ESSENTRA

AGENTIC RAG CHATBOT WITH MODEL CONTEXT PROTOCOL

Task Title Task Title: Agentic RAG Chatbot for Multi-Format Document QA using Model Context Protocol

GitHub Link : https://github.com/TirumalaManav/essentra-ai

Built by: TIRUMALA MANAV

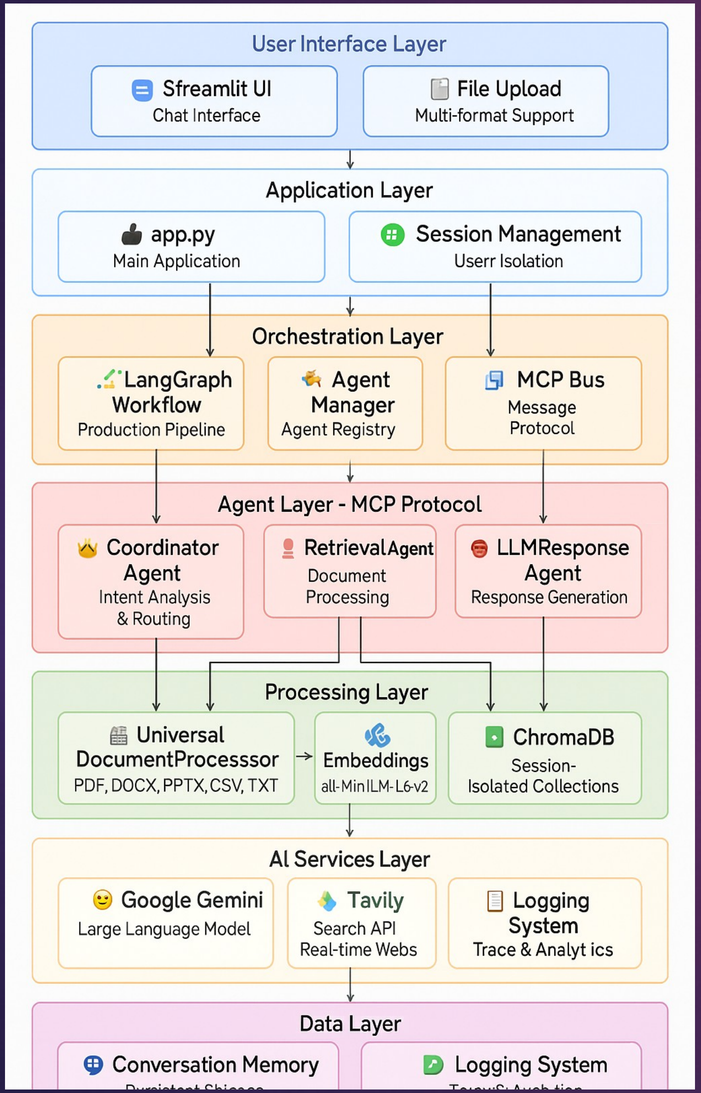
👤 Manav Tirumala
✉ Email: thirumalamanav123@gmail.com
🔗 LinkedIn: linkedin.com/in/tirumalamanav
🐙 GitHub: github.com/TirumalaManav

Coding Task and Solution

- THE CHALLENGE
- • Organizations struggle with document intelligence across multiple formats
- • Traditional Chatbots lack structured agent communication
- • No standardized protocol for inter-agent messaging
- • Limited scalability and production readiness

- THE SOLUTION: ESSENTRA
- ☑ 5 Specialized Agents with MCP Protocol
- ☑ Multi-Format Document Processing (PDF, DOCX, PPTX, CSV, TXT, MD)
- ☑ LangGraph Workflow Orchestration
- ☑ Production-Ready Architecture with Chroma DB Vector Storage

