

İkinci Aşama Teknik Görev – Laravel Mini Uygulama

Aşağıdaki gereksinimleri karşılayan küçük bir Laravel uygulaması geliştirmenizi bekliyoruz. Bu görev, yazılım geliştirme yaklaşımınızı, API ile çalışma becerinizi, backend & frontend entegrasyonunuzu ve kod kalitenizi değerlendirmek için hazırlanmıştır.

Görev Tanımı

Laravel kullanarak basit bir ürün listeleme ve sepet yönetimi uygulaması geliştirin. Uygulama, **api.hyperteknoloji.com.tr** adresindeki API üzerinden ürünleri çekecek ve kullanıcıya minimal bir arayüzde sunacaktır.

API Entegrasyonu

- Tüm API istekleri **backend (Laravel)** üzerinden yapılmalıdır. Frontend doğrudan API ile konuşmamalıdır.
 - API erişimi için **ApiKey** ve **Bearer Token** kullanılacaktır. (Detaylar API dokümantasyonunda bulunmaktadır.)
 - API dokümantasyonu: <https://api.hyperteknoloji.com.tr/documentation>
- ApiKey:** b5724e7b7a2c487080af6dc239c9766e
- Token:**
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJMb2dpblR5cGUOiOiIiwiQ3VzdG9tZXJJRCI6IjI1Mzc2IiwiaW9ja3R5b250dHlifQ.CanVMlhU_iPbgOdY5iJulsyfL0FLXnPoHMBzn-r_5mY

Ürün Listeleme Gereksinimleri

- Ürünler API'den **sayfalama (pagination)** kullanılarak çekilmelidir.
- Backend tarafında performans için **cache mekanizması** uygulanmalıdır. Örneğin cache süresi, kullanılan cache anahtarları, sayfa bazlı cache ve cache invalidation stratejisinin açıklanması beklenir.
- Backend, sayfalı ürün listesini frontend'e JSON (SPA tercih ediliyorsa) veya Blade template ile sunabilir.
- Arayüzde ürünler **kart tasarımı** ile listelenmelidir. Kartlarda:
 - Ürün görseli
 - Ürün fiyatı

- “Sepete Ekle” butonu

Sepet İşlemleri

- Kullanıcı ürünleri sepete ekleyebilmelidir.
- Sepet verileri **veritabanı (migrations + models)** veya **cookie**/local storage içinde tutulabilir. Hangi yöntemi seçtiğinizi ve nedenini README’de açıklayın.
- Sepet sayfasında eklenen ürünlerin listesi gösterilmelidir. Toplam tutar hesaplanmalıdır.

Teknik Beklentiler

- Laravel 9 ve üzeri kullanabilirsiniz.
- Temiz kod, SOLID prensipleri, servis katmanı veya repository pattern tercih sebebidir.
- API isteklerinde hata yakalama ve logging mekanizmaları olmalıdır.
- Cache kullanımı açıkça gösterilmelidir (örn. Cache::remember, TTL, cache anahtarı stratejisi).
- Gerekli migration, model, controller, service ve view dosyaları oluşturulmalıdır.
- Minimal, responsive bir tasarım beklenir (Tailwind/Bootstrap gibi frameworkler kullanılmamalıdır).

Teslimat

1. **GitHub deposu (public veya private)** — erişim bilgisi paylaşılmalıdır.
2. **README.md** içinde kurulum adımları, çalıştırma talimatları ve tercihlerin kısa açıklaması.
3. Kısa bir **teknik not**: kullandığınız cache stratejisi, sepete veri saklama yöntemi, bilinen eksiklikler.
4. Çalışan uygulamanın ekran görüntüleri veya kısa bir demo (opsiyonel).

Değerlendirme Kriterleri

- API entegrasyonunun doğruluğu ve güvenliği (ApiKey/Bearer yönetimi)
- Cache ve performans yaklaşımları
- Kod düzeni, katmanlı mimari ve test edilebilirlik
- Teslimat kalitesi

Süre

Görevi teslim etmek için **48 saat** süreniz vardır.