Üretime hazırlık kontrol listesi

Yapay zekayla yazılan bir uygulamayı gerçek kullanıcıya açmadan önce 15 maddelik denetim

C.E.O. Bee

Uygulama: _____ Tarih: _____

KIRMIZI Kaldıysa canlıya almayın.

KOD OKUMADAN Yanıt ve kanıt istemekle taranabilir.

Düzeltme yükü: S tek ayar ya da küçük değişiklik, M birkaç dosya ya da ekran ve test, L temel bir şeyi yeniden kurmak. Süre ya da fiyat değildir.

	MADDE	NASIL KONTROL EDİLİR	GEÇTİ SAYILIR
☐	**1 Sırlar: API anahtarı, .env, istemci paketine sızma** KIRMIZI KOD OKUMADAN Düzeltme yükü: S	Depo geçmişinde ve derlenmiş pakette canlı anahtar arayın (örneğin sk_live_). eyJ ile başlayan dizilerin içinde "service_role" yazıyor mu bakın. Tarayıcıda Ağ sekmesindeki isteklere göz atın. **Kod okumadan:** Hiç anahtar paylaşıldı mı, değiştirildi mi? Kanıt: anahtarların son değiştirilme tarihi.	Geçmişte ve pakette canlı anahtar yok. Açığa çıkmış her anahtar iptal edilip yenilendi; dosyadan silmek yetmez, anahtar depo geçmişinde kalır.
☐	**2 Kimlik doğrulama ve oturum** KIRMIZI Düzeltme yükü: S–M	Hazır kimlik servisi ya da olgun kütüphane kullanılıyor mu? Kendi yazılmışsa parolalar yavaş bir özet algoritmasıyla (argon2, bcrypt) saklanıyor mu? Oturum belirteci çerezde (HttpOnly, Secure, SameSite) mi, localStorage'da mı? Sıfırlama bağlantısını iki kez kullanmayı ve art arda yanlış parola göndermeyi deneyin.	Parolalar özetlenmiş; belirteç HttpOnly ve Secure bir çerezde (localStorage'daysa gerekçesi yazılı ve madde 4 geçti); sıfırlama bağlantısı tek kullanımlık ve süreli; yanlış denemeler sınırlı.
☐	**3 Yetkilendirme: nesne düzeyi** KIRMIZI Düzeltme yükü: M; kaydın sahibi yoksa L	İki hesapla sınayın: A ile açtığınız kaydın kimliğini B'ye ait bir kaydın kimliğiyle değiştirip isteği yeniden gönderin. Yönetim uçlarını normal kullanıcı oturumuyla çağırın. Tarayıcıdan doğrudan veritabanına konuşan servislerde (Supabase gibi) satır düzeyi güvenliğin (RLS) açık olduğuna bakın.	B'nin verisi A'ya hiçbir yoldan dönmüyor (403 ya da 404); yönetim uçları rolü sunucuda, her istekte denetliyor; her tabloda RLS açık ve ilke tanımlı, tabloyu açan görünüm yok ya da security_invoker ile tanımlı.
☐	**4 Girdi doğrulama ve enjeksiyon** KIRMIZI Düzeltme yükü: S–M	Metin birleştirilen sorguları ve ham HTML basılan yerleri arayın. Formlara ' ve <img src=x onerror=alert(1)> yazıp sonucu izleyin. Yüklenen dosyanın türünün, boyutunun ve nereye yazıldığının sunucuda denetlendiğini doğrulayın.	Tüm veritabanı erişimi parametreli ya da güvenilir bir ORM üzerinden; kullanıcı metni HTML olarak değil düz metin olarak gösteriliyor; yüklemelerde tür ve boyut sınırı sunucuda uygulanıyor.
☐	**5 Veritabanı: şema, kısıt, indeks, geçiş** Düzeltme yükü: M; veri modeli yanlışsa L	Depodaki geçiş (migrasyon) dosyalarından boş bir veritabanı kurulabiliyor mu? Önemli tablolarda ilişki, benzersizlik ve NOT NULL kısıtlarına, sık filtrelenen sütunlarda indekse bakın. Üretim şemasını depodakiyle karşılaştırın; örnek veri ya da test kullanıcıları üretimde kalmış mı?	Boş veritabanı depodaki geçişlerle kurulup uygulama açılıyor; ilişkili kayıtlar kısıtlarla korunuyor; üretim şeması depoyla aynı.
☐	**6 Yedek ve geri yükleme testi** KIRMIZI KOD OKUMADAN Düzeltme yükü: S–M	Yedek nerede, ne sıklıkla alınıyor, kimin hesabında? Son yedeği yeni ve boş bir veritabanına geri yükleyin; satır sayısına ve en son kayda bakın; süreyi not edin. Yüklenen dosyaların da yedeği var mı? **Kod okumadan:** Son yedek ne zaman geri yüklendi? Kanıt: geri yükleme tarihi ve süresi.	Son geri yükleme denemesinin tarihi ve süresi yazılı; geri yüklenen veri eksiksiz ve okunabilir; yedek uygulamanın çalıştığı hesaptan ayrı bir yerde.
☐	**7 Ortamlar ve dağıtım** KOD OKUMADAN Düzeltme yükü: M	Geliştirme ve üretim için ayrı veritabanı ve anahtarlar var mı? Önizleme sürümleri üretim verisine bağlı mı? Önceki sürüme tek adımda dönülebiliyor mu (bir kez sınayın)? Alan adı, DNS kayıtları ve TLS sertifikası kimin hesabında? **Kod okumadan:** Bir test değişikliği canlı veriye dokunabilir mi? Kanıt: ayrı test ortamı ve geri dönüş denemesi.	Test amaçlı bir değişiklik üretim verisine dokunamıyor; son sürüme geri dönüş sınandı ve çalıştı; yapılandırma ortam değişkenlerinde.
☐	**8 Hata yönetimi, log ve izleme** Düzeltme yükü: S–M	Bilerek bozun: veritabanı bağlantısını kesin, geçersiz girdi gönderin. Kullanıcı ne görüyor, siz nasıl haberdar oluyorsunuz? Hata izleme ve çalışma süresi kontrolü var mı, bildirim bir insana ulaşıyor mu? Log'larda parola, erişim belirteci, anahtar, bağlantı dizesi ya da kart verisi olmamalı.	Hata kullanıcıya anlaşılır bir mesajla, size ayrıntısıyla ulaşıyor; kritik hata bir kişiye bildirim olarak düşüyor; log'larda gizli bilgi yok.

