

CONVERTING PDF TO AUDIOBOOK USING DATA SCIENCE AND MACHINE LEARNING

PROJECT PROGRESS REPORT

OF PROJECT-1 (IT795)

BACHELOR OF TECHNOLOGY in Information Technology

SUBMITTED BY

Gourav Emmanuel (13000217092)

Malay Sourav (13000217082)

Nausheen Parween (13000217074)

Krishnendu Patra (13000217087)

Under the Supervision of Dr. Subhamita Mukherjee



Department of Information Technology
Techno Main Salt Lake.
Kolkata -700091.

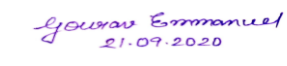
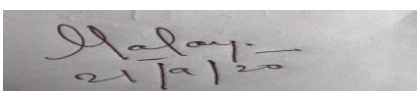

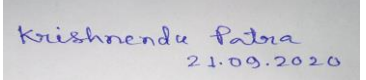


Department of Information Technology
Techno Main Salt Lake
EM-4/1, Salt Lake City, Kolkata-700091

BONAFIDE CERTIFICATE

Certified that this synopsis for the project titled "CONVERTING PDF TO AUDIOBOOK USING DATA SCIENCE AND MACHINE LEARNING" is a part of the project work being carried out by "Gourav Emmanuel, Malay Sourav, Nausheen Parween, Krishnendu Patra" under my supervision.

Full Signature of the Candidates (with date)

1. 
2. 
3. 
4. 

(Signature of the Supervisor)

(Signature of the Head of the Department)

ACKNOWLEDGEMENT

We take this opportunity to express my profound gratitude and deep regards to Supervisor Dr. Subhamita Mukherjee for her exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by her time to time shall carry us a long way in the journey of life on which we are about to embark.

We are obliged to my project team members for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of our assignment.

CONTENTS

Sl.no	Topics	Page no:
1	PROBLEM DEFINITION	5
2	WHAT IS AUDIOBOOK	5
3	LITERATURE REVIEW	5-6
4	APROACH	7
5	BRIEFING	7
6	WORKFLOW DIAGRAM	8
7.	CODE	9-16
	A INTERFACE	
	B ACCESSING THE UPLOADED PDF OF USER	
	C EXTRACTING TEXT FROM PDF	
	D TEXT CLEANING	
	E CONVERTING THE EXTRACTED TEXT INTO AUDIO	
	F UPLOADING THE GENERATED AUDIOBOOK ON THE INTERFACE	
8	OUTPUT	17-18
9	LIBRARIES USED AND THEIR WORKING	19-20
10	CONCLUSION	21
11	REFERENCES	22

1. PROBLEM DEFINITION:

Struggling adult learners find reading to be pure drudgery. Disliking the act of reading is an urgent issue to solve because discovering joy from reading is essential to comprehension. Adult students who are not able to comprehend their textbooks or other forms of literature in various content areas will not learn the concepts and vocabulary necessary to be successful on the test nor in secondary education. With the change in the paradigm of education and digital education getting a nod ahead of traditional mode has made audiobooks a better option to lure students into learning and providing easier access to content for blind people as well. Therefore, we decided to address the problem using Data Science and Machine Learning to provide an efficient solution.

2. WHAT IS AUDIOBOOK?

Walking—it's one of covid-19's greatest (and only) pleasures, isn't it? These days, you can do anything on foot: listen to the news, take meetings, even write notes (with voice dictation). The only thing you can't do while walking is read machine learning research papers.

Lately, Podcasts have gained much popularity among people because of its ease of access and the flexibility it provides as podcasts can be listened to anywhere. Similarly, Audiobooks too have become a popular thing among people though it's not in an episodic but mostly restricted to a book but provides similar flexibility as a podcast. The significance of Audiobooks is that it provides flexibility and portability to the user and can even be used by blind people as well without any human assistance. There have been certain methods in the market which were used to serve the same purpose such as converting the file to epub format and then upload the file to Google Playbooks and then use the read aloud feature. But this method had its own constraint that the file could not be paused or resumed and needed to be started over again.

