

# EduVox: AI Voice Tutor

"Your Voice-Enabled Learning Companion"

**Author:** Selvin Tuscano

**Date:** April 2025

## Documentation Report

---

### Table of Contents

1. [Abstract](#)
  2. [Problem Statement](#)
  3. [System Architecture Diagram](#)
  4. [Implementation Details](#)
  5. [Performance Metrics](#)
  6. [Challenges and Solutions](#)
  7. [Future Improvements](#)
  8. [Ethical Considerations](#)
  9. [Streamlit App Screenshots](#)
  10. [Conclusion](#)
  11. [References](#)
  12. [License](#)
- 

### Project Resources

**GitHub Repository:** [https://github.com/selvintuscano/EduVox\\_AI\\_VoiceTutor](https://github.com/selvintuscano/EduVox_AI_VoiceTutor)

#### **Presentations:**

1. PowerPoint Project Presentation:  
[https://drive.google.com/file/d/1HXs68IqLxASY7ypSj7peMQnk0Ix\\_9FpB/view?usp=drivesdk](https://drive.google.com/file/d/1HXs68IqLxASY7ypSj7peMQnk0Ix_9FpB/view?usp=drivesdk)
2. Code and Streamlit App Walkthrough:  
[https://drive.google.com/file/d/1H-iwAAAZsFp0\\_RJ\\_pFPG0nw-ldDV5r78/view?usp=drivesdk](https://drive.google.com/file/d/1H-iwAAAZsFp0_RJ_pFPG0nw-ldDV5r78/view?usp=drivesdk)  
**Note:** Haven't deployed Streamlit App as it contains OpenAI API key and eleven labs keys. Have provided screenshots of Streamlit App below

# 1. Abstract

EduVox is an innovative AI-powered voice tutoring system designed to transform traditional learning methodologies into interactive, personalized educational experiences. By leveraging advanced technologies in natural language processing, speech recognition, and retrieval-augmented generation, EduVox creates a dynamic learning environment that responds intelligently to user queries through both voice and text interfaces. The system processes educational materials into a vectorized knowledge base, enabling contextually relevant responses drawn from specific learning materials rather than generic information. With features including voice interaction, personalized responses, content summarization, quiz generation, multilingual support, and PDF export functionality, EduVox represents a comprehensive solution to the challenges of modern education. This documentation outlines the system architecture, implementation details, performance considerations, challenges overcome, and future directions for this voice-enabled learning companion.

## 2. Problem Statement

Traditional learning methods often present significant limitations in today's educational landscape:

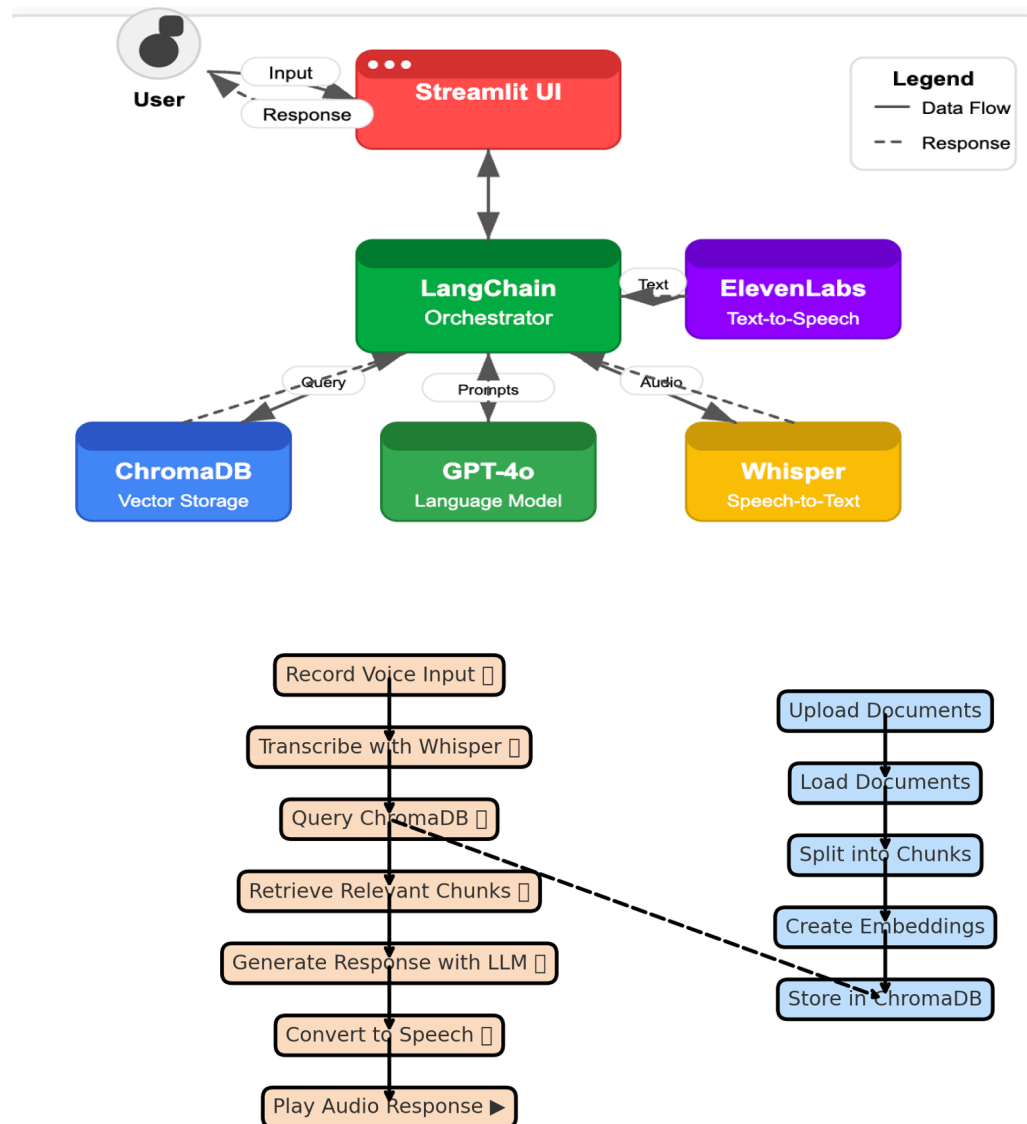
1. **Lack of Personalization:** Standard educational resources and approaches fail to adapt to individual learning needs, styles, and paces, resulting in sub-optimal learning experiences for many students.
2. **Limited Interactivity:** Traditional textbooks and static content provide few opportunities for active engagement, question-asking, or immediate feedback, which are crucial components of effective learning.
3. **Accessibility Barriers:** Existing educational resources may be challenging to access or use for learners with different abilities, language proficiencies, or technological limitations.
4. **Contextual Relevance:** Students frequently struggle to find precise, contextually relevant information when studying complex topics, leading to inefficient learning processes.
5. **Engagement Challenges:** Maintaining student engagement with conventional learning methods is increasingly difficult in an era of interactive digital experiences.
6. **Absence of Comprehensive Solution:** Prior to EduVox, there was no integrated system that combined voice interaction, artificial intelligence, and dynamic content generation to provide on-demand, adaptive educational support.

