

# Project Statement for Milestone 4

## Big Lens Massive Solutions

Antonio Barber, Adam Shtrikman, Sher Chhi Ly

### Overview:

At this stage of the project, teams should have developed a prototype of an end-to-end application with graphical user interface. Teams should have integrated the graphical user interface with NoSQL database and Hadoop/Spark framework.

In this report, teams are expected to provide updates on the graphical user interface and overall application.

### Report Topics:

The report should cover the following subtopics and answer the questions listed:

#### User Interface and Data Visualization:

**Describe the user interface and data visualization components of the software. The user interface should be interactive - take inputs from a user and provide outputs using algorithm results. The user interface should include data visualization, if applicable.**

The knowledge-based search engine consists of six key sections, each designed to enhance data discovery and analysis. The Knowledge-Based Subgraph Search feature allows users to enter a keyword and retrieve a paginated list of subgraphs from a Neo4j database, including entities, their relationships, and related entities that match the given keyword. Graph Visualization represents data in the form of nodes and edges, providing a visual representation of relationships and patterns within the graph. When any URI link is clicked on the web page, the corresponding graph is displayed in the Graph Visualization area, with nodes limited to 10 to avoid clutter, as some entities have hundreds of nodes. Find Similar Entities identifies entities that share specific relationships or characteristics with a given entity, including Relationship Results, which retrieves entities connected through a specified relationship type within a defined depth, and Multi-Hop Results, which explores entities reachable within a set number of hops from the starting entity. Find Within Two Connections checks whether two entities are directly connected or share a common neighbor, helping identify closely related nodes. Subgraph Matcher matches a subgraph pattern within the knowledge graph using a Cypher query to find specific structures or relationships. Finally, Node Similarity and Scoring calculates similarity between two nodes in the knowledge graph using cosine similarity based on shared neighbors, providing insights into how closely related two nodes are. All sections use a loading message to indicate progress, such as a loading bar, a spinning circle, or simply stating "loading results." All results are limited to 10 items per page; if more results are available, users can navigate using "Next" and "Previous" buttons.

## Knowledge-Based Subgraph Search

The user interface for the Knowledge-Based Subgraph Search allows users to input a keyword to initiate a search for subgraphs in a Neo4j database, containing entities, their relationships, and related entities that match the keyword. The retrieved results are displayed in a paginated list, including details such as the entity label, URI, and relationships. Users can click on an entity to visualize its relationships within a graph, rendered using Cytoscape.js. The graph includes nodes representing entities and edges for relationships. Pagination buttons ("Back" and "Next") enable users to navigate through the search results efficiently, and a loading indicator provides feedback during the data retrieval process.

