Claude Code ile İşletmenin Canlı 3D Dashboard'unu Kur

Oct 24, 2026 · @talha enes ünüvar

Ne kuracaksın

Yaklaşık 1,5 saatte Claude Code, işletmenin canlı verisiyle çalışan animasyonlu bir 3D dünya kuruyor. Müşteriler insan ya da ev, ürünler raf ya da palet, kargolar yolda giden kamyonet, ciro da dükkanın ya da ofisin oluyor. Bir objeye tıklıyorsun, parlıyor ve gerçek rakamlarınla bir stats kartı açılıyor. Rakamlar birkaç dakikada bir kendiliğinden yenileniyor, kopyala-yapıştır yok.

Kod bilmen gerekmiyor. Kodu Claude Code yazıp çalıştırıyor; sen sorularını cevaplayıp işini kontrol ediyorsun. Arka planda birbirini besleyen üç küçük parça kuruluyor:

1. **Sync script:** iyzico, Trendyol, Shopify ya da Google Sheets'ten 5 dakikada bir en güncel rakamları çekiyor.
2. **Veri dosyası (data.json):** bu rakamları tutuyor.
3. **3D dashboard:** veri dosyasını dakikada bir okuyup kartları güncelliyor.

Beş adımda kuruyorsun: veriyi bağla, dünyayı tasarla, Three.js ile kur, canlı hale getir, sonra istersen paylaş.

Başlamadan önce

İhtiyacın olan dört şey: Claude Code, ücretli bir Claude planı, boş bir klasör ve verine erişim. Önce bunları işaretle:

- [ ] **Claude Code.** Claude masaüstü uygulamasındaki Code sekmesi ya da terminalde Claude Code. Ücretli plan gerekiyor (Pro, Max, Team veya Enterprise).
- [ ] **En güçlü model.** En güncel Opus'u seç. 3D kısım uzun bir kod; güçlü model ilk seferde doğru kurma ihtimalini ciddi artırıyor.
- [ ] **Boş bir klasör.** Örneğin Belgeler içinde `3d-dashboard`. Claude Code'u bu klasörde aç ve tüm adımları aynı oturumda yap.
- [ ] **Verine erişim** (aşağıdaki tablo). Mümkünse sadece okuma yetkili anahtar kullan; değilse sync script'e "sadece okuma" kuralı koyduracağız.
- [ ] **Opsiyonel: işletmenin bir fotoğrafı.** Klasöre kaydet; dünya senin mekanına benzesin.

Bilgisayarında eksik bir şey varsa (sync script'i çalıştıran Node.js gibi) Claude Code söyler ve kurmadan önce izin ister.

Canlı rakamlar nereden gelecek? Tek uygulamada her şey nadiren olur. Türkiye'de Stripe'ı doğrudan kullanamıyorsun (kaynak), o yüzden tabloyu yerel kaynaklarla kurdum.

Kaynak	Ne verir	Erişimi nasıl alırsın
Trendyol	Siparişler, ürünler, stok, iadeler	Satıcı Paneli → Hesap Bilgilerim → Entegrasyon Bilgileri: Satıcı ID, API Key, API Secret. Sadece admin kullanıcıda görünür (Trendyol dokümanı)
iyzico	Ödemeler, ciro, iadeler	iyzico merchant panelindeki API anahtarları (API Key + Secret Key)
Shopify	Siparişler, ürünler, stok	Admin → Settings → Apps and sales channels → Develop apps → sadece read scope'larla app oluştur (orders, products, inventory) → kur ve access token'ı kopyala
Google Sheets	Geri kalan her şey: stok, kargo, rezervasyon, hedefler	Sheet'te File → Share → Publish to web, CSV olarak yayınla ve linki kopyala
Başka uygulama (ikas, Shopier, Hepsiburada, muhasebe programın)	Kargo, rezervasyon, personel, fatura	API'si varsa Claude Code kullanabilir. Yoksa Excel/CSV olarak Google Sheet'e aktar

Dikkat: Trendyol ve iyzico anahtarları tam yetkili; sadece okuma yetkili anahtar seçeneği yok gibi düşün. Bu yüzden 1b'deki prompt, script'e yalnızca veri okuyan (GET) istekler attırıyor.

Anahtarlara şifre gibi davran. Chat mesajına, ekran görüntüsüne ya da dashboard'un kendisine asla yapıştırma. 1. adım nereye koyacağını gösteriyor.

