Neural RAG System Architecture & Flow

1. TOOLS USED

Below is the list of tools that power your document-based chatbot:

- FastAPI (Python) (The Backend Framework.)

Acts as the bridge between the website (frontend) and the AI (backend). It handles file uploads and user questions.

- Ollama (Local AI Runner.)

Runs the Large Language Model (LLM) directly on your computer. This ensures privacy and zero costs (no OpenAI key needed).

- Qdrant (Vector Database.)

A specialized "smart" database that stores documents as numbers (vectors) so it can find answers by meaning, not just keywords.

- all-MiniLM-L6-v2 (Embedding Model.)

Translates human language into a list of numbers that the computer can understand and compare.

- RapidOCR (Optical Character Recognition.)

The "eyes" of the system. It reads text from scanned images and handwritten documents during upload.

2. TECHNIQUES USED

- RAG (Retrieval-Augmented Generation)

The main strategy. Instead of guessing, the AI finds the right document part (Retrieval) and then explains it (Generation).

- Embeddings

The process of turning text into a "mathematical fingerprint" so the computer can see how similar two pieces of text are.

- Vector Search

Searching the database using meaning. If you search for "automobile", it can find documents about "cars" because they are mathematically close.

- Chunking

Breaking long documents into small pieces (chunks). This helps the AI focus on the specific section containing the answer.

3. STEP-BY-STEP WORKING

Step 1: Document Upload - You upload a PDF/Docx on the website.

Step 2: Chunking - The system splits the text into small 500-character pieces.

Neural RAG System Architecture & Flow

Step 3: Embedding Creation - Each piece is converted into numbers (vectors).

Step 4: Storing in Qdrant - These numbers are saved in the smart database.

Step 5: User Question - You ask a question (e.g., "What is the premium?").

Step 6: Vector Search - The system finds the top 3 most relevant document pieces.

Step 7: Relevance Check - The system pulls the actual text for those 3 pieces.

Step 8: LLM Answer - The AI reads the pieces and gives you the final answer based ONLY on that text. If not found, it says "Query not found in document."

4. EXAMPLE DOCUMENT

Sample Insurance Policy: LI-9988
- Insured: John Doe
- Coverage: Life Insurance
- Benefit Amount: $500,000
- Monthly Premium: $45
- Deductible: $0

5. EXAMPLE QUESTION & ANSWER

Question (In Document): "How much is the monthly premium?"
Answer: "The monthly premium is $45."

Question (NOT In Document): "What is my car plate number?"
Answer: "Query not found in document."