INFORMATION & COMMUNICATION TECHNOLOGY ACADEMY OF KERALA
ICTAK

Internship Report

LLM Based Knowledge-Graph Builder

Artificial Intelligence and Machine Learning Specialist- June 2025

Prepared by :

Abhishek Krishna

Rohit S

Shebin Sabu

Steev Shaji

Table of Contents

Table of Figures

Abstract

The rapid growth of digital learning platforms has increased the need for intelligent systems that can organize institutional knowledge in a structured and interactive form. Knowledge graphs have emerged as a powerful approach for representing relationships between courses, trainers, skills, and learners. Motivated by this need, the present work focuses on designing an LLM-based Knowledge Graph Builder specifically for ICT Academy of Kerala (ICTAK), enabling automated extraction, structuring, and visualization of academic information. The core idea is to convert unstructured sources like brochures, resumes, and student enrollment data, into a machine-readable relational graph.

To accomplish this, the project employs PDF text extraction, OCR, resume parsing, skill mining, and AI-assisted mapping techniques. The extracted entities are processed using Python, NetworkX, and an LLM-driven reasoning layer to build a dynamic knowledge graph. A PyVis-powered visualization layer and a Streamlit user interface enable interactive exploration and filtering of the graph. The system successfully generates course–module networks, trainer–skill mappings, and student–course relationships while supporting keyword search, node-type filters, and multi-hop expansion. The results demonstrate a scalable and efficient solution for institutional knowledge consolidation and decision support.

The findings indicate that automated knowledge graph construction significantly reduces manual effort and enhances information accessibility. In the future, this work can be extended with full LLM-powered semantic extraction, skill-gap analytics, real-time updates, and integration with ICTAK's LMS systems.

Introduction

In recent years, educational institutions have increasingly transitioned toward digital ecosystems where academic data, training resources, and organizational information are stored across multiple platforms and formats. As institutions grow, the ability to organize, interconnect, and retrieve relevant information becomes essential for enhancing decision-making, curriculum planning, talent allocation, and learner support. Traditional information systems typically store data in isolated tables or documents, which limits their ability to capture meaningful relationships among entities such as courses, modules, trainers, skills, students, and learning outcomes. This gap has motivated the adoption of knowledge graphs, a modern and highly efficient representation method that connects information through semantic relationships and graph structures.

A knowledge graph provides a powerful way to model institutional knowledge by linking related entities in an interpretable and machine-readable form. Unlike conventional databases, knowledge graphs prioritize relationships, such as trainer teaches course, course has module, and student enrolled in program, allowing seamless navigation and deeper insights. With the rise of artificial intelligence, Large Language Models (LLMs) have further strengthened this field by enabling automated extraction of information from unstructured files such as brochures, resumes, reports, and textual descriptions. This integration of AI with graph-based modeling is transformative for organizations like ICT Academy of Kerala (ICTAK), which manage a large volume of training programs and associated human resource data.

Scope and Objectives

The purpose of this project is to design and develop an LLM-Based Knowledge Graph Builder for ICTAK, enabling automated construction, structuring, and visualization of institutional knowledge from raw data sources. The system focuses on extracting relevant information from unstructured PDF brochures, trainer resumes, and student database files to generate a unified knowledge graph that represents the complete academic ecosystem of ICTAK. The scope includes automated extraction of course names, module lists, trainer skills, trainer–course mappings, and student enrollment details.

The main objectives of this work are:

- Objective 1: Automated Information Extraction
- Objective 2: Trainer Skill and Profile Mining
- Objective 3: Entity Mapping and Relationship Construction
- Objective 4: Knowledge Graph Construction
- Objective 5: Interactive Visualization System
- Objective 6: Support for Institutional Insights

Background

Definition of the Problem

ICT Academy of Kerala (ICTAK) conducts numerous technical training programs, certification courses, and upskilling initiatives across the state. These programs involve multiple stakeholders, course designers, trainers, coordinators, partner institutions, and enrolled students. However, most of the institutional academic information exists in unstructured or semi-structured formats, such as PDF brochures, trainer resumes, student enrollment logs, and Excel sheets. Retrieving insights from this data requires manual reading, data entry, and cross-verification, which is time-consuming, error-prone, and inefficient.

