**Use Case: RAG-Based Project Onboarding Assistant for Tech Companies**

**Problem:**

In many tech companies, developers frequently join new projects. When onboarding, they must understand:

* Project architecture
* Dev environment setup
* Codebase structure
* API usage
* Deployment procedures
* Recent sprint outcomes or backlog items

However, challenges include:

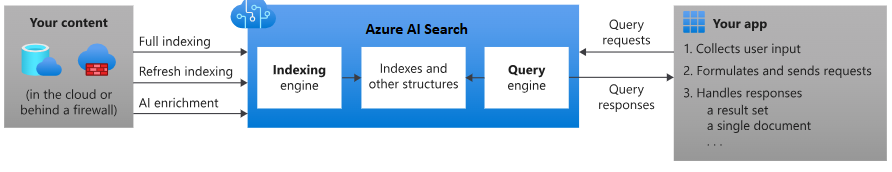
* Documentation is scattered across tools (Notion, GitHub, SharePoint)
* Seniors are busy and hard to reach
* Docs become stale and hard to search

**Solution:**

A **RAG (Retrieval-Augmented Generation)–powered AI Assistant** that lets developers ask natural-language questions like:

* “How do I run the application locally?”
* “What does the token API do?”
* “What are the Deployment procedures?”

The system retrieves the most relevant content from internal documents, augments the query, and uses a GPT model to generate accurate, contextual answers.



Example: Running the Application Locally  
User Query: "How do I run the application locally?"  
System Prompt: "You are a helpful onboarding assistant. The user wants to know how to run the application locally."  
Retrieved Context: "To run the application locally, follow these steps: 1. Clone the repository from GitHub. 2. Install dependencies using npm install. 3. Start the development server using npm start."  
Final Prompt: "You are a helpful onboarding assistant. The user wants to know how to run the application locally. To run the application locally, follow these steps: 1. Clone the repository from GitHub. 2. Install dependencies using npm install. 3. Start the development server using npm start."

**Document Ingestion Pipeline**

Add a preprocessing step to chunk and clean documents before indexing. Using tools like Azure Form Recognizer for PDFs or Notion API for structured content extraction.

Steps:  
1. Extract content from various sources (GitHub, Notion, PDFs).  
2. Clean and preprocess the content (remove noise, split into chunks).  
3. Index the cleaned chunks into Azure AI Search.

**Users:**

* **New developers** joining a project
* **QA or DevOps engineers** onboarding on infrastructure
* **Project managers** reviewing documentation context

**Tools & Technologies:**

| **Component** | **Tools used** |
| --- | --- |
| Embeddings | Azure OpenAI (text-embedding-ada-002) |
| Retriever | Azure AI Search – Content ingested into Search Index |
| LLM | Azure OpenAI (gpt-4.1) |
| Storage | GitHub, Notion, PDF docs |
| Frontend | Internal web app or chatbot |

**RAG Workflow:**

[1] User Asks a Question

[2] Embed the Query

- Use `text-embedding-ada-002`

[3] Perform Vector Search

- Search in pre-indexed doc chunks

- Return top N most relevant chunks

[4] Construct the Prompt

- System Prompt "You are a helpful onboarding assistant..."

- Add user query and retrieved context

[5] Call GPT-4.1 (LLM) via Chat Completion API

- Model reads context and answers question

[6] Return the Answer

- Show answer + sources