### **Hackathon Assignment: Detecting and Extracting Tables from PDFs**

#### **Objective:**

Develop a tool to detect and extract tables from system-generated PDFs without using Tabula, Camelot, or converting the PDF into images. The extracted tables should then be stored in an Excel sheet. The tables may have borders, no borders, and irregular shapes.

#### **Requirements:**

1. **Input:**
   * System-generated PDFs with tables.
   * Tables may have borders, no borders, or irregular shapes.
2. **Output:**
   * Excel sheets with the extracted tables.
3. **Constraints:**
   * Do not use Tabula or Camelot.
   * Do not convert PDFs to images for table extraction.

#### **Guidelines:**

1. **Language and Libraries:**
   * Use any programming language of your choice.
   * You may use PDF processing libraries.
   * For Excel manipulation, you can use libraries like openpyxl or pandas.
2. **Table Detection:**
   * Identify tables within the PDF.
   * Handle tables with borders and without borders.
   * Handle irregular-shaped tables.
3. **Table Extraction:**
   * Extract the table content accurately, maintaining the rows and columns structure.
   * Consider edge cases such as merged cells, multi-line cells, and irregular shapes.
4. **Storage:**
   * Store the extracted tables in an Excel sheet, maintaining the integrity of the table structure.
   * Each PDF should result in an individual Excel file or a consolidated file with multiple sheets.
5. **Documentation:**
   * Provide clear instructions on how to run your tool.
   * Include any dependencies or setup steps required.

#### **Evaluation Criteria:**

1. **Accuracy:**
   * Correct detection and extraction of tables.
   * Handling of both regular and irregular-shaped tables.
2. **Efficiency:**
   * Performance of the tool in processing multiple PDFs.
   * Time taken to extract and store tables in Excel sheets.
3. **Code Quality:**
   * Readability and maintainability of the code.
   * Proper documentation and comments.
4. **User Experience:**
   * Ease of use.
   * Clear error handling and messages.
5. **Innovation:**
   * Creative approaches to detecting and extracting tables.
   * Any additional features or improvements beyond the basic requirements.

#### **Submission:**

* Submit your source code along with a README file.
* Include sample PDFs and the corresponding output Excel files for testing purposes.
* Provide a short presentation or demo video explaining your solution.

#### **Example Workflow:**

1. **Loading PDF:**
   * Use a PDF library to read and parse the PDF content.
2. **Table Detection:**
   * Implement algorithms to identify table structures within the PDF text and layout.
3. **Table Extraction:**
   * Extract the table data, considering various table formats and structures.
4. **Excel Export:**
   * Write the extracted table data to an Excel sheet using an Excel manipulation library.
5. **Validation:**
   * Compare the extracted tables with the original tables in the PDF for accuracy.