The core problem arises from the lack of a unified system that can integrate, structure, and relate this dispersed information automatically. Without a relational view of ICTAK's training ecosystem, activities such as assigning trainers, understanding skill gaps, designing new course curricula, identifying module overlaps, and guiding students become significantly harder. There is a clear need for an automated system capable of extracting knowledge from raw documents and representing the relationships among entities such as courses, modules, trainers, skills, and students.

Therefore, the problem addressed in this project is:

"How can we automatically extract structured knowledge from ICTAK's academic resources and represent it as an interactive knowledge graph powered by AI and LLM-assisted reasoning?"

Basic Background Relevant to the Problem

Knowledge Extraction and Unstructured Data

Most institutional data exist in PDF brochures, resumes, and text documents. These files have inconsistent formats, layouts, and terminology. Extracting structured information requires combining:

- PDF text extraction
- Optical Character Recognition (OCR)
- Pattern matching
- AI-driven concept identification
- Rule-based parsing for modules and skills

The challenge is to transform this unstructured text into clean, usable data.

Knowledge Graphs

A knowledge graph (KG) is a graph-based data model that organizes data into:

- Nodes → representing entities
- (Course, Module, Trainer, Skill, Student)
- Edges → representing relationships
- (teaches, enrolled_in, has_module, skilled_in)

Karnaugh graphs, ontologies, semantic web technologies (RDF, OWL), and graph databases like Neo4j are traditionally used for knowledge graph modeling. However, for this project, the focus is on a lightweight, Python-based KG using NetworkX, tailored for ICTAK's training ecosystem.

Large Language Models (LLMs) in Knowledge Engineering

LLMs provide capabilities such as:

- semantic text extraction
- pattern recognition
- skill inference
- module classification
- relationship prediction

While the current version uses rule-based extraction, the project is designed for scalable LLM integration in the future.

Graph Visualization

Traditional tables cannot express relationships. Tools such as PyVis enable:

- interactive node-based visual representation
- keyword search
- neighborhood expansion
- color-coded entity types
- improved clarity of academic structure

This helps administrators and trainers navigate the training ecosystem easily.

Computational Approaches / Relevant Resources

The system is implemented using widely adopted AI and data-processing technologies.

1. PDF Processing

- pdfplumber – reads text from PDF documents
- pytesseract (OCR) – extracts text from scanned brochures
- Pillow (PIL) – handles images and page rendering

These together ensure reliable extraction from both text-based and scanned PDFs.

2. Natural Language Processing

- Keyword matching
- Regex-based module extraction
- Skill identification using predefined vocabularies
- Potential for spaCy and LLM-based expansion

3. Knowledge Graph Construction

- NetworkX DiGraph is used to build a directed graph containing nodes and relations.
- Entities are classified into categories: Course, Trainer, Student, Skill, Module.
- Relationships include:

  course → module (has_module)

  trainer → skill (skilled_in)

  trainer → course (teaches)

  student → course (enrolled_in)

4. Visualization

- PyVis for interactive, browser-based visualization
- Dynamic filtering, hierarchical layout, force-directed layout

5. Web Interface

- Streamlit is used to build an interactive web app for:
  - loading data
  - filtering by keyword
  - selecting node types
  - multi-hop expansion
  - downloading results

6. Hardware/Software Requirements

- Python 3.10+
- Tesseract OCR installed locally
- Adequate compute for OCR on larger PDFs
- Browser for graph rendering

Impact of the Problem

Global Perspective

Globally, organizations such as Google, Microsoft, Amazon, and LinkedIn rely heavily on knowledge graphs for:

- search systems
- recommendation engines
- skill graphs
- enterprise knowledge modeling
- intelligent assistants

Academic institutions abroad increasingly use semantic knowledge systems for curriculum design, competency mapping, and resource allocation.