Adım 1: Canlı veriyi bağla

Bu adımın sonunda sync script gerçek rakamlarını çekip bir veri dosyasına kaydediyor ve sen bunları kendi panellerinle karşılaştırmış oluyorsun.

1a. Anahtarların için güvenli bir yer aç. Bunu Claude Code'a yapıştır:

```
Bu klasörü canlı bir 3D dashboard projesi olarak kur. .env adında gizli bir ayar dosyası oluştur ve anahtarlarım için boş alanlar aç: TRENDYOL_SELLER_ID, TRENDYOL_API_KEY, TRENDYOL_API_SECRET, IYZICO_API_KEY, IYZICO_SECRET_KEY, SHOPIFY_STORE, SHOPIFY_TOKEN, SHEET_CSV_URL (kullanmadıklarımı çıkarırım). .env asla yüklenmesin ya da paylaşılmasın diye bir .gitignore ekle. Anahtarları bu chat'e yapıştırmamı isteme. .env dosyasını nasıl açıp kendim dolduracağımı söyle.
```

.env dosyasını aç, her anahtarı kendi `=` işaretinin sonrasına yapıştır ve kaydet. Kullanmadıklarını boş bırak.

1b. Rakamlarını çek. Bunu yapıştır:

```
.env'deki kaynaklardan işletme verilerimi okuyan ve en güncel rakamları data.json'a kaydeden bir sync script yaz. Script SADECE okuma yapsın: yalnızca GET/listeleme istekleri atsın, hiçbir sipariş, ürün, fiyat, stok ya da ödeme kaydını değiştirmesin.

Bir işletme sahibinin önemseyeceği rakamları şu gruplarla kaydet: müşteriler, ürünler ve stok, siparişler ve ciro (TL), kargolar ve bulduğun diğer her şey. Her rakam için değeri, zaman aralığını (bugün, bu hafta, bu ay) ve nereden geldiğini kaydet. Sync zamanını da kaydet.

Sadece toplamlar ve sayılar. Müşteri adı, e-posta, adres, telefon ya da kart bilgisi asla kaydetme; anahtarlarımı asla ekrana yazdırma.

Bir kez çalıştır. Bana her rakamı aralığı ve kaynağıyla bir tabloda göster, sonra işletmenin tam resmi için neyin eksik olduğunu ve onu nereden alabileceğimi söyle.
```

Devam etmeden önce rakamları kontrol et. İki üç tanesini kendi panelinle karşılaştır: bu ayın cirosu, bugünün siparişleri gibi. Her birinin zaman aralığına bak. Yanlış ya da eksik bir şey varsa şimdi söyle ("Ciro sadece bu ay olmalı", "Stoku Google Sheet'imden ekle") ve sync'i tekrar çalıştırt. Sonra kuracağın her şey bu rakamları gösterecek.

KVKK notu: Müşteri kişisel verisini data.json'a hiç almamak, sonradan dashboard'u paylaştığında da seni korur.

Adım 2: İşletmeni 3D dünya olarak tasarla

Bu adımda Claude Code dünyayı kodsuz, sadece kelimelerle planlıyor: içinde ne var, hangi obje hangi canlı rakamı gösteriyor. Planı burada doğru kurmak, sonradan baştan yapmaktan kurtarıyor.

İşletmeni Claude Code'a göster. Klasöre bir fotoğraf koy (dükkan önü, depo, tezgah, stüdyo) ya da birkaç cümleyle anlat. Örnek: "Trendyol'da satış yapan bir giyim markası. Üç yükleme kapılı mavi bir depo, önünde stok paletleri, iki kargo kamyoneti ve bir ofis."

Prompt 2. Köşeli parantezleri doldur ve yapıştır:

```
[İşletme türü, ör. Trendyol'da satış yapan bir giyim markası / bir kahveci]
işletiyorum. [Bu klasördeki fotoğrafa bak: foto.jpg. / Görünüşü şöyle: ...]

İşletmemi interaktif bir dashboard için küçük bir 3D izometrik dünya olarak
tasarla, masanın üstündeki bir maket gibi. Fotoğrafımı ya da tarifimi temel
al.

data.json'daki verinin her parçasını tıklanabilir bir objeye çevir:
müşteriler, ürünler ve stok, kargolar ve kamyonetler, ciro, personel ve uygun
olan diğer her şey. Her obje için bana şunları ver:
- dünyada nasıl göründüğü ve nerede durduğu
- kartında göstereceği 3-4 stat, her biri data.json'daki bir rakama bağlı
- nasıl hareket ettiği (yürüyen insanlar, tur atan kamyonetler, çalışan bir
tezgah ya da konveyör)

Markama uyan bir renk paleti de öner. Planı plan.md olarak kaydet ve henüz
kod yazma.
```