System Architecture Overview



User Interface Layer

Sfreamlit UI
Chat Interface

File Upload
Multi-format Support

Application Layer

app.py
Main Application

Session Management
Userr Isolation

Orchestration Layer

LangGraph Workflow
Production Pipeline

Agent Manager
Agent Registry

MCP Bus
Message Protocol

Agent Layer - MCP Protocol

Coordinator Agent
Intent Analysis & Routing

RetrievalAgent
Document Processing

LLMResponse Agent
Response Generation

Processing Layer

Universal DocumentProcesssor
PDF, DOCX, PPTX, CSV, TXT

Embeddings
all-MinILM-L6-v2

ChromaDB
Session-Isolated Collections

Al Services Layer

Google Gemini
Large Language Model

Tavily
Search API
Real-time Webs

Logging System
Trace & Analyt ics

Data Layer

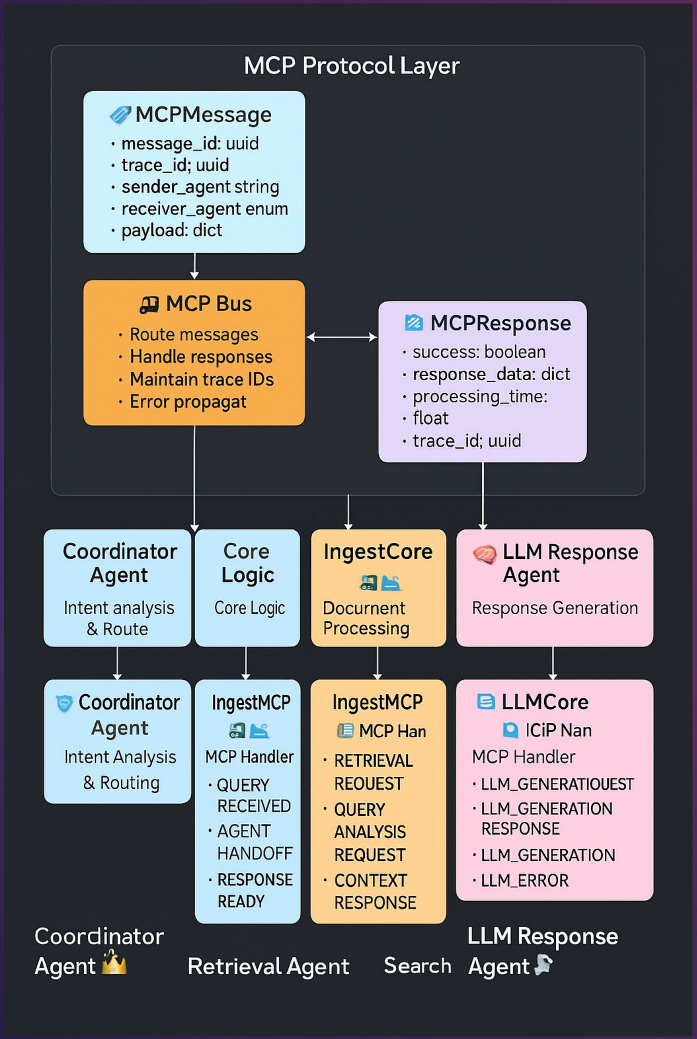
Conversation Memory

Logging System

- LAYERED ARCHITECTURE:
- User Interface Layer: Streamlit UI + File Upload
- Application Layer: Main App + Configuration + Session Management
- Orchestration Layer: LangGraph Workflow + MCP Bus
- Agent Layer: 5 Specialized Agents with MCP Communication
- Processing Layer: Document Processor + Embeddings + Vector Store
- AI Services: Google Gemini 1.5 Flash + Tavily Search API
- Data Layer : Conversation Memory and Logging System

Note: The architecture diagrams above were created using Canvas and SVG code to enhance visualization and improve readability, taking the project's structure into careful consideration