3. LITERATURE REVIEW

- **Miller, D. (n.d.). Pdf to mp3 converter. Retrieved March 09, 2021, from** Dorian Miller used components of third party and python language for the development of the software. To develop it the methods used by Dorian Miller

were LAME, SAPI (Microsoft Speech Application Programming Interface) and Microsoft's TTS engine.[9]

- **Hamiti, M., & Dika, A. (2010). Learning opportunities through generating speech from written texts.**

Mentor Hamiti M and Pro. Agni Dika in 2010 wrote a paper which can translate Albanian Language into an audiobook. This was done to help those users who has difficulty in eye vision. To make this project a success they used basic unit of Albanian Language and special letter found in the acoustic files.[10]

- **Isewon, I., Oyelade, J., & Oladipupo, O. (2014). Design and implementation of text to speech conversion for visually impaired people.**

This paper was contribution of 3 people Olufunke Oladipupo ,Itunuoluwa Isewon, AJelili Oyelade . In 2014 they wrote paper about the software that they have made called the TextToSpeech Robot in which extracted the text and converted it into speech using java programing language because it is platform independent. To implement the mentioned software the used Digital Signal Processing (DSP) and Natural Language Processing (NLP) .[11]

- **Chan, N. (n.d.). Text-to-speech conversion. doi:10.5353/th_b3120958**

S. Venkateswarlu et al in 2016 wrote a paper in which he showed low cost methods using which we can convert the given text into speech. The model consisted of two modules voice and image processing which was implemented using the concept of text to speech synthesizer and optical character recognition in Raspberry pi.[12]

- **M. (2020). PDF TO AUDIO CONVERTOR. PDF TO AUDIO CONVERTOR,**

Mr. Manohar M. in 2020 wrote a paper in which he made an application which can read only those text which the user wants. The user has to provide a page number to the application which the user wants the text to be converted into speech. To implement it he used the predefined libraries of python such as PyPDF2 and Textract.[13]

4. APPROACH:

Initially, we decided to take the entire project as a single one and we found it difficult to find an appropriate solution. After multiple sessions of brainstorming with our mentor, Dr. Subhamita Mukherjee, we found some breakthrough and managed to make some progress.

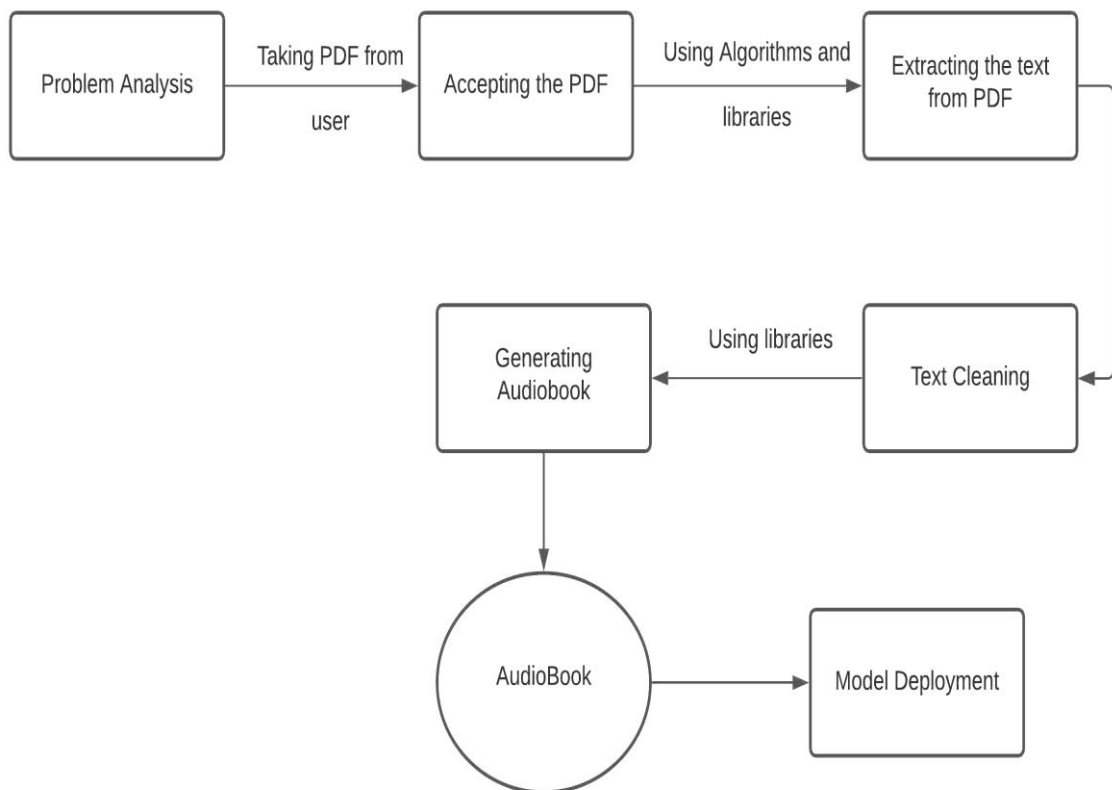
In our approach, we are trying to take a PDF and then extract all the text from the PDF into string. Then we'll be generating the audiobook by converting the string generated to an audio file using the pyttsx3 which is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and creates a mp3 file of the given text which we can play.

