

# SimplifAI Transforms Knowledge Access with AI-Powered Notion Project Assistant

---

## Company Background

SimplifAI is an AI-first software and automation development agency that builds intelligent solutions for marketing and business operations. Operating through two main divisions—SimplifAI for technical development and Butterfly Digital for marketing services—the company serves small and medium-sized businesses with comprehensive digital solutions.

As the agency scaled, internal knowledge management became a critical challenge. With project documentation, requirements, and action items scattered across Notion databases and Google Drive, team members spent valuable time searching for project context, technical decisions, and implementation details. The leadership team recognized that solving this internal challenge would not only improve their own operations but could become a valuable product offering for clients facing similar knowledge accessibility issues.

---

## Challenge

SimplifAI's growth created significant knowledge management friction that impacted team productivity and project execution:

### Context Switching Overhead

Team members regularly needed to pause development work to search through Notion project cards, PRD documents, and implementation notes to answer simple questions like *"What's the status of feature X?"* or *"Why did we choose this approach?"* This constant context switching disrupted deep work and extended project timelines.

### Scattered Knowledge Sources

Critical project information lived in multiple locations—Notion databases for project tracking, Google Drive for technical documentation, and individual team member memories. No single source provided comprehensive answers to project-related questions, forcing team members to check multiple systems and piece together information manually.

## **Onboarding Bottlenecks**

New team members or those joining mid-project faced steep learning curves. Understanding project history, technical decisions, and current status required extensive documentation review or interrupting colleagues with questions, neither of which scaled efficiently.

## **Time-Sensitive Information Needs**

During client demos and stakeholder meetings, leadership needed instant access to project details—implementation timelines, feature status, technical architecture decisions. Manual searches through Notion cards created awkward delays or led to incomplete answers that undermined confidence.

## **Document Retrieval Accuracy**

Standard keyword search in Notion and Drive often returned irrelevant results or missed context buried within larger documents. Team members couldn't efficiently find specific technical details or past decisions without reading through entire documents.

## **Business Impact**

These knowledge access challenges reduced billable hours, slowed project velocity, increased onboarding time, and created dependency on specific team members who held institutional knowledge. The company needed a way to make project knowledge instantly accessible without disrupting existing workflows.

---

## **Solution & Approach**

SimplifAI built a custom RAG (Retrieval-Augmented Generation) chatbot that transforms their Notion workspace and Google Drive into a conversational knowledge interface. The solution enables natural language queries about any project, returning accurate, source-cited answers in seconds.

### **Strategic Approach**

#### **Why This Approach vs. Alternatives**

Rather than implementing a generic enterprise search tool or third-party knowledge base, SimplifAI chose to build a custom RAG solution for several strategic reasons:

- **Control Over Data Pipeline:** Direct integration with their specific Notion structure and Drive organization ensured optimal indexing of their unique project documentation format.
- **Customized Context Understanding:** Fine-tuned prompt engineering could be tailored to their internal terminology, project structures, and question patterns.
- **Citation Requirements:** Built-in source attribution was critical for technical accuracy and verification—generic chatbots often lack reliable citation mechanisms.
- **Privacy and Security:** Keeping the solution self-hosted avoided exposing proprietary client project details to third-party services.
- **Extensibility:** A custom build allows future enhancements like specific role-based access, integration with other internal tools, and potential productization.

The team deliberately chose **Qdrant** over pgvector for vector storage based on superior search performance and dedicated vector database optimization. For reranking, they implemented **BM25 (Best Match 25)** rather than paid services like Cohere, balancing accuracy with cost efficiency for their query volume.

---

## Implementation Phases

### Phase 1: Architecture & Data Integration (2025-09-28 to 2025-10-02)

- Researched RAG architecture patterns and vector database options
- Selected OpenAI's **text-embedding-3-large** model for semantic understanding
- Established n8n workflows for automated data ingestion from Google Drive
- Configured Qdrant vector database with proper metadata schema
- Built initial data digestion pipeline with automatic vector point deletion when source documents are removed

### Phase 2: Core RAG Development (2025-10-05 to 2025-10-14)

- Finalized embedding and deletion patterns for data consistency
- Implemented metadata indexing on Qdrant for precise filtering
- Fixed critical bugs in vector search configuration
- Developed streaming response architecture for real-time chat experience
- Created initial RAG prompts optimized for technical project queries
- Built Server-Sent Events (SSE) pipeline for UI streaming