EduVox addresses these challenges by creating an accessible, interactive, voice-enabled AI tutor that leverages advanced technologies to provide personalized, context-aware educational assistance across various learning scenarios.

### 3. System Architecture Diagram

EduVox employs a sophisticated architecture that integrates multiple AI technologies to deliver a seamless learning experience. The system is built around a central LangChain orchestrator that coordinates interaction between various components.

#### Core Architecture Overview



## Data Flow

1. **User Interaction:** Users interact through voice or text via the Streamlit UI
  2. **Knowledge Base Setup:** Documents are processed and stored as vector embeddings in ChromaDB
  3. **Query Processing:**
    - Voice inputs are transcribed using Whisper
    - Queries are processed by LangChain to retrieve relevant information from ChromaDB
    - GPT-4o generates contextually relevant responses
  4. **Response Delivery:**
    - Text responses are displayed on the UI
    - ElevenLabs converts responses to natural-sounding speech
    - PDF export functionality provides downloadable content
- 

## 4. Implementation Details

### Key Components

#### 1. Document Processing System

The DocumentProcessor class handles the ingestion and preparation of learning materials:

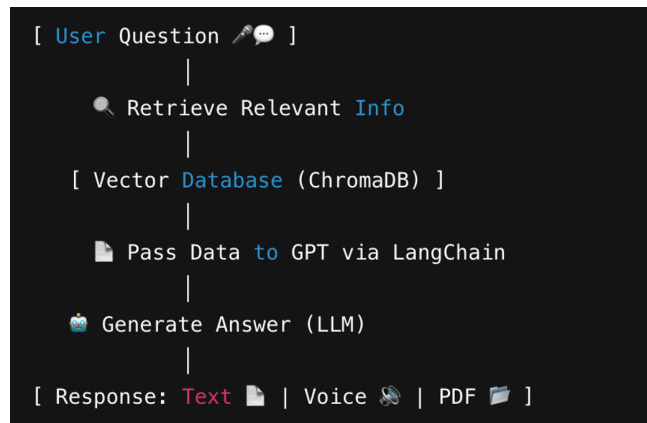
- Supports multiple file formats (PDF, TXT, MD)
- Uses recursive character text splitting with configurable chunk size and overlap
- Creates vector embeddings using OpenAI's embedding model
- Stores processed document chunks in ChromaDB for efficient retrieval

For dataset you can upload any document of choice

#### 2. Retrieval-Augmented Generation (RAG)

EduVox implements a RAG architecture to provide accurate, context-aware responses:

- User queries are converted to vector embeddings
- Similar document chunks are retrieved from ChromaDB using semantic search
- Retrieved content is passed to the language model along with the query
- LLM generates responses based on both its pre-trained knowledge and retrieved context

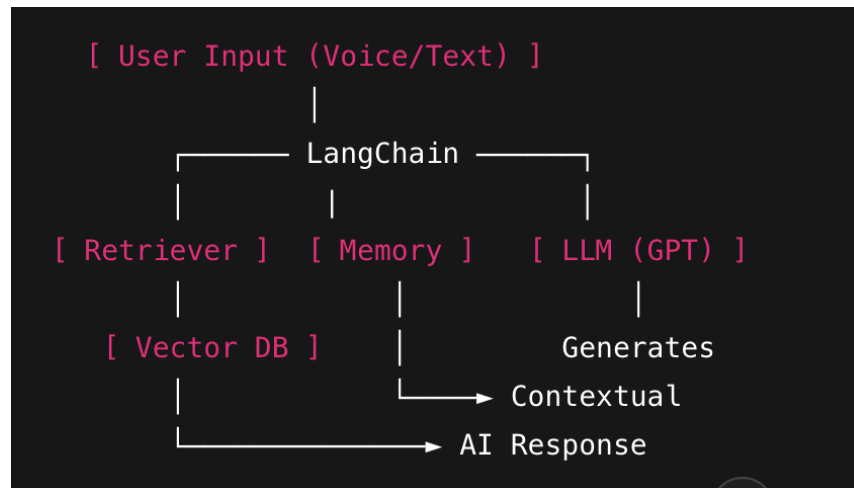


### 3. LangChain Framework Integration

LangChain serves as the orchestration layer for EduVox, connecting various components into a coherent system. Key LangChain implementations include:

1. **Conversation Chain Management:**
  - ConversationalRetrievalChain combines the retrieval system with language model generation
  - Maintains context through ConversationBufferMemory to ensure coherent dialogue across multiple interactions
  - Handles query reformulation and optimization for better retrieval results
2. **Document Processing Pipeline:**
  - Leverages LangChain's document loaders (PyPDFLoader, DirectoryLoader, TextLoader, UnstructuredMarkdownLoader) to support multiple file formats
  - Uses RecursiveCharacterTextSplitter with configurable chunk size (1000) and overlap (200) parameters
  - Creates a standardized document format that maintains metadata across processing steps
3. **Retrieval System:**
  - Implements LangChain's Retriever interface for semantic search
  - Configures retrieval parameters for optimal performance (e.g., similarity threshold, number of documents)
  - Supports filtering and reranking of retrieval results
4. **Prompt Management:**
  - Utilizes customized prompt templates for different response modes (default, summarize, quiz, simplify)
  - Injects retrieved context into prompts for the language model
  - Manages prompt construction for multilingual support

LangChain functions as the "glue" of EduVox, receiving user input, retrieving relevant data from ChromaDB, passing context to GPT, managing conversation history, and returning contextually appropriate responses to the user.



## 4. ChromaDB Vector Database

ChromaDB provides the vector storage and retrieval capabilities essential to EduVox's RAG architecture:

- 1. Vector Representation:**
  - Stores document chunks as vector embeddings (created using OpenAI's embedding model)
  - Maintains metadata alongside embeddings for source tracking and context preservation
  - Persists embeddings to disk for rapid application restart and session continuity
- 2. Efficient Semantic Search:**
  - Implements nearest-neighbor search algorithms for quick similarity matching
  - Supports configurable search parameters (top k results, similarity metrics)
  - Performs vector comparisons to find contextually relevant information based on semantic meaning rather than keyword matching
- 3. Vector Database Operations:**
  - Provides an API for adding, querying, and managing document embeddings
  - Supports batch operations for efficient document processing
  - Enables persistence mechanisms for long-term storage
- 4. Integration with Document Processing:**
  - The DocumentProcessor class interfaces directly with ChromaDB
  - `create_vector_store` method initializes and populates the database
  - Vector store is persisted to a specified directory ("knowledge\_base") for reuse across sessions

## 5. Voice AI Integration

Two-way voice communication capabilities:

- **Speech-to-Text:** Whisper model transcribes user's voice queries with high accuracy

- **Text-to-Speech:** ElevenLabs multilingual voice synthesis creates natural-sounding responses
- Multiple voice options (Rachel, Domi, Bella, Antoni, etc.) for personalized experience

## 6. Learning Modes

Multiple response generation strategies:

- **Default mode:** Straightforward answers to queries
- **Summarize:** Condensed overviews of complex topics
- **Quiz:** Automatic generation of multiple-choice questions for assessment
- **Simplify:** Explanations using simplified language for better understanding

## 7. Multilingual Support

- Translation capabilities for cross-language learning
- Currently supports English and Hindi
- Extensible framework for adding more languages

## Technologies Used

- **Frontend:** Streamlit
- **Language Models:** GPT-4o-mini (OpenAI)
- **Embedding Model:** OpenAI Embeddings
- **Vector Database:** Chroma
- **Speech-to-Text:** Whisper
- **Text-to-Speech:** ElevenLabs
- **Audio Processing:** sounddevice, soundfile
- **PDF Generation:** FPDF
- **Framework:** LangChain for orchestration
- **Document Processing:** PyPDFLoader, DirectoryLoader, TextLoader, UnstructuredMarkdownLoader

## Code Architecture

The codebase is organized into modular classes:

- `DocumentProcessor`: Handles document ingestion and vector store creation
  - `AIVoiceTutor`: Core class managing interactions, voice processing, and response generation
  - UI components separated into distinct functions for maintainability
-

## 5. Performance Metrics

EduVox demonstrates several performance characteristics:

### Response Time

- Voice recording: Configurable duration (1-10 seconds)
- Transcription: Near real-time processing with Whisper
- Query processing: Retrieval + generation typically completes within seconds
- Voice synthesis: Generated in real-time by ElevenLabs

### Accuracy

- Whisper provides high-quality transcription across different accents and speech patterns
- Vector search retrieves contextually relevant information based on semantic similarity
- GPT-4o-mini delivers coherent and accurate responses based on retrieved context

### Scalability

- ChromaDB enables efficient storage and retrieval of document embeddings
- The system architecture supports growing document collections
- Modular design allows for component upgrades without full system redesign