National Perspective (India)

In India, adoption of knowledge graphs is growing rapidly, especially in:

- EdTech platforms (e.g., BYJU’s, UpGrad, NPTEL)
- AI-driven government projects
- Skill development frameworks (NSDC)
- University course catalog systems

Yet, many institutions still rely on manual record-keeping and siloed data management, resulting in inefficiencies. ICTAK, as a leading training organization, stands to benefit significantly from an internal knowledge graph system that improves transparency, planning, and data-driven decision-making.

Impact on ICTAK

An automated KG system helps ICTAK:

- understand trainer expertise distribution
- identify skill gaps
- analyze course overlaps
- visualize student enrollments
- optimize training delivery

It forms the foundation for advanced AI services like personalized course pathways, trainer recommendations, and analytics dashboards.

Challenges and Future Prospects

Challenges

- Inconsistent PDF formats across brochures
- Scanned PDFs with low text quality
- Resume skill extraction variability
- Semantic ambiguity
- Graph scalability
- Dependency on keyword-based matching

Future Prospects

- Integration of LLMs
- Ontology-based modeling
- Graph database integration
- Real-time updates
- Advanced analytics
- AI-based Curriculum Design

Review of Literature

The development of automated knowledge graph systems has become a significant research area in artificial intelligence owing to their ability to represent structured knowledge from heterogeneous data sources. Hogan et al. (2021) provided one of the most comprehensive surveys on knowledge graphs, explaining their foundations, construction methodologies, applications, and challenges. Their work highlights how KGs serve as the backbone of semantic search, recommendation engines, digital assistants, and enterprise knowledge management, which aligns directly with the goals of this project. Paulheim (2017) focused on the refinement of knowledge graphs, showcasing techniques for improving accuracy, resolving inconsistencies, and enhancing completeness—an essential aspect when constructing graphs from noisy sources like resumes or PDFs.

Ji et al. (2020) surveyed representation learning methods for knowledge graphs, covering embedding approaches, neural models, and graph reasoning frameworks. This study emphasizes how modern KG systems rely not only on symbolic relations but also on statistical representation, which opens future possibilities for ICTAK to integrate predictive analytics into their knowledge ecosystem. Similarly, Weikum et al. (2021) discussed the evolution of knowledge graphs, identifying trends such as machine learning–augmented extraction pipelines and the integration of contextual embeddings for richer semantic modeling. This reinforces the relevance of transitioning from purely rule-based extraction to hybrid LLM-supported extraction.

More recently, Peng et al. (2023) explored opportunities and challenges in large-scale knowledge graph construction, highlighting issues such as data inconsistencies, missing links, and entity ambiguity. These challenges are especially relevant in ICTAK's context, where brochure formats and trainer resume lack standardization. Emerging work on LLM-driven knowledge engineering further strengthens this direction. Kommineni (2024) presented a framework where human experts and LLMs collaboratively assist ontology and KG construction, demonstrating improved automation in entity extraction, relation identification, and schema design. This directly supports our goal of building an LLM-powered pipeline for institutional knowledge mapping.

Bowen Zhang and Soh (2023) introduced the "Extract, Define, Canonicalize" pipeline, an LLM-based KG construction workflow that systematically extracts entities, assigns semantic meaning, and resolves duplicates using generative models. This approach addresses challenges like varying naming conventions and incomplete skill tagging in resumes, making it highly relevant for ICTAK's brochure and trainer-skill extraction. Likewise, KG-LLM models for link prediction (2023) have demonstrated how LLMs can strengthen graph completeness by predicting missing trainer-course mappings or skill relevance, an area that the current system could benefit from in future iterations.