### Phase 3: Citation & Context Enhancement (2025-10-15 to 2025-10-21)

- Added citation functionality linking answers to source documents
- Fixed data pipeline connecting Qdrant metadata to UI display
- Implemented custom vector search strategy (external reranking unnecessary)
- Developed Notion API integration for direct project card access

- Created dual-path retrieval: vector search + Notion MCP for comprehensive coverage
- Implemented document digestion workflow converting Notion cards to searchable Drive documents

#### **Phase 4: Conversational Intelligence (2025-10-22 to 2025-10-23)**

- Built chat history storage enabling contextual, multi-turn conversations
- Tuned response style for technical accuracy balanced with readability
- Enhanced context awareness so follow-up questions reference prior conversation

#### **Phase 5: Production Polish (2025-10-26 to 2025-10-29)**

- Deployed application to Netlify with production configuration
  - Refined chat UI for professional presentation
  - Created usage documentation for team onboarding
  - Conducted live demo with CTO for validation
  - Fine-tuned system instructions and tool-calling features for GPT-5 model
  - Prepared for upcoming database property enhancements (task ownership, due dates, assignments)
- 

### **Key Technologies**

- **Frontend:** Next.js with Server-Side Rendering
  - **Backend:** n8n automation workflows
  - **Vector Database:** Qdrant for semantic search with metadata filtering
  - **Embedding Model:** OpenAI **text-embedding-3-large** via OpenRouter
  - **LLM:** GPT-5 for response generation with streaming support
  - **Integrations:** Notion API, Google Drive API, Notion MCP
  - **Deployment:** Netlify with optimized build configuration
- 

### **Core Deliverables**

#### **1. Intelligent Conversational Interface**

- Natural language query processing for project-related questions
- Multi-turn conversations with context retention
- Streaming responses for immediate feedback
- Source citations with direct links to Notion cards and Drive documents

#### **2. Dual-Path Retrieval System**

- Vector semantic search via Qdrant for conceptual understanding

- Notion MCP integration for structured project data access
- Custom BM25-based reranking without external dependencies
- Metadata filtering for precise scope control

### **3. Automated Knowledge Sync**

- Real-time ingestion from Google Drive to vector database
- Automatic Notion project card conversion to searchable documents
- Deletion workflow maintaining data consistency
- Sheet integration tracking ingestion status

### **4. Enterprise-Grade Architecture**

- OAuth 2.0 secure authentication
  - Encrypted token storage
  - Horizontal scaling capability for multiple workspaces
  - At-least-once ingestion with idempotent operations
- 

## **Implementation Timeline**

### **Week 1: Research & Planning (2025-09-28 to 2025-09-30)**

- Researched RAG architecture patterns and reranking algorithms
- Established project timeline and identified missing resources
- Created technical specification document (PRD)

### **Week 2: Foundation (2025-10-01 to 2025-10-07)**

- Built core data pipeline connecting Google Drive to Qdrant via n8n
- Implemented OpenAI embedding integration and automated document ingestion
- Fixed critical sheet workflow bug
- Began initial UI development with Next.js

### **Week 3: RAG Core (2025-10-08 to 2025-10-14)**

- Developed complete RAG workflow with streaming response support
- Created optimized prompts for technical query handling
- Configured metadata structure on Qdrant and fixed errors
- Integrated embedding, deletion, and chat components

### **Week 4: Search Optimization (2025-10-15 to 2025-10-21)**

- Implemented custom vector search strategy without external reranking
- Added comprehensive citation support with source attribution
- Integrated Notion API for direct project card access
- Built content digestion pipeline converting Notion cards to Drive documents
- Fixed critical sync and cleanup workflows

### **Week 5: Conversational Intelligence (2025-10-22 to 2025-10-26)**

- Implemented chat history storage enabling multi-turn contextual conversations
- Integrated Notion MCP as alternate retrieval path
- Developed custom reranking based on document relevance
- Tuned response style for technical precision and readability

### **Week 6: Production & Demo (2025-10-27 to 2025-10-29)**

- Finalized UI polish and internal prompt optimization
- Demoed to CTO (Redwan Hossain) with positive validation
- Deployed to Netlify with production configuration
- Created comprehensive usage documentation
- Fine-tuned for GPT-5 model with enhanced tool-calling

### **Ongoing: Enhancement Pipeline (2025-11-02+)**

- Planning next iteration features: task ownership, assignee tracking, status indicators, enhanced database property access
  - Addressing edge cases in due date handling and project summary displays
- 

## **Results & Metrics**

### **Dramatic Efficiency Gains**

#### **Knowledge Access Speed**

The chatbot transformed project information retrieval from a multi-minute manual search process to **sub-5-second conversational queries**. Team members now ask questions in natural language and receive accurate, cited answers instantly.

