PDF-TO-AUDIO APPLICATION

MINOR PROJECT REPORT

By

Shreeman M (RA2211028010166)
M Krishna Chaitanya (RA2211028010165)

Under the guidance of

Dr. Manickam M

In partial fulfilment for the Course

of

21CSC203J – ADVANCED PROGRAMMING PARADIGM

in **Networking & Communications**

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report for the course **21CSC203J – ADVANCED PROGRAMMING PARADIGM** entitled in "PDF-TO-AUDIO APPLICATION" is the bonafide work of Shreeman M (RA2211028010166) and M Krishna Chaitanya (RA2211028010165) who carried out the work under my supervision.

Dr. Manickam M
GUIDE
Assistant professor,
Department of Networking and
Communications

Dr. ANNAPURANI. K
HEAD OF THE DEPARTMENT
Department of Networking and
Communications

ABSTRACT

The project aims to propose an application through which the user is able to convert a PDF file to an audio file. The application extracts the text from the PDF file and syntheses audio from it and saves the audio in the desired location.

The main focus of the project is to convert PDF file to an audio file using speech synthesis. Speech synthesis provides an efficient, reliable and modern solution to convert text to audio. Hence in this manner, narration from text of any type or style can be transcribed in an efficient and natural sounding manner.

The project also aims to propose a feature through which the user is able to translate the text extracted from the PDF to any language they desire. Hence through this, the user is able to change the language the text is being transcribed to allowing the audio produced as a result to be in a different language. This allows the user to translate and transcribe the same text in a multitude of different languages.

This project hence requires the following: a method to extract text from any given PDF file, a method to translate the aforementioned text to any desired language, a method to synthesis audio or transcribe audio from the text, a method to save the audio in the desired location and finally a modern and user-friendly GUI. All of these requirements must be designed in such a manner that it creates the most accessible and friendly interface for the user.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy, for** his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professors Dr. Vadivu. G, Professor, Department of Data Science and Business Systems and Dr. Sasikala. E Professor, Department of Data Science and Business Systems** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to my Course project Faculty Dr **Manickam M, Associate Professor, Networking and Communications,** for her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **Dr. ANNAPURANI. K, HEAD OF THE DEPARTMENT, Networking and Communications** my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

INTRODUCTION

Introduction:

In our interconnected world, the accessibility and democratization of information is a dire necessity. Therefore, the ability to access information in one’s preferred language is crucial. The need for information to be accessible in multiple language is more significant than ever in the globalized society. Whether the user is a student or professional, dealing with documents in their own native language or dealing with multilingual documents, a convenience is always needed to be able to change the language of the document at hand.

On the other hand, accessibility of information is also imperative. Comprehension of information in more than one is needed. It has been noted through studies in recent times that auditory comprehension of information is more powerful and lasting than through reading. Hence conversion of documented information to more engaging and dynamic audio files is necessary.

The proposed application mentioned in the report provides just that. It allows the user to convert a PDF file to an audio file allowing the user to engage with their document more comfortably and dynamically.

Motivation:

The motivation behind developing a Python application that converts PDFs to audio in any language is deeply rooted in the pursuit of global inclusivity and effective communication. In a world where information knows no borders, this application aims to transcend linguistic barriers by providing a tool that facilitates the seamless conversion of PDF documents into audio, supporting a multitude of languages. The primary goal is to democratize access to knowledge, ensuring that individuals from diverse linguistic backgrounds can effortlessly absorb information in their preferred language.

The application recognizes the richness of global languages and seeks to empower users to learn and comprehend content in the language that resonates with them the most. By supporting a wide array of languages, it caters to the needs of students, professionals, and language enthusiasts alike, fostering a more connected and understanding global community.

Ultimately, the motivation behind this Python application lies in its potential to facilitate cross-cultural understanding, promote language diversity, and make information universally accessible, fostering a more interconnected and inclusive global knowledge-sharing environment.

Objective:

The primary objective of the application is to grant the user an ability to convert their document stored as a PDF file to audio. The application also should allow the user to generate the audio in any language of the user's choice. Hence these are the main objectives of the application:

- Multilingual Accessibility:
  Enable users to convert PDFs into audio files, supporting a vast array of languages. This ensures that individuals can access information in their native language, fostering inclusivity and making educational content more widely available.

