



AI Modules for FastAPI Project

This module provides core AI functionalities to be integrated into a FastAPI backend. It includes LLM loading, PDF processing, and answer generation using a RAG (Retrieval-Augmented Generation) pipeline.



Model Configuration

Component	Description
LLM Model	Vicuna 7B - lmsys/vicuna-7b-v1.5
Embeddings	bkai-foundation-models/vietnamese-bi-encoder



AI Functions Available

There are **3 main functions** that the backend (BE) team can use:

- `load_llm()`
 - Loads and returns the LLM pipeline (with quantization and tokenizer setup).
- `process_pdf(llm, pdf_path, embeddings)`
 - Accepts a PDF file path, performs semantic chunking, creates a retriever for question answering.
- `generate_answer(rag_chain, question)`
 - Uses a LangChain `rag_chain` pipeline to answer a question based on retrieved context.



Testing the AI Functions

1. Run the **standalone AI tests**

This will test your AI logic independently:

```
python -m ai_modules.test_ai_modules
```

2. Run the **FastAPI-integrated AI tests**

This will test AI functions as part of the API workflow:

```
python -m ai_modules.test_ai_modules_with_fastAPI
```



Folder Structure (Simplified)

```
project/
├── ai_modules/
│   ├── __init__.py
│   ├── load_llm.py           # Contains load_llm()
│   ├── pdf_processor.py     # Contains process_pdf()
│   ├── generate.py          # Contains generate_answer()
│   ├── test_ai_modules.py   # standalone AI function test
│   └── test_ai_modules_with_fastAPI.py # FastAPI-integrated AI tests
├── backend/
│   └── main.py
```