

Continue AI-Powered Chatbot for Quick Access to Jenkins Resources

Google Summer of Code Program 2026 Project Proposal

Harmit Jetani
harmitjetani@gmail.com
Harmit07

Project Abstract. This project enhances Jenkins by developing an AI-powered chatbot to provide users with intuitive access to documentation, plugins, and troubleshooting resources. Leveraging a robust Retrieval-Augmented Generation (RAG) pipeline, the chatbot will utilize LLMs to deliver accurate, context-aware assistance, significantly streamlining workflows and reducing the learning curve for Jenkins users.

Project Description.

1. The Problem Statement

Jenkins possesses a massive, highly detailed ecosystem of documentation, plugins, and community resources. However, navigating this vast amount of information can be overwhelming, especially for newer users. Traditional keyword search often falls short when users are trying to troubleshoot specific pipeline errors or configure complex plugins. There is a critical need for an intelligent system that understands the context of a user's query and retrieves exact, synthesized answers directly within the Jenkins environment.

2. Proposed Architecture (RAG Pipeline)

To solve this, I will build a Retrieval-Augmented Generation (RAG) chatbot directly integrated into Jenkins. The architecture consists of three main pillars:

Data Ingestion & Vector Storage: I will develop a pipeline to ingest Jenkins documentation (from jenkins.io, Antora sites, and plugin wikis). The text will be processed, chunked into semantically meaningful segments, converted into embeddings, and stored in a Vector Database (such as ChromaDB or FAISS) for highly efficient semantic search.

The Intelligence Engine (LangChain4j & LLM): Leveraging Java's ecosystem, I will utilize LangChain4j to manage the LLM orchestration natively within the backend. When a user submits a query, the system will perform a similarity search against the vector database, retrieve the most relevant documentation chunks, and pass them as context to the LLM (e.g., using OpenAI's API or a local Llama 3 model via Ollama) to generate a highly accurate, hallucination-free response.

User Interface: A lightweight, responsive JavaScript/TypeScript frontend embedded directly into the Jenkins UI, allowing users to ask questions without breaking their workflow.

3. Benefits to Jenkins

This project will drastically reduce the time users spend context-switching between Jenkins and external documentation tabs. By providing instant, context-aware answers and troubleshooting steps, it lowers the barrier to entry for beginners while significantly accelerating the workflow of experienced DevOps engineers.

Project Deliverables

- **Jenkins Chatbot Plugin:** A functional backend engine integrating LangChain4j for LLM orchestration natively within Jenkins.
- **RAG Data Pipeline:** An automated ingestion pipeline that parses, chunks, and embeds Jenkins documentation into a Vector Database for semantic search.
- **Frontend Interface:** An intuitive, responsive JavaScript/TypeScript chat UI embedded directly within the Jenkins dashboard.

- **Enterprise-Grade Testing:** Comprehensive JUnit tests, integration tests, and developer documentation to ensure long-term maintainability.

Proposed Schedule

During each phase, please make it clear whether code will be ready to be submitted for inclusion. Also ensure that your code will be available for review almost every day. Do not work long periods (more than 2-3 days) without submitting code to the git repository for your project. You should normally commit code to GitHub everyday or almost every day (perhaps except on weekends).

March 16 - March 31, 2026 (Application Period)

- Finalize the proposal architecture based on mentor feedback from the Discourse forum.
- Continue solving "good first issues" in the Jenkins ecosystem to deepen my understanding of the codebase.
- Research LangChain4j compatibility constraints with Jenkins core.

April 1 - April 30 (Acceptance Waiting Period)

- Build a standalone local prototype of the RAG pipeline (chunking sample Markdown files and querying them via an LLM).
- Study the Jenkins UI framework and existing plugin frontend conventions to plan the UI injection.

May 1 - May 24 (Community Bonding Period)

- Establish weekly communication channels and meeting times with mentors.
- Finalize the exact Vector Database and LLM endpoints to be used.
- Set up the project repository, CI/CD pipelines, and initial Jenkins plugin skeleton.

May 25 - July 6 (Standard Coding Period)

- Weeks 1-2 (Data Pipeline): Develop the ingestion module to parse documentation, chunk the text, and store embeddings.
- Weeks 3-4 (Backend Engine): Integrate LangChain4j into the plugin. Build the similarity search and retrieval logic to fetch context based on user prompts.
- Weeks 5-6 (Integration & Midterm Prep): Connect the retrieval logic to the LLM API. Implement prompt engineering to prevent hallucinations. Prepare the backend code for the Phase 1 Evaluation.

