Chatbot System - Development Guide

Overview

We are building a chatbot with two main components:

1. **Data Ingestion System** (runs locally)
2. **Chatbot** (deployed on Cloudflare)

---

1. Data Ingestion System

The purpose of this system is to prepare data (from PDFs) and store embeddings for chatbot retrieval.

Steps:

1. **Read PDF**
   - Extract text content from the PDF files.
   - Use libraries like pdf2md.

2. **Convert to Text & Clean**
   - Normalize formatting (remove extra spaces, newlines, etc.).

3. **Chunking**
   - Split text into smaller chunks (e.g., 500–1000 characters with some overlap).
   - Helps embeddings and retrieval work better.

4. **Create Embeddings**
   - Use **OpenAI Embeddings API** (e.g., `text-embedding-3-small` or `text-embedding-3-large`).

5. **Store in Cloudflare Vectorize**
   - Push embeddings + metadata (document name, page number, chunk ID, original text) into **Cloudflare Vectorize**.

---

2. Chatbot (Cloudflare Worker)

The chatbot runs serverless using **Cloudflare Workers** and connects with external APIs for retrieval + inference.

Components:

1. **User Query Handling**
   - Accept user input (query).

2. **Embedding User Query**
   - Create embedding for user query using **OpenAI Embeddings API**.

3. **Vector Search**
   - Use **Cloudflare Vectorize** to search for most relevant text chunks.

4. **Inference (Response Generation)**
   - Send query + retrieved chunks to **Groq API** (LLM inference).
   - Model generates the answer.

5. **Return Response**
   - The chatbot returns the response to the user.

Tech Stack

- **Cloudflare Worker** → for chatbot runtime.
- **Cloudflare Vectorize** → for vector database storage + retrieval.
- **Groq API** → for fast LLM inference.
- **OpenAI API** → for embeddings.
- **Local NodeJS Script** → for data ingestion (PDF → Text → Embedding → Store).

Workflow Diagram (Simplified)

```
[PDF Files]
   ↓
[Data Ingestion System (local)]
   - Read PDF
   - Chunk text
   - Create embeddings (OpenAI)
   - Store embeddings (Cloudflare Vectorize)
```

[Chatbot (Cloudflare Worker)]
- Take user query
- Create embedding (OpenAI)
- Search embeddings (CF Vectorize)
- Send context + query → Groq API
- Return answer