

AI Company Internal Chatbot with Role-Based Access Control (RBAC)

1. Executive Summary

This project delivers a secure, internal AI-powered chatbot designed to provide controlled access to company knowledge. By integrating Retrieval-Augmented Generation (RAG) with Role-Based Access Control (RBAC), employees can access only department-authorized information. The solution uses FastAPI, ChromaDB, and Streamlit, fully built on free and open-source technologies.

2. Project Objectives & Scope

The primary objective was to process natural language queries and retrieve department-specific information securely. Core requirements included secure authentication, strict role-based data access, a complete RAG pipeline, and zero-cost infrastructure using open-source tools.

3. System Architecture & Technology Stack

The system follows a modular architecture separating data ingestion, backend logic, and frontend presentation.

4. Technical Implementation

Phase 1: Data Preparation & Ingestion

Documents were preprocessed, chunked (300–512 tokens), and tagged with role-based metadata to ensure accurate retrieval.

Phase 2: Vector Database & Semantic Search

ChromaDB stores vector embeddings generated using SentenceTransformers. Role-based filters ensure unauthorized data is never retrieved.

Phase 3: Backend API & Security

FastAPI endpoints handle authentication using JWT tokens and orchestrate the RAG workflow securely.

Phase 4: RAG Pipeline & Mock Mode

A fallback mock mode allows the system to operate without an LLM by returning raw retrieved content, ensuring reliability and cost control.

Phase 5: Frontend Interface

Streamlit provides a clean chat interface with session management and source citations for transparency.

5. RBAC Policy

Roles such as C-Level, Finance, HR, Engineering, and Marketing have strictly enforced access

controls. Unauthorized queries return zero results, ensuring complete data isolation.

6. Challenges & Solutions

Data privacy was ensured via metadata filtering at the database level. LLM dependency was mitigated using mock mode, and context limits were handled through optimized chunking strategies.

7. Conclusion

The project meets all objectives, delivering a secure, scalable, and extensible internal knowledge chatbot. The system converts static documents into an interactive, role-secured knowledge base.