Knowledge Graph Representation and Reasoning (KR)

The first component of our “Agentic Calling System for UOS”: **The Admission Queries Knowledge Graph (KG)**

Why we are starting with building the **“Answering Admission Queries”** use case:

1. The domain is well-bounded.
2. Information is static + rule based.
3. Ideal starting point for building a KG.
4. Easy to test and infer through our call agent.

Compete Roadmap of how I’ll be building the KG.

Part 1: Understanding what type of KG, we would need for our project.

A Knowledge Graph (KG) is:

- Structured network of entities (nodes)
- These nodes are connected by relationships (edges)
- And enriched with attributes (properties)

In our case, the KG will allow the AI agent to answer the following:

- “UOS ka admission kab start hota hai?”
- “CS ka merit 2023 me kitna tha?”
- “Fee structure kya hai?”
- “Documents kon kon se required hain?”
- “Admission open ya closed hain?”

This basically mean the KG will connect things like:

Programs-Merit-Year-Fee-Conditions-Deadlines

Part 2: What Data do we need to build the KG?

So, to handle the first use case (Admission Queries), the KG should contain the information from some of the following sources:

1. **Study Programs Information**
    - All departments (CS, IT, Math, English, BBA, etc.)
    - All degrees offered (BS, MSc, MPhil, PhD)
    - Duration
    - Seats (Maybe Optional)

2. **Admission Cycles**
    For each year (2021, 2022, 2023, 2024, 2025….):
    - Admission start date
    - Admission close date
    - Entry test date (IDTS we have any)

3. **Merit Lists**
    For each program:
    - 1st merit list
    - 2nd merit list
    - Reporting lists
    - Final closing merit
    - Previous years' closing merit

4. **Eligibility Criteria**
    Each program has conditions:
    - Minimum Marks
    - Required subjects
    - Equivalence rules

5. **Fee Structure**
    - Admission Fee
    - Semester Fee

6. **Required Documents**
    - CNIC
    - Photos
    - Matric and Intermediate certificates
    - Others (Domicile, etc.)

Part 3: From where we are collecting the Data From.

1. **UOS Website (This will be our primary source)**
    - Admission Section
    - Past Merit lists
    - Program details
    - Prospectus PDF

2. **Prospectus (Important to this use-case)**
   Usually contains:
    - Programs
    - Eligibility
    - Fee Structure
    - Admission rules

3. **University Social Media Posts**
   Often Share:
    - Deadlines
    - Merit Lists
    - Announcements

4. **Admin**
    - Have to ask for past merit lists
    - Admission criteria tables

Part 4: Structuring the Knowledge Graph (KG).

Step by step process on how to build a KG schema.

Step 1: Identify Entities (Nodes)

Examples:

- Program
- Department
- Merit List
- Admission Cycle
- Fee Structure
- Eligibility Rule
- Document Requirement

Step 2: Identify Relationships (Edges)

Examples:

- Program **belongs_to** Department
- Program **has_merit** MeritList
- Program **requires** EligibilityRule
- Program **has_fee** FeeStructure
- AdmissionCycle **has_document** DocumentRequirements

Step 3: Identify Attributes (Properties)

Examples:

- **Program** properties:
  - name
  - duration
  - morning/evening
- **MeritList** properties:
  - year
  - merit_value

  - merit_stage (1st, 2nd, …, final)
- **EligibilityRule** properties:
  - min_percentage
  - required_subjects

Part 5: Building the prototype of KG

I’ll be focusing on starting **small,** just one department for now.

Department of Computer Science (CS)

- BS CS
- BS AI
- BS DS

For each program:

- Merit list (2021 – 2025)
- Eligibility criteria
- Fee structure
- Required documents

(I can build this using Neo4j easily)

Part 6: Step by Step plan to build the complete KG.

Step 1: Collect Data

- Download latest UOS prospectus
- Open UOS website “Admissions” sections.
- Collect merit lists from last 3-4 years.
- Gather program details.

Step 2: Clean and Organize the Data

Make a simple Excel sheet:

- **Sheet1:** Programs
- **Sheet2:** Merit Lists
- **Sheet3:** Fee Structure
- **Sheet4:** Eligibility Rules
- **Sheet5:** Documents

Step 3: Convert Excel to Nodes & Relationships

I’ll use python or Neo4j browser to:

- CREATE nodes
- CREATE relationships

Step 4: Build Graph Query Templates

For example:

Query: “CS ka merit kya tha?”

KG Query: *Match (p:program {name:”BS CS”})-[:HAS_MERIT]->(m:MeritList {year: 2025})*

RETURN m.value;

Step 5: Connect Knowledge Graph to GraphRAG

I’ll use LangChain/LlamaIndex:

- Build embeddings
- Use KG + embeddings for hybrid retrieval
- Test natural-language queries

Step 6: Integrate with Call Agent

- User speaks query
- ASR > Text
- NLU > Intent (Admission Query)
- GraphRAG > Answer
- TTS > Speech

Part 7: Simple Roadmap

1. **Building Admission KG (CS DEPT. Only)**
   Entities:
   - BSCS
   - Admission Cycle
   - Merit List
   - Eligibility
   - Fee Structure
   - Documents
2. **Building GraphRAG Retrieval System**
   - Load KG
   - Embed text
   - Query using natural language
3. **Connect to Call Agent Prototype**
   Create minimal ASR – KG – TTS loop.