**A**

**PROJECT REPORT**

**ON**

# AUDIOBOOK USING GUI

**SUBMITTING TO:**

1. Mr. Mayur Dev Sewak
   General Manager, Operations
   Eisystems Services
2. Ms. Mallika Srivastava
   Trainer, Data Science & Analytics Domain
   Eisystems Services

**SUBMITING BY:**

B.DEEKSHITH

# LIST OF CONTENTS



# LIST OF FIGURES:

# 1. ABSTRACT

Audiobooks are ideal for anyone who, like most of us, likes to listen rather than read. It is just not possible to purchase and store them in your home on a bookshelf. Audiobooks are also a good method to relax your eyes and take a break from the continual stimulation of digital devices. Others can take advantage of them to save time. For example, keep up with books while doing different tasks at same time. It has the potential to not only alleviate problems for millennials, but also to be a highly useful tool for visually impaired people. The ability to transform any material into an audiobook is a true gift to civilization. Our technology can be put to use in the development of such tools. Text-to-speech and other read-aloud programmes are widely utilised to assist students in developing their reading comprehension abilities.

A PDF to audio system is a screen reader programme that has been designed and eveloped specifically for the purpose of effective audio communication. International Organization for Standardization (ISO) established PDFs as an open standard document format for the aim of displaying and transmitting information securely (ISO). One of the most convenient formats for electronic communication and information transmission is the document format. It's critical if we want to improve accessibility for screen readers by including audio into our material. Among the features PDF documents provide are text links and buttons as well as audio and video files. Many languages may be supported by the PDF to audio technology, which will allow users to hear text being read aloud (spoken).

# 2. SUMMARY

The Audiobook Player with GUI is a Python-based project that aims to create an interactive application for listening to audiobooks. The project utilizes a graphical user interface (GUI) to provide users with a user-friendly platform for managing and playing audiobook files.

# 3. INTRODUCTION

Audiobooks are recorded versions of a book's text that we listen to rather than read. Audiobooks can be literal word- for-word transcriptions of novels or shortened versions that exclude unnecessary language. For instance, we may convert short stories otherwise novels

into audiobooks so that scholars can rapidly obtain a description of the work. Text-to-speech related audio reading tools are widely accepted across the world and they are implemented in an attempt to encourage students listening skills. They help children in listening short stories at bed time as parents can now convert any story stories or light novels into audiobook. PDF to audio system is a screen interpretation application designed for an effective audio communication. PDFs are an open standard document format used internationally,maintained by the International Organization for Standardization (ISO).

PDF is one of the most suitable methods for electronic communication. And they are also very easy to share and exchange through electronic information conversation system. PDF documents are intended to contain links, buttons, forms, audios.

Optical-character-recognition is what is meant by the term "OCR." A generally used method for recognising text embedded in images, such as scanned documents and photos, is optical character recognition (OCR). When a picture contains written text, optical character recognition (OCR) technology is used to extract the text from the image. It is conceivably most recognised for its use of optical character recognition to turn printed paper documents into machine-readable text files. Word processors like Microsoft Word and Google Docs may be used to alter the text on a scanned paper document that has been treated using Optical Character Recognition. The following PDF to audiobook converter may benefit users by giving replacements to reading PDFs for the blind, for young people, for lethargic readers, for busy people to listen during their commute and so on and so forth.

# 4. DETAILS OF PROCESS

**4.1 GUI Design:** Design and implement the graphical user interface using the Tkinter library. Include text input fields, buttons, and dropdown menus for voice selection and playback control.

**4.2 Text-to-Speech Conversion:** Utilize the pyttsx3 library to convert the input text into audio files.

**4.3 Playback Functionality:** Use the PyAudio library to enable audio playback and control, including play, pause, stop, and speed adjustment.

**4.4 Save Audio:** Implement functionality to save the generated audio as an MP3 file using appropriate libraries or modules.

**4.5 Error Handling:** Implement error handling mechanisms to catch and handle exceptions gracefully, displaying informative error messages to the user.

# 5. SYSTEM REQUIREMENTS

## 5.1 Hardware Requirements

- RAM: 6 GB
- Processor: intel i5

## 5.2 Software Requirements

- Operating System: Windows 10
- Platforms used : Python 3.10 ,command prompt

# 6. WORK FLOW

**6.1 Graphical User Interface**: The project utilizes the Tkinter library to develop an interactive GUI, allowing users to navigate and control the audiobook playback effortlessly.

