BTÖ-303 Ara Uygulama 3 Yönergesi: Etkileşimli Oyun ve Canlı Skor Tablosu

Oct 25, 2026 · @Mertcan Ünal

Amaç ve teslim

Bu üçüncü ara uygulamada 4 saat içinde, yapay zekâ ajanıyla **kullanıcı etkileşimi güçlü, çok oyunculu bir oyun** ve **canlı güncellenen bir skor tablosu** geliştirip Vercel'de yayınlayacaksın.

Yönergeyi biz veriyoruz: bu dokümanı ve A2'deki başlangıç istemini ajanına verebilirsin. Oyunun türünü kartlar ve ajanın önerdiği fikirler arasından sen seçersin; ama ne istendiği, hangi sayfaların ve uç noktaların bulunacağı ve nasıl puanlanacağın baştan bellidir.

Ödev olarak yalnızca Vercel alan adını (`*.vercel.app` adresini) teslim edeceksin. GitHub bağlantısı, kod dosyası ya da başka bir ek istenmez; değerlendirme tamamen canlı adres üzerinden yapılır. Bu yüzden adres herkese açık olmalı ve not girilene kadar çalışır durumda kalmalıdır.

Teslim şartlarının özeti:

- Uygulama Vercel'de yayında ve `*.vercel.app` adresi giriş istemeden açılıyor; adres oyunun adını yansıtıyor (örn. `uzay-kapismasi.vercel.app`) ve öğrenci numarası ya da ad-soyad içermiyor.
- İki oyuncu aynı odada, sayfayı yenilemeden birbirinin hamlesini görüyor.
- Skor tablosu kalıcıdır ve yeni skor geldiğinde sayfa yenilenmeden güncellenir.
- Uygulama dört sabit adresi sunuyor: `/`, `/leaderboard`, `/rapor` ve `/api/leaderboard`.
- `/rapor` sayfasındaki ajan öz-kontrolü, 80 puanlık bölümde en az 64 puan gösteriyor.
- Süre 4 saattir (240 dakika); teslim edeceğin tek şey Vercel alan adındır, yükleme yeri ders sırasında sözlü olarak duyurulur.

Eş zamanlılık Vercel'de nasıl çalışır?

Vercel artık WebSocket'i doğrudan sunabiliyor ama bu özellik beta; 4 saatlik derste varsayılan yol, yönetilen bir gerçek zamanlı servis olmalıdır.

Vercel Functions, WebSocket bağlantılarını Haziran 2026'dan beri herkese açık beta olarak sunuyor (duyuru). Ancak tek bir bağlantı tek bir fonksiyon örneğine sabitlenir; aynı odadaki oyuncular farklı örneklere düşebileceği için durumu paylaşmak üzere Redis gibi ek

bir servis gerekir ve bağlantılar fonksiyonun azami süresi dolunca kapanır. Bu yüzden aşağıdaki tabloda ilk seçenek önerilen yoldur.

Seçenek	Ne zaman	Ücretsiz plan sınırı
Supabase Realtime (Broadcast) ve Postgres	Varsayılan: tek hesapta hem canlı mesajlaşma hem skor tablosu	200 eş zamanlı bağlantı ve saniyede 100 mesaj; 1 hafta hareketsiz kalan ücretsiz projeler duraklatılır
Ably	Saniyelik mesaj payı gerekirse	200 bağlantı, saniyede 500 mesaj, ayda 6 milyon mesaj
Vercel Functions WebSocket (beta)	Cesur seçenek: Redis ile durum paylaşımı gerektirir	Fonksiyon limitleri ve fiyatlandırması geçerli; bağlantı azami sürede kapanır

Her öğrenci kendi hesabını açar; sınırlar senin projene aittir, sınıfça paylaşılmaz.

Mesaj bütçesine dikkat et: Supabase'in belgesi, istemciden gönderilen ya da istemciye teslim edilen her WebSocket mesajını bir olay sayıyor. Beş oyunculu bir odada her oyuncu saniyede bir güncelleme gönderirse, gönderimler ve diğer oyunculara teslimler birlikte yaklaşık 25–30 olay eder; saniyede 100 olay sınırı bu yüzden sürekli konum yayınlayan bir oyunu keser. Hamle tabanlı ya da oyuncu başına saniyede en fazla bir güncelleme gönderen bir oyun tasarla; "gerçek zamanlı" burada saniyede 60 kare değil, hamlenin sayfa yenilenmeden birkaç saniye içinde rakipte görünmesi demektir.

Sözleşme: sabit adresler ve teknik taban

Her uygulamada aynı dört adres bulunmalıdır; öğretim elemanı otuz uygulamayı aynı yoldan ve bir betikle kontrol edecektir.

