

PrimeStride Atlas

AI-Powered Knowledge Management System

Product Documentation

Version 1.0 | February 2026

primestrideatlas.com

Confidential — PrimeStride AI

Table of Contents

1. Executive Summary

PrimeStride Atlas is an AI-powered knowledge management system designed for teams and organizations. It enables teams to upload, organize, and discover insights from their collective knowledge base through intelligent document management, semantic search, AI-powered chat assistance, and interactive knowledge graph visualization.

The platform transforms how organizations manage institutional knowledge by combining traditional document management with cutting-edge AI capabilities, making it easy for every team member to find the information they need, when they need it.

Key Capabilities

- **Document Management:** Upload, create, edit, and organize documents with version tracking, tagging, and categorization.
- **AI Chat Assistant:** Ask natural language questions and receive answers sourced directly from your documents using Retrieval-Augmented Generation (RAG).
- **Semantic Search:** Find documents by meaning, not just keywords, powered by OpenAI embeddings and cosine similarity.
- **Knowledge Graph:** Visualize document relationships and topic clusters with an interactive, AI-labeled graph.
- **Multi-Tenant Teams:** Role-based access control with organization management, team invitations, and member roles.
- **Feedback & Analytics:** Track document usefulness with a built-in feedback system and learning dashboard.

2. Technology Stack

PrimeStride Atlas is built on a modern, production-grade technology stack optimized for performance, scalability, and developer experience.

Frontend	Next.js 14 + React 18	Server-side rendering, app router, React components
Language	TypeScript	Type safety across the entire codebase
Authentication	Clerk	Multi-tenant auth with Google OAuth, session management
Database	Supabase (PostgreSQL)	Relational data, real-time subscriptions, storage
File Storage	Supabase Storage	Document file uploads (PDF, DOCX, etc.)
AI / LLM	OpenAI GPT-4o-mini	Chat responses, document summaries, cluster naming
Embeddings	text-embedding-3-small	1536-dimensional vectors for semantic search
Graph Visualization	React Flow	Interactive knowledge graph rendering
Email	Resend	Transactional emails for team invitations
Deployment	Vercel	Automatic deployments from GitHub, edge network
Webhooks	Svix	Clerk webhook verification for user sync
Document Parsing	Mammoth + unPDF	Extract text from DOCX and PDF uploads

Codebase Overview

The project consists of approximately **18,300+ lines of TypeScript/React code** across 47 API routes, 24 pages, 9 reusable components, and 8 library modules.

3. System Architecture

Multi-Tenancy Model

PrimeStride Atlas follows the Slack/Linear model of one workspace per organization. Each organization is an isolated tenant with its own documents, members, teams, and AI-generated data (embeddings, clusters). Users can belong to multiple organizations and switch between them.

Organization Hierarchy

- **Organization:** The top-level tenant. Each org has its own document library, knowledge graph, and member list.
- **Members:** Users belong to organizations with roles: Owner, Admin, or Member.
- **Teams:** Sub-groups within an organization for document access control. Documents can be org-wide or team-scoped.

Authentication Flow

Authentication is handled by Clerk with the following flow:

- User signs up or signs in via Clerk (email/password or Google OAuth).
- Clerk webhook fires to `/api/webhooks/clerk`, syncing the user to Supabase profiles.
- A new organization and organization_members record are created automatically.
- The AuthContext provider fetches the user profile and org membership via `/api/profile`.
- Clerk middleware protects all non-public routes, redirecting unauthenticated users.

Data Flow: Organization Resolution

All API routes use a shared `getUserOrganization()` helper that safely resolves the current user's active organization, handling cases where users belong to multiple organizations. This replaced the previous `.single()` pattern that caused failures for invited members.

