

# Multi-Channel E-commerce Management Portal

## Project Overview

### Tech Stack

- **Database:** Neon (PostgreSQL)
- **Version Control:** Git
- **Deployment:** Railway App
- **Channels:** Best Buy, Amazon, Shopify

### Core Requirements

- Modular architecture for isolated sales data processing
- Real-time dashboards for sales, inventory, and product performance
- AI-powered analytics and insights
- High-end, professional interface

---

## System Architecture

### 1. Database Schema (Neon PostgreSQL)

sql

## *-- Core Tables Structure*

### *-- Channels*

```
channels (  
  id, name, api_config, status, created_at, updated_at  
)
```

### *-- Products (Master catalog)*

```
products (  
  id, sku, name, description, brand, category,  
  base_price, cost, weight, dimensions, created_at  
)
```

### *-- Channel-specific product mappings*

```
channel_products (  
  id, product_id, channel_id, channel_sku,  
  channel_price, status, sync_status  
)
```

### *-- Inventory*

```
inventory (  
  id, product_id, channel_id, quantity,  
  reserved_quantity, last_updated  
)
```

### *-- Sales Orders*

```
sales_orders (  
  id, channel_id, channel_order_id, customer_info,  
  order_date, status, total_amount, items, shipping_info  
)
```

### *-- Order Items*

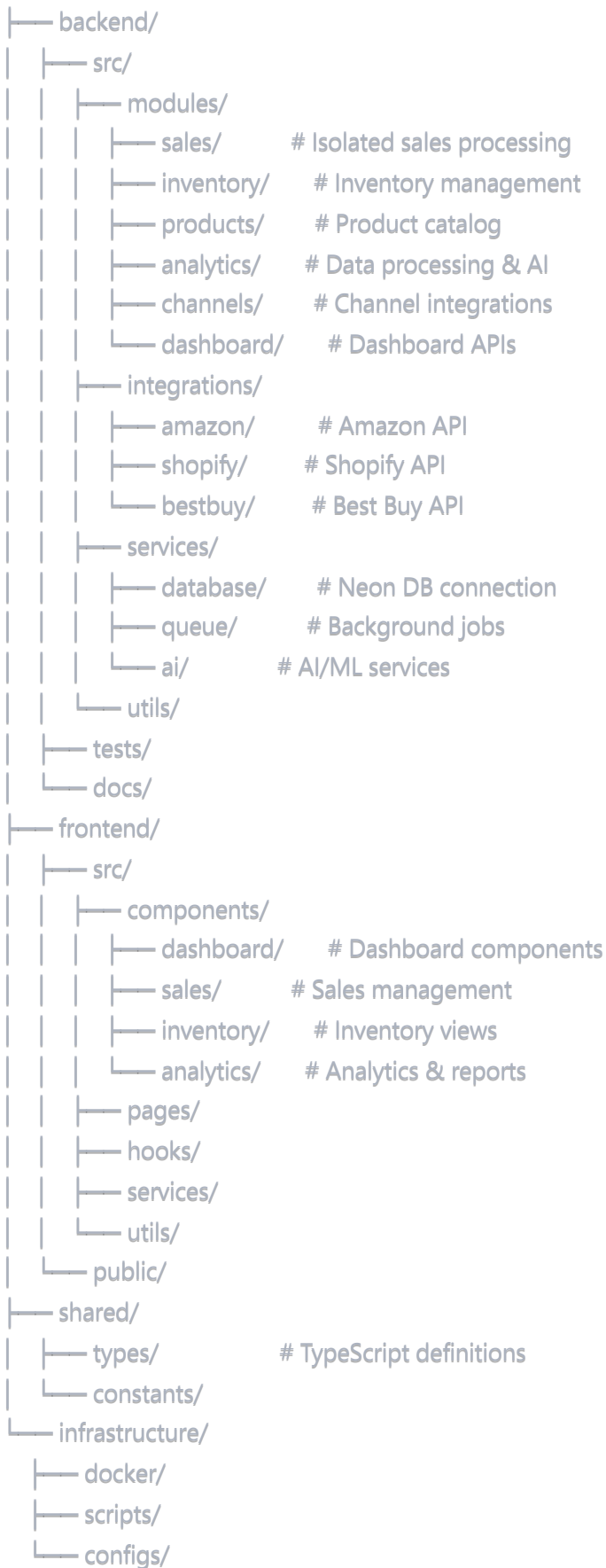
```
order_items (  
  id, order_id, product_id, quantity,  
  unit_price, total_price  
)
```

### *-- Analytics Tables*

```
daily_sales_summary (  
  date, channel_id, product_id, quantity_sold,  
  revenue, profit, avg_order_value  
)
```

```
-- AI Insights
ai_insights (
  id, type, data, confidence_score,
  generated_at, status
)
```

## 2. Application Structure



---

## Project Milestones

## **Phase 1: Foundation & Core Infrastructure (Weeks 1-3)**

### **Deliverables:**

- Database schema design and implementation in Neon
- Basic authentication and user management
- Core API structure with modular architecture
- Git repository setup with branching strategy
- Railway deployment pipeline
- Basic frontend shell with routing

### **Key Tasks:**

- Set up Neon database with optimized schema
- Implement database migrations and seeding
- Create base API controllers and middleware
- Set up TypeScript configuration
- Configure Railway deployment with environment management
- Design and implement authentication system