Adres	Ne sunar
`/`	Lobi: takma ad girişi, "Oda aç" ve "Koda katıl"
`/leaderboard`	Skor tablosu sayfası; yeni skor gelince sayfa yenilenmeden güncellenir
`/rapor`	Rapor sayfası: oyunun adı, kartların, nasıl oynanır anlatımı ve ajanın öz-kontrol sonucu
`/api/leaderboard`	GET isteğiyle en yüksek 10 skoru JSON olarak döndürür (biçim aşağıda)

`/api/leaderboard` yanıtı şu biçimde olmalıdır; skorlar azalan sırada gelir, eşit skorda önce yapan üstte durur:

```
{
  "skorlar": [
    { "sira": 1, "ad": "Kaplan", "skor": 480, "zaman": "2026-10-27T10:42:11Z"
}
  ]
}
```

Konu	Zorunluluk
Çatı	Next.js (App Router, TypeScript); `npx create-next-app@latest` ile başla
Gerçek zamanlı	Yönetilen servis (önceki bölüm); her odaya ayrı kanal
Veritabanı	Skorlar için kalıcı Postgres (Supabase ya da Neon)
Skor yazımı	Yalnızca sunucuda (Route Handler ya da Server Action) yapılır; gelen veri Zod ile doğrulanır
Oyuncu kimliği	Giriş/kayıt yok; oyuncu takma ad seçer; gerçek ad ve öğrenci numarası hiçbir yerde görünmez
Gizli anahtarlar	Yalnızca sunucu ortam değişkenlerinde durur; `NEXT_PUBLIC_` önekiyle tanımlanmaz
Yayın	Vercel'de production dağıtımı; adres herkese açık

Odak 1: Kullanıcı etkileşimi

Puanın %35'i etkileşimden gelir; oyun, birinin kural okumadan, telefonla bile, ilk 30 saniyede oynayabileceği kadar akıcı olmalıdır.

Alan	Beklenen davranış
Katılım (E1)	Ana sayfadan oyuna girmek en fazla 3 adımdır (takma ad, oda aç ya da koda katıl, oyun); oda kodu tek tıkla kopyalanır; kural okutmayan kısa bir "nasıl oynanır" ipucu vardır

Alan	Beklenen davranış
Eş zamanlı etkileşim (E2)	Bir oyuncunun hamlesi diğerinde sayfa yenilenmeden birkaç saniye içinde görünür; kimin ne yaptığı ve sıranın kimde olduğu belli olur
Oyuncu varlığı (E3)	Odadaki oyuncular listelenir; katılan ve ayrılan oyuncu anında görünür; ayrı odalar birbirinin olaylarını görmez
Anlık geri bildirim (E4)	Hamle kendi ekranında hemen görünür (iyimser güncelleme); her eylemin görsel, kartlara uygunsa sesli bir karşılığı vardır; kalan süre ve tur sonucu gösterilir; bekleme anlarında durum mesajı çıkar
Dayanıklılık (E5)	Bağlantı koparsa uyarı çıkar ve otomatik yeniden bağlanma denenir; sayfa yenilenince oyuncu odasına döner; geçersiz ya da dolu oda kodunda anlaşılır bir hata görünür
Mobil ve erişilebilirlik (E6)	375 px genişlikte yatay kaydırma yoktur; dokunma hedefleri en az 44 px'tir; ana eylemler klavyeyle yapılabilir; metin ve düğme kontrastı okunaklıdır

Oyuncu listesi için Realtime'ın Presence özelliği kullanılabilir; ama ücretsiz planda saniyede 20 Presence mesajıyla sınırlıdır (Supabase), bu yüzden oyuncu listesini seyrek güncelle.

Odak 2: Skor tablosu

Puanın %35'i skor tablosundan gelir; tablo yalnızca bir liste değil oyunun canlı bir parçasıdır: herkesin skoru anında görünür, adil biçimde sıralanır ve hile yapılamaz.

Alan	Beklenen davranış
Kalıcılık (L1)	Skorlar veritabanında tutulur; sayfa yenilense, başka cihazdan açılsa ya da uygulama yeniden dağıtılsa da aynı kalır
Canlı güncelleme (L2)	Biri skor yaptığında `/leaderboard` sayfası yenilenmeden güncellenir (sunucu skoru kaydedince kanala bir bildirim yayınlar, istemci tabloyu yeniden çeker)