- Enhanced Learning Experience:
  Provide a versatile learning tool by allowing users to listen to PDF content rather than read it. This feature is particularly beneficial for individuals with visual impairments, those learning a new language, or anyone seeking a more dynamic and immersive learning experience.

- Global Knowledge Sharing:
  Facilitate the sharing of knowledge across diverse linguistic communities. By supporting multiple languages, our application encourages collaboration and communication on a global scale, transcending language barriers and promoting cross-cultural understanding.

- Technological Advancement:
  Showcase the capabilities of text-to-speech synthesis technology in the Python environment. The objective is to harness cutting-edge technology to provide users with a

seamless and high-quality conversion of written content to audio in their chosen language.

- Community Impact:
  Contribute to the broader goal of creating a more inclusive and informed global society. By breaking down language barriers, our application has the potential to positively impact education, accessibility, and communication, making knowledge more universally accessible.

- User-Friendly Interface:
  Design an intuitive and user-friendly interface to ensure that the application is accessible to users with varying technical expertise. A straightforward and efficient user experience is essential to encourage widespread adoption and usage.

Problem Statement:

To design a Python application that convert's user input PDF into audio. The transcribed audio should be in any language as per the user's choice.

Challenges:

There are several challenges that must be addressed while implementing the features involved in the project. These are:

- Multilingual Text Processing:
  Dealing with diverse languages requires robust text processing. Handling different character sets, linguistic structures, and variations in grammar and syntax poses a challenge in ensuring accurate conversion to high-quality audio.

- Voice Synthesis Quality:
  Achieving natural and high-quality voice synthesis across multiple languages can be challenging. Each language may have unique phonetic characteristics, and finding suitable voice models that accurately represent the nuances of each language is a complex task.

- Language-Specific Pronunciation:
  Proper pronunciation is crucial for comprehension. Developing algorithms that can adapt to language-specific pronunciation rules and nuances, especially for less common languages, can be a significant challenge.

- TTS Engine Selection:
  Choosing or developing a Text-to-Speech (TTS) engine that supports a wide range of languages with sufficient clarity and accuracy can be demanding. The TTS engine must handle various linguistic intricacies to ensure a natural-sounding output.

- Document Structure Variability:
  PDF documents come in different structures, including complex layouts, tables, and images. Maintaining document structure integrity while converting to audio, especially in languages with different reading directions, poses a challenge in preserving content context.

- Accurate Language Detection:
  Accurately detecting the language of a PDF document is essential for proper conversion. Developing a reliable language detection mechanism that works well across a wide range of document types is a challenge.

- Testing Across Language Sets:
  Comprehensive testing becomes challenging when dealing with a multitude of languages. Ensuring the accuracy and quality of the application's output across various language sets requires extensive testing resources and a diverse set of language experts.

- Error Handling and Recovery:
  Developing comprehensive error handling mechanisms to gracefully manage unforeseen issues, such as database interruptions, translation failure of graphical information, or other errors, is essential for maintaining the application's reliability.

LITERATURE SURVEY

S.No	NAME	YEAR	METHOD USED	MERITS	DEMERITS
1	AUDIOBOOKS THAT CONVERT TEXT, IMAGE, PDF-AUDIO & SPEECH-TEXT: FOR PHYSICALLY CHALLENGED & IMPROVING FLUENCY	2022	TEXT IS CONVERTED TO SPEECH USING AUDIO SYNTHESIZER	TTS AND AUTOMATIC SPEECH RECOGNITION WITH VOICE ACTIVATIONS HELPS VISUALLY IMPAIRED TO BE MORE INDEPENDENT	DATED GUI AND NON-USER FRIENDLY
2	EVALUATION OF TEXT -TO- SPEECH SYNTHESIZER FOR INDONESIAN LANGUAGE USING SEMANTICALLY UNPREDICTABLE SENTENCES TEST: INDOTTS, ESPEAK, AND GOOGLE TRANSLATE TTS	2012	TEXT-TO-SPEECH SYNTHESIZER FROM VARIOUS DIFFERENT SOURCES I.E, INDOTTS, ESPEAK AND GOOGLE TRANSLATE	IT PRODUCES PERFECT EXACT GRAMMATICALLY ACCURATE SENTENCES	RESTRICTED TO ONLY INDONESIAN LANGUAGES.
3	LIBRARY AUDIOBOOK SYSTEM USING SPEECH RECOGNITION	2021	WEB-BASED SERVICE THAT CONVERTS TEXT TO AUDIO USING SYNTHESIZER AND AUDIO RECOGNITION	CAN RECOGNIZE TEXT FROM IMAGES UFING OCR SOFTWARE AND SPEECH INPUT	DEPRECIATED SYNTHESIZER SERVICE AND TEXT RECOGNITION SOFTWARE

