Implementation Plan: Brain Sparks Educational Recommender

Rugogamu Noela S23B8/016 B22775
Institution: Uganda Christian University (UCU)

December 2025

1. Project Overview

1.1 Project Name

Brain Sparks - Personalized Educational Recommender for Ugandan Learners

1.3 Project Scope

- Natural language query processing (Understand)
- Knowledge graph construction and traversal (Reason)
- Content based recommendation engine (Reason)
- User feedback integration (Learn)
- Web based interactive interface (Interact)

2. Work Breakdown Structure (WBS)

Level 1: Project Phases

```
Brain Sparks Project
 1.0 Problem Analysis & Design
 2.0 Cognitive System Implementation
 3.0 Evaluation & Documentation
 4.0 Deployment & Presentation
```

Level 2: Work Packages

1.0 Problem Analysis & Design (Part A - Week 1)

Table 1: WBS for Part A

WBS Code	Task	Deliverable	Days
1.1	Problem Analysis	`problem_analysis.pdf`	2
1.2	Data Pipeline & Content Curation	`01_data_pipeline.ipynb`, Dataset	2
1.3	NLP Engine Implementation (Understand)	`nlp_utils.py` (Topic, Intent, Context)	3

2.0 Cognitive System Implementation (Part B - Week 2)

Table 2: WBS for Part B and Evaluation

WBS Code	Task	Deliverable	Duration (Days)
2.1	Knowledge Graph (Reason)	`kg_utils.py`, KG	2
2.2	Recommender & Feedback (Learn)	`recommender.py`, `feedback.json`	2

Table 2 – continued

WBS Code	Task	Deliverable	Duration (Days)
2.3	Interactive UI (Interact)	`app.py` (Streamlit app)	2
3.1	Evaluation & Testing	`evaluation_report.pdf`, 20 Test Queries	2
3.2	Final Documentation	`ethical_analysis.pdf`, GitHub Upload	1

3. Project Timeline (Actual Execution)

Actual Execution Dates: November 18 – December 1, 2025

The project was executed in a two week sprint, structured around the four cognitive pillars.

0.1 Week 1: Foundations and The Understand Pillar (Approx. Nov 18-24)

- **Days 1-2 (Nov 18-19)**: Problem Analysis (`problem_analysis.pdf`) and defining the Cognitive Pipeline.
- **Days 3-4 (Nov 20-21)**: Dataset acquisition, cleaning, and transformation using `01_data_pipeline.ipynb`. This included curating 87 educational resources across 20 topics.
- **Days 5-7 (Nov 22-24)**: Implementation of the Understand pillar in `nlp_utils.py`. This module handles Natural Language Processing (NLP) to extract Topic, User Intent (such as 'learn', 'apply'), and Uganda Context.
- **Visualization**: The initial data distribution was visualized:

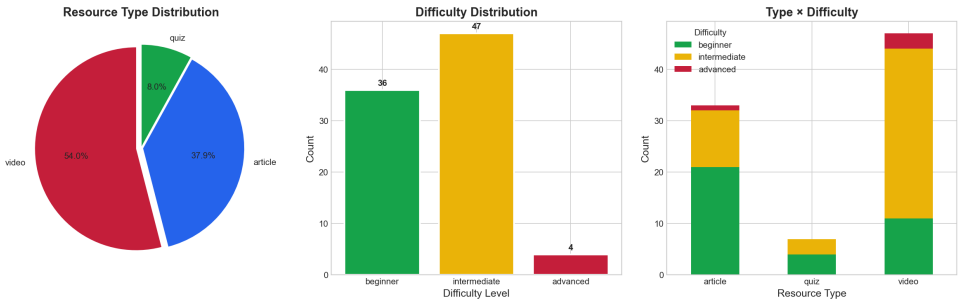

Figure 1: Educational Resources by Topic (Horizontal bar chart ranking topics by count, with Machine Learning at 9, Web Development at 8, Artificial Intelligence at 7, and others down to 2).