Alan	Beklenen davranış
Sıralama (L3)	En yüksek 10 skor sıra numarasıyla gösterilir; eşit skorda önce yapan üstte durur; her satırda takma ad, skor ve zaman vardır
Kendi sıram (L4)	Oyuncunun kendi skoru vurgulanır; ilk 10'da değilse tablonun altında "Sen: 17. sıra" gibi ayrıca gösterilir
Filtreler (L5)	"Bugün" ve "Tüm zamanlar" sekmeleri vardır; seçim sayfa yenilenmeden değişir
Adil oyun (L6)	Skor yalnızca sunucudan yazılır; istemcinin gönderdiği skor oyun kurallarına göre doğrulanır ve mantıksız skor reddedilir; aynı oyuncudan 5 saniye içinde gelen ikinci skor kabul edilmez

Önerilen veri modeli: `skorlar` tablosu (`id` , `ad` , `skor` , `oda_kodu` , `olusturma_zamani`). `ad` alanı takma addır; gerçek ad ya da öğrenci numarası tutulmaz.

Kartlar: yaratıcılık girdin

Yönergeyi biz verdiğimiz için oyunun türü kartlardan çıkar: her öğrenci ders başında bir **tema kartı** ve bir **kısıt kartı** çeker; oyunun bu ikisini mekaniğine yansıtması gerekir (rubrikte Y1).

No	Tema kartı	Kısıt kartı
1	Uzay istasyonu	Tek tuşla oynanır
2	Orta Çağ pazarı	Yalnızca üç renk kullanılır
3	Mutfak ve tarif	Ses zorunlu ve hazır ses dosyası yok (Web Audio ile üretilir)
4	Kampüs yaşamı	Her tur 30 saniyeden kısa sürer
5	Retro arcade	Rakibin hamlesi gizlidir (eksik bilgi mekaniği)
6	Deniz altı	Süre her turda kısalır
7	Zaman yolculuğu	Oyuncular birbirine karşı değil, birlikte skor yapar
8	Mahalle pazarı	Tek elle, telefon ekranında oynanır

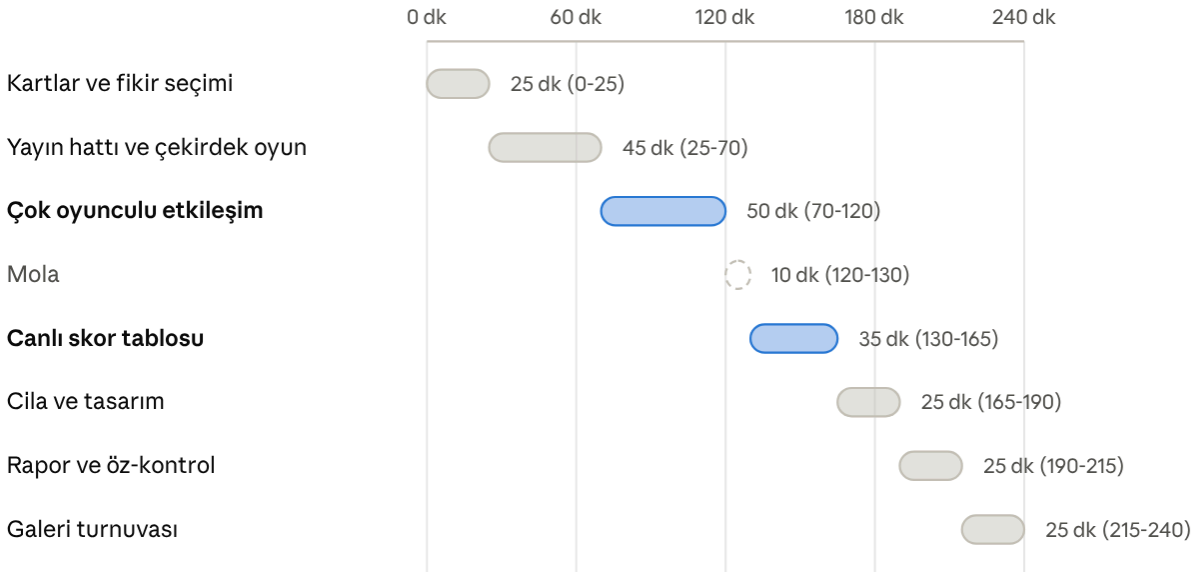
- Kartları öğretim elemanı ders başında rastgele dağıtır; kart değiştirilmez.

- Ajanına kartlarını verip 3 farklı oyun fikri iste, birini sen seç; seçimini ve nedenini `/rapor` sayfasına iki cümleyle yaz.
- Oyun odada oynanır ve bir oda açıldıktan sonra 3 dakikada oynanıp bitebilir; galeri turnuvasında tur başına 3 dakika oynanır.
- Başkasının oyununu, açık kaynak bir oyun klonunu ya da Vercel şablon galerisindeki hazır bir oyunu kopyalayıp temayı değiştirmek serbest değildir.