## **Phase 2: Channel Integrations (Weeks 4-6)**

### **Deliverables:**

- Amazon Seller Central API integration
- Shopify API integration
- Best Buy Partner API integration
- Data synchronization services
- Error handling and logging system

### **Key Tasks:**

- Implement OAuth flows for each platform
- Create data mapping services for product catalogs
- Build order import/sync mechanisms
- Implement inventory synchronization
- Create webhook handlers for real-time updates

- Add comprehensive error tracking

### **Phase 3: Sales Data Processing Module (Weeks 7-9)**

#### **Deliverables:**

- Isolated sales data processing engine
- Order management system
- Sales reporting infrastructure
- Data validation and cleansing

#### **Key Tasks:**

- Build modular sales processing pipeline
- Implement order status tracking
- Create sales aggregation services
- Add data quality checks and validation
- Build sales reporting APIs
- Implement audit trails

### **Phase 4: Dashboard Development (Weeks 10-12)**

#### **Deliverables:**

- Real-time sales dashboard
- Interactive charts and visualizations
- Key performance indicators (KPIs)
- Responsive design for all devices

#### **Key Tasks:**

- Design dashboard UI/UX
- Implement real-time data streaming
- Create interactive charts (sales trends, channel performance)
- Build KPI calculation engine
- Add filtering and date range selection
- Optimize for mobile and tablet views

### **Phase 5: Inventory Management (Weeks 13-15)**

**Deliverables:**

- Multi-channel inventory tracking
- Low stock alerts and automation
- Inventory forecasting
- Reorder point calculations

**Key Tasks:**

- Build inventory tracking across all channels
- Implement automated stock level monitoring
- Create forecasting algorithms
- Add inventory movement history
- Build reorder notifications
- Implement bulk inventory updates

**Phase 6: Product Performance Analytics (Weeks 16-18)****Deliverables:**

- Product performance metrics
- Profitability analysis
- Sales velocity tracking
- Channel-specific performance comparison

**Key Tasks:**

- Calculate product ROI and margins
- Track sales velocity and trends
- Compare performance across channels
- Identify top and underperforming products
- Build performance prediction models
- Create automated performance reports

**Phase 7: AI Analytics Engine (Weeks 19-22)****Deliverables:**

- Predictive analytics for sales forecasting

- Automated insights and recommendations
- Market trend analysis
- Pricing optimization suggestions

**Key Tasks:**

- Implement machine learning models for sales prediction
- Build recommendation engine
- Create automated insight generation
- Add natural language report generation
- Implement anomaly detection
- Build AI-powered pricing recommendations

**Phase 8: Advanced Features & Optimization (Weeks 23-25)****Deliverables:**

- Performance optimization
- Advanced reporting suite
- API rate limiting and caching
- Comprehensive testing suite

**Key Tasks:**

- Optimize database queries and indexing
- Implement Redis caching layer
- Add comprehensive unit and integration tests
- Build automated backup systems
- Create performance monitoring
- Implement advanced security features

**Phase 9: Testing & Deployment (Weeks 26-28)****Deliverables:**

- Full system testing
- Production deployment
- User training materials



- Documentation

### Key Tasks:

- Conduct end-to-end testing
  - Performance and load testing
  - Security penetration testing
  - Deploy to production environment
  - Create user documentation
  - Conduct user acceptance testing
- 

## Technical Specifications

### Frontend Technology Stack

- **Framework:** React 18 with TypeScript
- **UI Library:** Material-UI or Chakra UI for professional appearance
- **Charts:** Recharts or Chart.js for data visualization
- **State Management:** Zustand or Redux Toolkit
- **Real-time:** WebSocket connections for live updates

### Backend Technology Stack

- **Runtime:** Node.js with TypeScript
- **Framework:** Express.js or Fastify
- **ORM:** Prisma with Neon PostgreSQL
- **Authentication:** JWT with refresh tokens
- **Queue:** Bull Queue with Redis
- **Monitoring:** Winston logging with Sentry

### DevOps & Infrastructure

- **CI/CD:** GitHub Actions with Railway deployment
  - **Monitoring:** Application performance monitoring
  - **Backup:** Automated database backups
  - **Scaling:** Horizontal scaling capabilities on Railway
-

# Key Features by Module

## Sales Dashboard

- Real-time sales metrics across all channels
- Revenue trends and comparisons
- Order volume tracking
- Customer acquisition metrics
- Geographic sales distribution

## Inventory Management

- Real-time stock levels across channels
- Automated reorder notifications
- Inventory movement tracking
- ABC analysis for inventory prioritization
- Stockout prevention algorithms

## Product Performance

- Individual product profitability
- Sales velocity and trend analysis
- Channel-specific performance metrics
- Competitor price monitoring
- Product lifecycle tracking

## AI Analytics

- Sales forecasting with 95% accuracy targets
- Automated anomaly detection
- Personalized business insights
- Predictive inventory management
- Dynamic pricing recommendations

---

## Success Metrics

- **System Performance:** <2s page load times, 99.9% uptime

- **Data Accuracy:** Real-time sync with <5min delay
- **User Experience:** Intuitive interface with <3 clicks to key data
- **Business Impact:** 20% improvement in decision-making speed

This comprehensive plan provides a solid foundation for your high-end business management portal. Each phase builds upon the previous one while maintaining the modular architecture you requested for sales data processing.