BHARATIYA ANTARIKSH HACKATHON 2025

Powered by H2S

Team Name : **SMART BUILDERS**

Team Leader Name : **NAVYA NAYER**

Problem Statement : **Problem Statement - 4: Designing a Chain-of-Thought-Based LLM System for Solving Complex Spatial Analysis Tasks Through Intelligent Geoprocessing Orchestration**

TEAM MEMBERS

Powered by H2S

Team Leader:

Name: NAVYA NAYER
College: AMITY UNIVERSITY, NOIDA, UTTAR PRADESH

Team Member-1:

Name: MANVENDRA SINGH TANWAR
College: AMITY UNIVERSITY, NOIDA, UTTAR PRADESH

Team Member-2:

Name: BHANVI NAYER
College: AMITY UNIVERSITY, NOIDA, UTTAR PRADESH

Team Member-3:

Name: DIKSHA KHANDELWAL
College: AMITY UNIVERSITY, NOIDA, UTTAR PRADESH

IDEA

IDEA/SOLUTION:

GeoChain is a smart assistant for geospatial analysis. It helps users ask questions in plain English and automatically builds the steps needed to get map-based answers, just like a GIS expert would.

- Uses LLMs + RAG + step-by-step reasoning to plan spatial workflows
- Picks the right tools (like QGIS, GeoPandas) and runs them under the hood
- Outputs clear maps, downloadable layers, and reasoning logs
- If something breaks, it learns and retries, just like a human would
- Runs everything inside a simple, interactive Streamlit interface

GeoChain doesn't just automate GIS. It makes it smart, explainable, and accessible turning spatial decision-making into an AI-powered conversation.

PROBLEM RESOLUTION:

Problem	How GeoChain Helps
GIS tools are complex and expert-only	Users ask questions in plain English, no GIS expertise needed. GeoChain builds and runs workflows for them.
Manual workflows are time-consuming	GeoChain automatically plans and executes multi-step GIS tasks end-to-end, saving hours of manual effort.
No reasoning in traditional tools	Uses LLMs with Chain-of-Thought (CoT) to explain every step logically; mimicking expert reasoning.
Tools and data are fragmented	Integrates GeoPandas, QGIS, GDAL, OSM, and Bhoonidhi data into a unified AI-driven pipeline.
AI outputs lack transparency	Generates human-readable CoT logs and machine-executable YAML/JSON, every decision is traceable.

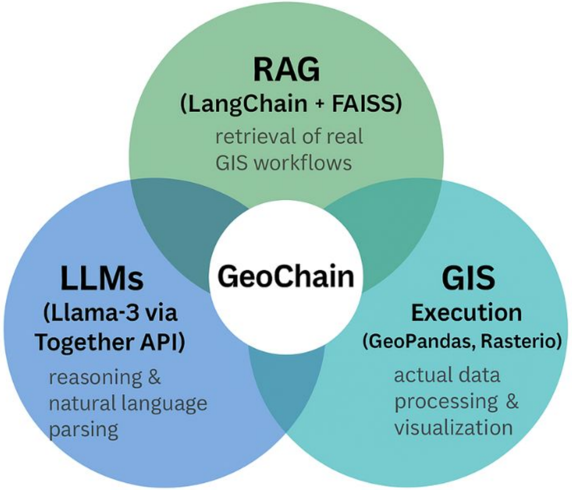
DEMO VIDEO LINK : https://youtu.be/xoknGqB7CQo

OPPORTUNITY

Powered by H2S

Existing Problems

GIS platforms require technical expertise (Python, QGIS, etc.)	Manual workflow design is slow and error-prone
LLM-GIS integrations often use toy datasets, not scalable	No transparency in AI decision-making)
Lack of real-time support for diverse spatial queries	No unified platform combining reasoning, retrieval & execution



What GeoChain Does Differently

01 End-to-end Automation
Natural language input to spatial output—zero manual scripting.

02 Chain-of-Thought Reasoning for Transparency
LLM breaks down the user query into interpretable, stepwise operations.

03 RAG-Powered Knowledge Retrieval
Dynamically pulls relevant geospatial methods (from QGIS docs, GeoPandas, GDAL) using FAISS vector search.

