Project Title: ASTRA-Q — AI Help Bot for Information Retrieval from a Knowledge Graph Based on Static/ Dynamic Web Portal Content (ISRO Hackathon)

Team Name: AstraMind\ **Portal:** MOSDAC

Phase-wise Workflow

Phase 1: UI Development *(Status: \~80% Complete)*

- **Technology:** ReactJS (with support for Streamlit as fallback/demo)
- **Components:**
- Landing page with animated space background (twinkling & shooting stars, parallax)
- Login / Signup
- Chat interface (main interaction hub)
- Admin dashboard (upload new documents, manage source URLs)
- Knowledge graph visualization (basic Neo4j view or static graph preview)
- Settings/help/about pages
- **Next UI Tasks:**
- Integrate backend API with chatbot
- Display answer sources
- Add user query history

Phase 2: Web Crawling & Data Extraction

- **Tech Stack:**
- Playwright (JavaScript-rendered pages)
- BeautifulSoup (Static HTML pages)
- **Workflow:**
- Crawl target pages on MOSDAC for datasets, products, and documentation
- Extract:
    - Titles, body text, table data
    - Links to downloadable PDFs, DOCs, and datasets
    - Metadata (region, satellite, parameters)
- Store as:
    - JSON metadata
    - Raw text files or structured HTML segments
- **Next Tasks:**
- Set up crawler scheduler
- Normalize data formats

Phase 3: Document Parsing & Semantic Embedding

- **Libraries:** PyMuPDF / pdfminer.six, python-docx, LangChain, FAISS

• **Steps:**
• Parse downloaded documents (PDF/DOCX)
• Chunk text (based on headings or token size)
• Embed using OpenAI Embeddings
• Store in FAISS vector database for retrieval
• **Output:** A searchable embedding index
• **Next Tasks:**
• Test retrieval with real queries
• Add multilingual chunk embedding support (future)

Phase 4: Knowledge Graph Construction

• **Tools:** Neo4j + Protégé
• **Schema Example:**
• Nodes:
    ◦ Satellite, Product, Parameter, Region
• Relationships:
    ◦ (Satellite)-[:PRODUCES]->(Product)
    ◦ (Product)-[:OBSERVES]->(Parameter)
    ◦ (Product)-[:COVERS]->(Region)
• **Workflow:**
• Manually or semi-automatically extract entities and relations from documents/metadata
• Populate Neo4j DB
• Query with Cypher via LangChain
• **Next Tasks:**
• Finalize ontology in Protégé
• Add batch import script

Phase 5: RAG Chatbot Backend

• **Tech Stack:** FastAPI (or LangChain Server), LangChain, OpenAI/Mistral
• **Retrieval Pipeline:**
• User query →
• Semantic search in FAISS
• Optionally combine with Neo4j KG answers
• Final prompt + context sent to LLM
• **Features:**
• Multi-turn memory support
• Source document references
• Fallback response generation
• **Next Tasks:**
• Deploy RAG pipeline on local or cloud backend
• Connect with frontend chat component

Phase 6: Integration & Testing

- **End-to-End Checks:**
• Query → Retrieval → LLM → Answer → UI Display
- **Test Cases:**
• Static page queries ("Where is INSAT-3D SST data?")
• Dynamic content ("Download rainfall product for Tamil Nadu")
• Geospatial questions ("Show LST map of Kerala")
• Follow-up questions ("Show same for May 2023")
- **Next Tasks:**
• Unit tests for each pipeline block
• Frontend user feedback collection

Phase 7: Optional Add-ons

- **Multilingual Support:** Translate user input and bot output using Indic NLP
- **Geospatial Visualization:** Leaflet.js or CesiumJS map overlays
- **Low-Code Admin Panel:** GUI for onboarding new websites or datasets

---

Deployment & Demo

- Host frontend (Vercel/Netlify), backend (Render/Fly.io/localhost)
- Include:
• Sample queries & responses
• Screenshots/video walkthrough
• Documentation + architecture diagrams

Summary

This workflow provides a step-by-step structure for implementing the ASTRA-Q solution. Each phase builds toward a robust, multilingual, AI-powered help bot for satellite data retrieval on the MOSDAC portal, and potentially for other ISRO portals and scientific websites.

Next Immediate Tasks:

1. Finish crawler for static + dynamic MOSDAC content
2. Start FAISS embedding and test LangChain search
3. Begin KG schema design in Protégé & Neo4j
4. Connect frontend chat to backend for MVP test