**SAP Nexus AI**

Mevut verilerle geliştirebileceğimiz "SAP Nexus AI" platformunun gerçekçi özelliklerini şöyle sıralayabiliriz:

1. Stok ve Envanter Yönetimi Modülü:

* Mevcut stok seviyelerinin gerçek zamanlı takibi
* Ürün bazında minimum stok seviyesi hesaplaması ve uyarı sistemi
* Rezerve edilen stokların otomatik yönetimi
* Stok tükenmesi riski analizi ve erken uyarı sistemi

1. Sipariş Analizi ve Optimizasyonu:

* Müşteri bazında sipariş pattern analizi
* Bölgesel sipariş dağılımlarının analizi
* Sipariş-stok ilişkisi optimizasyonu
* Otomatik sipariş önceliklendirme sistemi

1. İş Zekası ve Raporlama:

* Bölgesel performans analizleri
* Müşteri segmentasyonu ve analizi
* Ürün bazlı satış performans raporları
* Özelleştirilebilir dashboard'lar

1. Tahminleme ve Planlama:

* Ürün bazında satış tahmini
* Stok ihtiyaç planlaması
* Bölgesel talep tahmini
* Sezonsal trend analizi

1. Müşteri İlişkileri Yönetimi:

* Müşteri davranış analizi
* Müşteri segmentasyonu
* Sipariş geçmişine dayalı müşteri öngörüleri
* Çapraz satış fırsatları tespiti

Platform Özellikleri:

* SAP sistemi ile tam entegrasyon
* Gerçek zamanlı veri senkronizasyonu
* Rol bazlı erişim kontrolü
* Özelleştirilebilir raporlama arayüzü
* Mobil uyumlu tasarım

Önemli Notlar:

1. Mevcut veri setinin sınırlı olması nedeniyle, başlangıçta odaklanılacak temel özellikler yukarıda listelenmiştir.
2. Platform'un başarılı olabilmesi için ihtiyaç duyulan ek veriler:

* Daha uzun dönemli satış geçmişi
* Detaylı müşteri bilgileri
* Ürün kategorileri ve özellikleri
* Fiyat geçmişi
* Tedarik süreleri
* Maliyet verileri

1. Geliştirme Stratejisi:

* İlk aşamada mevcut verilerle çalışan temel özelliklerin geliştirilmesi
* Müşteri geri bildirimleri doğrultusunda özelliklerin genişletilmesi
* Yeni veri kaynaklarının eklenmesiyle fonksiyonalitenin artırılması

1. Başlangıç için Önerilen Odak Noktaları:

* Stok optimizasyonu
* Temel sipariş analizi
* Basit tahminleme modelleri
* Temel raporlama özellikleri

**SAP Nexus AI - Kapsamlı Sistem Dokümantasyonu**

**İçindekiler**

1. [Proje Özeti](#1-proje-%C3%B6zeti)
2. [Sistem Mimarisi](#2-sistem-mimarisi)
3. [Teknoloji Yığını](#3-teknoloji-y%C4%B1%C4%9F%C4%B1n%C4%B1)
4. [Frontend Mimarisi](#4-frontend-mimarisi)
5. [Backend Servisleri](#5-backend-servisleri)
6. [ML/AI Bileşenleri](#6-mlai-bile%C5%9Fenleri)
7. [Veri Mimarisi](#7-veri-mimarisi)
8. [Deployment ve DevOps](#8-deployment-ve-devops)
9. [Güvenlik ve Uyumluluk](#9-g%C3%BCvenlik-ve-uyumluluk)
10. [Kalite Güvence](#10-kalite-g%C3%BCvence)
11. [Operasyonel Prosedürler](#11-operasyonel-prosed%C3%BCrler)

**1. Proje Özeti**

SAP Nexus AI, işletmelerin SAP sistemlerindeki verilerini kullanarak operasyonel verimliliği artırmayı hedefleyen, yapay zeka destekli bir iş zekası ve optimizasyon platformudur. Platform, stok yönetimi, talep tahmini, sipariş optimizasyonu gibi kritik iş süreçlerini otomatize eder ve optimize eder.

**1.1 Temel Hedefler**

* Operasyonel verimliliği artırma
* Manuel süreçleri otomatize etme
* Veri odaklı karar verme süreçlerini destekleme
* İş süreçlerini optimize etme
* Maliyetleri düşürme

**1.2 Başlıca Özellikler**

* Akıllı stok yönetimi ve optimizasyonu
* İleri düzey talep tahmini
* Otomatik sipariş yönetimi
* Gerçek zamanlı analitik ve raporlama
* Dinamik dashboard'lar
* Özelleştirilebilir uyarı sistemi

**2. Sistem Mimarisi**

**2.1 Mimari Prensipler**

* Microservices tabanlı mimari
* Event-driven tasarım
* Domain-driven design
* Cloud-native yaklaşım
* Loose coupling, high cohesion
* Horizontally scalable yapı