# Knowledge Based Search Engine

## Knowledge-Based Subgraph Search - Takes a keyword and returns a

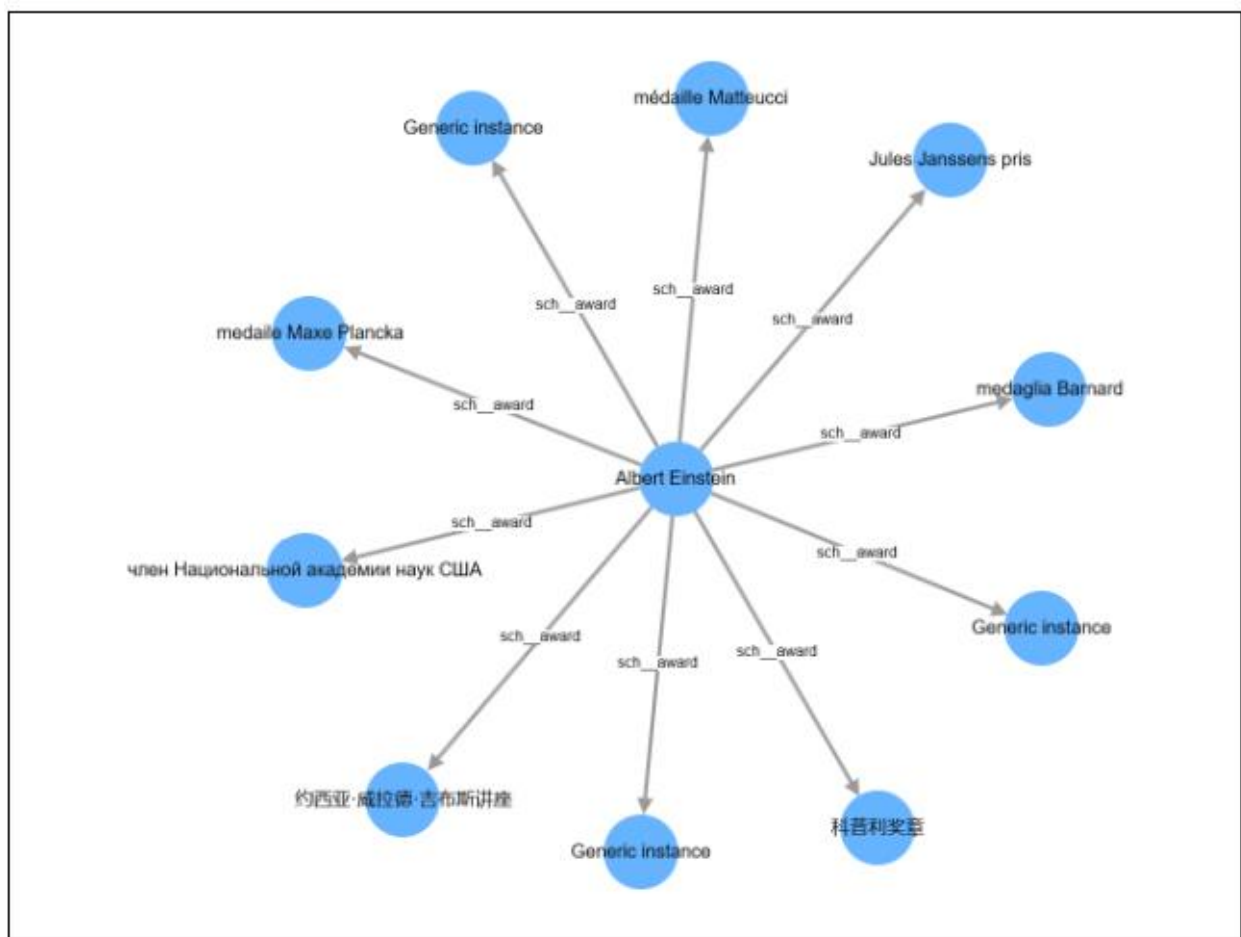
Enter Keyword:

- **Entity Label:** Albert Einstein-ziekenhuis  
**Entity URI:** [http://yago-knowledge.org/resource/Albert\\_Einstein\\_Israelite\\_Hospital](http://yago-knowledge.org/resource/Albert_Einstein_Israelite_Hospital)  
**Relation:** sch\_\_worksFor  
**Related Entity Label:** Nise Yamaguchi  
**Related Entity URI:** [http://yago-knowledge.org/resource/Nise\\_Yamaguchi](http://yago-knowledge.org/resource/Nise_Yamaguchi)
- **Entity Label:** Albert Einstein  
**Entity URI:** [http://yago-knowledge.org/resource/Albert\\_Einstein](http://yago-knowledge.org/resource/Albert_Einstein)  
**Relation:** sch\_\_memberOf  
**Related Entity Label:** Indiako Zientzien Akademia Nazionale  
**Related Entity URI:** [http://yago-knowledge.org/resource/Indian\\_National\\_Science\\_Academy](http://yago-knowledge.org/resource/Indian_National_Science_Academy)

## Graph Visualization

The user interface for the Graph Visualization component allows users to view relationships and patterns in the form of nodes and edges. It displays a visual representation of entities as nodes and their relationships as edges, providing an interactive view of data. Users can observe connections and interactions between entities, with nodes displayed in a consistent size, color, and label style to enhance readability. Additionally, when the mouse hovers over a node, it displays additional information about that node, allowing users to gain more context and insights into specific entities. The layout optimizes the graph for easy exploration, balancing the spacing between nodes and ensuring that relationships are clear. This visualization approach helps users intuitively understand and navigate complex data relationships.