5. BRIEFING:

At this point, we have been able to generate the Audiobook from the Pdf. We have create an interface using which the user can upload the pdf of which he wants the audiobook. To make things simple we have given two options to the user ie. Text and image pdf the user has to upload the type of pdf according into these options. After the pdf has been upload we extract the text from the pdf and convert it into string. To do this we have used Pypdf2 and Textract for Text pdf and pdfplumber for image pdf. After the text is extracted the string thus formed is raw and requires cleaning for this we have used NLTK and autocorrect libraries along with replace function. Once the text is cleaned we use the pyttsx3 library to generate the cleaned text into audiobook. The audiobook thus generated will be uploaded on the interface for the user to download and play it on their audio player whenever he wants.

Then we deployed the codes using FLASK. The interaction between the interface and the backend in done using 3 functions of Flask Render_template, Flash and send_file.

6. WORKFLOW DIAGRAM:



7. CODE:

A. INTERFACE

A.1. Html syntax

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@300;400;600;700;800&dis
lay=swap" rel="stylesheet">
    <title>PDF to AudioBook</title>
    <link href="static/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="static/assets/css/fontawesome.css">
    <link rel="stylesheet" href="static/assets/css/main.css">
    <link rel="stylesheet" href="static/assets/css/animated.css">
    <link rel="stylesheet" href="static/assets/css/owl.css">
  </head><body>
<div id="js-preloader" class="js-preloader">
  <div class="preloader-inner">
    <span class="dot"></span>
    <div class="dots">
      <span></span><span></span> <span></span>
    </div></div> </div>
<header class="header-area header-sticky wow slideInDown" data-wow-duration="0.75s"
```

```

data-wow-delay="0s">

<div class="container">

<div class="row">

<div class="col-12">

<nav class="main-nav">

<a href="index.html" class="logo">

<h4>PDFtoAudiobook </h4> </a>

<ul class="nav">

<li class="scroll-to-section"><a href="#top" class="active">Home</a></li>

<li class="scroll-to-section"><a href="#features">Execution</a></li>

<li class="scroll-to-section"><a href="#about">About Project</a></li>

<li class="scroll-to-section"><a href="#services">Team Members</a></li>

<!-- <li class="scroll-to-section"><a href="#portfolio">Portfolio</a></li>

<li class="scroll-to-section"><a href="#contact">Contact Us</a></li> -->

<!-- <li class="scroll-to-section"><div class="main-blue-button"><a
href="#contact">Get Your Quote</a></div></li> --> </ul>

<a class='menu-trigger'>

<span>Menu</span> </a>

</nav> </div></div></div></header>

<div class="main-banner wow fadeIn" id="top" data-wow-duration="1s" data-wow-
delay="0.5s">

<div class="container">

<div class="row">

<div class="col-lg-12">

<div class="row">

<div class="col-lg-6 align-self-center">

<div class="left-content header-text wow fadeInLeft" data-wow-duration="1s" data-
wow-delay="1s">

</div></div>

<div class="col-lg-12">

<h2>Want to listen to your PDF contents on the go? Use our platform to generate an

```

Audiobook and play it on any device.</h2></div>

B.2. CSS style

