Excel Processor API using FastAPI and LLMs

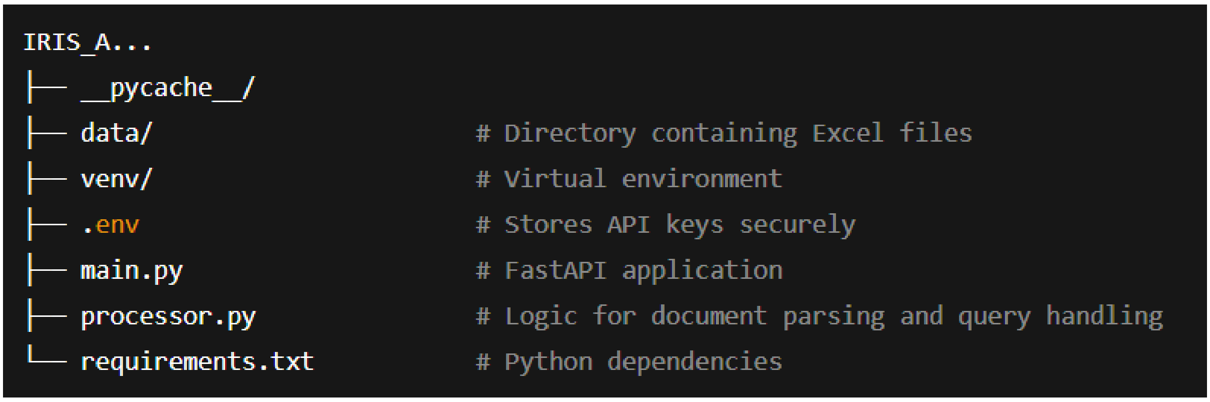
Project Overview:

The **Excel Processor API** is a FastAPI-based application designed to intelligently parse and extract information from Excel spreadsheets using large language models (LLMs). It enables users to:

- List all tables detected in an Excel file.
- Retrieve row names from a specific table.
- Calculate the sum of numeric values in a selected row.

This project uses **LlamaParse** for parsing .xls files and **Groq's LLaMA model** to process natural language queries semantically.

Project Structure:



```
IRIS_A...
├── __pycache__/
├── data/                   # Directory containing Excel files
├── venv/                   # Virtual environment
├── .env                    # Stores API keys securely
├── main.py                 # FastAPI application
├── processor.py            # Logic for document parsing and query handling
└── requirements.txt        # Python dependencies
```

Core Technologies:

Component	Technology Used
Backend Framework	FastAPI
LLM Provider	Groq LLaMA
File Parser	LlamaParse
Embedding Model	BAAI/bge-small-en-v1.5
Vector Index	LlamaIndex
Deployment Tool	Uvicorn

Potential Improvements

1. **Support for Additional Excel Formats**
   Currently, the application supports .xls files. Adding compatibility for .xlsx and .csv formats would enhance versatility and broaden use cases.
2. **Advanced Data Operations**
   Extend functionality to:
   - Compute averages, min/max values, standard deviation.
   - Compare financial metrics across tables or rows.
   - Extract charts or graphical summaries.
3. **UI Integration**
   Developing a frontend interface (e.g., using React or Streamlit) would allow users to upload files, select tables, and view results interactively without needing API knowledge.

Missed Edge Cases:

The model is unable to parse duplicate table names containing similar data. FAISS embeddings could have helped more. This implementation does not work well enough with unorganized data where multiple tables are present in a single sheet. Preparing multiple csv files for different tables using LLMs only could have resulted in much more accurate parsing.