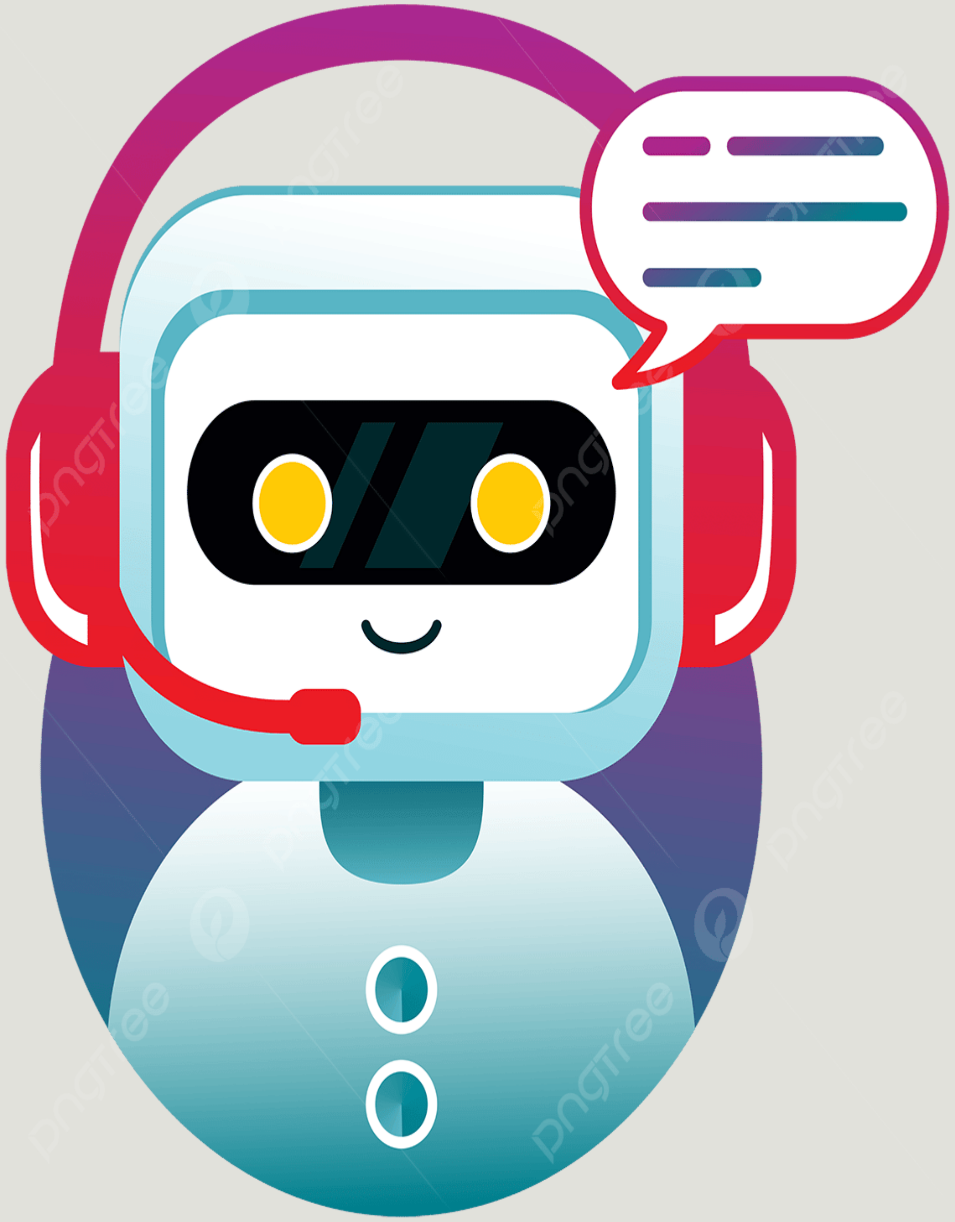
