



# **An Ontology-Driven Question Answering System for Computer Network Module**

**M.I.M.Nowshad**

**(Reg. No.: MS20908942)**

**MSc in IT**

**Specialized in Information Technology**

**Supervisor:**

Mr. Samantha Rajapaksha

Senior Lecturer

Department of IT

Faculty of Graduate Studies and Research

December 2021

**Department of IT**

**Faculty of Graduate Studies and Research**

**Sri Lanka Institute of Information Technology**

# Dedication

I am so indebted to express my deepest gratitude and dedication to some of the special people who worked with me, guided, and encouraged me on this important moment. Without their support and guidance, it would not have been possible to complete the project on time. Therefore, the help, advice, and guidance provided by the persons below are always appreciated.

- ❖ To my beloved parents who have always been supportive of me spiritually and mentally all the time.
- ❖ To the family who supported me morally and financially - especially to my wife Mrs. Fathima Sajitha.
- ❖ Most importantly to my supervisor who consistently gave instructions, guided, and fine-tuned me into this project from the beginning. - This project would not have been feasible without his consistent advice and guidance.

Finally, I humbly dedicate this dissertation to the Almighty Allah SWT. On the basis of being a firm believer that there is nothing in this world equal to his grace, blessing, and forgiveness.

**M.I.M.Nowshad, December 2021**

# Declaration

The dataset used in this thesis was obtained from the website of the National Institute of Education by access their official website <http://www.nie.lk>.

- (<http://www.nie.lk/pdffiles/tg/eGr12TG%20ICT.pdf>,
- <http://www.nie.lk/pdffiles/tg/eGr13TG%20ICT.pdf>)

The ontology database was created using these data in Protégé software. However, the data analysis, model design, development, and interpretation are solely my own work. Any contributions from the research papers, other sources, and diagrams are explicitly referenced in the text.

I am aware of and understand the Sri Lanka Institute of Information Technology (SLIIT), university's policy on plagiarism and certify that this thesis is my own work, except where indicated by the reference, and the work presented has not been submitted in support of another degree or qualification from this or any other university or institute of learning.

**M.I.M.Nowshad, December 2021**

# Quote

**“The essence of success begins with real exploration”**

- **Mohamed Nowshad**

# Acknowledgment

I owe my heartfelt gratitude at the outset to my esteemed supervisor Eng. U.U. Samantha Rajapaksha, Coordinator/Senior Lecturer, Faculty of Graduate Studies and Research, Sri Lanka Institute of Information Technology, has supported and advised my MSc study and research in many ways since its inception. with only the help of my supervisor's proper guidance, motivation, enthusiasm, and sharing of subject knowledge, I was able to complete this dissertation in a timely and elegant manner.

My heartfelt gratitude also goes to each and every lecturer at the Sri Lanka Institute of Information Technology who have consistently conducted the lectures to us in the midst of many challenges over the past two years.

I am truly grateful to SLIIT for taking such great care to complete this course on time even in the midst of the epidemics situation that has spread in our country. They have employed a number of strategies and techniques to achieve this goal without interrupting or delaying the learning-teaching activities and in which they succeeded.

I would also like to thank my beautiful family at this moment: To my parents who gave birth to me, Mr. M.M. Ismail, and Mrs. M.H.N. Lathifa. Their prayers and moral supports are the greatest strength I have, and also very thankful to my father-in-law and mother-in-law for their patience and supports throughout this journey.

Last but not the least, I would like to be greatly thankful to my wife, Mrs. M.F. Sajitha, and to my loving kids. The cooperation and encouragement they have given me to complete this journey successfully are unparalleled.

**M.I.M.Nowshad, December 2021**

# Table of Contents

Dedication .....	II
Declaration .....	III
Acknowledgment .....	V
Table of Contents .....	VI
List of Figures .....	IX
List of Tables .....	X
List of Abbreviations, Acronyms.....	XI
Abstract .....	XII
Chapter 01: Introduction .....	1
1.1 Chapter Overview .....	2
1.2 Problem Domain .....	2
1.3 Features of the prototype.....	3
1.4 Resource Requirements.....	4
1.4.1 Hardware Requirements .....	4
1.4.2 Software Requirements .....	4
1.4.3 Data Requirements .....	4
1.5 Project Document Structure .....	5
1.6 Chapter Summary.....	6
Chapter 2: Research Questions .....	7
2.1 Chapter Overview.....	8
2.2 Project Aim.....	8
2.3 Research Questions .....	8
2.4 Chapter Summary .....	9
Chapter 3: Literature Review.....	10
3.1 Chapter Overview.....	11

3.2 Background .....	11
3.2.1 Natural Language Processing - NLP.....	11
3.2.2 Question Answering System – QAS.....	13
3.2.3 Ontology .....	17
3.2.4 RDF, RDFS, and OWL.....	19
3.2.5 SPARQL .....	22
3.3 Literature Review .....	23
3.4 Chapter Summary.....	27
Chapter 4: Research Objectives .....	28
4.1 Chapter Overview.....	29
4.2 Main objective .....	29
4.3 Sub objectives.....	29
4.4 Chapter summary .....	30
Chapter 5: Research Methodology.....	31
5.1 Chapter Overview.....	32
5.2 The phases of the QA System .....	32
5.3 Data Processing .....	33
5.4 Question Processing .....	41
5.5 Ontology Mapper .....	43
5.6 Answer Retrieval.....	43
5.7 Chapter Summary.....	48
Chapter 6: Results and Evaluation.....	49
6.1 Chapter Overview.....	50
6.2 Results and Evaluation .....	50
6.3 Chapter Summary.....	53
Chapter 7: Timeline .....	54
Chapter 8: Discussion and Conclusion .....	55

Appendix A: Model Development.....	57
Appendix B: SPARQL Queries and Results.....	59
Appendix C: The Sample Test Questions.....	65
Appendix D: Complete RDFS for Network Module Ontology.....	67
Bibliography .....	96



# List of Figures

Figure 1.1: The Feature of QA Model .....	3
Figure 3.2: Text Pre-Processing Process .....	12
Figure 3.3: General Architecture of the QA System.....	13
Figure 4.1: The Structure of RDF .....	20
Figure 5.1: The Proposed QA System Architecture .....	33
Figure 5.2: Ontology Classes in Protégé.....	35
Figure 5.3: The Complete Class Diagram In Ontology .....	36
Figure 5.4: Object Properties of the Ontology .....	37
Figure 5.5: Data Properties of the Ontology .....	38
Figure 5.6: Data Restriction.....	39
Figure 5.7: Part of the Result of Protege in SPARQL Server.....	40
Figure 5.8: Ontology Information in Python .....	41
Figure 5.9: SPARQL query and Answer in Protege Software.....	44
Figure 5.10: Sparql Query and Answer in Fuseki Server .....	44
Figure 5.11: Sample Code Snippet of Owlready2 and SPARQL Query .....	45
Figure 5.12: Sample Answer for the SPARQL Query in Python .....	46
Figure 5.13: Question Screen of Web-Based System.....	47
Figure 5.14: Answer Screen of the Web-Based System .....	48
Figure 6.1: The Performance Comparison Chart.....	52
Figure 7.1: The Timeline Of The Project.....	54

# List of Tables

Table 3.1: The difference between RDBMS and ontology.....	19
Table 3.2: The Comparison of RDF, RDFS, AND OWL.....	21
Table 5.1: Summary of the Ontology Metrics .....	35
Table 5.2: The Sample Question Analysis.....	42
Table 6.1: Web-Based QA System Results .....	50
Table 6.2: Ontology-Based QA System Results.....	51
Table 6.3: Performance Indicators of Both Systems .....	52

# List of Abbreviations, Acronyms

<b>NLP</b>	-	Natural Language Processing
<b>IR</b>	-	Information Retrieval
<b>IE</b>	-	Information Extraction
<b>OWL</b>	-	Web Ontology Language
<b>QAS</b>	-	Question Answering System
<b>SRL</b>	-	Semantic Role Labelling
<b>KB</b>	-	Knowledge Base
<b>W3C</b>	-	Wide Web Consortium
<b>RDF</b>	-	Resource Description Framework
<b>RDFS</b>	-	Resource Description Framework Schema
<b>SPARQL</b>	-	SPARQL protocol and RDF query language
<b>URIs</b>	-	Uniform Resource Identifiers
<b>WWW</b>	-	World Wide Web
<b>XML</b>	-	eXtensible Markup Language
<b>AQUA</b>	-	Automated QUestion Answering System
<b>NER</b>	-	Named Entity Recognition
<b>AI</b>	-	Artificial Intelligence
<b>POST</b>	-	Part of Speech Tagging
<b>FAQ</b>	-	Frequently Asked Questions
<b>NLTK</b>	-	Natural Language Toolkit
<b>API</b>	-	Application Programming Interface
<b>HTML</b>	-	Hyper Text Markup Language
<b>REFO</b>	-	Regular Expressions For Objects

# Abstract

Data is found everywhere around us. Furthermore, these data are commonly found in structured, semi-structured and unstructured formats. Data are used in our daily lives in a variety of ways. Various search engines are used to search and retrieve the data we need. However, instead of search engines giving accurate and precise answers to the user queries, they often returned the list of links. Then, the user clicks on each link one by one to get the answer he wants. This method of searching can sometimes not giving the precise answers to the user queries, or the user may have to spend more time in searching for the answer, and hence the users may experience discomfort.

This situation can be avoided by using the semantic concept. The normal web data are machine-readable and can be understood by humans, whereas the semantic web information are machine-readable and understandable. Semantic web technologies are often used to create data stores, build vocabularies, and write rules for manipulating data. Ontology is the main component of the semantic web. Ontology can be described as the structure of knowledge-representation of a particular domain or subject. It clearly describes concepts, roles, instances, and the relationships between them.

The Question Answering System (QAS) is used to extract the exact and precise answer to the user queries from the data repository. The system can be developed using different techniques like NLP with IR, reasoning with the NLP, web-based QAS, ontology-based QAS, and more. This particular question answering system is developed using the ontology model.

This ontology-driven question answering (QA) system provides the facility to the students for finding accurate answers to their queries. Teachers also can use this system to assess the knowledge and skill of students whenever they want. This QA system mainly focuses to apply in the education domain especially in ICT subject areas. It defines a set of questions and relevant answers which are organized in a structured way with the help of Natural Language Processing (NLP).

# Chapter 01: Introduction

## Content

- Chapter Overview
- Problem Domain
- Features of the Prototype
- The Requirements of Resources
- Thesis Document Structure
- Chapter Summary

## 1.1 Chapter Overview

This chapter aims to provide an insight into the project scope undertaken. The chapter begins with a brief introduction of the question answering system based on the domain-specific ontology, the importance of the question answering system, and the different types of QAS systems available. It also describes the limitations of the systems. This chapter also described the high-end features of the proposed system. Finally, the chapter overview is added in which it summarizes the chapter's contents.

## 1.2 Problem Domain

Normally users use search engines to find out the answers to their queries. The search engine is a web program that collects and organizes content according to the user's query. These search engines have billions of websites and web pages that help to answer the queries of different users. It identifies and adds websites to its database using three main techniques such as crawling, indexing, and serving search results. The most relevant and best search results will display as the list from its database. Then the user needs to click the link one by one to get accurate answers for his queries.

The Question Answering System (QA-System) finds out accurate and precise answers for the user queries from its data repository. This data repository can be stored in both non-structured and structured data [1]. The QA system can be developed using different methods such as heuristic approaches, machine learning, and ontologies. The researches that are based on the ontology methods are very popular in recent times since the resulting QA systems can benefit from knowledge modeling [2].

The QA system that is based on the ontology can use data in a "knowledge-based" repository that is developed by the ontology. An ontology can be demarcated as "a formal explicit description of concepts in a domain of discourse (classes). Properties of each concept describe various features and attributes of the concept (slots), and restrictions on slots (facets) ontologies together with a set of individual instances of classes constitutes a knowledge base" [1].

The domain can be categorized as closed (dependent) or open (independent). In addition to this classification, there are some other systems that consist of natural language interfaces, frameworks,

tools, and using combined techniques for information retrieval (IR). However, this method only returns the probable answer highlighted. Because this system uses “shallow” methods like keyword search, template, etc. [3]

The common vocabulary of basic concepts and relations between the concepts can be defined with ontology. This kind of knowledge representation provides more benefit to extract information about the domain. The question answering system can easily be implemented using the ontology as the ontology consists of structured data [4].

The QA systems can have some limitations, mainly to facilitate between the user's requirements for information in their own language and the available semantic data, and limitations relevant to machine translation. The necessity of the previous knowledge with the used ontology schema and the manual steps performed in the process are some of the other challenges. The query construction can be enriched using the ontology schema also [2].

The subject-related QA systems are available in few numbers due to some reasons. One of the reasons can be the unclarity of how the ontology-driven processing and reasoning map with the subject domain. The QA system is implemented in certain languages such as English, Chinese, and Arabic at a large scale and this situation remains because of the application challenges of the NLP. Techniques such as syntactic analysis and Natural Language Processing were used in few subject-related approaches of question answering models previously. However, those systems were not very efficient. Therefore, this study intends to expand its provision in this particular area to a certain extent.

### 1.3 Features of the prototype

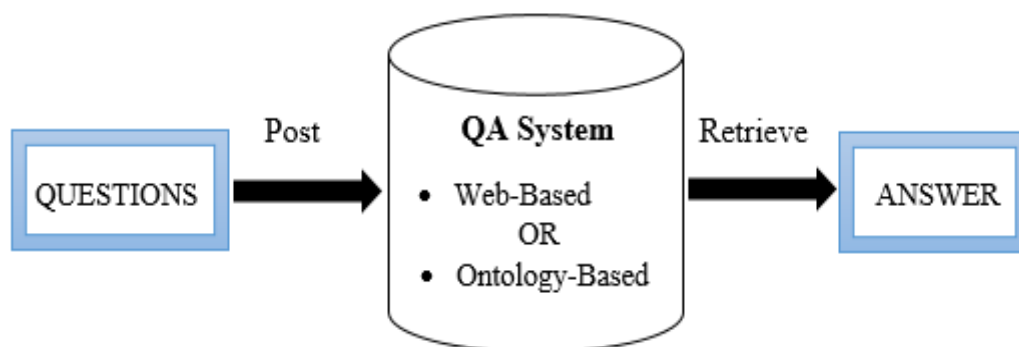


FIGURE 1.1: THE FEATURE OF QA MODEL

Basically, any “Question Answering System (QAS)” comprises mainly three components such as “question analysis, document analysis, and answer retrieval”. This particular proposed model also complies with that pipeline. The end-users enter the question in their native language using the user interface (UI) and get the precise answer for that query. The question will be answered using the resources like websites, corpus, and ontologies, etc. The diagram with the detailed information is given in the methodology section as well.

## **1.4 Resource Requirements**

The following resources are minimum essential to conduct this research work. However, the requirements are subject to change according to the circumstances.

### **1.4.1 Hardware Requirements**

The laptop or a PC with certain hardware configurations is essential for the system implementation, testing, and for the other documentation work also (e.g., Thesis preparation, Proposal preparation).

### **1.4.2 Software Requirements**

- **Protégé Software** (version to be determined) which is used to create the ontology knowledge base
- **Apache Jena Fuseki** is a standalone SPARQL server which is used to run the SPARQL queries to get the answer from the created ontology.
- **Python IDE** (version to be determined) e.g., PyCharm and different python packages to design user interface, data analysis, and deploy the system

### **1.4.3 Data Requirements**

- Grade 12, and Grade 13 ICT subject data are available in the National Institute of Education, Sri Lanka official website which can be accessed and used for this thesis work.
- A set of questions that can be used as the input of the system and for testing purposes.



## **1.5 Project Document Structure**

The final thesis will consist of the following chapters from chapter one to chapter six followed by the references and appendix part.

### **Chapter 2 - Research question**

This is one of the essential initial steps which helps to get insight into the project. It lays a solid basement for the entire project and directs the project in the right direction. This chapter also figures out the importance of this proposed QAS system.

### **Chapter 3 - Research Objectives**

The Research objectives chapter describes the main objectives of the proposed system. The objectives are categorized as the main objective and sub-objectives.

### **Chapter 4 - Literature review**

The Literature review chapter consists of detailed reviews of previous studies done by different authors in this specific domain. The recent research papers, conference papers, and other sources are utilized to draft this chapter to obtain a clear understanding of the subject. The chapter explains various approaches of question answering systems in detail and compares their uniqueness. This will also explain the techniques of quality measurement of the system from different perspectives.

### **Chapter 5 - Methodology**

This is the core section of the project, and it describes the methodology adopted to carry out the system successfully. The necessary steps taken to implement the system will discuss in detail under the methodology section. The rationale for selecting the appropriate technique for each section will also discuss in this chapter.

### **Chapter 6 – Results and Evaluations**

In this particular chapter, the results of both systems such as web-based and ontology-based QA systems will be discussed in detail. Various performance parameters like Accuracy, Precision, Recall, and F-Measure are compared with both systems with the help of a test dataset.

## **Chapter 7 – Timeline**

The important milestones and tasks will clearly denote in the timeline chapter. These factors need to be set out at the beginning for the successful planning and finishing of the project on time. The major deliverable of the project is organized in a chronological sequence in the timeline.

## **Chapter 8 – Discussion and Conclusion**

This chapter articulates the summary of both systems with the help of performance parameters. It will also highlight the future consideration of the study in the same domain. This will help future researchers to continue the work without any serious issues.

### **1.6 Chapter Summary**

This chapter provides complete insight into the problem domain, the prototype of the proposed system, its features, and the necessity of the question answering system. It also explains the project scope, how in-scope and out-scope engaged with the project. The resource requirements such as hardware, software, and data requirements also describe in this chapter.

The chapter concludes with the project document structure which describes how this document is organized and each part gives a brief introduction of its purpose and contents.

# Chapter 2: Research Questions

## Content

- Chapter Overview
- Project Aim
- Research Questions
- Chapter Summary

## **2.1 Chapter Overview**

The question answering systems which are related to a particular subject domain are available in few numbers due to few reasons. One of the reasons can be the unclarity of how the ontology-driven processing and reasoning map with the subject domain. Therefore, this chapter mainly focuses on the motive of this research work in the field of ICT-related subjects. This closed-domain subject-specific ontology QAS research will be focused on implementing a system to answer the user queries in the computer network module in the ICT subject precisely.

## **2.2 Project Aim**

The aim of this project is to design and develop a question answering system for the Computer Network module in the ICT subject domain with the help of ontology. In addition, the web-based QA system is also incorporated into the proposed model. Users can easily distinguish between these two systems by using them.

The proposed system needs to provide accurate answers to the user's questions quickly. One of the main aims of the project is to ensure that the answers provided by the system are identical to the answers the user is looking for.

## **2.3 Research Questions**

Research questions are a fundamental aspect of scientific research. The reason is that a research question provides a possible way to investigate a complex scientific problem. A good research question is essential to guide a research project. These research questions help to point out exactly what we are trying to find with this research, and to focus on the purpose of the research work.

These research questions can be very helpful in defining the scope of the research as well. It is not desirable that the number of research questions be too high or too low. In both these cases the research may face several complications in many ways . Therefore, it is essential that the number of research questions be such that the research can be completed within a specified period of time.

This particular QA system expects to find out the answers to the following research questions. Such as;

1. What are the real benefits of this proposed QAS system?
2. How the proposed QAS model helps IT students, to get the exact answers to their queries?
3. How it differs with the web-based QA system?

## **2.4 Chapter Summary**

This chapter clearly reveals the purpose of the ontology-based question answering system and describes the research questions of the project. These research questions help to achieve the goal of the overall system and it will facilitate the direction of the development methodology as well. The goal of this study can be achieved by getting clear answers to the research questions of this study.

# Chapter 3: Literature Review

## Content

- Chapter Overview
- Background
- Literature Review
- Chapter Summary

## **3.1 Chapter Overview**

This third chapter of the thesis explains several important terms such as “Natural Language Processing (NLP), Question Answering (QA) system’s components, ontology development, RDF, RDFS, and SPARQL queries” that are most frequently used throughout this project with some background. Also, the Literature review part focuses on the previous related studies done by different researchers which are relevant to the problem domain. These solutions and techniques could be re-used to develop the system in an efficient manner without compromising the accuracy and the quality of this particular research work.

## **3.2 Background**

### **3.2.1 Natural Language Processing - NLP**

Users are more convenient to use their own Natural Languages for communication. Computer user interfaces (UI) also become into the picture under these circumstances, such that users can easily communicate with the computers with their languages using the interface rather than using specialized computer commands. Modern NLP techniques facilitate the use of natural language in the programming ideas, thus the accessibility of non-professional users in programming has also increased [5].

The “Natural Language processing – NLP” is considered as a part of “Artificial Intelligent” and it’s main concern is the interaction of human language and the machine. There is a number of sub-topics that exist in NLP such as “tokenization”, “stop-words removal”, “stemming”, “lemmatization”, “part of speech - POS” analysis, semantic analysis, and parsing which are used to increase the machine understandability [6] [7].

The research [8] excerpt that the NLP mainly concerns two things. The first one is the communications between computers and human languages and secondly sharing the information.

The automatic or sometimes semi-automatic processing of human languages is known as Natural Language Processing, and it is basically multidisciplinary. The NLP is used in different branches of the computer science field such as formal language theory, compiler techniques, theorem proving, machine learning and human-computer interaction, and AI [9].

### 3.2.1.1 NLP Techniques for data cleaning

Various techniques can be employed to increase the machine understandability of NLP. Data cleaning is also one of the techniques used to enhance the understandability of the system. Some of the popular data cleaning techniques are discussed under this topic [7] [10];

- **Tokenization:** Each sentence is split into pieces (words) and each piece is called a token. A token can be issued to the sentences also. The Tokenizing process is also changing the question sentence to lowercase, delete unnecessary characters and symbols in the sentence.
- **Stop-word removal:** In this process, words that have less meaning, or no meaning are eliminated from the sentence. The deletion will take place when the words are matched with the stop-word dictionary.
- **Stemming:** Different forms of words are replaced with their basic root word by eliminating the affixes (e.g., prefix, suffix, etc.).
- **Lemmatization:** The challenges of the stemming process can be rectified in lemmatization by referring to the dictionary for the word meaning.

The stem words are stored in a table called index term dictionary. This table has only two columns for the term (i.e., keyword) and posting (i.e., file name).

The above text preprocessing can be depicted graphically as follows:

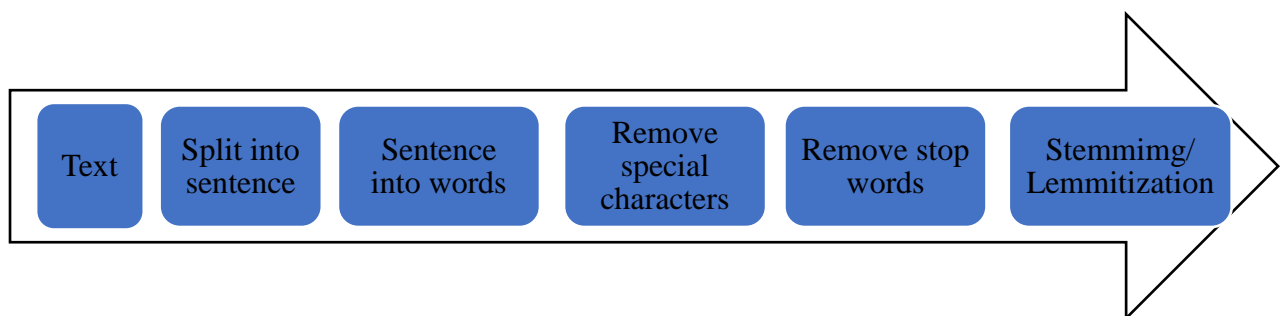


Figure 3.1: Text Pre-Processing Process



The NLP techniques such as POS tagging, stemming, lemmatizing, etc. are widely used in QA systems as these techniques help to analyze and convert natural language questions into machine-understandable queries [30].

### 3.2.2 Question Answering System – QAS

The field of Question answering becomes a popular research area nowadays for its practical applications. Because unlike other search engines the QAS returns accurate and precise answers to the user queries [11].

The typical architecture of question answering system can be depicted graphically as follows [12];

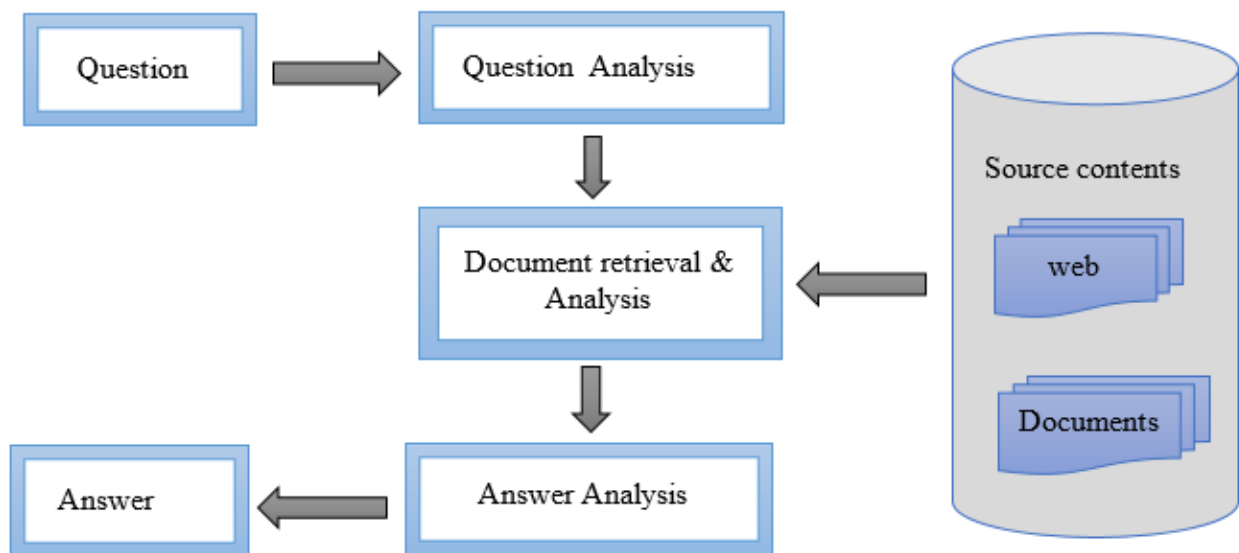


FIGURE 3.2: GENERAL ARCHITECTURE OF THE QA SYSTEM

Almost all QA system works in the same way. That means the system receives input as the question, which is written in their own language, then that problem is analyzed to get the answer, and finally, the system will display the exact answer for the query. Most of the QA systems are compliant with the same question answering pipeline as it has a minimum of three components itself such as question processing, document processing, and answer processing [8] [13].

