



**Work Integrated Learning Programmes Division**  
**M.Tech. in AIML**  
**NLP Applications**  
**S2-24\_AIMLCZG519**

**Assignment 1 – PS-2**

**PART-A**

**Problem Statement: Healthcare Diagnosis Network**

**Objective:**

**Design and implement a web-based Knowledge Graph application that models and visualizes relationships among symptoms, diseases, treatments, and medical specialists. The application should allow users to input and explore how medical concepts are interconnected to support diagnostic reasoning and clinical planning.**

**Requirements:**

**Web Interface (3 Marks)**

**Front-End Development:**

1. Design a clean and responsive interface using HTML, CSS, and JavaScript.
2. Provide input fields for users to enter:
  - a. Symptom names
  - b. Disease names
  - c. Treatment options
  - d. Specialist roles

**User Input:**

3. Users should be able to:
  - a. Add a medical concept and define its relationships (e.g., "*Fever*" indicates "*Influenza*", "*Influenza*" treated by "*Antivirals*", "*Antivirals*" prescribed by "*Infectious Disease Specialist*").

- b. Upload data in JSON/CSV format to load clinical knowledge or patient case data.

**Use Case to Consider for Input:**

4. Diagnostic Decision Support Network: Mapping relationships among symptoms, diagnoses, treatments, and medical roles to assist clinicians in making accurate and timely decisions.
5. Fields for input must include: Source Entity, Relationship Type (e.g., *"indicates"*, *"treated by"*, *"managed by"*), and Target Entity (symptom, disease, treatment, or specialist).

**Graph Query and Visualization (3 Marks)**

**Back-End Implementation:**

1. Use Flask to handle web requests and application logic.
2. Use networkx to build and manage the medical knowledge graph.
3. Allow users to:
  - a. Query possible diagnoses for a given symptom.
  - b. Find recommended treatments for a specific disease.
  - c. Identify specialists associated with a given disease or treatment.

**Visualization Requirements:**

1. Use D3.js or an equivalent visualization library to render the graph interactively.
2. Ensure the graph auto-updates with each new medical concept or relationship.
3. Display clusters of symptoms, diseases, and treatments based on known medical groupings (e.g., respiratory, neurological).

**Integration (2 Marks)**

1. Ensure seamless communication between the front end and back end.
2. Support both real-time data entry and batch file uploads.
3. Display query results clearly and ensure the graph reflects changes instantly.

**Task B: Enhancement Plan (2 Marks)**

Develop an enhancement plan that addresses:

1. Scalability: Efficiently manage and visualize thousands of medical concepts and relationships.
2. Performance Improvements: Implement indexing for faster queries and batch processing of uploads.
3. Advanced Visualization: Use clustering for disease categories, severity filters, or evidence-based relationship strength indicators.
4. UI Improvements: Add search, filtering by body system or symptom severity, node highlighting, and an export feature for clinical reports.

## **PART – B**

### **Literature Survey (5 Marks)**

**Conduct a literature review on Graph Neural Networks for Knowledge Graphs, satisfying the following requirements -**

1. Theoretical Understanding and Definitions
2. Key Methods and Approaches
3. GNN Architectures
4. Evaluation Metrics and Benchmarks

### **Deliverables:**

#### **PART - A**

- A well-documented code (Python and frontend) for the knowledge graph application.
- Instructions for running the application locally.
- A brief report explaining the design choices and any challenges faced during implementation.
- A set screenshots that explains the entire flow of the application to be included in the report.
- Task-B to be submitted as a .pdf document.

#### **PART – B**

A well-documented literature review to be presented as a .pdf document