```
@import
url('https://fonts.googleapis.com/css2?family=Open+Sans:wght@300;400;600;7
00;800&display=swap');

html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p, blockquote, div
pre, a, abbr, acronym, address, big, cite, code, del, dfn, em, font, img, ins, kbd, q,
s, samp, small, strike, strong, sub, sup, tt, var, b, u, i, center, dl, dt, dd, ol, ul, li,
figure, header, nav, section, article, aside, footer, figcaption {
    margin: 0;
    padding: 0;
    border: 0;
    outline: 0;
}.grid:after {
    content: "";
    display: block;
    clear: both;}
.grid-sizer,
.grid-item {width: 50%;}
.grid-item {float: left;}
.grid-item img { display: block;
    max-width: 100%;}
.clearfix:after { content: ".";
    display: block;
    clear: both;
    visibility: hidden;
    line-height: 0;
    height: 0;}
```

```

.clearfix { display: inline-block;}

html[xmlns] .clearfix {display: block;}

* html .clearfix {height: 1%;}

ul, li {padding: 0;
margin: 0;
list-style: none;}

header, nav, section, article, aside, footer, hgroup {
display: block;}

* { box-sizing: border-box;}

html, body {font-family: 'Open Sans', sans-serif;
font-weight: 400;
background-color: #fff;
-ms-text-size-adjust: 100%;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;}

a { text-decoration: none !important;}

h1, h2, h3, h4, h5, h6 {
margin-top: 0px;
margin-bottom: 0px;}

ul { margin-bottom: 0px;}

p {font-size: 15px;
line-height: 30px;
color: #2a2a2a;}

img { width: 100%;
overflow: hidden;
}

```

B. ACCESSING THE UPLOADED PDF OF USER

```
@app.route('/', methods=['POST'])
```

```

def upload_file():

    if request.method == 'POST':

        if 'file' not in request.files:

            flash('No file part')

            return redirect(request.url)

        file = request.files['file']

        if file.filename == "":

            flash('No file selected for uploading')

            return redirect(request.url)

        if file and allowed_file(file.filename):

            filename = secure_filename(file.filename)

            file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))

            global oname

            outname= time.strftime("%Y%m_%M%S")

            oname= "output"+outname+".mp3"

            fname= "input"+outname+".pdf"

            os.rename(r'D:\\Final Year Project\\PDFtoAudiobook\\uploads\\'+ file.filename,
r'D:\\Final Year Project\\PDFtoAudiobook\\uploads\\'+ fname)

            flash('File successfully uploaded')

            open_filename = open('D:\\Final Year Project\\PDFtoAudiobook\\uploads\\'+ fname,
'rb')

```

C. EXTRACTING TEXT FROM PDF

C.1. Extracting text from text pdf

```

doc_details = PyPDF2.PdfFileReader(open_filename)

doc_details.getDocumentInfo()

total_pages = doc_details.numPages

count = 0

text = ""

while(count < total_pages):

    page_text = doc_details.getPage(count)

```

```
count += 1

text += page_text.extractText()
```

C.2. Extracting text from image pdf

```
doc_details = PyPDF2.PdfFileReader(open_filename)

doc_details.getDocumentInfo()

total_pages = doc_details.numPages

error+= 'Total number of pages in PDF:' +str(total_pages)

count = 0

text = ""

with pdfplumber.open('D:\\FinalYearProject\\PDFtoAudiobook\\uploads\\'+ fname) as pdf:

    while(count < total_pages):

        page=pdf.pages[count]

        count += 1

        text=page.extract_text()
```

D. TEXT CLEANING

D.1. Removing all the HTML syntax from our extracted data

```
text=" ".join(filter(lambda x:x[0]!='#', text.split()))
text
```

D.2. Removing all the numeric digits from the text

```
text = " ".join([i for i in text if not i.isdigit()])
text
```

D.3. Replacing ™ with ‘

```
text = text.replace("™", "'")

text
```

D.4. Removing all non - english words

```
from autocorrect import Speller

from nltk.tokenize import word_tokenize

def removal(text):

    spell = Speller(lang='en')

    texts = spell(text)

    return ' '.join([w.lower() for w in word_tokenize(text)])
```

D.5. Removing all the punctuations except ‘.’ and ‘,’