The research [14] is implemented a QAS model that used three phases such as user query, query expansion technique, and entity linking method. However, the research [15] describes that question answering essentially consists of 4 modules such as question processing, document processing, paragraph extraction, and answer extraction.

The role of the first module of QAS is to convert the natural language question into a set of keywords. The relevant documents that are likely to contain answers are retrieved from the corpus in the second module. The paragraph extraction part uses algorithms that take a document as a question and find passages from a document that contains an answer. The last part of this QAS pipeline takes the returned passages as input and tries to retrieve an exact phrase to return as an answer.

### 3.2.2.1 Types of QAS

Several types of question answering systems exist in reality. This classification will be made based on many factors like the type of response expected, used technology, and so on.

Wellington Franco et al., denoted different types of question answering systems such as, “Closed-Domain Question Answering (QADR), QA for Comprehension Reading (QACR), Community Question Answering (QAC), QA Over Domain Ontologies, Ontology-Based QA (QASOBO), and Linked Open Data Question Answering (QALOD)” [16]. Each type of QAS has a unique feature and used different methods to implement.

The research on “A Survey of Text Question Answering Techniques” discusses the different question answering types. In this study, the researcher divide the QA systems into two groups such as QA system based on NLP and IR, and QA systems reasoning with NLP [8].

The researchers of [8] and [17] describe different kinds of “question answering systems” such as “Web-Based Question Answering Systems, IR / IE Based Question Answering Systems, Restricted Domain Question Answering systems, and Rule-Based Question Answering Systems”.

- **Web-Based Question Answering:** The web is one of the good sources to get information. This type of web-based QA system are using search engines that retrieve the web pages which likely consisting of answers to the questions. The “Web-based QA systems” vastly implemented for “open-domain” however, the domain-specific web-based systems also exist. But the numbers are less compared to the other one. The vast amount of information available on the internet makes it easy for these types of question answering systems to get quick answers to simple, factual questions with their help. This can actually be considered an advantage found in these types of question answering systems.

- **“IR / IE Based Question Answering Systems”:** The “IR (Information Retrieval) based QA systems” most probably retrieve the passages or topic in a collection from the source as the answer to the user questions. Whereas the Information Extraction (IE) systems are utilizing the natural language processing (NLP) systems to extract relevant information from the documents. This system can answer only “wh-type” of questions.
- **Restricted Domain Question Answering systems (RDQA):** Linguistic support is required for this system to understand the natural language text to answer the questions precisely. This kind of system is likely to have specific characteristics such as high system accuracy and minimum redundancy than the other types of QA systems. RDQA systems provide more accurate answers than the open-domain QA system as the availability of domain-dependent information.
- **“Rule-Based Question Answering Systems”:** The “rule-based QA system” can be considered as an extension of the “IR-based QA system”. The “Rule-Based QA systems” widely used different techniques of NLP to ensure the accuracy of the obtained answers and the systems do not use any specific techniques like “deep language understanding”.

The QA systems are classified based upon the available resource for answers. The closed domain QA systems provide more accurate and exact answers compared to the open domain QA systems. The answers by the system to the queries have an impact on the type of questions. This particular research categorizes the question types into six such as (1) Factoid type questions, (2) List type questions, (3) Confirmation Questions, (4) Causal Questions, (5) Hypothetical Questions, (6) Complex questions [18].

### 3.2.2.2 Performance Evaluation of QAS

Subjective impressions are not enough to make decisions about the quality of the system. Rather, it is necessary to use some quantifiable metrics with a labeled evaluation dataset to evaluate the system reliably. Some datasets provide one answer to one question and others provide multiple options for one question. However, the QA system being developed here is of the first type.

The factors such as “accuracy, precision, recall, and F-measure” are used to compare the performance of different QA systems [12] [19] [20]. It is desirable to use the same number and the same questions in order to reasonably evaluate both QA systems.

“Accuracy is the ratio between the number of correctly classified questions and the total number of sample questions”.

$$\text{Accuracy} = \frac{\text{No of correctly classified Questions}}{\text{Total no. of testing samples}}$$

The value of dividing the sum of accurate answers by the sum of related answers gives the precision. The below equation can be used to compute the precision.

$$\text{Precision} = \frac{\text{No. of Correct Answers}}{\text{Total No. of Related Answers}}$$

The value of recall can be computed using the below equation. It explains how generally you predicted efficaciously.

$$\text{Recall} = \frac{\text{No. of Related Record Retrieved}}{\text{Total No. of Questions}}$$

F-measure is sometimes referred to as “F1-score or F-score” and using this parameter, the accuracy of the test can be measured. The formula is given below.

$$\text{F-measure} = \frac{2 * \text{harmonic mean. Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

By evaluating the four factors described above in the same situation with the two QA systems, it will be much easier for users to know the best system. The users can get the more benefits from that particular QA system by knowing this fact easily.

### **3.2.3 Ontology**

Ontology can be defined as the knowledge base of the specific domain where the details are structured properly. It gives a general understanding to the user agents. Defining classes is the key step of implementing ontology. The superclass-subclass relation of the ontology can be used to extract the exact information. Ontologies can be constructed using the most popular tool called “Protégé”. It is a knowledge requisition system that is freely available. It is written in Java language. Complex user interfaces can be created using swings. Classes can be created with various instances with their object properties and data properties [21].

The collection of terminological or assertional axioms is the ontology. The relationships between multiple concepts can be described in terminological axioms and the relationships between the individuals and concepts can be expressed using the assertional axioms [33].

Ontology softwares can have different features like classes (concepts), properties (Object and Data properties), individuals, data types, annotation properties, and so on. The metadata of the entities, and the relationships can be added in the annotation section. Also, the author details, the data of the modifications, and the description of the entities can be included there. Annotation properties are different from other properties as annotation properties do not interfere with the reasoning. When defined in classes the annotations are not inherited by subclasses [22].

Ontologies can have two main purposes such as automatic reasoning and reuse of the knowledge. Automatic reasoning become feasible since the set of concepts, relations, and their properties can be described in a formal way whereas all ontologies share the same namespace and can be linked together which is leading to the semantic web. That can be reuse whenever required.

Owlready is a python package which can be used to access the ontology files in python language. Owlready is a “ontology-oriented programming” in python, and it includes a graph database with an OWL semantic level. Currently the Owlready supports to limited number of file formats such as RDF/XML, OWL/XML, and N-Triples.

The ontology for any subject domain can also be created easily in a programming language using the Owlready package. The Ontologist has the facility to add some constraints in an OWL file like add constraints in the Protégé editor. Also, the Owlready is able to blend the object model with the ontology in a single model [23].

### **3.2.3.1 Ontology Development**

Reference [1] identified few reasons to form an ontology by the developers. The following are some of them.

- The general understandings of information can be shared with other individuals or mediators
- Existing domain knowledge can be reused
- The domain assumptions can be made precisely
- The domain knowledge and the operational knowledge can be isolated
- The knowledge of the domain can be analyzed meaningfully

However, ontology building cannot be easily done as it takes a lot of time, effort, and expert people in that particular domain. The ontology building process should be subjected to a number of paces in practice which will be discussed in the methodology section in detail.

### **3.2.3.2 Protégé**

Protégé is the prominent editor for ontology, and it has different versions (e.g., “ protégé 2000, Protégé 3.1, protégé 3.4 beta, protégé 4.0 beta, Protégé 4.3”, etc.). It is an open-source tool for editing and managing Ontologies and is a widely used domain-independent, freely available, platform-independent technology for developing and managing ontologies. In addition to this, there are some other tools for ontology editing such as DOME, Onto Lingua, and Altova Semantic Works [21].

Table 3.1: The difference between RDBMS and ontology

Characteristics	Relational database	Knowledge base
Structure	Schema	Ontology statements
Data	Rows	Instance statements
Language	DDL (Data Definition Language)	Ontology statements
Query language	SQL	SPARQL
Relationship	Foreign key	Multidimensional
Logic	Triggers	Formal logic statement
Uniqueness	Keys of tables	URI

Research [20] also highlights the differences between the relational database management system (RDBMS) and ontology (“Knowledge Base”) in a tabular way that can give a clear brief idea of these differences.

Protégé is an “open source OWL editor” tool that is developed by Stanford University and the built-in plug-in is used to develop, assign restrictions, and reason the proposed ontology. It can run on a variety of platforms, supports personalized user-interface extensions, merge the open knowledge base connectivity (OKBC) knowledge, model. It works easily as a relational database, XML, RDF, and is used by individuals and various research groups [25].

### 3.2.4 RDF, RDFS, and OWL

- **RDF**

The Resource Description Framework (RDF) is a data model, and entities or resources can be described using this RDF as it provides a unified standard. In simple terms, it can also describe as a way and means to express things. RDF is represented as SPO (Subject-Predicate-Object) three tuple, sometimes called a sentence (statement). The following image describes the structure of the RDF data model.

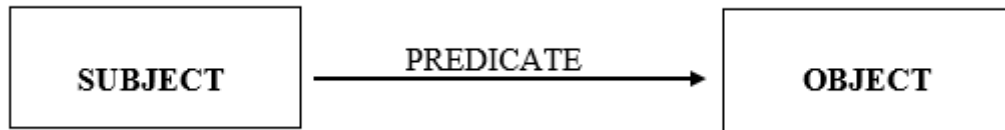


FIGURE 3.3: THE STRUCTURE OF RDF

Resource information are described in an easy-to-understand graphic format. This particular framework is known as the Resource Description Framework (RDF). RDF refers to the metadata of WWW resources. So that, it was identified sources with Uniform Resource Identifier (URI). XML language is used to write the RDF documents, and this is called RDF/XML. It makes RDF information easy to exchange between different types of computers in different operating systems and application languages. The RDF information is characterized in the form of triples such that subject-predicate-object form. All the elements of the triple are resources except the last one (e.g., object). The last element can be literal also. “Literal in the RDF sense is a constant string value such as string or number” [24].

The data model for objects (i.e., resources) and relations between them can be described using RDF, and it provides a simple semantics for this data model. These data models can be represented in an XML syntax [25]

#### ▪ **RDFS**

The semantic extension of RDF is known as the RDF Schema (RDFS). The Group of related resources and relationships of these resources can be described using the RDFS mechanism. The “`rdfs: Class`, `rdfs: Resource`, and `rdfs: Property`” resources are used to declare classes, resources, and properties in RDFS [22].

Some systems attempt to utilize semantically annotated web pages to answer the queries. RDF or RDFS framework can be used to write these annotations and it uses the semantic annotations to perform the deduction and reduce the number of possible answers to a question [26].

RDFS is a vocabulary that describes the properties and classes of RDF resources, with semantics for generalization hierarchies of such properties and classes [25].



- **OWL**

The OWL stands for “Web Ontology Language”. It is used to develop different ontologies that are compatible with the WWW. Various applications can use this OWL to process the contents of the information. This is more effective than simply providing information to users or humans. The OWL provides additional vocabulary with semantics details. This is very useful for facilitating machine interpretation of web content than that supported by “XML, RDF, and RDFS”. The “OWL Lite, OWL DL, and OWL Full” can be considered as different flavours of OWL [25].

The OWL is an extension of RDF, and RDFS. The OWL is a language that is based on RDF/XML syntax. It includes the definition of classes, properties, and their instances [11] [27].

These three data models have different characteristics. They can each be used for different occasions depending on their nature. The comparison between these three models can be tabulated as follows [27];

TABLE 3.2: THE COMPARISON OF RDF, RDFS, AND OWL

<b>Properties</b>	<b>RDF</b>	<b>RDFS</b>	<b>OWL</b>
Class	Yes	Yes	Yes
Data	Yes	Yes	Yes
Object	Yes	Yes	Yes
Domain and Range	Yes	Yes	Yes
Annotation	No	Yes	Yes
Individual	Yes	Yes	Yes
Graph	RDF	RDFS	Onto
Inverse	No	Yes	Yes
Inference	Poor	Good	Better
Indexing	Poor	Good	Better
Searching	Poor	Good	Better
SPARQL Query	No	Yes	Yes
DL Query	No	Yes	Yes
Prefix	Rdf	Rdfs	Owl

### 3.2.5 SPARQL

SPARQL is pronounced as “sparkle”. It is a query language like SQL of relational databases, but it is specific to the RDF databases, which is used to search in RDF graph. It contains both the “SPARQL Protocol and RDF query language”. The stored data in the form of RDF triples (e.g., S-V-O triples) can be retrieved and manipulated using the SPARQL semantic query language. Resources are represented with URIs which can be abbreviated as PREFIX names. The standard of RDF vocabulary is the “FOAF (Friend-Of-A-Friend)” and that helps to describe the people, their deeds, and their relationships with other people and objects. Another RDF version is DBpedia which is extracted information from Wikipedia. These datasets can be used for SPARQL querying. The accuracy of the system is improved by using the DBpedia ontology for extracting some sort of information about the domain [28].

Automatic SPARQL generation is very useful as manual SPARQL query creation is a more time-consuming task. The Quepy, a python framework is used to create auto SPARQL queries for the research. That converts natural language input queries into SPARQL queries. The Quepy is a python framework that provides the basic structure of SPARQL to convert questions of everyday language to questions of a database query language. It can be easily tailored to a variety of questions in natural language and database queries [28].

The research [29] has proposed tools to transform users’ questions into SPARQL query language to find the answer. They created an application called Quepy. This application converts the natural language question into SPARQL in order to query linked open data such as DBpedia or Freebase.

The SPARQL query plugins can be installed in the protégé software to extract the relevant data. Otherwise, the ontology can be uploaded into a separate SPARQL server like “Apache Jena Fuseki” and retrieve the answer by writing SPARQL query in that server. The tested queries and the obtained answers are tabulated and annexed as Appendix B.

A SPARQL query comprises mainly two segments or clauses such as “SELECT” and “WHERE”. Each clause can be used for different purposes. The “SELECT” clause recognizes all the variables that are to be seen in the output of the query. The basic graph pattern against the original data graph can be obtained using the “WHERE” clause. This WHERE clause can have many triple patterns like “<s, p, o>”. Each letter in the triple pattern describes the unique thing, i.e., the character “s” represents the “subject”, the character “p” indicates the “predicate”, and the character “o” implies

the “object”. The subject, predicate, and the object can be variables, resources (“written as URIs”) or literal values in the SPARQL query.

### 3.3 Literature Review

The Question and Answering system has evolved since the days when the “Semantic Web” became popular since the QA system development is a very common application of the semantic web. Semantic provides the meaning of a word in a sentence and semantic analysis is trying to expose the meaning of words and sentences in the real-world environment [30].

This section briefly discusses some question answering systems that are previously implemented and their techniques.

The research [31] has implemented a system that answers natural language questions using multiple ontologies. They combined several knowledge bases using a natural language parser to transform questions into SPARQL queries or another query language. This semantic ontology QA system can return accurate results. Users can choose one knowledge base out of three knowledge bases such as Freebase, DBpedia, and Linked Open Data. The main purpose of this system which uses three knowledge bases is to maximize the precision and accuracy of the results.

**QUASE** – It stands ‘Q**U**estion A**N**swering S**Y**stem for E**D**ucation’ which is a question answering system developed in a closed domain with a finite set of documents for answering natural language questions. This model incorporates several modules such as machine learning, NLP techniques, indexing-based retrieval mechanism, and ontology processing. The system mainly focuses on factoid questions to find different types of answers including human, location, numeric, abbreviation, and entity, and the question classification has been done using the question taxonomy. This QUASE system provides users two options for getting answers such as a correct answer or a nearly correct answer[32].

**ADANS** - This is an ontology-specific (i.e., Agriculture Domain) semantic question answering system that uses multiple techniques such as NLP, and semantic web technologies to answer the natural language questions of the users. The question is converted into a SPARQL query which is the loosening part of the system. Then, it helps to retrieve the answers within the acceptable time. This QA system is very convenient (i.e., Portable) and can be used for other domains also with small configuration changes [33].

**AQUA** - This acronym stands for “Automated Question Answering” which is a question answering system developed using several technologies such as hand-crafted ontology, Logic, and NLP. This AQUA model implements a framework that combines logic queries and information retrieval. The main aim of this system is to find correct textual answers to the user’s question quickly [26].

Maria Rahim et al proposed a semantic-based QA system on travel ontology. This system was implemented to answer the factoid questions from the tourism industry. The natural language question is converted into NL triples after detecting the type of question and then generate a SPARQL query based on that natural language triples. Then these SPARQL queries are answered using the ontology [34].

**Ontology-based MCQ** - The research [35] has evaluated the feasibility of ontology-based MCQ models which are developed by inexperienced educators. Their verdicts confirm the feasibility of developing that kind of ontology-based MCQ system and it can generate educationally useful questions and answers based on the level of difficulties.

**Python-QAS:** Marcos Oliveira Ramos et al implemented a web-based domain-specific question answering system for “learning python” programming language. The website was developed using Django and PostgreSQL which is a free and open-source relational database that was used as the database engine [36].

**HERO** ontology – The acronym HERO stands for “Higher Education Reference Ontology”. This HERO ontology has been implemented to overcome the issues of application ontologies. Because most of the application ontologies are too specific and cannot be reusable. The reference ontology of the higher education sector would be an important tool for various stakeholders who have a similar interest in higher education [37].

Authors A.S. Omarbekova et al., [38] have developed a QA system based on the knowledge base. Algebra is the knowledge base that is developed using the protégé editor software. This application-based system is used in smart learning such that students’ knowledge on a given subject area can be checked using this system.

The research [39] has developed a IT domain specific Question Answering system which has used 16 types of questions and an inferencing method is designed and implemented for each of the question types. The users and QA system and the ontology were connected via the user interface.

This QA system has exploited a triplet method for ontology lookup instead of natural language sentences for internal processing.

**AquaLog** – It is a portable ontology-based QA system capable to use one or more knowledge bases to answer user questions. This QA system utilized several techniques such as the “GATE NLP platform, string metric algorithms, WordNet and ontology-based relation similarity service (RSS)” to understand the user queries. In this system, the natural language queries were converted into the query triple format and “Relation Similarity Service (RSS)”. The RSS can convert the query triple format into “Onto-Triple” form. This QA model is based on the triple form such that “<Subject, Predicate, Object>” type. The Precision, Recall, and also failure types were used to evaluate the performance of the AquaLog QA system separately. This domain specific question answering system was able to attain the average of 63.5 % [40].

**SMART** – The acronym stands for “Semantic web information Management with Automated Reasoning Tool”. It is a web-based and ontology-driven QA system, which is easy to use for the semantic web, and the purpose of this system is to provide tools to life Scientists to represent, integrate, manage and query heterogeneous and distributed biological knowledge. The SMART system used Java Programming Language with open-source and semantic web including OWL, API, and Protégé for the implementation [41].

**PowerAqua** – It is a QA system that uses multiple ontologies to answer the natural language questions from the users. Hence, this system is not limited to a single ontology like other QA systems. PowerAqua comprises three major elements such as “Linguistic Component” which help to analyze the natural language query and translates it into linguistic triples, “Power Map” which identifies the type of ontology from the database, and “Triple Similarity Services” which helps to analyze the ontologies and generate the appropriate results [42].

**QASYO** – This acronym is represented to “Question Answering System for YAGO Ontology”. It is used the “YAGO ontology” which was constructed using the “oracle database and visual C#” as the knowledge base for answering NL questions from the end-users. However, this system is considered an old system as it uses keyword search methods.

QASYO system architecture comprises four segments such as “question classifier, linguistic component, query generator, and query processor”. The linguistic component creates a parse tree that divides the sentence based on grammatical forms like question words, nouns, objects, verbs. Using the Query Simplification Technique, the system simplifies NL questions into simple queries.

The “WordNet 2.1” was used as the knowledge base of this system. It was found that 91% of the total questions could be answered, of which 40% were based on knowledge [43].

Research [44] proposed an “ontology-based QA system for banking domain” which uses the template to generate answers. The answers to Frequently Asked Questions (FAQs) are stored in the ontology and it is considered as the knowledge base of this system. This research was used several techniques such as regular expression pattern creation, and linguistic structure to develop a chat application of digital assistant for the banking domain to answer user queries. The system is used the keyword searching method to execute the ontology. This system had limited data set as it used FAQ as the knowledge base. The system was able to achieve 75% of the overall accuracy level for twenty test data.

The research [45] created an ontology in the music domain for the retrieval of semantic information. The ontology was constructed using the protégé 5.0 software and this ontology was tested using the Descriptive Logic ontology query language.

The coverage of the examination paper is determined using ontology. The research [46] described this study where ontologies are developed to the relevant subjects to calculate the depth. This automated system would help to keep the standard and quality of the examination papers. The examination papers were evaluated using the path of the words which is already saved in the ontology. Also, the feedback of the subject experts was obtained for the evaluation purpose and the accuracy of the ontology-based method is compatible with their values.

The research [47] is proposed an ontology-based system for durian pests and diseases retrieval. The system consists of two main information, such as the knowledge of durian pests and diseases and diagnosis of durian diseases, and the suggestions for the treatments. The empirical results yielded 100% of precision, 88.33% of recall, and 93.8% of F-measure.

**IQAS** - This acronym stands “Intelligent Question Answering System” which is used in the concept of ontology and semantic search engines. The system searches keyword-related information on the web and the results are saved in the database for the easy extraction of the users. The IQAS system has two types of users such as students and teachers. Students get the service from this system as they raise questions and received the answers, and the teachers stored limited questions and answers into the knowledge base at the beginning. The system provides more accurate results [48].

The research [49] implemented a web-based ontology for information searching. This system use keywords to search and retrieve data from the ontology. The keywords can be entered on the website which is developed by adding a semantic application to provide more precise and most relevant information about the IT profession field. The ontology of the national occupational map of ICT Mapping facilitates the development of structured data relations schemes by grouping concepts, characterizing relations, and defining system boundaries on concepts within the National Occupational Map of ICT.

Sheen-Mok Lee et al proposed the ontology-based question answering system for the IT domain. They have been used sixteen types of questions and an “inferencing method” is designed and developed for each type of question. Instead of a complete sentence, a triplet is used in this method to the ontology lookup. First, the “question-type lexical triplet” is given as the input and then it is converted into a “question-type ontological triplet”. The answers are also received in triplet format (“answer-type ontological triplet”) and it is converted into the “answer-type lexical triplet” output [50].

The authors W. Ahmed and B. Anto have proposed a question answering system called “An automatic web-based Question Answering (QA) system for e-learning and education” in 2017. They have implemented a method to identify the question type to extract the precise answer. The efficiency of this automated web-based QA system for e-learning is enhanced as the system yields only the exact answer rather than full documents. This web-based QA system helps school children and learners who need to get information from the web for their homework and assignments in any subject [51].

### **3.4 Chapter Summary**

This chapter initially discussed some predominant background topics which are very frequently come across throughout any question answering system. Better understandings of those terms definitely will be very helpful to construct the QA system without any ambiguity. Furthermore, this chapter has highlighted some previous QA models which are developed by different people, their techniques, and different usages briefly. These studies are helpful to find out the existing research methodologies and to identify the gaps. The current research work can be easily carried out by learning and choosing these studies correctly.

# Chapter 4: Research Objectives

## Content

- Chapter Overview
- Main Objective
- Sub objectives
- Chapter Summary



## **4.1 Chapter Overview**

This chapter describes the research objectives of the project work. Research objectives should be set at the beginning of any project to avoid ambiguity of project goals. The research questions should be clearly answered at the end of the project by the implemented system. Besides, it gives direction to the project and is very helpful to define the project scope as well. The research scope cannot be too broad or too narrow. When the scope is too large then it will consume more time to complete and carries a high probability to deviate from the project path. If the scope is too small, then it is really hard to get adequate information from that. Therefore, the scope of the study should be appropriate in size, otherwise, the research will become less useful. Hence, setting the right research objectives is a challenging task itself.

## **4.2 Main objective**

The main objective of this research is to develop a question answering system (QAS) in the field of education, especially for information technology-related subjects. In this research work, computer network an ICT module is considered. The system should provide an accurate and precise answer to the end user's questions in their own language in this particular domain within short time. The ontology and semantic concepts are used to build this system.

## **4.3 Sub objectives**

Achieving the main objective completely at one time can be difficult in many cases. Therefore, these sub-objectives are used extensively to achieve the main objective easily. This proposed system development can have different sub-objectives, such as:

- Study the existing question answering systems approaches which are based on the ontology to determine the requirements of Computer Network Module domain .
- Build a domain dependent ontology for the computer network module of ICT subject which is the basic of the QA system.