**2.2 Sistem Bileşenleri**

1. Veri Katmanı
   * Veri alım servisleri
   * ETL pipeline'ları
   * Veri doğrulama sistemleri
   * Veri depolama servisleri
2. İşlem Katmanı
   * ML model servisleri
   * İş mantığı servisleri
   * Event processing servisleri
   * Caching servisleri
3. Sunum Katmanı
   * API Gateway
   * Frontend uygulaması
   * Bildirim servisi
   * Raporlama motoru

**3. Teknoloji Yığını**

**3.1 Backend Teknolojileri**

* Ana Dil: Python 3.11+
* API Framework: FastAPI
* Message Broker: Apache Kafka
* Ana Veritabanı: PostgreSQL
* Time Series DB: TimescaleDB
* Cache: Redis
* Container: Docker & Kubernetes

**3.2 Frontend Teknolojileri**

* Framework: Next.js 14+
* State Management: Redux Toolkit
* UI Library: Tailwind CSS + shadcn/ui
* Görselleştirme: Apache ECharts
* Table Management: TanStack Table
* Type System: TypeScript

**3.3 ML/AI Teknolojileri**

* ML Framework: Scikit-learn & PyTorch
* Model Management: MLflow
* Optimization: Optuna
* Feature Store: Feast
* Monitoring: Prometheus + Grafana

**4. Frontend Mimarisi**

**4.1 Ana Komponentler**

1. Dashboard Sistemi
   * KPI görselleştirmeleri
   * Interaktif grafikler
   * Özelleştirilebilir widget'lar
   * Gerçek zamanlı güncelleme
2. Analitik Modülleri
   * Detaylı veri analizi
   * Özelleştirilebilir raporlar
   * Drill-down yetenekleri
   * Export fonksiyonları
3. Yönetim Arayüzleri
   * Stok yönetimi
   * Sipariş takibi
   * Tahmin görüntüleme
   * Ayar yönetimi

**4.2 UX/UI Özellikleri**

* Responsive tasarım
* Tema desteği
* Erişilebilirlik standartları
* Performans optimizasyonları

**5. Backend Servisleri**

**5.1 Çekirdek Servisler**

1. Data Integration Service
   * SAP connector
   * Data validation
   * Transformation pipeline
   * Error handling
2. Business Logic Service
   * Order processing
   * Inventory management
   * Business rules engine
   * Event processing
3. Analytics Service
   * Report generation
   * KPI calculations
   * Data aggregation
   * Export functionality

**6. ML/AI Bileşenleri**

**6.1 Model Mimarisi**

1. Stok Optimizasyonu Modeli
   * Algoritma: Ensemble (LightGBM + LSTM)
   * Giriş verileri: Stok hareketleri, sipariş trendi
   * Çıktılar: Optimal stok seviyeleri, reorder noktaları
2. Talep Tahmin Modeli
   * Algoritma: Prophet + GradientBoost Ensemble
   * Giriş verileri: Sipariş geçmişi, sezonsal faktörler
   * Çıktılar: 30/60/90 günlük tahminler

**6.2 Model Yönetimi**

* Version control
* A/B testing
* Performance monitoring
* Automated retraining

**7. Veri Mimarisi**

**7.1 Veri Kaynakları**

* SAP sistem verileri
* Batch dosya işlemleri
* Gerçek zamanlı akışlar
* Harici veri kaynakları

**7.2 Veri İşleme**

* Stream processing
* Batch processing
* ETL pipeline'ları
* Data validation

**8. Deployment ve DevOps**

**8.1 CI/CD Pipeline**

* GitLab CI
* Automated testing
* Container registry
* Deployment automation

**8.2 Monitoring**

* System health monitoring
* Performance metrics
* Error tracking
* Usage analytics

**9. Güvenlik ve Uyumluluk**

**9.1 Güvenlik Önlemleri**

* JWT authentication
* Role-based access control
* Data encryption
* API security

**9.2 Uyumluluk**

* GDPR uyumluluğu
* Veri saklama politikaları
* Audit logging
* Security compliance

**10. Kalite Güvence**

**10.1 Test Stratejisi**

* Unit testing
* Integration testing
* Performance testing
* Security testing

**10.2 Kalite Metrikleri**

* Code coverage
* Performance benchmarks
* Error rates
* User satisfaction

**11. Operasyonel Prosedürler**

**11.1 Deployment Prosedürleri**

* Release management
* Rollback prosedürleri
* Database migration
* Configuration management

**11.2 Bakım Prosedürleri**

* Backup stratejisi
* Monitoring setup
* Incident response
* Capacity planning

**GELISTIRME ADIMLARI**

**SAP Nexus AI - Development Roadmap**

**Phase 0: Project Setup and Infrastructure (Sprint 0)**

**0.1 Development Environment Setup**

1. Initialize Git repository with proper branching strategy
2. Set up development, staging, and production environments
3. Configure CI/CD pipeline with GitLab CI
4. Establish Docker and Kubernetes infrastructure
5. Configure monitoring stack (Prometheus + Grafana)