### User Experience

- Multi-modal input/output (voice/text)
  - Interactive quiz generation for assessment
  - PDF export for offline reference
  - Customizable voice options for personalized experience
-



## 6. Challenges and Solutions

Challenge	Solution Implemented
API Rate Limits	Optimized API calls with batching techniques and implemented retries for critical requests
Audio Processing Delays	Used efficient libraries (Whisper, sounddevice) and optimized recording durations to balance performance and accuracy
Large Document Handling	Implemented smart chunking with LangChain's Text Splitter and leveraged ChromaDB for scalable vector storage
Maintaining Context	Integrated ConversationBufferMemory in LangChain to ensure coherent, contextual responses across multiple queries
Multilingual Voice Output	Utilized ElevenLabs multilingual TTS and fallback mechanisms for unsupported cases
Temporary File Management	Automated cleanup using Python's tempfile and os modules to manage storage efficiently

---

## 7. Future Improvements

Several areas for enhancement have been identified:

### 1. Expanded Language Support

- Add support for additional languages beyond English and Hindi
- Improve translation quality for technical and domain-specific terminology
- Implement language detection for automatic response language selection

### 2. Enhanced UI Design

- Create a more intuitive and accessible interface
- Develop mobile-responsive design for cross-device usage
- Add visualization components for data and concept explanation

### 3. Adaptive Learning Features

- Implement user progress tracking
- Develop personalized learning paths based on user performance
- Create spaced repetition algorithms for improved retention

### 4. Mobile Deployment

- Package the application for mobile platforms
- Optimize performance for resource-constrained environments
- Implement offline capabilities for learning without continuous internet access

## **5. Advanced RAG Techniques**

- Implement hybrid search (keyword + semantic)
  - Add re-ranking algorithms for more relevant retrieval
  - Explore multi-step reasoning for complex queries
- 

# **8. Ethical Considerations**

## **Privacy and Data Security**

- User audio recordings should be processed locally where possible
- Clear data retention policies should be established
- Implement encryption for sensitive data

## **Accessibility**

- Voice interface provides access for users with certain disabilities
- Multiple input modalities (voice/text) increase accessibility
- Simplification mode helps users with different comprehension levels

## **Educational Integrity**

- Quiz generation should align with educational standards
- System should encourage critical thinking, not just information retrieval
- Clear attribution of sources for educational content

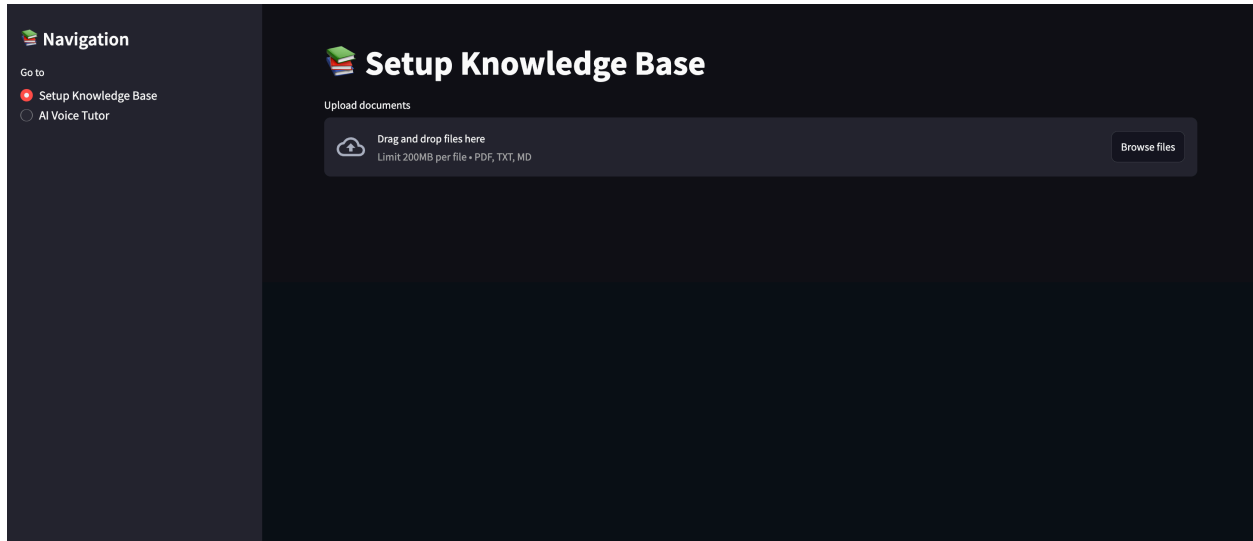
## **AI Transparency**

- Users should understand when they are interacting with AI
- Limitations of the system should be clearly communicated
- Mechanism for human review of AI-generated content when necessary