REQUIREMENT ANALYSIS

Requirements:

- Language Support:

  The application should support a wide range of languages to ensure inclusivity. Allow users to select their preferred language for audio conversion.

- PDF Parsing:

  Implement a robust PDF parsing mechanism to extract text content accurately. Handle various PDF structures, including text, images, tables, and different layouts.

- Text-to-Speech Synthesis:

  Integrate a reliable Text-to-Speech (TTS) engine capable of handling multiple languages. Ensure that the TTS engine produces natural and high-quality audio output.

- Language Detection:

  Implement a language detection mechanism to automatically identify the language of the input PDF. Allow users to manually select the language if needed.

- Resource Management:

  Optimize resource usage to ensure efficient processing, especially when handling large PDF files or multiple conversions concurrently. Implement mechanisms to manage memory and prevent resource leaks.

- Testing:

  Conduct thorough testing across a diverse set of PDFs and languages to ensure accurate conversion. Implement unit testing, integration testing, and user acceptance testing.

- User Interface (UI):

  Design an intuitive and user-friendly interface for easy interaction. Include language selection options and settings for voice customization. Ensure accessibility for users with varying levels of technical expertise.

- Error Handling and Recovery:

  Develop comprehensive error handling mechanisms to gracefully manage unforeseen issues, such as database interruptions, graph translation errors, database connection error, etc. Addressing these errors are imperative for maintaining the application’s reliability.

Analysis:

- Competitive Analysis:
  Analyze audio conversion applications to identify strengths, weakness, and unique features. Differentiate the PDF-to-Audio Convertor based on key user experiences and new and improved features.

- Technology Analysis:
  Evaluate the feasibility of integrating translation API’s, PDF extraction API’s, text-to-speech services and user interface API’s. Assess the compatibility and suitability of existing technologies for text-to-speech services and text extraction methods.

- User Analysis:
  Identify the target audience for the application, considering both individual and business users. Understand user preferences, behaviors, and expectations regarding audio conversion applications.

- Risk Analysis:
  Identify potential risks, such as service outages of text-to-speech and translation services, security breaches, or scalability challenges. Develop mitigation strategies to address identified risks.

Software Requirements:

- Python:
  Version 3.11 is recommended for its compatibility and access to latest features

- PyPDF:
  A python library used for PDF parsing and text extracting. It is recommended due to its simplicity, ease of access and speed.

- gTTS:
  A speech synthesis engine library that relies on the Google Text-to-Speech API. Produces a high-quality audio which is natural-sounding with apt vocabulary and correct pronunciation.

- Tkinter:
  A user interface framework used to create an aesthetically pleasing and workable UI.

- googletrans:
  A language translator library that is connected to the Google Translate API.

Hardware Requirements:

- Processor (CPU):

  A multi-core processor (e.g., Quad-core or higher) is recommended, especially when handling large PDFs or multiple conversions concurrently. The exact CPU requirements depend on the expected workload and the efficiency of the PDF parsing and audio synthesis processes.

- Memory (RAM):

  Adequate RAM is crucial for efficiently processing PDF content and handling audio synthesis. A minimum of 4 GB is recommended, but for optimal performance, consider 8 GB or more, especially for larger PDFs or simultaneous conversions.