Planı kontrol et. Her stat data.json'daki gerçek bir rakamı göstermeli, uydurma bir rakamı değil. Eksik olanı ekle ("kahve makinesini ekle, bugün yapılan içecek sayısını göstersin"), yerini hak etmeyeni çıkar. 6-10 tıklanabilir obje yeterli; fazlası dünyayı kalabalıklaştırıp yavaşlatıyor.

Adım 3: Three.js ile kur

Çoğu kişinin hata yaptığı adım bu. Claude'dan sadece "bir dashboard" istersen grafikli düz bir web sayfası alırsın. **Three.js**'i adıyla söylersen (web için ücretsiz, açık kaynak 3D kütüphanesi) Claude Code gerçek bir 3D sahne kuruyor: derinliği ve gölgesi olan objeler, hareket eden şeyler ve her objede tıklama algılama.

Prompt 3. Bunu yapıştır:

```
plan.md'deki dünyayı Three.js ile interaktif bir web sayfası olarak kur.

Dünya:
- Plandaki her objeyi yumuşak yuvarlatılmış kenarlı, low-poly bir 3D model
olarak yap; yumuşak ışık ve gölge kullan, plandaki renk paletine uy.
- İzometrik tarzda bir kamera kullan. Sürükleyerek döndürebileyim, scroll ya
da pinch ile zoom yapabileyim; açılışta tüm dünya görünsün.
- Her şey animasyonlu olsun: yürüyen insanlar, rotasında giden kamyonetler,
çalışan makineler. Hiçbir şey hareketsiz durmasın.

Tıklama:
- Plandaki her obje tıklanabilir olsun. Üstüne gelince cursor değişsin.
Tıklayınca obje parlasın ve yanında frosted-glass bir stats kartı açılsın.
- Boş alana ya da başka bir objeye tıklamak kartı kapatsın.

Canlı veri:
- Her karttaki her rakam data.json'dan gelsin. Hiçbir rakamı koda gömme.
- data.json'ı 60 saniyede bir yeniden oku ve açık kartları sayfa yenilemeden
güncelle. Değişen rakamı animasyonla göster.
- Sync zamanından "X dk önce güncellendi" yazan küçük bir CANLI noktası
göster. Veri 30 dakikadan eskiyse nokta griye dönsün.
- Para birimini TL olarak, Türkçe sayı formatıyla göster (1.250,50 ₺).

Laptop'ta ve telefonda çalışsın. Bunun için bir local server başlat,
tarayıcımda aç ve bana linki ver.
```

Ne göreceksin. Claude Code dosyaları yazıyor, küçük bir local server başlatıyor ve dashboard'u tarayıcında `http://localhost:3000` gibi bir linkte açıyor. Server önemli: tarayıcılar, yanında açıldığı veri dosyasını okumasını engelliyor. Sayfa boş geliyor ya da hata veriyorsa aşağıdaki Sorun giderme bölümüne bak.

Dene. Dünyayı sürükleyip çevir, her objeye tıkla ve kartının 1. adımdaki rakamlarla aynı olduğunu kontrol et. Değişiklikleri düz Türkçe iste ("depoyu büyüt", "kamyonetleri yavaşlat"). Claude Code aynı dosyaları düzenliyor, baştan başlamıyorsun.

Adım 4: Canlı hale getir

Şu an rakamlar sadece Claude Code sync'i çalıştırınca güncelleniyor. Bu adım sync'i bir zamanlayıcıya bağlıyor, dashboard kimse dokunmadan kendini güncel tutuyor.