- To Implement the QA-system prototype with the help of ontology and NLP
- Comparing the proposed ontology-based question answering system with the web-based QA system.
- Evaluating the system with sample dataset.

These sub-objectives can be achieved gradually throughout the project. Also, these objectives will be facilitated to achieve the main objective of the project easily.

#### **4.4 Chapter summary**

This chapter focuses on achieving project goals by dividing the main objective of the research into sub- objectives. The main objective of the project can be achieved successfully by methodically implementing the sub-objectives of the project.

# Chapter 5: Research Methodology

## Content

- Chapter Overview
- The phases of the QA system
- The Data Processing
- The Query Processing
- Map the ontology (“Ontology Mapper”)
- Retrieve the Answer
- Chapter Summary

## 5.1 Chapter Overview

The design and methodology of this proposed question answering system is articulated in this chapter. The QA system developed here is based on ontology. The ontology is created for an ICT subject module known as Computer Network. This can be considered as the domain of knowledge. In this approach, the natural language questions need to be converted into SPARQL queries and this is also one of the main concerns in the system.

The users can ask questions in their natural language using the user interface and the system need to provide the accurate and concise answers to that queries. A number of stages can be involve in that answering process such as question analysis and answer extraction from the specific domain (e.g., computer network) Ontology. Making it “ontology-portable” is considered to be the most important characteristic of the system. The process of creating any question and answer system using ontology is a process consisting of various components such as “Data Processing, Question Processing, Ontology Mapper, and Answer Retrieval”. Each of these elements is interpreted individually and precisely in separate sections. These chapters also explain the exact reason for choosing the appropriate method for each category.

## 5.2 The phases of the QA System

Question and answer systems are one of the most important applications of information retrieval processes. Retrieving information from a particular knowledge base is more specific than retrieving information from a set of documents. Therefore, the ontology concept is considered to be the most appropriate application for information retrieval tasks. Basically, any question answering system comprises mainly four components as mentioned above, and this particular QA model also complies with that pipeline. The proposed system aims to generate specific answers in natural language to queries asked in natural language. SPARQL query generation for a natural language question is an important part of generating the natural language answers. SPARQL endpoints can be used to generate the answers for SPARQL queries. The following diagram clearly depicts these stages of the proposed question answering system. Each phase of the model and its techniques explain in detail in this chapter.

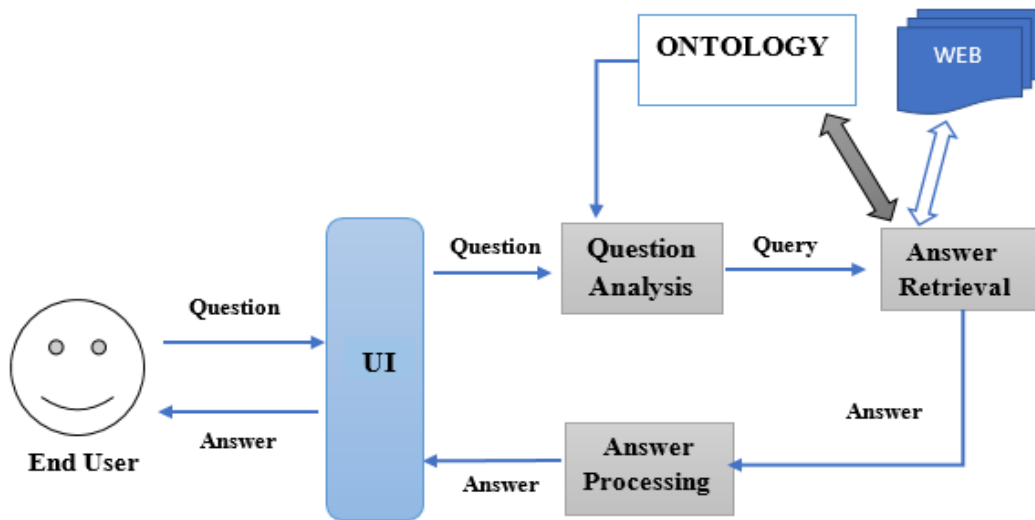


FIGURE 5.1: THE PROPOSED QA SYSTEM ARCHITECTURE

### 5.3 Data Processing

Data processing is liable to maintain and manage the “Knowledge Base – KB” which can have different ontologies. The ontology can contain the “schema” and it is mostly stored in a file. Building ontology is the main part of this stage. A “top-down approach” can be used to construct the ontology. In this method, first the abstract concepts are identified. After that only the specific concepts are focused. An ontology for computer network module is constructed manually as the domain which can signifies the basic knowledge base for our system. This ontology can be used to discuss the proposed approach to transform questions in natural language into “SPARQL queries” and to express model outcomes throughout this thesis.

Various tools such as “Hozo, DOML, and AltovaSemantic Works, etc.,” are available for developing ontologies. However, in this project the Protégé software which is the most popular and widely used ontology development editors has been used to construct the ontology. It can be easily defined several things using this software such as: “ontology concepts (classes), properties, taxonomies, various restrictions, and class instances”. It supports the use of many ontology representative languages such as OWL, RDF, RDFS.

This project has been developed using the software called protege the ontology content for the computer network module. This computer network is a unit of ICT course in grade 12. This

ontology is structured using data from this unit. The ontology is made in OWL format so it can be easily mapped to a database.

The ontology building process should be subjected to the following paces in practice [1] [18] [52].

1. **Decide the domain and the ontology scope** – The process of ontology development begins with deciding the domain and identifying its scope. The scope of the design model can be limited by answering few queries that are listed below.
  - The domain and the scope of the ontology – The domain of this ontology is a module (Computer Network) in ICT subject of Grade 12 curriculum
  - The use the ontology - The ontology is to provide a schema-base of Computer Network module which is used in our QA system
  - Type of questions expected to answer - The ontology would provide answers to questions relating to the Computer Network domain and like:
    - I. What is the Computer Network?
    - II. What is the purpose of a computer network?
    - III. What are the types of networks?
    - IV. How to define the reference model?
    - V. What is the topology?
    - VI. Why do we need an IP address?
  - Who will use and maintain the ontology? - The ontology will be available to the QA system to provide basic information for questions in Computer Network Module subject domain.
2. **Overview of the ontology** – Here, the ontology is designed from the scratch as a prototype model to serve in the implementation of the system. The reason for this development is the absence of the existing ontology in the same domain. Therefore, identify the relevant topics and data in this domain that are needed to the process of answering questions is very essential. This ontology was represented in OWL.

This created ontology contains general characteristics. It can add or remove any information about the networking module easily and without affecting the overall structure of the ontology whether at the level of classes or properties. So, it can be reused by other applications interested in the same domain.

The overall ontology details of this domain can be tabulated as below:

TABLE 5.1: SUMMARY OF THE ONTOLOGY METRICS

“Axioms”	513
“Logical axiom count”	330
“Declaration axiom count”	171
“Class count”	54
“Object property count”	25
“Data property count”	46
“Individual count”	47
DL expressivity	ALC(D)

At this stage, we must list out all the essential terms that are used in the ontology is listed. These terms are collected to form a comprehensive list that does not bother about the overlap of the concepts, relation among the terms, and properties of the concepts.

3. **Demarcate classes and hierarchy** – At this stage, classes (concepts) that are used in the ontology can be defined. These classes are not selected randomly, but they are selected depending on the computer network module domain. Figure 6 contains seven Ontology concepts mentioned in our domain. Choosing these concepts has a direct relation with user requirements used in the process of answering questions in the selected domain.

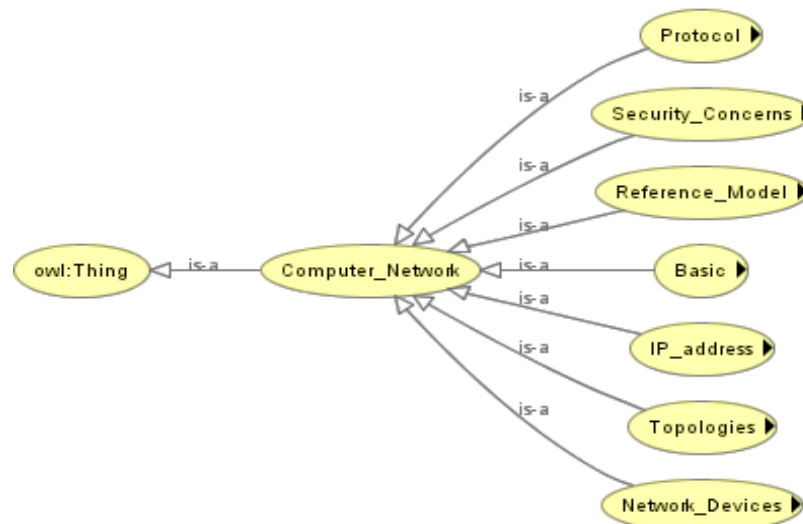


FIGURE 5.2: ONTOLOGY CLASSES IN PROTÉGÉ

Each of these classes consists of sub-classes and some sub-classes consist of sub-subclass as well. This complete comprehensive class diagram can be derived using the OWLViz from the protégé software. That complete diagram can be depicted as follows:



FIGURE 5.3: THE COMPLETE CLASS DIAGRAM IN ONTOLOGY



4. **Demarcate classes-slots properties** - The competency questions from Step-1 cannot be answered using the classes alone because it does not provide adequate information to answer such queries. The internal structure of the concepts needs to be described prior to define the classes. It is essential to decide the class in the list that describes by the properties.

The properties can be object properties and data properties. The object property is used to connect with other classes or instances and the data property is used to create a relation to classes with literal. In this particular ontology, we have used 25 object properties and 46 data properties. Both properties need to define the domain and the range of the classes. All data properties consist of the range as a string and have a different domain.

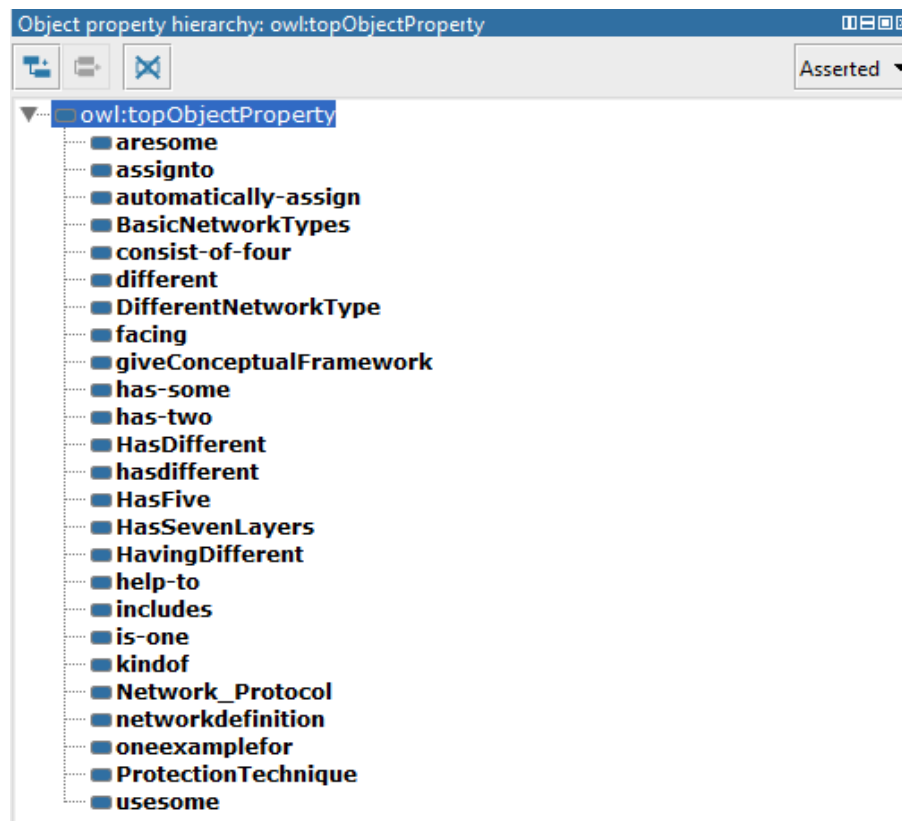


FIGURE 5.4: OBJECT PROPERTIES OF THE ONTOLOGY

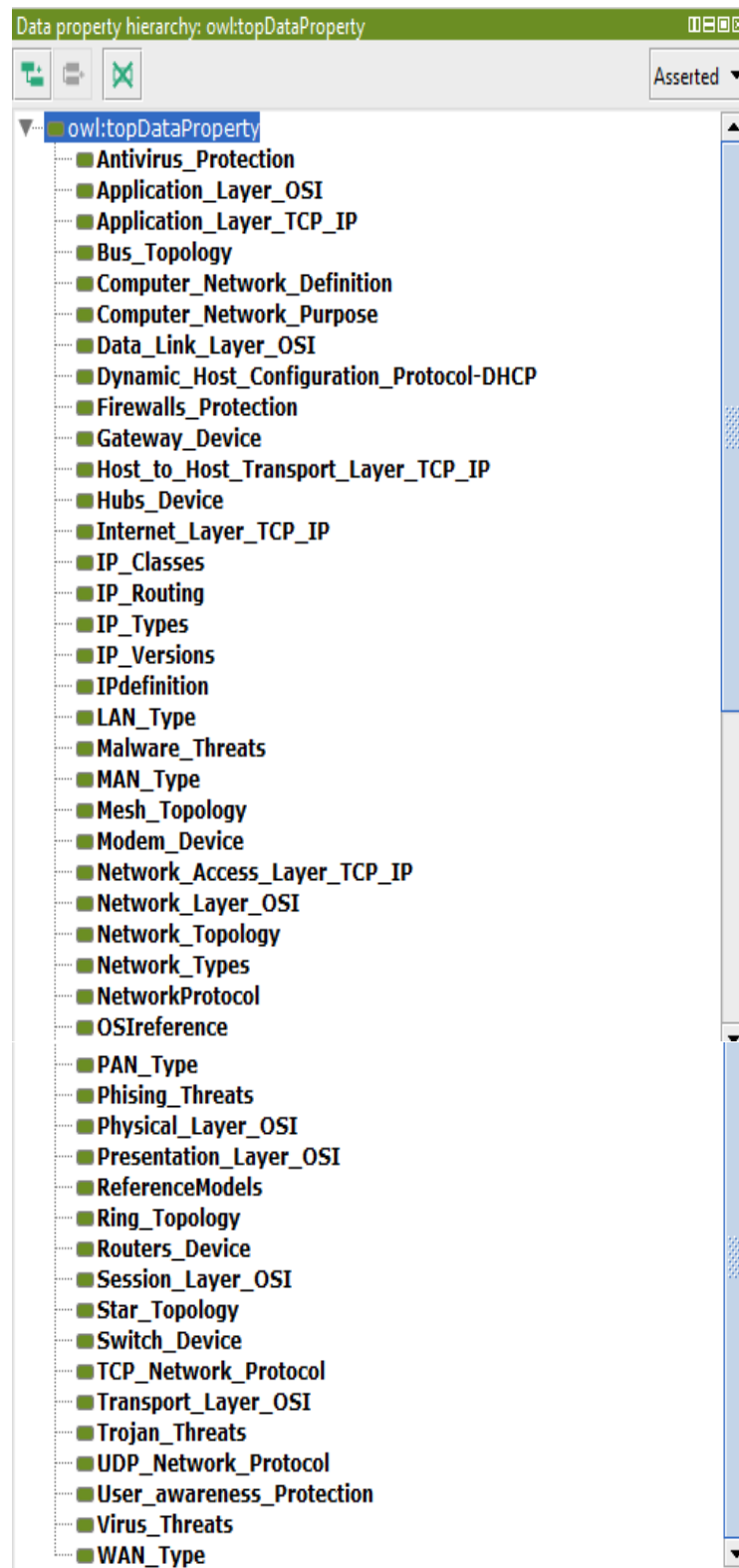


FIGURE 5.5: DATA PROPERTIES OF THE ONTOLOGY

5. **“Define the facets of the slots”** – Here, “the facets are added to the properties and these facets include value type, allowed values, the number of the values (cardinality), and other

features of the values the slot can take”. In this case, all the slot values are strings. For example, the value type of data property Malware\_Threats for domain Threats class is a string and the number of the values (cardinality) has a minimum cardinality of 1.

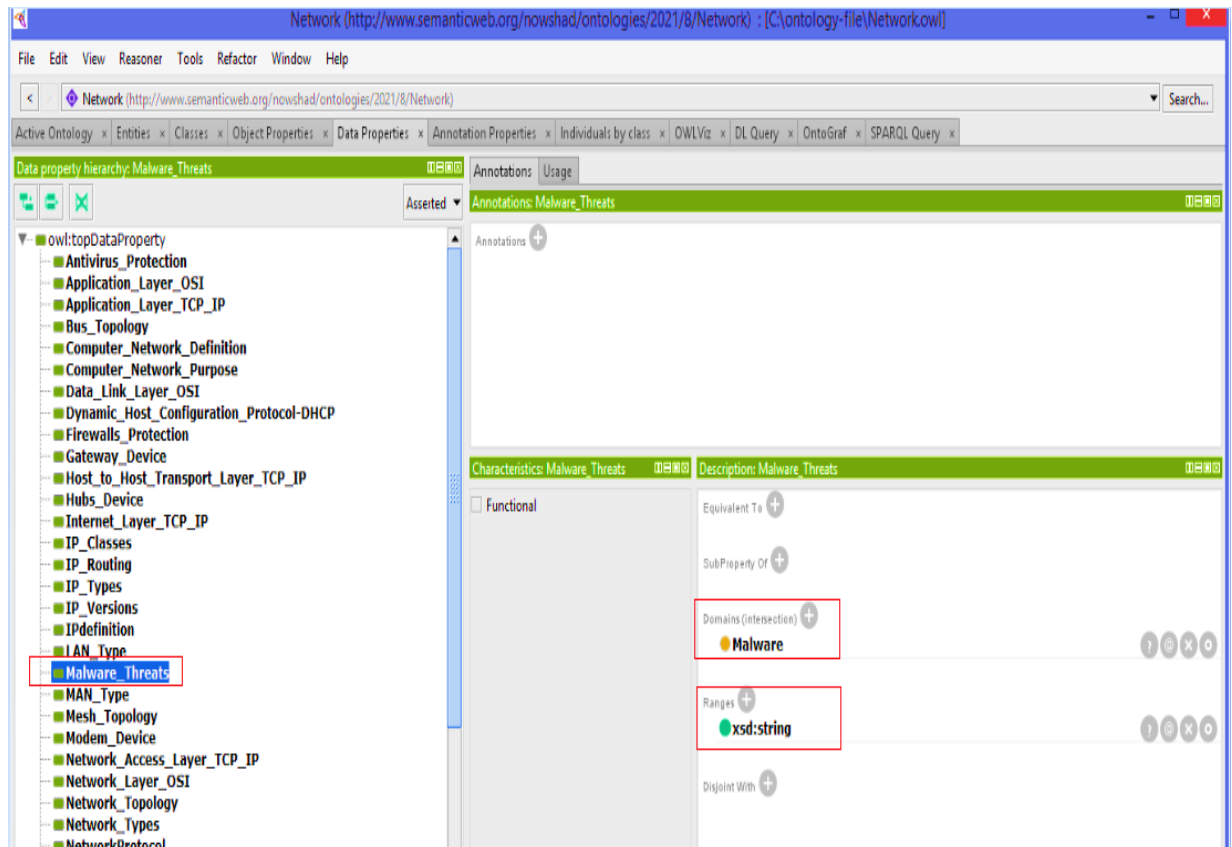


FIGURE 5.6: DATA RESTRICTION

6. **“Creating instances”** – At this pace we used to create class instances in the hierarchy. The factors such as “choosing a class, creating an instance of that class, and filling in the slot values” are required to define an instance. Creating instances (individuals) of classes in the hierarchy can be considered as the last step of building ontology. In this particular ontology, we have 47 individual counts.
  
7. **“Apply Ontology Reasoner”** - After creating instances, and storing them, we apply an ontology reasoner (e.g., HermiT reasoner) on the ontology. This is necessary to detect the consistency of the created ontology. It can also recognize “new relations from existing ones. The reasoner is able to identify the different types of ontological relations such as transitive, symmetric, inverse, and functional properties and use them to add new facts. Note that we are used the reasoner only to explore new relations and axioms for ontology

and not used it in the analysis of the question”. Finally, the ontology in RDF or OWL data format needs to be exported into the required interface for the execution of queries after saving the ontology in RDF/OWL format.

This ontology information can be viewed in the SPARQL server called Jena Fuseki either in a table format or in raw response. First, the developed Network ontology need to be uploaded into this server, then that ontology take as the dataset to produce the result. The result consist of subject, predicate, and object as three column. The following SPARQL query is used to obtained the results.

```
SELECT * WHERE {  
  ?subject ?predicate ?object .  
  ?predicate a owl:ObjectProperty .  
}
```

The following figure shows the sample result of above query.

QUERY RESULTS

Table Raw Response

Showing 1 to 46 of 46 entries

Search:  Show 50 entries

	subject	predicate	object
1	table:GatewayDevice	table:are	table:GatewayDevice
2	table:HubsDevices	table:are	table:HubsDevices
3	table:ModemDevice	table:are	table:ModemDevice
4	table:RoutersDevice	table:are	table:RoutersDevice
5	table:SwitchDevice	table:are	table:SwitchDevice
6	table:IPaddressdefinition	table:assign-to	table:IPaddressdefinition
7	table:DHCPjob	table:automatically-assign	table:DHCPjob
8	table:IPtypes	table:can-be-diffeent	table:IPtypes
9	table:AntivirusProtection	table:can_be_the	table:AntivirusProtection
10	table:FirewallsProtection	table:can_be_the	table:FirewallsProtection
11	table:UserAwarenessProtection	table:can_be_the	table:UserAwarenessProtection

FIGURE 5.7: PART OF THE RESULT OF PROTEGE IN SPARQL SERVER

Likewise, this ontology can be connected into the python programming language using the owlready2 and flask and able to view the whole information about the ontology on the website which is created in HTML. The ontology information will be displayed in the localhost. The result can be shown in the following image.

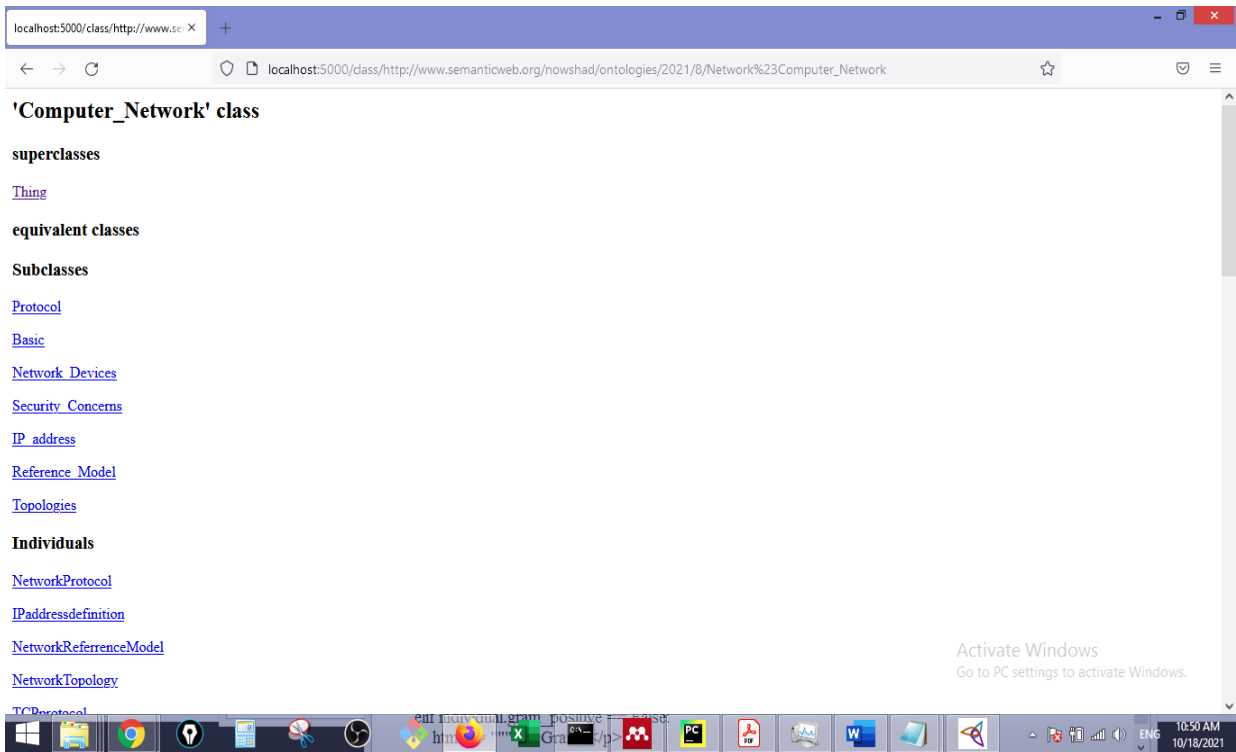


FIGURE 5.8: ONTOLOGY INFORMATION IN PYTHON

This dynamic website is created using the flask a python module which can be used to create websites easily. This Flask module allows to associate a URL path on a website with a Python function. The corresponding HTML page can be returned as the result using the function call in python. To obtain the above output three functions are defined with the relevant URL path with three different types of pages such as the “ontology” page, the “class” page, and the “individual” page on the website. You can run this website using "Werkzeug Server". Thus, when running this website, it will run in the local host browser on port number 5000

## 5.4 Question Processing

The question answering process starts with the question analysis phase. Normally, the same question can be asked in different ways in natural language. Besides, the user’s input can be an assertive or interrogative sentence. Therefore, the understandings of the semantic relationship of words in a sentence are very important. In this stage, the question type also needs to be defined clearly. The following table shows the sample of that question classification with its meaning.