**0.2 Project Structure Setup**

1. Initialize Next.js frontend project with TypeScript
2. Set up FastAPI backend project
3. Configure PostgreSQL and TimescaleDB
4. Set up Redis cache
5. Initialize MLflow for model management

**0.3 Base Configuration**

1. Configure authentication system (JWT)
2. Set up role-based access control
3. Implement logging and monitoring
4. Configure error handling and reporting
5. Set up automated testing framework

**Phase 1: Data Processing Pipeline (Sprints 1-3)**

**1.1 Data Ingestion**

1. Implement Excel file readers for:
   * MATERIAL\_LIST\_.xlsx
   * STOCK\_.xlsx
   * DELIVERY\_.xlsx
   * ORDERS\_.xlsx
   * OPEN\_ORDERS\_.xlsx
2. Create data validation schemas
3. Implement ETL pipelines
4. Set up data quality checks

**1.2 Data Storage**

1. Design and implement database schema
2. Create data models and migrations
3. Implement data versioning
4. Set up backup and recovery procedures

**1.3 Data Processing**

1. Implement data transformation pipelines
2. Create data aggregation services
3. Set up real-time processing capabilities
4. Implement data caching strategy

**Phase 2: Core Backend Services (Sprints 4-6)**

**2.1 API Development**

1. Design RESTful API architecture
2. Implement core API endpoints
3. Create API documentation
4. Implement rate limiting and security measures

**2.2 Business Logic**

1. Implement inventory management service
2. Create order processing service
3. Develop delivery tracking system
4. Implement business rules engine

**2.3 Integration Services**

1. Create SAP integration layer
2. Implement event processing system
3. Set up message queues
4. Create notification service

**Phase 3: ML Model Development (Sprints 7-9)**

**3.1 Stock Optimization Model**

1. Implement data preprocessing pipeline
2. Develop baseline model
3. Create model training pipeline
4. Implement model evaluation system
5. Set up automated retraining

**3.2 Demand Forecasting Model**

1. Create feature engineering pipeline
2. Implement Prophet + GradientBoost ensemble
3. Develop model validation system
4. Create prediction API
5. Implement model monitoring

**3.3 Model Operations**

1. Set up model versioning
2. Implement A/B testing framework
3. Create model deployment pipeline
4. Set up performance monitoring

**Phase 4: Frontend Development (Sprints 10-12)**

**4.1 Core Components**

1. Implement authentication and authorization UI
2. Create dashboard layout and navigation
3. Develop base components library
4. Implement state management

**4.2 Data Visualization**

1. Create interactive charts and graphs
2. Implement real-time updates
3. Develop customizable dashboards
4. Create reporting interface

**4.3 User Interface**

1. Implement inventory management interface
2. Create order management system
3. Develop forecasting visualization
4. Create system administration interface

**Phase 5: Integration and Testing (Sprints 13-14)**

**5.1 Integration Testing**

1. Implement end-to-end tests
2. Create integration test suite
3. Set up automated testing pipeline
4. Perform load testing

**5.2 Security Testing**

1. Conduct security audit
2. Implement penetration testing
3. Perform vulnerability assessment
4. Set up security monitoring

**5.3 Performance Optimization**

1. Optimize database queries
2. Implement caching strategy
3. Optimize frontend performance
4. Fine-tune ML model inference

**Phase 6: Deployment and Documentation (Sprint 15)**

**6.1 Deployment**

1. Create deployment documentation
2. Set up production environment
3. Configure monitoring and alerts
4. Implement backup procedures

**6.2 Documentation**

1. Create technical documentation
2. Write user manuals
3. Prepare API documentation
4. Create maintenance guides

**Phase 7: Training and Handover (Sprint 16)**

**7.1 Training**

1. Prepare training materials
2. Conduct technical training
3. Create user training videos
4. Document best practices

**7.2 System Handover**

1. Perform system audit
2. Create handover documentation
3. Set up support procedures
4. Establish maintenance schedule

**Development Guidelines**

**Code Quality Standards**

* Maintain minimum 80% test coverage
* Follow SOLID principles
* Implement proper error handling
* Use typed interfaces
* Document all public APIs

**Performance Requirements**

* Page load time < 3 seconds
* API response time < 500ms
* ML model inference time < 1 second

**Security Standards**

* Implement JWT authentication
* Use role-based access control
* Encrypt sensitive data
* Follow OWASP security guidelines

**Monitoring Requirements**

* System health monitoring
* Performance metrics tracking
* Error logging and alerting
* User activity monitoring

**Daily Development Workflow**

1. Code review process
2. Testing procedures
3. Deployment checklist
4. Documentation updates