Prompt 4. Bunu yapıştır:

```
Dashboard'u kendi kendine çalışır hale getir:
- Sync script'i 5 dakikada bir çalıştır ve local server'ı açık tut.
- İkisi de bilgisayarım açılınca otomatik başlasın (Windows'ta Görev Zamanlayıcı / Task Scheduler, Mac'te bir background job).
- Her sync'i sync.log'a yaz. Bir sync başarısız olursa son iyi rakamları koru, son başarılı sync zamanını dashboard'da göster ve bir sonrakinde tekrar dene.
- Düz Türkçe kısa bir README.md yaz: dashboard nasıl açılır, nasıl durdurulup yeniden başlatılır, çalıştığı nasıl kontrol edilir.

Sonra her şeyi bundan sonra çalışacağı şekilde yeniden başlat ve bana çalıştığını göster.
```

Gerçekten canlı mı, kontrol et:

- [ ] CANLI etiketindeki "X dk önce güncellendi" 5 dakikada bir sıfırlanıyor.
- [ ] Kaynakta bir rakamı değiştir (Google Sheet'te bir hücre düzenle ya da yeni sipariş bekle) ve kartın 5 dakika içinde sayfa yenilemeden değiştiğini gör.
- [ ] Bilgisayarını yeniden başlat, dashboard linkini tekrar aç; hâlâ güncelleniyor.

5 dakikalık aralık API sınırlarının çok altında kalıyor; örneğin Trendyol aynı endpoint'e 10 saniyede 50 isteğe izin veriyor (kaynak). Daha hızlı istersen dakikada bir iste. Bilgisayarın açık ve uyanık olduğu sürece çalışır. Her zaman açık ve ekibinin her yerden açabileceği bir dashboard için 5. adıma geç.

Adım 5 (opsiyonel): Paylaş ve günlük özet ekle

Ekibin için internete koy. Ekibin tek bir linki herhangi bir laptop ya da telefondan açıyor, bilgisayarın kapalıyken bile rakamlar canlı kalıyor. Bunu yapıştır:

```
Bu dashboard'u ekibimin her yerden açabileceği şekilde internete koy. Önerdiğin ücretsiz ya da düşük maliyetli bir host kullan (Vercel veya Netlify gibi) ve bana ne kadar tutacağını söyle.
- Anahtarlarımı host'un gizli environment ayarlarında tut, asla sayfada ya da kodda değil.
- Bilgisayarımdaki zamanlayıcı yerine, sayfa istediğinde taze rakamları çek ve 2 dakika cache'le; böylece zamanlayıcı olmadan hep canlı kalsın.
- Şifreyle koru.
Hesabı kendim açayım diye beni adım adım yönlendir.
```

Hosting hesabını sen açıyorsun; Claude Code seni yönlendiriyor ve gerisini hallediyor. Linki ve şifreyi sadece ekibinle paylaş: sayfa gerçek cironu gösteriyor.

Not: Trendyol her istekte User-Agent başlığı istiyor, test ortamı ayrıca IP yetkisiyle çalışıyor (kaynak). Host'a taşıdıktan sonra Trendyol verisi gelmezse bunu Claude Code'a söyle.

Claude'un yazdığı günlük özet ekle. Sync rakamları taşıyor. Zamanlanmış bir Claude Code görevi de bu rakamları okuyup ne anlama geldiklerini söyleyebilir. Bunu yapıştır:

```
Her sabah 07:00'de çalışan zamanlanmış bir görev kur: bugünkü data.json'ı son
7 günle karşılaştır, neyin değiştiğini ve olağan dışı bir şey varsa 3
satırlık kısa bir özet halinde summary.json'a yaz. Bunu dashboard'un üstünde
ince bir banner olarak göster. Karşılaştırma yapabilmek için her günün
rakamlarının bir kopyasını sakla.
```

Her çalışma Claude kullanımından biraz yiyor, günde bir yeterli. Canlı rakamlar her durumda sync script'ten gelmeye devam ediyor.

Sorun giderme ve geliştirmeler

Çoğu sorun, gördüğünü aynı oturumda Claude Code'a düz Türkçe anlatınca çözülüyor. Log'u ve dosyaları kendisi okuyabiliyor.

