

Veerify Mimari Tasarım Raporu

1. Proje Özeti

Proje Adı: Veerify

Proje Tipi: Web Uygulaması + Browser Extension

Geliştirme Süresi: 10-12 hafta

2. Misyon & Vizyon

Misyon

Kullanıcıların karşılaştığı haberlerin güvenilirliğini, birden fazla güvenilir kaynaktan otomatik olarak toplanan verilerle karşılaştırarak değerlendiren, veri odaklı bir doğrulama sistemi geliştirmek. Sistem, haberin ne kadar güvenilir kaynak tarafından doğrulandığını analiz ederek kullanıcılara objektif bir güvenilirlik skoru sunmayı hedefler.

Vizyon

İnternette dolaşımdaki haberlerin güvenilirliğini anlık olarak değerlendirebilen, kullanıcı dostu bir platform oluşturarak dezenformasyonla mücadelede bireyleri güçlendirmek. Uzun vadede Türkçe haber ekosisteminde en kapsamlı doğrulama veri tabanını oluşturmak.

Temel Değer Önerisi

- Çoklu Kaynak Doğrulama:** Tek bir kaynağa güvenmek yerine, birden fazla güvenilir kaynaktan veri toplama
 - Veri Odaklı Yaklaşım:** Her doğrulama kararı somut veri ve istatistiklere dayanır
 - Şeffaflık:** Kullanıcıya hangi kaynakların haberi doğruladığını açıkça gösterme
 - Trend Takibi:** Hangi haberlerin viral olduğunu ve doğruluk oranlarını izleme
-

3. Temel Fonksiyonel Gereksinimler

3.1 Kullanıcı Fonksiyonları

- Kullanıcı kayıt ve giriş sistemi
- Haber URL'i veya metin girişi ile doğrulama talebi
- Doğrulama sonuçlarını görüntüleme (güvenilirlik skoru, kaynak listesi)
- Geçmiş doğrulama taleplerini görüntüleme
- Güncel trendleri ve popüler haberleri görüntüleme

3.2 Browser Extension Fonksiyonları (İleri Aşama)

- Aktif haber sayfasını otomatik algılama
- Tek tıkla doğrulama başlatma
- Popup'ta hızlı sonuç görüntüleme
- Sayfa içi güvenilirlik badge'i gösterme

3.3 Admin/Sistem Fonksiyonları

- Güvenilir kaynak listesi yönetimi
 - Sistem loglarını inceleme
 - Kullanıcı aktivite analizi
 - Scraping performans metrikleri
-

4. Sistem Mimarisi

4.1 Genel Mimari Yaklaşım

Mikroservis-Uyumlu Monolitik Yaklaşım: Başlangıçta tek bir backend uygulaması olarak başlayıp, servisleri modüler şekilde tasarlayarak ileride mikroservislere geçiş yapılabilir hale getirme.

5. Teknoloji Stack

5.1 Frontend Layer

Web Application

Framework: React

Neden React?

- Geniş topluluk desteği ve dokümantasyon

- Component-based mimari ile modüler geliştirme
- Extension geliştirmede de JavaScript bilgisi tekrar kullanılabilir
- Hızlı prototipleme için zengin kütüphane ekosistemi

Ek Teknolojiler:

- **State Management:** React Context API (başlangıç için) / Redux Toolkit (gerekirse)
- **Routing:** React Router v6
- **UI Framework:** Tailwind CSS (hızlı styling) + shadcn/ui (hazır componentler)
- **HTTP Client:** Axios
- **Form Management:** React Hook Form + Zod (validation)
- **Charts:** Recharts (trend görselleştirme için)

Browser Extension

Teknoloji: Vanilla JavaScript (Manifest V3)

Neden Vanilla JS?

- Extension için hafiflik kritik
- Build süreci gerektirmez
- React kullanımı extension'da gereksiz overhead yaratır

Extension Bileşenleri:

- **Content Script:** Sayfa içeriğini okuma
- **Background Service Worker:** API iletişimi
- **Popup:** Kullanıcı arayüzü
- **Storage API:** Kullanıcı ayarları için

5.2 Backend Layer

Primary Backend Framework

Framework: FastAPI (Python 3.11+)

Neden FastAPI?

