AI-Powered Knowledge Graph Generation

In a University Context

using Neo4j Graph Database

Group 13 (Data Dynamos)

Chen Yao Quan A0238895H | Lai Ze Ming A0240558A | Liao Jun Yang A0239360E
Lim Qi Shan Chloe A0240530X | Lim Xin Yu A0238763U Ng Hannah A0239466R | Tan Xin Pei Alicia A0258615X

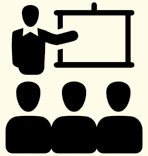
Problem Statement

Lack of a Comprehensive Module Recommendation System

- **Absence of Personalized Module Recommendation System**
  Currently, there is no all-in-one solution that integrates module selection with students' career and skill goals.

- **Misalignment Between Academics and Career Paths**
  Students struggle to select modules that match desired job roles and skill requirements.

- **Absence of Tailored Guidance**
  Lack of career-specific recommendations leads to inefficient academic choices.



- **Solution Goal**
  Develop a comprehensive recommendation system to help students choose modules aligned with their career aspirations and skill needs.

Entity & Relationship Extraction

- **Diverse ECM Data Formats Supported**
  Handles a variety of formats: CSV, JSON, XLSX, and PDF.

CSV	Original file types	Entities extracted
00- mock student data	csv	Student, Module, Faculty,
01 - mock module info	json	Module, Department, Faculty, Skill, Semester
02- mock department list	csv	Department, Faculty
03 - mock staff info	csv	Staff, Module, Department
04 - mock module reviews	json	Module, Staff, Skill
05 - nus undergraduate programs	csv	Major, Degree
06 - jobs and relevant skillset	xlsx	Job, Skill
07 - jobs and tech	xlsx	Job, Skill
08 - jobs and skills	xlsx	Job, Skill
09 - graduate employment survey	csv, pdf	Degree, Faculty, University

- **Configuration-Based Extraction for entity and relationship mapping**
  Customizable setup by mapping key entities to relevant relationships based on specific needs.

```yaml
relationship_mappings:
  student_faculty:
    from_col: student_entities
    to_col: faculty_entities
    from_type: Student
    to_type: Faculty
    relationship_type: STUDYING_UNDER
  student_major:
    from_col: student_entities
    to_col: major_entities
    from_type: Student
    to_type: Major
    relationship_type: MAJOR_IN
```

- **Skills Extraction**
  Accurate skills identification using exact matching (spaCy & regex).
  Fallback: Fuzzy matching for when exact matches fail.

Entity & Relationship Extraction

- **Used regex patterns to extract staff entities**
  - Reduces false positives for better data accuracy

```
# Regex pattern to capture staff names with titles like 'Prof', 'Dr', 'Lecturer', 'Tutor'
staff_pattern = re.compile(
    r"\b(Prof|Professor|Dr|Lecturer|Tutor|Instructor)\s*[A-Z][a-z]+(?:\s+[A-Z][a-z]+)?",
    re.IGNORECASE,
)
```

- **High Performance & Scalability**
  - Multiprocessing
  - Parallel & Chunk-Based Processing
    - ➔ Supports large datasets while maintaining efficiency

Graph Construction Engine

Expected output after entity and relationship extraction

Module entity: [('ABM5001','MODULE')]

Skill entity: [('Leadership','SKILL')]

Relationship: [{'from_type': 'Module', 'from_id': 'ABM5001', 'to_type': 'Skill', 'to_id': 'Leadership', 'type': 'SKILL_TAUGHT'}]

Graph Construction and Integration

- General Function which extracts out the entity type in the list
- It then creates the individual nodes in the neo4j database by referring to the ontology_config.json file which contains all the relevant entities and relationships which we want to have in the database
- Ontology example:

Graph Construction Engine

Use of buttons for users to access graph construction engine/ graph database

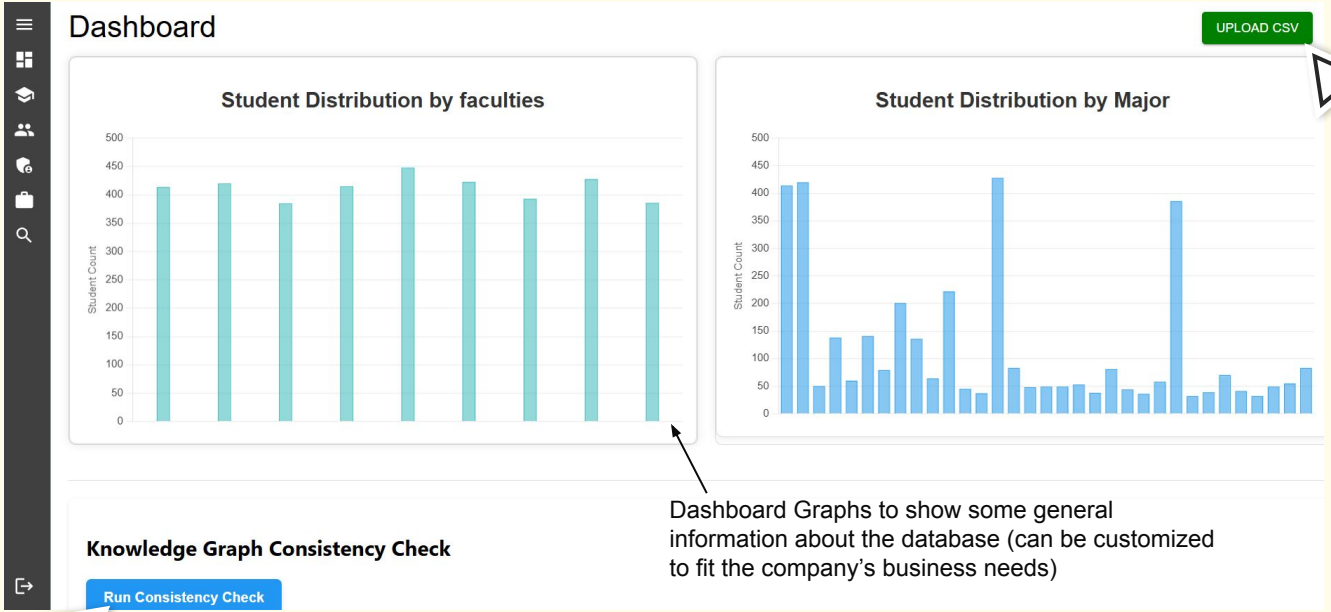
Deletes the individual node and all its relationships using it unique property (e.g. for Module, its unique property would be its Module Code)

Produces a form which the user can fill up to create the individual node and all its associated relationship

Allows the user to update any of the properties of that individual node which the user can access by inputting its unique property

Allows the user to upload any csv file to run the entity extraction and graph construction function based on the ontology and configurations set by the user

Dashboard Page



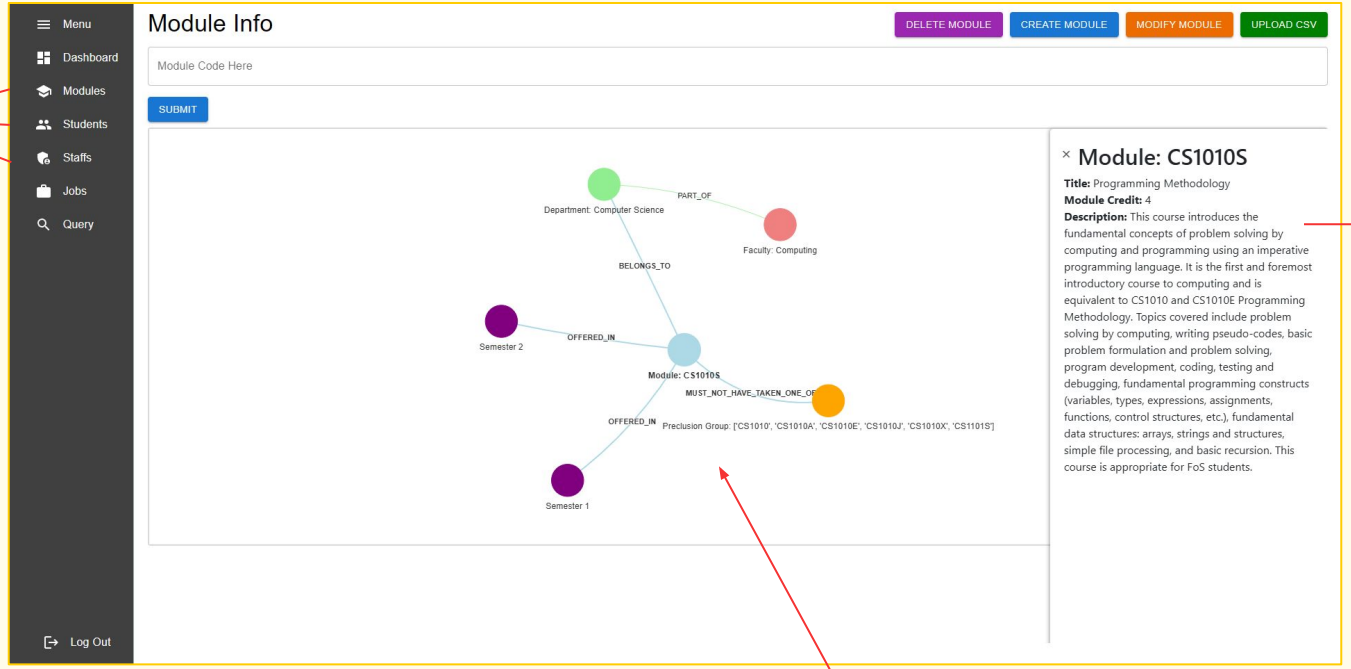
Runs the entity extraction and graph construction algorithms to integrate the data into the neo4j database

