**Module 1 - Dataset Loading**: Loads and validates all JSON files with comprehensive error handling and data verification.

**Module 2 - Tool Definitions**: Three core tools that:

* summarize\_patient\_record(): Extracts patient demographics, calculates age, and formats medical information with human-readable descriptions
* summarize\_policy\_guideline(): Retrieves policy coverage rules and formats them clearly
* check\_claim\_coverage(): Analyzes patient records against policy guidelines to determine coverage

**Module 3 - System Instruction**: Comprehensive prompt that defines the workflow sequence, decision criteria, and exact output format requirements.

**Module 4 - LLM Integration**: Uses LangChain's ChatOpenAI with configurable parameters optimized for consistent, factual responses.

**Module 5 - Agent Creation**: ReAct-style agent using LangGraph's create\_react\_agent() method with proper tool integration.

**Module 6 - Validation**: Processes validation records, analyzes results, and provides performance insights for agent improvement.

**Module 7 - Testing**: Generates final test responses and creates the required submission.csv file.

Key features of the implementation:

1. **Robust Error Handling**: Each module includes comprehensive error handling and logging
2. **Medical Code Translation**: Automatically converts ICD-10 and CPT codes to human-readable descriptions
3. **Age Calculation**: Dynamically calculates patient age from birth date and service date
4. **Structured Analysis**: Systematic evaluation of coverage criteria (age range, gender, diagnosis/procedure matching, preauthorization)
5. **Professional Output**: Generates responses in the exact format you specified with clear decisions and detailed reasoning

The system is designed to be modular, maintainable, and easily extensible. You can run individual components for testing or execute the full pipeline through the main() function.

To use this system:

1. Set your OpenAI API key as an environment variable
2. Ensure all JSON files are in the working directory
3. Run main() to execute the complete pipeline
4. Check the generated submission.csv for final results

The agent will systematically process each claim by analyzing patient demographics, medical codes, and policy requirements to make coverage determinations with detailed reasoning.