Blood Test Report Analyzer - Bug Analysis Report

Project Overview

This is a FastAPI-based blood test report analyzer using CrewAI framework. The application allows users to upload PDF blood test reports and get analysis from AI agents.

Critical Bugs Identified

1. main.py Issues

Bug 1: Syntax Error - Missing Closing Parenthesis

- **Location**: Line 14 in `run_crew()` function
- **Issue**: Missing closing parenthesis in Crew instantiation
- **Current Code**:

  ```
  medical_crew = Crew(
      agents=[doctor],
      tasks=[help_patients],
      process=Process.sequential,
  ```

- **Fix**: Add missing closing parenthesis

  ```
  medical_crew = Crew(
      agents=[doctor],
      tasks=[help_patients],
      process=Process.sequential,
  )
  ```

Bug 2: Indentation Error

- **Location**: Line 16-17 in `run_crew()` function
- **Issue**: `result` assignment and `return` statement have incorrect indentation
- **Fix**: Proper indentation after Crew instantiation

Bug 3: Missing Closing Brace

- **Location**: Line 44 in `analyze_blood_report()` function
- **Issue**: Missing closing brace for return statement
- **Fix**: Add proper closing brace after return dictionary

2. agents.py Issues

Bug 4: Undefined Variable Assignment

- **Location**: Line 9
- **Issue**: `llm = llm` assigns undefined variable to itself
- **Fix**: Import proper LLM or define it

  ```
  from langchain.llms import OpenAI
  llm = OpenAI()
  ```

Bug 5: Incorrect Parameter Name

- **Location**: Agent definitions
- **Issue**: Using `tool=` instead of `tools=`
- **Fix**: Change parameter name to `tools=`

Bug 6: Missing Closing Parentheses

- **Location**: All Agent definitions
- **Issue**: Missing closing parentheses for Agent constructors
- **Fix**: Add closing parentheses

Bug 7: Incorrect Tool Reference

- **Issue**: `BloodTestReportTool().read_data_tool` creates instance but references method incorrectly
- **Fix**: Create proper tool instance

3. tools.py Issues

Bug 8: Missing Import

- **Location**: Line 20
- **Issue**: `PDFLoader` used but not imported
- **Fix**: Add import statement

  ```
  from langchain.document_loaders import PDFLoader
  ```

Bug 9: Missing Self Parameters

- **Location**: All class methods
- **Issue**: Methods missing `self` parameter
- **Fix**: Add `self` as first parameter to all methods

  ```
  async def read_data_tool(self, path='data/sample.pdf'):
  ```

Bug 10: Unused Import

- **Location**: Line 4
- **Issue**: `from crewai_tools import tools` imported but unused
- **Fix**: Remove unused import

4. task.py Issues

Bug 11: Missing Closing Parentheses

- **Location**: All Task definitions
- **Issue**: Missing closing parentheses in Task constructors
- **Fix**: Add closing parentheses

Bug 12: Import Error

- **Location**: Line 3
- **Issue**: Importing `verifier` but it may not be properly defined
- **Fix**: Ensure verifier agent is properly defined in agents.py

Bug 13: Incorrect Tool Reference

- **Location**: Task tool assignments
- **Issue**: Using class method instead of instance
- **Fix**: Use proper tool instance

5. requirements.txt Issues

Bug 14: Missing Dependencies

- **Issue**: Missing critical dependencies like `langchain`, `pypdf`, etc.
- **Fix**: Add missing dependencies

Severity Classification

Critical (Application Won't Start)

- Bug 1: Syntax Error in main.py
- Bug 4: Undefined LLM variable
- Bug 8: Missing PDFLoader import

High (Runtime Errors)

- Bug 2: Indentation issues
- Bug 6: Missing closing parentheses
- Bug 9: Missing self parameters

Medium (Functionality Issues)

- Bug 5: Incorrect parameter names
- Bug 7: Incorrect tool references
- Bug 11-13: Task definition issues

Low (Code Quality)

- Bug 10: Unused imports
- Bug 14: Missing optional dependencies

Fix Priority Order

1. Fix syntax errors in main.py (Bug 1, 2, 3)
2. Define LLM variable in agents.py (Bug 4)
3. Add missing imports in tools.py (Bug 8)
4. Fix method signatures (Bug 9)
5. Fix agent definitions (Bug 5, 6, 7)
6. Fix task definitions (Bug 11, 12, 13)
7. Clean up unused imports (Bug 10)
8. Update requirements.txt (Bug 14)

Testing Strategy

1. **Syntax Validation**: Ensure all Python files have valid syntax
2. **Import Testing**: Verify all imports work correctly
3. **Class Instantiation**: Test that all classes can be instantiated
4. **API Endpoints**: Test FastAPI endpoints respond correctly
5. **File Processing**: Test PDF file upload and processing
6. **Agent Interaction**: Verify CrewAI agents work together

Recommendations

1. Use a Python linter (flake8, pylint) to catch syntax errors
2. Set up pre-commit hooks for code quality
3. Add comprehensive unit tests
4. Use type hints for better code documentation
5. Implement proper error handling and logging
6. Add input validation for API endpoints