

Rylie SEO Hub - Implementation Plan & Roadmap

Project Overview

Objective: Transform Rylie SEO Hub into a fully functional white-labeled SEO platform that serves as a conversational middlepiece between dealerships and SEO WORKS.

Timeline: 23 days (June 23 - June 15, 2025) **Team Size:** 2-3 experienced developers + 1 project manager **Budget Estimate:** 45,000—65,000 (based on senior developer rates)

Development Phases

PHASE 1: Foundation & Onboarding (Days 1-8)

Objective: Establish agency onboarding and installation requirement systems

Day 1-2: Project Setup & Database Design

Tasks: - Set up development environment and team access - Design database schema for new models - Create migration scripts for new tables - Set up feature flags for gradual rollout

Deliverables: - Updated Prisma schema with new models - Database migrations - Development environment setup - Feature flag configuration

Database Models to Add:

```

model DealershipOnboarding {
  id: String @id @default(cuid())
  agencyId: String
  dealershipName: String
  contactEmail: String
  contactPhone: String
  website: String?
  status: String // pending, in_progress, completed
  installationRequirements: Json?
  createdAt: DateTime @default(now())
  updatedAt: DateTime @updatedAt
}

model OnboardingForm {
  id: String @id @default(cuid())
  agencyId: String
  name: String
  fields: Json // Dynamic form fields
  isActive: Boolean @default(true)
  createdAt: DateTime @default(now())
}

model InstallationRequirement {
  id: String @id @default(cuid())
  onboardingId: String
  requirementType: String
  description: String
  priority: String // high, medium, low
  status: String // pending, completed
  assignedTo: String?
  createdAt: DateTime @default(now())
}

```

Day 3-5: Agency Dashboard Development

Tasks: - Create agency admin dashboard - Build dealership management interface - Implement onboarding form builder - Add dealership listing and status tracking

Deliverables: - Agency dashboard UI - Dealership management system - Form builder component - Onboarding workflow

Key Components: - `/src/app/admin/dealerships` - Dealership management - `/src/app/admin/onboarding` - Onboarding forms - `/src/components/admin/DealershipList.tsx` - `/src/components/admin/OnboardingForm.tsx` - `/src/components/admin/FormBuilder.tsx`

Day 6-8: Installation Requirements System

Tasks: - Build installation requirement collection - Create requirement templates - Implement data validation and storage - Add SEO WORKS API integration for requirements

Deliverables: - Installation requirement system - Template management - API integration for requirement delivery - Data validation framework

API Endpoints: - `POST /api/admin/dealerships` - Create new dealership - `GET /api/admin/dealerships` - List dealerships - `POST /api/admin/onboarding-forms` - Create onboarding form - `POST /api/admin/installation-requirements` - Submit requirements

PHASE 2: Chat Integration & AI Enhancement (Days 9-16)

Objective: Implement seamless chat-to-order workflow with AI intent recognition

Day 9-11: AI Intent Recognition

Tasks: - Enhance AI prompting for service request detection - Implement natural language processing for order details - Create intent classification system - Add parameter extraction from conversations

Deliverables: - Enhanced AI service with intent recognition - Order parameter extraction - Intent classification system - Conversation context management

Technical Implementation:

```
// Enhanced AI service
interface IntentRecognition {
  intent: 'order_request' | 'question' | 'status_check'
  confidence: number
  parameters: {
    taskType?: string
    description?: string
    urgency?: string
    estimatedHours?: number
  }
}

// Updated chat API
const aiResponse = await aiService.generateResponseWithIntent(
  chatMessages,
  model,
  tenantContext
)
```

Day 12-14: Chat-to-Order Integration

Tasks: - Integrate order creation within chat interface - Build order confirmation and tracking in chat - Implement real-time order status updates - Add order management UI components in chat

Deliverables: - Seamless chat-to-order workflow - Order tracking within conversations - Real-time status updates - Enhanced chat UI components

Key Components:

-	/src/components/chat/OrderCreation.tsx	-
	/src/components/chat/OrderStatus.tsx	-
	/src/components/chat/OrderConfirmation.tsx	-
	/src/app/api/chat/route.ts	Enhanced

Day 15-16: Workflow Testing & Refinement

Tasks: - End-to-end workflow testing - User experience optimization - Error handling and edge cases - Performance optimization

Deliverables: - Tested chat-to-order workflow - Optimized user experience - Comprehensive error handling - Performance benchmarks

PHASE 3: Automation & Reporting (Days 17-23)

Objective: Implement automatic reporting system and final optimizations

Day 17-19: Report Generation System

Tasks: - Build GA4 data processing engine - Create report generation templates - Implement automated report scheduling - Add report delivery system

Deliverables: - GA4 data processing system - Report generation engine - Automated scheduling system - Multi-channel delivery (email, chat, dashboard)

Technical Components:

```
// Report generation service
interface ReportGenerator {
  generateSEOREport(agencyId: string, dateRange: DateRange): Promise<Report>
  scheduleReport(agencyId: string, schedule: ReportSchedule): Promise<void>
  deliverReport(report: Report, deliveryMethod: DeliveryMethod): Promise<void>
}

// Report templates
interface ReportTemplate {
  id: string
  name: string
  sections: ReportSection[]
  branding: AgencyBranding
}
```

Day 20-21: Report Delivery Integration

Tasks: - Integrate report delivery with chat interface - Add email notification system - Implement report history and archiving - Create report customization options

Deliverables: - Chat-integrated report delivery - Email notification system - Report history management - Customization interface

Day 22-23: Final Testing & Deployment

Tasks: - Comprehensive system testing - Performance optimization - Security audit - Production deployment - User training materials

Deliverables: - Fully tested system - Performance optimizations - Security validation - Production deployment - User documentation

Technical Architecture Updates

New API Endpoints

```
/api/admin/dealerships
├─ GET / - List dealerships
├─ POST / - Create dealership
├─ PUT /:id - Update dealership
└─ DELETE /:id - Remove dealership

/api/admin/onboarding
├─ GET /forms - List onboarding forms
├─ POST /forms - Create form
├─ GET /forms/:id - Get form
└─ POST /submit - Submit onboarding

/api/admin/installation
├─ GET /requirements - List requirements
├─ POST /requirements - Create requirement
└─ PUT /requirements/:id - Update requirement

/api/chat/enhanced
├─ POST /intent - Analyze intent
├─ POST /order - Create order from chat
└─ GET /orders - Get chat orders

/api/reports
├─ GET / - List reports
├─ POST /generate - Generate report
├─ GET /:id - Get specific report
└─ POST /schedule - Schedule report
```

Database Schema Updates

```
-- New tables to be added
CREATE TABLE dealership_onboarding (
  id VARCHAR PRIMARY KEY,
  agency_id VARCHAR NOT NULL,
  dealership_name VARCHAR NOT NULL,
  contact_email VARCHAR NOT NULL,
  contact_phone VARCHAR,
  website VARCHAR,
  status VARCHAR DEFAULT 'pending',
  installation_requirements JSONB,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

CREATE TABLE onboarding_forms (
  id VARCHAR PRIMARY KEY,
  agency_id VARCHAR NOT NULL,
  name VARCHAR NOT NULL,
  fields JSONB NOT NULL,
  is_active BOOLEAN DEFAULT true,
  created_at TIMESTAMP DEFAULT NOW()
);

CREATE TABLE installation_requirements (
  id VARCHAR PRIMARY KEY,
  onboarding_id VARCHAR NOT NULL,
  requirement_type VARCHAR NOT NULL,
  description TEXT NOT NULL,
  priority VARCHAR DEFAULT 'medium',
  status VARCHAR DEFAULT 'pending',
  assigned_to VARCHAR,
  created_at TIMESTAMP DEFAULT NOW()
);

CREATE TABLE report_schedules (
  id VARCHAR PRIMARY KEY,
  agency_id VARCHAR NOT NULL,
  report_type VARCHAR NOT NULL,
  schedule_cron VARCHAR NOT NULL,
  delivery_method VARCHAR NOT NULL,
  is_active BOOLEAN DEFAULT true,
  created_at TIMESTAMP DEFAULT NOW()
);
```