**6.2 Audiobook Selection**: Users can browse their computer's file system within the GUI and select the desired audiobook file for playback. Common audio formats like MP3 or WAV are supported.

**6.3 Text-to-Speech Conversion:** The project incorporates the pyttsx3 library to convert the book's text into synthesized speech, providing a seamless listening experience.

**6.4 Playback Controls:** The application includes controls such as play, pause, stop, and navigation options (e.g., forward and backward) to give users full control over the playback of the audiobook.

**6.5 Customizable Audio Settings**: Users can personalize their listening experience by adjusting audio settings such as playback speed and volume through intuitive sliders or controls provided in the GUI.

**6.6 Bookmarking and Resuming:** The application enable users to bookmark their progress within an audiobook and easily resume playback from the saved position at a later time.

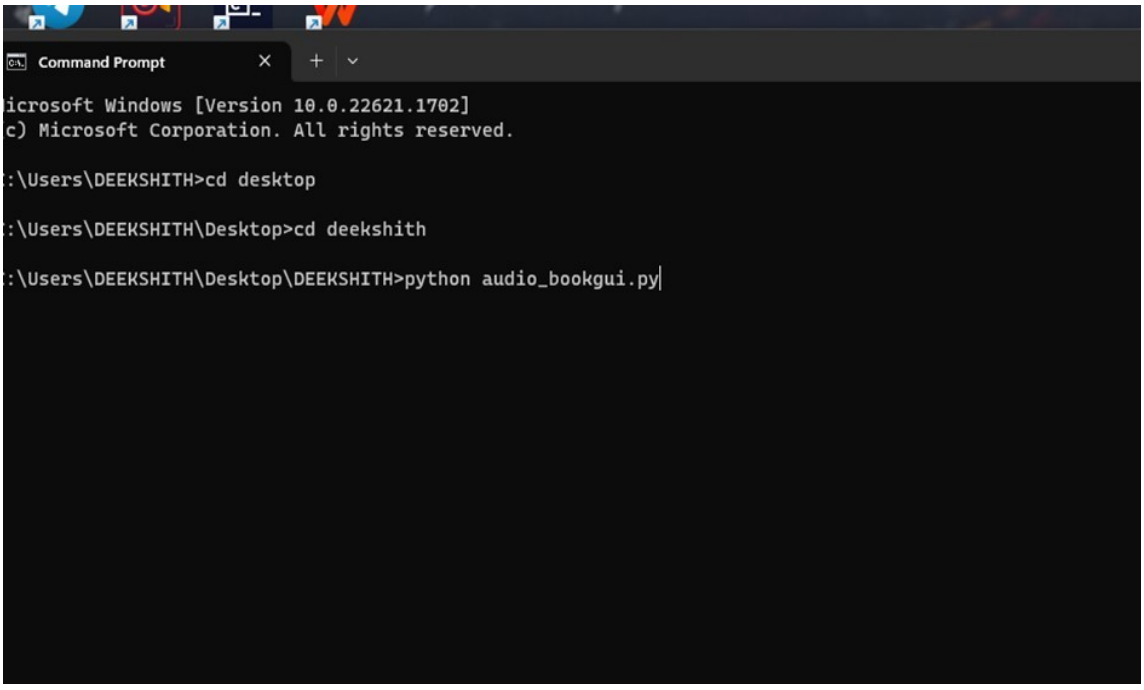
## 7. METHODOLOGIES USED

**7.1 Python:** The core programming language for developing the application.

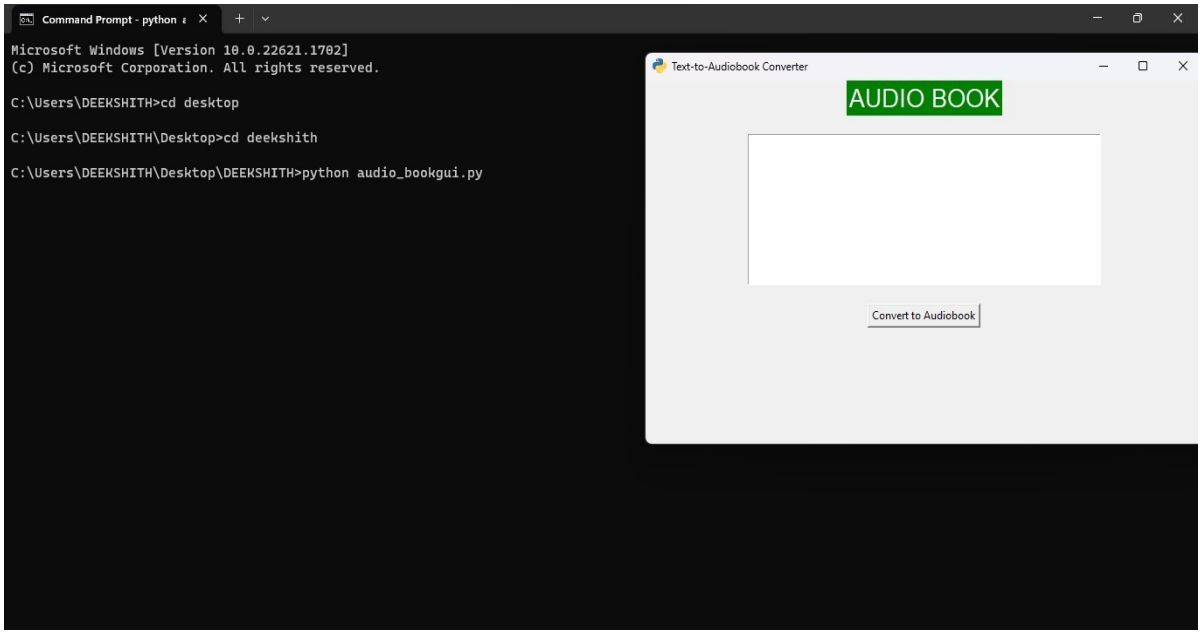
**7.2 Tkinter:** A Python GUI toolkit used to create the graphical user interface.