Classical information extraction studies also provide foundational relevance. Etzioni et al. (2005) introduced automated web-scale extraction techniques through the KnowItAll system, setting early precedents for large-scale entity extraction. Meanwhile, Wang et al. (2019) reviewed text mining strategies for PDF documents, particularly emphasizing OCR challenges, layout inconsistency, and noise handling, issues directly addressed in the PDF extraction modules of this project.

Together, these studies establish the academic foundation for automated extraction, knowledge graph construction, LLM-assisted refinement, and interactive visualization. They validate the feasibility and importance of developing a KG-based institutional intelligence framework, such as the one proposed for ICTAK.

References

- Etzioni, O., Banko, M., Soderland, S., & Weld, D. S. (2005). Unsupervised named-entity extraction from the web: An experimental study. Artificial Intelligence, 165(1), 91–134.
- Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutiérrez, C., … Young, A. (2021). Knowledge graphs. ACM Computing Surveys, 54(4), 1–37. https://doi.org/10.1145/3447772
- Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2020). A survey on knowledge graphs: Representation, acquisition, and applications. arXiv preprint arXiv:2002.00388.
- Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic Web, 8(3), 489–508. https://doi.org/10.3233/SW-160218
- Peng, C., Xia, Z., Naseriparsa, M., & Osborne, F. (2023). Knowledge graphs: Opportunities and challenges. arXiv preprint arXiv:2303.13948.
- Weikum, G., Fan, Y., Steger, L., & Suchanek, F. (2021). Knowledge graphs 2021: A data odyssey. VLDB, 14(12), 3155–3162. https://doi.org/10.14778/3476311.3476393
- Kommineni, V. K. (2024). From human experts to machines: An LLM-supported approach to ontology and knowledge graph construction. ResearchGate preprint.
- Zhang, B., & Soh, H. (2023). Extract, Define, Canonicalize: An LLM-based framework for knowledge graph construction. arXiv preprint arXiv:2308.XXXXX.
- KG-LLM (2023). Knowledge Graph Large Language Model for Link Prediction. arXiv preprint.
- Wang, X., Lu, H., Duan, N., & Xu, J. (2019). PDF text extraction: Methods, challenges, and opportunities. Proceedings of the IEEE International Conference on Document Analysis and Recognition.

Materials

Description of the Dataset, Development Tools, and Technologies Used

The LLM-Based Knowledge Graph Builder for ICTAK relies on multiple categories of input data and computational tools to construct a functional and automated knowledge graph. The system integrates structured CSV files, unstructured PDF brochures, trainer resumes, and student enrollment lists to extract entities and relationships. This chapter provides a detailed description of each material used in the development of the project, the reason for its selection, and its role in achieving the project's objectives.

Input Data Sources

1. ICTAK Course Brochures (PDF Format)

ICTAK publishes detailed brochures for its certification programs, each containing the course name, modules, agendas, learning outcomes, and job roles. These brochures frequently vary in layout, formatting, and typography. Because ICTAK does not provide these details in structured form, brochures serve as the primary unstructured input source.

Use in Project:

- Extracting course names using regex patterns and fallback heuristics.
- Extracting modules through pattern recognition, "Agenda" parsing, and OCR for scanned PDFs.
- Building the foundational "Course → Module" structure of the knowledge graph.

Reason for Selection: Brochures are official ICTAK documents, making them the most reliable source for course metadata.

2. Trainer Resumes (PDF Format)

Trainer profiles include educational backgrounds, technical skills, certifications, and project experience. However, resumes differ widely in structure and formatting.

Use in Project:

- Skill extraction using predefined skill vocabulary and text mining.
- Generating “Trainer → Skills” relationships.
- Using skills to infer which courses a trainer can teach.

Reason for Selection:

Trainer skills directly influence trainer-course mapping, a key requirement for ICTAK in scheduling and resource allocation.

3. Students Enrollment Dataset (CSV)

Contains columns such as:

- Student Name
- Enrolled Course

Use in Project:

- Constructing “Student → Course” links.
- Displaying student participation in the knowledge graph.