04 Real GIS Execution, Not Just Recommendations
Executes workflows using Python GIS libraries like GeoPandas and Rasterio.

05 Supports Attribute-Based & Proximity Queries
Understands conditions like "rivers within 2 km of Teesta in Sikkim."

06 Visual Debug + Editable Plans
Exposes intermediate steps and lets user review or edit before execution.

07 Automatic Data Linking & Conversion
Handles file formats, spatial joins, and reprojections without user intervention.

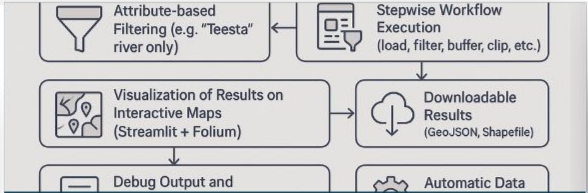
08 Extensible Framework
New tools and data sources can be plugged in easily—scalable for disaster, climate, or defense missions.

PROPOSED SOLUTION & FEATURES

Powered by H2S

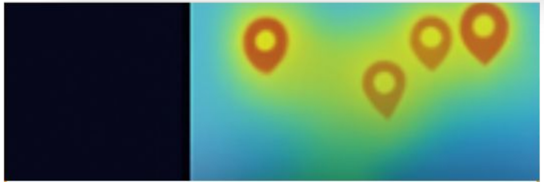
Our solution enables intelligent, explainable, and actionable geospatial workflows

Natural Language Geospatial Queries



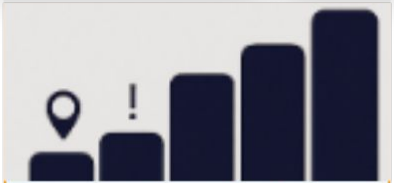
RAG-powered LLM Workflow Planning

Stepwise Workflow Execution
(load, filter, buffer, clip, etc.)

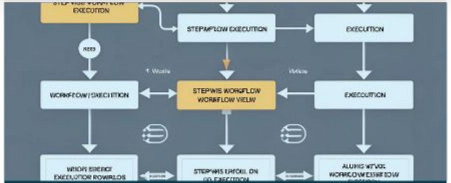


Visualization of Result on Interactive Maps
(Streamlit + Folium)

Attribute-based Filtering
(e.g., “Teesta” river only)



Downloadable Results
(GeoJSON, Shapefile)



Chain-of-Thought Reasoning Display
(via LLAMA)

USE CASES AND FLOW DIAGRAM

Powered by H2S

USE CASES

01 FLOOD RISK MAPPING

Our system automatically identifies flood-prone areas by analyzing river networks from OSM and elevation data from Bhoonidhi DEM. It buffers rivers, filters low-lying zones, and intersects these layers to generate accurate risk maps. This helps disaster management authorities, NGOs, and urban planners prepare effective response plans.

02 SITE SUITABILITY

It finds optimal locations for hospitals, schools, or warehouses by analyzing population density, proximity to major roads, and avoidance of flood zones. The system combines multiple spatial layers and ranks sites based on user-defined criteria, supporting faster and smarter urban infrastructure planning.