**Graph Visualization** – Represents data as nodes and edges to reveal relationships and patterns.



## Find Similar Entities

The user interface for the "Find Similar Entities" feature allows users to enter an entity URI to search for entities that share specific relationships or characteristics with the given entity. Upon submitting the URI, the system retrieves results in three categories: Similar Entities, which lists entities with similar characteristics; Relationship Results, which displays entities connected to the given entity through a specified relationship type within a defined depth; and Multi-Hop Results, which shows entities reachable within a set number of hops from the starting entity. These results are presented in a list format, with pagination options to navigate through multiple pages. Each result includes the URI, which users can click on to visualize the relationships in a graph, offering an interactive way to explore and analyze entity connections.

**Find Similar Entities -** Identifies entities sharing specific relationships or characteristics with a given entity.

<http://yago-knowledge.org/re>

[http://yago-knowledge.org/resource/Manuel\\_Cardona](http://yago-knowledge.org/resource/Manuel_Cardona) **Label:** Manuel Cardona  
[http://yago-knowledge.org/resource/Franco\\_Rasetti](http://yago-knowledge.org/resource/Franco_Rasetti) **Label:** Franco Rasetti  
[http://yago-knowledge.org/resource/William\\_Ramsay](http://yago-knowledge.org/resource/William_Ramsay) **Label:** William Ramsay  
[http://yago-knowledge.org/resource/Augusto\\_Righi](http://yago-knowledge.org/resource/Augusto_Righi) **Label:** Augusto Righi  
[http://yago-knowledge.org/resource/Pief\\_Panofsky](http://yago-knowledge.org/resource/Pief_Panofsky) **Label:** Wolfgang K. H. Panofsky  
[http://yago-knowledge.org/resource/Orso\\_Mario\\_Corbino](http://yago-knowledge.org/resource/Orso_Mario_Corbino) **Label:** Orso Mario Corbino  
[http://yago-knowledge.org/resource/Henry\\_Augustus\\_Rowland](http://yago-knowledge.org/resource/Henry_Augustus_Rowland) **Label:** Henry Augustus Rowland  
[http://yago-knowledge.org/resource/Lawrence\\_Bragg](http://yago-knowledge.org/resource/Lawrence_Bragg) **Label:** William Lawrence Bragg  
[http://yago-knowledge.org/resource/Arthur\\_Compton](http://yago-knowledge.org/resource/Arthur_Compton) **Label:** Arthur Compton  
[http://yago-knowledge.org/resource/William\\_Thomson\\_u002C\\_1st\\_Baron\\_Kelvin](http://yago-knowledge.org/resource/William_Thomson_u002C_1st_Baron_Kelvin) **Label:** William Thomson, Lord Kelvin

**Relationship Results -** Retrieves entities connected through a specified relationship type within a defined depth.

[http://yago-knowledge.org/resource/William\\_Thomson\\_u002C\\_1st\\_Baron\\_Kelvin](http://yago-knowledge.org/resource/William_Thomson_u002C_1st_Baron_Kelvin) **Traversal Depth:** 2  
[http://yago-knowledge.org/resource/Bruno\\_Touschek](http://yago-knowledge.org/resource/Bruno_Touschek) **Traversal Depth:** 2  
[http://yago-knowledge.org/resource/John\\_William\\_Strutt\\_u002C\\_3rd\\_Baron\\_Rayleigh](http://yago-knowledge.org/resource/John_William_Strutt_u002C_3rd_Baron_Rayleigh) **Traversal Depth:** 2  
[http://yago-knowledge.org/resource/Arnold\\_Sommerfeld](http://yago-knowledge.org/resource/Arnold_Sommerfeld) **Traversal Depth:** 2  
[http://yago-knowledge.org/resource/Theodor\\_W\\_u002E\\_Haensch](http://yago-knowledge.org/resource/Theodor_W_u002E_Haensch) **Traversal Depth:** 2  
[http://yago-knowledge.org/resource/Wolfgang\\_Pauli](http://yago-knowledge.org/resource/Wolfgang_Pauli) **Traversal Depth:** 2  
[http://yago-knowledge.org/resource/Heinrich\\_Hertz](http://yago-knowledge.org/resource/Heinrich_Hertz) **Traversal Depth:** 2  
[http://yago-knowledge.org/resource/Pierre\\_Curie](http://yago-knowledge.org/resource/Pierre_Curie) **Traversal Depth:** 2  
[http://yago-knowledge.org/resource/Guglielmo\\_Marconi](http://yago-knowledge.org/resource/Guglielmo_Marconi) **Traversal Depth:** 2  
[http://yago-knowledge.org/resource/Abdus\\_Salam](http://yago-knowledge.org/resource/Abdus_Salam) **Traversal Depth:** 2