TABLE 5.1: THE SAMPLE QUESTION ANALYSIS

Question Word	Function	Question Focus
Who	Asking what or which person or people (subject)	Person
Whom	Asking what or which person or people (object)	Person
Whose	Asking about the ownership	Person
Where	Asking in or at what place or position	Location

When the user post his question in a natural language in the user interface, the system must process it and transform it to the query language. Several frameworks such as NLTK and Quepy can be used to process natural language question and transform it to a query language. In this research, we have used Quepy python package for the analysis purpose. It has the feature of the NLTK and powerful tool for the natural language processing exist in the python. The answer which is given by the systems depends on the query processing. Therefore, the user’s questions need to be analyzed clearly and all keywords should be identified to get the precises answer from the ontology.

This analysis processes of natural language query can use the techniques such as: the “normalization, tokenization, remove stop words, stemming, and part of speech tagging”. These pre-processing operations have been utilized to the terms that are exist in the “Ontological Dictionary”. The aim of pre-processing the question is to prepare it for mapping with terms in the Ontological Dictionary.

When a question is asked, the question processing component does the aforementioned pre-processing tasks initially. Note that the preprocessing operation in Ontological Dictionary is done once but, in the question, it is done every time a user asks a question.

All ontological concepts including classes, properties and instances can be extracted, preprocessed, and stored in the Ontological Dictionary. This will facilitated to the fast access and matching to the ontology. Therefore, all contents of the developed ontology and RDF store with their names and URIs retrieved to develop the Ontological Dictionary. This dictionary can be considered as the knowledge base which can be used to match the user query with the ontological concepts. This ontological dictionary should provide possible matching concepts from the ontology for the particular word of the user’s query. The main purpose of developing this dictionary is to speed up the search functions and to help for the translation NL queries to SPARQL. In this project, the Ontological Dictionary consist of more than 200 terms from ontology and RDF store that have been selected in computer network domain. Classes, properties, instances, or literals can be

included in the dictionary. This ontological dictionary terms also subjected to the preprocessing operations such as: Normalization, Removing stop words, Tagging, and Stemming.

## **5.5 Ontology Mapper**

Each word in Natural Language queries need to be mapped with the ontological dictionary after pre-processing both the ontological dictionary and user question. We can obtain a list of ontology terms once the mapping of user question and ontological dictionary success. As a result, we can determine the type of each word in question whether it is "class", "property", "instance" or "literal".

## **5.6 Answer Retrieval**

SPARQL query is used to extract answers from the ontology. During the last stage, the answers for a natural language query can be retrieved which is in the form of the triplet. The SPARQL query can be written in the protégé software where it has a separate tab for the SPARQL query. When running the SPARQL query in the protégé then it returned the short answer from the developed ontology.

The other way to perform this task is to use Apache Jena Fuseki Server. Apache Jena Fuseki is an open-source, standalone, SPARQL server that is used to execute the SPARQL query against the ontology. First, we have to load our dataset in this case Network ontology file into the server. Once the dataset upload is successful, then the SPARQL query can be easily run on that. This query also retrieves the short answers from the local ontology databases to the questions which are posed by the users.

The following two images show the SPARQL query, and the answer obtained from the protégé software.

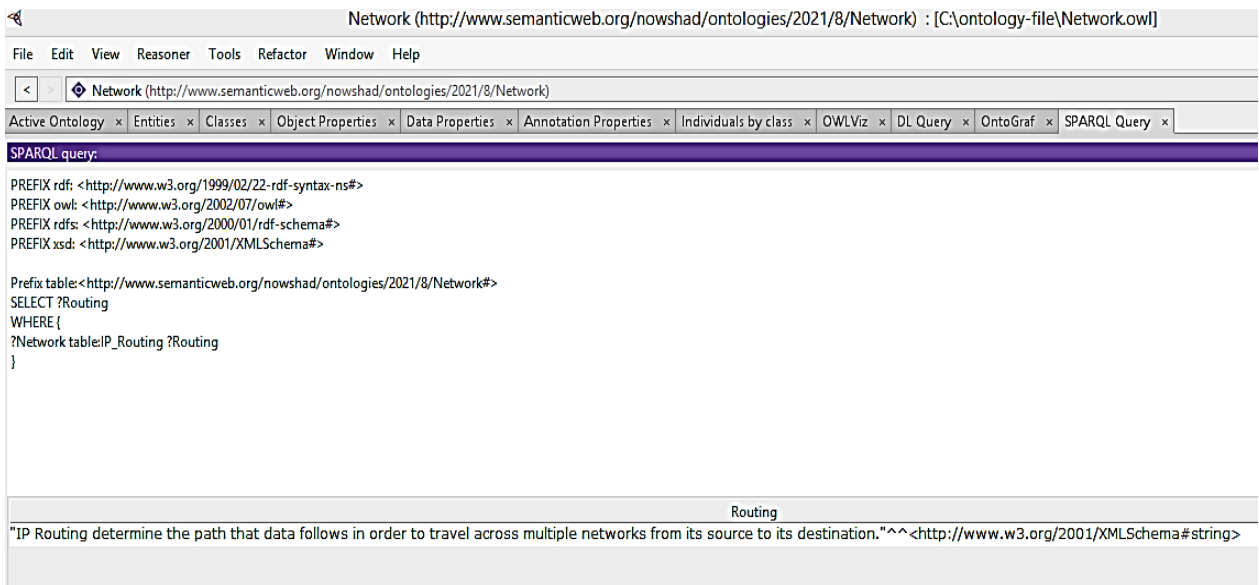


FIGURE 5.9: SPARQL QUERY AND ANSWER IN PROTEGE SOFTWARE

The protégé software does not seem to support some queries like Insert, Update, and Delete information graph and therefore all the testing is carried out in the Jena Fuseki server. The following image illustrates Jena Fuseki's screen where we have the facility to write the SPARQL query, execute, and obtain the results.

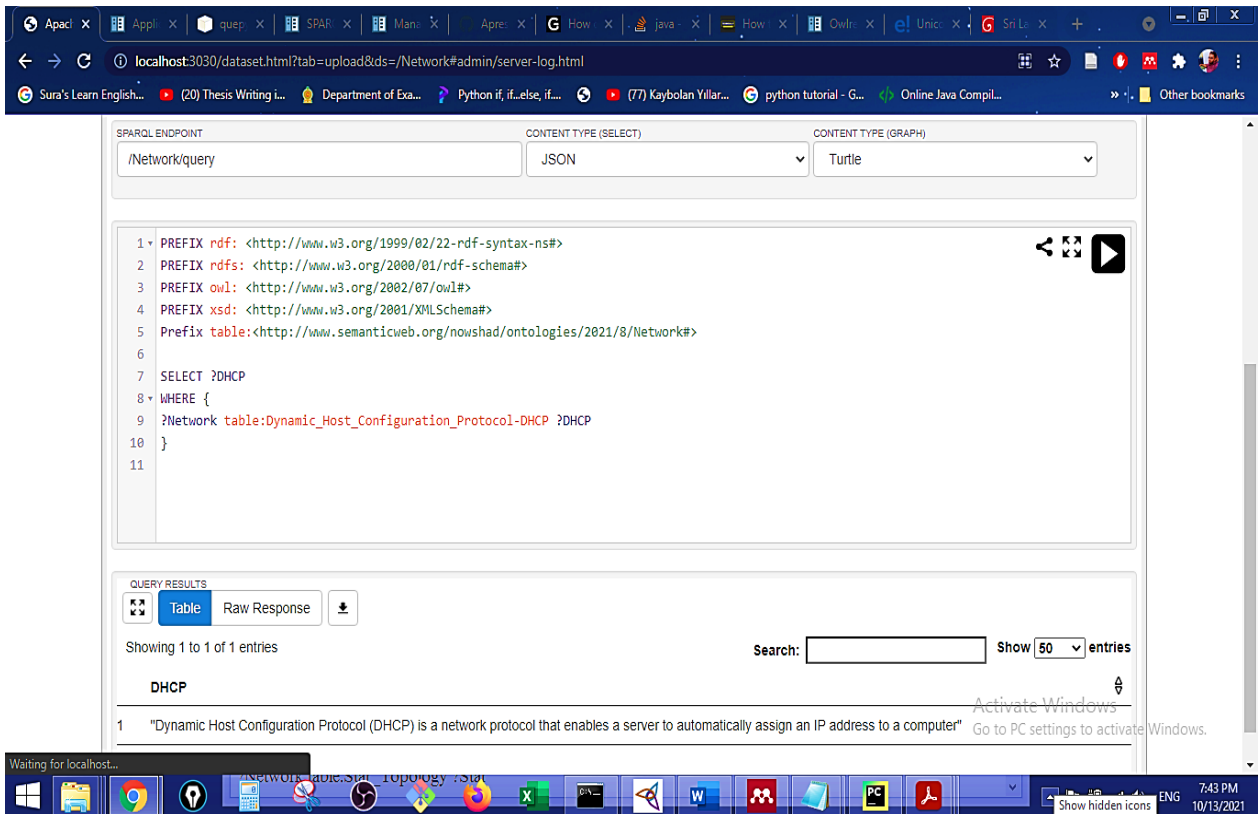


FIGURE 5.10: SPARQL QUERY AND ANSWER IN FUSEKI SERVER



Several python packages such as owlready2, RDFLib, and more can be used to access the OWL ontologies in python. In the owlready2 package, the OWL ontologies can be loaded as the python objects, modify them, save them, and perform reasoning via Hermit. However, the RDFLib library package is used to work with RDF files, and using this python package the answer for the particular query can be obtained from the graph structure (i.e., ontology).

RDFlib package consist of a SPARQL engine, and this engine can be used with Owlready. The SPARQL query can be used to perform more complex searches compared to the ordinary search() method of Owlready. However, the search() method can be more utilized for simple searches as that provides better performance. The RDFlib graph object has the query() method that can be utilized to execute a SPARQL query and returns the result in RDFlib format. The following code snippet of a sample query made through owlready2, and SPARQL query is given below:

```

main.py x Project2.py x semantic.py x
project2
1 import self
2 import base64
3 from owlready2 import *
4 class SparqlQueries:
5     def __init__(self):
6         my_world = World()
7         # path to the owl file is given here
8         my_world.get_ontology("C://ontology-file/Network.owl").load()
9         sync_reasoner(my_world) #reasoner is started and synchronized here
10        self.graph = my_world.as_rdfLib_graph()
11    def search(self):
12        #Search query is given here
13        #Base URL of your ontology has to be given here
14        query = "base <http://www.semanticweb.org/novshad/ontologies/2021/8/Network> " \
15              "SELECT ?s ?p ?o " \
16              "WHERE { " \
17              " ?s ?p ?o . " \
18              "}"
19        #query is being run
20        resultslist = self.graph.query(query)
21        #creating json object
22        response = []
23        for item in resultslist:
24
25            s = str(item['s']).toPython()
26            s = re.sub(r'.*#', "", s)
27
28            p = str(item['p']).toPython()
29            p = re.sub(r'.*#', "", p)
30
31            o = str(item['o']).toPython()
32            o = re.sub(r'.*#', "", o)
33            response.append({'s': s, 'p': p, 'o': o})
34
35        print(response) #just to show the output
36        return response
37    runQuery = SparqlQueries()
38    runQuery.search()

```

FIGURE 5.11: SAMPLE CODE SNIPPET OF OWLREADY2 AND SPARQL QUERY

When we run the above query in python, then it will be returned the answers in the triplet format like S-P-O format. The sample answer of the query is given in the below picture:

```
C:\Users\Nowshad\PycharmProjects\Project2\venv\lib\site-packages\owlready2\hermit\Hermit.jar org.semanticweb
.Hermit.cli.CommandLine -c -O -D -I file:///C:/Users/Public/Documents/Wondershare/CreatorTemp/tmpfwpi7y4_
* Owlready2 * Hermit took 1.385033369064331 seconds
* Owlready2 * Reparenting Network.NetworkProtocol: {owl.DatatypeProperty} => {Network.Protocol}
* Owlready2 * (NB: only changes on entities loaded in Python are shown, other changes are done but not listed)
[{'s': 'http://anonymous', 'p': 'type', 'o': 'Ontology'}, {'s': 'http://www.semanticweb
.org/nowshad/ontologies/2021/8/Network', 'p': 'type', 'o': 'Ontology'}, {'s': 'BasicNetworkTypes', 'p':
'type', 'o': 'ObjectProperty'}, {'s': 'DifferentNetworkType', 'p': 'type', 'o': 'ObjectProperty'}, {'s':
'HasDifferent', 'p': 'type', 'o': 'ObjectProperty'}, {'s': 'HavingDifferent', 'p': 'type', 'o':
'ObjectProperty'}, {'s': 'HavingDifferent', 'p': 'domain', 'o': 'Computer_Network'}, {'s': 'HavingDifferent',
'p': 'range', 'o': 'Network_Devices'}, {'s': 'Network_Protocol', 'p': 'type', 'o': 'ObjectProperty'}, {'s':
'Network_Protocol', 'p': 'domain', 'o': 'Computer_Network'}, {'s': 'Network_Protocol', 'p': 'range', 'o':
'Protocol'}, {'s': 'are', 'p': 'type', 'o': 'ObjectProperty'}, {'s': 'are', 'p': 'domain', 'o': 'Gateway'},
{'s': 'are', 'p': 'domain', 'o': 'Hubs'}, {'s': 'are', 'p': 'domain', 'o': 'Modem'}, {'s': 'are', 'p':
'domain', 'o': 'Routers'}, {'s': 'are', 'p': 'domain', 'o': 'Switch'}, {'s': 'are', 'p': 'range', 'o':
'Network_Devices'}, {'s': 'assign-to', 'p': 'type', 'o': 'ObjectProperty'}, {'s': 'assign-to', 'p': 'domain',
'o': 'IP_address'}, {'s': 'assign-to', 'p': 'range', 'o': 'Computer_Network'}, {'s': 'automatically-assign',
'p': 'type', 'o': 'ObjectProperty'}, {'s': 'automatically-assign', 'p': 'domain', 'o': 'DHCP'}, {'s':
'automatically-assign', 'p': 'range', 'o': 'IP_address'}, {'s': 'can-be-diffeent', 'p': 'type', 'o':
'ObjectProperty'}, {'s': 'can-be-diffeent', 'p': 'domain', 'o': 'IP_address'}, {'s': 'can-be-diffeent', 'p':
```

FIGURE 5.12: SAMPLE ANSWER FOR THE SPARQL QUERY IN PYTHON

In our model, we have used two systems such as; web-based which is a domain-independent QA system, and an ontology-based (Domain-Specific) QA system as shown in the system diagram. Therefore, a web-based system also needs to be implemented using the python programming language. This web-based system is developed using a number of packages such as “NLTK, Flask, gunicorn, unicodedata, wolframalpha, gevent, and wtforms”. Every package has a specific purpose for usage.

NLTK stands National Language Toolkit is a python program which is used to work with human language data processing. This program consist of packages which help to make machine understand the human language and respond appropriately. Gunicorn (“Green Unicorn”) is a python web server that is compatible with several other web servers [57]. The “unicodedata” module give the access to the database of unicode characters (UCD) which contains all unicode characters [59]. The wolframalpha can be described as “an API which can compute expert-level answers using Wolfram's algorithms, knowledgebase and AI technology”. The gevent is a “Coroutine-based concurrency library for Python that uses greenlet to provide a high-level synchronous API on top of the libev or libuv event loop” [60]. WTForms is “a flexible forms validation and rendering library for Python web development. It can work with whatever web

framework and template engine you choose. It supports data validation, CSRF protection, internationalization (I18N), and more” [61].

This system search for the answer from Wikipedia and the Google sites and returned the answer. The default port of Fuseki is 3030, and the browser access is "http://localhost:3030/", whereas the default port for the developed web-based QA system is 9191, and it can browser access "http://localhost:9191/".

The following image shows the web-based QA system question entering the screen.

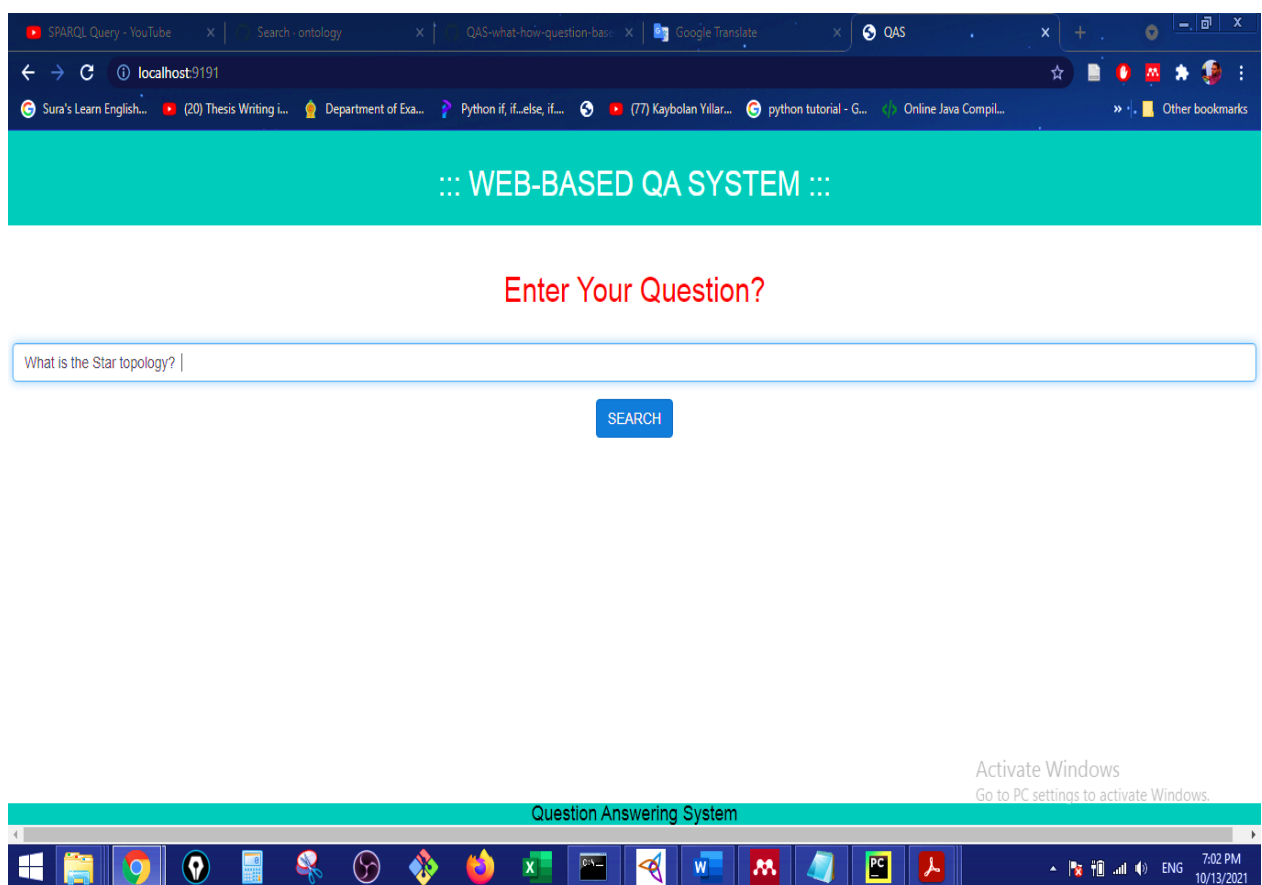


FIGURE 5.13: QUESTION SCREEN OF WEB-BASED SYSTEM

We have used a separate HTML file to display the answer. In that web page, both the question and the relevant answer will be displayed. The answer to the above sample question is returned on the screen as in the following figure.

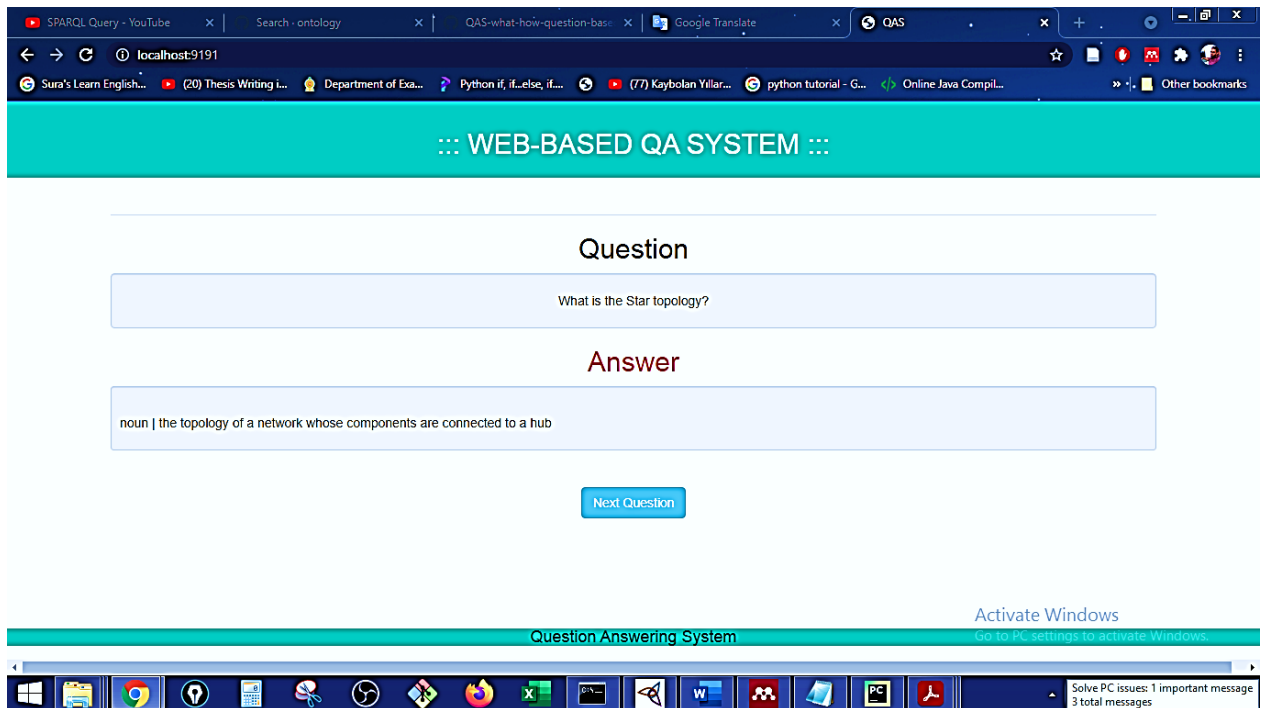


Figure 5.14: Answer Screen of the Web-Based System

The web-based system is able to return answers mostly to the general questions within a short period of time compared to the ontology-based system.

The users have the opportunity to search for answers using these two QA systems. This allows users to find out which of these two systems provides the answers they expect. In addition, they can see the difference in the time it takes for the two systems to respond. All of these can be taken as unique advantages of this QA system developed here.

## 5.7 Chapter Summary

This chapter discussed the methodologies of both systems such as web-based and ontology-based systems in detail. Besides, the technologies that are used in different stages in these QA systems were discussed. The differences between these two methods in terms of time-consuming and accurate answer delivery were briefly highlighted also under this chapter.

# Chapter 6: Results and Evaluation

## Content

- Chapter Overview
- Research results and Evaluation
- Chapter Summary

## 6.1 Chapter Overview

This chapter clearly highlights the performance of the developed QA system. The performance of both systems is discussed with the help of four factors that characterize performance such as accuracy, precision, recall, and f-measure. Based on these performance indicators also users can clearly distinguish between QA systems.

## 6.2 Results and Evaluation

An equal number of test questions were asked in both the systems and the answers given by both of them were obtained. Also, questions were asked in a variety of ways by users. Each time the responses provided by the systems were recorded and the performance of the systems was evaluated.

A better approach is needed to prepare test questions for students' knowledge assessment. The questions prepared should not be vague or confusing to the students. In that regard, various techniques can be explored during question development. However, they should all be thoroughly tested by the system. Here only 45 simple questions were used as test questions.

Based on this 45 sample questions were used. These sample test questions are annexed to Appendix C. The performance of the systems was computed based on the answers obtained to these forty-five sample questions.

The web-based QA system's results can be classified into four groups such as Correct Answer, Nearly correct answer, Couldn't find the answer, and Incorrect answer, and the obtained results can be tabulated as below:

TABLE 6.1: WEB-BASED QA SYSTEM RESULTS

Answer's Category	Quantity
Correct Answers	11
Nearly Correct Answers	6
Couldn't Find the Answers	23
Incorrect Answers	5

Here in this QA model, the non-comprehensive answers are considered to be the nearly correct answers. For instance, when the question is asked like “What is the computer network?” and the system returned the answer as “a network of computers”. This answer is not a comprehensive one, hence we take this kind of answer as the Nearly or Partially correct answer.

Furthermore, the system sometimes did not provide the answers at all and says to try something else. In this scenario, we considered this situation as “Couldn’t Find the Answer”. Otherwise, the system returned complete incorrect answers to some questions. For example, this system responded, “espionage network | old boy network | reticulum | support system | chicken wire | gauze | grillwork | hairnet | netting | reseau | safety net | save-all | snood | spark arrester | sparker |” as the answer for the query of “What is the network types?”. This is not the answer users are expected for that question, and hence this type of answer is classified as “Incorrect Answers”.