Ne görüyorsun	Claude Code'a ne diyorsun
Boş sayfa ya da hata	"Dashboard boş. Aç, hatayı bul, düzelt ve geri kalan her şeyi aynı bırak."
Rakamlar güncellenmiyor (CANLI noktası gri)	"Dashboard güncellenmeyi bıraktı. sync.log'a bak, nedenini bul, düzelt ve aynı şekilde tekrar olmayacağından emin ol."
Anahtar ya da yetkiyle ilgili sync hatası (ör. Trendyol 401)	"Sync [Trendyol] anahtarımın reddedildiğini söylüyor. Hangi bilgiyi yanlış girmiş olabileceğimi adım adım söyle."
Bir rakam yanlış	"[Ciro] kartı [bu ay için X ₺] göstermeli. Sync'in bunu nasıl hesapladığını kontrol et, düzelt ve yeni rakamı panelimdekiyle yan yana göster."
Objeye tıklayınca bir şey olmuyor	"[Kamyonet]'e tıklayınca kartı açılmıyor. Plandaki her objeyi, küçük parçaları dahil tıklanabilir yap."
Kart objeyi kapatıyor ya da ekrandan taşıyor	"Stats kartını ekranın içinde, objenin yanında tut; asla üstünde değil."

Ne görüyorsun	Claude Code'a ne diyorsun
Düz ya da sade görünüyor	"Premium bir 3D render gibi görünsün: daha yumuşak gölgeler, yuvarlak kenarlar, her objede daha fazla detay ve daha zengin bir renk paleti."
İşletmeme benzemiyor	Klasöre bir fotoğraf daha ekle: "Dünyayı foto-2.jpg'ye daha çok benzet. Tıklanabilir objeleri ve canlı rakamları koru."
Telefonda yavaş ya da takılıyor	"Telefonda akıcı çalışsın: küçük objelerde daha az gölge, uzaktaki modelleri sadeleştir."

Canlıya çıktıktan sonra geliştirmeler:

- **Uyarılar:** "Bir ürünün stoğu 20 adedin altına düşünce rafı kırmızı yanıp sönsün ve kartında uyarı çıksın."
- **Trendler:** "Her sync'in geçmişini tut ve her karta son 7 günün sparkline'ını ekle."
- **Toplam barı:** "Üste bugünün cirosu, siparişleri ve yeni müşterileriyle ince bir bar ekle."
- **Satışı kutla:** "Yeni sipariş gelince depodan bir kamyonet çıksın ve üstünde küçük bir +₺ tutarı belirsin."
- **TV modu:** "Ofisteki ekran için tam ekran bir mod ekle: kamera objeden objeye uçsun, her kartı 5 saniye açsın."

İşletme türüne göre fikirler

Bunları Prompt 2 için başlangıç noktası olarak kullan; kendi objelerini ve rakamlarını koy.

İşletme	Dünya	Tıklayınca gördüğün
Pazaryeri satıcısı (Trendyol, Hepsiburada)	Depo, stok paletleri, kargo kamyonetleri, müşteri evleri	Ciro ve sipariş; ürün başına stok; kargodaki paketler; iade oranı
Kendi e-ticaret siten (ikas, Shopify)	Depo, raflar, kargo, vitrin	Ciro; sepet ortalaması; en çok satanlar; yeni ve tekrar gelen müşteri
Kahveci	Tezgah, kahve makinesi, masalar, sıra	Bugünkü satış; yapılan içecek; en çok satanlar; en yoğun saat

İşletme	Dünya	Tıklayınca gördüğün
Restoran	Salon, mutfak, bar, kurye motorları	Bu akşamki kuver; masa dönüş süresi; en çok satan yemekler; paket servis siparişleri (Yemeksepeti, Getir)
Spor salonu / stüdyo	Stüdyo odaları, resepsiyon, giren üyeler	Bugünkü giriş; ders rezervasyonu; yeni üye; iptaller
Kuaför / klinik	Koltuklar ya da odalar, resepsiyon, bekleme alanı	Bugünkü randevular; koltuk başına ciro; tekrar randevu oranı; gelmeyenler
Ajans / hizmet işi	Ofis katları, müşteri başına bir masa, proje panosu	Zamanında giden projeler; bütçeye karşı harcanan saat; vadesi gelen faturalar; pipeline
İçerik üreticisi	Stüdyo, kamera, gelen mesaj kutusu, marka masası	Bu haftaki izlenme; takipçi artışı; aktif marka işbirlikleri; bülten aboneleri
Emlak ofisi	Satılık ve satıldı tabelalı ev sırası	Aktif ilanlar; gösterim randevuları; teklifler; ilanda kalma süresi

Kaynaklar

- Orijinal rehber: "Build a Live 3D Dashboard of Your Business with Claude Code", @nocode.joshua
- Stripe'ın Türkiye'de doğrudan kullanılamaması
- Trendyol Partner API: Authorization ve istek sınırları