Echo-Verse

EchoVerse is an AI-powered audiobook creation tool that converts user-provided text into expressive audio using IBM Granite LLM and IBM Watson Text-to-Speech (TTS) for narration. The system is built with Python and Streamlit, featuring a simple, interactive UI that supports text input, file uploads, tone/voice customization, real-time playback, and MP3 downloads.

Key Features:

- **Input Options**: Paste text or upload .txt files.
- **Audio Output**: Rewritten text converted into high-quality MP3 audio, playable and downloadable.
- **Dual Display**: Original vs. rewritten text shown side by side for comparison.
- **Session History**: “Past Narrations” panel tracks all outputs within a session for replay/download.
- **No Login Required**: Temporary session state, cleared after exit.

Technical Architecture:

1. **Input Layer** – Textbox or file upload handled by Streamlit.
2. **Text-to-Speech** – IBM TTS API generates narration in real time.
3. **Session State** – Tracks generated outputs for the active session.
4. **UI Layer** – Streamlit interface with dropdowns, columns, audio player, and styled background.
5. **Security** – API keys stored securely in .env files.

Development Workflow:

- **Milestone 1**: Integrated IBM Granite.
- **Milestone 2**: Integrated IBM TTS with voice control & MP3 playback or download.

- **Milestone 3**: Designed Streamlit UI with text or file input, dual display, and playback.
- **Milestone 4**: Implemented session state for narration history.

Outputs:

- Fast end-to-end pipeline (6–8 seconds for typical input).
- Stable audio generation across voices, with occasional truncation on long texts.
- Robust error handling for invalid inputs, large files, and API timeouts.
- Polished, user-friendly interface with themed visuals.

AI TOOLS USED:

1. Hugging Face

- **What it is**:
  Hugging Face is an open-source AI platform and model hub where thousands of pre-trained models (NLP, vision, speech, etc.) are shared. It also provides APIs and libraries to easily integrate models into applications.

- **Why we used it**:
  We used Hugging Face to connect with IBM Granite speech to text and handle text processing efficiently. It simplifies model loading, prompt chaining, and API interaction, making it easier to focus on building the audiobook pipeline rather than training models from scratch.

2. Streamlit

- **What it is**:
  Streamlit is a Python-based framework to quickly build interactive web apps without needing frontend coding (HTML, CSS, JS).

- **Why we used it**:
  We used it to create an easy-to-use interface where users can paste or upload text, select tone or voice, see original and rewritten text, and play or download audio.

3. API Key

- **What it is**:
  An API key is a secure code that authenticates your app when connecting to external services (like IBM Granite).

- **Why we used it**:
  We needed an API key to securely access IBM Granite and Text-to-Speech services. This ensures only authorized requests are processed and keeps our app safe.

4. IBM Granite Model

- **What it is**:
  IBM's Granite is a family of large language models, designed for text generation, summarization, rewriting, and etc.

- **Why we used it**:
  We used Granite for tone-specific text rewriting. Given user input and tone selection (neutral, suspenseful, inspiring,humorous and formal). This makes the narration more expressive and suitable for audiobook creation.

Conclusion:

EchoVerse delivers a complete, low-friction workflow for audiobook creation. It meets goals of accessibility, content repurposing, and creative flexibility by blending LLM-powered tone rewriting with high-quality voice synthesis. While token limits affect very long inputs, the system remains reliable and showcases the potential of generative AI for expressive, accessible content transformation.