Project Documentation: Document Search Bot

1. Project Overview

Project Name: Document Search Bot

Objective: Develop a React-based frontend integrated with a Flask backend to facilitate document search and chatbot interaction.

2. Design & Approach

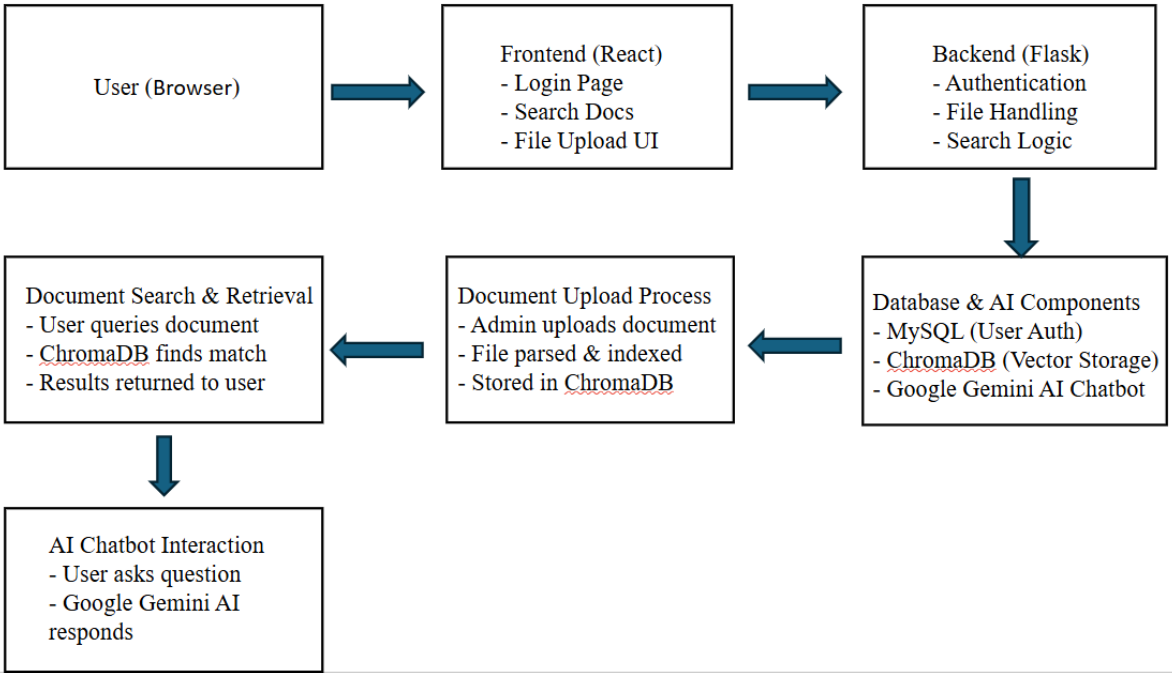
Architecture:

- The project follows a client-server architecture:
- Frontend: React.js with react-chatbot-kit for chatbot interaction.
- Backend: Flask with MySQL for authentication and ChromaDB for vector storage.
- Authentication: Role-based authentication (admin for document management, user for search and view access).
- Chatbot: Uses Google Gemini AI for intelligent responses.

3. Tech Stack

- **Frontend:** React.js, Axios, React-Router, Tailwind CSS
- **Backend:** Flask, MySQL, ChromaDB, LangChain, Ollama Embeddings
- **AI Model:** Google Gemini AI
- **Storage:** ChromaDB (Vector Store for document embeddings)

4. Project Flow Diagram



5. Features

- User Authentication & Role Management
- Users can log in using credentials stored in MySQL.
- Admins can upload and delete documents.
- Users can only search and view documents.

6. Document Processing & Search

- Files are parsed and stored in ChromaDB as vector embeddings.
- Users can search documents using text-based queries.
- Results are ranked using a custom similarity function.

7. Chatbot Integration

- React chatbot interacts with users for document-related queries.
- Uses Google Gemini AI for intelligent responses.

8. Test Cases & Results

Test Case	Description	Expected Result	Actual Result	Status
Login with valid credentials	User enters correct username/password	Redirected to dashboard	Redirected successfully	✅ Pass
Login with invalid credentials	Incorrect username/password entered	Show error message	Error message displayed	✅ Pass
Upload document (Admin)	Admin uploads PDF/DOCX file	File successfully stored	File uploaded successfully	✅ Pass
Search document	User enters a search query	Relevant results displayed	Correct results displayed	✅ Pass
Delete document (Admin)	Admin deletes a file	File removed successfully	File deleted successfully	✅ Pass

9. Deployment & Setup Instructions

Backend Setup:

- Install dependencies: pip install -r requirements.txt
- Start Flask server: python app.py

Frontend Setup:

- Install dependencies: npm install
- Start React server: npm start