**Project Citadel Overview**

*Based on Pydantic AI Intelligent Documentation Crawler & RAG Agent*

**Functional Perspective**

**Core Purpose:**  
Citadel enables automated, efficient, and intelligent ingestion of documentation from web resources (websites, txt/Markdown files, sitemaps) and supports advanced, LLM-powered semantic search and question answering via a user-friendly Streamlit web interface.

**Key Functionalities:**

- Automatic web crawling of documentation sites, txt/Markdown, or sitemap URLs

* Chunking of collected content by document structure for optimal retrieval
* Fast, parallel crawling and data deduplication
* Submission of content and metadata (headers, chunk index, source) into a vector database for semantic, context aware retrieval
* Streamlit interface to surface semantic search and question answering against the curated documentation
* Extensible scripts for easily deploying or customizing all stages of the workflow

**User Experience:**

* Simple command-line interface for ingestion and setup
* Instantly deployable web app for search and QA
* Flexible parameters for scaling batch sizes, parallelism, and chunk sizes

**Technical Perspective**

**Architecture & Components:**

* Python-based, modular codebase
* Utilizes Crawl4AI for deep and recursive web crawling
* Hierarchical chunker splits documentation by headers, then by size
* Vector storage and semantic search powered by ChromaDB
* Models and embeddings are flexible and can use OpenAI or user defined options
* Streamlit provides the frontend user interface for search and QA

**Integration/Scalability:**

* Designed to handle large corporate documentation, with batching and concurrency controls
* Easily swap out models or database config for enterprise scaling or integration
* Scripts and parameters allow tuning to fit dataset size and performance needs
* Pluggable architecture can extend to other vector DBs, LLMs, or UIs

**Deployment Stack:**

* Python 3.11+, OpenAI API, ChromaDB, Streamlit, Playwright (for crawling)
* Environment managed through .env files for easy configuration

**Project Status and Next Steps**

**Proof-of-Concept Achievements:**

* Demonstrated crawling, chunking, and vectorization on real-world documentation
* Working user-facing interface for instant, context-rich retrieval
* Modular, extensible codebase for rapid prototyping

**Preparedness for Next Phase:**

* Foundation established for enterprise-level onboarding, scalability, and integration
* Robust enough to test on expanded document sets and new model endpoints
* Code structure supports quick extension or deployment into production

**Next Milestones:**

* Harden for enterprise security and scale (fault-tolerance, error handling, user auth)
* Extend to additional DBs, LLMs, and potentially multi-lingual content
* Optimize memory, speed, and debugging for industrial scenarios

This summary should get all stakeholders up-to-speed on where Project Citadel is now and why its plugin-style, high-performance documentation ingestion & LLM-QA stack is a smart base for enterprise RAG deployments.