

okanocodes feat: added project report ✓ ecf0d2e · 5 days ago

public	initial commit	last week
server	feat: add basic email/password auth ...	5 days ago
src	feat: add demo user quick-login butt...	5 days ago
.gitignore	refactor: fix AI speech. Added eleven ...	5 days ago
.oxlintrc.json	initial commit	last week
CEX: AI Çağrı Asistanı Bireysel ...	feat: added project report	5 days ago
README.md	feat: add basic email/password auth ...	5 days ago
index.html	initial commit	last week
package-lock.json	initial commit	last week
package.json	refactor: fix AI speech. Added eleven ...	5 days ago
tsconfig.app.json	initial commit	last week
tsconfig.json	initial commit	last week
tsconfig.node.json	initial commit	last week
vite.config.ts	initial commit	last week

README



CEX — AI Çağrı Platformu

Önceden tanımlanmış senaryolara ve kural setlerine göre otomatik olarak arama yapan veya gelen aramaları cevaplayan bir AI çağrı platformu MVP'si. Arayüz tamamen Türkçe'dir.

Mimari

- Frontend** (repo kökünde): React + Vite + TypeScript + Tailwind. Sayfalar: Panel, Senaryolar, Arama Günlükleri, Arama Başlat, Ayarlar.
- server/**: Node.js + Express + WebSocket backend. STT ve LLM çağrıları Hugging Face üzerinden, TTS (metinden sese) çağrıları ise ElevenLabs üzerinden yapılır — canlı HF Inference Providers kataloğunda güvenilir Türkçe konuşan bir TTS modeli bulunmadığı için bu parça HF dışında tutuldu (bkz. aşağıdaki not). API anahtarları/token'lar yalnızca burada tutulur, tarayıcıya asla gönderilmez.

- **Veritabanı:** Postgres. Tüm senaryolar/aramalar/ayarlar `DATABASE_URL` ile bağlanılan bir Postgres'te tutulur — yerel bir dosyada değil, bu sayede deploy/restart'larda veri kaybolmaz.
- **Arama modu:** MVP varsayılan olarak **tarayıcı simülasyonu** ile çalışır — mikrofonla konuşup AI'nın sesli yanıtını dinlediğiniz, gerçek telefon hattı gerektirmeyen bir döngü (basılı-tut-konuş / turn-based). Gerçek telefon araması (Twilio) için altyapı hazır şekilde eklenmiştir ancak Ayarlar sayfasından kendi Twilio hesap bilgilerinizi girmeden aktif olmaz.
- **Kimlik doğrulama:** Basit e-posta/şifre ile kayıt+giriş (JWT). Bu bir erişim kapısıdır — giriş yapan herkes aynı paylaşılan senaryo/arama/ayarlar verisini görür, kullanıcı başına veri izolasyonu yoktur. Şifre sıfırlama/e-posta doğrulama MVP kapsamı dışı.

Kurulum

```
npm run install:all # hem frontend hem server bağımlılıklarını kurar
cp server/.env.example server/.env
```



`server/.env` içine dört şeyi doldurun:

1. `HF_TOKEN` — <https://huggingface.co/settings/tokens> (STT + LLM için)
2. `ELEVENLABS_API_KEY` — <https://elevenlabs.io/app/settings/api-keys> (TTS için)
3. `DATABASE_URL` — bir Postgres bağlantı adresi. İki seçenek:
 - **Yerel geliştirme:** `docker run -d --name cex-postgres -e POSTGRES_PASSWORD=cex -e POSTGRES_DB=cex -p 5433:5432 postgres:16-alpine` çalıştırıp `postgres://postgres:cex@localhost:5433/cex` kullanın.
 - **Ücretsiz barındırılan:** [Neon](#) veya [Supabase](#) üzerinde bir proje açıp verilen bağlantı adresini yapıştırın — Supabase kullanıyorsanız **"Direct connection"** değil, **"Session pooler"** adresini kullanın (bkz. Deploy bölümü).
4. `JWT_SECRET` — `openssl rand -hex 32` ile üretilmiş rastgele bir değer (giriş/kayıt token'larını imzalamak için).

HF token ve ElevenLabs anahtarı uygulama içinden Ayarlar sayfasına da girilebilir; `DATABASE_URL` ve `JWT_SECRET` ise yalnızca `.env` 'den okunur.

İlk kullanımda tarayıcıda **Kayıt Ol** sayfasından bir hesap oluşturun — kayıt olan herkes uygulamanın tamamına erişebilir (bkz. yukarıdaki Kimlik doğrulama notu).

Geliştirme

```
npm run dev
```



Bu komut frontend'i (<http://localhost:5173>) ve backend'i (<http://localhost:4000>) birlikte başlatır; Vite, `/api` ve `/ws` isteklerini otomatik olarak backend'e yönlendirir.