**Multi-Hop Results -** Explores entities reachable within a set number of hops from the starting entity.

[http://yago-knowledge.org/resource/Fellow\\_of\\_the\\_Royal\\_Society\\_of\\_Canada](http://yago-knowledge.org/resource/Fellow_of_the_Royal_Society_of_Canada) **Hops:** 3  
[http://yago-knowledge.org/resource/Accademia\\_dei\\_Lincei](http://yago-knowledge.org/resource/Accademia_dei_Lincei) **Hops:** 3  
[http://yago-knowledge.org/resource/Spanish\\_language\\_generic\\_instance](http://yago-knowledge.org/resource/Spanish_language_generic_instance) **Hops:** 3  
[http://yago-knowledge.org/resource/Mexican\\_Academy\\_of\\_Sciences](http://yago-knowledge.org/resource/Mexican_Academy_of_Sciences) **Hops:** 3  
[http://yago-knowledge.org/resource/Honorary\\_Doctorate\\_Of\\_The\\_Masaryk\\_University\\_Q64756748\\_generic\\_instance](http://yago-knowledge.org/resource/Honorary_Doctorate_Of_The_Masaryk_University_Q64756748_generic_instance) **Hops:** 3  
[http://yago-knowledge.org/resource/American\\_Physical\\_Society](http://yago-knowledge.org/resource/American_Physical_Society) **Hops:** 3  
[http://yago-knowledge.org/resource/Honorary\\_Doctor\\_Of\\_The\\_Autonomous\\_University\\_Of\\_Barcelona\\_Q47035798\\_generic\\_instance](http://yago-knowledge.org/resource/Honorary_Doctor_Of_The_Autonomous_University_Of_Barcelona_Q47035798_generic_instance) **Hops:** 3  
[http://yago-knowledge.org/resource/Royal\\_Society\\_of\\_Canada](http://yago-knowledge.org/resource/Royal_Society_of_Canada) **Hops:** 3  
[http://yago-knowledge.org/resource/Max\\_Planck\\_Institute\\_for\\_Solid\\_State\\_Research](http://yago-knowledge.org/resource/Max_Planck_Institute_for_Solid_State_Research) **Hops:** 3  
[http://yago-knowledge.org/resource/Guggenheim\\_Fellowship\\_generic\\_instance](http://yago-knowledge.org/resource/Guggenheim_Fellowship_generic_instance) **Hops:** 3

## Find Within Two Connections

The user interface for the "Find Within Two Connections" feature allows users to input two entities and determine whether they are directly connected or share a common neighbor. After entering the start and end URIs, users can submit the form to check the connection status. If a relationship or shared neighbor exists, the interface provides a summary of the connections. Furthermore, if a shortest path is found between the two entities, the interface offers an option to display this path on the existing graph visualization area, providing users with a clear view of the relationships within the knowledge graph.