- Otomatik API dokümantasyonu (Swagger/OpenAPI)
- Yüksek performans (Starlette + Pydantic)
- Async/await desteği (I/O-heavy işlemler için ideal)
- Type hints ile güvenli geliştirme
- NLP ve ML kütüphaneleriyle uyum

Kütüphaneler:

- **FastAPI** - Web framework
- **SQLAlchemy** - ORM (PostgreSQL için)
- **Alembic** - Database migrations
- **PyMongo** - MongoDB driver
- **Redis-py** - Redis client
- **Celery** - Distributed task queue
- **Pydantic** - Data validation
- **python-jose** - JWT handling
- **passlib** - Password hashing
- **httpx** - Async HTTP client

5.3 Data Collection & Processing Layer

Web Scraping

Framework: Scrapy + BeautifulSoup4

Neden Scrapy?

- Profesyonel web scraping için endüstri standardı
- Built-in rate limiting, concurrent requests
- Middleware sistemi ile kolay özelleştirme
- robots.txt otomatik saygı

BeautifulSoup4: Basit, tek sayfa parse işlemleri için

Scraping Stratejisi:

1. **Scheduled Scraping:** Her 30 dakikada major haber sitelerini tara
2. **On-Demand Scraping:** Kullanıcı talebi geldiğinde spesifik URL'leri tara
3. **Rate Limiting:** Her site için özel rate limit (robots.txt + etik kurallar)

Hedef Haber Kaynakları (Türkiye için):

- Güvenilir kaynaklar: BBC Türkçe, Deutsche Welle Türkçe, Euronews TR
- Ana akım medya: CNN Türk, NTV, Hürriyet, Sözcü, Cumhuriyet
- Doğrulama siteleri: [Teyit.org](https://teyit.org), Doğruluk Payı

News APIs

Kullanılacak API'ler:

- **[NewsAPI.org](https://newsapi.org):** 50+ Türkçe kaynak, günde 100 ücretsiz request
- **GDELT Project:** Global haber veritabanı, ücretsiz
- **Bing News Search API:** Microsoft'un haber API'si

NLP & Text Analysis

Framework: spaCy + Sentence Transformers

spaCy (v3.7+):

- Türkçe dil modeli: `tr_core_news_lg`
- Named Entity Recognition (NER)
- Part-of-speech tagging
- Dependency parsing

Sentence Transformers:

- Model: `sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2`
- Haber benzerliği hesaplama için
- Semantic search için embedding oluşturma

İşlem Pipeline:

1. Metin temizleme (HTML tag'leri, özel karakterler)
2. Tokenization ve normalizasyon
3. Embedding oluşturma (512-dim vector)
4. Cosine similarity hesaplama
5. Benzer haberleri bulma

5.4 Data Layer

PostgreSQL (Primary Database)

Versiyon: PostgreSQL 15+

Neden PostgreSQL?

- ACID uyumluluğu (veri tutarlılığı kritik)
- JSON desteği (hybrid data için)
- Full-text search yetenekleri
- Excellent indexing (performance)
- Ücretsiz ve açık kaynak

PostgreSQL Extensions:

- **pgvector:** Vector similarity search için
- **pg_trgm:** Fuzzy text search için
- **uuid-ossdp:** UUID generation için

MongoDB (Semi-Structured Data)

Versiyon: MongoDB 7.0+

Neden MongoDB?

- Flexible schema (scraping sonuçları değişken yapıda olabilir)
- JSON-like documents (API response'ları direkt saklanabilir)
- Horizontal scaling capability
- Yüksek yazma performansı (logging için ideal)

Redis (Caching & Session Management)

Versiyon: Redis 7.2+

Neden Redis?

- Sub-millisecond latency
- Rich data structures (Strings, Hashes, Lists, Sets, Sorted Sets)
- Built-in TTL support
- Pub/Sub for real-time features
- Celery broker olarak da kullanılabilir

Monitoring & Logging

- **Application Logging:** Python logging + structlog
- **Log Aggregation:** MongoDB (custom solution) veya ELK Stack (advanced)

Version Control & CI/CD

- **Git:** GitHub/GitLab
- **Branch Strategy:** Git Flow (main, develop, feature/, *hotfix*/)
- **CI/CD:** GitHub Actions (basit pipeline)