However, the ontology-based QA system’s results were categorized into three groups such as Correct Answer, Incorrect Answer, and No Answer. These results also can be tabulated as follows:

TABLE 6.2: ONTOLOGY-BASED QA SYSTEM RESULTS

Answer’s Type	Quantity
Correct Answer	39
Incorrect Answer	03
No Answer Returned	03

Based on these results, the performance of the systems can be evaluated. Four main performance indicators such as Accuracy, Precision, Recall, and F-Measure can be computed using the above results. Each of these factors has a different purpose itself.

Accuracy takes into account whether the model is being properly trained and how it generally works. The precision describes the ability of the model to correctly predict the positives out of all the positive predictions it made. The recall expresses the ability of the system to correctly predict the positives out of actual positives. The F-measure describes the system as a function of precision and recall.

While each of these factors gives meaning individually, the explanation that all of these factors combine is very useful.

TABLE 6.3: PERFORMANCE INDICATORS OF BOTH SYSTEMS

Performance Indicator	Web-Based QA System	Ontology-Based QA System
Accuracy	38%	87%
Precision	0.4	0.93
Recall	0.8	0.93
F-Measure	0.5	0.93

The accuracy is expressed as a percentage and the value of the other performance parameters is found to be between 0 and 1. Value 1 is considered to be optimum and 0 is considered to be worst.

The figures which are tabulated in the above table can be graphically shown to better understandings as below image.

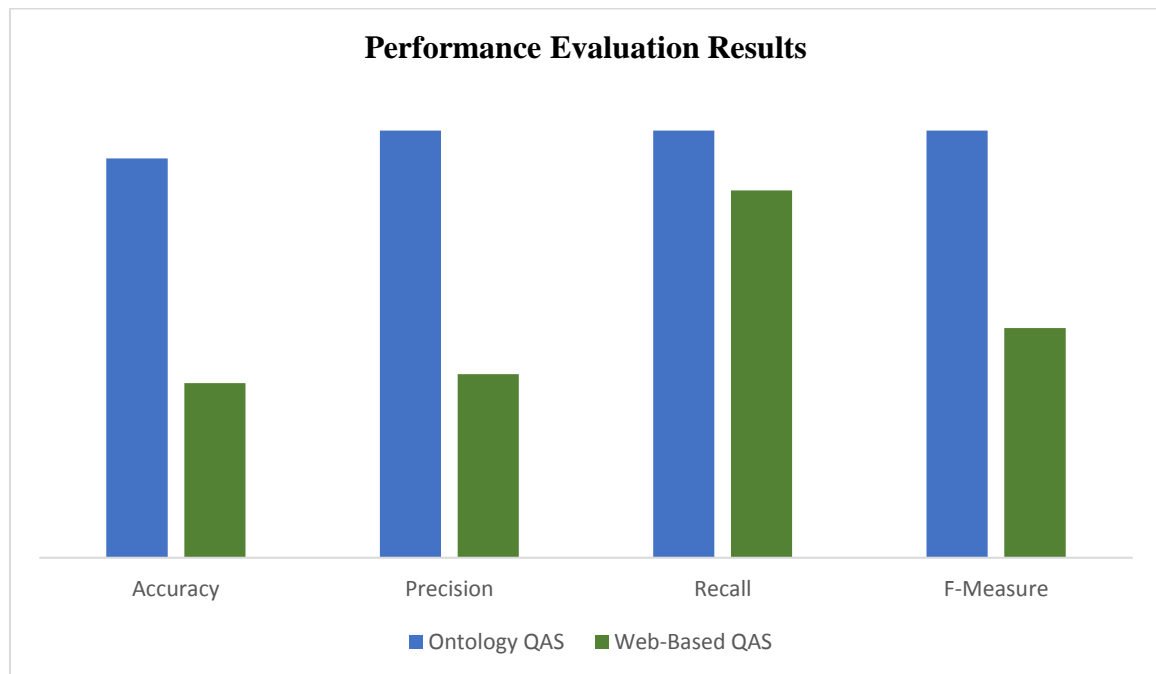


FIGURE 6.1: THE PERFORMANCE COMPARISON CHART

In this calculation, our ontology-based system gives the highest value for all performance indicators than the other one. Therefore, the ontology-based question answering system can be the optimum system for users to find the answers to their queries.



## **6.3 Chapter Summary**

This chapter discussed the results of both question answering systems in detail based on the performance factors such as accuracy, precision, recall, and f-measure. This computation is carried out manually in both systems as we have used a small dataset. The most important thing to note here is that these two QA systems are only assessed with the same sample query dataset.

# Chapter 7: Timeline

The following Gantt chart shows the tasks of the project and the timeline required for those tasks. Although some tasks are dragged out compared to the pre-determined timetable, the whole project is being completed on time.

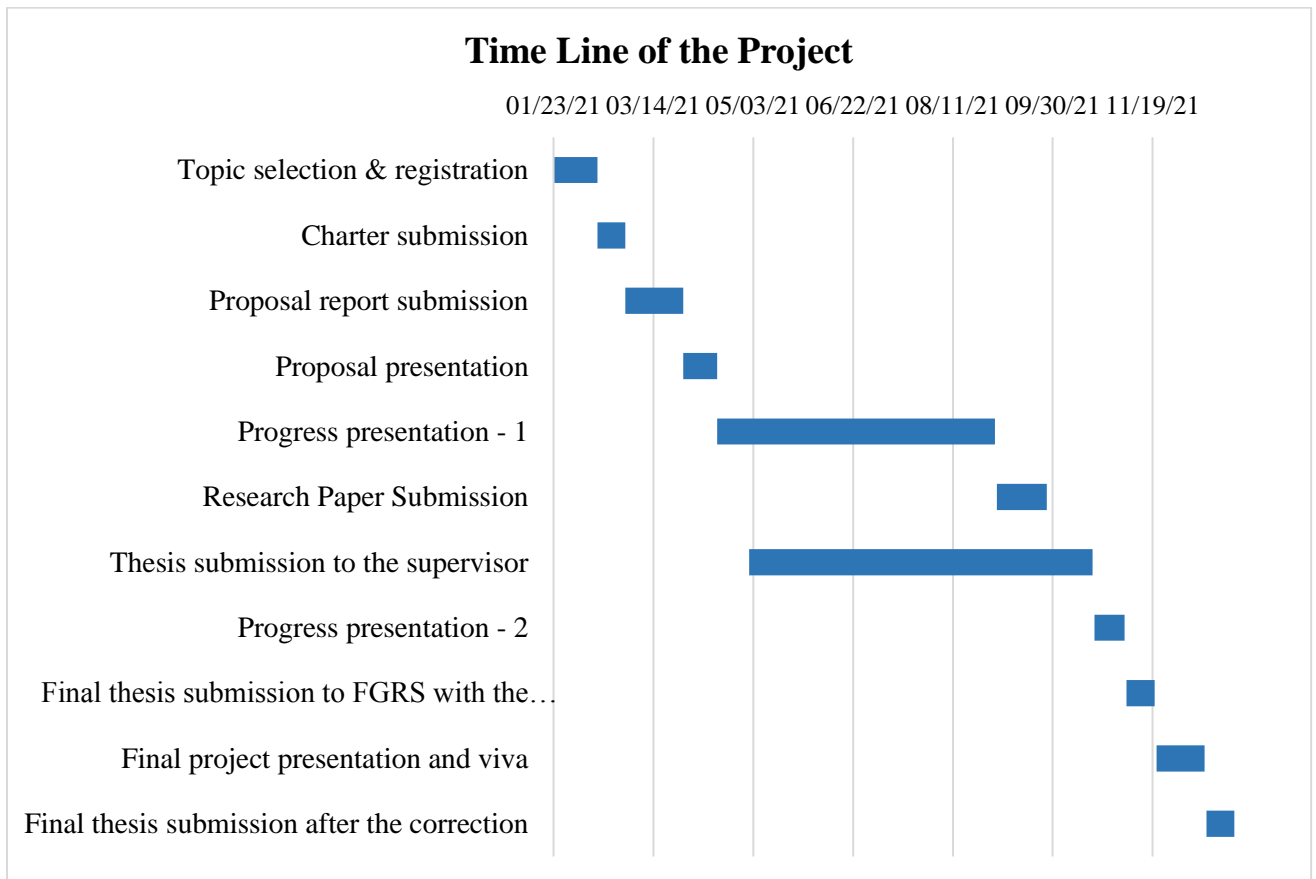


FIGURE 7.1: THE TIMELINE OF THE PROJECT

# Chapter 8: Discussion and Conclusion

The user can enter questions in the natural language completely or just use the keyword in both QA systems. The ontology-based system will be provided the same answer in both situations. However, in the domain-independent QA model, this situation is seen to be completely negative. For example, if acronyms are used in some query, it will not answer the query. But the system answers when it is queried in its full form.

Example: The query of "What is the LAN?" did not give any answer, whereas when this query changed into "What is the Local Area Network?", then the web-based system gives the correct answer to that. In some cases, the acronym gives the wrong answer, and the message appears that the system could not find the result even when asked in its complete form. Due to this, the user is likely to have confusion in getting clear answers to all the questions using this QA system. However, the fact that these confusing cases are not found in the "ontology-based QA system" will be an advantage.

The "Web-Based QA system" consumes more time compared to the ontology-based system as it consists of many resources. The domain-specific or ontology-based QA system has higher accuracy than the open-domain QA system. Especially when the user asks a very specific question in any particular domain (e.g., Computer Network Module), then the open-domain system returned a number of incorrect answers or could not find the answer. Because the web-based system is generally able to provide correct answers for common general questions. However, an ontology-based QA system gives very precise answers in that particular domain. The performance matrix of the system further confirms this outcome.

This ontology can be extended to accommodate other modules like Operating Systems, DBMS, and Data Communication, etc., in the ICT subject in the future. This can be done with small modifications of the current ontology. Otherwise, separate ontologies can be developed for each module and integrate all of them to answer the particular user query.

Besides, this ontology-based QA system can be implemented in other local languages of Sri Lanka like Sinhala or Tamil. Then, most users can greatly benefit from those QA systems. This QA model can be incorporated with some machine learning algorithms to improve its efficiency and usage across multiple dynamic knowledge-base.

The instances of the ontology can be constructed and saved in a database separately in the “RDF format”. Then, no need to access the ontology all the time when the question is asked. So, it helps the system to be scalable and reusable for other application in the similar domain.

# Appendix A: Model Development

## A.1 Frameworks / Software Tools

The essential tools which are used to develop this QA system are listed below. It can be downloaded on their official site and the links are also given.

### A.1.1 Protégé software

Protégé is a free and open-source tool that is used to construct domain models and knowledge-based applications with ontologies. It has multiple versions, such as Protege Desktop, Web Protégé, etc. It also provides a platform for developers to use the terminologies in end-user applications

**Access:** <https://protege.stanford.edu/>

### A.1.2 PyCharm IDE

It is a dedicated Python Integrated Development Environment (IDE) that provides a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

**Access:** <https://www.jetbrains.com/pycharm/download/#section=windows>

### A.1.3 NLTK

Natural Language Toolkit is an excellent tool (python package) to work with human language data, and the NLTK module is used for natural language processing. It is one of the most powerful NLP libraries, which contains packages to make machines understand human language and reply to it with an appropriate response

**Access:** <https://www.nltk.org/>

### A.1.4 Pandas

It is a fast, powerful, flexible, and easy-to-use open-source software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

**Access:** <https://pandas.pydata.org/>

### **A.1.5 NumPy**

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

**Access:** <https://numpy.org/>

### **A.1.6 Scikit-Learn**

Scikit-learn is a free software machine learning library for the Python programming language. It is a simple and efficient tool for predictive data analysis which is built on NumPy, SciPy, and matplotlib. It features various classification, regression and clustering algorithms including support vector machines.

**Access:** <https://scikit-learn.org>

### **A.1.7 Quepy**

It is a python framework which is used to transform natural language questions to queries in a database query language like SPARQL. It can be easily customized to different kinds of questions in natural language and database queries.

**Access:** <https://quepy.readthedocs.io/en/latest/tutorial.html#defining-the-regex>

### **A.1.8 Apache Jena Fuseki**

Apache Jena is a free and open-source Java framework that is used to build Semantic Web and Linked Data applications. Apache Jena Fuseki is a SPARQL server. It can run as an operating system service, as a Java web application (WAR file), and as a standalone server.

<http://jena.apache.org/download/index.cgi>

# Appendix B: SPARQL Queries and Results

S. No	SPARQL Query	Answer
1.	<pre>SELECT ?Definition WHERE { ?Network table:Computer_Network_Definition ?Definition }</pre>	"two or more computers that are linked together using a communication medium"
2.	<pre>SELECT ?Purpose WHERE { ?Network table:Computer_Network_Purpose ?Purpose }</pre>	"The main purpose of networking is to sharing resources."
3.	<pre>SELECT ?Malware WHERE { ?Network table:Malware_Threats ?Malware }</pre>	"Malwares are the softwares that is written for malicious purposes"
4.	<pre>SELECT ?Trojan WHERE { ?Network table:Trojan_Threats ?Trojan }</pre>	"Any malicious computer program used to invade into a computer by misleading users known as Trojan"
5.	<pre>SELECT ?Phising WHERE { ?Network table:Phising_Threats ?Phising }</pre>	"An attempt to obtain sensitive information such as usernames, passwords, and credit card details by pretending as a trustworthy person."
6.	<pre>SELECT ?Virus WHERE { ?Network table:Virus_Threats ?Virus }</pre>	"Viruses are programs which enters into the system, run and perform malicious activities unknowingly."
7.	<pre>SELECT ?Firewalls WHERE { ?Network table:Firewalls_Protection ?Firewalls }</pre>	"Firewalls are systems that act against unauthorized accesses to protected data."
8.	<pre>SELECT ?Antivirus WHERE {</pre>	

	?Network table:Antivirus_Protection ?Antivirus }	"Antivirus softwares are software that detect and quarantine the malicious software that tries to harm a computer"
9.	SELECT ?User_awareness WHERE { ?Network table:User_awareness_Protection ?User_awareness }	"Computer users must be properly educated to protect the network devices against malicious attacks and unauthorized accesses."
10.	SELECT ?IP_address WHERE { ?Network table:IPdefinition ?IP_address }	"IP addresses are unique addresses assigned to each device on the network."
11.	SELECT ?Versions WHERE { ?Network table:IP_Versions ?Versions }	"IP Version 4 (IPv4) and IP Version 6 (IPv6)"
12.	SELECT ?Classes WHERE { ?Network table:IP_Classes ?Classes }	"There are five network classes, Class A to Class E. Class D and Class E which are reserved for research and development purposes"
13.	SELECT ?Types WHERE { ?Network table:IP_Types ?Types }	Two type of IP address are there such as Public IP addresses and Private IP addresses."
14.	SELECT ?Routing WHERE { ?Network table:IP_Routing ?Routing }	"IP Routing determine the path that data follows in order to travel across multiple networks from its source to its destination."
15.	SELECT ?DHCP WHERE { ?Network table:Dynamic_Host_Configuration_Protocol- DHCP ?DHCP }	"Dynamic Host Configuration Protocol (DHCP) is a network protocol that enables a server to automatically assign an IP address to a computer"
16.	SELECT ?Bus WHERE { ?Network table:Bus_Topology ?Bus }	"In a bus topology all nodes are connected to the linear cable with terminators at each end."
17.	SELECT ?Ring WHERE { ?Network table:Ring_Topology ?Ring	"Each station is directly connected only to two of its



	} }	neighbors and messages sent between two stations pass through all of the stations in between"
18.	SELECT ?Star WHERE { ?Network table:Star_Topology ?Star }	"A star topology is designed with each node is connected directly to a central network hub or switch."
19.	SELECT ?Mesh WHERE { ?Network table:Mesh_Topology ?Mesh }	"Hosts in point-to-point connection with every other host or may also have hosts which are in point-to-point connection with few hosts only."
20.	SELECT ?Topologies WHERE { ?Network table:Network_Topology ?Topologies }	"The structure of the network of how all the components are interconnected to each other. There are two types of topology: physical and logical topology. Physical topology is the geometric representation of all the nodes in a network"
21.	SELECT ?OSI WHERE { ?Network table:OSIreference ?OSI }	"Open System Interconnection (OSI) Model consist of seven layers such as Application, Presentation, session, transport, network, data link, and physical layer"
22.	SELECT ?Application WHERE { ?Network table:Application_Layer_OSI ?Application }	"Application layer provides network services to applications"
23.	SELECT ?Session WHERE { ?Network table:Session_Layer_OSI ?Session }	"Session layer responsible for Inter-host communication"
24.	SELECT ?Presentation WHERE { ?Network table:Presentation_Layer_OSI ?Presentation }	"Presentation layer represent the data (Data representation)"
25.	SELECT ?Transport WHERE {	

	?Network table:Transport_Layer_OSI ?Transport }	"Transport layer provides end-to-end connection reliability"
26.	SELECT ?NetworkLayer WHERE { ?Network table:Network_Layer_OSI ?NetworkLayer }	"Network layer address the best path"
27.	SELECT ?Physical WHERE { ?Network table:Physical_Layer_OSI ?Physical }	"Physical layer establish the binary transmission (wires, connectors, voltage data rates, etc.)"
28.	SELECT ?Datalink WHERE { ?Network table:Data_Link_Layer_OSI ?Datalink }	"Data link layer access to the media"
29.	SELECT ?InternetLayer WHERE { ?Network table:Internet_Layer_TCP_IP ?InternetLayer }	"Internet layer defines the datagram and handles the routing of data"
30.	SELECT ?Transport_Layer WHERE { ?Network table:Host_to_Host_Transport_Layer_TCP_IP ?Transport_Layer }	"Host-to-Host Transport Layer provides end to end data delivery services"
31.	SELECT ?Network_Layer WHERE { ?Network table:Network_Access_Layer_TCP_IP ?Network_Layer }	"Network Access Layer consists of routing for accessing physical network"
32.	SELECT ?Application_Layer WHERE { ?Network table:Application_Layer_TCP_IP ?Application_Layer }	"Consists of applications and processes that uses the network."
33.	SELECT ?UDP_Protocol WHERE { ?Network table:UDP_Network_Protocol ?UDP_Protocol }	"User Datagram Protocol (UDP) is simple and suitable for query based communications and it is not connection oriented"
	SELECT ?TCP_Protocol	

34.	WHERE { ?Network table:TCP_Network_Protocol ?TCP_Protocol }	"Transmission Control Protocol (TCP) is a connection oriented protocol and uses sequenced acknowledgment with retransmission of packets when necessary"
35.	SELECT ?Gateway WHERE { ?Network table:Gateway_Device ?Gateway }	"Gateway is a device which connects two or more networks."
36.	SELECT ?Switch WHERE { ?Network table:Switch_Device ?Switch }	"A switch receives data in one of its incoming connections and forwards the data only on the outgoing connection which connects to the destination device"
37.	SELECT ?Hubs WHERE { ?Network table:Hubs_Device ?Hubs }	"A hub receives data in one of its incoming connections and then shall forward the data to all of its outgoing connection."
38.	SELECT ?Modem WHERE { ?Network table:Modem_Device ?Modem }	"Modem is a hardware device that connects a computer or router to a broadband network."
39.	SELECT ?Routers WHERE { ?Network table:Routers_Device ?Routers }	"Devices that are capable of communicating with similar devices over the network, collaborate among themselves and find paths for arriving data"
40.	SELECT ?Protocol WHERE { ?Network table:NetworkProtocol ?Protocol }	"A network protocol is an established set of rules that determine how data is transmitted between different devices in the same network"
41.	SELECT ?Network_Types WHERE { ?Network table:Network_Types ?Network_Types }	"A computer network can be categorized by their size. A computer network is mainly of four types such as: PAN,LAN,MAN, and WAN"
	SELECT ?MAN	

42.	<pre>WHERE { ?Network table:MAN_Type ?MAN }</pre>	<p>"A metropolitan area network is a network that covers a larger geographic area by interconnecting a different LAN to form a larger network. In MAN, various LANs are connected to each other through a telephone exchange line. It has a higher range than Local Area Network(LAN)"</p>
43.	<pre>SELECT ?PAN WHERE { ?Network table:LAN_Type ?LAN }</pre>	<p>"Personal Area Network is a network arranged within an individual person, typically within a range of 10 meters. It is used for connecting the computer devices of personal use. Personal computer devices that are used to develop the personal area network are the laptop, mobile phones, media player and play stations."</p>
44.	<pre>SELECT ?WAN WHERE { ?Network table:WAN_Type ?WAN }</pre>	<p>"A Wide Area Network is a network that extends over a large geographical area such as states or countries. A Wide Area Network is not limited to a single location, but it spans over a large geographical area through a telephone line, fibre optic cable or satellite links. The internet is one of the biggest WAN in the world."</p>
45.	<pre>SELECT ?LAN WHERE { ?Network table:LAN_Type ?LAN }</pre>	<p>"Local Area Network is a group of computers connected through a communication medium such as twisted pair, coaxial cable, etc. to each other in a small area such as building, office"</p>
46.	<pre>SELECT ?ReferenceModel WHERE { ?Network table:ReferenceModels ?ReferenceModel }</pre>	<p>"Reference models give a conceptual framework that standardizes communication between heterogeneous networks."</p>

# Appendix C: The Sample Test Questions

1. What is the computer network?
2. What is the purpose of computer networking?
3. What is the network types?
4. What is the Malware?
5. What is the trojan?
6. What is the Phishing?
7. What is the virus?
8. What is the Antivirus?
9. What is the Firewalls?
10. What is the role of user awareness in network protection?
11. What is the IP address?
12. What are the IP versions?
13. How many IP address classes are there?
14. How many IP address types are there?
15. What is routing?
16. What is the DHCP?
17. What is the topology?
18. What is the Bus topology?
19. What is the Star topology?
20. What is the ring topology?
21. What is the mesh topology?
22. What is the Reference Model of a computer network?
23. What is the OSI Reference Model?
24. What is the job of the Application layer?
25. What is the job of the Session layer?
26. What is the job of the presentation layer?
27. What is the job of the transport layer?
28. What is the role of the network layer?
29. What is the job of the physical layer?
30. What is the job of the data link layer?
31. What is the Internet layer in the TCP/IP model?

32. What is the role of the transport layer in the TCP/IP model?
33. What is the role of the network layer in the TCP/IP model?
34. What is the role of the application layer in the TCP/IP model?
35. What is the meaning of protocol?
36. What is the UDP protocol?
37. What is the TCP protocol?
38. What is the gateway?
39. What is the switch?
40. What is the hubs?
41. What is the modem?
42. What is the routers?
43. What is the LAN?
44. What is the MAN?
45. What is the WAN?