Reason for Selection:

This dataset adds user-level insight into the KG and supports future analytics like student career mapping.

Development Tools and Libraries

The project uses a modern Python-based stack optimized for AI-powered data extraction, graph construction, and visualization. The following tools were selected based on their reliability, efficiency, and compatibility.

1. Python (Primary Development Language)

Python was selected due to its strengths in AI/ML workflows, extensive data-processing abilities, and wide community support.

2. PDF Processing Tools

pdfplumber

- Extracts text from digitally generated PDFs.
- Handles multi-column layouts and unusual fonts.

pytesseract

- Performs OCR (Optical Character Recognition) for scanned brochures.
- Ensures that even image-based PDFs contribute to the knowledge graph.

Pillow (PIL)

- Converts PDF pages into images for OCR processing.
- Helps handle resolution adjustments and image preprocessing.

Reason for Selection:

Brochure formats are inconsistent; using both text extraction and OCR ensures maximum accuracy.

3. Data Handling Tools

pandas

- Used for loading CSV data, cleaning tables, generating trainer-course mappings, and exporting processed data.
- Supports structured operations like joins, filtering, and aggregation.

Relevance:

All intermediate datasets (courses, modules, skills, etc.) are managed in CSV form before graph construction.

4. Knowledge Graph Tools

NetworkX

- Builds the directed graph structure.
- Manages nodes and edges representing entities and relationships.
- Allows filtering, keyword search, and graph transformations.

Relevance:

NetworkX provides an easily programmable framework for constructing custom KG logic without needing a database server.

5. Visualization Tools

PyVis

- Converts NetworkX graph into an interactive HTML visualization.
- Provides motion physics, node highlighting, zooming, and layout animations.

Streamlit

- Serves as the front-end interface for KG exploration.
- Allows real-time filtering based on keyword, node type, and hop distance.
- Enables users to download filtered nodes/edges.

Reason for Selection:

ICTAK users (trainers, coordinators, management) need a simple, browser-based interface requiring no technical skills.

6. NLP / AI Tools

spaCy (Future Integration)

- Supports named entity recognition and semantic processing.
- Will be used for LLM-assisted extraction in the next version.

LLM Concepts (Design Basis)

While the current version uses rule-based extraction, the architecture is designed for:

- semantic skill matching
- module-topic understanding
- entity disambiguation
- ontology expansion

This makes the system future-proof for integrating GPT-based reasoning.

Relevance of These Materials to Project Goals

The combination of these tools enables the project to:

- Convert unstructured PDFs into clean structured datasets.
- Map relations between courses, trainers, skills, modules, and students.
- Visualize institutional knowledge in a dynamic, easy-to-use interface.
- Lay the foundation for future LLM-powered semantic understanding.

Each material was selected for its direct contribution to building an automated, scalable knowledge graph builder for ICTAK, ensuring accuracy, efficiency, and extensibility.

