PDF to Audiobook Converter

1. Introduction

Reading long documents or academic materials can often be time-consuming and tiring. To make reading more accessible and convenient, the PDF to Audiobook Converter was developed. This application converts text from PDF files into speech, allowing users to listen to documents instead of reading them. It is particularly useful for multitasking, visually impaired users, and students who prefer auditory learning.

2. Abstract

The project focuses on transforming a text-based PDF document into an audiobook using Python. The system extracts text from PDF pages and reads it aloud using the Text-to-Speech (TTS) engine. A user-friendly graphical interface built with Tkinter enables users to upload a PDF, control playback (Play, Pause, Stop), and adjust parameters like voice type, rate, and volume. The application employs multi-threading to ensure smooth GUI responsiveness during playback and provides real-time progress tracking. Future enhancements include adding support for exporting audio directly to MP3 using gTTS.

3. Tools Used

Tool / Library	Purpose
Python 3.x	Programming language used
Tkinter	For creating the Graphical User Interface (GUI)
PyMuPDF (fitz)	For extracting text from PDF pages
pyttsx3	For offline text-to-speech conversion
Threading	To prevent GUI freezing during audio playback
OS module	For basic file handling operations

4. Steps Involved in Building the Project

1. Environment Setup – Installed required libraries (fitz, pyttsx3, tkinter).
2. GUI Design – Created a clean, responsive window using Tkinter widgets such as buttons, frames, labels, and progress bars.
3. PDF Selection and Text Extraction – Implemented file selection and used fitz to extract text from the selected PDF.
4. Text Cleaning – Processed raw extracted text to remove extra spaces and line breaks for smooth speech.
5. Speech Conversion – Integrated pyttsx3 for converting cleaned text into speech, supporting both male and female voices.
6. Playback Control – Added buttons for Play, Pause, and Stop functions with progress indication.
7. Multithreading – Used threads to run speech playback without freezing the GUI.
8. Export Option – Added a placeholder for future MP3 export functionality using gTTS.

5. Conclusion

The PDF to Audiobook Converter successfully transforms digital reading into an auditory experience, making study materials more accessible and engaging. It demonstrates effective integration of GUI design, file handling, and speech synthesis in Python. The project can be extended to include multi-language support, direct MP3 export, and voice customization features. Overall, this project serves as a practical application of Python for enhancing accessibility and convenience in reading.