Warm-up Task: Web Builder using PaddleOCR & ERNIE

Objective:

The goal of this warm-up task is to demonstrate an end-to-end pipeline that converts PDF documents into a deployable web page using OCR, document understanding, and large language models.

Overall Workflow:

1. Input PDF Document

A PDF document is provided as input. This document may contain structured text, headings, tables, and layout information.

2. Text and Layout Extraction using PaddleOCR-VL

PaddleOCR-VL is used to extract textual content along with layout-level information such as paragraphs, headings, lists, and structural blocks from the PDF pages.

The output of this stage is a structured representation of the document content.

3. Conversion to Markdown

The extracted text and layout information are converted into clean, well-structured Markdown format.

Headings, bullet points, code blocks, and sections are preserved to maintain document readability and hierarchy.

4. Web Page Generation using ERNIE Model

The Markdown content is passed to the ERNIE language model.

ERNIE is used to generate a complete HTML-based web page from the Markdown input.

This includes semantic structuring, section organization, and content refinement suitable for a web interface.

5. Static Website Deployment

The generated HTML content is prepared as a static website.

The website is deployed using GitHub Pages, enabling public access through a hosted URL.

Dataset Context:

The dataset used for this workflow has been web-scraped and curated from Arduino official documentation and Arduino user forums.

This ensures technical accuracy and real-world relevance for the generated content.

Outcome:

The final output is a fully functional static web page generated automatically from a PDF document, demonstrating OCR, document understanding, language modeling, and deployment integration.