**7.3 pyttsx3:** A Python library used for text-to-speech conversion.

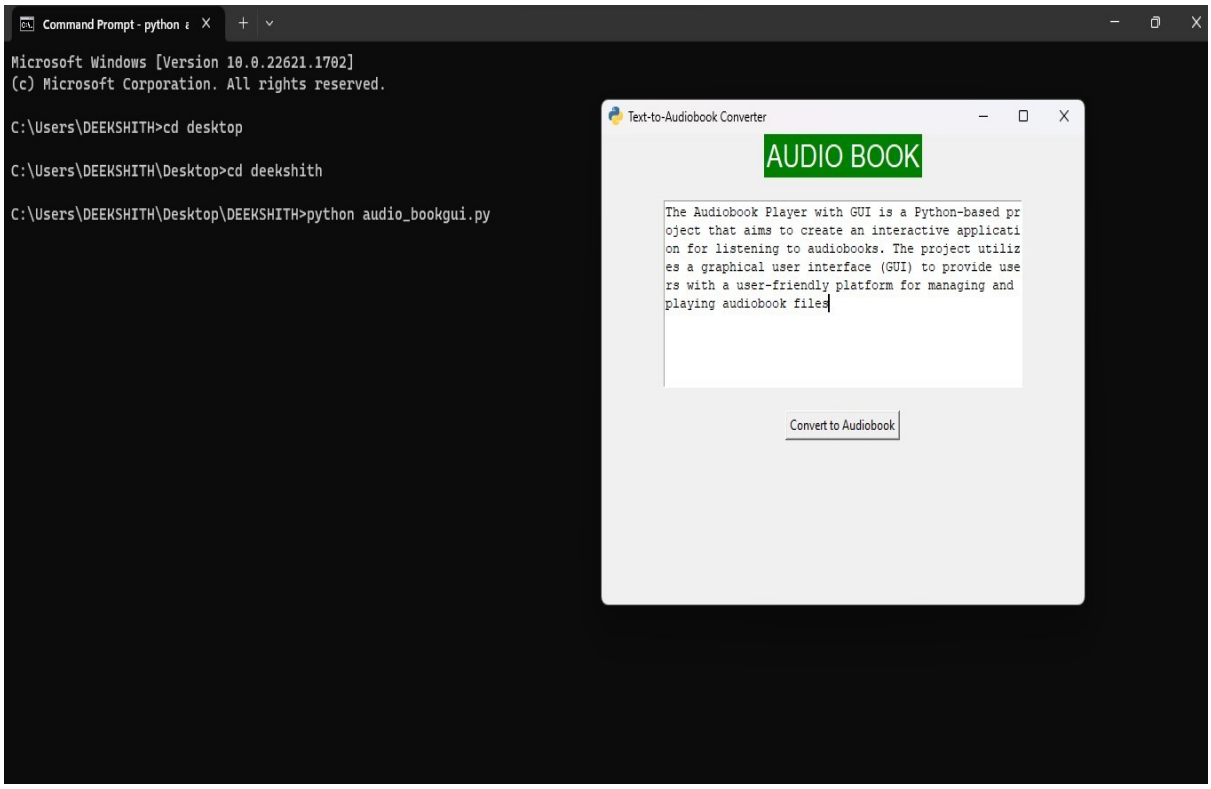
## 8. INPUT OUTPUT DATASETS/SCREENSHOTS



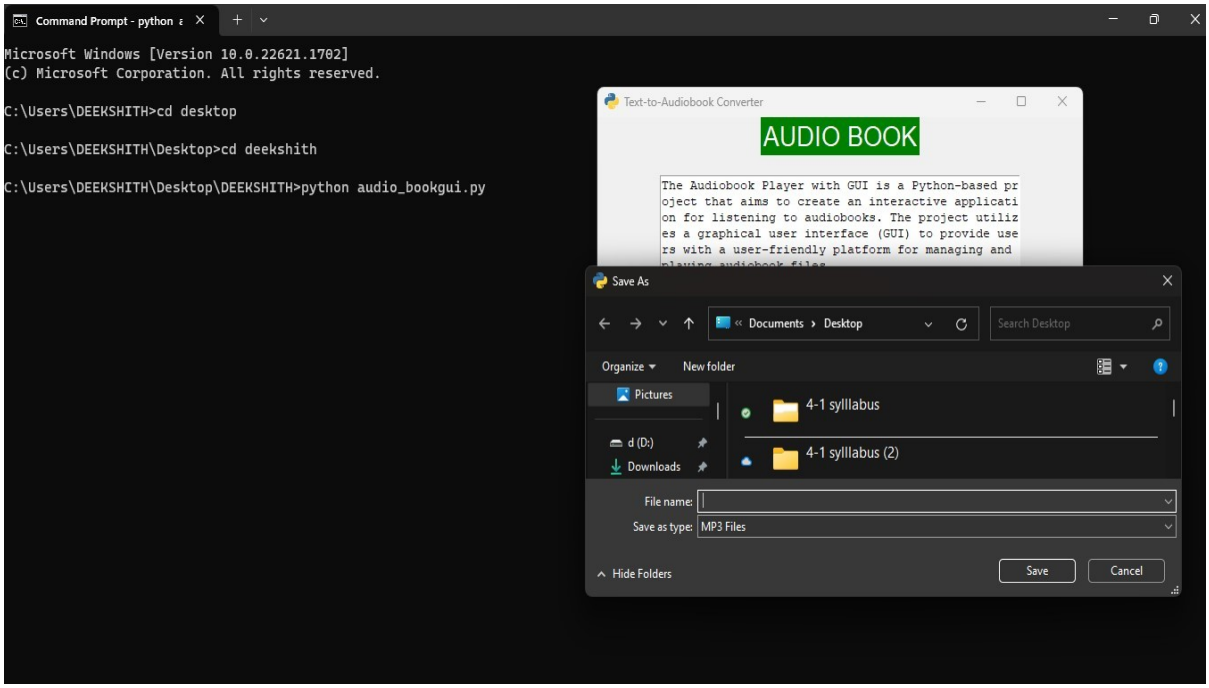

8.1 accessing audiobook using command prompt



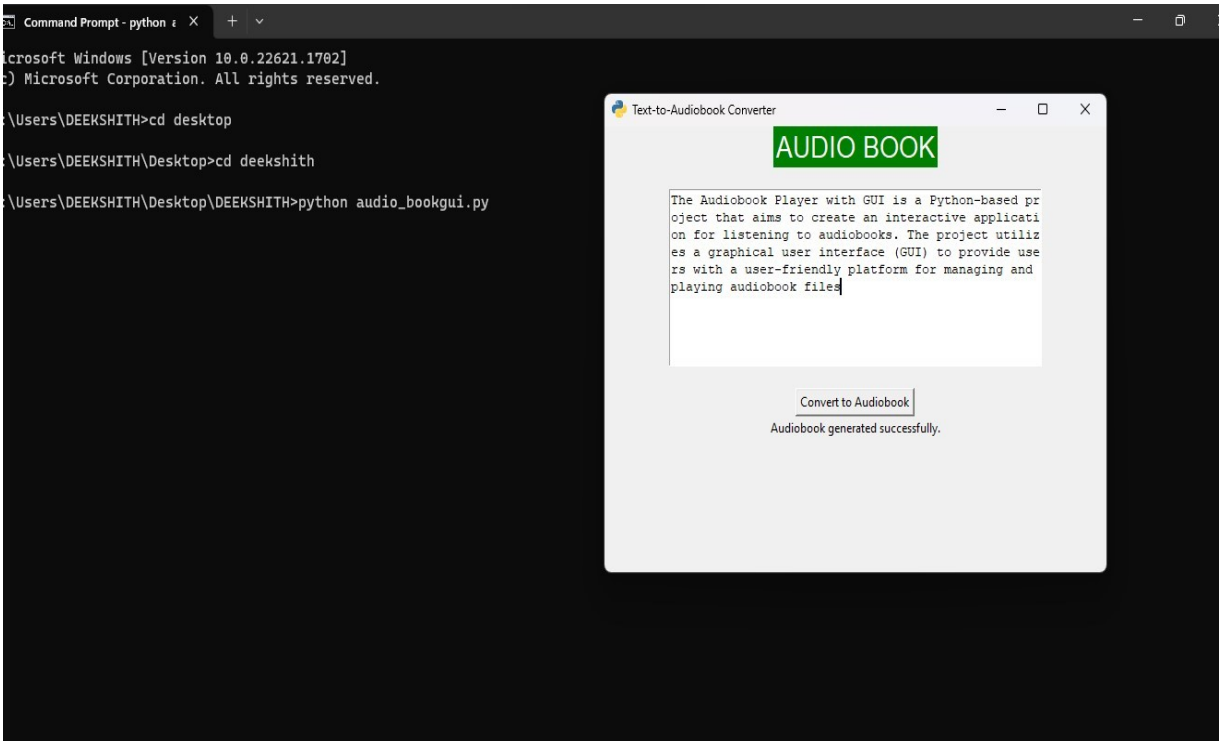

8.2 audio-book using command prompt




8.3 text into entry box




8.4 save audio to desired location




8.5 message on successful audiobook generation

# 9. TEXT CODE

audio_bookgui.py - C:\Users\DEEKSHITH\Desktop\DEEKSHITH\audio_bookgui.py (3.7.0)

File Edit Format Run Options Window Help

```
import tkinter as tk
from tkinter import filedialog