Agent Architecture and MCP Protocol

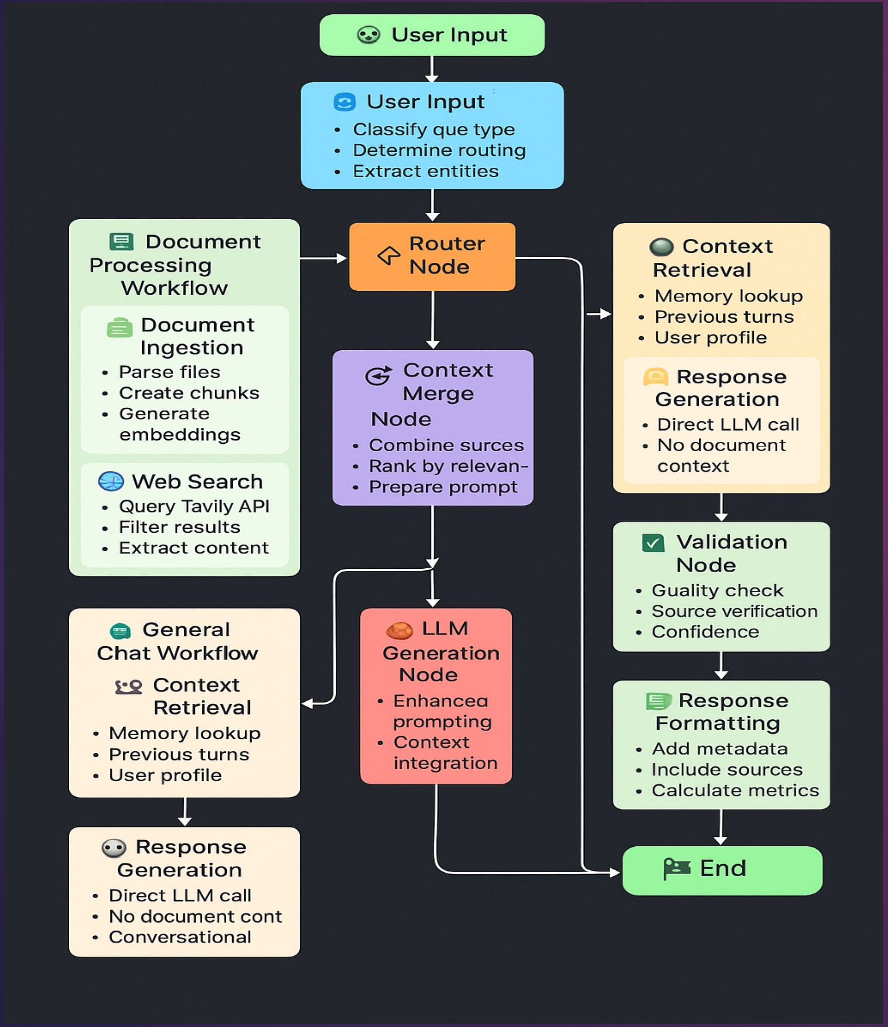


- 5 INTELLIGENT AGENTS:
- CoordinatorAgent - Intent analysis & routing decisions
- IngestionAgent - Multi-format document processing
- RetrievalAgent - Semantic search & context ranking
- LLMResponseAgent - Enhanced response generation
- WebSearchAgent - Real-time information retrieval

- MCP PROTOCOL FEATURES:
- • Structured message passing with trace IDs
- • Error propagation and handling
- • Processing time tracking
- • Response validation

Note: The architecture diagrams above were created using Canvas and SVG code to enhance visualization and improve readability, taking the project's structure into careful consideration

LangGraph Workflow and Message Flow



User Input

User Input
- Classify que type
- Determine routing
- Extract entities

Document Processing Workflow

Document Ingestion
- Parse files
- Create chunks
- Generate embeddings

Web Search
- Query Tavily API
- Filter results
- Extract content

Router Node

Context Merge Node
- Combine surces
- Rank by relevan-
- Prepare prompt

Context Retrieval
- Memory lookup
- Previous turns
- User profile

Response Generation
- Direct LLM call
- No document context

Validation Node
- Guality check
- Source verification
- Confidence

Response Formatting
- Add metadata
- Include sources
- Calculate metrics

General Chat Workflow

Context Retrieval
- Memory lookup
- Previous turns
- User profile

LLM Generation Node
- Enhancea prompting
- Context integration

Response Generation
- Direct LLM call
- No document cont
- Conversational

End

INTELLIGENT ROUTING:

- Intent Analysis Node - Classifies query type and entities
- Router Node - Determines optimal processing pathway
- Context Merge Node - Combines multiple information sources
- Validation Node - Quality assurance and confidence scoring
- Memory Update Node - Persistent conversation storage

WORKFLOW TYPES:

📄 Document Processing Workflow

🌐 Web Search Workflow

💬 General Chat Workflow

Note: The architecture diagrams above were created using Canvas and SVG code to enhance visualization and improve readability, taking the project's structure into careful consideration

