HLRD

Here is a comprehensive High-Level Requirements Document (HLRD) to support the technical documentation of your Model Documentation Agent:

High-Level Requirements Document (HLRD): Model Documentation Agent

1. Executive Summary

The Model Documentation Agent is an intelligent, CLI-based system designed to automate the generation of technical documentation for complex codebases—particularly financial model implementations. By combining traditional file system parsing with prompt-engineered interactions with LLMs (Anthropic Claude models), the agent creates structured, readable documentation artifacts (Markdown) with minimal human intervention. This HLRD defines the high-level functional, non-functional, and user requirements for its development and deployment.

---

2. Business Objectives

- **Reduce Manual Documentation Effort**: Automate the generation of accurate and comprehensive technical documentation, significantly reducing analyst and developer hours spent on manual write-ups.
- **Standardize Documentation Across Teams**: Promote consistency in the structure, depth, and style of model documentation within large organizations (e.g., financial institutions).
- **Enable Scalable Documentation Coverage**: Allow scaling to cover thousands of models or scripts across departments through automation and templating.

---

3. Scope

In-Scope Features

- Codebase ingestion from a folder structure.
- Multi-step documentation generation using LLMs (file summaries, hierarchical summaries, structured outline, section drafts).
- Markdown document generation using a user-specified or default JSON template.
- CLI orchestration and real-time monitoring of progress.
- Modular architecture supporting future plugin of new models, templates, or documentation logic.

Out-of-Scope (v1)

- Real-time collaborative editing.
- Non-Python codebase support.
- Formal integration with CI/CD pipelines (manual triggering only in v1).

---

4. Stakeholders

Role	Responsibility
Technical Documentation Teams	Consume output documentation for regulatory submissions.
Model Developers	Trigger documentation generation for their own codebases.
Risk & Compliance Officers	Verify generated documentation meets audit readiness.
Engineering/IT Admins	Install, configure, and maintain the tool on enterprise systems.

---

5. Functional Requirements

ID	Requirement Description
F-01	The system shall accept a codebase folder path via CLI.
F-02	The system shall load a documentation template from a JSON file.

F-03	The system shall summarize each relevant code file using an LLM.
F-04	The system shall generate a hierarchical overview of the entire codebase using LLM output.
F-05	The system shall use the summaries and the template to generate a documentation outline.
F-06	The system shall draft each section/subsection based on outline + summaries using LLM calls.
F-07	The system shall save the intermediate results in JSON and `.txt` format for auditability and inspection.
F-08	The system shall generate a final Markdown file matching the template structure.
F-09	The system shall allow user to toggle real-time monitoring of intermediate file creation.
F-10	The system shall retry failed LLM interactions up to a configurable number of times with exponential backoff.
F-11	The system shall store outputs in timestamped directories for traceability.

6. Non-Functional Requirements

ID	Requirement Description
NFR-01	
NFR-02	The system shall support API-based LLM communication using Anthropic Claude models.
NFR-03	The system shall complete a single documentation run (500–2000 lines of code) in under 10 minutes.
NFR-04	The system shall ensure sensitive API keys are never written to logs or committed to version control.
NFR-05	The system shall be modular and extensible to allow for new LLMs or output formats in future versions.
NFR-06	The system shall comply with enterprise documentation practices (e.g., file manifest, version history).
NFR-07	The output directory structure shall remain consistent and machine-readable.

7. User Stories

As a Model Developer:

- I want to point the agent at my project directory, so that it creates a documentation draft with minimal effort.
- I want to include a custom documentation template for consistency with my team's standards.

As a Compliance Officer:

- I want the final document to include metadata, a file manifest, and hierarchical summaries, so that I can validate the model's compliance posture.

---

8. Technical Constraints

- **LLM Backend**: Must use Anthropic Claude models via API; key management is done via `.env` files.
- **File Support**: Initially supports only `.py` files for summarization. Support for `.ipynb` or other languages will be planned for future versions.
- **Template Format**: Must be JSON and follow a hierarchical structure aligned with section/subsection headers.
- **Output Format**: Markdown only (`.md`). PDF/HTML conversions are out of scope for v1.

---

9. Assumptions

- The user has access to a valid Anthropic API key with sufficient quotas.
- The codebases to be documented are self-contained Python repositories and don't rely on dynamic imports or runtime introspection for understanding.
- Intermediate drafts can be inspected by users but are not editable mid-process.

---

10. Risk and Mitigation Strategy

Risk	Mitigation
LLM rate limits or outages	Built-in retry mechanism with exponential backoff
Code parsing limitations (e.g., very dynamic code)	Plan for AST parsing enhancements in future releases
Sensitive data leakage (e.g., API keys)	`.env` exclusion from version control; `clean_sensitive_info.py` utility
Output quality variance across LLM versions	Centralized configuration of model version; fallback prompts as backup

11. Future Considerations

- Extend to multiple code languages beyond Python (R, MATLAB, SQL, etc.)
- Generate diagrams (e.g., class hierarchies, call graphs) using AST or static analysis.
- Integrate with IDEs (e.g., VS Code plugin).
- Enterprise workflow integration via REST APIs or CI/CD triggers.
- Introduce feedback loop where users can comment and refine generated documentation before finalization.