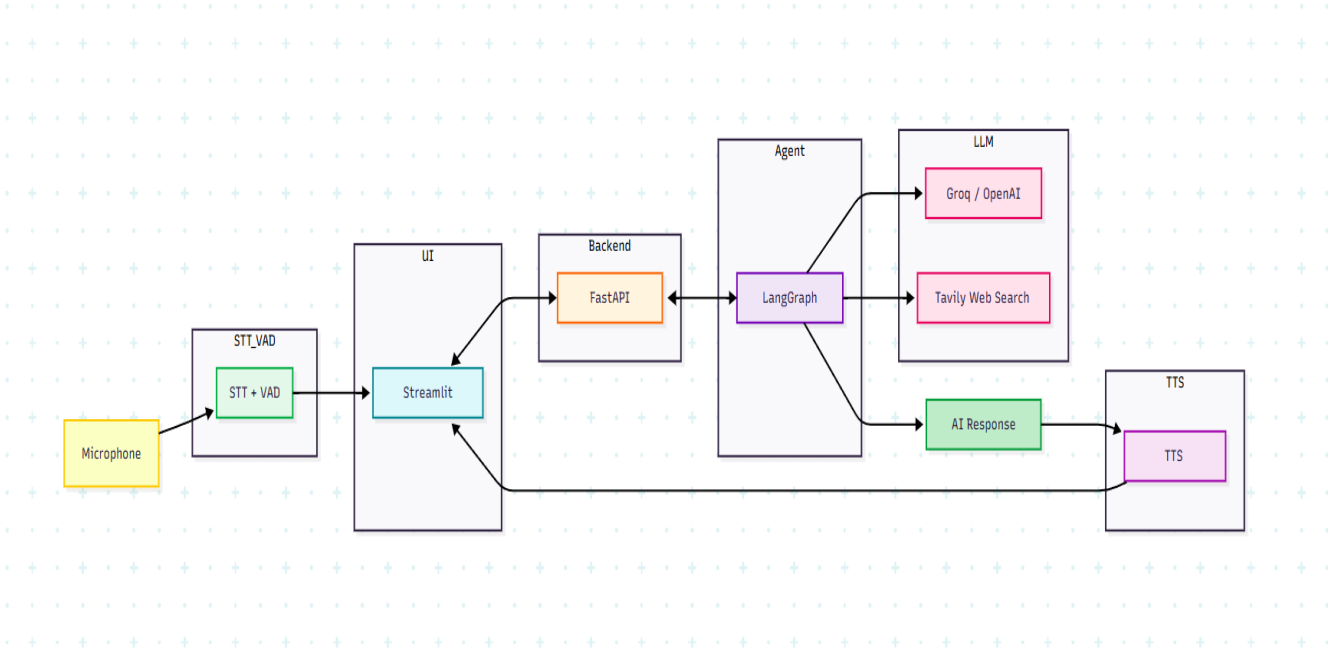
□ System Architecture Document: EDU-GEN AI Assistant