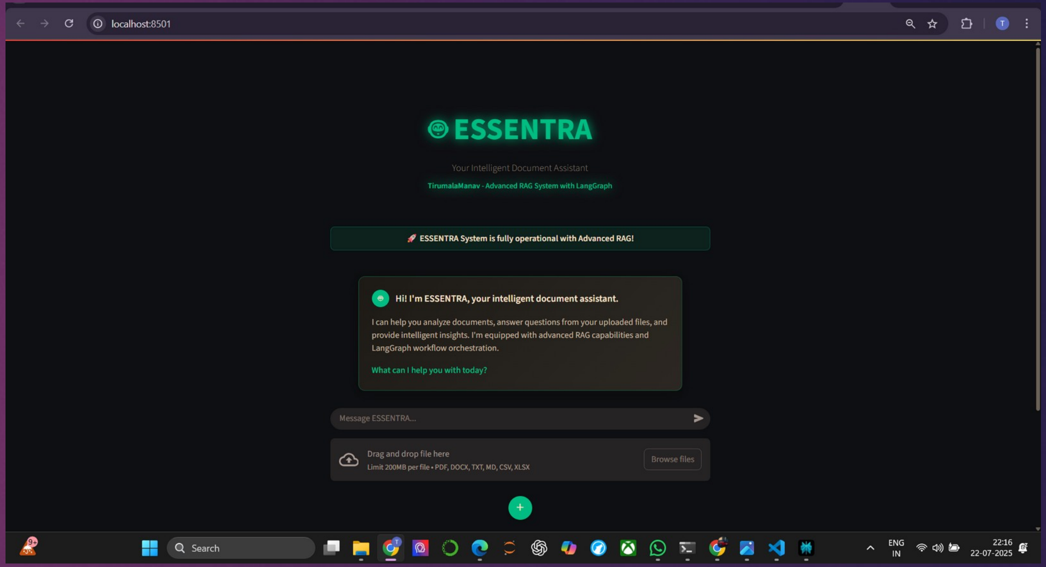
Technology Stack

CORE TECHNOLOGIES:

Frontend: Streamlit with ChatGPT-like UI
Orchestration: LangGraph Workflow Engine
Communication: Custom MCP Protocol Implementation
LLM: Google Gemini 1.5 Flash
Web Search: Tavily Search API
Vector DB: ChromaDB with Session Isolation
Embeddings: Sentence Transformers (all-MiniLM-L6-v2)

FEATURES:

- ➢ Async Processing & Performance Optimization
- ➢ Conversation Memory with Persistence
- ➢ Session Management & User Isolation
- ➢ Comprehensive Logging & Analytics
- ➢ Production-Grade Error Handling


localhost:8501
ESSENTRA
Your Intelligent Document Assistant
TirumalaManav - Advanced RAG System with LangGraph
Message ESSENTRA...
Drag and drop file here
Limit 200MB per file • PDF, DOCX, TXT, MD, CSV, XLSX
Browse files
Search
ENG
IN
22:16
22-07-2025