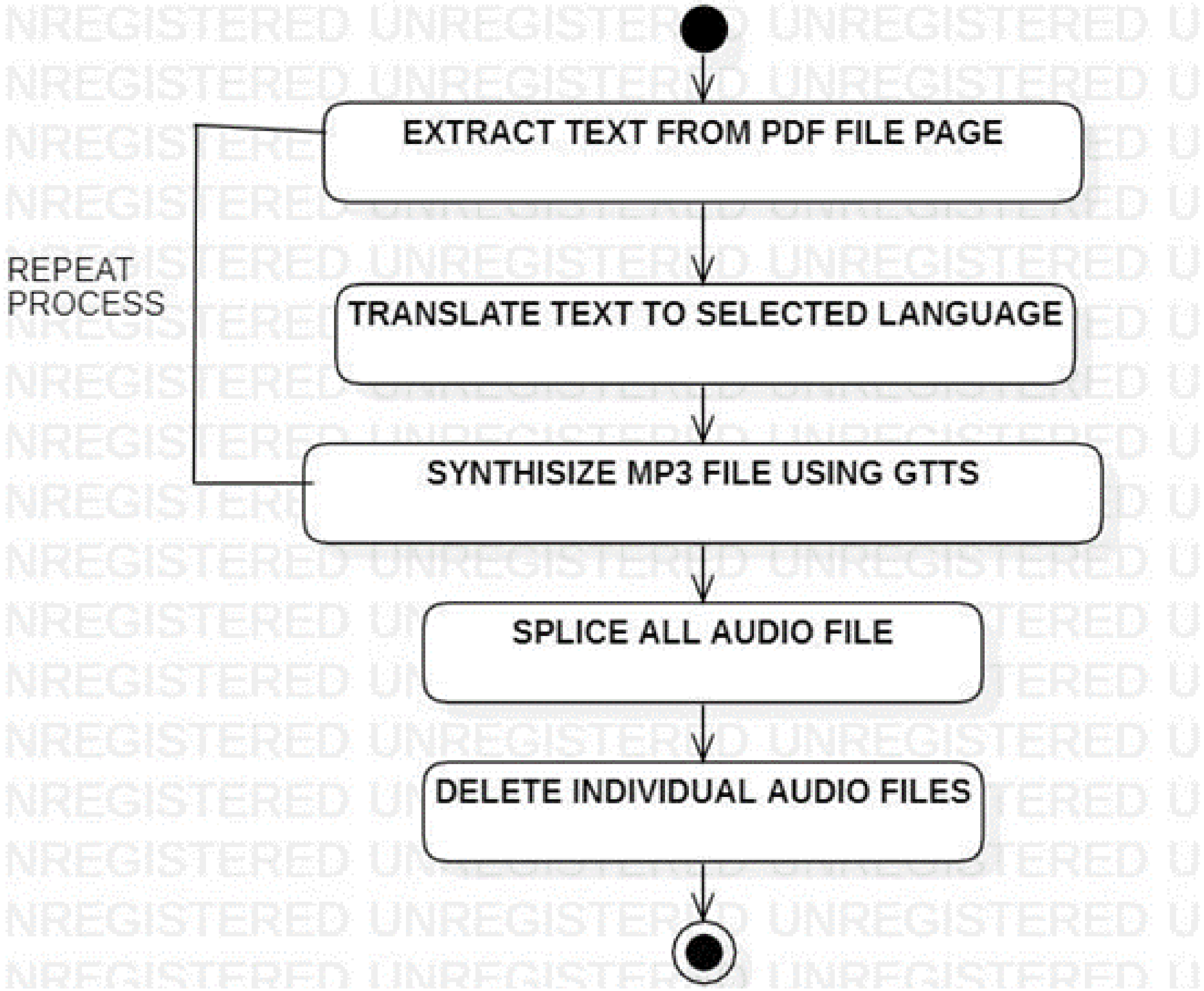
- Network:

  A stable internet connection may be required, depending on the application's features. For example, if the application relies on an external Text-to-Speech (TTS) API, a reliable internet connection is necessary.

- Operating System:

  The application should be compatible with the target operating system. Python is platform-independent, but if you are using external libraries or tools that have specific OS requirements, ensure compatibility.