```
punctuations = "!()-[]{};:'<>/?@#$$%^&*~'"

for x in punctuations:

    text = text.replace(x, "")

text
```

E. CONVERTING THE EXTRACTED TEXT INTO AUDIO

Converting the cleaned text into audiobook using pyttsx3

```
mytext = text

audiobook = pyttsx3.init()

# change_voice(audiobook, "nl_BE", "VoiceGenderFemale")

audiobook.save_to_file(mytext, 'D:\\FinalYearProject\\PDFtoAudiobook\\uploads\\'+
oname)

audiobook.runAndWait()

audiobook.stop()
```

F. UPLOADING THE GENERATED AUDIOBOOK ON THE INTERFACE

```
@app.route('/output')  
  
def downloadFile ():  
    global oname  
    path = "D:\\Final Year Project\\PDFtoAudiobook\\uploads\\"+ oname  
    return send_file(path,as_attachment=True)
```


8. OUTPUT

a. INTERFACE

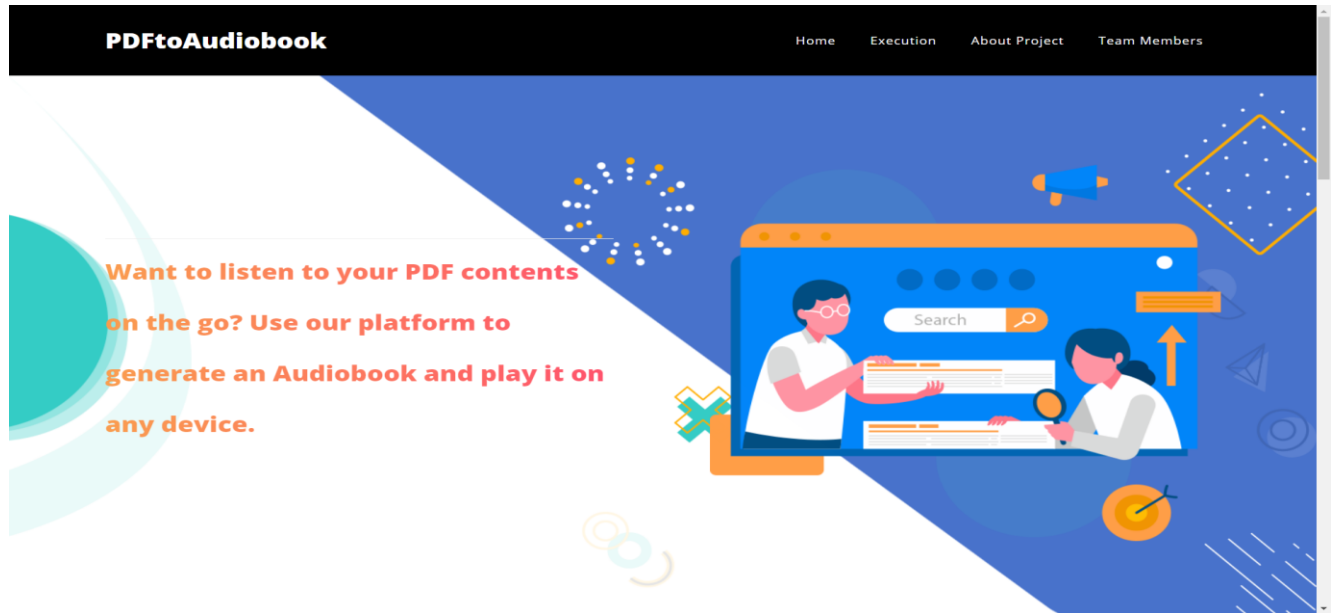


FIG 8.1 interface look

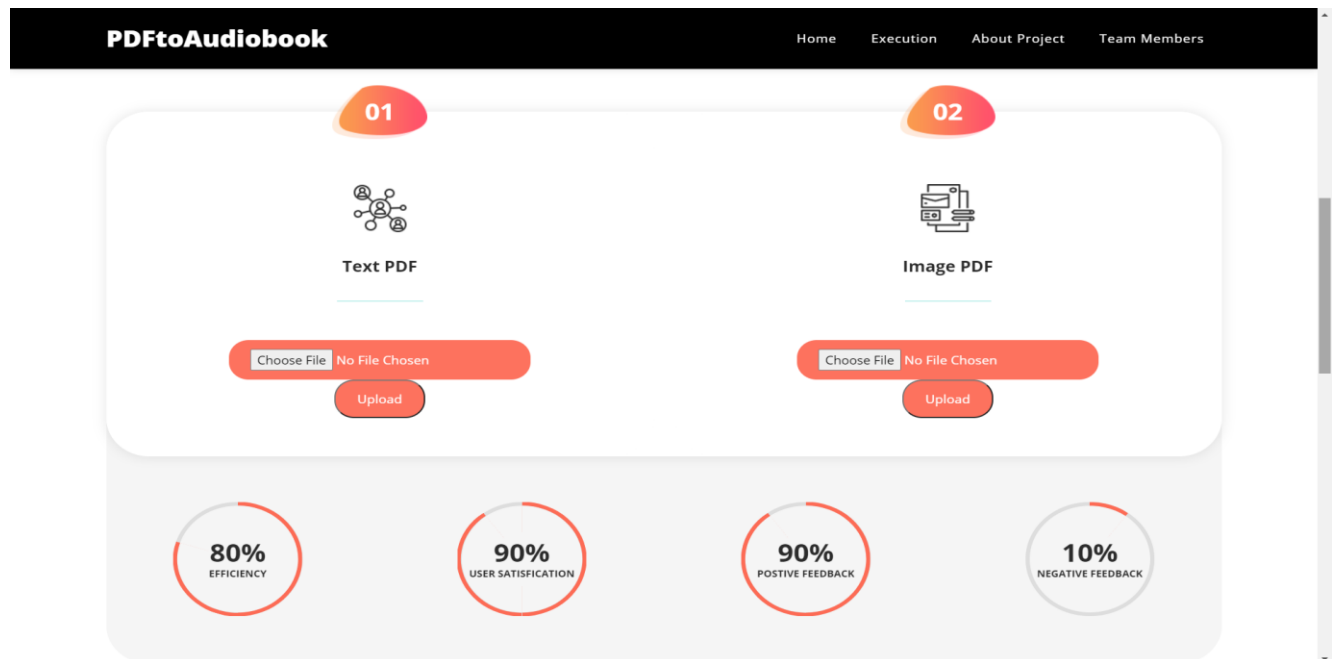


FIG 8.2 continuation of interface look

b. TEAM MEMBER ICON ON THE TOP RIGHT CORNER

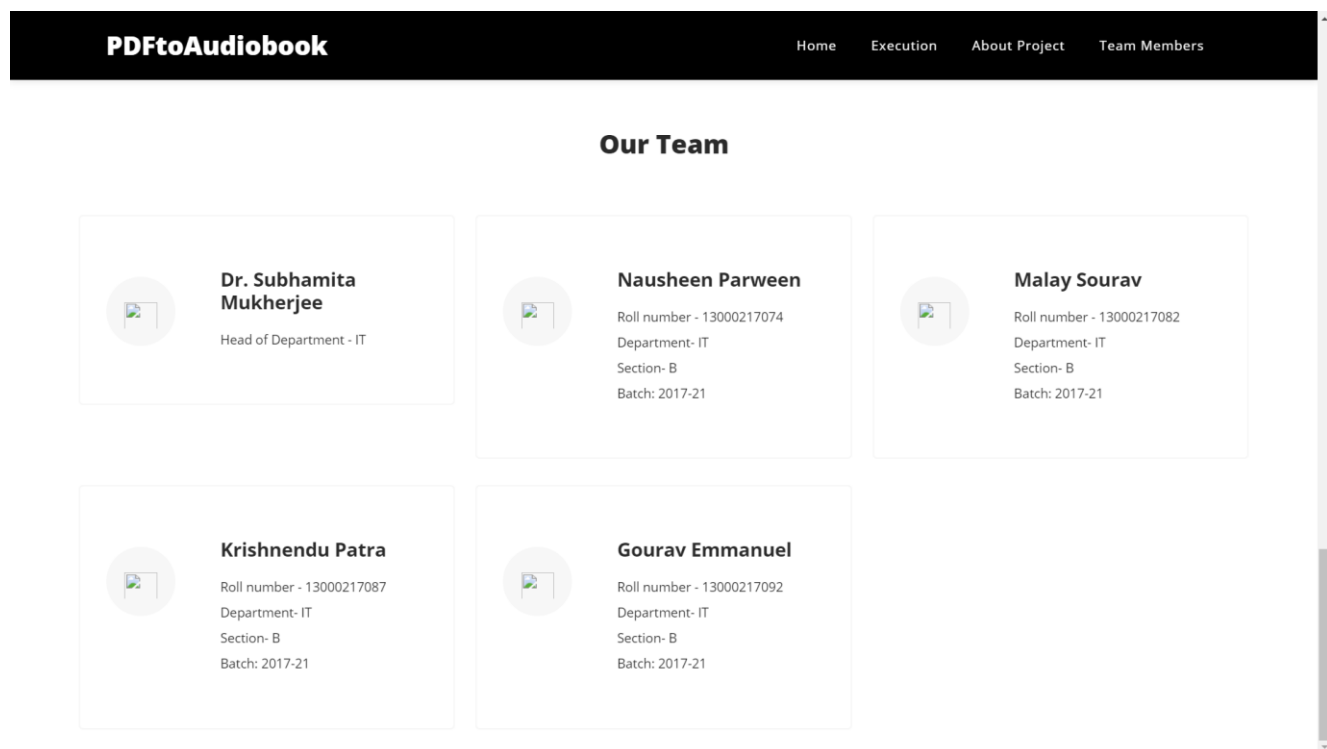
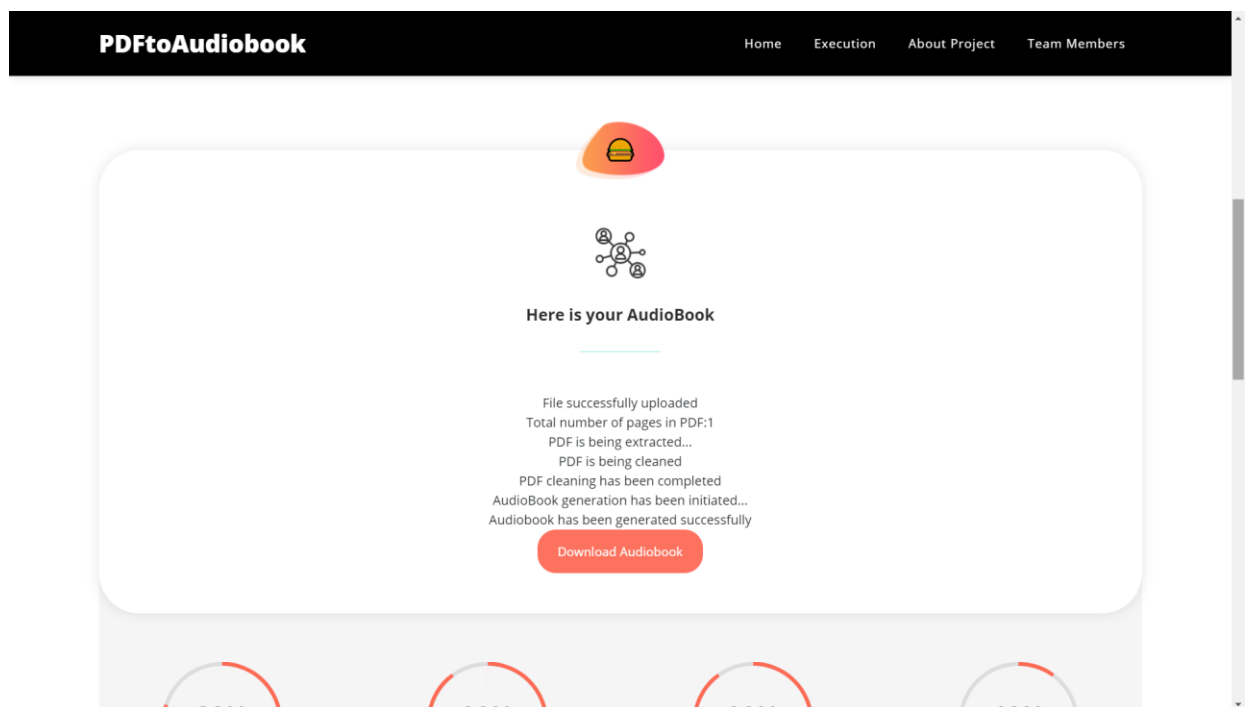


FIG 8.3. SCREENSHOT OF THE INTERFACE WHEN YOU CLICK ON TEAM MEMBERS

c. WHEN THE PDF IS GENERATED



9. LIBRARIES USED AND THEIR WORKING

- **PyPDF2:** PyPDF2 is a pure-python PDF toolkit originating from the pyPdf project. It can extract data from PDF files, or manipulate existing PDFs to produce a new file. Here, in our project we are implementing this library to read the PDF and open it to perform multiple applications on it