**Find Within Two Connections -** Checks if two entities are directly connected or share a common neighbor.

Enter Start URI:  Enter End URI:

### Results:

Neighbors of [http://yago-knowledge.org/resource/State\\_of\\_Bahrain](http://yago-knowledge.org/resource/State_of_Bahrain): 2 neighbors

Neighbors of [http://yago-knowledge.org/resource/Order\\_of\\_King\\_Abdulaziz](http://yago-knowledge.org/resource/Order_of_King_Abdulaziz): 68 neighbors

Relationship Status: The nodes [http://yago-knowledge.org/resource/State\\_of\\_Bahrain](http://yago-knowledge.org/resource/State_of_Bahrain) and [http://yago-knowledge.org/resource/Order\\_of\\_King\\_Abdulaziz](http://yago-knowledge.org/resource/Order_of_King_Abdulaziz) are not within 2 connections.

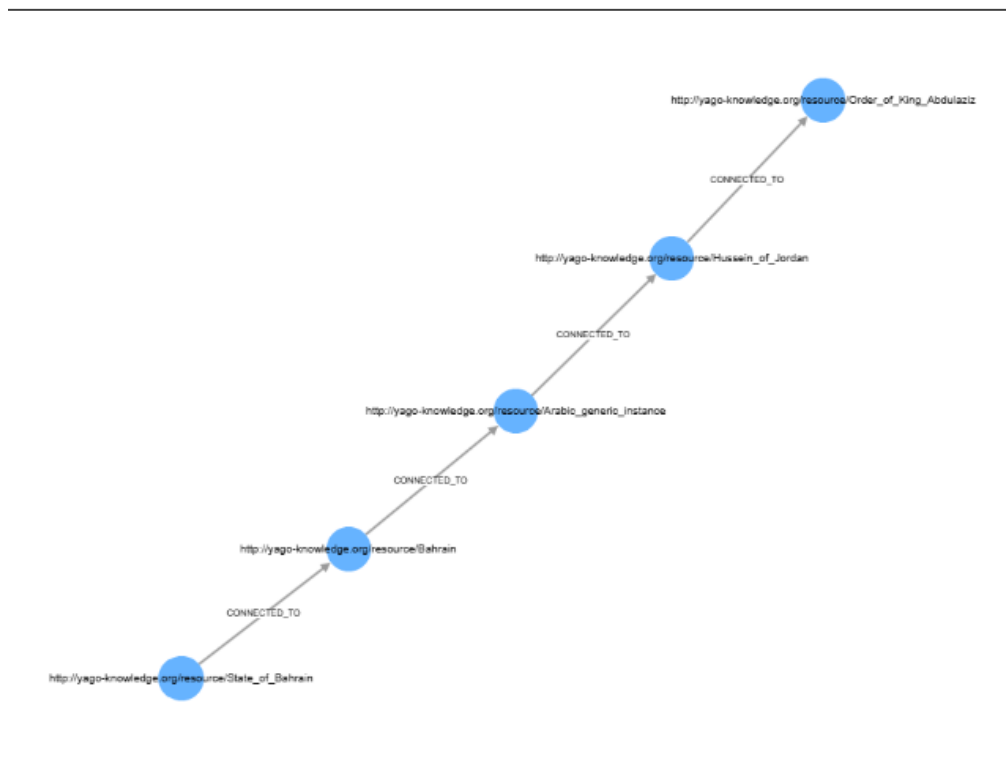
### Shortest Path:

Path Length: 4

Path Nodes (URIs):

- [http://yago-knowledge.org/resource/State\\_of\\_Bahrain](http://yago-knowledge.org/resource/State_of_Bahrain)
- <http://yago-knowledge.org/resource/Bahrain>
- [http://yago-knowledge.org/resource/Arabic\\_generic\\_instance](http://yago-knowledge.org/resource/Arabic_generic_instance)
- [http://yago-knowledge.org/resource/Hussein\\_of\\_Jordan](http://yago-knowledge.org/resource/Hussein_of_Jordan)
- [http://yago-knowledge.org/resource/Order\\_of\\_King\\_Abdulaziz](http://yago-knowledge.org/resource/Order_of_King_Abdulaziz)

**Graph Visualization -** Represents data as nodes and edges to reveal relationships and patterns.



## Subgraph Matcher

The user interface for the Subgraph Matcher allows users to search for a specific subgraph pattern within the knowledge graph using a Cypher query. Users input an entity name, which generates a Cypher query to match subgraph patterns in the Neo4j database. The results are presented in a list, showing details like the start node, the relationship type, and the related node. Each result allows users to visualize the corresponding subgraph by clicking on the nodes. Additionally, the interface provides "Next" and "Previous" buttons to navigate through multiple pages of search results, making it easy for users to explore a large set of matching subgraphs.