Resource Requirements

Development Team

Required Roles: - **Senior Full-Stack Developer** (Lead) - \$150-200/day - **Frontend Developer** (React/Next.js specialist) - \$120-150/day

- **Backend Developer** (API/Database specialist) - \$120-150/day - **Project Manager** (Part-time) - \$100-120/day

Total Daily Cost: \$490-620 **Total Project Cost:** \$11,270-14,260 (23 days)

Infrastructure Requirements

Additional Services: - **Report Generation Service** - \$50-100/month - **Email Delivery Service** (Mailgun/SendGrid) - \$20-50/month - **File Storage** (Cloudinary/AWS S3) - \$20-40/month - **Monitoring & Analytics** (Enhanced) - \$30-60/month

Third-Party Integrations

Required APIs: - **Google Analytics 4 API** - Free (with rate limits) - **OpenRouter API** - \$0.002-0.02 per 1K tokens - **Email Service API** - \$0.50-1.00 per 1K emails - **PDF Generation Service** - \$0.01-0.05 per document

Risk Mitigation Strategies

HIGH RISK: Timeline Pressure

Mitigation: - Parallel development streams - Daily standup meetings - Continuous integration/deployment - Feature flag rollouts for testing - Contingency plan for automatic reporting

MEDIUM RISK: AI Integration Complexity

Mitigation: - Start with simple intent recognition - Use existing OpenRouter models - Implement fallback mechanisms - Gradual complexity increase - User feedback integration

MEDIUM RISK: GA4 API Limitations

Mitigation: - Implement caching strategies - Rate limit management - Alternative data sources - Graceful degradation - Manual report fallback

LOW RISK: Multi-tenant Data Isolation

Mitigation: - Comprehensive testing - Database constraint validation - Audit logging - Security reviews - Tenant-specific testing

Success Metrics

Technical Metrics

- **System Uptime:** >99.5%
- **API Response Time:** <500ms average
- **Chat Response Time:** <2 seconds
- **Report Generation Time:** <30 seconds
- **Error Rate:** <1%

Business Metrics

- **Onboarding Completion Rate:** >80%
- **Chat-to-Order Conversion:** >60%
- **User Satisfaction:** >4.5/5
- **Agency Adoption:** 100% of pilot agencies
- **Dealership Engagement:** >70% monthly active

User Experience Metrics

- **Time to First Order:** <5 minutes
 - **Onboarding Completion Time:** <15 minutes
 - **Report Delivery Success:** >95%
 - **Support Ticket Reduction:** >50%
 - **User Retention:** >85% monthly
-

Deployment Strategy

Staging Environment

Week 1: Development environment setup **Week 2:** Feature testing and integration
Week 3: User acceptance testing

Production Rollout

Phase 1: Agency admin features (Day 15) **Phase 2:** Chat integration (Day 18) **Phase 3:** Automatic reporting (Day 21) **Phase 4:** Full system launch (Day 23)

Rollback Plan

- Feature flags for instant rollback
 - Database backup before each deployment
 - Previous version containers maintained
 - Monitoring alerts for immediate detection
 - Emergency contact procedures
-

Post-Launch Support

Immediate Support (Days 24-30)

- 24/7 monitoring and support
- Daily check-ins with agencies
- Bug fixes and hotfixes
- Performance optimization
- User training sessions

Ongoing Maintenance

- Weekly system health reports

- Monthly feature updates
- Quarterly security audits
- Bi-annual architecture reviews
- Continuous user feedback integration

The implementation plan provides a structured approach to delivering the required functionality within the aggressive timeline while maintaining quality and minimizing risks.