#### **Context Switching Elimination**

Developers and project managers no longer interrupt deep work to hunt for project details. The chat interface provides **immediate access** to technical decisions, implementation status, and project history without leaving current workflow.

## Meeting Preparation Time Reduction

Leadership team members preparing for client demos or stakeholder meetings now spend **minutes instead of hours** gathering project context, with source citations enabling quick verification.

---

## Technical Performance & Accuracy

### Citation Reliability

Every response includes **direct links** to source Notion cards and Google Drive documents. Dual-path retrieval (vector search + Notion MCP) ensures **comprehensive coverage** of unstructured and structured project data.

### Conversational Context Retention

Multi-turn conversation support allows natural follow-up questions without re-establishing context. The system maintains **conversation continuity** through intelligent chat history management.

### Search Quality

Custom BM25-based reranking delivers **high-precision results** without external API costs. The system successfully retrieves relevant information across **50+ project cards** and associated Drive documents.

---

## Operational Impact

### Onboarding Acceleration

New team members now interact with the chatbot to understand project history and current status, **reducing the burden** on existing team members.

### Institutional Knowledge Preservation

Project decisions and technical rationale are now **permanently searchable**, reducing organizational risk and enabling knowledge transfer at scale.

### Workflow Integration

Notion MCP integration and automated Drive sync ensure the chatbot stays **continuously updated** with latest project changes, eliminating verification anxiety.

---

## System Reliability

## Data Consistency

Automated deletion workflows maintain vector database accuracy, ensuring **no stale information** misleads users.

## Streaming Performance

Server-Sent Events architecture delivers **real-time response streaming**, maintaining engagement during answer generation.

## Scalability Foundation

Architecture supports **horizontal scaling** for multiple workspaces and increased query volume.

---

## Business Value

The PKMS chatbot directly addresses the time-to-information bottleneck that constrained operational efficiency. By making project knowledge instantly accessible through natural conversation, the system **reclaims valuable hours weekly**, enables faster decision-making, and creates a scalable foundation for knowledge management.

---

## Client Testimonial

*"This chatbot has fundamentally changed how our team accesses project information. What used to take 10 minutes of searching through Notion and Drive now happens in seconds through a simple question. The citation links give us confidence in the answers, and the conversational interface feels natural. We're not just saving time—we're preserving institutional knowledge that would otherwise be lost."*

— Redwan Khan, CTO, SimplifAI

---

## Key Takeaways

### 1. RAG Architecture Requires Thoughtful Trade-offs

Custom BM25 reranking proved critical for cost management while maintaining quality. **Thoughtful algorithm selection beats premium APIs**—SimplifAI's query patterns and document corpus didn't require external reranking, saving operational costs.

### 2. Dual-Path Retrieval Solves Different Knowledge Types

Vector semantic search (for unstructured docs) combined with Notion MCP (structured data) created **complementary retrieval strategies**. Hybrid approaches outperform single-method solutions for diverse knowledge sources.

### 3. Citations Transform User Trust and Adoption

Linking every answer to specific Notion cards and Drive documents became the **most valued capability**, enabling instant verification and confidence in responses.

### 4. Streaming Responses Are Non-Negotiable for Chat UX

Server-Sent Events real-time streaming improved perceived performance. Users feel engaged watching answers appear token-by-token, highlighting that **psychological performance often matters more than raw speed**.

### 5. Internal Tools Become Product Prototypes

Solving internal knowledge challenges created **directly transferable architecture and learnings** for client projects, validating potential product offerings while delivering immediate operational value.

---

**Project Duration:** 2025-09-28 to 2025-10-29 (5 weeks active development)

**Development Team:** SimplifAI Technical Team

**Service Type:** Custom Software Development - RAG/AI Application

**Current Status:** Production deployment with ongoing enhancements

**Technologies:** Next.js, Qdrant, OpenAI, Notion API, n8n, Netlify, Google Drive API