# Enhanced Prompt Ops Platform - Complete Development Specification

## Project Overview

The Enhanced Prompt Ops Platform is an enterprise-grade, white-label SaaS solution for visual prompt engineering and AI workflow orchestration. This platform combines the power of React Flow for visual design, Supabase for scalable data management, and n8n for backend workflow automation to create a comprehensive prompt operations platform.

**Market Opportunity**: $3.48B prompt engineering market by 2029 with 32.4% CAGR
**Target Users**: Enterprise AI teams, digital agencies, SMB developers
**Core Value Proposition**: Visual-first prompt management with enterprise governance and white-label capabilities

## Core Objectives

1. **Democratize Prompt Engineering**: Enable non-technical users to create sophisticated AI workflows through visual drag-and-drop interfaces
2. **Enterprise-Grade Governance**: Provide compliance monitoring, security controls, and cost optimization for enterprise AI deployments
3. **Multi-LLM Integration**: Support seamless switching between OpenAI, Anthropic, Google, and open-source models
4. **White-Label Revenue Model**: Enable agencies and consultants to resell the platform under their own brand
5. **Real-Time Collaboration**: Support team-based prompt development with live editing and version control

## Technical Architecture

### Frontend Stack

- **Framework**: Next.js 14 with App Router
- **UI Library**: React 18 + Tailwind CSS 3.4+
- **Visual Editor**: React Flow 12.7+ for drag-and-drop workflow design
- **State Management**: Zustand for client state, React Query for server state
- **Real-Time Updates**: Supabase Realtime for live collaboration
- **Icons**: Lucide React for consistent iconography

### Backend Infrastructure

- **Database**: Supabase (PostgreSQL) with Row Level Security
- **Authentication**: Supabase Auth with SSO support
- **Workflow Engine**: n8n for AI workflow orchestration
- **API Layer**: Next.js API routes with tRPC for type safety
- **File Storage**: Supabase Storage for assets and exports

### AI Integration Layer

- **Multi-LLM Router**: Custom service supporting OpenAI, Anthropic, Google, and Hugging Face
- **Prompt Execution**: Distributed execution with failover support
- **Cost Tracking**: Real-time token usage monitoring and optimization
- **Security Scanning**: Automated sensitive data detection in prompts

## User Interface Design Principles

### Non-Technical User Focus

- **Intuitive Drag-and-Drop**: React Flow-based canvas with clear visual connections
- **Smart Defaults**: Pre-configured templates for common use cases
- **Progressive Disclosure**: Show advanced features only when needed
- **Contextual Help**: In-line tooltips and guided onboarding flows

### Visual Design Language

- **Clean Minimalism**: Uncluttered interface with focus on content
- **Consistent Spacing**: 8px grid system for visual harmony
- **Accessibility First**: WCAG 2.1 AA compliance with keyboard navigation
- **Responsive Design**: Works seamlessly on desktop, tablet, and mobile

## Core Features Specification

### 1. Visual Prompt Chain Designer

### Primary Interface Components

```
Left Sidebar (Node Palette):
├── Basic Nodes
│   ├── Prompt Node (AI interaction)
│   ├── Input Node (user data entry)
│   ├── Output Node (final result)
```

```
│   └── Router Node (conditional logic)
├── Advanced Nodes
│   ├── Validator Node (quality checking)
│   ├── Transformer Node (data processing)
│   ├── Integration Node (external APIs)
│   └── Cache Node (response caching)
└── Template Library
    ├── Customer Support
    ├── Content Generation
    ├── Data Analysis
    └── Document Processing

Center Canvas:
├── Infinite Canvas (React Flow)
├── Smart Grid Snapping
├── Multi-Select Operations
├── Zoom/Pan Controls
└── Connection Validation

Right Panel (Properties):
├── Node Configuration
├── LLM Model Selection
├── Parameter Tuning
├── Test Console
└── Version History
```

### Key Functionality

- **Drag-and-Drop Node Creation**: Simple palette-to-canvas workflow
- **Visual Connection System**: Colored connections showing data flow
- **Real-Time Validation**: Immediate feedback on configuration errors
- **Template Import/Export**: Shareable workflow configurations
- **A/B Testing Framework**: Split-test different prompt variations