03 URBAN PLANNING

The system assists in creating zoning plans for residential, commercial, and green areas while considering environmental constraints like pollution zones or protected forests. It enables smart city projects by visualizing sustainable layouts and helping decision-makers plan balanced urban growth

User Query

User enters a geospatial query

RAG Retrieval

Retrieves relevant GIS documentation

LLM Planning

Generates a step-by-step workflow

Workflow Execution

Executes the workflow using GIS tools

Result Visualization

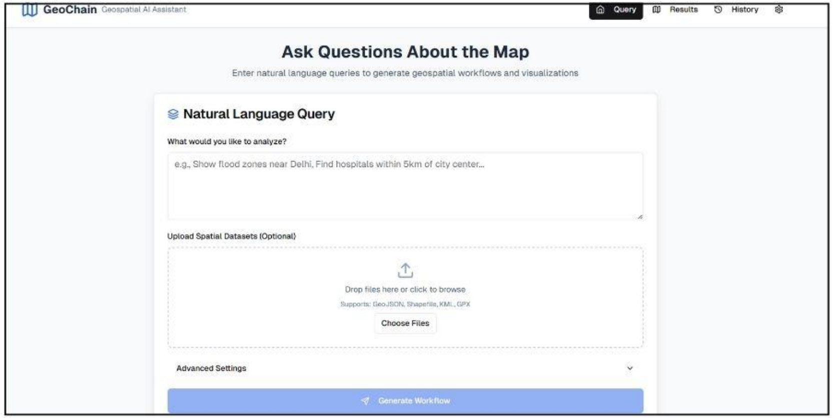
Displays results on an interactive map

Optional Features

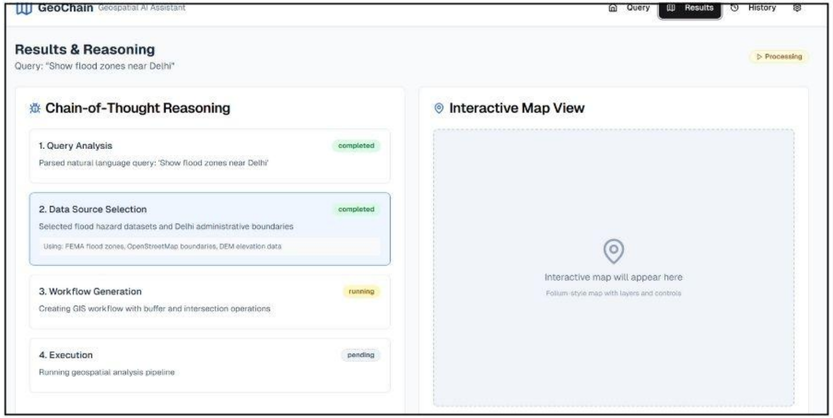
Includes advanced features like voice input