0.2 Week 2: Reasoning, Learning, Interaction, and Evaluation (Nov 25-Dec 1)

- **Day 8-9 (November 25-26): Reasoning Pillar and Knowledge Graph (Milestone 3)**
  - Implementation of the **Reason** pillar in `kg_utils.py` using `NetworkX`. The `EducationalKnowledgeGraph` class was created, adding nodes for topics, resources, and applications, and edges like `'is_about'`, `'applies_to'`, and prerequisites.
  - The final graph size included **335 Total Nodes** and **1028 Total Edges**.

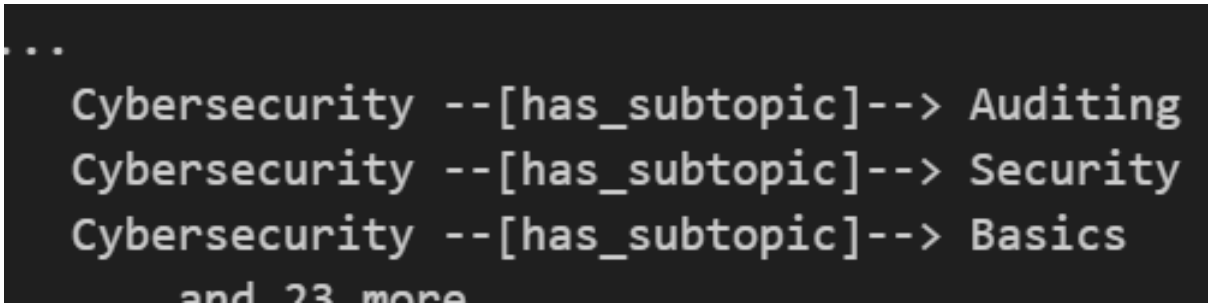

Figure 2: Relationship extraction: A snippet illustrating the semantic relationships extracted and stored in the graph, specifically showing 'Cybersecurity' connected to subtopics like 'Auditing', 'Security', and 'Basics' via the `--[has_subtopic]-->` edge.