import pyttsx3

def convert():
    text = text_entry.get("1.0", tk.END).strip()

    if text:
        engine = pyttsx3.init()

        output_file = filedialog.asksaveasfilename(defaultextension=".mp3", file
        if output_file:
            engine.save_to_file(text, output_file)
            engine.runAndWait()
            status_label.config(text="Audiobook generated successfully.")
        else:
            status_label.config(text="No output file selected.")
    else:
        status_label.config(text="No text entered.")



# Create the GUI window
t= tk.Tk()
t.title("Text-to-Audiobook Converter")
t.iconbitmap("python.ico")

text_label=tk.Label(t,text="AUDIO BOOK",bg="green",fg="white",font="Arial 20")
```

```
text_label=tk.Label(t,text="AUDIO BOOK",bg="green",fg="white",font="Arial 20")
text_label.pack()


# Create and configure the text entry widget
text_entry = tk.Text(t, height=10, width=50)
text_entry.pack(pady=20)



# Create and configure the convert button
convert_button = tk.Button(t, text="Convert to Audiobook", command=convert)
convert_button.pack()


# Create the status label
status_label = tk.Label(t, text="")
status_label.pack()

# Start the GUI event loop
t.mainloop()
```

# 10. REFERENCES


- www.pycharm.com
- www.google.com
- www.ijedr.com