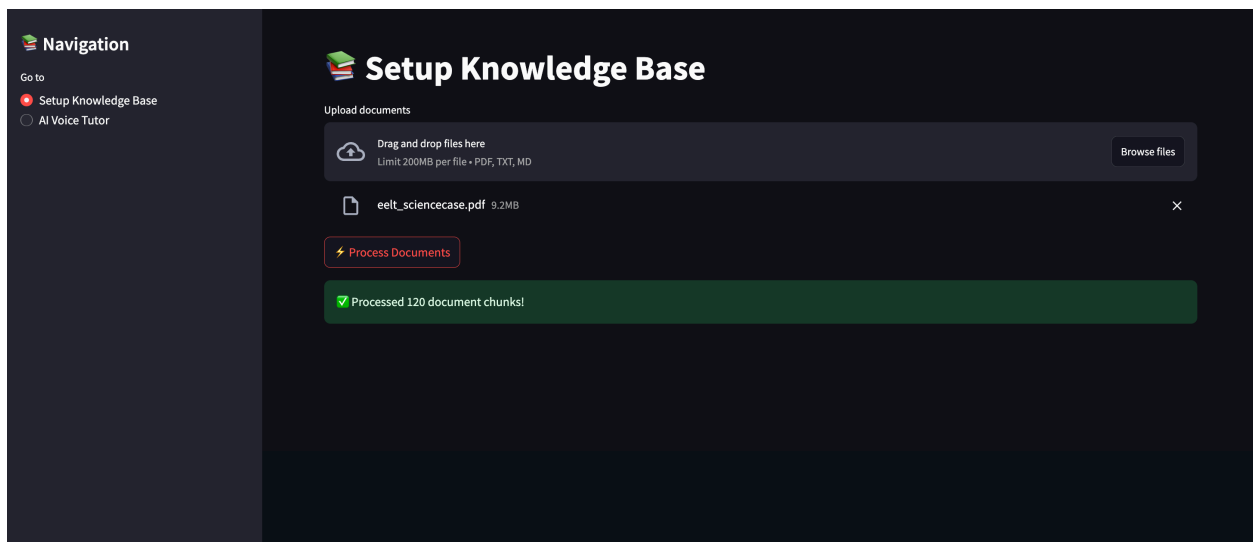
## **Content Moderation**

- Prevent generation of harmful, misleading, or inappropriate content
  - Implement content filters appropriate for educational contexts
  - Allow teacher/administrator oversight for educational deployment
-

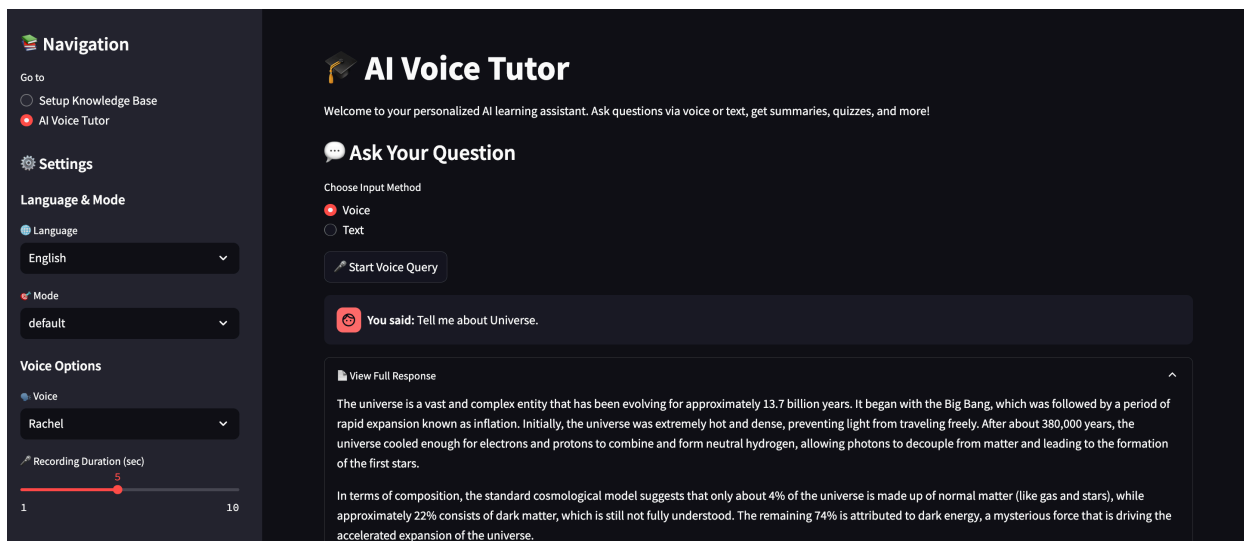
## 9. Streamlit App Screenshots



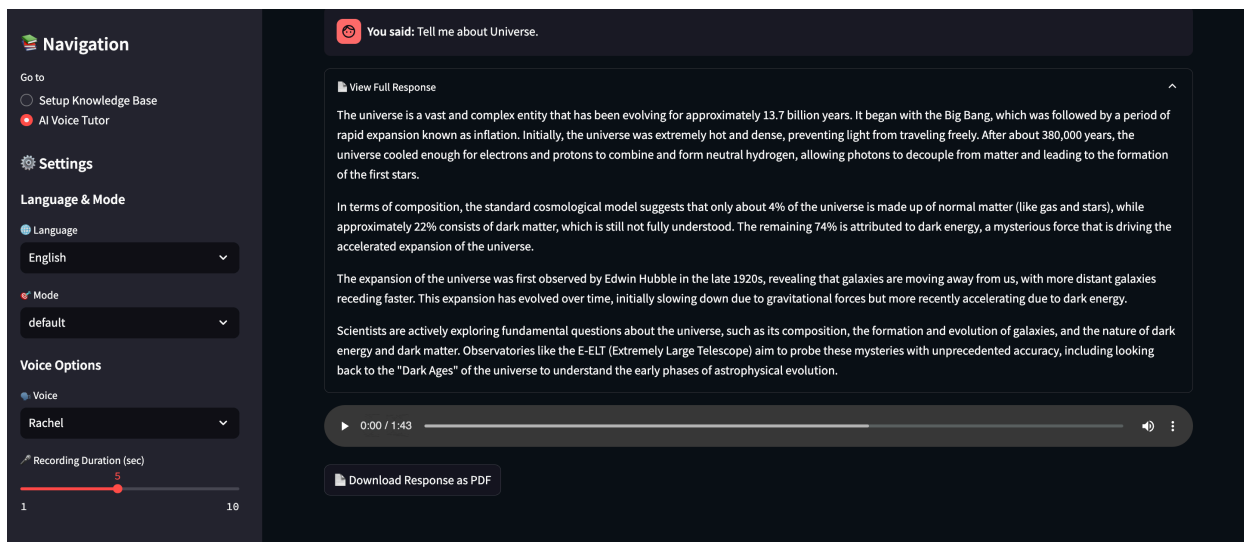
a **Knowledge Base Setup** screen where users can upload documents (PDF, TXT, MD formats) and process them into searchable chunks, as indicated by the successful processing of 120 document chunks.