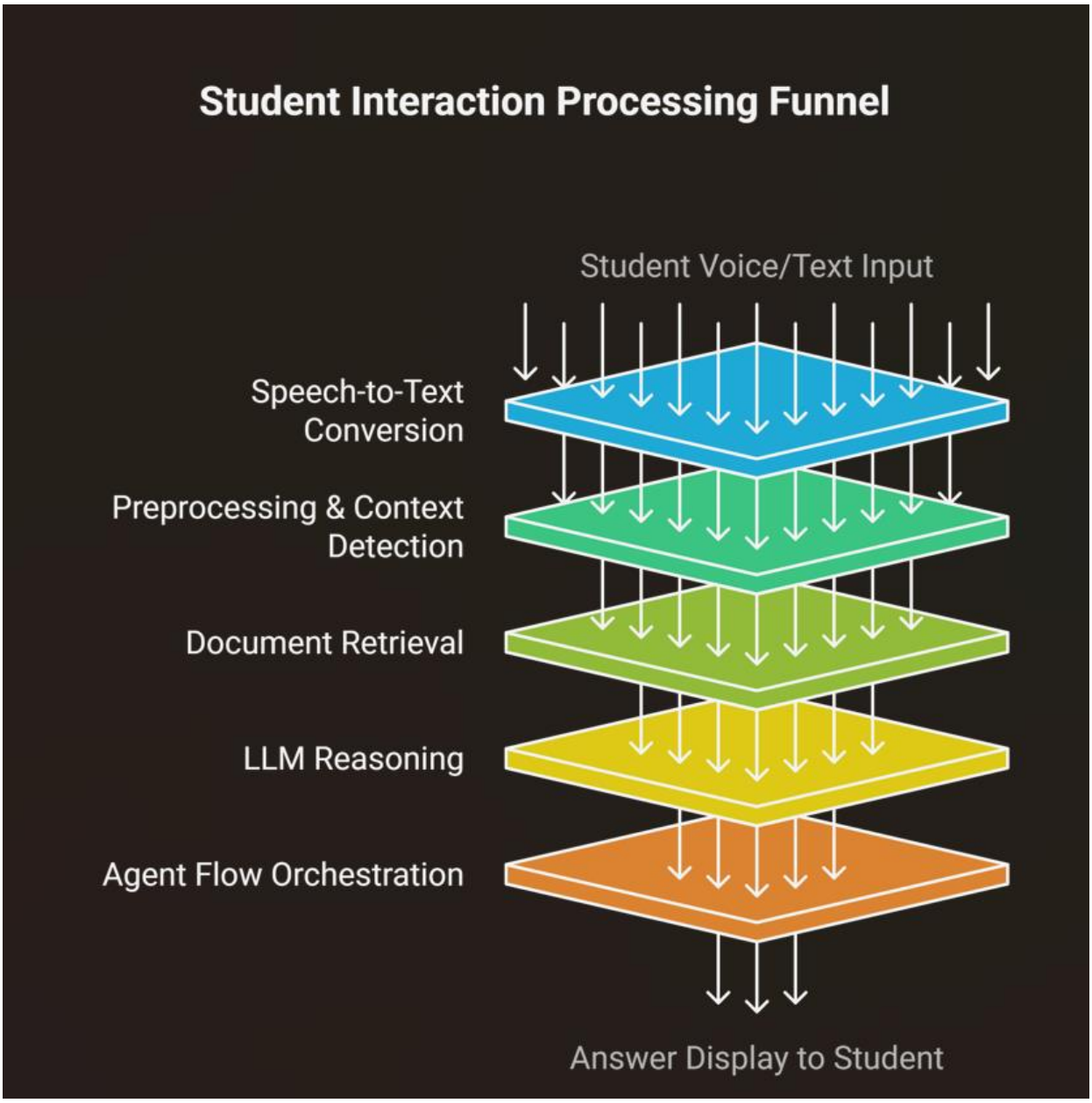
□ Overview

The **EDU-GEN AI Assistant** is a voice-based, multilingual AI chatbot system that enables real-time communication with users. It integrates various AI tools and modules including LangGraph (for agent orchestration), LLM APIs (Groq, OpenAI), STT/VAD (Speech Recognition), and TTS (ElevenLabs).

This architecture ensures modularity, scalability, and responsiveness suitable for multi-user educational and conversational use cases.

□ High-Level Architecture





□ Core Components

1. Frontend (``)

- Built using **Streamlit**
- UI allows:
    - Text or voice input

    - Language and model selection
    - TTS output
    - Quiz generation & download
- Maintains session state per user for chat history, language, and last query.

2. Voice Agent (``)

- CLI-based voice interface
- Uses `speech_recognition` for STT and ambient noise filtering
- Sends queries to FastAPI backend
- Uses ElevenLabs TTS for spoken response

3. Backend API (``)

- Built with **FastAPI**
- Exposes `/chat` and `/generate_quiz` endpoints
- Validates input using Pydantic models
- Adds language-specific system prompts dynamically
- Generates quizzes and PDFs using ReportLab

4. AI Agent Core (``)

- Manages LangGraph REACT agent
- Dynamically chooses between OpenAI and Groq LLMs
- Integrates **Tavily Search Tool** if `allow_search=True`
- Handles:
    - Memory state (via messages array)
    - Tool selection
    - Response parsing from AIMessage

□ Data Flow Diagram (Voice Path)

1. **Voice Input (Microphone)**
2. □ Transcribed using `speech_recognition` → Text
3. □ Sent to `/chat` endpoint with model config
4. □ LangGraph agent handles logic & memory
5. □ AI response parsed → returned to frontend

6. □ Converted to speech using ElevenLabs
7. □ Played back to user

□ Multi-User Handling

- `Streamlit's session_state` keeps track of:
  - Chat history
  - Selected language, voice, and model
  - TTS audio path
- No persistent storage or DB is used; state is per session

□ System Design Deliverables

Include the following in your GitHub repo:

- `system_design.pdf` or `system_design.png`:
  - A visual diagram (similar to the ASCII above)
  - Explain component interaction
  - Highlight voice & text routes
- This Markdown/README file for detailed explanation

✓Key Technologies

- **LangGraph + LangChain**: REACT agent for stateful reasoning
- **Groq / OpenAI GPT**: LLM inference
- **Tavily**: External search tool integration
- **ElevenLabs**: Voice synthesis (TTS)
- **SpeechRecognition (Google)**: Voice transcription
- **FastAPI**: REST API service
- **Streamlit**: Web frontend
- **ReportLab**: Quiz PDF generation

□ Conclusion

This architecture is modular, flexible, and voice-first — making it ideal for educational chatbots, multilingual tutors, or assistive agents.

Let me know if you want me to generate a **visual diagram (PNG)** from this architecture!