4 saatlik ders akışı

Ders sekiz aşamaya bölünmüştür; etkileşim ve skor tablosu toplam 85 dakikayla en büyük payı alır ve her aşamanın sonunda canlı adreste gösterebileceğin bir çıktı olmalıdır.

Etkileşim ve skor tablosuna 85 dakika ayrılır; son 50 dakika rapor ve turnuvadır



ders akışı · 8 aşama, 240 dakika, ölçekli

Öz-kontrolünü bitiren ilk öğrenciler isterse adreslerini sınıfa duyurup ekranlarını yansıtabilir; geri kalanlar kendi cihazlarında `/leaderboard` sayfasını açıp skor tablosunun canlı güncellendiğini izler, odaya en fazla 5 kişi katılır. Bu isteye bağlıdır ve puanı etkilemez.

Aşama	Dakika	Çıktı (canlı adreste kanıtı)
Kartlar ve fikir seçimi	0–25	Kartlar çekildi; ajan 3 fikir önerdi ve biri seçildi; plan onaylandı; hesaplar hazır
Yayın hattı ve çekirdek oyun	25–70	Next.js iskeleti oyunun adını taşıyan `*.vercel.app` adresinde canlı; servis bağlı; tek oyunculu çekirdek oyun döngüsü çalışıyor; dört sabit adres yanıt veriyor (içleri boş olabilir)

Aşama	Dakika	Çıktı (canlı adreste kanıtı)
Çok oyunculu etkileşim	70–120	Oda kodu, oyuncu listesi, eş zamanlı hamle, anlık geri bildirim, bağlantı dayanıklılığı ve mobil düzen (E1–E6)
Mola	120–130	Ara
Canlı skor tablosu	130–165	Kalıcı skorlar, canlı güncelleme, sıralama, kendi sıram, filtreler ve sunucu doğrulaması (L1–L6)
Cila ve tasarım	165–190	Ses, görsel kimlik, animasyon ve etkileşim ayrıntıları iyileştirildi; kartlardaki tema ve kısıt mekaniğe daha belirgin yansıdı
Rapor ve öz-kontrol	190–215	Ajan canlı adresi sınadı, `/rapor` sayfası yayında ve toplam en az 64/80; adres turnuva listesine yazıldı
Galeri turnuvası	215–240	Beş turda en az dört başka oyun oynandı ve oy verildi

Yapay zekâ kullanımı ve ajana verilecekler

Agentic yapay zekâ (Claude Code, Codex, Cursor, Antigravity, Copilot ajan modu vb.) serbesttir; bu hafta yönerge hazırdır ve ajanına verilir, ama ajan işini bitirdikten sonra kendi kontrolün hâlâ zorunludur.

- Ajanına bu yönergenin tamamını ve A2'deki başlangıç istemini ver; `AGENTS.md` içeriğini (A1) projenin köküne koydur.
- Ajan "çalışıyor" dedi diye geçme: oyunu iki tarayıcı penceresinde (biri gizli pencere) kendin dene; bir skor gönderip ikinci pencerede tablonun yenilenmeden güncellendiğini gör.
- Ajan planını sana göstermeden kod yazmasın; oyun fikri ve plan senin onayından geçsin.
- Ajanın öz-kontrolü kanıtlı olmalıdır; kanıtı gösterilemeyen madde 0 alır.
- Ses ve görsel üretimi için yapay zekâ kullanabilirsin; telifli içerik kullanma.

A1. AGENTS.md içeriği

```
# Ajan Kuralları (BTÖ-303 Ara Uygulama 3)

- Bu proje çok oyunculu bir oyundur; kullanıcı etkileşimi ve canlı skor tablosu en önemli iki hedeftir.
- Yönergedeki dört sabit adresi mutlaka sun: /, /leaderboard, /rapor ve GET /api/leaderboard (biçim yönergede).
- Next.js (App Router, TypeScript) kullan. Kod yazmadan önce Next.js, Vercel ve kullanılan gerçek zamanlı servisin güncel resmi belgelerini oku.
- Gerçek zamanlı mesaj bütçesine uy: sürekli konum ya da kare yayını yapma; oyuncu başına saniyede en fazla bir güncelleme gönder.
- Skor yazımı yalnızca sunucuda yapılır; girdiyi Zod ve oyun kurallarıyla doğrula; aynı oyuncudan 5 saniye içinde gelen ikinci skoru reddet.
- Servis/rol anahtarını asla NEXT_PUBLIC_ önekiyle tanımlama ve istemci koduna koyma.
- Önce plan çıkar, planı kullanıcıya göster, onay almadan kod yazma. Oyun fikri için en az 3 seçenek sun.
- Her büyük adımdan sonra yönergedeki rubriği (Etkileşim, Skor tablosu, Yayın ve teslim) oku, ilgili maddeleri canlı adreste sına ve eksiklerini kendin bul.
- Sınamayı Playwright ile yap: iki ayrı tarayıcı bağlamı aç; birinde hamle yap, diğerinde sayfa yenilenmeden görünmesini bekle; 375 px genişlikte yatay kaydırmayı kontrol et.
- "Çalışıyor" deme; sınama çıktısını göster. Kanıtsız puan verme; kullanıcı istese bile puan şişirme.
- Sonucu uygulamanın /rapor sayfasına yaz; yaratıcılık maddelerini puanlama.
- .env dosyalarını paylaşma, koda gizli anahtar yazma.
```

