AI Inference Platform

Modular, pluggable AI inference platform with support for declarative flow definitions, multimodal inputs, approval-based async workflows, and fast LLM inference (local and remote).

📦Features

- 🧩Modular flow design (each AI capability lives in its own folder)
- Config-driven API registration
- YAML-based DSL for defining flow steps
- 🎙️Pause/resume flow execution using Redis + callback
- Multimodal input support (image, PDF, audio, text)
- LLM support via LangChain (OpenAI, Ollama, etc.)
- Auto-generated catalog endpoint
- 🧨CLI tools for local flow execution and validation

🗂️Folder Structure

```
ai_inference_platform/
├── main.py               # FastAPI app
├── config.yaml           # Lists enabled flows and API prefixes
├── flow_registry.py      # Loads and validates flows from /flows
├── router_factory.py     # Dynamically registers routers
│
├── core/
│   ├── base_flow.py      # Abstract base class for flows
│   ├── schema.py         # Pydantic schemas for validation
│   ├── utils.py          # Common utilities
│   └── state_store.py    # Redis storage for paused flows
│
├── flows/
│   ├── ocr_po/
│   │   ├── flow.py
│   │   ├── model.py
│   │   ├── prompt.txt
│   │   ├── meta.yaml
│   │   └── dsl.yaml
│   └── audio_transcribe/
│       └── ...
│
├── cli/
│   ├── run_flow.py       # CLI for running flows
│   └── validate_flows.py # Flow and metadata validation
│
```

```
└── docs/
    └── architecture.md   # Optional architecture diagram
```

🗜️ Quick Start

```
# Start the FastAPI server
uvicorn main:app --reload

# Run a flow manually
python cli/run_flow.py ocr_po --file invoice.pdf

# Validate all flows
python cli/validate_flows.py
```

🔦 Async Callback Flow

1. Flow pauses at an `approval` step
2. Saves state in Redis under `flow:{flow_id}`
3. Sends request to `approval_api` with a callback URL
4. External system hits `/callback/{flow_id}` with approval result
5. Flow is resumed from next step and finished

🛠️ Supported DSL Step Types

- `ocr` : Run OCR on image/pdf
- `llm` : Run LLM prompt (OpenAI, Ollama)
- `combine` : Merge values into a templated input
- `approval` : Trigger external approval request with callback

🔗 Dependencies

- Python 3.10+
- FastAPI
- LangChain
- Transformers
- Ollama (optional)
- Redis
- pdf2image, pytesseract (optional OCR fallback)

📖License

MIT License (or org-specific)

For more information, see the full PRD in `docs/` or contact the project lead.