeğe dönüşebileceğini kanıtlar.



Küçük sürümü kendin test et ve planladığın gibi yapıp yapmadığını kontrol et. Test, neyin çalıştığını ve neyin çalışmadığını gösterir. Normal metinle, boş metinle ve çok uzun metinle görev eklemeyi dene. Ne olduğunu yaz. Erken test etmek sonradan zaman kazandırır, çünkü problemleri küçük bir programda bulmak daha kolaydır.



Testin bulduğu problemleri düzelt. Küçük sorunları erken düzeltmek, uygulama büyüdükçe onların büyümesini engeller. Örneğin, boş bir görev ekleniyorsa, bunu durduran bir kontrol ekle. Her seferinde tek bir şeyi değiştir ve yeniden test et. Her düzeltme, küçük sürümü daha güvenilir ve bir sonraki özelliğe daha hazır hale getirir.



Bir görev listesi uygulamasında önce görev ekleme yeteneğini hazırla. Bu, bütün fikrin küçük, test edilebilir bir parçasıdır. Metin alanını okuyup yeni bir görev oluşturan kodu yaz. Başka bir şey yapmadan önce eklemenin çalıştığını test et. Her seferinde bir parça oluşturmak, programı anlaşılır ve düzeltmesi kolay tutar.



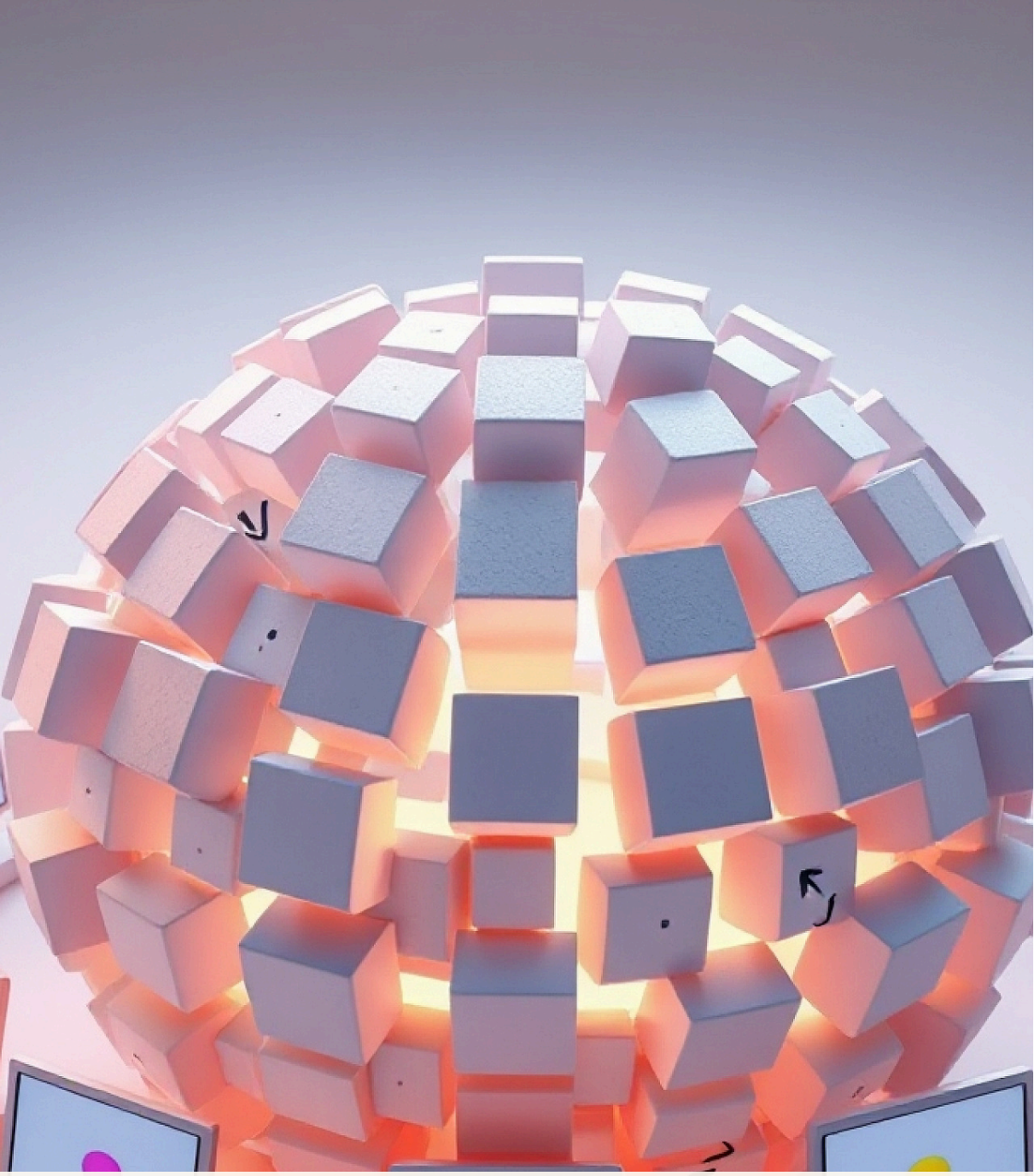
Sonra, eklenen görevleri görüntülemeyi hazırla. Artık kullanıcı, birinci özelliğin ikinciyle birlikte çalışmasının sonucunu görebilir. Her görevi ekranda bir liste halinde gösteren kodu yaz. Eklenen görevlerin doğru göründüğünü test et. Artık iki küçük parça birlikte çalışıyor ve uygulama işe yarar hissettirmeye başlıyor.