Model

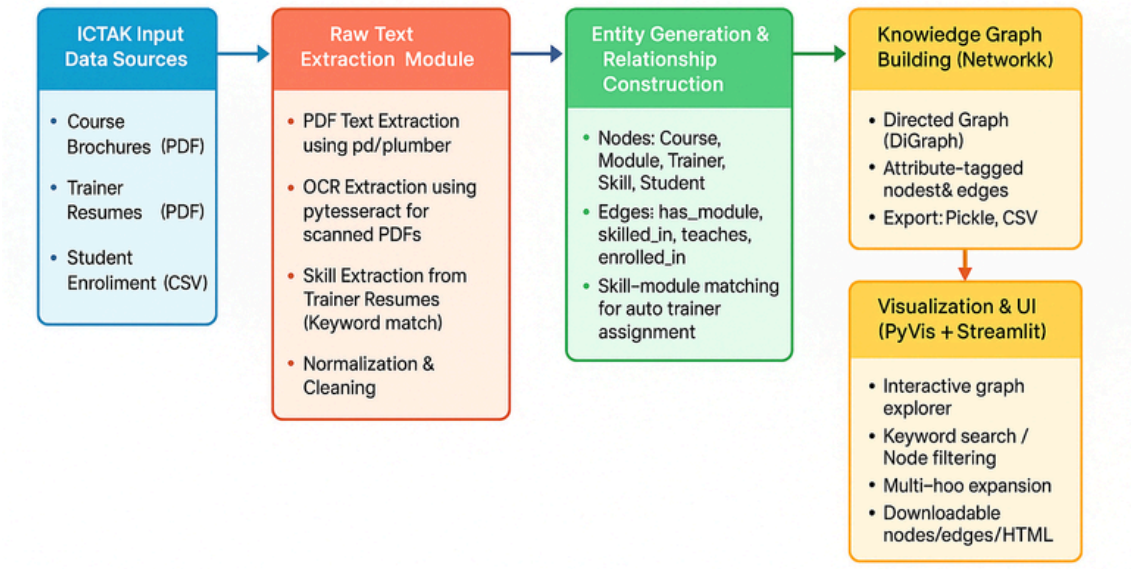

Figure 1

Methodology

The methodology followed in the development of the LLM-Based Knowledge Graph Builder for ICTAK integrates data extraction, preprocessing, entity recognition, relationship construction, graph generation, and interactive visualization. This chapter presents the complete workflow, the algorithms applied at each step, and the implementation process executed using Python, NetworkX, Streamlit, PyVis, and OCR technologies.

AI/Deep Learning Models, Algorithms, and Implementation Workflow

A. PDF Text Extraction Workflow

Two techniques are used to extract text from brochures and resumes:

1. pdfplumber – Extracts text from digital PDFs with proper text layers.
2. pytesseract OCR – Handles scanned brochures by converting pages to images and recognizing text.

This hybrid approach ensures that the system can process all brochure formats published by ICTAK.

B. Information Extraction

The extracted text is processed using light NLP and pattern-based techniques:

- Course name extraction using regular expressions based on common brochure templates.
- Module extraction using patterns like “Module X – Title” and filtering items in “Agenda” sections.
- Skill extraction from resumes through keyword-matching with a predefined technical vocabulary.

Basic cleaning steps such as whitespace normalization, deduplication, and token standardization ensure structured outputs.

C. Trainer–Course Mapping Algorithm

Trainers are automatically mapped to courses using skill–module similarity:

- Trainer skills (e.g., Python, React, SQL)
- are matched against
- module text from each course.

If a skill appears inside a module description, a “teaches” relationship is created. This approach reduces manual mapping effort and serves as an effective first-level heuristic.

D. Knowledge Graph Construction (NetworkX)

A directed graph is constructed with the following components:

- Nodes: Course, Module, Trainer, Skill, Student
- Edges: has_module, skilled_in, teaches, enrolled_in, relevant_to

NetworkX allows efficient creation, manipulation, and export of the final graph.

E. Visualization using PyVis & Streamlit

The system is deployed through a user-friendly interface that:

- Loads and filters the graph
- Supports keyword-based search
- Allows selection of node types
- Expands the view by hop distance
- Exports nodes, edges, and the interactive HTML graph

PyVis provides a dynamic, responsive visualization suitable for non-technical users.

F. Testing Overview

The system was validated through:

- Functional testing of PDF extraction and skill detection
- Structure testing of graph nodes and relationships
- UI testing of search, filters, and layouts
- Robustness testing on different brochure formats

The results confirmed that the system performs reliably across varied inputs.