- **Autocorrect:** Autocorrect library is used to correct the spellings of words present in any of the files. Here, we are using this library to rectify the words of a PDF so that the user doesn't face issues with the pronunciation when the Audiobook is generated.
- **Texttract:** It is a document analysis service that detects and extracts printed text, and handwriting, structured data, such as fields of interest and their values, and tables from images and scans of documents. Here, we are using the library to extract the texts of each page and then add in the string variable which will be later used to generate the Audiobook. Texttract uses parser to extract the code using multiple modules which run in the background. The main functionality of the parser is to encode and decode the texts extracted and add in the destination file or variable.
- **NLTK Tokenize:** This tokenizer divides a text into a list of sentences by using an unsupervised algorithm to build a model for abbreviation words, collocations, and words that start sentences. Here, we have used this to convert non-English words to English words.

- **FLASK:** We have used FLASK to package our code for deployment. We have integrated our code in one file along with some additional libraries and implementations. After the complete execution, the user will get an option to download the audiobook and play it on any multimedia device.
- **Pdfplumber:** This library can be used for different purpose while extracting text. If we want to extract text or tabular data from any document, we can use this library.
- **Pyttsx3:** It is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. It is a very easy to use tool which converts the entered text into speech. The pyttsx3 module supports two voices first is female and the second is male which is provided by “sapi5” for windows.

7.Conclusion:

In our project, we achieved our target of providing a platform to the user to generate an audiobook from a PDF. We did this so by using python libraries to complete sub-tasks such as extracting the text, cleaning them and finally generating an audiobook.

During the project, we realised that we need to extract text from images as well because some of the PDFs might be of picture books. Therefore, we decided to provide 2 different options to the user, one being to generate audiobook from Text PDF and the other one from Image PDF. In order to achieve this, we created two routes in the interface, so that both the operations are carried out separately.