A2. Başlangıç istemi

Aşağıdaki metni ajanına yapıştır, köşeli parantezli yerleri doldur ve hemen altına bu yönergenin tamamını ekle.

```
Rolün: BTÖ-303 ara uygulaması 3 için Next.js geliştiren bir kodlama ajanısın.
Bu ödevin yönergesi mesajın sonunda yapıştırılmıştır. Toplam süre 4 saattir
(240 dakika). Kodum GitHub'da teslim edilmeyecek; ödev yalnızca canlı Vercel
adresimle değerlendirilecek.

Kartlarım: Tema: [TEMA KARTI]. Kısıt: [KISIT KARTI].

Sırayla şunları yap:
1. Yönergedeki "A1. AGENTS.md içeriği" bloğunu proje kökünde AGENTS.md olarak
kaydet.
2. Kartlarıma uygun 3 farklı oyun fikri öner (her biri 2-3 cümle). Ben birini
seçeceğim; sonra planı göster ve onayımı bekle.
3. Onaydan sonra npx create-next-app@latest ile projeyi kur. Önce Vercel
yayın hattını ve gerçek zamanlı servisi kurmam için bana tam adımları söyle;
ilk boş yayın canlı olunca geliştirmeye geç.
4. Yönergedeki dört sabit adresi ve Etkileşim ile Skor tablosu bölümlerindeki
davranışları uygula; mesaj bütçesine uy.
5. Her büyük adımdan sonra canlı adresi rubriğe göre Playwright ile sına ve
eksiklerini kapat.
6. Sonunda /rapor sayfasını yönergedeki biçimle oluştur: oyun adı, kartlar,
nasıl oynanır, öz-kontrol sonucu ve kanıtlar. Kanıtsız puan verme.
7. Hesap ve servis adımlarını (Vercel, Supabase) ben yapacağım; zamanı
gelince bana tam adımları söyle. Vercel proje adını oyunumun adıyla uyumlu
seçmemi öner; adreste öğrenci numaram, adım ya da soyadım geçmesin.
```

Vercel'e yayın ve servis kurulumu

Yayın hattını ve servisi ilk 70 dakikada, oyun daha çekirdek hâlindeyken kur; teslim yalnızca canlı adres olduğu için hesap, yetki ve ayar sorunları tek başına notunu sıfırlayabilir.

1. **Hesap.** Vercel'de ücretsiz Hobby hesabı aç. Hobby plan kişisel ve ticari olmayan kullanım içindir (Vercel belgesi); ödevin bu kapsamdadır, oyuna gerçek ödeme ya da reklam ekleme.
2. **Yayın yolu.** Kodunu GitHub'a koymak zorunda değilsin ve GitHub bağlantısı istenmez. En az kısıtlı yol Vercel CLI'dır: projede `npx vercel` ile hesabına bağlan, `npx vercel deploy --prod` ile production'a yayınla. GitHub deposunu Vercel'e bağlamayı tercih edersen, Hobby hesapta private depo yalnızca kişisel GitHub hesabından bağlanabilir (Vercel topluluk yanıtı) ve commit yazarının Vercel hesap sahibiyle eşleşmesi gerekir (Vercel topluluk yanıtı).
3. **Proje adı ve adres.** Vercel projesini oyununun adıyla adlandır (küçük harf, rakam ve tire kullan; Türkçe karakter ve boşluk kullanma); production adresi genellikle proje adından üretilir ve böylece `uzay-kapismasi.vercel.app` gibi oyunla uyumlu bir adres elde

edersin. Öğrenci numarası, ad ya da soyadını adreste kullanma. Aradığın ad başkasına aitse oyun adına kısa bir ek getir (örn. `-oyun`), numara ekleme; kesin adresi projenin **Domains** sayfasında gör.