## Subgraph Matcher

Match a subgraph pattern in the knowledge graph using a Cypher query.

Example inputs:

Albert\_Einstein

United\_States

Moon\_Landing

Enter Entity Name:

- **Match 21:** Start Node: [http://yago-knowledge.org/resource/Albert\\_Einstein](http://yago-knowledge.org/resource/Albert_Einstein)  
Related Node: [http://yago-knowledge.org/resource/Accademia\\_nazionale\\_delle\\_scienze](http://yago-knowledge.org/resource/Accademia_nazionale_delle_scienze)  
Relationship: sch\_memberOf
- **Match 22:** Start Node: [http://yago-knowledge.org/resource/Albert\\_Einstein](http://yago-knowledge.org/resource/Albert_Einstein)  
Related Node: [http://yago-knowledge.org/resource/Indian\\_National\\_Science\\_Academy](http://yago-knowledge.org/resource/Indian_National_Science_Academy)  
Relationship: sch\_memberOf
- **Match 23:** Start Node: [http://yago-knowledge.org/resource/Albert\\_Einstein](http://yago-knowledge.org/resource/Albert_Einstein)  
Related Node: [http://yago-knowledge.org/resource/Prussian\\_Academy\\_of\\_Sciences](http://yago-knowledge.org/resource/Prussian_Academy_of_Sciences)  
Relationship: sch\_memberOf
- **Match 24:** Start Node: [http://yago-knowledge.org/resource/Albert\\_Einstein](http://yago-knowledge.org/resource/Albert_Einstein)  
Related Node: [http://yago-knowledge.org/resource/French\\_Academy\\_of\\_Sciences](http://yago-knowledge.org/resource/French_Academy_of_Sciences)  
Relationship: sch\_memberOf
- **Match 25:** Start Node: [http://yago-knowledge.org/resource/Albert\\_Einstein](http://yago-knowledge.org/resource/Albert_Einstein)  
Related Node: [http://yago-knowledge.org/resource/Royal\\_Swedish\\_Academy\\_of\\_Sciences](http://yago-knowledge.org/resource/Royal_Swedish_Academy_of_Sciences)  
Relationship: sch\_memberOf
- **Match 26:** Start Node: [http://yago-knowledge.org/resource/Albert\\_Einstein](http://yago-knowledge.org/resource/Albert_Einstein)  
Related Node: [http://yago-knowledge.org/resource/American\\_Academy\\_of\\_Arts\\_and\\_Sciences](http://yago-knowledge.org/resource/American_Academy_of_Arts_and_Sciences)  
Relationship: sch\_memberOf
- **Match 27:** Start Node: [http://yago-knowledge.org/resource/Albert\\_Einstein](http://yago-knowledge.org/resource/Albert_Einstein)  
Related Node: [http://yago-knowledge.org/resource/National\\_Academy\\_of\\_Sciences](http://yago-knowledge.org/resource/National_Academy_of_Sciences)  
Relationship: sch\_memberOf
- **Match 28:** Start Node: [http://yago-knowledge.org/resource/Albert\\_Einstein](http://yago-knowledge.org/resource/Albert_Einstein)  
Related Node: [http://yago-knowledge.org/resource/Bavarian\\_Academy\\_of\\_Sciences\\_and\\_Humanities](http://yago-knowledge.org/resource/Bavarian_Academy_of_Sciences_and_Humanities)  
Relationship: sch\_memberOf
- **Match 29:** Start Node: [http://yago-knowledge.org/resource/Albert\\_Einstein](http://yago-knowledge.org/resource/Albert_Einstein)  
Related Node: [http://yago-knowledge.org/resource/Royal\\_Netherlands\\_Academy\\_of\\_Arts\\_and\\_Sciences](http://yago-knowledge.org/resource/Royal_Netherlands_Academy_of_Arts_and_Sciences)  
Relationship: sch\_memberOf
- **Match 30:** Start Node: [http://yago-knowledge.org/resource/Albert\\_Einstein](http://yago-knowledge.org/resource/Albert_Einstein)  
Related Node: [http://yago-knowledge.org/resource/German\\_Democratic\\_Party](http://yago-knowledge.org/resource/German_Democratic_Party)  
Relationship: sch\_memberOf