We designed a website to provide an interface to the user, which contains sub-sections such as execution, project description and team member details. Finally, when the execution gets completed, the user will have an option to download an Audiobook and keep it on their device.

References

1. Gilbert, Williams & McLaughlin, 1996:- Use of assisted reading to increase correct reading rates and decrease error rates of students with learning disabilities. *Applied Behavior Analysis*, 29(2), 255-257.
2. Wolfson, 2008
3. Whittingham, Huffamn, Christensen & McAllister, 2012
4. Yarosz & Barnett, 2001
5. COABE, 2013
6. Binder & Lee, 2012;
7. Hudson, Lane & Pullen, 2005
8. Hasbrouck, 2006
9. Miller, D. (n.d.). Pdf to mp3 converter. Retrieved March 09, 2021, from http://www.cs.unc.edu/Research/assist/et/projects/mp3/final_pdf2mp3.html
10. Hamiti, M., & Dika, A. (2010). Learning opportunities through generating speech from written texts. *Procedia - Social and Behavioral Sciences*, 2(2), 4319-4324. doi:10.1016/j.sbspro.2010.03.686
11. Isewon, I., Oyelade, J., & Oladipupo, O. (2014). Design and implementation of text to speech conversion for visually impaired people. *International Journal of Applied Information Systems*, 7(2), 25-30. doi:10.5120/ijais14-451143
12. Chan, N. (n.d.). Text-to-speech conversion. doi:10.5353/th_b3120958
13. M. (2020). PDF TO AUDIO CONVERTOR. *PDF TO AUDIO CONVERTOR*, 02(12 DEC 2020), 563-566.