🟦 Tech Scoping Write-Up: WhimsyBot – For Es Magico AI Studio

💡 Framework & Tools Used

- LangChain: For managing RAG pipelines and prompt engineering.
- Together AI: Hosted LLM backend with support for large, capable open-source models.
- HuggingFace Embeddings: sentence-transformers/all-MiniLM-L6-v2 for embedding user queries and context.
- Chroma: As the vector database to store and retrieve document chunks.
- gTTS: For text-to-speech generation.
- Deep Translator: For automatic translation and language detection.
- Streamlit: For building an interactive and clean chatbot UI.
- SpeechRecognition + PyAudio: For handling microphone input and converting speech to text.

⚙️ Flow of the Application

1. Document Loading & Indexing
   - Three PDFs (Alice in Wonderland, Gulliver's Travels, Arabian Nights) are loaded using PyPDFLoader.
   - Content is split into manageable chunks with overlap for better semantic retention.
   - Embeddings are created and stored in a persistent Chroma DB directory.
2. Chat Interface
   - User can type or speak a query via the UI.
   - Speech input is recognized using the microphone and Google Web Speech API.
3. Multilingual Support
   - Input is auto-detected and translated to English before querying.
   - Final answer is translated back to the original input language, if necessary.
4. LLM Querying
   - LangChain's RetrievalQA chain with Together AI's LLM retrieves relevant story chunks.
   - Prompt enforces a funny, creative tone and avoids meta-commentary.
   - Custom fallback message is used if the model cannot answer.
5. Output Generation
   - Cleaned and context-rich answers are shown.
   - Generated answer is converted into audio (gTTS) and played.
   - If a valid answer is returned (not fallback), an image related to the query is generated using Together's image API.

🔧 Key Design Decisions

- RAG (Retrieval-Augmented Generation) was chosen to restrict the LLM's knowledge base to the given PDFs, ensuring accuracy.
- Prompt Engineering was used to ensure a humorous storytelling tone.
- Multilingual + Audio I/O + Image Generation was added as bonus features to enhance accessibility and engagement.
- Modular Code Structure allows easy replacement of the LLM, embedding model, or vector database.

🔁 Extensibility

- Swap out the LLM or embedding model with a single line of code.
- Add more PDFs or story domains easily with PyPDFLoader.
- Additional languages and audio synthesis models can be plugged in without architectural changes.