# Appendix D: Complete RDFS for Network Module Ontology

```
“<?xml version="1.0"?>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.semanticweb.org/nowshad/ontologies/2021/8/Network"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  ontologyIRI="http://www.semanticweb.org/nowshad/ontologies/2021/8/Network">
  <Prefix name="" IRI="http://www.semanticweb.org/nowshad/ontologies/2021/8/Modal-1#"/>
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
  <Prefix name="xml" IRI="http://www.w3.org/XML/1998/namespace"/>
  <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
  <Prefix name="Network" IRI="http://www.semanticweb.org/nowshad/ontologies/2021/8/Network#"/>
  <Annotation> <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <Literal xml:lang="en" datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">
  This is an ontology for Computer Network module which is a ICT subject of Grade 12</Literal>
  </Annotation>
  <Declaration> <NamedIndividual IRI="#IPversions"/> </Declaration>
  <Declaration> <Class IRI="#User_awareness"/> </Declaration>
  <Declaration> <DataProperty IRI="#Network_Access_Layer_TCP_IP"/> </Declaration>
  <Declaration> <Class IRI="#LAN"/> </Declaration>
  <Declaration> <Class IRI="#Network_Access_Layer"/> </Declaration>
  <Declaration> <Class IRI="#Modem"/> </Declaration>
  <Declaration> <Class IRI="#Protocol"/> </Declaration>
  <Declaration> <Class IRI="#Bus"/> </Declaration>
  <Declaration> <Class IRI="#OSI_Model"/> </Declaration>
  <Declaration> <DataProperty IRI="#Star_Topology"/> </Declaration>
  <Declaration> <Class IRI="#Types"/> </Declaration>
  <Declaration> <NamedIndividual IRI="#OSIapplication"/> </Declaration>
  <Declaration> <NamedIndividual IRI="#TCPinternet"/> </Declaration>
  <Declaration> <Class IRI="#Basic"/> </Declaration>
  <Declaration> <DataProperty IRI="#Gateway_Device"/> </Declaration>
  <Declaration> <Class IRI="#Classes"/> </Declaration>
  <Declaration> <DataProperty IRI="#Presentation_Layer_OSI"/> </Declaration>
  <Declaration> <DataProperty IRI="#Computer_Network_Definition"/> </Declaration>
  <Declaration> <DataProperty IRI="#MAN_Type"/> </Declaration>
  <Declaration> <Class IRI="#Network_Devices"/> </Declaration>
  <Declaration> <DataProperty IRI="#IP_Versions"/> </Declaration>”
```

```

“ <Declaration>      <NamedIndividual IRI="#SwitchDevice"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#TCPapplication"/> </Declaration>
<Declaration>      <DataProperty IRI="#Network_Layer_OSI"/> </Declaration>
<Declaration>      <Class IRI="#Switch"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#FirewallsProtection"/> </Declaration>
<Declaration>      <Class IRI="#WAN"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#StarTopology"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#RoutersDevice"/> </Declaration>
<Declaration>      <Class IRI="#Application_layer"/> </Declaration>
<Declaration>      <Class IRI="#Application_Layer"/> </Declaration>
<Declaration>      <ObjectProperty IRI="#consist"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#IPclasses"/> </Declaration>
<Declaration>      <DataProperty IRI="#Mesh_Topology"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#MANnetwork"/> </Declaration>
<Declaration>      <DataProperty IRI="#OSireference"/> </Declaration>
<Declaration>      <Class IRI="#Antivirus"/> </Declaration>
<Declaration>      <ObjectProperty IRI="#BasicNetworkTypes"/> </Declaration>
<Declaration>      <Class IRI="#Security_Concerns"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#MeshTopology"/> </Declaration>
<Declaration>      <DataProperty IRI="#Modem_Device"/> </Declaration>
<Declaration>      <DataProperty IRI="#Routers_Device"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#RingTopology"/> </Declaration>
<Declaration>      <DataProperty IRI="#Transport_Layer_OSI"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#MalwareThreats"/> </Declaration>
<Declaration>      <Class IRI="#TCP/IP_Model"/> </Declaration>
<Declaration>      <DataProperty IRI="#Physical_Layer_OSI"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#TCPnetwork"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#TCPtransport"/> </Declaration>
<Declaration>      <DataProperty IRI="#Bus_Topology"/> </Declaration>
<Declaration>      <Class IRI="#TCP"/> </Declaration>
<Declaration>      <DataProperty IRI="#Virus_Threats"/> </Declaration>
<Declaration>      <Class IRI="#IP_address"/> </Declaration>
<Declaration>      <ObjectProperty IRI="#is-example-for"/> </Declaration>
<Declaration>      <ObjectProperty IRI="#are"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#GatewayDevice"/> </Declaration>
<Declaration>      <DataProperty IRI="#Application_Layer_TCP_IP"/> </Declaration>
<Declaration>      <ObjectProperty IRI="#help-to"/> </Declaration>
<Declaration>      <ObjectProperty IRI="#facing"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#NetworkPurpose"/> </Declaration>
<Declaration>      <DataProperty IRI="#NetworkProtocol"/> </Declaration>
<Declaration>      <Class IRI="#Data_link_Layer"/> </Declaration>
<Declaration>      <ObjectProperty IRI="#HavingDifferent"/> </Declaration>
<Declaration>      <Class IRI="#Host_to_Host_Transport_Layer"/> </Declaration>
<Declaration>      <Class IRI="#Presentation_Layer"/> </Declaration>
<Declaration>      <DataProperty IRI="#UDP_Network_Protocol"/> </Declaration>
<Declaration>      <ObjectProperty IRI="#is-a"/> </Declaration>”

```



```

“ <Declaration>      <ObjectProperty IRI="#includes"/> </Declaration>
<Declaration>      <DataProperty IRI="#Switch_Device"/> </Declaration>
<Declaration>      <DataProperty IRI="#TCP_Network_Protocol"/> </Declaration>
<Declaration>      <DataProperty IRI="#Malware_Threats"/> </Declaration>
<Declaration>      <Class IRI="#Transport_Layer"/> </Declaration>
<Declaration>      <Class IRI="#UDP"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#TCPprotocol"/> </Declaration>
<Declaration>      <ObjectProperty IRI="#hasdifferent"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#OSIphysical"/> </Declaration>
<Declaration>      <DataProperty IRI="#Computer_Network_Purpose"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#OSIreferencemodel"/> </Declaration>
<Declaration>      <Class IRI="#Hubs"/> </Declaration>
<Declaration>      <Class IRI="#Phising"/> </Declaration>
<Declaration>      <Class IRI="#Virus"/> </Declaration>
<Declaration>      <Class IRI="#Star"/> </Declaration>
<Declaration>      <Class IRI="#Reference_Model"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#NetworkReferenceModel"/> </Declaration>
<Declaration>      <DataProperty IRI="#WAN_Type"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#UserAwarenessProtection"/> </Declaration>
<Declaration>      <Class IRI="#Threats"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#OSIdatalink"/> </Declaration>
<Declaration>      <DataProperty IRI="#Firewalls_Protection"/> </Declaration>
<Declaration>      <Class IRI="#Gateway"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#NetworkDefinition"/> </Declaration>
<Declaration>      <ObjectProperty IRI="#DifferentNetworkType"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#OSItransport"/> </Declaration>
<Declaration>      <Class IRI="#PAN"/> </Declaration>
<Declaration>      <DataProperty IRI="#Session_Layer_OSI"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#VirusThreats"/> </Declaration>
<Declaration>      <ObjectProperty IRI="#has-two"/> </Declaration>
<Declaration>      <DataProperty IRI="#IP_Classes"/> </Declaration>
<Declaration>      <Class IRI="#Mesh"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#HubsDevices"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#DifferentComputerNetworks"/> </Declaration>
<Declaration>      <Class IRI="#Definition"/> </Declaration>
<Declaration>      <DataProperty IRI="#Host_to_Host_Transport_Layer_TCP_IP"/>
</Declaration>
<Declaration>      <NamedIndividual IRI="#WANnetwork"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#DHCPjob"/> </Declaration>
<Declaration>      <ObjectProperty IRI="#has-some"/> </Declaration>
<Declaration>      <ObjectProperty IRI="#assign-to"/> </Declaration>
<Declaration>      <ObjectProperty IRI="#is-one"/> </Declaration>
<Declaration>      <DataProperty IRI="#Phising_Threats"/> </Declaration>
<Declaration>      <NamedIndividual IRI="#TrojanThreats"/> </Declaration>
<Declaration>      <Class IRI="#Protection"/> </Declaration>
<Declaration>      <DataProperty IRI="#Data_Link_Layer_OSI"/> </Declaration>”

```

```

“ <Declaration> <NamedIndividual IRI="#PANnetwork"/> </Declaration>
<Declaration> <Class IRI="#NetworkTypes"/> </Declaration>
<Declaration> <Class IRI="#Internet_Layer"/> </Declaration>
<Declaration> <DataProperty IRI="#Ring_Topology"/> </Declaration>
<Declaration> <DataProperty IRI="#IP_Types"/> </Declaration>
<Declaration> <ObjectProperty IRI="#automatically-assign"/> </Declaration>
<Declaration> <NamedIndividual IRI="#ModemDevice"/> </Declaration>
<Declaration> <Class IRI="#Ring"/> </Declaration>
<Declaration> <Class IRI="#Malware"/> </Declaration>
<Declaration> <Class IRI="#Firewalls"/> </Declaration>
<Declaration> <Class IRI="#Network_Layer"/> </Declaration>
<Declaration> <DataProperty IRI="#Hubs_Device"/> </Declaration>
<Declaration> <NamedIndividual IRI="#UDPprotocol"/> </Declaration>
<Declaration> <Class IRI="#Physical_Layer"/> </Declaration>
<Declaration> <NamedIndividual IRI="#NetworkTopology"/> </Declaration>
<Declaration> <DataProperty IRI="#Network_Topology"/> </Declaration>
<Declaration> <DataProperty IRI="#Dynamic_Host_Configuration_Protocol-DHCP"/>
</Declaration>
<Declaration> <DataProperty IRI="#Internet_Layer_TCP_IP"/> </Declaration>
<Declaration> <DataProperty IRI="#Trojan_Threats"/> </Declaration>
<Declaration> <NamedIndividual IRI="#IProuting"/> </Declaration>
<Declaration> <DataProperty IRI="#PAN_Type"/> </Declaration>
<Declaration> <Class IRI="#Purpose"/> </Declaration>
<Declaration> <DataProperty IRI="#IP_Routing"/> </Declaration>
<Declaration> <NamedIndividual IRI="#IPaddressdefinition"/> </Declaration>
<Declaration> <Class IRI="#Topologies"/> </Declaration>
<Declaration> <ObjectProperty IRI="#can_be_the"/> </Declaration>
<Declaration> <NamedIndividual IRI="#OSIsession"/> </Declaration>
<Declaration> <ObjectProperty IRI="#Network_Protocol"/> </Declaration>
<Declaration> <Class IRI="#DHCP"/> </Declaration>
<Declaration> <DataProperty IRI="#Antivirus_Protection"/> </Declaration>
<Declaration> <NamedIndividual IRI="#LANnetwork"/> </Declaration>
<Declaration> <DataProperty IRI="#IPdefinition"/> </Declaration>
<Declaration> <DataProperty IRI="#Network_Types"/> </Declaration>
<Declaration> <ObjectProperty IRI="#can-be-diffeent"/> </Declaration>
<Declaration> <Class IRI="#Routers"/> </Declaration>
<Declaration> <Class IRI="#Routing"/> </Declaration>
<Declaration> <ObjectProperty IRI="#giveConceptualFramework"/> </Declaration>
<Declaration> <NamedIndividual IRI="#AntivirusProtection"/> </Declaration>
<Declaration> <DataProperty IRI="#LAN_Type"/> </Declaration>
<Declaration> <NamedIndividual IRI="#IPtypes"/> </Declaration>
<Declaration> <ObjectProperty IRI="#has-five"/> </Declaration>
<Declaration> <DataProperty IRI="#ReferenceModels"/> </Declaration>
<Declaration> <ObjectProperty IRI="#HasDifferent"/> </Declaration>
<Declaration> <NamedIndividual IRI="#NetworkProtocol"/> </Declaration>
<Declaration> <NamedIndividual IRI="#PhishingThreats"/> </Declaration>”

```

```

“ <Declaration> <Class IRI="#Trojan"/> </Declaration>
<Declaration> <Class IRI="#Session_Layer"/> </Declaration>
<Declaration> <NamedIndividual IRI="#BusTopology"/> </Declaration>
<Declaration> <DataProperty IRI="#User_awareness_Protection"/> </Declaration>
<Declaration> <NamedIndividual IRI="#OSInetwork"/> </Declaration>
<Declaration> <Class IRI="#Versions"/> </Declaration>
<Declaration> <NamedIndividual IRI="#TCP-IP_referencemodel"/> </Declaration>
<Declaration> <Class IRI="#Computer_Network"/> </Declaration>
<Declaration> <ObjectProperty IRI="#has"/> </Declaration>
<Declaration> <ObjectProperty IRI="#is-the"/> </Declaration>
<Declaration> <Class IRI="#MAN"/> </Declaration>
<Declaration> <NamedIndividual IRI="#OSIpresentation"/> </Declaration>
<Declaration> <DataProperty IRI="#Application_Layer_OSI"/> </Declaration>
<SubClassOf> <Class IRI="#Antivirus"/> <Class IRI="#Protection"/> </SubClassOf>
<SubClassOf> <Class IRI="#Application_Layer"/> <Class IRI="#TCP/IP_Model"/>
</SubClassOf>
<SubClassOf> <Class IRI="#Application_layer"/> <Class IRI="#OSI_Model"/> </SubClassOf>
<SubClassOf> <Class IRI="#Basic"/> <Class IRI="#Computer_Network"/> </SubClassOf>
<SubClassOf> <Class IRI="#Bus"/> <Class IRI="#Topologies"/> </SubClassOf>
<SubClassOf> <Class IRI="#Classes"/> <Class IRI="#IP_address"/> </SubClassOf>
<SubClassOf> <Class IRI="#Computer_Network"/> <Class abbreviatedIRI="owl:Thing"/>
</SubClassOf>
<SubClassOf> <Class IRI="#DHCP"/> <Class IRI="#IP_address"/> </SubClassOf>
<SubClassOf> <Class IRI="#Data_link_Layer"/> <Class IRI="#OSI_Model"/> </SubClassOf>
<SubClassOf> <Class IRI="#Definition"/> <Class IRI="#Basic"/> </SubClassOf>
<SubClassOf> <Class IRI="#Firewalls"/> <Class IRI="#Protection"/> </SubClassOf>
<SubClassOf> <Class IRI="#Gateway"/> <Class IRI="#Network_Devices"/> </SubClassOf>
<SubClassOf> <Class IRI="#Host_to_Host_Transport_Layer"/> <Class IRI="#TCP/IP_Model"/>
</SubClassOf>
<SubClassOf> <Class IRI="#Hubs"/> <Class IRI="#Network_Devices"/> </SubClassOf>
<SubClassOf> <Class IRI="#IP_address"/> <Class IRI="#Computer_Network"/> </SubClassOf>
<SubClassOf> <Class IRI="#Internet_Layer"/> <Class IRI="#TCP/IP_Model"/> </SubClassOf>
<SubClassOf> <Class IRI="#LAN"/> <Class IRI="#NetworkTypes"/> </SubClassOf>
<SubClassOf> <Class IRI="#MAN"/> <Class IRI="#NetworkTypes"/> </SubClassOf>
<SubClassOf> <Class IRI="#Malware"/> <Class IRI="#Threats"/> </SubClassOf>
<SubClassOf> <Class IRI="#Mesh"/> <Class IRI="#Topologies"/> </SubClassOf>
<SubClassOf> <Class IRI="#Modem"/> <Class IRI="#Network_Devices"/> </SubClassOf>
<SubClassOf> <Class IRI="#NetworkTypes"/> <Class IRI="#Basic"/> </SubClassOf>
<SubClassOf> <Class IRI="#Network_Access_Layer"/> <Class IRI="#TCP/IP_Model"/>
</SubClassOf>
<SubClassOf> <Class IRI="#Network_Devices"/> <Class IRI="#Computer_Network"/>
</SubClassOf>
<SubClassOf> <Class IRI="#Network_Layer"/> <Class IRI="#OSI_Model"/> </SubClassOf>
<SubClassOf> <Class IRI="#OSI_Model"/> <Class IRI="#Reference_Model"/> </SubClassOf>
<SubClassOf> <Class IRI="#PAN"/> <Class IRI="#NetworkTypes"/> </SubClassOf>
<SubClassOf> <Class IRI="#Phising"/> <Class IRI="#Threats"/> </SubClassOf>”

```

```

“ <SubClassOf> <Class IRI="#Physical_Layer"/> <Class IRI="#OSI_Model"/> </SubClassOf>
<SubClassOf> <Class IRI="#Presentation_Layer"/> <Class IRI="#OSI_Model"/> </SubClassOf>
<SubClassOf> <Class IRI="#Protection"/> <Class IRI="#Security_Concerns"/> </SubClassOf>
<SubClassOf> <Class IRI="#Protocol"/> <Class IRI="#Computer_Network"/> </SubClassOf>
<SubClassOf> <Class IRI="#Purpose"/> <Class IRI="#Basic"/> </SubClassOf>
<SubClassOf> <Class IRI="#Reference_Model"/> <Class IRI="#Computer_Network"/>
</SubClassOf>
<SubClassOf> <Class IRI="#Ring"/> <Class IRI="#Topologies"/> </SubClassOf>
<SubClassOf> <Class IRI="#Routers"/> <Class IRI="#Network_Devices"/> </SubClassOf>
<SubClassOf> <Class IRI="#Routing"/> <Class IRI="#IP_address"/> </SubClassOf>
<SubClassOf> <Class IRI="#Security_Concerns"/> <Class IRI="#Computer_Network"/>
</SubClassOf>
<SubClassOf> <Class IRI="#Session_Layer"/> <Class IRI="#OSI_Model"/> </SubClassOf>
<SubClassOf> <Class IRI="#Star"/> <Class IRI="#Topologies"/> </SubClassOf>
<SubClassOf> <Class IRI="#Switch"/> <Class IRI="#Network_Devices"/> </SubClassOf>
<SubClassOf> <Class IRI="#TCP"/> <Class IRI="#Protocol"/> </SubClassOf>
<SubClassOf> <Class IRI="#Threats"/> <Class IRI="#Security_Concerns"/> </SubClassOf>
<SubClassOf> <Class IRI="#Topologies"/> <Class IRI="#Computer_Network"/> </SubClassOf>
<SubClassOf> <Class IRI="#Transport_Layer"/> <Class IRI="#OSI_Model"/> </SubClassOf>
<SubClassOf> <Class IRI="#Trojan"/> <Class IRI="#Threats"/> </SubClassOf>
<SubClassOf> <Class IRI="#Types"/> <Class IRI="#IP_address"/> </SubClassOf>
<SubClassOf> <Class IRI="#UDP"/> <Class IRI="#Protocol"/> </SubClassOf>
<SubClassOf> <Class IRI="#User_awareness"/> <Class IRI="#Protection"/> </SubClassOf>
<SubClassOf> <Class IRI="#Versions"/> <Class IRI="#IP_address"/> </SubClassOf>
<SubClassOf> <Class IRI="#Virus"/> <Class IRI="#Threats"/> </SubClassOf>
<SubClassOf> <Class IRI="#WAN"/> <Class IRI="#NetworkTypes"/> </SubClassOf>
<SubClassOf> <Class IRI="#TCP/IP_Model"/> <Class IRI="#Reference_Model"/> </SubClassOf>
<DisjointClasses> <Class IRI="#TCP"/> <Class IRI="#UDP"/> </DisjointClasses>

<ClassAssertion> <Class IRI="#Antivirus"/> <NamedIndividual IRI="#AntivirusProtection"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Bus"/> <NamedIndividual IRI="#BusTopology"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#DHCP"/> <NamedIndividual IRI="#DHCPjob"/> </ClassAssertion>

<ClassAssertion> <Class IRI="#NetworkTypes"/> <NamedIndividual IRI="#Different Computer
Networks"/> </ClassAssertion>

<ClassAssertion> <Class IRI="#Firewalls"/> <NamedIndividual IRI="#FirewallsProtection"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Gateway"/> <NamedIndividual IRI="#GatewayDevice"/>
</ClassAssertion>”

```

```

“<ClassAssertion> <Class IRI="#Hubs"/><NamedIndividual IRI="#HubsDevices"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#IP_address"/> <NamedIndividual IRI="#IPaddressdefinition"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Classes"/> <NamedIndividual IRI="#IPclasses"/> </ClassAssertion>

<ClassAssertion> <Class IRI="#Routing"/><NamedIndividual IRI="#IProuting"/></ClassAssertion>

<ClassAssertion> <Class IRI="#Types"/> <NamedIndividual IRI="#IPtypes"/> </ClassAssertion>

<ClassAssertion><Class IRI="#Versions"/><NamedIndividual IRI="#IPversions"/> </ClassAssertion>

<ClassAssertion> <Class IRI="#LAN"/><NamedIndividual IRI="#LANnetwork"/> </ClassAssertion>

<ClassAssertion> <Class IRI="#MAN"/><NamedIndividual IRI="#MANnetwork"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Malware"/> <NamedIndividual IRI="#MalwareThreats"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Mesh"/> <NamedIndividual IRI="#MeshTopology"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Modem"/> <NamedIndividual IRI="#ModemDevice"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Definition"/> <NamedIndividual IRI="#NetworkDefinition"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Protocol"/> <NamedIndividual IRI="#NetworkProtocol"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Purpose"/> <NamedIndividual IRI="#NetworkPurpose"/>
</ClassAssertion>

<ClassAssertion><Class IRI="#Reference_Model"/> <NamedIndividual IRI= "#Network
ReferenceModel"/> </ClassAssertion>

<ClassAssertion> <Class IRI="#Topologies"/> <NamedIndividual IRI="#NetworkTopology"/>
</ClassAssertion>

<ClassAssertion><Class IRI="#Application_layer"/> <NamedIndividual IRI="#OSIapplication"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Data_link_Layer"/> <NamedIndividual IRI="#OSIdatalink"/>
</ClassAssertion>”

```

```

“<ClassAssertion> <Class IRI="#Network_Layer"/> <NamedIndividual IRI="#OSInetwork"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Physical_Layer"/> <NamedIndividual IRI="#OSIphysical"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Presentation_Layer"/><NamedIndividual IRI="#OSIpresentation"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#OSI_Model"/> <NamedIndividual IRI="#OSIreferencemodel"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Session_Layer"/> <NamedIndividual IRI="#OSIsession"/>
</ClassAssertion>

<ClassAssertion><Class IRI="#Transport_Layer"/> <NamedIndividual IRI="#OSItransport"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#PAN"/> <NamedIndividual IRI="#PANnetwork"/>
</ClassAssertion>

<ClassAssertion><Class IRI="#Phising"/> <NamedIndividual IRI="#PhishingThreats"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Ring"/> <NamedIndividual IRI="#RingTopology"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Routers"/> <NamedIndividual IRI="#RoutersDevice"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Star"/><NamedIndividual IRI="#StarTopology"/> </ClassAssertion>

<ClassAssertion> <Class IRI="#Switch"/> <NamedIndividual IRI="#SwitchDevice"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#TCP/IP_Model"/> <NamedIndividual IRI="#TCP-
IP_referencemodel"/> </ClassAssertion>

<ClassAssertion> <Class IRI="#Application_Layer"/> <NamedIndividual IRI="#TCPapplication"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Internet_Layer"/><NamedIndividual IRI="#TCPinternet"/>
</ClassAssertion>

<ClassAssertion> <Class IRI="#Network_Access_Layer"/> <NamedIndividual IRI="#TCPnetwork"/>
</ClassAssertion>”

```

```

“ <ClassAssertion><Class IRI="#TCP"/> <NamedIndividual IRI="#TCPprotocol"/>
  </ClassAssertion>

  <ClassAssertion><Class IRI="#Host_to_Host_Transport_Layer"/><NamedIndividual
IRI="#TCPtransport"/> </ClassAssertion>

  <ClassAssertion> <Class IRI="#Trojan"/><NamedIndividual IRI="#TrojanThreats"/>
  </ClassAssertion>

  <ClassAssertion><Class IRI="#UDP"/> <NamedIndividual IRI="#UDPprotocol"/>
  </ClassAssertion>

  <ClassAssertion><Class IRI="#User_awareness"/> <NamedIndividual IRI="# UserAwareness
Protection"/> </ClassAssertion>

  <ClassAssertion> <Class IRI="#Virus"/> <NamedIndividual IRI="#VirusThreats"/>
  </ClassAssertion>

  <ClassAssertion> <Class IRI="#WAN"/> <NamedIndividual IRI="#WANnetwork"/>
  </ClassAssertion>

  <ObjectPropertyAssertion><ObjectProperty IRI="#can_be_the"/> <NamedIndividual
IRI="#AntivirusProtection"/> <NamedIndividual IRI="#AntivirusProtection"/>
  </ObjectPropertyAssertion>

  <ObjectPropertyAssertion> <ObjectProperty IRI="#is-one"/> <NamedIndividual
IRI="#BusTopology"/> <NamedIndividual IRI="#BusTopology"/>
  </ObjectPropertyAssertion>

  <ObjectPropertyAssertion> <ObjectProperty IRI="#automatically-assign"/>
    <NamedIndividual IRI="#DHCPjob"/> <NamedIndividual IRI="#DHCPjob"/>
  </ObjectPropertyAssertion>

  <ObjectPropertyAssertion> <ObjectProperty IRI="#BasicNetworkTypes"/>
    <NamedIndividual IRI="#DifferentComputerNetworks"/> <NamedIndividual
IRI="#DifferentComputerNetworks"/> </ObjectPropertyAssertion>

  <ObjectPropertyAssertion> <ObjectProperty IRI="#can_be_the"/> <NamedIndividual IRI
="#FirewallsProtection"/> <NamedIndividual IRI="#FirewallsProtection"/>
  </ObjectPropertyAssertion>

  <ObjectPropertyAssertion><ObjectProperty IRI="#are"/><NamedIndividual IRI="#GatewayDevice"/>
    <NamedIndividual IRI="#GatewayDevice"/> </ObjectPropertyAssertion>

  <ObjectPropertyAssertion> <ObjectProperty IRI="#are"/><NamedIndividual IRI="#HubsDevices"/>”

```

```

“<NamedIndividual IRI="#HubsDevices"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#assign-to"/> <NamedIndividual
IRI="#IPaddressdefinition"/> <NamedIndividual IRI="#IPaddressdefinition"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#has-five"/> <NamedIndividual IRI="#IPclasses"/>
<NamedIndividual IRI="#IPclasses"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#help-to"/><NamedIndividual IRI="#IProuting"/>
<NamedIndividual IRI="#IProuting"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#can-be-diffeent"/> <NamedIndividual
IRI="#IPtypes"/> <NamedIndividual IRI="#IPtypes"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion><ObjectProperty IRI="#has-two"/><NamedIndividual IRI="#IPversions"/>
<NamedIndividual IRI="#IPversions"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#BasicNetworkTypes"/>
<NamedIndividual IRI="#LANnetwork"/> <NamedIndividual IRI="#LANnetwork"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#BasicNetworkTypes"/>
<NamedIndividual IRI="#MANnetwork"/> <NamedIndividual IRI="#MANnetwork"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#is-a"/> <NamedIndividual
IRI="#MalwareThreats"/> <NamedIndividual IRI="#MalwareThreats"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#is-one"/><NamedIndividual
IRI="#MeshTopology"/><NamedIndividual IRI="#MeshTopology"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion><ObjectProperty IRI="#are"/><NamedIndividual IRI="#ModemDevice"/>
<NamedIndividual IRI="#ModemDevice"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion><ObjectProperty IRI="#is-the"/><NamedIndividual IRI="#
NetworkDefinition"/> <NamedIndividual IRI="#NetworkDefinition"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion><ObjectProperty IRI="#Network_Protocol"/><NamedIndividual
IRI="#NetworkProtocol"/> <NamedIndividual IRI="#NetworkProtocol"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#includes"/><NamedIndividual
IRI="#NetworkPurpose"/><NamedIndividual IRI="#NetworkPurpose"/> </ObjectPropertyAssertion>”

