BIG DATA ANALYTICS - PROJECT REPORT

Surf AI (LLM Powered Chatbot)

Contents

Project Title: Surf AI (LLM Powered Chatbot)

1. Team Members

1) Hammad Javed - MSDS24018
2) Musham Ahmad Malik - MSDS24049
3) Sana Ilyas - MSDS24058
4) Ume Habiba - MSDS24069

2. Problem Statement

This project aims to build a scalable AI-powered chatbot system that handles both natural language queries and user-uploaded files, delivering contextual responses by leveraging a Retrieval-Augmented Generation (RAG) pipeline. The goal is to address the challenge of contextual understanding in chatbot interactions while supporting various data formats (text, JSON, files), and to demonstrate how big data tools efficiently manage high data volume and variety.

3. High-Level Architecture Diagram

The system includes components for user input (query and file), a chatbot for processing, and distributed storage using DynamoDB (Key-Value), Chroma (Vector DB), Amazon S3 (Object Store), and Amazon RDS (Relational DB). Following is the high-level architecture diagram of the system:

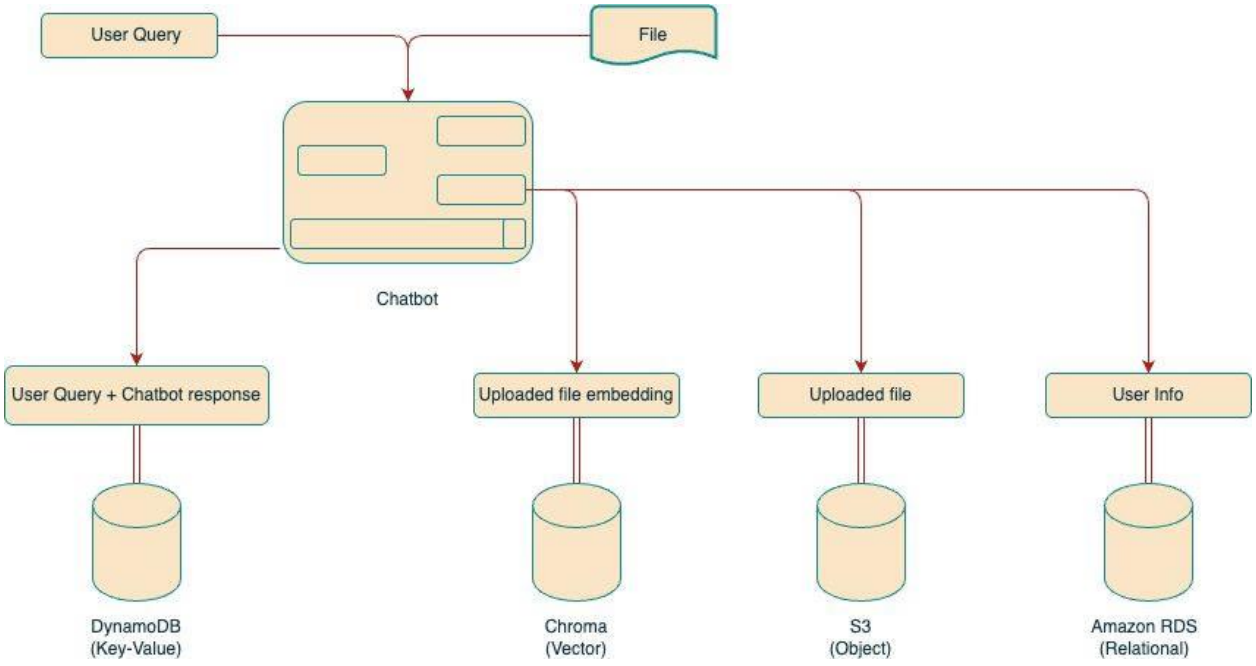

Figure 1: High-Level Architecture Diagram for Chatbot

4. Data Flow Pipeline

Following is the data flow diagram for the data flow diagram for chatbot pipeline:

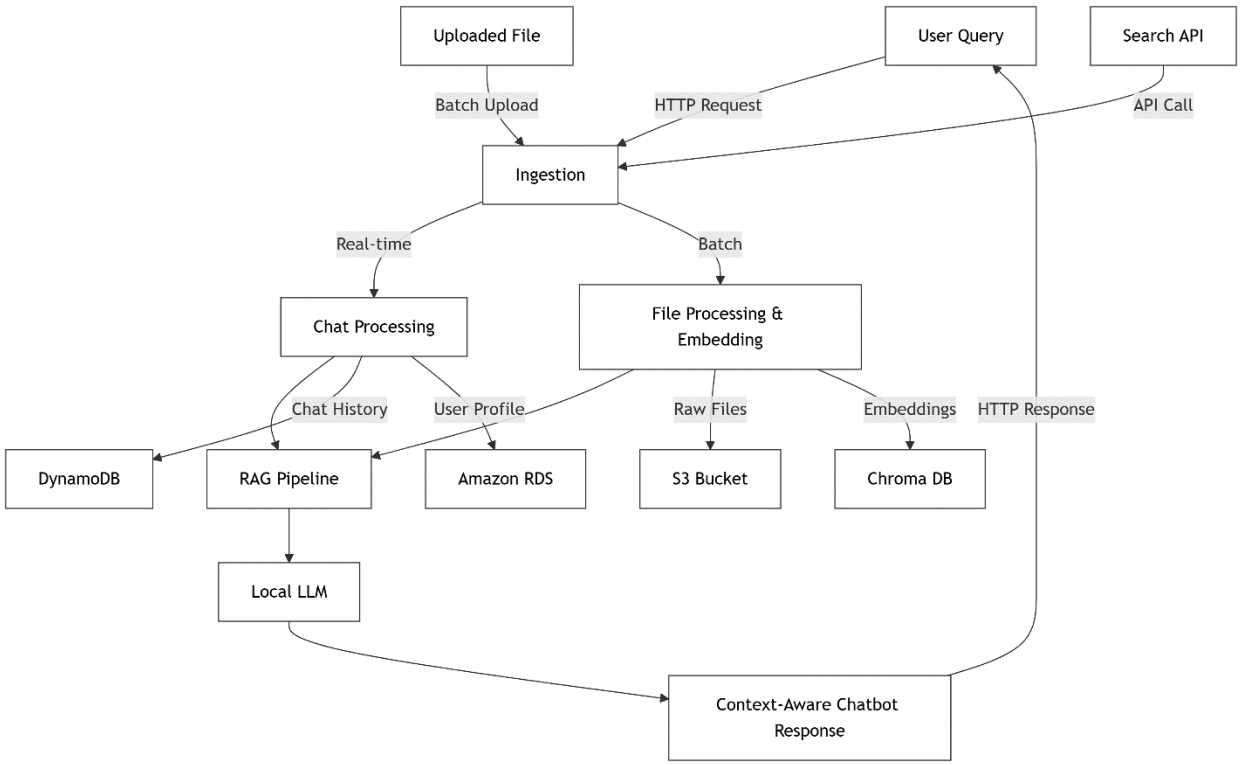

Figure 2: Data Flow Diagram for Chatbot Pipeline

1. Inputs:

- User query (text)
- Uploaded file (PDF, image, etc.)
- Search API (external content)

2. Ingestion:

- Real-time for chat
- Batch for file processing and embedding

3. Processing:

- RAG pipeline using **LangChain**
- Local LLM deployment using **Ollama**

4. Storage:

- Raw files → S3
- Embeddings → Chroma
- User chat history → DynamoDB
- User profile info → Amazon RDS

5. Output:

- Context-aware chatbot response based on embeddings + LLM

5. Tool & Technology Stack

Tool/Tech	Purpose	Justification
Ollama	Self-hosted large language models (LLMs)	Private and customizable inference layer
LangChain	Building RAG pipeline	Integrates LLMs with context from external sources
Amazon S3	Object storage for uploaded files	Reliable, scalable, and fast retrieval
Chroma	Vector DB for file/query embeddings	Efficient semantic similarity search
DynamoDB	NoSQL DB for chat history	Scalable, fast key-value access for chat logs
Amazon RDS	SQL DB for user profiles	Supports structured user metadata and relational queries

6. Storage Mapping Table

Data Type	Format	Storage System	Reason
User queries & responses	Structured (text)	DynamoDB	Fast key-value access and scalability
Uploaded file embeddings	Vectors (numeric)	Chroma	Vector-based semantic search for retrieval

Data Type	Format	Storage System	Reason
Uploaded files	Unstructured (PDF, etc.)	S3	Object storage optimized for large file access
User info	Structured (tables)	Amazon RDS	Relational structure suited for profile management