AIPHAISTOS Project Overview

Introduction

AIPHAISTOS is a backend-driven AI-powered document question answering system designed to understand technical, architectural, and contractual PDFs. Led by Sanjay Gokhale under the guidance of Dr. Stefan Heinemann, this project focuses on building a FastAPI-powered backend with semantic search, hybrid vector + keyword retrieval, and LLM-based answering capabilities.

Features Implemented

- PDF upload, parsing, chunking, and semantic vector embedding
- FAISS-based similarity search with Whoosh BM25 hybrid retrieval
- Blueprint OCR and YOLO-based layout parsing for engineering drawings
- Multiple LLM integrations (MiniLM, Mistral, Yi, TinyLLaMA, Zephyr via Ollama)
- Local reranking using OpenAI and Zephyr for better relevance
- Custom scoring, deduplication, and fallback strategies for robust answers
- Swagger UI for easy endpoint interaction

Architecture Overview

[Insert architecture diagram image here showing PDF -> Chunking -> Embedding -> FAISS+BM25 -> LLM]

Demo Screenshots (UI Images)

Screenshot 1: Upload PDF

Screenshot 2: Query Interface

Screenshot 3: Answer Response

Challenges Faced

Despite successful retrieval, the LLM-generated answers were often too generic or contextually weak. Blueprint files had noisy OCR output, and long documents suffered from token and

embedding dilution. Answer precision lacked ChatGPT-level clarity and relevance due to LLM limitations on local CPUs.

Additionally, training or fine-tuning even small models like Zephyr requires a significant amount of domain-specific data and GPU-based computational power. This is currently a major bottleneck. To scale the project beyond local limitations, we need access to high-performance GPU infrastructure for fine-tuning and inference. The domain-specific nature of technical drawings, architectural plans, and contractual documents makes this task both necessary and urgent.

Future Plan

1. Fine-tune smaller LLMs (Zephyr, TinyLLaMA) using domain-specific data via LoRA
2. Collect high-quality technical, architectural, and contract datasets
3. Utilize high-GPU compute (cloud or lab) to train or fine-tune models
4. Improve blueprint parsing with better image-text alignment
5. Deliver a ChatGPT-style semantic QA experience across all PDF types