Self■Healing RAG Pipeline Project (Pre■GSoC Build)

This PDF outlines a complete, small but powerful end■to■end Self■Healing RAG (Retrieval■Augmented Generation) Pipeline project.
The goal is to prepare a strong portfolio project before GSoC applications open.

---
1. Project Overview

A Self■Healing RAG system can:
• Detect retrieval or generation failures
• Automatically fix missing embeddings
• Re-index corrupted data
• Switch between retrievers or models
• Log issues and self-correct

---
2. Architecture Summary

Modules:
• Data Loader
• ETL + Chunking Engine
• Embedding Generator
• VectorDB (ChromaDB)
• Retriever
• LLM Generator
• Self-Healing Engine
• Logging System

---
3. Development Roadmap (Before GSoC)

Phase 1 — Setup (Anaconda)
• conda create -n ragheal python=3.10
• pip install langchain chromadb openai pypdf tiktoken

Phase 2 — ETL System
• Clean text, chunk data, metadata injection

Phase 3 — Embeddings + Vector Store
• Build Chroma index
• Add re-indexing functions

Phase 4 — RAG Query Pipeline
• Retriever + LLM answer
• Add fallback retriever

Phase 5 — Self■Healing System
• Detect missing embeddings
• Detect 0-result retrieval

• Auto-regenerate chunks
• Retry logic + fallback

Phase 6 — Final demo project

---
4. Next Steps After First Version

• Add multi-agent verification
• Add LangGraph orchestration
• Add evaluation metrics
• Add monitoring dashboard
• Prepare proposal version

---
5. Recommended Folder Structure
---
project/
data/
etl/
modules/
vectorstore/
logs/
main.py
notebook.ipynb

This project will be your strongest GSoC preparation asset.