Kullanım akışı

0. **Kayıt Ol** sayfasından bir hesap oluşturun (veya varsa **Giriş Yap**) — uygulamanın geri kalanı girişe kapalıdır.
1. **Ayarlar** sayfasından Hugging Face token'ınızı ve ElevenLabs API anahtarınızı girin (girilmemişse aramalar STT/LLM/TTS hatası verir).
2. **Senaryolar** sayfasından bir senaryo oluşturun: sistem promptu, karşılama mesajı ve isteğe bağlı kurallar (anahtar kelime eşleşince LLM çağrılmadan sonlandırma / özel yanıt / değişken kaydetme).
3. **Arama Başlat** sayfasından senaryoyu seçip test araması başlatın; mikrofon butonunu basılı tutup konuşun, bırakınca AI yanıtlar.

4. **Arama Günlükleri**'nde geçmiş çağrıların transkriptini, toplanan değişkenleri ve tahmini API kullanımını görebilirsiniz.

Maliyet/token optimizasyonu

Her API çağrısı bir maliyet/kota tükettiği için orchestrator şu önlemleri uygular: kural eşleşen dönüşlerde LLM'i tamamen atlama, LLM'e yalnızca son ~6 turluk kaydırmalı geçmiş gönderme, sabit `max_tokens` sınırı, tekrarlanan metinler için TTS ses önbellesi, sessiz kayıtları hiç göndermeme ve çağrı başına maksimum tur sınırı (Ayarlar'dan yapılandırılabilir).

Neden TTS için Hugging Face değil ElevenLabs?

Araştırma sonucu: Hugging Face Inference Providers üzerinde canlı (live) durumda olan TTS modelleri arasında güvenilir Türkçe desteği sunan tek bir model bile yok. `ResembleAI/chatbox` gibi çokdilli olduğu iddia edilen modellerin bile HF üzerinden erişilebilen tek canlı ucu İngilizce'ye sabit (`chatbox`'ın gerçek çokdilli modu farklı, HF'de kayıtlı olmayan bir `fal.ai` rotasında). Bu yüzden STT (Whisper) ve LLM (Qwen) Hugging Face'te kalırken, TTS için doğrulanmış Türkçe desteği olan ElevenLabs (`eleven_multilingual_v2`) kullanılıyor.

Deploy

Uygulama tek bir servis olarak deploy edilir: `server/` derlenmiş halde hem API/WebSocket'i hem de derlenmiş React uygulamasını (repo kökündeki `dist/`) aynı porttan sunar — ayrı bir statik host veya CORS ayarına gerek yoktur.

```
npm run build:all # frontend + server derlenir
npm start        # server/dist/index.js, hem API'yi hem derlenmiş frontend'i tek porttan sunar
```



Railway, Render, Fly.io gibi bir platforma deploy ederken:

1. Build komutu: `npm run install:all && npm run build:all`
2. Start komutu: `npm start`
3. Ortam değişkenleri: `HF_TOKEN` , `ELEVENLABS_API_KEY` , `JWT_SECRET` , `DATABASE_URL` (Neon/Supabase gibi bir Postgres'e işaret etsin), `PORT` (çoğu platform bunu otomatik sağlar)
4. WebSocket desteği açık olmalı — seçtiğiniz platformun bunu desteklediğini doğrulayın (Railway/Render/Fly.io hepsi destekler).
5. **Supabase kullanıyorsanız:** `DATABASE_URL` için "Direct connection" değil "**Session pooler**" veya "**Transaction pooler**" adresini kullanın (Supabase dashboard → Connect). Direct connection yalnızca IPv6'ya çözümleniyor ve çoğu platformun (Render dahil) giden IPv6 desteği olmadığı için `ENETUNREACH` hatasına yol açıyor.

Veritabanı artık Postgres olduğu için veri, platformun dosya sistemi ephemeral olsa bile redeploy/restart'larda kaybolmaz — tek şart `DATABASE_URL` 'nin kalıcı bir Postgres'e işaret etmesi.

Twilio (gerçek arama) — opsiyonel, ileri seviye

`server/src/providers/twilioProvider.ts` ve `server/src/routes/twilioWebhooks.ts` dosyalarında ne yapılması gerektiği belgelenmiştir: kendi Twilio hesabınız, satın alınmış bir telefon numaranız ve genel erişilebilir bir webhook URL'niz (örn. `ngrok http 4000`) olması gerekir. Bu kimlik bilgileri girilip Ayarlar'da mod "twilio" yapılmadan bu yol devre dışı kalır.

Notlar / bilinen sınırlamalar (MVP kapsamı)

Releases

No releases published

Packages

No packages published

Contributors 1



okanocodes Okan Şahin

Languages

• TypeScript 99.5% • Other 0.5%