### Node Types Specification

#### Prompt Node

```
interface PromptNode {
  id: string;
  type: 'prompt';
  data: {
    title: string;
    prompt: string;
    model: 'gpt-4' | 'claude-3-sonnet' | 'gemini-pro' | 'llama-3.1';
    temperature: number;
    maxTokens: number;
    systemPrompt?: string;
    variables: VariableMapping[];
  };
```

```
  position: { x: number; y: number };
}
```

**Router Node**

```
interface RouterNode {
  id: string;
  type: 'router';
  data: {
    title: string;
    conditions: Condition[];
    defaultPath: string;
  };
}
```

## 2. Enterprise Governance Layer

### Compliance Monitoring

- **Sensitive Data Detection**: Real-time scanning for PII, financial data, healthcare information
- **Regulatory Compliance**: GDPR, HIPAA, SOX compliance checks
- **Content Filtering**: Inappropriate content detection and blocking
- **Usage Audit Logs**: Comprehensive tracking of all prompt executions

### Role-Based Access Control

```
interface UserRole {
  id: string;
  name: string;
  permissions: {
    createPrompts: boolean;
    editPrompts: boolean;
    deployPrompts: boolean;
    viewAnalytics: boolean;
    manageUsers: boolean;
    accessAdmin: boolean;
  };
  promptLibraryAccess: string[]; // Library IDs
  costLimits: {
    monthly: number;
    perExecution: number;
  };
}
```

### Cost Optimization Engine

- **Real-Time Cost Tracking**: Token usage monitoring across all models
- **Budget Alerts**: Automated notifications for spending thresholds
- **Model Recommendations**: AI-powered suggestions for cost-effective model selection
- **Usage Analytics**: Detailed breakdowns of costs by user, project, and time period

## 3. Advanced Analytics Dashboard

### Real-Time Metrics

- **Execution Performance**: Response times, success rates, error frequencies
- **Cost Analysis**: Token usage, model costs, optimization opportunities
- **Quality Scoring**: LLM-as-a-judge evaluation frameworks
- **User Engagement**: Session duration, feature usage, collaboration patterns

### Dashboard Components

```
Top Navigation:
├── Time Range Selector
├── Tenant/Project Filter
├── Export Controls
└── Alert Center

Main Grid Layout:
├── KPI Cards (4-6 key metrics)
├── Performance Charts (time series)
├── Cost Breakdown (pie/bar charts)
├── Usage Heatmaps
├── Error Log Summary
└── Optimization Recommendations

Interactive Elements:
├── Drill-down Capabilities
├── Filter by User/Project
├── Custom Date Ranges
└── Comparative Analysis
```

### Analytics Features

- **Anomaly Detection**: Automatic identification of unusual patterns
- **Predictive Analytics**: Cost forecasting and usage predictions
- **Custom Metrics**: User-defined KPIs and tracking
- **Automated Reports**: Scheduled email/Slack notifications

## 4. White-Label Business Model

### Multi-Tenant Architecture

```
interface Tenant {
  id: string;
  name: string;
  domain: string; // custom.clientdomain.com
  branding: {
    logo: string;
    primaryColor: string;
    secondaryColor: string;
    fontFamily: string;
    customCSS?: string;
  };
  features: FeatureSet;
  billing: BillingConfiguration;
  settings: TenantSettings;
}
```

### Customization Options

- **Visual Branding**: Logo, colors, fonts, custom CSS
- **Feature Sets**: Configurable feature availability per tenant
- **Custom Domains**: Full white-label domain support
- **API Access**: Tenant-specific API keys and documentation
- **Billing Integration**: Stripe Connect for revenue sharing

### Partner Portal

- **Tenant Management**: Create and configure client instances
- **Revenue Dashboard**: Track commissions and usage metrics
- **Support Tools**: Client support ticket management
- **Marketing Assets**: White-label marketing materials and documentation

## Multi-LLM Integration System

### LLM Router Architecture

```
interface LLMProvider {
  id: string;
  name: string;
  apiEndpoint: string;
  authMethod: 'apiKey' | 'oauth' | 'custom';
  models: LLMModel[];
  pricing: PricingModel;
  capabilities: Capability[];
```

```
}

interface LLMModel {
  id: string;
  name: string;
  contextWindow: number;
  inputCost: number; // per 1K tokens
  outputCost: number; // per 1K tokens
  capabilities: ('text' | 'multimodal' | 'function-calling')[];
}
```