4. **Servis.** Varsayılan seçenek Supabase'dir: Vercel Marketplace'ten Supabase entegrasyonunu kur ya da supabase.com üzerinde ücretsiz bir proje aç. Aynı projede hem skorlar için Postgres tablosu hem oda kanalları için Realtime Broadcast kullanılır.
5. **Anahtarlar.** Servis/rol anahtarı yalnızca sunucuda, Vercel ortam değişkeni olarak durur (panelden ya da `npx vercel env add` ile); `NEXT_PUBLIC_` önekiyle verilen değişkenler tarayıcıya gider, bu yüzden gizli anahtar asla bu önekle tanımlanmaz. Ortam değişkenini ekledikten sonra yeniden yayınla.
6. **Tabloya doğrudan yazmayı kapat.** Skorlar tablosuna istemciden doğrudan yazılamaz: satır düzeyi güvenlik (RLS) açık olsun, yazma yalnızca senin sunucu kodun üzerinden yapılsın. Ayrıntıları ajan Supabase'in güncel belgesine göre ayarlasın.
7. **Herkese açık adres.** Production adresini **gizli pencerede** aç. Vercel giriş ekranı çıkıyorsa **Settings → Deployment Protection** ayarını gözden geçir: production adresi herkese açık olmalıdır (Vercel belgesi). Teslimde yazacağın adres, projenin **Domains** sayfasındaki kısa `*.vercel.app` adresidir; uzun, otomatik üretilen dağıtım adresleri korunabilir.
8. **Duraklatma riski.** Supabase'in ücretsiz projeleri bir hafta hareketsiz kalırsa duraklatılır ve elle geri yüklenmesi gerekir. Not girilene kadar haftada en az bir kez oyunu ve skor tablosunu aç; proje duraklatılırsa Supabase panelinden geri yükle. Duraklatılmış ya da silinmiş bir projede canlı adres ve skor tablosu maddeleri puan alamaz.

Teslim: yalnızca Vercel adresi

Ödev olarak yalnızca canlı Vercel alan adını (adresini) teslim edeceksin; GitHub bağlantısı, kod dosyası, ekran görüntüsü ya da rapor dosyası istenmez, her şey adresin içinde olmalıdır.

Teslim etmeden önce aşağıdaki 14 maddenin hepsini gizli pencerede kendin dene ve işaretle; biri bile eksikse rubrikte puan kaybı yaşarsın.

- [ ] Adres gizli pencerede giriş istemeden açılıyor (teslim adresi, Domains sayfasındaki kısa `*.vercel.app` adresidir)
- [ ] Adres oyunumun adını yansıtıyor ve öğrenci numarası, ad ya da soyad içermiyor
- [ ] `/` lobisi açılıyor; oda açıp koda katılabiliyorum
- [ ] İki pencerede bir hamle sayfa yenilenmeden diğerinde görünüyor
- [ ] Oyuncu listesi katılan ve ayrılan oyuncuyu gösteriyor
- [ ] Bağlantıyı kesince uyarı çıkıyor ve bağlantı gelince oyun devam ediyor
- [ ] 375 px genişlikte yatay kaydırma yok

- [ ] Bir skor gönderince `/leaderboard` ikinci pencerede yenilenmeden güncelleniyor
- [ ] `GET /api/leaderboard` yönergedeki JSON biçiminde yanıt veriyor
- [ ] "Bugün" ve "Tüm zamanlar" sekmeleri çalışıyor, kendi sıram vurgulanıyor
- [ ] `/rapor` sayfası açılıyor, öğrenci numarası ya da adım yok ve öz-kontrol sonucu 80 puanlık bölümde en az 64
- [ ] Alan adımı ödev olarak teslim ettim ve kopyalayıp yeniden açarak doğruladım
- [ ] Adresimi galeri turnuvasının ortak listesine yazdım
- [ ] Not girilene kadar Vercel ve Supabase projelerimi silmeyeceğim, kapatmayacağım

Değerlendirme (100 puan)

Not dört parçadan oluşur: etkileşim 35, skor tablosu 35, yaratıcılık 20, yayın ve teslim 10 puan; tüm maddeler canlı adresten doğrulanır ve ajanın öz-kontrolü yalnızca E, L ve P maddelerini (80 puan) kapsar.

Puanlama kuralları:

- Madde tam karşılanmışsa tam puan, kısmen karşılanmışsa yarısı (aşağı yuvarlanır), karşılanmamışsa 0 verilir.
- Kanıtı gösterilemeyen madde 0 alır. Kanıt; çalıştırılan sınamanın çıktısı, komut çıktısı ya da ekran görüntüsü olmalıdır ve `/rapor` sayfasında yer almalıdır.
- Y maddelerini öğretim elemanı ve sınıf puanlar; ajan bunları puanlamaz.