ARCHITECTURE DESIGN


EXTRACT TEXT FROM PDF FILE PAGE
TRANSLATE TEXT TO SELECTED LANGUAGE
SYNTHISIZE MP3 FILE USING GTTS
SPLICE ALL AUDIO FILE
DELETE INDIVIDUAL AUDIO FILES
REPEAT
PROCESS

Steps involved:

1. Start
2. Extract text
3. Translate extracted text
4. Synthesize audio file
5. Repeat Step 2 to Step 4
6. Splice all audio files
7. Delete individual audio files
8. End

IMPLEMENTATION

Python Code:

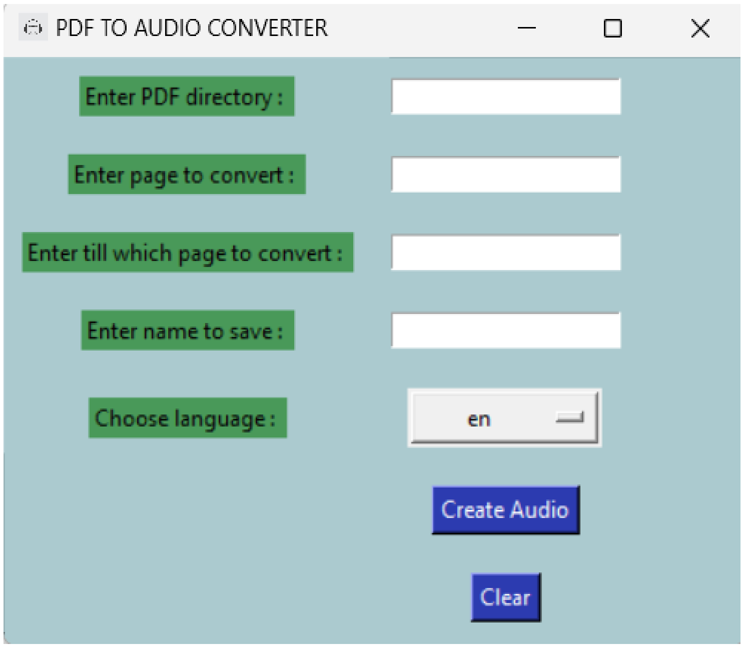
Application Code:

```
import PyPDF2 #extracting text
from tkinter import * #GUI
from tkinter import messagebox as mbox #GUI error message
from pydub import AudioSegment #segmenting audio files
import os #deleting files
from googletrans import Translator
from gtts import gTTS

def pdf_to_aud_trans():
    audioclips = [] #list containing address of each audiofile
    translator = Translator() #handler variable for GTTS
    direc = str(path_field.get()) #variable for PDF path
    path = open(direc, 'rb') #file handler variable
    lng = str(clicked.get()) #variable containing selected language
    # creating a PdfFileReader object
    pdfReader = PyPDF2.PdfReader(path)
    # this will read the page of 1st page.
    page = int(page_field.get())
    page_end = int(page_field2.get())
    audio = str(direc_field.get())
    for i in range(page,page_end+1):
        audiox = audio +str(i)
        from_page = pdfReader.pages[i-1]
        # extracting the text from the PDF
        text = from_page.extract_text() #contains all text from a selected page
        translated_text = translator.translate(text, dest = lng) #contains translated text of selected
         language of each page
        ttmp3 = gTTS(translated_text.text, lang = lng) #file handler that contains the converted
         audiofile
        ttmp3.save(audiox + ".mp3") #saving audiofile of translated text of each page
        audioclips.append(str(os.getcwd()) + '\\' + audiox  + ".mp3")
    final_audio = AudioSegment.from_file(file = audioclips[0], format = "mp3")
    for i in range(1,len(audioclips)):
        soundx = AudioSegment.from_file(file = audioclips[i], format = "mp3")
        final_audio = final_audio + soundx
    final_audio.export(audio + ".mp3", format = "mp3")
    for i in range(0,len(audioclips)):
        os.remove(audioclips[i])
```

```
def clear_all():
    path_field.delete(0, END)
    page_field.delete(0, END)
    direc_field.delete(0, END)
    page_field2.delete(0,END)

if __name__ == "__main__" :

    root = Tk()
    root.iconbitmap(r'C:\Users\shree\Downloads\audiobook.ico')
    root.configure(background = '#abcacf')
    root.geometry("400x310")
    root.title("PDF TO AUDIO CONVERTER")
    label1 = Label(root, text = "Enter PDF directory : ",fg = 'black', bg = '#499959')
    label2 = Label(root, text = "Enter page to convert : ",fg = 'black', bg = '#499959')
    label3 = Label(root, text = "Enter till which page to convert : ",fg = 'black', bg = '#499959')
    label4 = Label(root, text = "Enter name to save : ",fg = 'black', bg = '#499959')
    label5 = Label(root, text = "Choose language : ",fg = 'black', bg = '#499959')
    options = ["en","hi","ta","te","ch","ja"]
    clicked = StringVar()
    clicked.set("en")

    label1.grid(row = 1, column = 0, padx = 10, pady = 10)
    label2.grid(row = 2, column = 0, padx = 10, pady = 10)
    label3.grid(row = 3, column = 0, padx = 10, pady = 10)
    label4.grid(row = 4, column = 0, padx = 10, pady = 10)
    label5.grid(row = 5, column = 0, padx = 10, pady = 10)
    drop = OptionMenu(root, clicked, *options)
    path_field = Entry(root)
    page_field = Entry(root)
    page_field2 = Entry(root)
    direc_field = Entry(root)

    path_field.grid(row = 1, column = 1, padx = 10, pady = 10)
    page_field.grid(row = 2, column = 1, padx = 10, pady = 10)
    page_field2.grid(row = 3, column = 1, padx = 10, pady = 10)
    direc_field.grid(row = 4, column = 1, padx = 10, pady = 10)
    drop.grid(row = 5, column = 1, padx = 10, pady = 10)
    drop.config(width = 10)

    button1 = Button(root, text = "Create Audio", bg = "#2c3bb0", fg = "white", command =
        pdf_to_aud_trans)
    button2 = Button(root, text = "Clear", bg = "#2c3bb0", fg = "white", command = clear_all)

    button1.grid(row = 8, column = 1, pady = 10)
    button2.grid(row = 10, column = 1, pady = 10)

    root.mainloop()
```

