

### ♦ Project Name: Ghostart

A scalable **AI-powered LinkedIn content platform** designed for **thought leadership, employee advocacy, and brand content automation**.

### ♦ Tech Stack & Hosting

- ✓ Frontend: Next.js (App Router) on Vercel
- ✓ Styling: TailwindCSS
- ✓ Backend: Supabase (PostgreSQL, RLS, Edge Functions)
- ✓ Caching: Redis (User Data, AI Content, Engagement Metrics)
- ✓ AI Services: Python backend on Render (OpenAI-powered content)
- ✓ Queue System: Task scheduling for AI content, analytics, email triggers
- ✓ External APIs: LinkedIn API (OAuth, posting, analytics), Stripe (payments)
- ✓ Security & Monitoring: Sentry (error tracking), APM (response times, API health)

### ♦ Claude's Architecture Integrated

#### 1 Frontend (Next.js on Vercel)

- **App Router model** (modern approach)
- **Public & Protected Routes**
- **Edge Caching** for static & dynamic content
- ✓ **Action Items for Cursor:**
  - Implement **authentication guards** on protected routes.
  - Set up **Next.js Edge Middleware** for caching.

#### 2 Caching Layer (Redis)

- **User Data:** Profile info, preferences, AI exercise results
- **Content Data:** Cached AI-generated posts, templates, training materials
- **Analytics Data:** Performance tracking, LinkedIn engagement stats
- ✓ **Action Items for Cursor:**

- Set up **Redis integration** for fast data retrieval.
- Implement **cache invalidation rules** (e.g., when a user updates their profile).

### 3 Queue System

Handles **background processing** for:

- **AI-generated content** (async requests to OpenAI)
- **Scheduled posts** (posting via LinkedIn API)
- **Performance analytics** (aggregating engagement data)

#### ✓ Action Items for Cursor:

- Implement **Task Scheduling & Worker Processes**.
- Connect queue system to **AI post generation & scheduled content**.

### 4 Backend (Supabase)

- **PostgreSQL DB**
- **Row Level Security (RLS)** for user-specific data protection
- **Edge Functions** for real-time AI content updates
- **Authentication system (OAuth, JWT, Role-based Access)**

#### ✓ Action Items for Cursor:

- Ensure **user data is securely stored with RLS**.
- Implement **Supabase Edge Functions for AI processing triggers**.

### 5 AI Service (Python Backend on Render)

- Runs **OpenAI-powered content generation**
- Uses **retry mechanisms & fallback strategies**
- Automates **personalized LinkedIn post drafting**

#### ✓ Action Items for Cursor:

- Implement **AI content API endpoints**.
- Set up **error handling & retries** for OpenAI failures.