WIREFRAMES

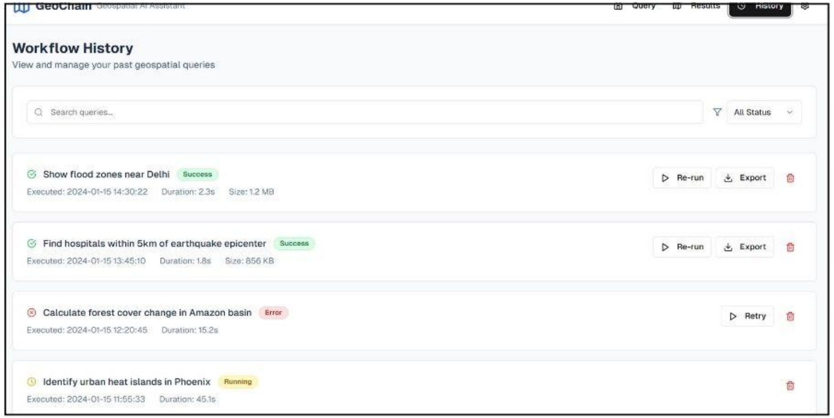
Powered by H2S



STEP 1: QUERY



STEP 2: RESULTS

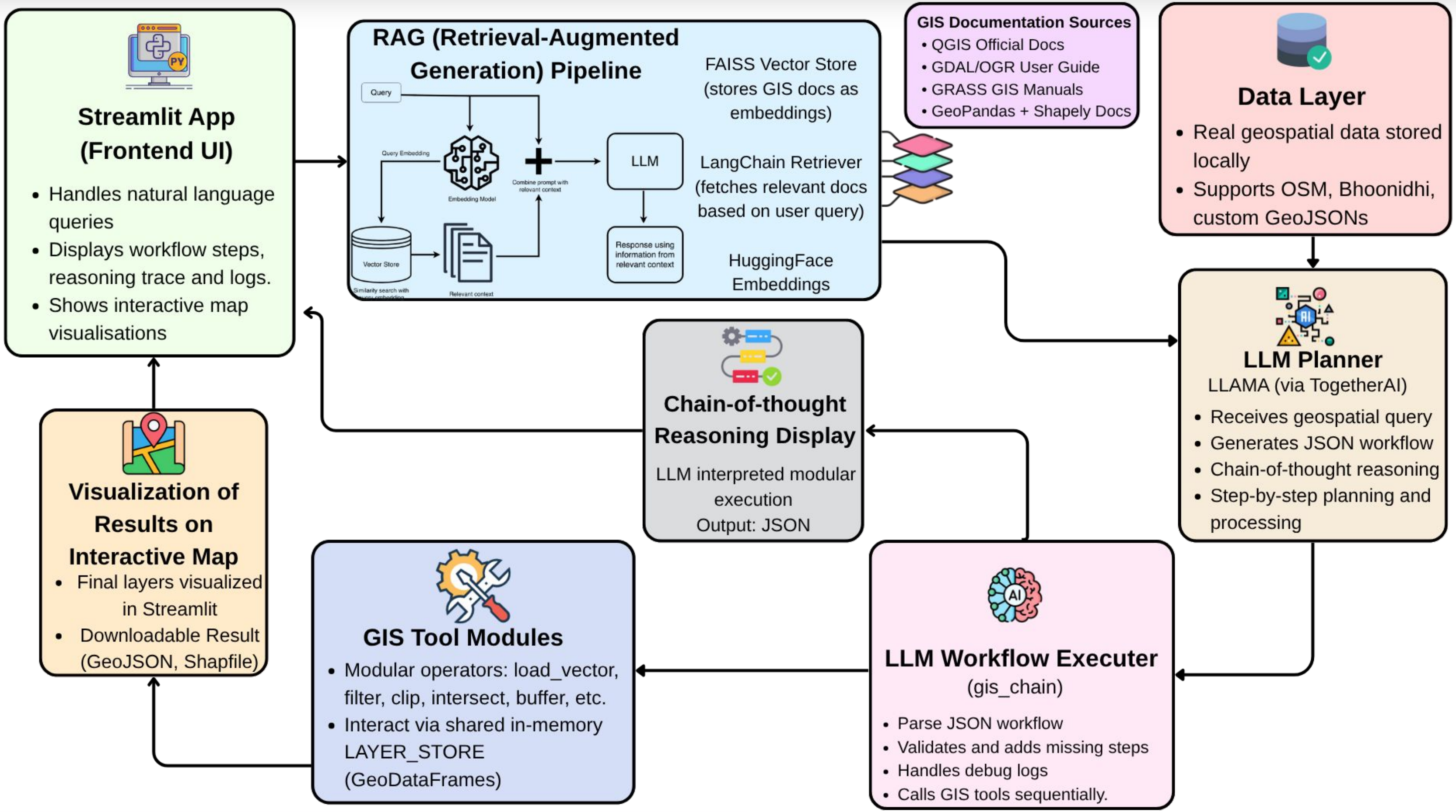


STEP 4: HISTORY

STEP 3: WORKFLOW OUTPUT

ARCHITECTURE DIAGRAM

Powered by



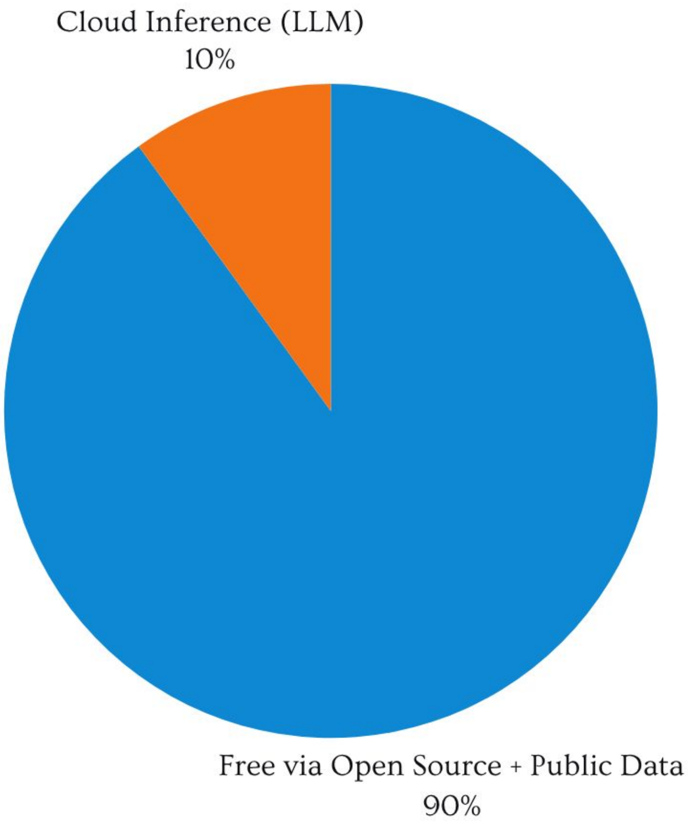
TECHNOLOGY STACK

Module	Tools / Frameworks	Purpose
1. Streamlit App (Frontend UI)	Streamlit-folium	Capture natural language queries- Show workflow steps & maps
2. RAG Pipeline	FAISS- LangChain Retriever- HuggingFace Embeddings	Store & search GIS document vectors- Inject relevant context to LLM
3. GIS Docs as Knowledge Base	QGIS Docs- GDAL/OGR Guides- GRASS Manuals- GeoPandas + Shapely Docs	Provide domain-specific knowledge for LLM reasoning
4. Data Layer	GeoJSON files- OSM- Bhoonidhi	Source of real geospatial layers
5. LLM Planner	TogetherAI API (LLAMA model)	Generate JSON workflow- Chain-of-Thought reasoning for GIS planning