### Supported Providers

1. **OpenAI**: GPT-4, GPT-4 Turbo, GPT-3.5 Turbo
2. **Anthropic**: Claude 3 Opus, Sonnet, Haiku
3. **Google**: Gemini Pro, Gemini Ultra
4. **Open Source**: Llama 3.1, Mistral 7B/22B, CodeLlama

### Smart Model Selection

- **Cost Optimization**: Automatic model selection based on cost/performance requirements
- **Capability Matching**: Route requests to models with required capabilities
- **Fallback Strategy**: Automatic failover to alternative models
- **Load Balancing**: Distribute requests across multiple providers

## Database Schema (Supabase)

### Core Tables

```
-- Organizations (Tenants)
CREATE TABLE organizations (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  name TEXT NOT NULL,
  domain TEXT UNIQUE,
  plan TEXT NOT NULL DEFAULT 'free',
  branding JSONB DEFAULT '{}',
  settings JSONB DEFAULT '{}',
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

-- Users
CREATE TABLE users (
  id UUID PRIMARY KEY REFERENCES auth.users(id),
  org_id UUID REFERENCES organizations(id),
  email TEXT NOT NULL,
  role TEXT NOT NULL DEFAULT 'member',
  permissions JSONB DEFAULT '{}',
  created_at TIMESTAMP DEFAULT NOW()
```

```
);

-- Prompt Chains
CREATE TABLE prompt_chains (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  org_id UUID REFERENCES organizations(id),
  name TEXT NOT NULL,
  description TEXT,
  version INTEGER DEFAULT 1,
  status TEXT DEFAULT 'draft',
  config JSONB NOT NULL,
  tags TEXT[],
  created_by UUID REFERENCES users(id),
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

-- Executions
CREATE TABLE prompt_executions (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  chain_id UUID REFERENCES prompt_chains(id),
  org_id UUID REFERENCES organizations(id),
  input_data JSONB,
  output_data JSONB,
  metadata JSONB,
  cost_data JSONB,
  started_at TIMESTAMP DEFAULT NOW(),
  completed_at TIMESTAMP,
  status TEXT NOT NULL
);

-- Analytics (Hourly Aggregation)
CREATE TABLE analytics_hourly (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  org_id UUID REFERENCES organizations(id),
  chain_id UUID REFERENCES prompt_chains(id),
  hour TIMESTAMP NOT NULL,
  executions INTEGER DEFAULT 0,
  total_cost DECIMAL DEFAULT 0,
  avg_latency_ms INTEGER,
  success_rate DECIMAL,
  unique_users INTEGER
);
```

### Row Level Security Policies

```
-- Users can only access their organization's data
CREATE POLICY "Users can view own org data" ON prompt_chains
  FOR SELECT USING (org_id = (SELECT org_id FROM users WHERE id = auth.uid()));

-- Role-based access for admin functions
CREATE POLICY "Admins can manage all" ON organizations
  FOR ALL USING (
    EXISTS (
      SELECT 1 FROM users
```

```
      WHERE id = auth.uid()
      AND role = 'admin'
    )
  );
```

## Implementation Phases

### Phase 1: Core Infrastructure (Weeks 1-4)

**Goals**: Establish foundation and basic functionality

- [ ] Supabase project setup with authentication
- [ ] Next.js application scaffolding
- [ ] Basic React Flow integration
- [ ] Simple prompt execution pipeline
- [ ] User management and organization setup

**Deliverables**:

- Working authentication system
- Basic visual prompt designer
- Simple prompt execution
- User/organization management

### Phase 2: Visual Designer & Multi-LLM (Weeks 5-8)

**Goals**: Complete visual editor and AI integration

- [ ] Full React Flow node system
- [ ] Node property panels and configuration
- [ ] Multi-LLM router implementation
- [ ] Real-time collaboration features
- [ ] Template library system

**Deliverables**:

- Complete visual prompt designer
- Multi-LLM integration
- Real-time collaboration
- Template system