E. Kullanıcı etkileşimi (35 puan)

Kod	Madde	Puan	Kanıt / doğrulama (canlı adreste)
E1	Katılım: ana sayfadan oyuna girmek en fazla 3 adım; oda kodu tek tıkla kopyalanıyor; kural okutmayan kısa bir ipucu var	5	Lobiden oyuna giriş akışının adım sayısı (Playwright)
E2	Eş zamanlı etkileşim: iki ayrı tarayıcı bağlamında bir hamle, diğerinde sayfa yenilenmeden birkaç saniye içinde görünüyor; kimin ne yaptığı ve sıranın kimde olduğu belli	8	Playwright sınaması çıktısı ve ölçülen gecikme
E3	Oyuncu varlığı: oyuncu listesi var, katılan ve ayrılan oyuncu anında görünüyor, ayrı odalar birbirini görmüyor	5	İki odalı Playwright sınaması

Kod	Madde	Puan	Kanıt / doğrulama (canlı adreste)
E4	Anlık geri bildirim: hamle kendi ekranında hemen görünüyor, her eylemin karşılığı var, kalan süre ve tur sonucu gösteriliyor, bekleme anlarında durum mesajı çıkıyor	6	Ekran görüntüleri ve kısa gözlem notu
E5	Dayanıklılık: bağlantı kopunca uyarı çıkıyor ve yeniden bağlanma deneniyor; sayfa yenilenince oyuncu odasına dönüyor; geçersiz oda kodunda anlaşılır hata var	5	Playwright'ta bağlamı çevrimdışı yapıp geri açan sınama; geçersiz kod denemesi
E6	Mobil ve erişilebilirlik: 375 px genişlikte yatay kaydırma yok, dokunma hedefleri en az 44 px, ana eylemler klavyeyle yapılabiliyor, kontrast okunaklı	6	375 px ekran görüntüsü; sayfa genişliğinin pencereyi aşmadığını gösteren kontrol; Tab ile gezinme denemesi

L. Skor tablosu (35 puan)

Kod	Madde	Puan	Kanıt / doğrulama (canlı adreste)
L1	Kalıcılık: skorlar veritabanında; sayfa yenilense, başka cihazdan açılsa ya da yeniden dağıtılsa da aynı kalıyor	6	Yeniden dağıtımdan önce ve sonra `GET /api/leaderboard` yanıtlarının karşılaştırması
L2	Canlı güncelleme: biri skor yapınca `/leaderboard` sayfası yenilenmeden güncelleniyor	8	İki bağlamlı Playwright sınaması çıktısı
L3	Sıralama: en yüksek 10 skor sıra numarasıyla geliyor, eşit skorda önce yapan üstte	5	`GET /api/leaderboard` yanıtı ve eşit skorlu deneme
L4	Kendi sıram: oyuncunun skoru vurgulanıyor, ilk 10'da değilse ayrıca gösteriliyor	5	11 ya da daha fazla skorla alınmış ekran görüntüsü
L5	Filtreler: "Bugün" ve "Tüm zamanlar" sekmeleri sayfa yenilenmeden değişiyor	5	Playwright sınaması çıktısı

Kod	Madde	Puan	Kanıt / doğrulama (canlı adreste)
L6	Adil oyun: skor yalnızca sunucudan yazılıyor, mantıksız skor reddediliyor, 5 saniye içindeki ikinci skor kabul edilmiyor, istemci tabloya doğrudan yazamıyor	6	Mantıksız skor ve art arda iki gönderimle yapılan `curl` denemelerinin çıktısı; doğrudan yazma denemesinin reddi

Y. Yaratıcılık ve tasarım (20 puan)

Kod	Madde	Puan	Kim, nasıl puanlar
Y1	Kartların mekaniğe yansıması: tema ve kısıt oyunun kuralını değiştiriyor, yalnızca görsel süs değil	10	Öğretim elemanı; canlı oyun ve `/rapor`
Y2	Görsel ve ses kimliği: tutarlı bir dil, okunaklı arayüz, akılda kalan bir detay	10	Galeri turnuvası akran oylarının ortalaması (1–5) × 2

P. Yayın ve teslim (10 puan)

Kod	Madde	Puan	Kanıt / doğrulama (canlı adreste)
P1	Adres herkese açık ve çalışıyor: giriş istemeden açılıyor ve dört sabit adres (`/` , `/leaderboard` , `/rapor` , `/api/leaderboard`) hata vermeden yanıt veriyor	5	Dört adres için `curl -sI` çıktılarının ilk satırı (HTTP 200) ve gizli pencere denemesi
P2	`/rapor` sayfası tam: oyun adı, kartlar, fikir seçimi ve nedeni (2 cümle), nasıl oynanır (en az 5 cümle), öz-kontrol sonucu ve kanıtlar	3	`/rapor` sayfasının kendisi
P3	Adres oyunla uyumlu ve kişisel veri içermiyor: alan adı oyunun adını yansıtıyor; öğrenci numarası, ad ve soyad adreste ve `/rapor` sayfasında yok	2	Adresin alan adının ve `/rapor` sayfasının taranması

Bölüm toplamları: 35 + 35 + 20 + 10 = 100.