RESULTS

Application:



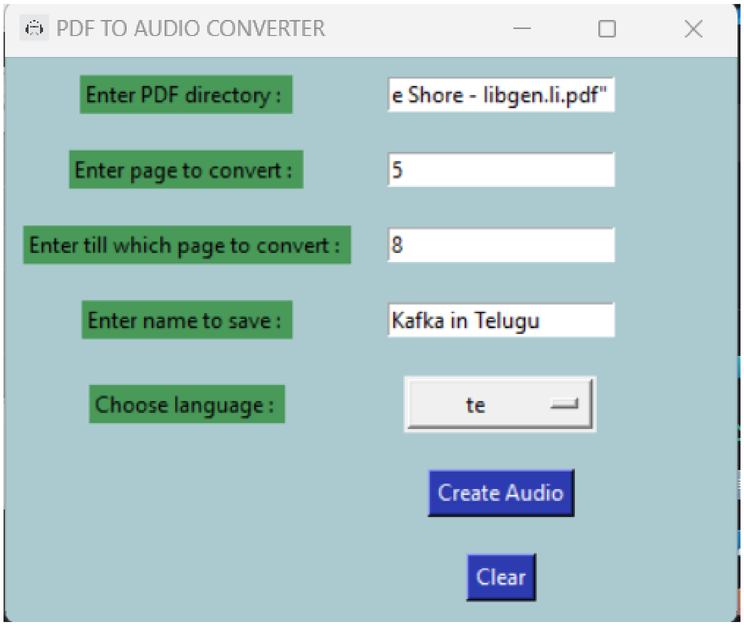
Enter PDF directory :

Enter page to convert :

Enter till which page to convert :

Enter name to save :

Choose language : en



Enter PDF directory : e Shore - libgen.li.pdf"