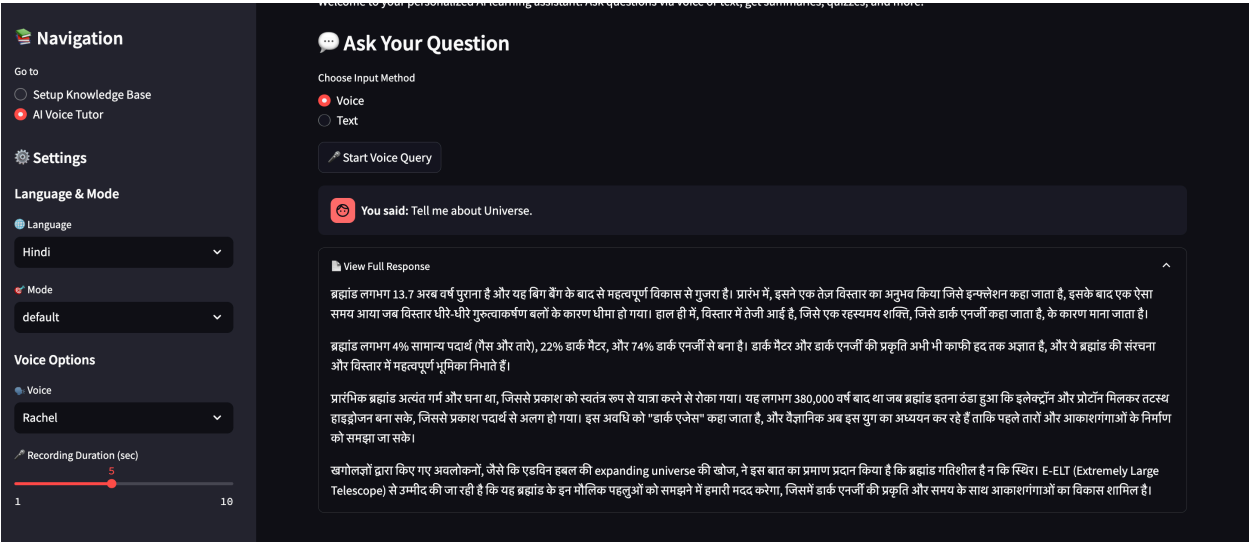
This Streamlit app screen showcases the **Tutor**, where users can ask educational questions via voice or text, customize language and voice settings, and receive detailed AI-generated responses, as seen in the example about the universe.



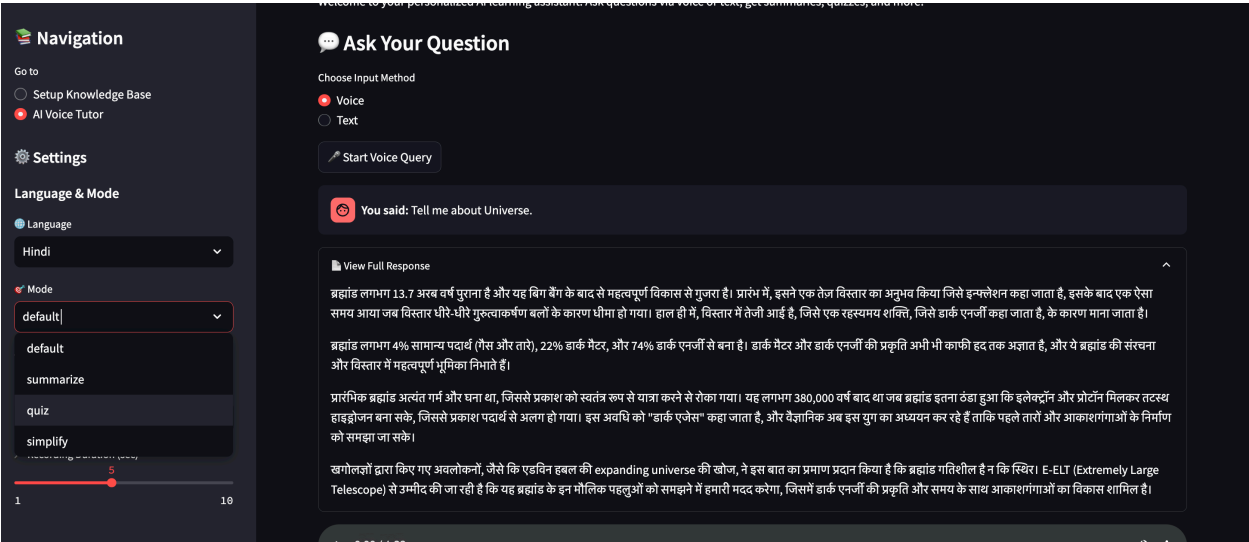
This screen displays the **detailed response view**, where the user receives an in-depth explanation, audio playback of the AI's answer, and an option to download the response as a PDF for future reference.