### Phase 3: Analytics & Governance (Weeks 9-12)

**Goals**: Enterprise features and analytics

- [ ] Analytics dashboard implementation
- [ ] Cost tracking and optimization
- [ ] Compliance monitoring system
- [ ] Role-based access controls
- [ ] Audit logging system

**Deliverables**:

- Enterprise governance layer
- Advanced analytics dashboard
- Compliance monitoring
- Cost optimization features

### Phase 4: White-Label & Scale (Weeks 13-16)

**Goals**: White-label capabilities and platform scaling

- [ ] Multi-tenant architecture implementation
- [ ] Custom branding system
- [ ] Partner portal development
- [ ] Performance optimization
- [ ] Documentation and onboarding

**Deliverables**:

- White-label platform
- Partner management portal
- Performance optimizations
- Complete documentation

## Success Metrics

### Technical Metrics

- **Response Time**: <500ms for prompt execution initiation
- **Uptime**: 99.9% availability
- **Scalability**: Support 1000+ concurrent users
- **Security**: SOC 2 Type II compliance

### Business Metrics

- **User Adoption**: 80% DAU/MAU ratio
- **Revenue Growth**: 30% MoM growth in ARR
- **Cost Efficiency**: 40% reduction in customer AI spend
- **Customer Satisfaction**: >4.5/5 NPS score

### Feature Metrics

- **Time to First Value**: <5 minutes for new users
- **Template Usage**: 70% of prompts use templates
- **Collaboration**: Average 3.2 collaborators per project
- **Cost Optimization**: 25% average cost reduction

## Security & Compliance

### Data Protection

- **Encryption**: AES-256 encryption at rest and in transit
- **Access Controls**: Zero-trust architecture with least privilege
- **Audit Logging**: Comprehensive activity tracking
- **Data Retention**: Configurable retention policies

### Compliance Standards

- **GDPR**: Full EU data protection compliance
- **SOC 2 Type II**: Security and availability controls
- **CCPA**: California privacy law compliance
- **HIPAA**: Healthcare data protection (optional)

### Security Features

- **Prompt Scanning**: Automated PII and sensitive data detection
- **Rate Limiting**: API abuse prevention
- **Vulnerability Scanning**: Regular security assessments
- **Incident Response**: 24/7 security monitoring

## Revenue Model

### Pricing Tiers

```
Free Tier:
- 3 prompt chains
- 1,000 executions/month
- Basic analytics
- Community support

Professional ($99/month):
- 50 prompt chains
- 50,000 executions/month
- Advanced analytics
- Email support
- Version control

Enterprise ($499/month):
- Unlimited prompt chains
- 500,000 executions/month
- White-label options
- Priority support
- SSO integration

Custom Enterprise:
- Unlimited everything
- On-premise deployment
- Custom integrations
- Dedicated support
- SLA guarantees
```

### White-Label Revenue Sharing

- **Partner Revenue**: 70/30 split (Partner/Platform)
- **Setup Fee**: $2,000 one-time white-label setup
- **Monthly Platform Fee**: $200/month per white-label instance
- **Overage Charges**: Standard execution pricing

## Development Guidelines

### Code Quality Standards

- **TypeScript**: 100% TypeScript coverage
- **Testing**: 80%+ test coverage with Jest/Playwright
- **Linting**: ESLint + Prettier configuration
- **Documentation**: JSDoc for all public APIs

### Development Workflow

- **Git Flow**: Feature branches with PR reviews
- **CI/CD**: GitHub Actions with automated testing
- **Deployment**: Vercel for frontend, Railway for n8n
- **Monitoring**: Sentry for error tracking

### Performance Requirements

- **Core Web Vitals**: LCP <2.5s, FID <100ms, CLS <0.1
- **Bundle Size**: <500KB initial JavaScript bundle
- **API Response**: <200ms for standard operations
- **Database Queries**: <50ms average query time

This specification provides a comprehensive blueprint for building the Enhanced Prompt Ops Platform using your preferred tech stack of Cursor.AI development, Node.js/React frontend, Supabase database, and n8n workflow automation. The platform addresses the market opportunity for prompt management while providing enterprise-grade features and white-label capabilities for maximum revenue potential.