Dashboard Graphs to show some general information about the database (can be customized to fit the company's business needs)

Runs the consistency checking algorithm for the neo4j database. The results will be then be returned in a user-friendly format under the button.

Modules/Staff/Students Page

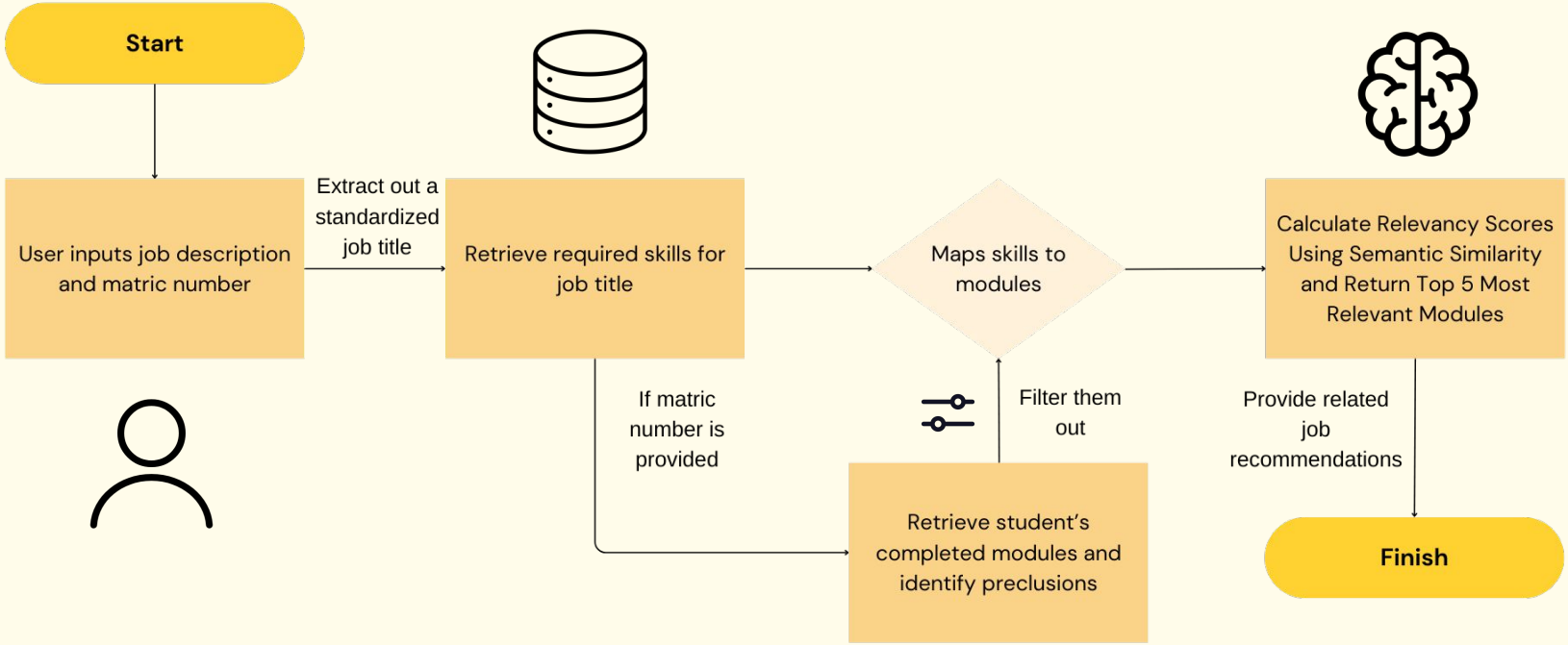
3 info graphs



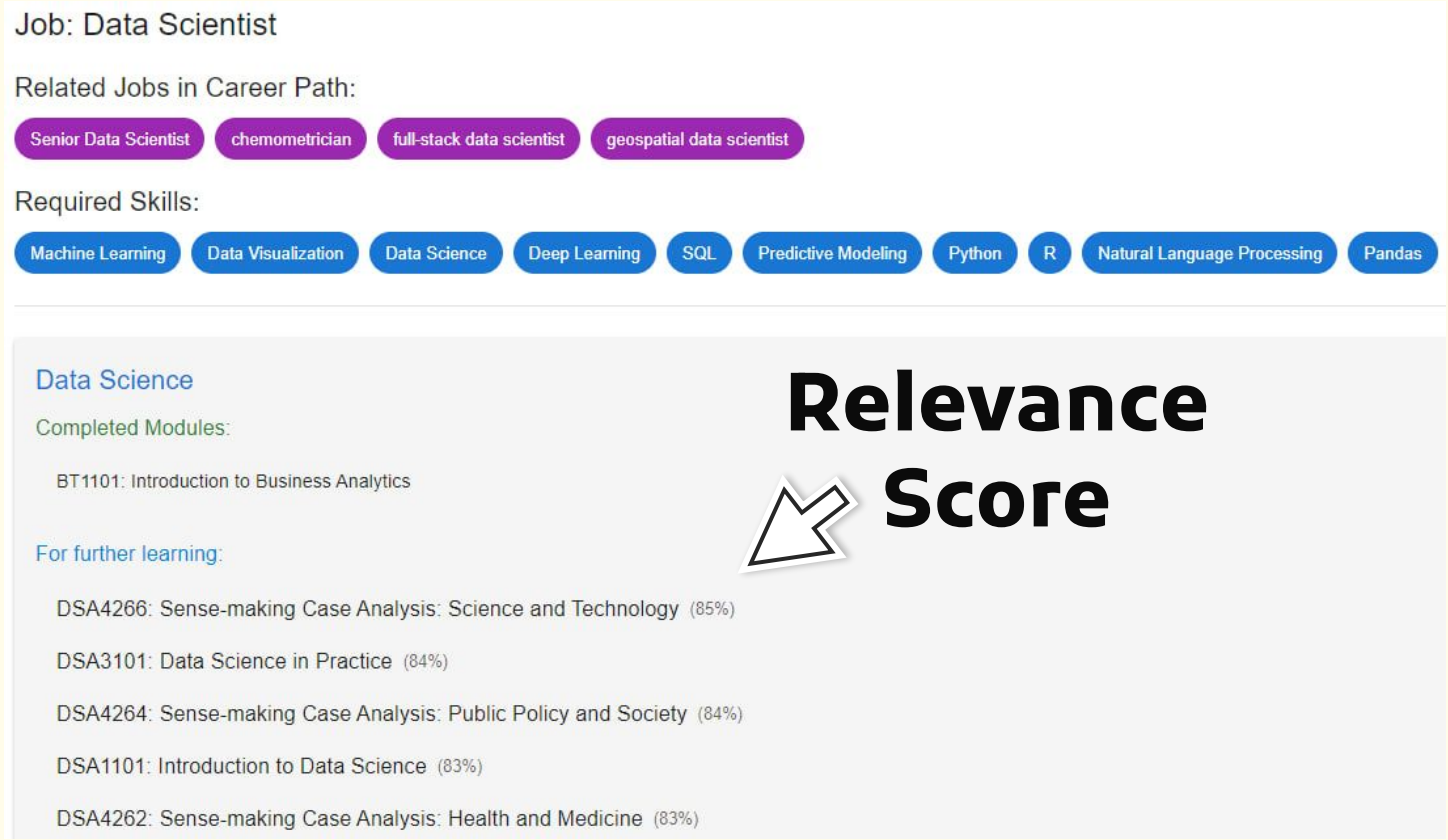
Details and descriptions

Interactive graph

Job Recommendations Engine



Job Recommendations Engine



Relevance Scoring:

- Uses Natural Language Processing for semantic similarity.
- Calculates similarity between skills and module descriptions.

Query (RAG) Engine

Type in any queries you have into the input

Query Bot

Input Prompt (*Case sensitive) *

tell me the modules that are taught by Marin Sergio Hernandez? Give me more more details about the modules

GET REPLIES

Response 1

description	This course is aimed at giving masters students in computer engineering an opportunity to work on a cutting edge project in collaboration with the industry over two se… …owledge and pick up new tools and skills in the process of solving problems that add value to themselves, the industry, and the society. Projects can involve a mix of research, design, and development aspects. Studen… …with an internal supervisor (who is a faculty member) and an external (industry) supervisor. Students will need to submit a report, as well as a presentation to the supervisors.
moduleCode	CEG5003
moduleCredit	8
title	Computer Engineering Project

Return a response

Purpose

Access information that specialised tabs cannot provide

STUDENTS

- Serves as valuable resource
- Present data summaries
- Compare Options

STAFFS

- Streamlines admin processes
- Manage various statistics
- Interacts with large datasets