Daha sonra görev silme yeteneğini ekle. Her yeni özellik, zaten sahip olduğun çalışan parçaların üzerine kurulur ve test edilir. Seçilen bir görevi listeden kaldıran kodu yaz. Silmenin çalıştığını ve kalan görevlerin hâlâ göründüğünü test et. Her seferinde bir özellik eklemek, program büyürken onu kararlı tutar.



Sonra kaydetmeyi ekle; böylece görevler korunur ve uygulama kapandığında kaybolmaz. Yine, bu tek parçayı oluştur ve devam etmeden önce test et. Görevleri saklayan ve uygulama başladığında yükleyen kodu yaz. Uygulamayı kapatıp yeniden açarak test et. Kaydetme, uygulamayı günlük kullanım için güvenilir kılar.



Büyük bir fikir, küçük, test edilebilir parçalara bölünerek geliştirilir. Her parça, bir sonraki başlamadan önce oluşturulur, test edilir ve düzeltilir. Bir görev listesi uygulamasında ekleme, görüntüleme, silme ve kaydetme ayrı parçalardı. Bu yaklaşım her uygulama fikri için işe yarar. Küçük adımlar seni kontrol altında tutar ve ilerlemeyi görünür kılar.



Her şeyi bir anda oluşturmaktan ve test etmeyi atlamaktan kaçın. Herhangi bir parça çalışmadan önce tüm özellikleri eklemeye çalışmak, problemleri bulmayı zorlaştırır. Her küçük parçayı oluştururken test et. Bir şey bozulursa, son değişiklikten kaynaklandığını bilirsin. Sabır ve küçük adımlar zaman ve hayal kırıklığından tasarruf sağlar.



Yolu özetle: problemi tanımla, veriyi, ekranları ve adımları tasarla, sonra küçük parçaları sırayla oluştur, test et ve düzelt. Bir görev listesi uygulamasında bir görev ekledin, görüntüledin, sildin ve kaydettin; hepsini tek tek. Büyük bir fikrin küçük, test edilebilir parçalarla büyüdüğünü unutma. Küçük başla, sık test et ve geliştirmeye devam et.