**Project Description: RAG System with Gemini, FastAPI & ReactJS**

We are building a **Retrieval-Augmented Generation (RAG) system** powered by **Google Gemini** that allows users to upload documents, ask questions, and receive accurate answers.

**🔹 Key Features**

1. **File Upload & Processing**
   * Users can upload **up to 5 files** (PDF, DOCX, TXT, etc.).
   * The system will parse and index the content for semantic retrieval.
2. **Question Answering (RAG)**
   * Users can ask any question related to the uploaded documents.
   * The system retrieves relevant context from the uploaded files.
   * Gemini API generates accurate, context-aware responses.
3. **Authentication System**
   * **Google OAuth2 authentication** for user login/signup.
   * User data (uploaded files, chat history, etc.) will be **synced to their Google account**.
4. **Frontend (ReactJS)**
   * Intuitive **chat-style UI** where users can upload files and interact with the assistant.
   * File upload panel (drag & drop or select files).
   * Display of chat history with context from documents.
   * Authentication flow with Google sign-in.
5. **Backend (FastAPI - Python)**
   * Handles authentication, file storage, and retrieval pipeline.
   * Uses **vector database (e.g., Pinecone, Weaviate, or FAISS)** for efficient document retrieval.
   * Integrates with Gemini for generating answers.