TASK 1 DOCUMENTATION

Libraries Used:

1. **PyMuPDF (fitz)**: Used for extracting text and images from PDF files.
2. **pdfminer**: Used as an alternative for text extraction from PDF files.
3. **Tika**: Used as another alternative for text extraction, especially for PDFs with complex layouts.
4. **pytesseract**: Utilized for Optical Character Recognition (OCR) to extract text from images.
5. **PIL (Python Imaging Library)**: Used to process and handle images for OCR.
6. **NLTK (Natural Language Toolkit)**: Employed for text preprocessing tasks such as sentence segmentation and tokenization.
7. **Google Generative AI (Gemini API)**: Integrated for advanced text generation and language understanding capabilities.

Preprocessing Steps:

8. **Text Cleaning**:
    - Removal of irrelevant characters, whitespace, and formatting using regular expressions.
    - Retention of specific characters from a whitelist (`WHITELIST`) to preserve relevant symbols.
9. **Sentence Segmentation**:
    - Text is divided into individual sentences using NLTK's `sent_tokenize()` function.
10. **Tokenization**:
    - Sentences are broken down into individual words or sub-word units using NLTK's `word_tokenize()` function.

OCR Engine Selection and Evaluation:

- **OCR Engine**: Tesseract OCR was chosen for its accuracy and robustness in recognizing text from images.
- **Evaluation Process**:
    - The OCR engine selection was based on:

    i. **Accuracy**: Tesseract OCR has demonstrated high accuracy in recognizing text from images, even in complex scenarios.
    ii. **Community Support**: Tesseract OCR is open-source and has a large community, ensuring continuous development and improvement.

- Evaluation of OCR performance was conducted through:
    - **Testing on Sample Documents**: Various PDF documents containing images with text were used to evaluate Tesseract OCR's accuracy and performance.
    - **Comparison with Other Engines**: Tesseract OCR was compared with alternative OCR engines to assess its performance in extracting text from images.

Conclusion:

The chosen libraries and preprocessing steps enable efficient and accurate conversion of PDF documents into machine-readable text. Integration with the Google Generative AI (Gemini API) enhances the document understanding process, allowing for advanced text generation and analysis capabilities. Tesseract OCR, with its high accuracy and community support, serves as a reliable tool for extracting text from images within the document conversion pipeline.