# PDF Note-taking system and editor with Python

[1]Harmit Saini, [2]Siddharth Singh, [3]Sarvesh Shirwalkar, [4]Shubham Mojidra, [5]Sukhada S. Aloni

[1,2,3,4]Student, [5]Faculty, A.P. Shah Institute of Technology, Thane, India, [1]20102125@Apsit.edu.in, [2]20102176@Apsit.edu.in, [3]20102109@Apsit.edu.in, [4]20102108@Apsit.edu.in, [5]ssaloni@Apsit.edu.in

**Abstract: The COVID-19 pandemic has wreaked havoc on every aspect of society. A huge chunk of daily chores and tasks have shifted to the digital world. Face-to-face classes have been canceled and moved online, bringing about the rise of online learning and hence technologies like Online conference apps, online notebooks, PDF editors, etc. As the students going through the experience of E-learning, we realize the need for a good PDF editor. A PDF editing software solution turns PDF into a living, editable document. PDFs can be created within or uploaded into, these educational streams. PDF editors often include password protection to view and edit documents and designated fields of education which is collaborative and allows users to edit any part of the document at the institution level. We propose a user-friendly tool, to ease the process of pdf editing for the students and teachers. Our tool includes basic security features like encryption and decryption along with our very own unique ideas such as Text-to-speech, note highlighter, and text recognition, all packed in one user-friendly software for desktops.**

***Keywords- PDF, PDF to audio, PDF highlighting, PDF editor, PDF security, COVID-19.***

## I. INTRODUCTION

With the rise of the deadly Corona Virus and things moving online, individuals have realized the need for PDF editors.

Our plan is to make things easier for the students and teachers who often struggle to find a good PDF editor which results in wasting their time. All the previous technologies available lack an accumulation of useful features in one place. This project eliminates most of the drawbacks of the existing system and will also help users to save time finding a useful free-to-use offline PDF editor. In today's world, the existing systems are either paid, filled with ads, or require authentication. People may feel their privacy is hindered. Most of the editors which are already present have limited features. There is a lack of free-to-use, bugs, and an advertisement-free PDF editor with a good UI.

We have implemented a PDF text editing feature using a notepad made with python. The text is extracted from the PDF file and is fetched into the text field, open to edit for users. The menu bar which sits on the top of the text field contains various features. As the user presses the save button, the text is collected and converted into a PDF which is saved into the chosen directory. The software also distinguishes between image-based and text-based pdfs. By default we assume the pdf to be text-based. In case, the function returns an empty string, we go on to treat the PDF as an image-based pdf. Image-based pdfs have different processing algorithms and use different libraries such as Tesseract, OpenCV, Poppler, etc. This bifurcation opens the door to processing scanned pdfs too. The resultant pdf is a text-based pdf and is stored as per the user's need.

The GUI of the notepad is implemented entirely via Tkinter.

Since everything is becoming digital, one of the most time-consuming tasks a student does, that is, jotting down notes, can also become digital. Our pdf editor comes with a feature to highlight text that the user may find essential. Apart from being incredibly efficient and time-saving, taking notes online also prevents the wastage of papers and stationery. This is a highlighting feature of this software. We work with the notes(highlighted text) that the user has taken down in various other features of this software.

The approach we followed to accomplish this was using existing features of the PyPDF2 and PyMuPDF libraries in python which includes functions like annotations to highlight the text of the PDF. But the logic we followed was to get the text inside the text field of the notepad and when the user click's on the highlight button, we get that selected text and then we save it in a list. Once the user clicks on the save button, we search for those occurrences inside that PDF that the user selected highlight the selected text. Then we allow users to choose where they want to save their file by opening a file dialog in Tkinter. Once the PDF is saved, the user can use that PDF for quick revision of notes, i.e. highlighted text of the PDF and listen to the audiobook of the same.

In modern times, people are stocked with work and often have trouble finding a window to read a book or notes, moreover, students who find it hard to concentrate while reading, may benefit from audio-visual stimuli to help the focus. We have implemented a PDF to Audio feature so that users can listen to the PDF as an audiobook. Taking this a step further, combining this feature with the PDF highlighting feature, we have created a unique 'notes to audiobook' feature which allows the client to listen to only the notes(highlighted text) from the PDF. This helps during those last-minute revisions.

There were two options for us to go with, GTTS and pyttsx3

DOI : 10.35291/2454-9150.2022.0165

modules of python. The reason we went with the pyttsx3 module is that it is easy to implement and requires comparatively less time than the gtts module. It has easy-to-use functions for speech conversion. The .wav file is generated and saved by extracting the text of the PDF and then parsing it to the pyttsx3 module's function to get the desired audio of the input text (which in this case is the PDF's text). Therefore, when the user clicks the play button, that .wav file is played which is nothing but the text-to-speech audio file. We chose the .wav file over the .mp3 file because the pygame module is buggy with its implementations on .mp3 files.