## Node Similarity and Scoring

The user interface for the "Node Similarity and Scoring" feature allows users to input two URIs representing nodes in the knowledge graph. Users enter the URIs into text fields and click a button to initiate the similarity calculation, which is based on cosine similarity between shared neighbors of the nodes. The interface provides a loading message during processing and, once completed, displays the similarity score between the nodes. Additionally, the results include clickable links for each URI that enable users to visualize the respective node and its connections in the graph, offering insights into the relationship between the two nodes.

**Find Within Two Connections -** Checks if two entities are directly connected or share a common neighbor.

Enter Start URI:  Enter End URI:

### Results:

Neighbors of [http://yago-knowledge.org/resource/State\\_of\\_Bahrain](http://yago-knowledge.org/resource/State_of_Bahrain): 2 neighbors

Neighbors of [http://yago-knowledge.org/resource/Order\\_of\\_King\\_Abdulaziz](http://yago-knowledge.org/resource/Order_of_King_Abdulaziz): 68 neighbors

Relationship Status: The nodes [http://yago-knowledge.org/resource/State\\_of\\_Bahrain](http://yago-knowledge.org/resource/State_of_Bahrain) and [http://yago-knowledge.org/resource/Order\\_of\\_King\\_Abdulaziz](http://yago-knowledge.org/resource/Order_of_King_Abdulaziz) are not within 2 connections.

Shortest Path:

Path Length: 4

Path Nodes (URIs):

- [http://yago-knowledge.org/resource/State\\_of\\_Bahrain](http://yago-knowledge.org/resource/State_of_Bahrain)
- <http://yago-knowledge.org/resource/Bahrain>
- [http://yago-knowledge.org/resource/Arabic\\_generic\\_instance](http://yago-knowledge.org/resource/Arabic_generic_instance)
- [http://yago-knowledge.org/resource/Hussein\\_of\\_Jordan](http://yago-knowledge.org/resource/Hussein_of_Jordan)
- [http://yago-knowledge.org/resource/Order\\_of\\_King\\_Abdulaziz](http://yago-knowledge.org/resource/Order_of_King_Abdulaziz)



## User Queries and Results:

Provide all user queries and their results to describe the overall functionality of the software.

### Node Similarity and Scoring (node\_similarity.py)

- **Function: calculate similarity**
- **User Query/Input:** Two URIs are provided to calculate their similarity.
- **Output:** A cosine similarity score based on shared neighbors.
- **UI Representation:** The similarity score is displayed as a numerical value, ranging from 0 to 1, along with clickable links to visualize both nodes. The closer the score is to 1, the more similar the nodes are.

### Subgraph Matcher (subgraph\_matcher.py)

- **Function: match subgraph**
- **User Query/Input:** A URI or entity name provided by the user to match a specific subgraph pattern.
- **Output:** A list of subgraph matches, showing node URIs, relationships, and related nodes.
- **UI Representation:** The results are presented as a visual representation of the matched subgraph, highlighting nodes and their relationships in an interactive graph.