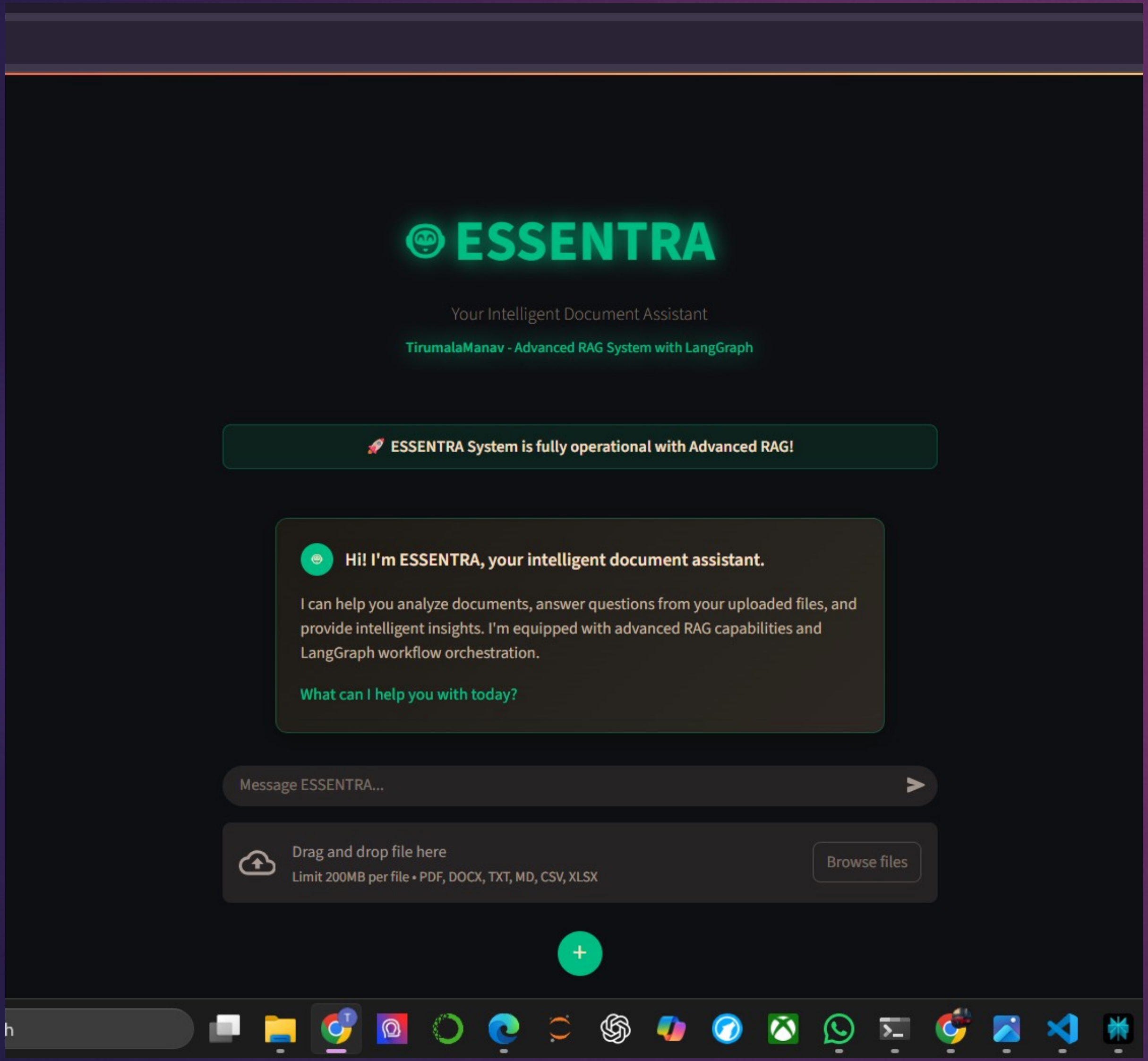
🚀 ESSENTRA System is fully operational with Advanced RAG!

Hi! I'm ESSENTRA, your intelligent document assistant.

I can help you analyze documents, answer questions from your uploaded files, and provide intelligent insights. I'm equipped with advanced RAG capabilities and LangGraph workflow orchestration.

What can I help you with today?

UI Screenshots and Features



ESSENTRA

Your Intelligent Document Assistant

TirumalaManav - Advanced RAG System with LangGraph

🚀 ESSENTRA System is fully operational with Advanced RAG!

Hi! I'm ESSENTRA, your intelligent document assistant.

I can help you analyze documents, answer questions from your uploaded files, and provide intelligent insights. I'm equipped with advanced RAG capabilities and LangGraph workflow orchestration.

What can I help you with today?

Message ESSENTRA...

Drag and drop file here

Limit 200MB per file • PDF, DOCX, TXT, MD, CSV, XLSX

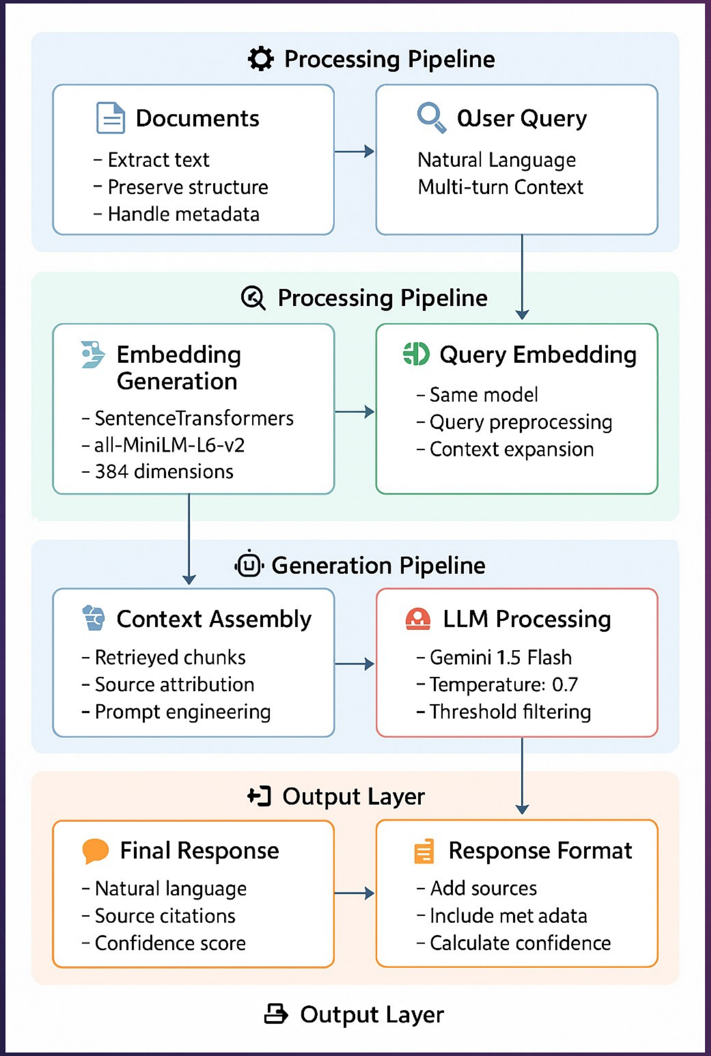
KEY UI FEATURES:

- Clean, Modern ChatGPT-Style Interface
- Drag & Drop File Upload with Progress Indicators
- Real-time Processing Feedback with Spinners
- Source Attribution with Confidence Scores
- Multi-turn Conversation Support
- Responsive Design with Smooth Animations

USER EXPERIENCE:

- Intuitive file upload for multiple formats
- Instant processing feedback
- Clear source citations
- Professional presentation

Data Processing Pipeline



Processing Pipeline

Documents
- Extract text
- Preserve structure
- Handle metadata

OUser Query
Natural Language
Multi-turn Context

Processing Pipeline

Embedding Generation
- SentenceTransformers
- all-MiniLM-L6-v2
- 384 dimensions

Query Embedding
- Same model
- Query preprocessing
- Context expansion

Generation Pipeline

Context Assembly
- Retrieyed chunks
- Source attribution
- Prompt engineering

LLM Processing
- Gemini 1.5 Flash
- Temperature: 0.7
- Threshold filtering

Output Layer

Final Response
- Natural language
- Source citations
- Confidence score

Response Format
- Add sources
- Include met adata
- Calculate confidence

Output Layer

Key Features

- Document Parsing - Extract text while preserving structure
- Smart Chunking - 500 tokens with 50 token overlap
- Embedding Generation - 384-dimensional vectors
- Vector Storage - Session-isolated ChromaDB collections
- Semantic Retrieval - Cosine similarity with ranking
- Context Assembly - Multi-source information fusion
- LLM Generation - Enhanced prompting with Gemini
- Response Formatting - Source citations & metadata

Note: The architecture diagrams above were created using Canvas and SVG code to enhance visualization and improve readability, taking the project's structure into careful consideration

Challenges Faced and Solutions

CHALLENGE 1: MCP Protocol Implementation

- ✕ Problem: Complex inter-agent communication with trace IDs
- ☑ Solution: Custom MCPBus with async message routing

CHALLENGE 2: Multi-Format Document Processing

- ✕ Problem: Different parsing strategies for each format
- ☑ Solution: UniversalDocumentProcessor with format handlers

CHALLENGE 3: Session Isolation for Multiple Users

- ✕ Problem: Shared vector storage causing data leaks
- ☑ Solution: Session-specific ChromaDB collections

CHALLENGE 4: Real-time Performance Optimization

- ✕ Problem: Large document processing blocking UI
- ☑ Solution: AsyncIO with progress indicators & caching

FUTURE ENHANCEMENTS AND MY JOURNEY

FUTURE ROADMAP:

- Multi-Modal Support - Image and video processing
- Graph RAG Implementation - Advanced retrieval strategies
- Collaboration Features - Multi-user document sharing
- Analytics Dashboard - Usage metrics and insights
- API Endpoints - External system integrations
- Advanced Agents - Specialized domain experts

MY DEVELOPMENT JOURNEY:

- Applied ALL my existing skills in Python, AI, LLMs
- Implemented NEW technologies: LangGraph, MCP Protocol, RAG
- Created professional architecture diagrams using SVG code + Canva
- Overcame technical challenges through research and experimentation
- Built enterprise-grade solution with production-ready architecture

🎯 LOOKING FORWARD:

I have given my absolute best to this project, combining technical excellence with creative problem-solving. I'm excited about the opportunity to contribute these skills and passion to your company, bringing the same level of dedication and innovation to real-world challenges.

THANK YOU