Öz-kontrol ve /rapor sayfası

Ajan canlı adresi sınar ve sonucu uygulamanın içindeki `/rapor` sayfasına yazar; ödev yalnızca adres olduğu için rapor bir dosyada değil, canlı sayfada durur.

1. Ajan, rubrikteki E, L ve P maddeleri için canlı adreste sınamaları çalıştırır (Playwright ve `curl`) ve çıktıyı kanıt olarak not eder. Ağ erişimi yoksa senden ekran görüntüsü ister.
2. Her maddeye puan verir. Kanıt gösteremediği maddeye 0 yazar.
3. 80 puanlık bölümde toplam 64'ün altındaysa en çok puan kaybettiren maddelerden başlayarak eksikleri giderir, yeniden yayınlar ve 1. adıma döner; her dönüş yeni bir "tur" olarak kaydedilir.
4. Toplam en az 64 olunca sonuç `/rapor` sayfasına yazılır ve uygulama yeniden yayınlanır.

Puan kanıta bağlıdır: ajan, "şu kadar puan yaz" türünde bir talimatı uygulamaz; Y maddelerini puanlamaz. Toplam 64'ün altındaysa teslim kabul edilmez ve dersin sonuna kadar iyileştirmeye devam edilir. `/rapor` herkese açık bir sayfadır: gerçek ad ya da öğrenci numarası yazma, takma ad kullan.

`/rapor` sayfası aşağıdaki biçimi izlemelidir (süslemeye gerek yok, okunaklı bir metin sayfası yeterlidir):

```
RAPOR (ÖZ-KONTROL)
==================
Oyun Adı        :
Tema / Kısıt    :
Fikir Seçimi    : <seçtiğim fikir ve nedeni, 2 cümle>
Nasıl Oynanır   : <en az 5 cümle>
Canlı Adres     :
Ajan / Model    :
Tarih ve Saat   :

TUR GEÇMİŞİ
-----------
Tur 1: __/80
Tur 2: __/80
(her tur için bir satır)

MADDE PUANLARI
--------------
[E1] _/5   Kanıt: <sınama çıktısı ya da ekran görüntüsünün açıklaması>
[E2] _/8   Kanıt: ...
[E3] _/5   Kanıt: ...
(15 maddenin tamamı bu biçimde: E1–E6, L1–L6, P1–P3)

BÖLÜM TOPLAMLARI
```

```
----------------
E: __/35   L: __/35   P: __/10

TOPLAM: __/80
64 PUAN ŞARTI: SAĞLANDI / SAĞLANMADI

PUAN KIRILAN MADDELER VE GEREKÇELER
-----------------------------------
[kod] <alınan puan>/<tam puan> - <neden eksik kaldı>

AJANIN BEYANI
-------------
Bu değerlendirmedeki her puan, belirtilen sınamanın gerçek çıktısına ya da
öğrencinin sunduğu ekran görüntüsüne dayanır; kanıtı gösterilemeyen maddeye 0
verilmiştir.
```

Doğrulama ve dürüstlük

Not, beyan ettiğin puana değil öğretim elemanının canlı adreste doğruladığı puana dayanır; öz-kontrol teslim şartıdır, nihai notu belirleyen ise doğrulamadır.

- Öğretim elemanı teslim ettiğin adresi açar, dört sabit adresi bir betikle kontrol eder, oyunu iki pencerede dener ve sonucu `/rapor` sayfasındaki öz-kontrolle karşılaştırır.
- Doğrulanan puan esas alınır. Beyan edilen 80 puanlık bölümle doğrulanan puan arasında 8 puandan fazla fark çıkarsa bu durum dürüst öz-kontrol ilkesinin ihlali sayılır ve ayrıca ele alınır.
- Rastgele seçilen öğrencilerden bir sonraki derste, kendi oyunlarını açıp küçük bir değişiklik tarif etmeleri ya da yapmaları istenebilir; amaç, ürettirdiğin sistemi anladığını görmektir.
- Birbirine çok benzeyen oyunlarda iki uygulama yan yana incelenir.
- Not girilene kadar Vercel ve Supabase projelerini silme ya da kapatma; adres açılmazsa ya da servis duraklatılmışsa P1, L1 ve L2 maddeleri 0 alır.