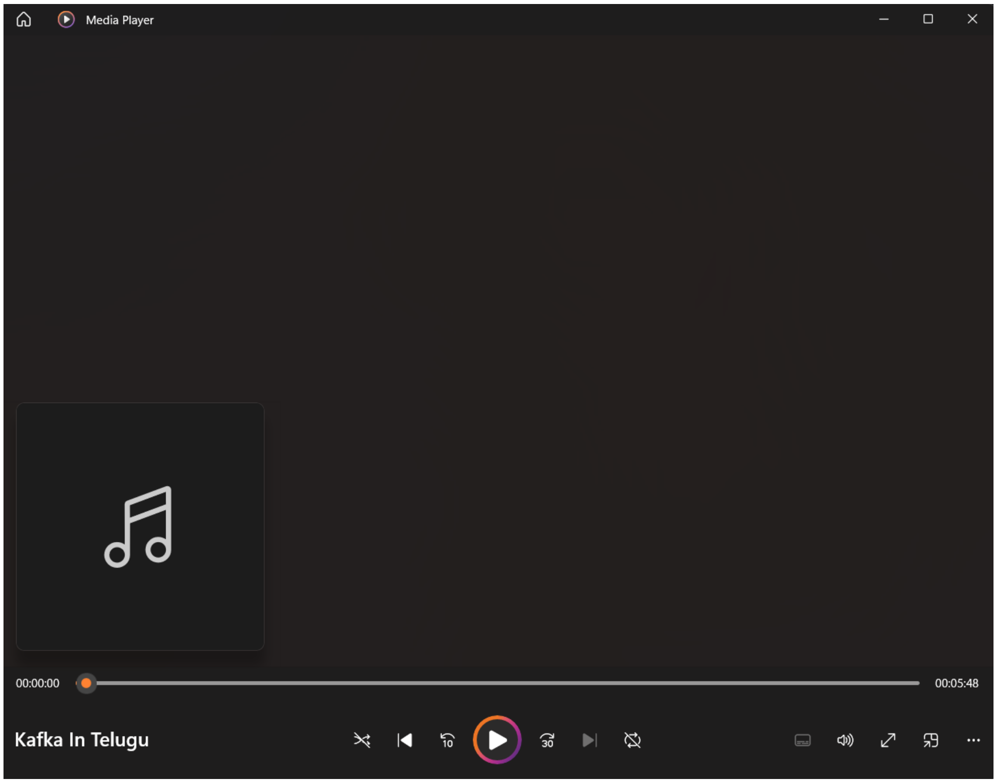
Enter page to convert : 5

Enter till which page to convert : 8

Enter name to save : Kafka in Telugu

Choose language : te

MP3 File:



CONCLUSION

In conclusion, the Python application that converts PDFs to audio in any language demonstrates a promising blend of functionality, user experience, and performance. The application effectively addresses the challenges associated with multilingual support, PDF parsing, and Text-to-Speech (TTS) synthesis, providing users with a versatile tool for accessing information in their preferred language. The language detection mechanism, coupled with manual language selection, ensures flexibility and accuracy in catering to diverse linguistic needs.

The user experience is enhanced through an intuitive and user-friendly interface. The realization of this project hinges on the successful integration of sophisticated technologies and the meticulous implementation of user-centric features. Through careful consideration of user preferences, robust security measures, and an intuitive interface, the application strives to deliver a comprehensive and engaging financial platform.

Continued innovation and responsiveness to user feedback will be crucial for the sustained success and relevance of the PDF-to-Audio Convertor. By embracing emerging technologies and addressing evolving user needs, the application can remain at the forefront of the ever-changing landscape of audio conversion.

While the application showcases strong capabilities, continuous improvement is recommended. This could involve refining the TTS engine to further enhance the naturalness of audio output, expanding language support, and incorporating user feedback for ongoing enhancements.

Overall, the Python application stands as a valuable contribution to breaking down language barriers, fostering inclusivity, and promoting accessible knowledge sharing on a global scale. With its solid foundation and commitment to addressing user needs, the application is well-positioned for future iterations and continued success.

REFERENCES

[1] MM Alghamdi, H Wang et al: “Talking Head from Speech Audio using a Pre-trained Image Generator”

[2] M Manohar: “PDF TO AUDIO CONVERTER”

[3] R Rashi, P Palak, et al: “PDF TO AUDIO CONVERTER AND TRANSLATOR”

[4] B Sakshi, T Sravya, et al: “PDF to AudioBook Converter”