6. Chain-of-Thought Display	JSON rendering in Streamlit	Show intermediate LLM reasoning output (steps, logs)
7. GIS Tool Modules	GeoPandas- Shapely- Custom tools (load_vector, buffer, clip, etc.)	Perform spatial operations- Store outputs in shared LAYER_STORE
8. Workflow Executor (gis_chain)	Python- Custom execution logic	Parse and run JSON steps- Add missing steps- Call tool functions
9. Visualization Module	folium- streamlit-folium	Render map output- Support GeoJSON/Shapefile download

ESTIMATED IMPLEMENTATION COST

Powered by H2S

COMPONENTS	COSTS
🖥️ STREAMLIT APP (FRONTEND)	₹0
📚 RAG STACK (FAISS, LANGCHAIN)	₹0
📄 DOCS (QGIS, GDAL, GEOPANDAS)	₹0
🤖 LLM PLANNER (MISTRAL/LLAMA API)	₹8K–₹12K (GPU)
🛠️ GIS TOOLS (CLIP, BUFFER, ETC.)	₹0
🌍 DATA LAYER (BHOONIDHI, OSM)	₹0
🗺️ OUTPUT (FOLIUM, GEOJSON)	₹0



DEMO VIDEO

Powered by H2S

https://youtu.be/xoknGqB7CQo

Powered by H2S

BHARATIYA ANTARIKSH HACKATHON 2025

THANK YOU