PDF to MS Word Replication Documentation

1. Objective

The objective of this assignment is to recreate a given legal PDF document into an MS Word document using Python, ensuring that the **content, structure, layout, spacing, and alignment** closely match the original PDF.

The solution focuses on **programmatic document generation** rather than manual editing or auto-conversion.

2. Problem Understanding

The provided PDF is a **legal mediation application form** containing:

- Center-aligned headers
- A structured, multi-row table
- Two parties (Applicant and Opposite Party)
- Conditional placeholders for addresses
- A "Details of Dispute" section

The PDF is primarily **table-driven**, so accurate replication requires recreating the table structure in MS Word.

3. Approach

Step 1: PDF Analysis

- Carefully reviewed the PDF to understand:
  - Number of columns and rows
  - Section headers and numbering
  - Field labels and placeholders
  - Overall form layout
- Identified that the entire form is built as **one main table**.

Step 2: Tool Selection

- **Python 3.x**
- **python-docx** for creating and formatting Word documents
- **Google Colab** for development and execution

Step 3: Document Creation

- Created the Word document from scratch using `python-docx`
- Added centered headers for the title and authority details
- Built a **three-column table** to match the PDF structure:
  - Column 1: Serial number
  - Column 2: Field label
  - Column 3: Field value / placeholder

- Used merged cells where required for section headers
- Preserved placeholders exactly as shown in the PDF (e.g. `{{client_name}}`, `{% if address1 ... %}`)

Step 4: Layout Accuracy

- Used table borders (`Table Grid`) to maintain form appearance
- Maintained section order and spacing
- Ensured all headings and labels match the PDF text

4. Features of the Solution

- Exact structural replication of the PDF
- Fully programmatic MS Word generation
- Editable and reusable placeholders
- Clean and readable output document
- Easy to extend for dynamic data filling

5. How to Run the Code

1. Open the Google Colab notebook
2. Install dependencies:
3. `pip install python-docx`
4. Run the notebook cells
5. The output file `Mediation_Application_Form_FINAL.docx` will be generated
6. Download and verify the document

6. Output

The generated MS Word document:

- Matches the PDF's layout and structure
- Contains all required sections
- Is clearly readable and professionally formatted
- Is suitable for legal and official use

7. Future Enhancements (Optional)

- Dynamic population of placeholders using user input or APIs
- Integration with Google Drive or Google Docs API
- Automated PDF-to-DOC comparison for validation

8. Conclusion

This implementation successfully recreates the provided PDF into an MS Word document using Python while maintaining layout accuracy, structural integrity, and readability. The approach demonstrates practical document automation skills and attention to detail.