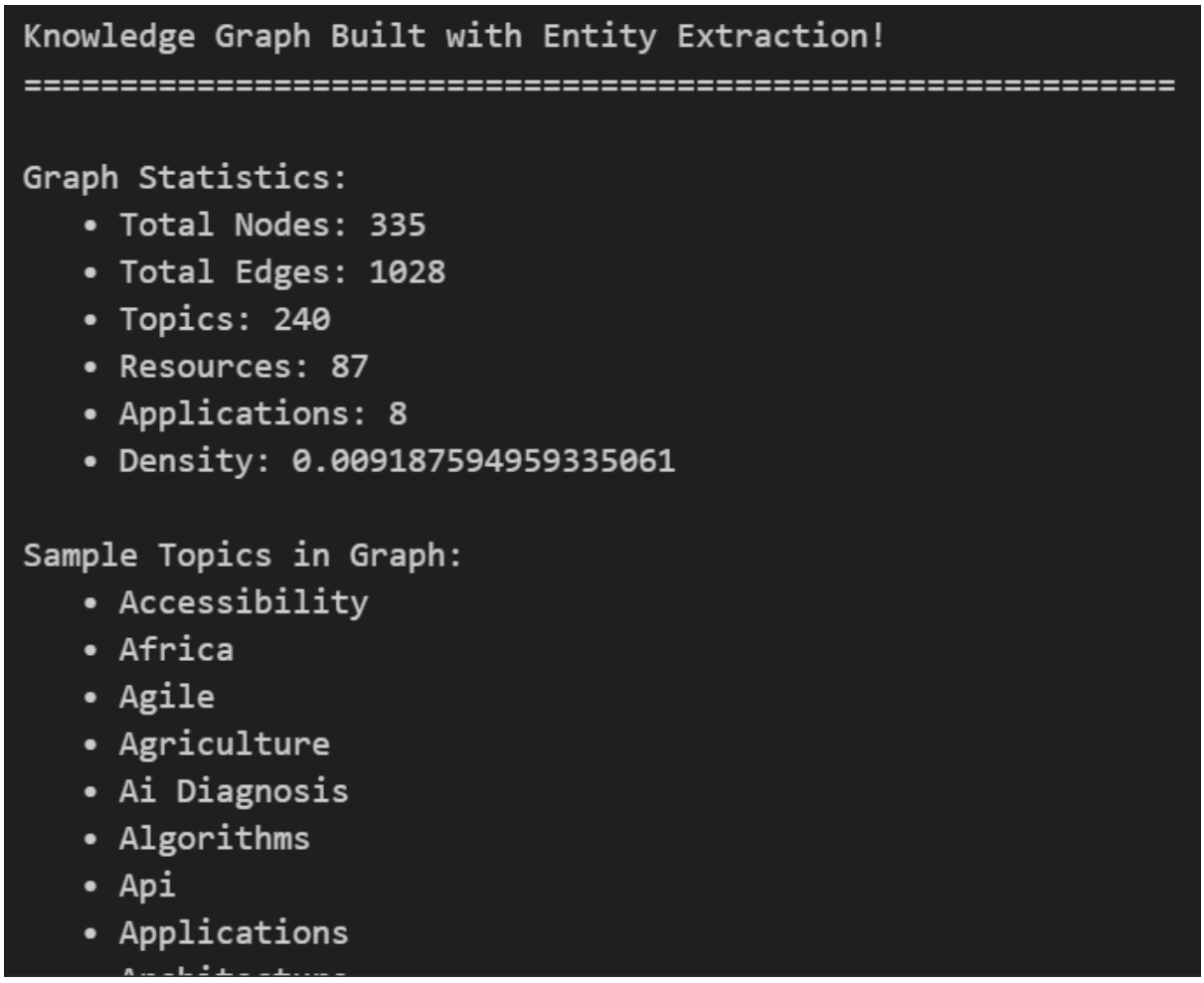

Figure 3: A console output detailing the structure and size of the Knowledge Graph built via entity extraction. Key metrics include 335 Total Nodes, 1028 Total Edges, and a list of sample topics like 'Agriculture', 'Algorithms', and 'Accessibility'.

- **Day 10-11 (November 27-28): Recommendation and Learning Pillar**
  - Creation of `recommender.py` using **TF-IDF** for content-based similarity scoring, integrated with knowledge graph traversal for context.

- Implementation of the **Learn** pillar using `FeedbackManager`, which stores user ratings in `feedback.json` and adjusts resource scores (boosting popular resources by up to 0.3).

- **Day 12-13 (November 29-30): Interaction Pillar and System Integration (Milestone 4)**
  - Development of the user interface in `app.py` using **Streamlit**. The UI includes: home (query input), results (structured learning paths with Explanation and Uganda relevance tabs), feedback submission, knowledge graph visualization (`PyVis`), and a statistics dashboard.

- **Day 14 (December 1): Evaluation, Refinement, and Documentation (Milestone 5)**
  - Generation of 20 test queries to measure performance metrics like `precision@3` using a `compare_with_baseline` function.
  - Visualization of results using `Plotly` charts.
  - Finalization of all reports (`evaluation_report.md`, `ethical_analysis.md`) and code upload to GitHub.

```
Testing Multiple Queries
============================================================

Query: "I want to learn machine learning for agriculture"
   → Topic: educational_technology (67%)
   → Uganda: Yes
   → Intent: learn

Query: "How can cybersecurity help protect mobile money in Uganda?"
   → Topic: fintech (67%)
   → Uganda: Yes
   → Intent: solve

Query: "Teach me about blockchain applications"
   → Topic: mobile_development (67%)
   → Uganda: No
   → Intent: learn

Query: "What is artificial intelligence?"
   → Topic: artificial_intelligence (33%)
   → Uganda: No
```

Figure 4: Multiple Query Analysis Results. Examples demonstrating the system's ability to analyze and categorize multiple user queries. Each query is mapped to a Topic, checked for Uganda Context, and assigned an Intent (such as 'learn' or 'solve').

4. Ethical Considerations

Throughout the implementation, ethics were prioritized by ensuring no personal data collection, maintaining a balanced dataset for fairness, and utilizing simple, accessible tools to keep the system available to all Ugandan learners.