```



```

“<ObjectPropertyAssertion><ObjectProperty IRI="#giveConceptualFramework"/>
<NamedIndividual IRI="#NetworkReferenceModel"/><NamedIndividual IRI=
"#NetworkReferenceModel"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#HasDifferent"/> <NamedIndividual
IRI="#NetworkTopology"/><NamedIndividual IRI="#NetworkTopology"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion><ObjectProperty IRI="#has"/> <NamedIndividual IRI="#OSIapplication"/>
<NamedIndividual IRI="#OSIapplication"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion><ObjectProperty IRI="#has"/> <NamedIndividual IRI="#OSIdatalink"/>
<NamedIndividual IRI="#OSIdatalink"/></ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#has"/> <NamedIndividual IRI="#OSInetwork"/>
<NamedIndividual IRI="#OSInetwork"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion><ObjectProperty IRI="#has"/>
<NamedIndividual IRI="#OSIphysical"/> <NamedIndividual IRI="#OSIphysical"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#has"/> <NamedIndividual
IRI="#OSIpresentation"/> <NamedIndividual IRI="#OSIpresentation"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#is-example-for"/> <NamedIndividual
IRI="#OSIreferencemodel"/> <NamedIndividual IRI="#OSIreferencemodel"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#has"/> <NamedIndividual IRI="#OSIsession"/>
<NamedIndividual IRI="#OSIsession"/></ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#has"/><NamedIndividual IRI="#OSItransport"/>
<NamedIndividual IRI="#OSItransport"/></ObjectPropertyAssertion>

<ObjectPropertyAssertion><ObjectProperty IRI="#BasicNetworkTypes"/>
<NamedIndividual IRI="#PANnetwork"/> <NamedIndividual IRI="#PANnetwork"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion><ObjectProperty IRI="#is-a"/> <NamedIndividual
IRI="#PhishingThreats"/> <NamedIndividual IRI="#PhishingThreats"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#is-one"/> <NamedIndividual
IRI="#RingTopology"/> <NamedIndividual IRI="#RingTopology"/> </ObjectPropertyAssertion>

<ObjectPropertyAssertion> <ObjectProperty IRI="#are"/> <NamedIndividual IRI="#RoutersDevice"/>
<NamedIndividual IRI="#RoutersDevice"/> </ObjectPropertyAssertion>”

```

```
“<ObjectPropertyAssertion><ObjectProperty IRI="#is-one"/> <NamedIndividual IRI="#StarTopology"/> <NamedIndividual IRI="#StarTopology"/> </ObjectPropertyAssertion>
```

```
<ObjectPropertyAssertion> <ObjectProperty IRI="#are"/><NamedIndividual IRI="#SwitchDevice"/>  
<NamedIndividual IRI="#SwitchDevice"/> </ObjectPropertyAssertion>
```

```
<ObjectPropertyAssertion><ObjectProperty IRI="#consist"/><NamedIndividual IRI="#TCPApplication"/><NamedIndividual IRI="#TCPApplication"/> </ObjectPropertyAssertion>
```

```
<ObjectPropertyAssertion> <ObjectProperty IRI="#consist"/> <NamedIndividual IRI="#TCPinternet"/> <NamedIndividual IRI="#TCPinternet"/> </ObjectPropertyAssertion>
```

```
<ObjectPropertyAssertion> <ObjectProperty IRI="#consist"/><NamedIndividual IRI="#TCPnetwork"/><NamedIndividual IRI="#TCPnetwork"/></ObjectPropertyAssertion>
```

```
<ObjectPropertyAssertion> <ObjectProperty IRI="#consist"/> <NamedIndividual IRI="#TCPtransport"/> <NamedIndividual IRI="#TCPtransport"/> </ObjectPropertyAssertion>
```

```
<ObjectPropertyAssertion> <ObjectProperty IRI="#is-a"/> <NamedIndividual IRI="#TrojanThreats"/>  
<NamedIndividual IRI="#TrojanThreats"/></ObjectPropertyAssertion>
```

```
<ObjectPropertyAssertion> <ObjectProperty IRI="#can_be_the"/> <NamedIndividual IRI="#UserAwarenessProtection"/> <NamedIndividual IRI="#UserAwarenessProtection"/>  
</ObjectPropertyAssertion>
```

```
<ObjectPropertyAssertion> <ObjectProperty IRI="#is-a"/> <NamedIndividual IRI="#VirusThreats"/>  
<NamedIndividual IRI="#VirusThreats"/> </ObjectPropertyAssertion>
```

```
<ObjectPropertyAssertion> <ObjectProperty IRI="#BasicNetworkTypes"/>  
<NamedIndividual IRI="#WANnetwork"/> <NamedIndividual IRI="#WANnetwork"/>  
</ObjectPropertyAssertion>
```

```
<DataPropertyAssertion> <DataProperty IRI="#Antivirus_Protection"/>  
<NamedIndividual IRI="#AntivirusProtection"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Antivirus softwares are software that detect and quarantine the malicious software that tries to harm a computer</Literal> </DataPropertyAssertion>
```

```
<DataPropertyAssertion> <DataProperty IRI="#Bus_Topology"/>  
<NamedIndividual IRI="#BusTopology"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">In a bus topology all nodes are connected to the linear cable with terminators at each end.</Literal> </DataPropertyAssertion>
```

```
<DataPropertyAssertion> <DataProperty IRI="#Dynamic_Host_Configuration_Protocol-DHCP"/>”
```

“<NamedIndividual IRI="#DHCPjob"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Dynamic Host Configuration Protocol (DHCP) is a network protocol that enables a server to automatically assign an IP address to a computer</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#Network\_Types"/> <NamedIndividual IRI="#DifferentComputerNetworks"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string"> A computer network can be categorized by their size. A computer network is mainly of four types such as: PAN,LAN,MAN, and WAN</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#Firewalls\_Protection"/> <NamedIndividual IRI="#FirewallsProtection"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string"> Firewalls are systems that act against unauthorized accesses to protected data.</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#Gateway\_Device"/> <NamedIndividual IRI="#GatewayDevice"/><Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Gateway is a device which connects two or more networks.</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#Hubs\_Device"/><NamedIndividual IRI="#HubsDevices"/><Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">A hub receives data in one of its incoming connections and then shall forward the data to all of its outgoing connection.</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#IPdefinition"/> <NamedIndividual IRI="#IPaddressdefinition"/> Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">IP addresses are unique addresses assigned to each device on the network.</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#IP\_Classes"/> <NamedIndividual IRI="#IPclasses"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">There are five network classes, Class A to Class E. Class D and Class E which are reserved for research and development purposes</Literal> </DataPropertyAssertion>

<DataPropertyAssertion><DataProperty IRI="#IP\_Routing"/> NamedIndividual IRI="#IProuting"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">IP Routing determine the path that data follows in order to travel across multiple networks from its source to its destination.</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#IP\_Types"/> <NamedIndividual IRI="#IPtypes"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Two type of IP address are there such as Public IP addresses and Private IP addresses.</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#IP\_Versions"/> <NamedIndividual IRI="#IPversions"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">IP Version 4 (IPv4) and IP Version 6 (IPv6)</Literal> </DataPropertyAssertion>”

```
“<DataPropertyAssertion> <DataProperty IRI="#LAN_Type"/> <NamedIndividual
IRI="#LANnetwork"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Local
Area Network is a group of computers connected through a communication medium such as twisted pair,
coaxial cable, etc. to each other in a small area such as building, office.</Literal>
</DataPropertyAssertion>
```

```
<DataPropertyAssertion><DataProperty IRI="#MAN_Type"/> <NamedIndividual IRI=
"#MANnetwork"/> <Literal datatypeIRI="http://www.w3.org/ 2001/XMLSchema#string">A
metropolitan area network is a network that covers a larger geographic area by interconnecting a different
LAN to form a larger network. In MAN, various LANs are connected to each other through a telephone
exchange line. It has a higher range than Local Area Network(LAN)</Literal> </DataPropertyAssertion>
```

```
<DataPropertyAssertion><DataProperty IRI="#Malware_Threats"/> <NamedIndividual
IRI="#MalwareThreats"/> <Literal datatypeIRI= "http://www.w3.org/2001/ XMLSchema#
string">Malwares are the softwares that is written for malicious purposes</Literal>
</DataPropertyAssertion>
```

```
<DataPropertyAssertion> <DataProperty IRI="#Mesh_Topology"/> <NamedIndividual
IRI="#MeshTopology"/> <Literal datatypeIRI="http://www.w3.org/ 2001/XMLSchema#string">Hosts in
point-to-point connection with every other host or may also have hosts which are in point-to-point
connection with few hosts only.</Literal> </DataPropertyAssertion>
```

```
<DataPropertyAssertion> <DataProperty IRI="#Modem_Device"/> <NamedIndividual
IRI="#ModemDevice"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Modem
is a hardware device that connects a computer or router to a broadband network.</Literal>
</DataPropertyAssertion>
```

```
<DataPropertyAssertion> <DataProperty IRI="#Computer_Network_Definition"/>
<NamedIndividual IRI="#NetworkDefinition"/><Literal datatypeIRI="http://www.w3.org/2001/
XMLSchema#string">two or more computers that are linked together using a communication
medium</Literal> </DataPropertyAssertion>
```

```
<DataPropertyAssertion> <DataProperty IRI="#NetworkProtocol"/><NamedIndividual
IRI="#NetworkProtocol"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">A
network protocol is an established set of rules that determine how data is transmitted between different
devices in the same network</Literal> </DataPropertyAssertion>
```

```
<DataPropertyAssertion> <DataProperty IRI="#Computer_Network_Purpose"/>
<NamedIndividual IRI="#NetworkPurpose"/> <Literal datatypeIRI="http://www.w3.org/2001/
XMLSchema#string">The main purpose of networking is to sharing resources.</Literal>
</DataPropertyAssertion>
```

```
<DataPropertyAssertion><DataProperty IRI="#ReferenceModels"/> <NamedIndividual
IRI="#NetworkReferenceModel"/> <Literal datatypeIRI="http://www.w3.org/2001"/>
```

“XMLSchema#string”>Reference models give a conceptual framework that standardizes communication between heterogeneous networks.</Literal> </DataPropertyAssertion>

<DataPropertyAssertion><DataProperty IRI="#Network\_Topology"/> <NamedIndividual IRI="#NetworkTopology"/><Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">The structure of the network of how all the components are interconnected to each other. There are two types of topology: physical and logical topology. Physical topology is the geometric representation of all the nodes in a network</Literal> </DataPropertyAssertion>

<DataPropertyAssertion><DataProperty IRI="#Application\_Layer\_OSI"/>  
<NamedIndividual IRI="#OSIapplication"/><Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Application layer provides network services to applications</Literal>  
</DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#Data\_Link\_Layer\_OSI"/> <NamedIndividual IRI="#OSIdatalink"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Data link layer access to the media</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#Network\_Layer\_OSI"/><NamedIndividual IRI="#OSInetwork"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Network layer address the best path</Literal></DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#Physical\_Layer\_OSI"/> <NamedIndividual IRI="#OSIphysical"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Physical layer establish the binary transmission (wires, connectors, voltage data rates, etc)</Literal>  
</DataPropertyAssertion>

<DataPropertyAssertion><DataProperty IRI="#Presentation\_Layer\_OSI"/> <NamedIndividual IRI="#OSIpresentation"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Presentation layer represent the data (Data representation)</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#OSIreference"/> <NamedIndividual IRI="#OSIreferencemodel"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Open System Interconnection (OSI) Model consist of seven layers such as Application, Presentation, session, transport, network, data link, and physical layer</Literal> </DataPropertyAssertion>

<DataPropertyAssertion><DataProperty IRI="#Session\_Layer\_OSI"/> <NamedIndividual IRI="#OSIsession"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Session layer responsible for Inter-host communication</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#Transport\_Layer\_OSI"/> <NamedIndividual IRI="#OSItransport"/><Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Transport layer provides end-to-end connection reliability</Literal> </DataPropertyAssertion>

“<DataPropertyAssertion> <DataProperty IRI="#PAN\_Type"/> <NamedIndividual IRI="#PANnetwork"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Personal Area

Network is a network arranged within an individual person, typically within a range of 10 meters. It is used for connecting the computer devices of personal use. Personal computer devices that are used to develop the personal area network are the laptop, mobile phones, media player and play stations.</Literal></DataPropertyAssertion>”

“<DataPropertyAssertion> <DataProperty IRI="#Phising\_Threats"/> <NamedIndividual IRI="#PhishingThreats"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">An attempt to obtain sensitive information such as usernames, passwords, and credit card details by pretending as a trustworthy person.</Literal> </DataPropertyAssertion>

<DataPropertyAssertion><DataProperty IRI="#Ring\_Topology"/> <NamedIndividual IRI="#RingTopology"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Each station is directly connected only to two of its neighbors and messages sent between two stations pass through all of the stations in between</Literal></DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#Routers\_Device"/> <NamedIndividual IRI="#RoutersDevice"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Devices that are capable of communicating with similar devices over the network, collaborate among themselves and find paths for arriving data</Literal> </DataPropertyAssertion>

<DataPropertyAssertion><DataProperty IRI="#Star\_Topology"/> <NamedIndividual IRI="#StarTopology"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">A star topology is designed with each node is connected directly to a central network hub or switch.</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#Switch\_Device"/> <NamedIndividual IRI="#SwitchDevice"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">A switch receives data in one of its incoming connections and forwards the data only on the outgoing connection which connects to the destination device</Literal> </DataPropertyAssertion>

<DataPropertyAssertion><DataProperty IRI="#Application\_Layer\_TCP\_IP"/> <NamedIndividual IRI="#TCPapplication"/><Literal datatypeIRI="http://www.w3.org/2001/ XMLSchema#string">Consists of applications and processes that uses the network.</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#Internet\_Layer\_TCP\_IP"/> <NamedIndividual IRI="#TCPinternet"/><Literal datatypeIRI="http://www.w3.org/2001/ XMLSchema#string">Internet layer defines the datagram and handles the routing of data</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#Network\_Access\_Layer\_TCP\_IP"/> <NamedIndividual IRI="#TCPnetwork"/><Literal datatypeIRI="http://www.w3.org/2001/ XMLSchema#string">Network Access Layer consists of routing for accessing physical network</Literal> </DataPropertyAssertion>

<DataPropertyAssertion> <DataProperty IRI="#TCP\_Network\_Protocol"/><NamedIndividual IRI="#TCPprotocol"/> <Literal datatypeIRI="http://www.w3.org/2001/MLSchema#string">”

“Transmission Control Protocol (TCP) is a connection oriented protocol and uses sequenced acknowledgment with retransmission of packets when necessary</Literal> </DataPropertyAssertion>”

```
<DataPropertyAssertion> <DataProperty IRI="#Host_to_Host_Transport_Layer_TCP_IP"/>
<NamedIndividual IRI="#TCPtransport"/><Literal datatypeIRI="http://www.w3.org/2001/
XMLSchema#string">Host-to-Host Transport Layer provides end to end data delivery services</Literal>
</DataPropertyAssertion>
```

```
<DataPropertyAssertion><DataProperty IRI="#Trojan_Threats"/> <NamedIndividual IRI=
"#TrojanThreats"/><Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Any malicious
computer program used to invade into a computer by misleading users known as Trojan</Literal>
</DataPropertyAssertion>
```

```
<DataPropertyAssertion> <DataProperty IRI="#UDP_Network_Protocol"/><NamedIndividual
IRI="#UDPprotocol"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">User
Datagram Protocol (UDP) is simple and suitable for query based communications and it is not connection
oriented</Literal> </DataPropertyAssertion>
```

```
<DataPropertyAssertion> <DataProperty IRI="#User_awareness_Protection"/> <NamedIndividual
IRI="#UserAwarenessProtection"/> <Literal datatypeIRI="http://www.w3.org/2001/
XMLSchema#string">Computer users must be properly educated to protect the network devices against
malicious attacks and unauthorized accesses.</Literal></DataPropertyAssertion>
```

```
<DataPropertyAssertion><DataProperty IRI="#Virus_Threats"/> <NamedIndividual IRI=
"#VirusThreats"/> <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Viruses are
programs which enters into the system, run and perform malicious activities unknowingly.</Literal>
</DataPropertyAssertion>
```

```
<DataPropertyAssertion> <DataProperty IRI="#WAN_Type"/><NamedIndividual IRI=
"#WANnetwork"/><Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">A Wide Area
Network is a network that extends over a large geographical area such as states or countries. A Wide Area
Network is not limited to a single location, but it spans over a large geographical area through a telephone
line, fibre optic cable or satellite links. The internet is one of the biggest WAN in the world.</Literal>
</DataPropertyAssertion>
```

```
<ObjectPropertyDomain> <ObjectProperty IRI="#HavingDifferent"/> <Class IRI=
"#Computer_Network"/> </ObjectPropertyDomain>
```

```
<ObjectPropertyDomain> <ObjectProperty IRI="#Network_Protocol"/> <Class IRI=
"#Computer_Network"/> </ObjectPropertyDomain>
```

```
<ObjectPropertyDomain> <ObjectProperty IRI="#are"/> <Class IRI="#Gateway"/>
</ObjectPropertyDomain>”
```

```

“<ObjectPropertyDomain> <ObjectProperty IRI="#are"/> <Class IRI="#Hubs"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain> <ObjectProperty IRI="#are"/> <Class IRI="#Modem"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain> <ObjectProperty IRI="#are"/> <Class IRI="#Routers"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain> <ObjectProperty IRI="#are"/><Class IRI="#Switch"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#assign-to"/> <Class IRI="#IP_address"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain> <ObjectProperty IRI="#automatically-assign"/> <Class IRI="#DHCP"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#can-be-diffeent"/> <Class IRI="#IP_address"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain> <ObjectProperty IRI="#can_be_the"/> <Class IRI="#Antivirus"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain> <ObjectProperty IRI="#can_be_the"/> <Class IRI="#Firewalls"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#can_be_the"/> <Class IRI="#User_awareness"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#consist"/> <Class IRI="#TCP/IP_Model"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#facing"/> <Class IRI="#Computer_Network"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain> <ObjectProperty IRI="#giveConceptualFramework"/><Class IRI=
"#Reference_Model"/> </ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#has"/> <Class IRI="#OSI_Model"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#has-five"/><Class IRI="#IP_address"/>
</ObjectPropertyDomain>”

```



```

“<ObjectPropertyDomain><ObjectProperty IRI="#has-some"/><Class IRI="#Computer_Network"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#has-two"/><Class IRI="#IP_address"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain> <ObjectProperty IRI="#hasdifferent"/><Class IRI="#Computer_Network"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#help-to"/> <Class IRI="#IP_address"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#includes"/> <Class IRI="#Basic"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#is-a"/> <Class IRI="#Malware"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#is-a"/> <Class IRI="#Phising"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#is-a"/><Class IRI="#Trojan"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain> <ObjectProperty IRI="#is-a"/><Class IRI="#Virus"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#is-example-for"/><Class IRI="#OSI_Model"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#is-example-for"/><Class IRI="#TCP/IP_Model"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#is-one"/><Class IRI="#Bus"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#is-one"/><Class IRI="#Mesh"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain> <ObjectProperty IRI="#is-one"/> <Class IRI="#Ring"/>
</ObjectPropertyDomain>

<ObjectPropertyDomain><ObjectProperty IRI="#is-one"/><Class IRI="#Star"/>
</ObjectPropertyDomain>”

```

```

“<ObjectPropertyDomain> <ObjectProperty IRI="#is-the"/> <Class IRI="#Basic"/>
</ObjectPropertyDomain>

<ObjectPropertyRange><ObjectProperty IRI="#HavingDifferent"/> <Class IRI=
"#Network_Devices"/> </ObjectPropertyRange>

<ObjectPropertyRange> <ObjectProperty IRI="#Network_Protocol"/> <Class IRI="#Protocol"/>
</ObjectPropertyRange>

<ObjectPropertyRange> <ObjectProperty IRI="#are"/> <Class IRI="#Network_Devices"/>
</ObjectPropertyRange>

<ObjectPropertyRange><ObjectProperty IRI="#assign-to"/><Class IRI="#Computer_Network"/>
</ObjectPropertyRange>

<ObjectPropertyRange><ObjectProperty IRI="#automatically-assign"/><Class IRI="#IP_address"/>
</ObjectPropertyRange>

<ObjectPropertyRange> <ObjectProperty IRI="#can-be-diffeent"/> <Class IRI="#Types"/>
</ObjectPropertyRange>

<ObjectPropertyRange><ObjectProperty IRI="#can_be_the"/><Class IRI="#Protection"/>
</ObjectPropertyRange>

<ObjectPropertyRange><ObjectProperty IRI="#consist"/><Class IRI="#Application_Layer"/>
</ObjectPropertyRange>

<ObjectPropertyRange><ObjectProperty IRI="#consist"/> <Class IRI=" #Host_to_Host_
Transport_Layer"/> </ObjectPropertyRange>

<ObjectPropertyRange><ObjectProperty IRI="#consist"/> <Class IRI="#Internet_Layer"/>
</ObjectPropertyRange>

<ObjectPropertyRange> <ObjectProperty IRI="#consist"/><Class IRI="#Network_Access_Layer"/>
</ObjectPropertyRange>

<ObjectPropertyRange> <ObjectProperty IRI="#facing"/> <Class IRI="#Threats"/>
</ObjectPropertyRange>

<ObjectPropertyRange> <ObjectProperty IRI="#giveConceptualFramework"/>
<Class IRI="#Computer_Network"/></ObjectPropertyRange>

<ObjectPropertyRange><ObjectProperty IRI="#has"/><Class IRI="#Application_layer"/>
</ObjectPropertyRange>”

```

```

“<ObjectPropertyRange> <ObjectProperty IRI="#has"/><Class IRI="#Data_link_Layer"/>
</ObjectPropertyRange>

<ObjectPropertyRange> <ObjectProperty IRI="#has"/><Class IRI="#Network_Layer"/>
</ObjectPropertyRange>

<ObjectPropertyRange> <ObjectProperty IRI="#has"/><Class IRI="#Physical_Layer"/>
</ObjectPropertyRange>

<ObjectPropertyRange> <ObjectProperty IRI="#has"/> <Class IRI="#Presentation_Layer"/>
</ObjectPropertyRange>

<ObjectPropertyRange> <ObjectProperty IRI="#has"/><Class IRI="#Session_Layer"/>
</ObjectPropertyRange>

<ObjectPropertyRange><ObjectProperty IRI="#has"/> <Class IRI="#Transport_Layer"/>
</ObjectPropertyRange>

<ObjectPropertyRange> <ObjectProperty IRI="#has-five"/> <Class IRI="#Classes"/>
</ObjectPropertyRange>

<ObjectPropertyRange><ObjectProperty IRI="#has-some"/><Class IRI="#Basic"/>
</ObjectPropertyRange>

<ObjectPropertyRange><ObjectProperty IRI="#has-two"/> <Class IRI="#Versions"/>
</ObjectPropertyRange>

<ObjectPropertyRange><ObjectProperty IRI="#hasdifferent"/> <Class IRI="#Topologies"/>
</ObjectPropertyRange>

<ObjectPropertyRange><ObjectProperty IRI="#help-to"/> <Class IRI="#Routing"/>
</ObjectPropertyRange>

<ObjectPropertyRange><ObjectProperty IRI="#includes"/><Class IRI="#Purpose"/>
</ObjectPropertyRange>

<ObjectPropertyRange> <ObjectProperty IRI="#is-a"/> <Class IRI="#Threats"/>
</ObjectPropertyRange>

<ObjectPropertyRange> <ObjectProperty IRI="#is-example-for"/><Class IRI="#Reference_Model"/>
</ObjectPropertyRange>

<ObjectPropertyRange> <ObjectProperty IRI="#is-one"/> <Class IRI="#Topologies"/>
</ObjectPropertyRange>”

```

```

“<ObjectPropertyRange> <ObjectProperty IRI="#is-the"/>    <Class IRI="#Definition"/>
</ObjectPropertyRange>

<DataPropertyDomain> <DataProperty IRI="#Antivirus_Protection"/> <Class IRI="#Antivirus"/>
</DataPropertyDomain>

<DataPropertyDomain>    <DataProperty IRI="#Application_Layer_OSI"/><Class IRI=
"#Application_layer"/> </DataPropertyDomain>

<DataPropertyDomain>    <DataProperty IRI="#Application_Layer_TCP_IP"/>
<Class IRI="#Application_Layer"/> </DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Bus_Topology"/>    <Class IRI="#Bus"/>
</DataPropertyDomain>

<DataPropertyDomain>    <DataProperty IRI="#Computer_Network_Definition"/>
    <Class IRI="#Definition"/> </DataPropertyDomain>

<DataPropertyDomain>    <DataProperty IRI="#Computer_Network_Purpose"/>
    <Class IRI="#Purpose"/> </DataPropertyDomain>

<DataPropertyDomain>    <DataProperty IRI="#Data_Link_Layer_OSI"/>
<Class IRI="#Data_link_Layer"/> </DataPropertyDomain>

    <DataPropertyDomain> <DataProperty IRI="#Dynamic_Host_Configuration_Protocol-DHCP"/>
    <Class IRI="#DHCP"/> </DataPropertyDomain>

<DataPropertyDomain>    <DataProperty IRI="#Firewalls_Protection"/> <Class IRI="#Firewalls"/>
</DataPropertyDomain>

<DataPropertyDomain>    <DataProperty IRI="#Gateway_Device"/>    <Class IRI="#Gateway"/>
</DataPropertyDomain>

<DataPropertyDomain>    <DataProperty IRI="#Host_to_Host_Transport_Layer_TCP_IP"/>
<Class IRI="#Host_to_Host_Transport_Layer"/> </DataPropertyDomain>

<DataPropertyDomain>    <DataProperty IRI="#Hubs_Device"/>    <Class IRI="#Hubs"/>
</DataPropertyDomain>

<DataPropertyDomain>    <DataProperty IRI="#IP_Classes"/>    <Class IRI="#Classes"/>
</DataPropertyDomain>

<DataPropertyDomain><DataProperty IRI="#IP_Routing"/>    <Class IRI="#Routing"/>
</DataPropertyDomain>”