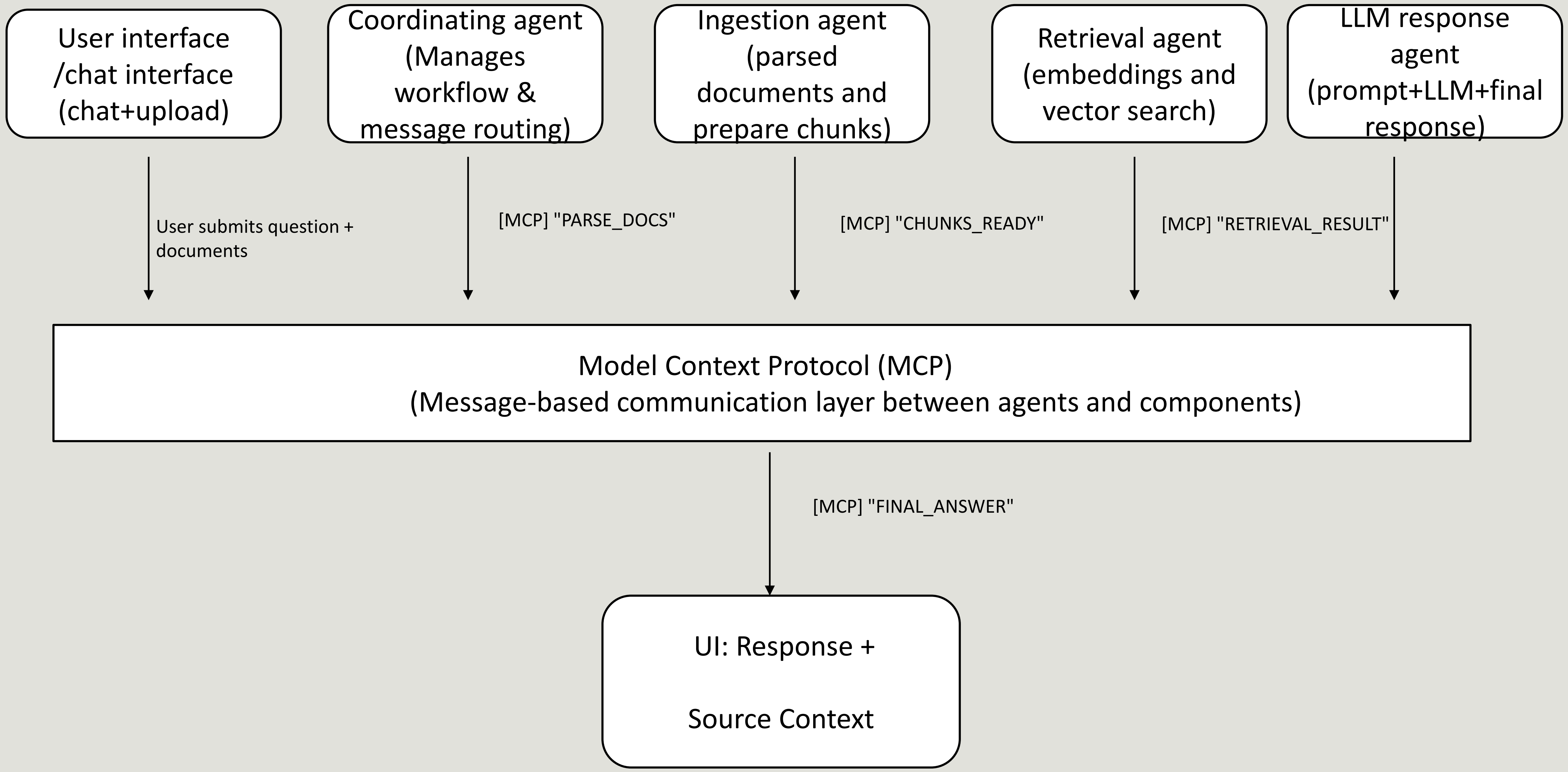