### Find Within Two Connections (within\_two.py)

- **Function: detailed shortest path**
- **User Query/Input:** The user provides two URIs to determine the shortest path between them.
- **Output:** The length of the shortest path and a list of nodes involved in the path.
- **UI Representation:** The shortest path is represented as a list of nodes, each of which is clickable, allowing the user to visualize the path as a whole in the graph.
- **Function: get neighbors count**
- **User Query/Input:** The user provides a node URI to calculate its neighbors.
- **Output:** The number of neighbors for the given node.
- **UI Representation:** The neighbor count is shown, with clickable links to explore each neighboring entity in more detail.
- **Function: are neighbors**
- **User Query/Input:** The user provides two URIs to check if they are directly connected.
- **Output:** A boolean value indicating if the nodes are neighbors.
- **UI Representation:** The result is displayed as a message that states whether or not the nodes are directly connected, and users can visualize both nodes through clickable links.
- **Function: has common neighbors**



- **User Query/Input:** The user provides two URIs to check if they share common neighbors.
- **Output:** A boolean result indicating whether there is a common neighbor.
- **UI Representation:** The output is displayed as a message, and links are available for visualizing the nodes and exploring any shared neighbors.

### Knowledge-Based Subgraph Search ([subgraph\\_search.py](#))

- **Function: search\_subgraph**
- **User Query/Input:** A keyword is entered by the user to search for matching entities and relationships.
- **Output:** A list of matching entities, including labels, URIs, relationships, and related entities.
- **UI Representation:** The results are presented in a paginated list format, with clickable links that let users visualize each entity in the graph. The pagination buttons allow users to navigate through multiple results.

### Find Similar Entities ([similar\\_search.py](#))

- **Function: explore\_multi\_hop**
- **User Query/Input:** A starting entity URI and a number of hops.
- **Output:** Entities reachable from the given starting entity within the specified number of hops.
- **UI Representation:** The reachable entities are presented along with their associated hop count. Users can click on links to visualize these entities in the graph.
- **Function: find\_similar\_entities**
- **User Query/Input:** A URI provided by the user to identify similar entities.
- **Output:** A list of similar entities that share certain attributes (e.g., awards) with the given entity.
- **UI Representation:** The similar entities are displayed as a list with clickable URIs, allowing users to explore the similar nodes further.
- **Function: relationship\_specific\_search**
- **User Query/Input:** Starting URI, type of relationship, and depth are provided to explore entities connected via specific relationships.
- **Output:** A list of connected entities showing their traversal depth.
- **UI Representation:** The connected entities are displayed as a list, including traversal depth, and users can explore these entities by clicking the links.

### Subgraph Visualization: [visualizeSubgraph](#) and [get\\_entity\\_subgraph](#)

- **Function: get\_entity\_subgraph (in subgraph\_search.py)**

- **User Query/Input:** The user provides a URI to visualize the associated subgraph.
- **Output:** Retrieves all the entities and their relationships linked to the specified URI.
- **UI Representation:** This is represented visually using an interactive graph format. Nodes are presented as circles, with edges depicting the relationships between them. The user can interact with the nodes to see details such as labels and URIs. This allows a comprehensive view of how an entity is connected within the graph.

#### Function: [visualizeSubgraph \(JavaScript Frontend Function\)](#)

- **User Query/Input:** Typically called when the user clicks on an entity from a list of search results to see its connections.
- **Output:** Retrieves and visualizes the corresponding subgraph.
- **UI Representation:** The output is displayed in the graphContainer element, utilizing Cytoscape.js for graph visualization. It includes nodes (entities) and edges (relationships), with styling options such as colors and arrow shapes to make the graph easy to interpret. The graph layout uses a force-directed arrangement, providing an intuitive representation of how entities are connected.

#### Source Code:

- a. Provide the source code of your application prototype, including user interface, data ingestion, data query and analytics algorithms, in a Zip file. Make sure to exclude the dataset.

#### Peer Evaluation:

Each team members should complete the CATME Peer Evaluation survey.

#### Grading:

- 50 pts: Team has successfully implemented an interactive user interface with visualization. Team has also successfully integrated the user interface with database and algorithms. Team has successfully tested the functionality of its solution with user-based queries. The project milestone document provides all information with relevant description, pseudo code / code snippets and diagrams / images.