July 6 - August 24 (Work Period for Standard Route)

- Weeks 7-8 (Frontend UI): Develop the JavaScript/TypeScript chat interface and ensure it injects cleanly into the Jenkins UI.
- Weeks 9-10 (Refinement): Connect the frontend to the backend endpoints. Implement chat history tracking and context-aware follow-ups.
- Weeks 11-12 (Testing & Final Polish): Write comprehensive unit and integration tests. Finalize user documentation. Prepare the final presentation slides and code submission for the Final Evaluation.

August 24 - November 2 (Extended Coding Period)

- Stretch Goals: My core architecture is designed to be completed within the standard 12-week timeframe. If the extended timeline is utilized, I will focus on advanced stretch goals, such as adding Jenkins Community Discourse threads to the vector database for broader context, or implementing advanced caching mechanisms to reduce latency.

Future Improvements

- **Role-Based Access Control (RBAC) Integration:** Ensuring the chatbot only suggests actions or provides troubleshooting steps that the specific user has the permission to execute within their Jenkins instance.
- **Community Forum Indexing:** Expanding the Vector Database to automatically ingest solved threads from the Jenkins Discourse forum, providing real-world troubleshooting context beyond official documentation.

Continued Involvement

I intend to remain a highly active contributor to the Jenkins project well beyond GSoC. My goal is to become a core

maintainer for the AI tools and plugins within the Jenkins ecosystem, helping to guide future AI integrations and eventually mentoring new contributors.

Conflict of Interests or Commitment(s)

I do not have any conflicts of interest. I am not participating in any other summer internships or full-time work engagements that would compete for my time. The code I develop will be contributed entirely to Jenkins without any copyright claims from an employer or university.

Major Challenges Foreseen

- LLM Hallucinations: Jenkins requires exact, highly technical configurations. If the LLM guesses an answer, it could break a user's pipeline. I will mitigate this by implementing strict prompt engineering that forces the model to rely only on the retrieved context, returning "I do not have the exact documentation for this" if a definitive answer isn't found in the vector database.
- Seamless UI Integration: Jenkins has a mix of modern and legacy interfaces. Ensuring the JavaScript/TypeScript chatbot widget renders perfectly without conflicting with existing page styles or breaking older plugins will require careful frontend isolation.

References

- Jenkins Plugin Development Tutorial (jenkins.io/doc/developer/tutorial)
- [LangChain4j Official Documentation](#)
- ChromaDB / Vector Database Architecture Guides

Relevant Background Experience

I have a good foundation in Java, full-stack development, and artificial intelligence, which aligns directly with the requirements for the AI Chatbot project. I have already demonstrated my ability to navigate the Jenkins codebase and testing environment by successfully getting my first pull request merged into the Acceptance Test Harness. Additionally, I have built LLM based intelligence assistant system for Indian laws In which I have used RAG.

Personal

I am currently a student at Charotar university of science and technology studying Computer Science and Engineering. I am in my 3rd year of my college. I am highly interested in Jenkins because it is the industry standard for CI/CD, and contributing to its core infrastructure gives me the opportunity to build enterprise-grade software that impacts millions of developers.

Availability and commitments

I am fully committed to treating GSOC as a full-time responsibility and will dedicate 35-40 hours per week to the project. My university exams conclude on March, leaving my summer completely free for this initiative.

Experience

Free Software Experience/Contributions (optional):

Jenkins (Merged PR): Fixed issue #2594 in the Acceptance Test Harness by ensuring the ByFactory.button uses exact text matching. [PR #2621](#).

Language Skill Set

- Java: Intermediate (Comfortable navigating large codebases and writing tests; actively expanding architectural knowledge)
- Spring Boot: Beginner/Intermediate (Currently learning and applying to backend project architectures)
- Python (AI/ML): Intermediate (Hands-on experience building custom data pipelines and predictive models)

- LLMs & Prompt Engineering: Intermediate (Experienced in integrating LLMs into domain-specific applications and RAG pipelines)
- JavaScript / HTML / CSS: Familiar (Capable of building and editing frontend UI components)

Related Research and Work Experience (if any):

- Secured First position in AI Hackathon([certificate](#))
- Made an Project on Stock market prediction using XGBoost.
- Was an intern at Codsoft as an App developer.
- Made an LLM based intelligence assistant system for Indian laws so citizens can understand Indians laws properly.

Reference Links and Web URLs (optional):

- <https://github.com/Harmit07>
- <https://www.linkedin.com/in/harmit-jetani-a17501283/>
- <https://community.jenkins.io/u/harmit/summary>