**Bugs Found and How I Fixed Them**

| **File** | **Bug / Issue** | **Fix Applied** |
| --- | --- | --- |
| agents.py | llm = llm undefined and circular | Replaced with explicit initialization: llm = ChatOpenAI(...) using OpenAI API key from .env |
| agents.py | Exaggerated, non-professional agent roles | Rewrote agent role, goal, and backstory to reflect a real-world general physician |
| tools.py | langchain.document\_loaders import failed | Added missing import: from langchain.document\_loaders import PDFLoader |
| tools.py | SerperDevTool import error | Removed unused SerperDevTool since internet search wasn't needed for PDF analysis |
| tools.py | async def read\_data\_tool unused correctly | Refactored to a normal method and ensured proper call from CrewAI task |
| task.py | Unrealistic or fake-sounding task description | Rewritten with an ethical, clinically sound task description and expected output |
| main.py | Uploaded file not used in agent logic | Passed correct file\_path to CrewAI input dictionary |
| main.py | Missing directory creation | Added os.makedirs("data", exist\_ok=True) before saving uploaded file |
| main.py | File not deleted after use | Added cleanup logic using os.remove(file\_path) in finally block |
| requirements.txt | Conflicting pinned versions caused install failure | Switched to flexible install for compatibility across Python versions (esp. Python 3.12) |
| Python 3.12 | Incompatible with FastAPI + Pydantic v1 | Recommended downgrading to Python 3.10 or 3.11 for full compatibility |

**Setup and usage instructions API documentation**

from docx import Document

# Create a Word document

doc = Document()

doc.add\_heading('Blood Test Report Analyzer', 0)

doc.add\_heading('⚙️ Setup and Usage Instructions', level=1)

doc.add\_paragraph("""

Follow these steps to install dependencies, configure your environment, and run the application locally.

""")

doc.add\_heading("1️⃣ Clone the Repository", level=2)

doc.add\_paragraph("""

git clone https://github.com/Vyom321/blood-test-analyzer.git

cd blood-test-analyzer

""")

doc.add\_heading("2️⃣ Create and Activate a Virtual Environment (Optional but Recommended)", level=2)

doc.add\_paragraph("Windows:\npython -m venv venv\nvenv\\Scripts\\activate")

doc.add\_paragraph("macOS/Linux:\npython3 -m venv venv\nsource venv/bin/activate")

doc.add\_heading("3️⃣ Install Required Dependencies", level=2)

doc.add\_paragraph("Use a flexible install to avoid version conflicts:")

doc.add\_paragraph("""

pip install fastapi uvicorn crewai crewai-tools langchain openai PyPDF2 python-dotenv pydantic==1.10.13

""")

doc.add\_paragraph("Or install from requirements.txt:")

doc.add\_paragraph("pip install -r requirements.txt")

doc.add\_heading("4️⃣ Configure Environment Variables", level=2)

doc.add\_paragraph("""

Create a .env file in the root of your project and add your OpenAI API key:

OPENAI\_API\_KEY=your-openai-api-key-here

""")

doc.add\_heading("5️⃣ Run the FastAPI Server", level=2)

doc.add\_paragraph("uvicorn main:app --reload\nServer: http://127.0.0.1:8000")

doc.add\_heading("6️⃣ Access Interactive API Docs", level=2)

doc.add\_paragraph("""

Swagger UI: http://127.0.0.1:8000/docs

ReDoc: http://127.0.0.1:8000/redoc

""")

doc.add\_page\_break()

# API Documentation

doc.add\_heading('📡 API Documentation', level=1)

doc.add\_heading('📤 POST /analyze', level=2)

doc.add\_paragraph("Upload a blood test report PDF and get a summarized analysis by the CrewAI-powered doctor agent.")

doc.add\_heading("🔸 Request Format", level=3)

table = doc.add\_table(rows=1, cols=4)

hdr\_cells = table.rows[0].cells

hdr\_cells[0].text = 'Parameter'

hdr\_cells[1].text = 'Type'

hdr\_cells[2].text = 'Required'

hdr\_cells[3].text = 'Description'

row\_cells = table.add\_row().cells

row\_cells[0].text = 'file'

row\_cells[1].text = 'UploadFile (PDF)'

row\_cells[2].text = '✅ Yes'

row\_cells[3].text = 'The blood test report in PDF format'

row\_cells = table.add\_row().cells

row\_cells[0].text = 'query'

row\_cells[1].text = 'string'

row\_cells[2].text = '❌ Optional'

row\_cells[3].text = 'A custom question (e.g., "Summarize my lipid profile")'

doc.add\_heading("🔸 Default Behavior", level=3)

doc.add\_paragraph('If query is not provided, it defaults to:\n"Summarize my Blood Test Report"')

doc.add\_heading("🔸 Sample cURL Request", level=3)

doc.add\_paragraph("""

curl -X POST http://127.0.0.1:8000/analyze \\

-F "file=@blood\_test\_report.pdf" \\

-F "query=Explain liver panel results"

""")

doc.add\_heading("🔸 Successful Response", level=3)

doc.add\_paragraph("""

{

"status": "success",

"query": "Explain liver panel results",

"analysis": "Your liver enzymes ALT and AST are within normal limits...",

"file\_processed": "blood\_test\_report.pdf"

}

""")

# Sve the Word document

word\_path = "/mnt/data/Setup\_and\_API\_Documentation.docx"

doc.save(word\_path)

word\_path