To make the user experience pleasant, we have integrated a music player with essential features to control the audio stream. Users would be able to listen to the audiobook or notes at their comfort with the ability to play and pause anytime. The file is also generated in the software's directory so that the user can use other music players to play the file. The music player functionality is implemented using the pygame module in python which has been used to control the audio stream of the player and the basic GUI is implemented using the Tkinter module. The GUI of the audio player has basic functions like a music player viz, Play, Pause, Restart, Stop, Close, etc. which provides user-friendly functionality.

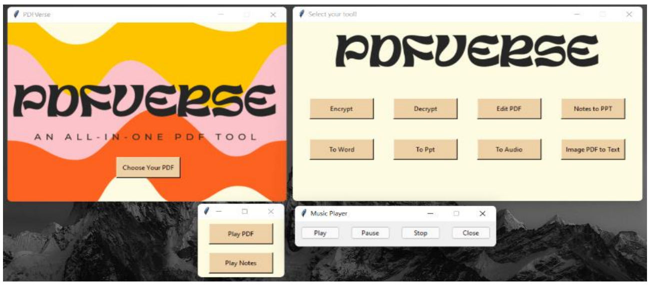
Moving on to more customary features, we have included Encrypt (adding a password to a pdf file), Decrypt (removing password from a pdf file) PDF to Word (.docx), and PDF to PowerPoint(.pptx). The users can convert their PDFs into Microsoft Word documents and PowerPoint presentations with a single click.

Faculties who need to present PowerPoint Presentations during the online classes may often need only important parts from a book (often in pdf format) to remember as teaching points. To assist users burdened with such tasks, we have implemented a Notes (highlighted text) to PowerPoint Presentation(.pptx) file.

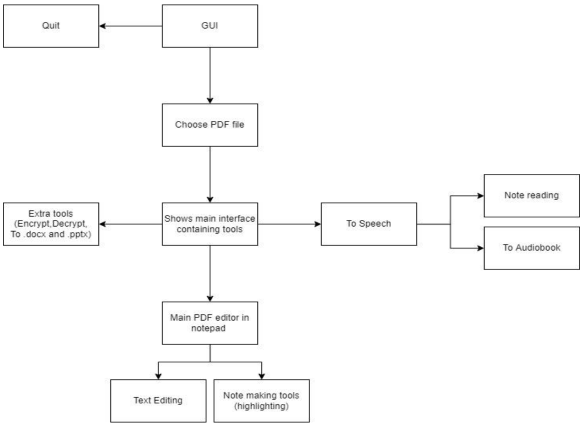
In this method, we make use of the pdf2docx library's parse method where the PDF file is converted into Docx format. We take the user's PDF file and then convert it into the word format with this function. The function generates a RAW page of every page of the PDF and then it converts it into word format using other parsing methods present inside the Converter class in the module.

The GUI is made using the Tkinter module in python.

Various widgets such as buttons, canvas, labels, etc were used. A picture of GUI is shown below:



The flow of the program is as follows:


Quit
GUI
Choose PDF file
Extra tools (Encrypt,Decrypt, To .docx and .pptx)
Shows main interface containing tools
To Speech
Note reading
To Audiobook
Main PDF editor in notepad
Text Editing
Note making tools (highlighting)

## II. CONCLUSION

We have fabricated a software that fulfills the major PDF editing needs in today's world. The note-taking system which is a blend of pdf editing and text-to-audio conversion eases the burdens of a common user. The support for hand-written text recognition from PDFs and dictating to PDFs can be considered as future scopes. The combination of various libraries based around PDFs and Machine Learning has helped us create a handy all-in-one PDF tool.

## REFERENCES

[1] T.Gnana Prakash | K. Anusha "Text Extraction from Image using Python" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-1 | Issue-6, October 2017, pp.310-317

M. (2020). PDF TO AUDIO CONVERTOR. PDF TO AUDIO CONVERTOR, 02(12 DEC 2020), 563-566.

[2] Chan, N. (n.d.). Text-to-speech conversion.

doi:10.5353/th_b3120958

Kumar, S. (2020, October 12). Build your Own audiobook in 7 lines of Python code. Retrieved March 09, 2021, from https://towardsdata/science.com/build-your-own-audiobook-in-7-lines-of-python-code-bfd805fca4b1

[3] Sakshi Bhargava | Sravya Tummala | |Shravani S | Shivam Koul | Nilima Kulkarni "PDF to AudioBook Converter", March 2021| IJIRT | Volume 7 Issue 10 | ISSN: 2349-6002

[4] M. A. Bhatti and A. Ahmad, "PDF to HTML Conversion: Having a Usable Web Document," 2006 1st International Conference on Digital Information Management, 2007, pp. 289-293, doi: 10.1109/ICDIM.2007.369212.

[5] X. Xueya and Z. Yanmei, "The research and application of the creation PDF document based on the iTextSharp," 2012 IEEE Symposium on Robotics and Applications (ISRA), 2012, pp. 108-110, doi: 10.1109/ISRA.2012.6219132.

[6] A. Revathi and N. A. Modi, "Comparative Analysis of Text Extraction from Color Images using Tesseract and OpenCV," 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), 2021, pp. 931-936, doi: 10.1109/INDIACom51348.2021.00167.

DOI : 10.35291/2454-9150.2022.0165