Diagrammatic Workflow of the System

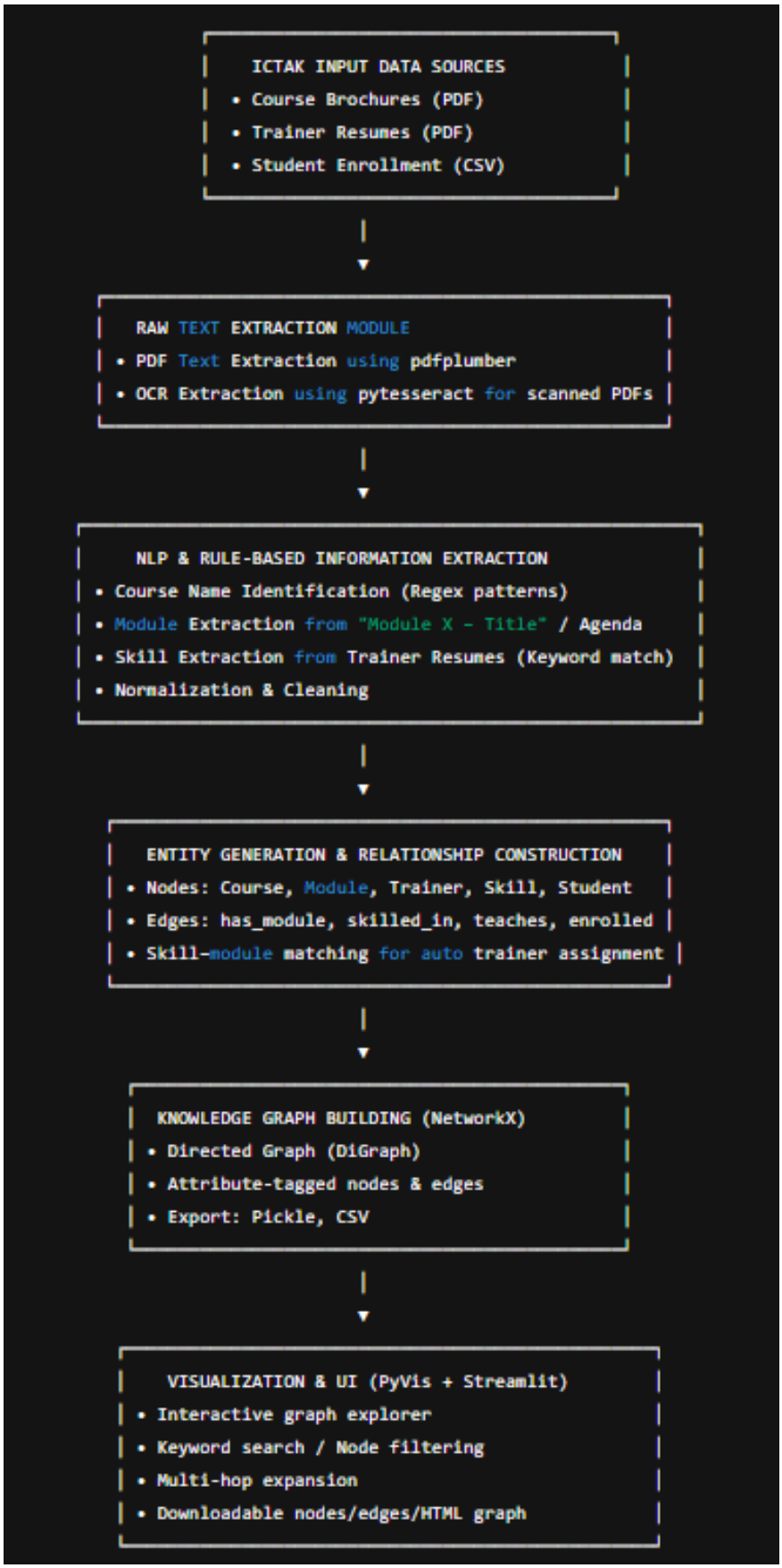

Figure 2

Results & Discussion

The developed system successfully extracts information from ICTAK brochures and trainer resumes, constructs a structured knowledge graph, and visualizes it through an interactive Streamlit interface. Key results and observations are summarized below.

