EduTutor AI: Requirements Specification

1. Project Title

EduTutor AI: Personalized Learning With Generative AI And LMS Integration

2. Objectives

- Provide personalized tutoring using IBM Watsonx.
- Integrate with Google Classroom to pull class content and assignments.
- Use Pinecone vector database for smart, semantic search across study material.
- Offer a responsive frontend interface via Streamlit.

3. Key Features

AI Features:
- Generative AI chatbot for learning assistance.
- Summarization of lengthy topics.
- Automatic quiz/question generation.

LMS Integration:
- OAuth login with Google account.
- Fetch classroom materials and assignments.
- Track individual learning progress.

Semantic Search (Pinecone):
- Store and retrieve notes/content using vector similarity.
- Retrieve the most relevant answers even if the question is phrased differently.

Frontend (Streamlit):
- Student dashboard with chatbot, upload materials, view assignments, and recommendations.
- Optional teacher dashboard to upload content and view progress.

4. User Roles

Student: Chat with AI, view assignments, receive recommendations.
Teacher (optional): Upload content, monitor progress, generate quizzes.
Admin (optional): Manage users and settings.

5. Tech Stack

- AI Engine: IBM Watsonx
- Vector DB: Pinecone
- LMS: Google Classroom API
- Frontend Framework: Streamlit (Python)
- Backend Code: Python
- Auth & Config: OAuth2, dotenv
- Data Processing: sentence-transformers, NumPy, Pandas

6. Security & Privacy

- OAuth 2.0 for authentication.
- No storage of login credentials.
- Encrypted or access-controlled uploaded content.
- Environment variables for API keys.

7. Future Extensions

- Add voice support via Whisper API or Web Speech.
- Add analytics dashboard for students.
- Enable teacher content management panel.
- Multilingual support.

8. Folder Structure (suggested)

```
EduTutorAI/
├── .env                # Environment variables
├── main.py             # Streamlit UI entry point
```

```
├── backend/
│   ├── ai_engine.py        # Watsonx interaction
│   ├── pinecone_utils.py   # Pinecone DB handlers
│   └── classroom_sync.py   # Google API interactions
├── data/                   # Uploaded notes or course materials
├── models/                 # Embedding/LLM models
├── utils/                  # Helper functions
└── requirements.txt        # Python dependencies
```