```

```

“ <DataPropertyDomain> <DataProperty IRI="#IP_Types"/> <Class IRI="#Types"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#IP_Versions"/> <Class IRI="#Versions"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#IPdefinition"/> <Class IRI="#Definition"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Internet_Layer_TCP_IP"/> <Class
IRI="#Internet_Layer"/> </DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#LAN_Type"/> <Class IRI="#LAN"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#MAN_Type"/> <Class IRI="#MAN"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Malware_Threats"/> <Class IRI="#Malware"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Mesh_Topology"/> <Class IRI="#Mesh"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Modem_Device"/> <Class IRI="#Modem"/>
</DataPropertyDomain>

<DataPropertyDomain><DataProperty IRI="#NetworkProtocol"/> <Class IRI="#Protocol"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Network_Access_Layer_TCP_IP"/> <Class IRI=
"#Network_Access_Layer"/> </DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Network_Layer_OSI"/> <Class IRI=
"#Network_Layer"/> </DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Network_Topology"/> <Class IRI="#Topologies"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Network_Types"/> <Class IRI="#NetworkTypes"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#OSIreference"/> <Class IRI="#OSI_Model"/>
</DataPropertyDomain>”

```

```

“<DataPropertyDomain> <DataProperty IRI="#PAN_Type"/> <Class IRI="#PAN"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Phising_Threats"/> <Class IRI="#Phising"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Physical_Layer_OSI"/> <Class IRI=
"#Physical_Layer"/> </DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Presentation_Layer_OSI"/> <Class IRI=
"#Presentation_Layer"/> </DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#ReferenceModels"/> <Class IRI=
"#Reference_Model"/> </DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Ring_Topology"/> <Class IRI="#Ring"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Routers_Device"/> <Class IRI="#Routers"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Session_Layer_OSI"/> <Class IRI=
"#Session_Layer"/> </DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Star_Topology"/> <Class IRI="#Star"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Switch_Device"/> <Class IRI="#Switch"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#TCP_Network_Protocol"/> <Class IRI="#TCP"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Transport_Layer_OSI"/> <Class
IRI="#Transport_Layer"/> </DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#Trojan_Threats"/> <Class IRI="#Trojan"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#UDP_Network_Protocol"/> <Class IRI="#UDP"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#User_awareness_Protection"/> <Class
IRI="#User_awareness"/> </DataPropertyDomain>”

```

```

“<DataPropertyDomain> <DataProperty IRI="#Virus_Threats"/> <Class IRI="#Virus"/>
</DataPropertyDomain>

<DataPropertyDomain> <DataProperty IRI="#WAN_Type"/> <Class IRI="#WAN"/>
</DataPropertyDomain>

<DataPropertyRange> <DataProperty IRI="#Antivirus_Protection"/> <Datatype abbreviatedIRI=
"xsd:string"/></DataPropertyRange>

<DataPropertyRange><DataProperty IRI="#Application_Layer_OSI"/> <Datatype abbreviatedIRI=
"xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Application_Layer_TCP_IP"/> <Datatype
abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Bus_Topology"/> <Datatype abbreviatedIRI=
"xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Computer_Network_Definition"/> <Datatype
abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Computer_Network_Purpose"/> <DatatypeabbreviatedIRI
= "xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Data_Link_Layer_OSI"/> <Datatype abbreviatedIRI=
"xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Dynamic_Host_Configuration_Protocol-DHCP"/>
<Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Firewalls_Protection"/> <Datatype abbreviatedIRI=
"xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Gateway_Device"/> <Datatype abbreviatedIRI=
"xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Host_to_Host_Transport_Layer_TCP_IP"/>
<Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Hubs_Device"/> <Datatype abbreviatedIRI=
"xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#IP_Classes"/> <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>”

```

```

“<DataPropertyRange> <DataProperty IRI="#IP_Routing"/> Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#IP_Types"/> <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#IP_Versions"/> <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#IPdefinition"/> <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Internet_Layer_TCP_IP"/><Datatype abbreviatedIRI=
"xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#LAN_Type"/> <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#MAN_Type"/> <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Malware_Threats"/> <Datatype abbreviatedIRI=
"xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Mesh_Topology"/> <Datatype abbreviatedIRI=
"xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Modem_Device"/> <Datatype abbreviatedIRI=
"xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#NetworkProtocol"/> <Datatype abbreviatedIRI=
"xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Network_Access_Layer_TCP_IP"/> <Datatype
abbreviatedIRI= "xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Network_Layer_OSI"/> <Datatype abbreviatedIRI=
"xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Network_Topology"/> <Datatype abbreviatedIRI=
"xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Network_Types"/> <Datatype abbreviatedIRI=
"xsd:string"/> </DataPropertyRange>”

```



“<DataPropertyRange> <DataProperty IRI="#OSIreference"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#PAN\_Type"/> <Datatype abbreviatedIRI="xsd:string"/>  
</DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Phising\_Threats"/> <Datatype abbreviatedIRI="xsd:string"/></DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Physical\_Layer\_OSI"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Presentation\_Layer\_OSI"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#ReferenceModels"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Ring\_Topology"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Routers\_Device"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Session\_Layer\_OSI"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Star\_Topology"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Switch\_Device"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#TCP\_Network\_Protocol"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Transport\_Layer\_OSI"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Trojan\_Threats"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#UDP\_Network\_Protocol"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>”

“<DataPropertyRange> <DataProperty IRI="#User\_awareness\_Protection"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#Virus\_Threats"/> <Datatype abbreviated IRI="xsd:string"/> </DataPropertyRange>

<DataPropertyRange> <DataProperty IRI="#WAN\_Type"/> <Datatype abbreviatedIRI="xsd:string"/> </DataPropertyRange>

<AnnotationAssertion><AnnotationProperty abbreviatedIRI="rdfs:label"/>  
<IRI>#HubsDevices</IRI><Literal datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">HubsDevice</Literal> </AnnotationAssertion>

<AnnotationAssertion> <AnnotationProperty abbreviatedIRI="rdfs:label"/> <IRI>#are</IRI>  
<Literal xml:lang="en" datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">aresome</Literal> </AnnotationAssertion>

<AnnotationAssertion> <AnnotationProperty abbreviatedIRI="rdfs:label"/> <IRI>#assign-to</IRI>  
<Literal xml:lang="en" datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">assignto</Literal> </AnnotationAssertion>

<AnnotationAssertion> <AnnotationProperty abbreviatedIRI="rdfs:label"/> <IRI>#can-be-diffeent</IRI>  
<Literal xml:lang="en" datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">different</Literal> </AnnotationAssertion>

<AnnotationAssertion> <AnnotationProperty abbreviatedIRI="rdfs:label"/> <IRI> #can\_be\_the</IRI>  
<Literal xml:lang="en" datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">ProtectionTechnique</Literal> </AnnotationAssertion>

<AnnotationAssertion> <AnnotationProperty abbreviatedIRI="rdfs:label"/>  
<IRI>#consist</IRI> <Literal xml:lang="en" datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">consist-of-four</Literal> </AnnotationAssertion>

<AnnotationAssertion> <AnnotationProperty abbreviatedIRI="rdfs:label"/> <IRI>#has</IRI>  
<Literal xml:lang="en" datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">HasSevenLayers</Literal> </AnnotationAssertion>

<AnnotationAssertion> <AnnotationProperty abbreviatedIRI="rdfs:label"/>  
<IRI>#has-five</IRI> <Literal xml:lang="en" datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">HasFive</Literal> </AnnotationAssertion>

<AnnotationAssertion> <AnnotationProperty abbreviatedIRI="rdfs:label"/>  
<IRI>#is-a</IRI> <Literal xml:lang="en" datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">kindof</Literal> </AnnotationAssertion>”

```
“ <AnnotationAssertion> <AnnotationProperty abbreviatedIRI="rdfs:label"/>
  <IRI>#is-example-for</IRI> <Literal xml:lang="en" datatypeIRI="http://www.w3.org/1999/02/22-rdf-
  syntax-ns#PlainLiteral">oneexamplefor</Literal> </AnnotationAssertion>

  <AnnotationAssertion> <AnnotationProperty abbreviatedIRI="rdfs:label"/> <IRI>#is-the</IRI>
  Literal xml:lang="en" datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-
  ns#PlainLiteral">networkdefinition</Literal> </AnnotationAssertion>
</Ontology>

<!-- Generated by the OWL API (version 4.2.5.20160517-0735) https://github.com/owlcs/owlapi -->”
```

# Bibliography

- [1] D. Kotkov, J. Veijalainen, and S. Wang, “How does serendipity affect diversity in recommender systems? A serendipity-oriented greedy algorithm,” *Computing*, vol. 102, no. 2, pp. 393–411, 2020, doi: 10.1007/s00607-018-0687-5.
- [2] W. Franco *et al.*, “Ontology-based question answering systems over knowledge bases: A survey,” *ICEIS 2020 - Proc. 22nd Int. Conf. Enterp. Inf. Syst.*, vol. 1, no. April, pp. 532–539, 2020, doi: 10.5220/0009392205320539.
- [3] M. Latifi, H. Rodríguez, and M. Sanchez-Marre, “ScoQAS: A semantic-based closed and open domain question answering system,” *Proces. Leng. Nat.*, vol. 59, pp. 73–80, 2017.
- [4] S. M. and P. C. R. M., Athira P., “Architecture of an Ontology-Based Domain- Specific Natural Language Question Answering System - india-Notes.pdf.” *International Journal of Web & Semantic Technology (IJWesT) Vol.4, No.4, October 2013, p. 9, 2013, [Online]. Available: <https://docplayer.net/298326-Architecture-of-an-ontology-based-domain-specific-natural-language-question-answering-system.html>.*
- [5] R. Mihalcea, H. Liu, and H. Lieberman, “NLP (Natural Language Processing) for NLP (Natural Language Programming),” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3878 LNCS, pp. 319–330, 2006, doi: 10.1007/11671299\_34.
- [6] G. Gangwal, “Question Answering System using Open Source Software Question Answering System using Open Source Software A Writing Project Presented to The Faculty of the Department of Computer Science San Jose State University of Requirements for the Degree Master of S,” 2012.
- [7] S. P. Lende and M. M. Raghuvanshi, “Question answering system on education acts using NLP techniques,” *IEEE WCTFTR 2016 - Proc. 2016 World Conf. Futur. Trends Res. Innov. Soc. Welf.*, 2016, doi: 10.1109/STARTUP.2016.7583963.
- [8] P. Gupta and V. Gupta, “A Survey of Text Question Answering Techniques,” *Int. J. Comput. Appl.*, vol. 53, no. 4, pp. 1–8, 2012, doi: 10.5120/8406-2030.
- [9] L. Synopsis, “Natural Language Processing : part 1 of lecture notes,” *Language (Baltim.)*, pp. 1–34, 2003.
- [10] R. A. Yunmar and I. Wayan Wiprayoga Wisesa, “Design of Ontology-based Question Answering System for Incompleted Sentence Problem,” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 258, no. 1, 2019, doi: 10.1088/1755-1315/258/1/012032.
- [11] H. Q. Vo, T.-H. Phung, and N. Q. Ly, “VQASTO: Visual Question Answering System for Action Surveillance based on Task Ontology,” pp. 273–279, 2021, doi: 10.1109/nics51282.2020.9335891.
- [12] H. Al-Chalabi, S. Ray, and K. Shaalan, “Semantic based query expansion for Arabic question answering systems,” *Proc. - 1st Int. Conf. Arab. Comput. Linguist. Adv. Arab. Comput. Linguist. ACLing 2015*, no. March 2016, pp. 127–132, 2016, doi: 10.1109/ACLing.2015.25.

- [13] S. Poonam, Tanwar and Rachel, Micheal and Shivani, "The Question Answering System Using NLP and AI," *Int. J. Sci. Eng. Res.*, vol. 7, no. 12, pp. 55–60, 2016.
- [14] S. Jayalakshmi and A. Sheshasaayee, "Automated question answering system using ontology and semantic role," *IEEE Int. Conf. Innov. Mech. Ind. Appl. ICIMIA 2017 - Proc.*, no. Icimia, pp. 528–532, 2017, doi: 10.1109/ICIMIA.2017.7975515.
- [15] W. A. Elnozahy, G. A. El Khayat, L. Cheniti-Belcadhi, and B. Said, "Question Answering System to Support University Students' Orientation, Recruitment and Retention," *Procedia Comput. Sci.*, vol. 164, pp. 56–63, 2019, doi: 10.1016/j.procs.2019.12.154.
- [16] W. Franco *et al.*, "Ontology-based question answering systems over knowledge bases: A survey," *ICEIS 2020 - Proc. 22nd Int. Conf. Enterp. Inf. Syst.*, vol. 1, no. Iceis, pp. 532–539, 2020, doi: 10.5220/0009392205320539.
- [17] R. K. Harpreet Kaur, "Novel Scoring System for Identify Accurate Answers for Factoid Questions," *Int. J. Sci. Res.*, vol. 2, no. 9, pp. 294–297, 2013, [Online]. Available: <https://www.ijsr.net/archive/v2i9/MTUwOTEzMDM=.pdf>.
- [18] A. C. O. Reddy and K. Madhavi, "A Survey on Types of Question Answering System," vol. 19, no. 6, pp. 19–23, 2017, doi: 10.9790/0661-1906041923.
- [19] P. A. E. A. Karunarathne, C. M. W. Arachchi, G. C. Perera, W. A. M. N. Bandara, and G. U. Ganegoda, "Ontology Based Answer Extraction for Customer Support in Transportation Service," no. December, pp. 67–72, 2019.
- [20] S. Palaniappan, U. K. Sridevi, and J. Subburaj, "Ontology based Question Answering system using JSON-LD for Closed Domain," vol. 119, no. 12, pp. 1969–1980, 2018.
- [21] V. Jain and M. Singh, "Ontology Development and Query Retrieval using Protégé Tool," *Int. J. Intell. Syst. Appl.*, vol. 5, no. 9, pp. 67–75, 2013, doi: 10.5815/ijisa.2013.09.08.
- [22] L. Jean-Baptiste, *Ontologies with Python*. 2021.
- [23] J. B. Lamy, "Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies," *Artif. Intell. Med.*, vol. 80, no. September, pp. 11–28, 2017, doi: 10.1016/j.artmed.2017.07.002.
- [24] A. W. AbuTaha, "An Ontology-Based Arabic Question Answering System," 2015, [Online]. Available: [https://www.mendeley.com/catalogue/ed7d5c7e-123c-378e-92b0-33f63c321836/?utm\\_source=desktop](https://www.mendeley.com/catalogue/ed7d5c7e-123c-378e-92b0-33f63c321836/?utm_source=desktop).
- [25] D. L. McGuinness and F. van Harmelen, "OWL Web Ontology Language Overview," *W3C Recomm.*, vol. 10, no. February, 2004, [Online]. Available: <http://www.w3.org/TR/owl-features/>.
- [26] M. Vargas-Vera, E. Motta, and J. Domingue, "AQUA: An Ontology-Driven Question Answering System," ...*Directions Quest. Answering*, 2003, [Online]. Available: <http://www.aaai.org/Papers/Symposia/Spring/2003/SS-03-07/SS03-07-009.pdf>.
- [27] J. A. Khan and S. Kumar, "Deep analysis for development of RDF, RDFS and OWL ontologies with protege," *Proc. - 2014 3rd Int. Conf. Reliab. Infocom Technol. Optim. Trends Futur. Dir. ICRITO 2014*, pp. 6–11, 2015, doi: 10.1109/ICRITO.2014.7014747.

- [28] K. Raju, "DOMAIN SPECIFIC ONTOLOGY BASED," vol. 1, no. 5, pp. 1–5, 2016, [Online]. Available: <https://1library.net/document/q75r4vvz-domain-specific-ontology-question-answering-kavitha-abstract-ijrmet.html>.
- [29] M. Rajosoa, R. Hantach, S. Ben Abbes, and P. Calvez, "Hybrid question answering system based on natural language processing and SPARQL query," *CEUR Workshop Proc.*, vol. 2487, pp. 94–102, 2019.
- [30] V. Mallamma and M. Hanumanthappa, "Semantical and Syntactical Analysis of NLP," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 3, pp. 3236–3238, 2014.
- [31] B. Hssina, B. Bouikhalene, and A. Merbouha, "Europe and MENA Cooperation Advances in Information and Communication Technologies," vol. 520, no. June 2020, pp. 103–112, 2017, doi: 10.1007/978-3-319-46568-5.
- [32] V. Mishra and N. Khilwani, "QUASE: AN Ontology-Based Domain Specific Natural Language Question Answering System," *Int. J. Recent Technol. Eng.*, vol. 8, no. 4, pp. 261–268, 2019, doi: 10.35940/ijrte.d6773.118419.
- [33] M. Devi and M. Dua, "ADANS: An agriculture domain question answering system using ontologies," *Proceeding - IEEE Int. Conf. Comput. Commun. Autom. ICCCA 2017*, vol. 2017-Janua, pp. 122–127, 2017, doi: 10.1109/CCAA.2017.8229784.
- [34] M. Rahim, Z. Turabee, Q. Rajput, and S. A. Khoja, "Semantic Based Question Answering System on Travel Ontology," *2019 6th Int. Conf. Soc. Networks Anal. Manag. Secur. SNAMS 2019*, pp. 67–74, 2019, doi: 10.1109/SNAMS.2019.8931886.
- [35] T. Alsubait, B. Parsia, and U. Sattler, "Generating multiple choice questions from ontologies: How far can we go?," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8982, pp. 66–79, 2015, doi: 10.1007/978-3-319-17966-7\_7.
- [36] M. Ramos, M. J. V. Pereira, and P. R. Henriques, "A QA System for learning Python," *Communiation Pap. 2017 Fed. Conf. Comput. Sci. Inf. Syst.*, vol. 13, pp. 157–164, 2017, doi: 10.15439/2017f157.
- [37] L. Zemmouchi-Ghomari and A. R. Ghomari, "Process of building reference ontology for higher education," *Lect. Notes Eng. Comput. Sci.*, vol. 3 LNECS, no. July 2013, pp. 1595–1600, 2013.
- [38] A. S. Omarbekova, G. S. Nurgazinova, A. A. Sharipbay, A. Barlybayev, and G. T. Bekmanova, "Automatic formation of questions and answers on the basis of the knowledge base," *Proc. 2017 Int. Conf. Eng. Technol. ICET 2017*, vol. 2018-Janua, pp. 1–3, 2018, doi: 10.1109/ICEngTechnol.2017.8308192.
- [39] K. Hwerbi, "An ontology-based chatbot for crises management: use case coronavirus," 2020, [Online]. Available: <http://arxiv.org/abs/2011.02340>.
- [40] V. Lopez, V. Uren, E. Motta, and M. Pasin, "AquaLog: An ontology-driven question answering system for organizational semantic intranets," *Web Semant.*, vol. 5, no. 2, pp. 72–105, 2007, doi: 10.1016/j.websem.2007.03.003.
- [41] A. De Leon Battista, N. Villanueva-Rosales, M. Palenychka, and M. Dumontier, "SMART: A web-based, ontology-driven, semantic web query answering application,"

- CEUR Workshop Proc.*, vol. 295, no. January 2014, 2007.
- [42] V. Lopez, E. Motta, M. Sabou, and M. Fernandez, “PowerAqua: A multi-ontology based question answering system–v1,” *OpenKnowledge Deliv. D*, vol. 8, p. 4, 2007, [Online]. Available: [http://technologies.kmi.open.ac.uk/poweraqua/okdeliverable/OK\\_D8.4\\_PowerAqua\\_vfinal.pdf](http://technologies.kmi.open.ac.uk/poweraqua/okdeliverable/OK_D8.4_PowerAqua_vfinal.pdf).
- [43] A. Moussa and R. Abdel-Kader, “QASYO: A Question Answering System for YAGO Ontology,” *Int. J. Database Theory Appl.*, vol. 4, no. 2, pp. 99–112, 2011, [Online]. Available: [http://www.sersc.org/journals/IJDTA/vol4\\_no2/9.pdf](http://www.sersc.org/journals/IJDTA/vol4_no2/9.pdf).
- [44] W. M. D. Weerabahu, M. A. G. A. S. Amalika, H. K. C. Dulakshi, and T. C. Sandanayake, “Ontology-Based Question Answering System for Banking Domain,” no. Reiter 1995, pp. 205–211, 2007.
- [45] N. Kaur, “Semantic Information Retrieval using String Ontology in Music Domain using Protege5 . 0,” pp. 47–52.
- [46] Y. T. Premathilaka, K. Banujan, and B. T. G. S. Kumara, “Ontology-Based Approach to Determine the Coverage of Examination Papers,” in *2020 International Conference on Decision Aid Sciences and Application, DASA 2020*, 2020, pp. 613–617, doi: 10.1109/DASA51403.2020.9317235.
- [47] P. Visutsak, “Ontology-Based Semantic Retrieval for Durian Pests and Diseases Control System,” vol. 11, no. 1, 2021, doi: 10.18178/ijmlc.2021.11.1.1019.
- [48] J. Xu and W. Zhang, “Study of Intelligent Question Answering System Based on Ontology.”
- [49] L. Prananingrum, “Web Based Ontology Implementation for Information Search System,” pp. 499–504, 2019.
- [50] S. Lee, P. Ryu, and K. Choi, “Ontology-based Question Answering System QA Processing Unit : Triplet,” *6th Int. Semant. Web Conf.*, 2007.
- [51] W. Ahmed and B. Anto, “an Automatic Web-Based Question Answering System for E-Learning,” *Inf. Technol. Learn. Tools*, vol. 58, no. 2, p. 1, 2017, doi: 10.33407/itlt.v58i2.1567.
- [52] G. Ashour, A. Al-Dubai, and I. Romdhani, “Ontology-based course teacher assignment within universities,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 7, pp. 720–728, 2020, doi: 10.14569/IJACSA.2020.0110787.
- [53] “Managing ontologies — Owlready2 0.34 documentation” <https://owlready2.readthedocs.io/en/latest/onto.html#> (accessed Sep. 10, 2021)
- [54] “How to Build a Question Answering System: a Step-by-Step Tutorial.” <https://intersog.com/blog/the-basics-of-qa-systems-from-a-single-function-to-a-pre-trained-nlp-model-using-python/> (accessed Sep. 15, 2021).

- [55] “Step 1. Create and run your first Python project | PyCharm.”  
<https://www.jetbrains.com/help/pycharm/creating-and-running-your-first-python-project.html#edit-file> (accessed Oct. 11, 2021).
- [56] “jena - How to define very simple own rule in fuseki step by step? - Stack Overflow.”  
<https://stackoverflow.com/questions/30217704/how-to-define-very-simple-own-rule-in-fuseki-step-by-step> (accessed Oct. 11, 2021).
- [57] “Gunicorn - Python WSGI HTTP Server for UNIX.” <https://gunicorn.org/> (accessed Oct. 11, 2021).
- [58] “Building a Question-Answering System from Scratch— Part 1 | by Alvira Swalin | Towards Data Science.” <https://towardsdatascience.com/building-a-question-answering-system-part-1-9388aadff507> (accessed Oct. 12, 2021).
- [59] “unicodedata — Unicode Database — Python 3.10.0 documentation.”  
<https://docs.python.org/3/library/unicodedata.html> (accessed Oct. 21, 2021).
- [60] “What is gevent? — gevent 21.8.1.dev0 documentation.” <http://www.gevent.org/> (accessed Oct. 21, 2021).
- [61] “WTForms — WTForms Documentation (2.3.x).” <https://wtforms.readthedocs.io/en/2.3.x/> (accessed Oct. 21, 2021).
- [62] D. Mukhopadhyay, A. Banik, S. Mukherjee, J. Bhattacharya, and Y.-C. Kim, “A Domain Specific Ontology Based Semantic Web Search Engine,” 2011, [Online]. Available: <http://arxiv.org/abs/1102.0695>.
- [63] M. Jabalameli, M. Nematbakhsh, and A. Zaeri, “Ontology-lexicon-based question answering over linked data,” *ETRI J.*, vol. 42, no. 2, pp. 239–246, 2020, doi: 10.4218/etrij.2018-0312.
- [64] D. Moldovan, M. Paşca, S. Harabagiu, and M. Surdeanu, “Performance issues and error analysis in an open-domain question answering system,” *ACM Trans. Inf. Syst.*, vol. 21, no. 2, pp. 133–154, 2003, doi: 10.1145/763693.763694.
- [65] F. Cujar-Rosero, D. S. Pinchao Ortiz, S. R. Timaran Pereira, and J. M. Guerrero Restrepo, “Fenix: A Semantic Search Engine Based on an Ontology and a Model Trained with Machine Learning to Support Research,” pp. 97–115, 2021, doi: 10.5121/csit.2021.110709.
- [66] G. Besbes, H. Baazaoui-Zghal, and H. Ben Ghezela, “An ontology-driven visual question-answering framework,” *Proc. Int. Conf. Inf. Vis.*, vol. 2015-Sept, pp. 127–132, 2015, doi: 10.1109/iV.2015.32.
- [67] A. Ayadi, A. Samet, F. D. B. De Beuvron, and C. Zanni-Merk, “Ontology population with deep learning-based NLP: A case study on the Biomolecular Network Ontology,” *Procedia Comput. Sci.*, vol. 159, pp. 572–581, 2019, doi: 10.1016/j.procs.2019.09.212.