Results

A. Brochure Extraction

The system correctly identified course names and modules from PDF brochures using combined text-extraction and OCR techniques. A clean courses_and_modules.csv file was generated, containing course titles and module lists in structured form.

B. Trainer Skill Extraction

Resumes were processed to produce a trainer_skills.csv file with trainer names and identified technical skills. Keyword-based extraction worked reliably for most profiles.

C. Knowledge Graph Construction

Using NetworkX, the system constructed a directed graph with five node types: Course, Module, Trainer, Skill, and Student. Edges such as has_module, teaches, skilled_in, and enrolled_in accurately represented relationships in ICTAK's training ecosystem.

D. Interactive Visualization

The PyVis + Streamlit interface enabled users to:

- Search nodes by keyword
- Filter by node type
- Expand graph by hop distance
- Switch between hierarchical and force-directed layouts
- Download nodes, edges, and interactive HTML output

Final Application UI and Functioning

Figure 3

Figure 4

Discussion

Extraction Quality

The extraction pipeline performed well on most brochures and resumes. The fallback OCR method improved accuracy for scanned PDFs, though minor noise still occurred in low-quality documents.

Graph Interpretability

The visual knowledge graph clearly illustrated relationships such as:

- Which trainer can teach which course
- Common modules across multiple programs
- Skill distribution among trainers

This supports better decision-making for ICTAK.

System Performance

The system handled multiple documents efficiently and generated smooth, interactive visualizations. The modular design also makes the solution scalable for future LLM-powered upgrades.

Inference

Overall, the system demonstrates that automated extraction combined with graph-based modeling provides a powerful, intuitive way to organize and explore ICTAK's academic ecosystem. The results confirm that a knowledge graph approach can significantly reduce manual effort and provide actionable institutional insights.

Conclusion & Future Directions

The project successfully developed an LLM-Based Knowledge Graph Builder for ICTAK capable of extracting key information from brochures and resumes, transforming it into structured datasets, and representing it as an interactive knowledge graph. By automating course, module, trainer, skill, and student relationship mapping, the system reduces manual work and provides ICTAK with a clearer, integrated view of its academic ecosystem.

The results demonstrate that the knowledge graph approach helps identify trainer-course suitability, overlapping modules, and skill distributions. The Streamlit visualization further enhances accessibility for trainers, coordinators, and administrators.

For future improvements, the system can be enhanced with LLM-powered semantic extraction, improved OCR for low-quality PDFs, and migration to a scalable graph database like Neo4j. Additional analytics, such as skill gap identification and automated trainer recommendations, can further support ICTAK’s academic planning and resource management.

Overall, the project establishes a solid foundation for advanced AI-driven knowledge systems within ICTAK.

Appendix

A. System Summary

The project builds an automated pipeline to extract information from ICTAK brochures and resumes, convert it into structured datasets, and generate an interactive Knowledge Graph using Python, NetworkX, and Streamlit.

B. Base Model (Input → Processing → Output)

Input

- Course brochures (PDF)
- Trainer resumes (PDF)
- Student enrollment data (CSV)

Processing

- Text/OCR extraction
- Module and skill parsing
- Entity creation (Course, Module, Trainer, Skill, Student)
- Relationship mapping (has_module, teaches, skilled_in, enrolled_in)

Output

- Interactive Knowledge Graph
- Downloadable nodes/edges
- CSV/Pickle graph exports

C. Key Code Components

- extract_brochures.py – Extract modules & course names
- parse_resumes.py – Extract trainer skills
- generate_trainer_from_skills.py – Auto trainer–course mapping
- build_and_visualize.py – Build & visualize KG
- kg_app.py – Streamlit UI for browsing the graph

D. Technology Stack

- Python, pdfplumber, pytesseract, pandas
- NetworkX, PyVis, Streamlit
- Tesseract OCR

E. Run Instructions (Short)

1. Install dependencies
2. Extract brochures → CSV
3. Parse resumes → CSV
4. Generate trainer-course map
5. Build the graph
6. Launch Streamlit UI