	MADDE	NASIL KONTROL EDİLİR	GEÇTİ SAYILIR
☐	**9 Bağımlılıklar: kullanılmayan, uydurma, açık, lisans** Düzeltme yükü: S–M	Üretim bağımlılıklarını bilinen açıklara karşı tarayın (npm projelerinde npm audit --omit=dev). Her paketin gerçekten var olduğunu, yayıncısının ve yaşının makul göründüğünü doğrulayın; kullanılmayanları ayıklayın. Kilit dosyası depoda mı? Lisanslar ticari kullanıma uygun mu?	Üretim bağımlılıklarında yüksek ya da kritik açık yok, ya da gerekçeli istisna yazılı; her paket doğrulandı; kilit dosyası depoda.
☐	**10 Testler: kritik akışlar ve anlamlı testler** Düzeltme yükü: M	Kritik akışları listeleyin (kayıt, giriş, ödeme, veri yazma): her birinin testi var mı? Bir kuralı bilerek bozun; hiçbir test kırılmıyorsa test işe yaramıyor demektir. Testler her değişiklikte otomatik çalışıyor mu?	Kritik akışlar için, kural bozulduğunda kırılan testler var ve her değişiklikte otomatik çalışıyor.
☐	**11 Performans ve ölçek** Düzeltme yükü: M	Yüzlerce ve binlerce kayıtlık test verisiyle liste ve arama sayfalarını açıp ölçün (tarayıcının Ağ sekmesi yeterli). Bir istekte kaç sorgu çalıştığına bakın. Liste uçlarında sayfalama, görsellerde boyutlandırma, dış servis çağrılarında zaman aşımı var mı?	Gerçekçi veri hacminde ana ekranlar, önceden yazdığınız hedef süre içinde açılıyor; listelerde sayfalama var; dış çağrılarda zaman aşımı tanımlı.
☐	**12 Ödeme ve hassas akışlar** KIRMIZI Düzeltme yükü: M; baştan yanlış kurulmuşsa L	Webhook ucuna imzasız sahte bir istek gönderin (reddedilmeli). Aynı olayı iki kez gönderin (tek sipariş oluşmalı). İstek gövdesindeki tutarı değiştirin (sunucu tutarı kendi hesaplamalı). Sipariş "ödendi" durumuna yönlendirme sayfasından değil, doğrulanmış bildirimden mi geçiyor?	Sahte, tekrarlanan ya da değiştirilmiş istek kayıt oluşturmuyor; tutar sunucuda hesaplanıyor; durum yalnızca doğrulanmış bildirimle değişiyor.
☐	**13 Kişisel veri ve KVKK** KIRMIZI KOD OKUMADAN Düzeltme yükü: M–L	Veri envanteri çıkarın: hangi kişisel veri, nerede duruyor, kim okuyabiliyor, hangi üçüncü taraflara gidiyor, ne kadar saklanıyor, nasıl siliniyor? İstemlere (prompt) kullanıcı verisi giriyor mu? Bir test kullanıcısını silip veritabanında, log'larda ve yedeklerde ne kaldığına bakın. **Kod okumadan:** Hangi kişisel veri, nerede, kimlerde? Kanıt: veri envanteri tablosu.	Her kişisel veri için yeri, erişimi, üçüncü tarafı ve silme yolu yazılı; silme talebi karşılanabiliyor; yapay zeka sağlayıcısına giden veri biliniyor. Hukuki danışmanlık değildir; hukuki değerlendirme için avukata danışın.
☐	**14 Maliyet ve kötüye kullanım** KOD OKUMADAN Düzeltme yükü: S–M	Ücretli servis anahtarlarının tarayıcıya gitmediğini doğrulayın (madde 1). Kendi uygulamanızın bir ucuna art arda istek gönderin; bir noktadan sonra 429 görmelisiniz (ucu ücretli bir servisi çağırıyorsa önce az sayıda istekle ya da test ortamında deneyin). Sağlayıcı konsolunda aylık harcama üst sınırı ve bütçe uyarısı var mı? Kayıt formunda bot koruması? **Kod okumadan:** Harcama üst sınırı var mı? Kanıt: sağlayıcı panelinden ekran görüntüsü.	Art arda istek kısıtlanıyor; harcama üst sınırı ve uyarısı tanımlı; ücretli anahtarlar yalnızca sunucuda.
☐	**15 Sahiplik ve sürdürülebilirlik** KOD OKUMADAN Düzeltme yükü: S–L	Şu listeyi doldurun: kod deposu, barındırma, alan adı ve DNS, veritabanı, e-posta ve SMS, ödeme hesabı, mağaza hesapları, yapay zeka sağlayıcı hesabı. Her birinin sahibi ve yöneticisi kim, kurumsal e-postaya bağlı mı? Kod platformdan dışa aktarılabiliyor mu; README'yi izleyince temiz bir bilgisayarda çalışıyor mu? **Kod okumadan:** Tüm hesapların yöneticisi kim? Kanıt: hesap listesi ve yönetici e-postaları.	Tüm hesaplar şirketin ya da ürün sahibinin adına; kod dışa aktarılmış ve bir depoda; bir mühendis, yapay zeka aracı olmadan, ilk gün projeyi çalıştırabiliyor.