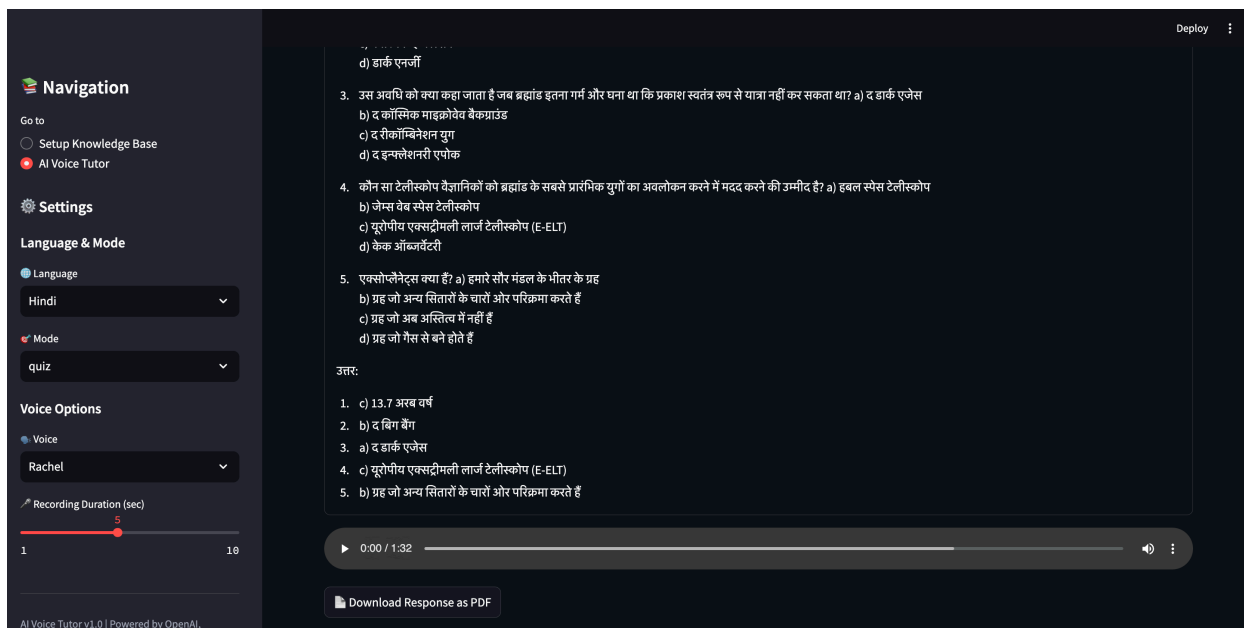
This screen highlights the **multilingual capability (Hindi)** of the Tutor, where the user receives a comprehensive AI-generated response in Hindi, demonstrating support for diverse languages within voice-based queries.



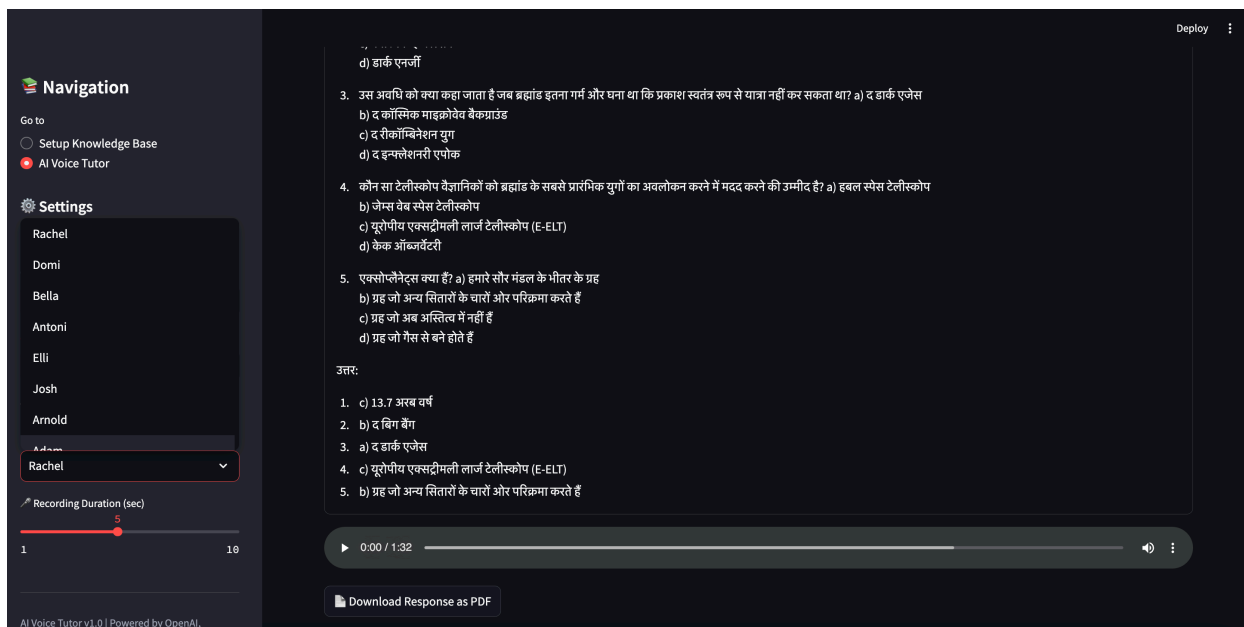
This screen demonstrates the **mode selection feature** of the AI Voice Tutor, allowing users to choose between different response modes like default, summarize, quiz, or simplify, enhancing personalized learning experiences in Hindi.



