Kisan Saathi – Technical Documentation

# Project Overview

Kisan Saathi is a local-first AI-powered assistant designed to support Indian farmers by answering agricultural queries. It leverages a combination of semantic search (FAISS), transformer-based embeddings, and a Streamlit-based interactive UI. The system prioritizes using a local dataset (Kisan Call Center or KCC) and gracefully falls back to live Internet search when local data is insufficient.

# Workflow Summary

* - Data Preprocessing - Cleans and combines raw KCC queries and answers.
* - Embedding & Indexing - Generates vector embeddings and builds a FAISS index.
* - Query Processing - Finds local matches using semantic similarity and falls back to Internet search if needed.
* - Answer Delivery via UI - Presents results in a user-friendly web interface.

# Data Preprocessing

* - Load raw dataset
* - Clean query and answer fields
* - Drop empty entries
* - Create combined chunks
* - Save cleaned dataset

# Vector Store Creation

* - Generate embeddings using SentenceTransformer
* - Build FAISS index and save it to disk

# Query Handling

* - Perform semantic search using FAISS
* - Check similarity threshold to determine fallback
* - Generate answer using QA model
* - Fallback to Internet search using DuckDuckGo

# User Interface with Streamlit

* - Input field for user query
* - Status indicator during processing
* - Display result from local data or Internet

# Project Directory Structure

KisanSaathi/  
├── data/  
│ ├── Raw/  
│ ├── Processed/  
│ └── Embeddings/  
├── src/  
│ ├── pipeline\_manager.py  
│ ├── kcc\_preprocessor.py  
│ └── rag\_pipeline.py  
├── app.py  
├── requirements.txt  
└── .env

# System Requirements

* - Python Version: 3.8+
* - Dependencies: pandas, sentence-transformers, faiss-cpu, transformers, streamlit, duckduckgo-search
* - Memory: 8 GB RAM minimum (16 GB recommended)

# Setup & Launch Guide

* - Step 1: Preprocess the Dataset: python src/kcc\_preprocessor.py
* - Step 2: Launch Streamlit App: streamlit run app.py
* - Step 3: Access in Browser: http://localhost:8501

# Planned Enhancements

* - Full Dataset Integration with cloud/distributed support
* - Better Semantic Search with hybrid methods
* - Multilingual Capabilities for Hindi and regional languages

# Acknowledgments

* - KCC Dataset for authentic agricultural data
* - Open-Source Tools: FAISS, PyTorch, Streamlit, Hugging Face, DuckDuckGo Search