Karar: kırmızı, sarı, yeşil

Kırmızı	**1, 2, 3, 4, 6, 12 ya da 13. maddelerden biri kaldı**	Canlıya almayın. Kullanıcı verisi ya da para söz konusu; önce bu madde kapanmalı. Gerekçeli istisna yoktur ("uygulanmaz" sayılanlar dışında).
Sarı	**Kırmızı maddeler geçti; diğerlerinden biri ya da birkaçı kaldı**	Sınırlı ve tanıdık bir kullanıcı grubuyla açın; her kalan madde için sahibi ve tarihi olan bir düzeltme planı yazın.
Yeşil	**Tüm maddeler geçti (kırmızı olmayanlarda gerekçeli istisna yazılı olabilir)**	Canlıya alın; denetimi büyük her değişiklikten sonra yineleyin.

Eşikler makalenin önerisidir, bir standart değildir. Uygulamanızda ilgili özellik yoksa (ödeme gibi) o madde "uygulanmaz" sayılır.

Projenizi anlatın: ceobee.dev

C.E.O. Bee · ceobee.dev

Kaynak makale: Vibe coding ile yapılan uygulama üretime hazır mı? 15 maddelik denetim · ceobee.dev/muhendislik/vibe-coding-uygulamasi-uretime-hazir-mi/ (komutlar ve gerekçeler makalede)