4. Database Schema

The application uses Supabase (PostgreSQL) with the following core tables:

organizations	Organization/tenant metadata	id, name, slug, created_by
organization_members	User-org mapping with roles	user_id, organization_id, role, is_active
profiles	User profiles synced from Clerk	id, email, full_name, avatar_url, role
documents	All documents with content	doc_id, title, content, organization_id, team_id, tags
document_embeddings	Vector embeddings for AI	doc_id, organization_id, embedding (1536-dim)
feedback	User feedback on documents	doc_id, user_id, is_helpful, comment
invitations	Pending/accepted team invites	email, organization_id, role, token, status
cluster_names	AI-generated graph labels	organization_id, cluster_index, cluster_name
graph_metadata	Knowledge graph state	organization_id, last_full_refresh
graph_refresh_log	Rate limiting for embeddings	organization_id, user_id, docs_processed

Access Control Model

Owner	Full CRUD on all docs	Invite, remove, change roles	Generate embeddings, view graph
Admin	Full CRUD on all docs	Invite, remove, change roles	Generate embeddings, view graph
Member	View org-wide + own team docs	View member list only	View graph, use chat & search

5. Features

--

5.1 Document Management

The document library is the core of PrimeStride Atlas. Users can create documents directly in the app, upload files (PDF, DOCX), edit content, and organize documents with tags, types, and domains.

--	--	--

List documents	GET /api/documents	Fetch all documents for the user's organization
Create document	POST /api/documents	Create with title, content, tags, type
View document	GET /api/documents/[docId]	Retrieve a single document by ID
Edit document	PUT /api/documents/[docId]	Update content, title, tags, status
Delete document	DELETE /api/documents/[docId]	Remove a document permanently
Upload file	POST /api/documents/upload	Upload PDF or DOCX to Supabase Storage
Parse file	POST /api/documents/parse	Extract text from uploaded PDF/DOCX
AI Summary	POST /api/documents/summarize	AI-generated summary of document content
AI Tags	POST /api/tags/suggest	AI-suggested tags based on content

5.2 AI Chat Assistant (RAG)

The AI Chat uses Retrieval-Augmented Generation to answer questions based on the organization's documents. It provides sourced, citation-backed answers rather than general knowledge.

RAG Pipeline

- User sends a question via POST /api/chat.
- The question is embedded using OpenAI text-embedding-3-small (1536 dimensions).
- Cosine similarity is calculated against all document embeddings in the organization.
- The top 5 most relevant documents (similarity > 0.3) are selected as context.
- Document content (truncated to 2000 chars each) is passed to GPT-4o-mini as system context.
- The model generates an answer citing specific documents, with relevance percentages.

The chat supports conversation history (up to 6 prior messages) for follow-up questions.

5.3 Search

- **Keyword Search:** Text matching against document titles and content with scoring based on title hits, exact phrases, and content frequency. Results include snippets and match explanations.
- **Semantic Search:** AI-powered search using embeddings. The query is embedded and compared against all document embeddings via cosine similarity. Falls back to keyword search if no embeddings exist.

Both modes support filtering by document type, domain, tag, and status.

5.4 AI Knowledge Graph

The Knowledge Graph provides an interactive visualization of document relationships and topic clusters using React Flow.

- **Nodes:** Each document appears as a node on the graph.
- **Edges:** Connections represent similarity > 0.5. Each document links to its top 3 most similar docs.
- **Clusters:** K-means clustering groups documents into topics. $K = \min(5, \max(2, \text{docs}/3))$.
- **Cluster Names:** GPT-4o-mini generates descriptive 2-4 word labels (e.g., 'Sales Management').

Rate Limiting

- Maximum 3 graph refreshes per user per 24 hours
- Maximum 500 documents processed per organization per 24 hours
- 24-hour cooldown per document before re-embedding

5.5 Team Management & Invitations

Invitation Flow

- Admin/Owner sends an invitation via POST `/api/invitations/send` with email and role.
- A branded email is sent via Resend with a unique invitation token link.
- Recipient clicks the link, opening the `/invite/[token]` acceptance page.
- If not signed in, they create a Clerk account or sign in first.
- Upon accepting, they are added to `organization_members` with the assigned role.

Member Management

- View all organization members with roles and join dates
- Change member roles between Member and Admin
- Remove members from the organization
- Resend or cancel pending invitations

5.6 Feedback & Learning Dashboard

Every document has a feedback mechanism (Helpful / Not Helpful with optional comments). This data feeds into the Learning Dashboard at `/learning`, which aggregates feedback counts per document to show which documents perform well and which need improvement.

