

Documentation: Docx to PDF Converter Application

Pareesh Sharma - 102116092 - CSE

Project for - RapidFort: Campus Recruitment Drive

Project Overview :

The Docx to PDF Converter is a web-based application built with Flask that allows users to upload .docx files, extract metadata, and convert them to .pdf. The converted PDF can then be downloaded via a generated link.

Features :

Upload .docx File:

Users can upload .docx files via a simple web interface.

Only .docx files are accepted.

Metadata Extraction:

Extracts metadata like the number of paragraphs from the uploaded .docx file.

PDF Conversion:

Converts the .docx file into a .pdf using pdfkit and wkhtmltopdf.

Download PDF:

Users can download the converted PDF via a provided link.

Technologies Used :

Backend:

Flask: Python microframework for handling requests and rendering templates.

Python Libraries:

docx: Extracts metadata and content from .docx files.

pdfkit: Converts HTML to PDF.

PyPDF2: Handles PDF operations (optional in the current version).

Frontend:

HTML, Bootstrap, JavaScript.

System Tools:

wkhtmltopdf: Required for HTML-to-PDF conversion via pdfkit.

Project Structure :

doc-to-pdf/

â”œâ”€â”€ app.py # Main Flask application

â”œâ”€â”€ templates/ # HTML templates

â””, â””â”€â”€ index.html # Frontend interface

â”œâ”€â”€ uploads/ # Temporary storage for uploaded files

â””â”€â”€ requirements.txt # Python dependencies

Frontend Details

index.html

Provides a simple form for uploading .docx files.

Sends a POST request to the /upload endpoint using JavaScript.

Dynamically displays metadata and a download link after successful upload.

Backend Logic

1. File Upload

Validates the uploaded file:

Ensures the file is present and has a .docx extension.

Saves the file in the uploads/ directory.

Extracts metadata using the python-docx library.

2. Conversion Process

Converts .docx to HTML using python-docx.

Converts HTML to PDF using pdfkit and wkhtmltopdf.

3. Error Handling

Returns appropriate error messages for:

Missing file.

Invalid file format.

Conversion failures.

Improvements to Consider

Password Protection and UI/UX:

Use PyPDF2 to add password protection to the generated PDF.

Styling and Formatting:

Improve the appearance of the converted PDF by styling the HTML input.

Asynchronous Processing:

Handle large files asynchronously to avoid blocking the server.

Example Workflow

Open the application in a web browser (<http://localhost:5000>).

Upload a .docx file.

View extracted metadata and download the converted PDF.

Sample Response

Metadata:

json

Copy code

```
{  
  "paragraph_count": 12,  
  "filename": "sample.docx"  
}
```

Download Link: </download/sample.pdf>

Screenshot :

Conclusion :

The Docx to PDF Converter provides a lightweight and efficient solution for converting .docx files to .pdf format. It can be enhanced further to support additional features like password protection, better error handling, and scaling for larger file sizes.