

TheFirstStepsTR.com - Proje Finalizasyon ve Canlıya Alma Planı

Bu döküman, teknik planların (Java Backend ve React Frontend) tamamlanmasının ardından projenin başarılı bir şekilde geliştirilmesi, güvenliğinin sağlanması ve canlıya alınması için gereken son adımları listeler.

1. Altyapı ve DevOps Planı (Deployment & Monitoring)

Adım	Açıklama	Teknoloji	Sorumluluk
1.1. CI/CD Pipeline	Kod değişikliklerinde otomatik test, build ve deployment'i tetikler.	GitHub Actions / GitLab CI	DevOps / Backend
1.2. Konteynerleştirme	Java Backend, PostgreSQL ve Redis servislerinin Docker imajlarının oluşturulması.	Docker / Docker Compose	Backend
1.3. Ortam Tanımları	Geliştirme (DEV), Test (TEST) ve Canlı (PROD) ortam değişkenlerinin yönetilmesi.	Spring Profiles (.yml) / Environment Variables	Backend / DevOps
1.4. Loglama ve İzleme	Hata takibi, performans monitörü ve metriklerin toplanması (Hataalarları hayatı önemdedir).	Prometheus & Grafana / ELK Stack (Opsiyonel)	DevOps
1.5. Frontend Hosting	React uygulamasının statik dosyalarının hız için bir CDN	Netlify / Vercel / Cloudflare Pages	Frontend / DevOps

	arkasında sunulması.		
--	----------------------	--	--

2. Güvenlik ve Veri Bütünlüğü (Security & Database)

Güvenlik, e-ticaret için en kritik adımdır.

2.1. API Güvenliği (Java Backend)

- Rate Limiting:** Kullanıcıları ve botları aşırı istek atmaktan korumak için API Gateway'e hız sınırlayıcı kuralların eklenmesi.
- CORS:** React uygulamasının Backend API'sine güvenli bir şekilde erişimini sağlayan CORS (Cross-Origin Resource Sharing) ayarlarının yapılması.
- Veri Maskelenme:** Loglarda hassas bilgilerin (kredi kartı son 4 hane, tam adres) maskelenmesi.

2.2. Veritabanı ve Firestore Güvenlik Kuralları

Canvas ortamında zorunlu olan **Firestore** için (kullanıcı sepeti, mesajlaşma gibi amaçlarla), verilerin sadece yetkili kullanıcılar tarafından okunup yazılabilmesi için kuralların tanımlanması:

- Özel Veri (Private):** Kullanıcıların kendi sipariş geçmişi gibi veriler sadece kendi userId'leri ile erişilebilir olmalıdır.
 - Yol:** /artifacts/{appId}/users/{userId}/{koleksiyon}
- Genel Veri (Public/Shared):** Ürün yorumları gibi herkesin görmesi gereken veriler.
 - Yol:** /artifacts/{appId}/public/data/{koleksiyon}

3. Kalite Güvencesi (QA) ve Test Stratejisi

Tıpkı Java'da yazdığınız Unit Testleri gibi, projenin her katmanı test edilmelidir.

Test Türü	Kapsam	Araçlar
3.1. Unit Test (Birim)	Backend'de iş mantığı (Service katmanı), Frontend'de küçük React bileşenleri.	JUnit 5 (Java), Jest / Vitest (React)
3.2. Integration Test (Entegrasyon)	Java API'sinin veritabanı (PostgreSQL) ve Redis ile doğru konuşup konuşmadığı.	Testcontainers (Java)

3.3. E2E Test (Uçtan Uca)	Kullanıcının ana sayfadan başlayıp, sepete ürün ekleyip ödeme yapma akışının simülasyonu.	Cypress / Playwright
3.4. Performans Testi	Yüksek trafik altında (Örn: Black Friday) sistemin tepki süresi ve eş zamanlı kullanıcı sayısı.	JMeter / Gatling (Java tabanlıdır)

4. Canlıya Geçiş (Go-Live Checklist)

- Domain & SSL:** thefirststepstr.com domaininin yönlendirilmesi ve zorunlu SSL sertifikasının (HTTPS) aktif edilmesi.
- CDN Optimizasyonu:** React statik dosyaları için Content Delivery Network ayarlarının yapılması.
- Hata Sayfaları:** 404, 500 ve Bakım Sayfalarının (Maintenance) tasarılanıp aktif edilmesi.
- GDPR/KVKK Uyumluluğu:** Çerez bildirimi ve gizlilik politikası sayfalarının eklenmesi.

Bu son dokümanla birlikte, **thefirststepstr.com** projesinin tüm ön hazırlıkları tamamlanmıştır. Artık klavyeyi elinize alıp, bu planları koda dökmeye başlayabilirsiniz!