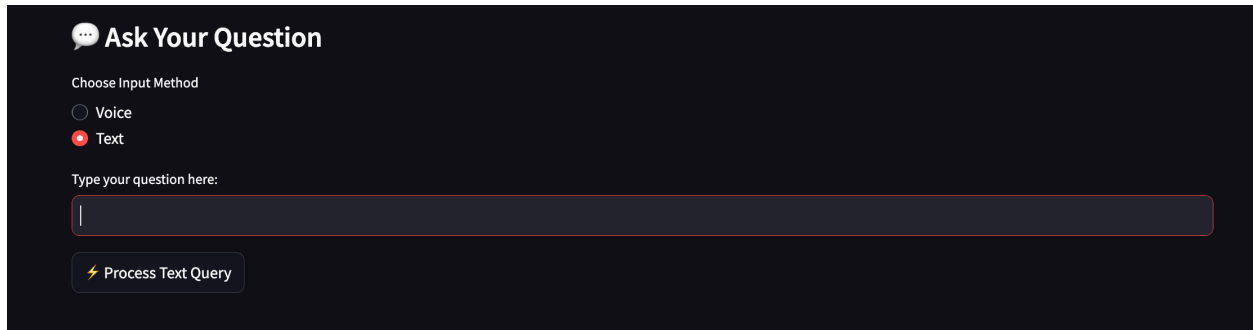
This screen showcases the **quiz generation mode** of the AI Voice Tutor, where the app automatically creates multiple-choice questions and answers in Hindi based on the user's query, along with audio playback and PDF download options for interactive learning.



This screen highlights the **voice customization feature** of the AI Voice Tutor, where users can select from various voice options (like Rachel, Domi, Bella, etc.) for audio responses, enhancing personalization.



This screen displays the **text input option** of the Tutor, allowing users to type their questions instead of using voice, providing flexibility in how learners interact with the AI for queries and educational assistance.



The screenshot shows a dark-themed interface titled "Ask Your Question" with a speech bubble icon. Below the title, it says "Choose Input Method" with two radio button options: "Voice" (unselected) and "Text" (selected, indicated by a red dot). Below this, it says "Type your question here:" followed by a long, empty text input field. At the bottom, there is a button labeled "Process Text Query" with a lightning bolt icon.

---

## 10. Conclusion

EduVox represents a significant advancement in AI-powered educational technology, bridging the gap between traditional learning methods and the personalized, interactive experiences that modern learners require. By leveraging cutting-edge technologies in natural language processing, speech recognition, and vector databases, EduVox creates a learning companion that is accessible, responsive, and adaptable to individual needs.

The system's architecture demonstrates how multiple AI technologies can be orchestrated through a unified framework to deliver a cohesive user experience. The implementation of Retrieval-Augmented Generation ensures that responses are grounded in factual information rather than hallucinations, making EduVox a reliable educational tool.

The challenges encountered during development highlight the practical considerations of deploying complex AI systems, while the solutions implemented show creative approaches to overcoming these obstacles. The identified future improvements provide a clear roadmap for evolving EduVox into an even more powerful learning platform.

As AI continues to transform education, tools like EduVox will play an increasingly important role in making knowledge more accessible and learning more engaging. However, as the ethical considerations section emphasizes, this transformation must be guided by principles that ensure privacy, accessibility, educational integrity, and transparency.

EduVox stands as a testament to how thoughtful application of AI can enhance human capabilities rather than replace them, particularly in the critical domain of education.

---

## 11. References

1. OpenAI. (2022). "Whisper: Robust Speech Recognition via Large-Scale Weak Supervision." <https://openai.com/research/whisper>
  2. ElevenLabs. (2023). "Speech Synthesis Documentation." <https://elevenlabs.io/docs>
  3. LangChain. (2023). "LangChain: Documentation." [https://python.langchain.com/docs/get\\_started/introduction](https://python.langchain.com/docs/get_started/introduction)
  4. Troya Sharma et al. (2022). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." Advances in Neural Information Processing Systems.
  5. Chroma. (2023). "ChromaDB: A database for building AI applications with embeddings." <https://www.trychroma.com/>
  6. Karpathy, A. (2022). "Building LLM applications for production." <https://karpathy.ai/zero-to-llm.pdf>
  7. Streamlit. (2023). "Streamlit Documentation." <https://docs.streamlit.io/>
  8. Brown, T. B., et al. (2020). "Language Models are Few-Shot Learners." Advances in Neural Information Processing Systems.
  9. Devlin, J., et al. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805.
  10. Lewis, P., et al. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." arXiv preprint arXiv:2005.11401.
- 

## 12. License

Copyright 2025 Selvin Tuscano

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.