Agentic RAG Chatbot for Multi-Format Document QA

Introduction

This project presents the **Agentic RAG Chatbot**, an intelligent question-answering system designed to efficiently handle multi-format documents by leveraging the **Model Context Protocol (MCP)**. By combining retrieval-augmented generation with MCP, the chatbot dynamically accesses and processes diverse document types—including PDFs, Word files, spreadsheets, and more—to deliver accurate, context-aware responses. This innovative integration enables the system to maintain relevant context across user interactions and improve answer precision.


User interface /chat interface (chat+upload)
Coordinating agent (Manages workflow & message routing)
Ingestion agent (parsed documents and prepare chunks)
Retrieval agent (embeddings and vector search)
LLM response agent (prompt+LLM+final response)
User submits question + documents
[MCP] "PARSE_DOCS"
[MCP] "CHUNKS_READY"
[MCP] "RETRIEVAL_RESULT"
Model Context Protocol (MCP) (Message-based communication layer between agents and components)
[MCP] "FINAL_ANSWER"
UI: Response + Source Context

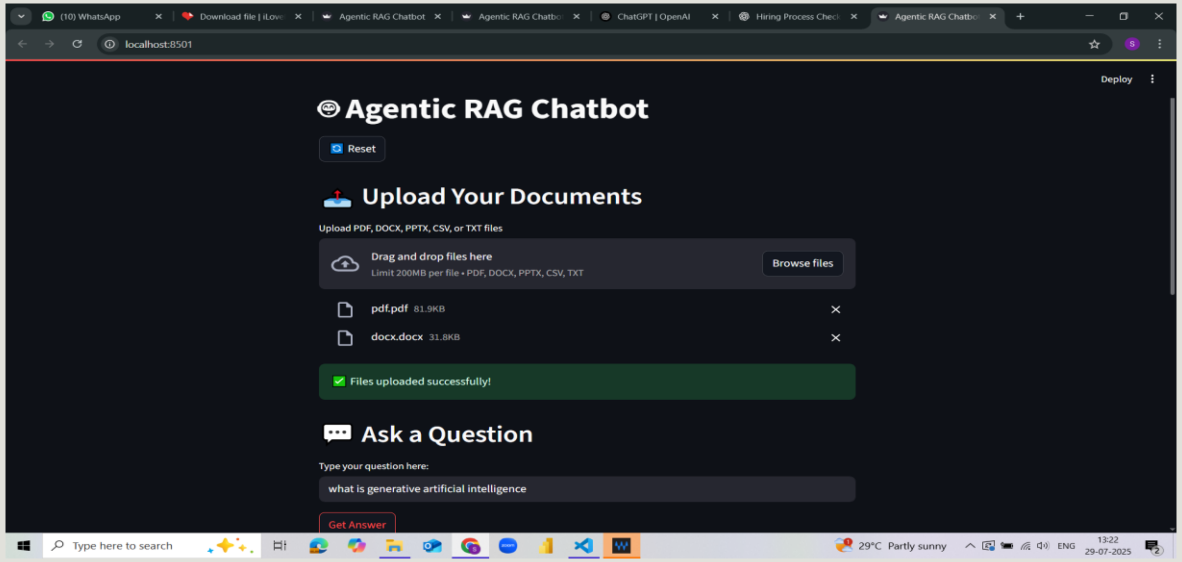
MCP protocol

```
import uuid

def create_mcp_message(sender, receiver, msg_type, payload, trace_id=None):
    return {
        "sender": sender,
        "receiver": receiver,
        "type": msg_type,
        "trace_id": trace_id or str(uuid.uuid4()),
        "payload": payload
    }
```

Category	Technology	Purpose
Frontend (UI)	Streamlit	Chat interface, file uploads, user interaction
Backend Logic	Python	Core development language
Agents	Custom Python Classes	IngestionAgent, RetrievalAgent, LLMResponseAgent
Messaging Layer	Model Context Protocol (MCP)	Message-passing between agents (in-memory / REST)
Document Parsing	PyMuPDF, python-docx, python-pptx, pandas	Parsing PDF, DOCX, PPTX, CSV, TXT files
Embeddings	Hugging Face Transformers	Generating vector representations of chunks
Vector Store	Vector DB with FAISS	Storing and retrieving document embeddings
LLM	google/flan-t5-base	Answer generation using retrieved context
File Handling	Python OS / tempfile / base64	Upload, read, and manage user documents
Deployment	Streamlit	App deployment and demo hosting
Version Control	Git + GitHub	Code management and collaboration

Results:



Agentic RAG Chatbot

Reset

Upload Your Documents

Upload PDF, DOCX, PPTX, CSV, or TXT files

Drag and drop files here
Limit 200MB per file • PDF, DOCX, PPTX, CSV, TXT

Browse files

pdf.pdf 81.9KB

docx.docx 31.8KB

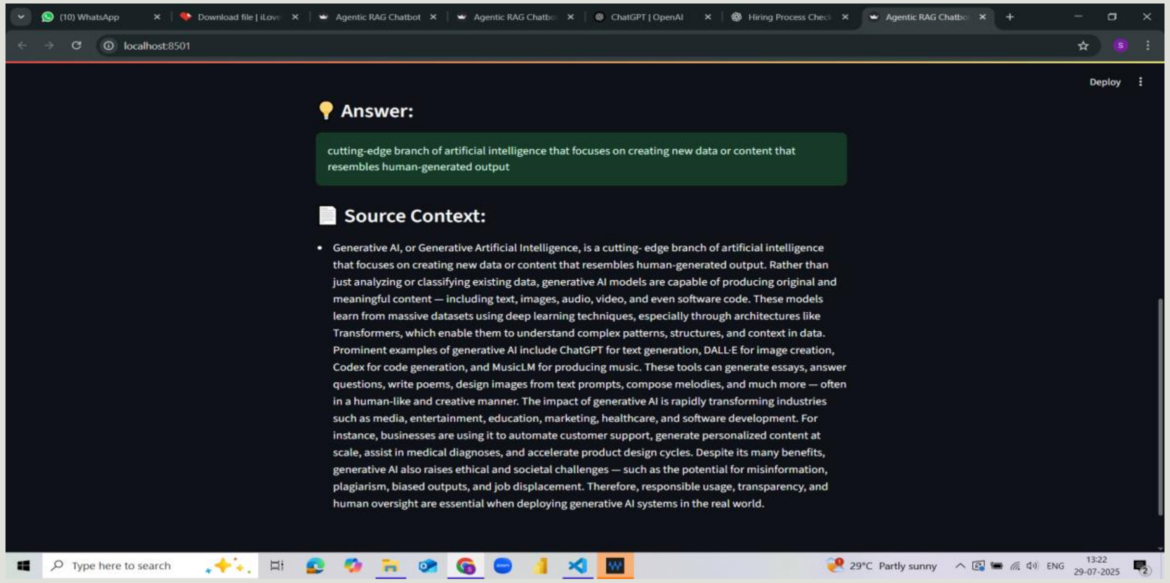
Files uploaded successfully!

Ask a Question

Type your question here:

what is generative artificial intelligence

Get Answer



Answer:

cutting-edge branch of artificial intelligence that focuses on creating new data or content that resembles human-generated output

Source Context:

- Generative AI, or Generative Artificial Intelligence, is a cutting- edge branch of artificial intelligence that focuses on creating new data or content that resembles human-generated output. Rather than just analyzing or classifying existing data, generative AI models are capable of producing original and meaningful content — including text, images, audio, video, and even software code. These models learn from massive datasets using deep learning techniques, especially through architectures like Transformers, which enable them to understand complex patterns, structures, and context in data. Prominent examples of generative AI include ChatGPT for text generation, DALL·E for image creation, Codex for code generation, and MusicLM for producing music. These tools can generate essays, answer questions, write poems, design images from text prompts, compose melodies, and much more — often in a human-like and creative manner. The impact of generative AI is rapidly transforming industries such as media, entertainment, education, marketing, healthcare, and software development. For instance, businesses are using it to automate customer support, generate personalized content at scale, assist in medical diagnoses, and accelerate product design cycles. Despite its many benefits, generative AI also raises ethical and societal challenges — such as the potential for misinformation, plagiarism, biased outputs, and job displacement. Therefore, responsible usage, transparency, and human oversight are essential when deploying generative AI systems in the real world.

Challenges Faced

Repetitive Content in Source Documents

Several documents contained repeated paragraphs or duplicated content across different formats (e.g., same text in PDF and PPTX).

This led to redundant chunks being passed to the LLM, increasing token usage and sometimes causing repetitive or verbose responses.

Solution: Implemented a deduplication step during document parsing to filter out repeated text before embedding and retrieval.