6. Pages & Navigation

Home / Landing	/	Product overview; redirects to /library if authenticated
Library	/library	Main document list with search and filters
New Document	/library/new	Create a new document form
View Document	/library/[docId]	Document viewer with feedback and related docs
Edit Document	/library/[docId]/edit	Edit document metadata and content
AI Chat	/chat	Conversational AI assistant with RAG
Search	/search	Advanced keyword and semantic search
AI Knowledge Graph	/ai-graph	Interactive document relationship visualization
Learning Dashboard	/learning	Feedback analytics and document quality metrics
Team Settings	/team	Member list, role management, invitations
Teams	/teams	Team creation and management
Admin	/admin	Organization settings and configuration
Accept Invitation	/invite/[token]	Invitation acceptance for new members
Sign In / Sign Up	/sign-in, /sign-up	Clerk-powered authentication pages

7. Components & Libraries

--

React Components

CreateDocumentForm	Rich form for creating documents with file upload, tags, and type selection
EditDocumentForm	Form for editing existing documents with all metadata fields
DocumentView	Document reader with feedback buttons and related documents sidebar
DocumentFeedback	Helpful / Not Helpful feedback buttons with comment support

OrgSwitcher	Dropdown to switch between organizations for multi-org users
UserMenu	User avatar dropdown with profile, settings, and sign-out
InvitationAcceptForm	Form displayed when accepting a team invitation
ProtectedRoute	Wrapper component that redirects unauthenticated users

Shared Libraries

AuthContext	React context providing user profile, role, org ID, and isAdmin flag
getUserOrganization	Shared helper resolving user's active org (handles multi-org)
ai-embeddings	Embedding generation, cosine similarity, k-means clustering
email-templates	Branded HTML email template for invitations via Resend
supabase	Supabase client initialization for server and client usage

8. Deployment & Infrastructure

- **Domain:** primestrideatlas.com (custom domain on Vercel)
- **Hosting:** Vercel with automatic deployments from GitHub main branch
- **Database:** Supabase hosted PostgreSQL with Row Level Security
- **File Storage:** Supabase Storage bucket for document uploads
- **Email:** Resend for transactional invitation emails
- **Auth Provider:** Clerk with Google OAuth via Google Cloud Console

Environment Variables

NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY	Clerk (client-side)
CLERK_SECRET_KEY	Clerk (server-side)
CLERK_WEBHOOK_SECRET	Clerk (webhook verification)
NEXT_PUBLIC_SUPABASE_URL	Supabase (project URL)
SUPABASE_SERVICE_ROLE_KEY	Supabase (admin access)
OPENAI_API_KEY	OpenAI (GPT-4 + embeddings)
RESEND_API_KEY	Resend (email sending)

9. Cost Estimates

Estimated monthly costs for a team of 100 users with 1,000 documents:

Vercel (Hosting)	Pro plan, automatic deployments	\$20/month
Supabase (Database)	Pro plan, PostgreSQL + Storage	\$25/month
Clerk (Auth)	Up to 10,000 MAU on Pro plan	\$25/month
OpenAI (Embeddings)	Initial embedding of 1,000 docs	~\$0.50 one-time
OpenAI (Chat)	~50 chat queries/day	~\$15-30/month
OpenAI (Summaries + Clusters)	On-demand generation	~\$6-12/month
Resend (Email)	Team invitation emails	Free tier

Total estimated cost: **\$90-115/month** for 100 users with 1,000 documents.

10. Planned Features

AI Enhancements

- **Automatic Document Summaries:** AI-generated abstracts created on upload.
- **Auto-Tagging:** GPT-4 suggests 3-5 relevant tags automatically on upload.
- **Enhanced Semantic Search:** Hybrid keyword + semantic scoring for better relevance.
- **AI Writing Assistant:** Grammar fixes, tone matching, and improvement suggestions.

Platform Improvements

- **Organization Switcher UI:** Polished dropdown for multi-org users.
- **Mobile Responsiveness:** Optimized layouts for phone and tablet.
- **Document Permissions:** Fine-grained access control at the document level.
- **Export Capabilities:** Export documents as PDF or DOCX.

Enterprise Features

- **SSO (SAML):** Single sign-on for enterprise customers.
- **Audit Logs:** Track all document and member actions for compliance.
- **Custom Branding:** White-label options per organization.
- **API Access:** Public REST API for third-party integrations.

PrimeStride Atlas — Built by PrimeStride AI
primestrideatlas.com