Dissertation on

Study on Implementing Search Techniques on Lexical Resources

Thesis submitted towards partial fulfilment of the requirements
for the degree of

Master of Technology in IT (Courseware Engineering)

Submitted by

Bidisa Pal

EXAMINATION ROLL NO.: M4CWE23007

UNIVERSITY REGISTRATION NO.: 147616 of 2018-19

Under the guidance of

Prof. Dr. Matangini Chattopadhyay

School of Education Technology
Jadavpur University

Course affiliated to
Faculty of Engineering and Technology

Jadavpur University

Kolkata – 700032

India

2023

M.Tech IT (Courseware Engineering)
Course affiliated to
Faculty of Engineering and Technology
Jadavpur University
Kolkata, India

CERTIFICATE OF RECOMMENDATION

This is certified that the thesis entitled "**Study on Implementing Search Techniques on Lexical Resources"** is a bonafide work carried out by **BIDISA PAL** under our supervision and guidance for partial fulfillment of the requirements for the degree of Master of Technology in IT (Courseware Engineering) in School of Education Technology, during the academic session 2022-2023.

_____

SUPERVISOR
School of Education Technology
Jadavpur University,
Kolkata - 700 032

_____

DIRECTOR
School of Education Technology
Jadavpur University,
Kolkata - 700 032

_____

DEAN - FISLM
Jadavpur University,
Kolkata - 700 032

M.Tech IT (Courseware Engineering)
Course affiliated to
Faculty of Engineering and Technology
Jadavpur University
Kolkata, India

CERTIFICATE OF APPROVAL **

This foregoing thesis is hereby approved as a credible study of an engineering subject carried out and presented in a manner satisfactory to warranty its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not endorse or approve any statement made or opinion expressed or conclusion drawn therein but approve the thesis only for purpose for which it has been submitted.

Committee of final examination
for evaluation of Thesis

** Only in case the thesis is approved.

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this thesis contains literature survey and original research work by the undersigned candidate, as part of her **Master of Technology in IT (Courseware Engineering)** studies.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by this rule and conduct, I have fully cited and referenced all materials and results that are not original to this work.

NAME: Bidisa Pal

EXAMINATION ROLL NO: M4CWE23007

THESIS TITLE: STUDY ON IMPLEMENTING SEARCH TECHNIQUES ON LEXICAL RESOURCES

SIGNATURE: **DATE**:

Acknowledgment

I express my sincere gratitude to my supervisor **Prof. Dr. Matangini Chattopadhyay** for her continuous support and help in my thesis work and related research. I am highly indebted to madam for her guidance and constant supervision as well as her valuable time towards necessary discussions, insights and overall viewpoints which went far beyond the narrow domain of work and helped me embark on new ideas. I could not have a better advisor or mentor for my research work other than her. I am also thankful for her constant motivation, enthusiasm, and immense knowledge.

I also thank to **Mr. Diptentu Dutta** for his all sort of help in resource, knowledge, and ideas sharing, correcting my code, and look after throughout my work in person and also over the virtual mode.
Besides, I would like to thank **Dr. Susovan Jana**, whose ideas and co-operation always help me in thinking out of the box about a problem with different perspective. These initial approaches are beneficial to me during my research whenever I encounter difficulties with a specific problem.

I must acknowledge the academic resources that I have got from Jadavpur University during my entire period of study in the School of Education Technology. I would like to thank all School of Education Technology faculty members, lab assistant, other research workers, my fellow classmates, and staff who were always available for help, ready to give suggestions when required.

Lastly, I shall remain grateful forever to my parents, family members, friends and well-wishers who always have faith in me and remain there for me with all my ups and downs. It is their constant encouragement, enthusiasm and support that have helped me throughout my academic career and especially during my project work.

Bidisa pal
Examination Roll No. M4CWE22024
Univ. Registration No. 147616 of 2018-19
M. Tech. IT (Courseware Engineering)
School of Education Technology
Jadavpur University, Kolkata – 700032

DEDICATED TO,

My Parents

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF ABBEVIATIONS

API: Application Programming Interface

LKE: LearnITy Knowledge Engine

DFD: Data Flow Diagram

DOM: Document Object Model

HTML: Hyper Text Markup Language

IDE: Integrated Development Environment

JSON: Java Script Object Notation

RAM: Random Access Memory

SDK: Software Development Kit

SQL: Structured Query Language

UI: User Interface

URL: Uniform Resource Locator

IRI: International Resource Identifiers

OWL: Web Ontology Language

RDF: Resource Description Framework

JWI: MIT Java Wordnet Interface

EER: Enhanced entity-relationship

NLP: Natural Language Processing

EXECUTIVE SUMMARY

The insight of the Semantic Web introduced ontologies as the main integrating tool for the management of the knowledge and semantic structure of various documents. This research work describes the hurdle to combining freely available online materials to build inclusive lexical resources. It also provides a useful user interface and RESTful API to access and manipulate the lexical resource using the proposed methodology.

The design and implementation of the LKE (LearnITy Knowledge Engine) system capability modules focus on recognizing and analyzing the most suitable and easy approach for extracting and processing different multilingual WordNet (also used as ontology) and ontology files. To extract relevant information such as synonyms, hyponyms, hypernyms, classes, and properties, from these files, we can utilize various API's such as MIT Java Wordnet Interface (JWI), WOLF, and Web Ontology Language (OWL). This extracted knowledge can then be linked and utilized for testing purposes in the search interface. The data obtained after extraction is stored in separate tables, creating its own LKEApplication database. This database is used to store the extracted files and data from WordNet and Ontologies.

This research involves several steps, which includes uploading various wordnet and ontology files (such as PWN, DebVisDic, XML-based, .owl, .rdf), extracting data, parsing the data using different APIs, and analyzing the results. This analysis is highly beneficial for text and word analysis, aiding in the development of the proposed ontology search tool. It can be efficiently exploitable in different artificial intelligence applications.

Besides all the mentioned features, this ontology extraction algorithm is created keeping in mind relatable possible usage scenarios of the ontologies for understanding and program analysis.

CHAPTER 1

1.0 INTRODUCTION

1.1 Overview

Experts often use existing words from the English Thesaurus in their lexical research. WordNet is a lexical asset at the core of many advanced Natural Language Processing (NLP) applications. WordNet has a complex concept regarding specific semantic domain knowledge.

Besides, Information Technology has seen an increase in the use of services, such as storing information online and accessing it through services in recent years. With this approach, the computational effort associated with performing operations on data is transparent to the programmer.

The programmer only needs to know what input to provide to the services and what output to expect. One of the most complex ways to represent data is with ontologies. From a programmer's perspective, the main challenge is linking these two approaches like, what input to provide to the services, and what output to expect from the data in the form of ontology.

1.2 Problem Statement

To design and develop a search algorithm on lexical resources such as - WordNet and Ontology based

1.3 Objectives

The objectives of my research work are as follows:

- ✓ To study existing LearnITy Knowledge Engine (LKE) application and how it works.
- ✓ Learn to set up the LKE framework inside the Eclipse IDE.
- ✓ Learn to set up the connection of the Tomcat server along with it.
- ✓ Study the lexical resources present on the internet.
- ✓ Learn to use APIs to use them for information extraction from the lexical resources.
- ✓ Learn to fetch matching search results based on the extracted information.

- ✓ Learn Java programming to develop the classes for information extraction and search result.
- ✓ Learn jQuery, XML, and JavaScript programming to develop the interfaces.

- ✓ Learn to write MySQL queries for storing extracted data in respective tables by executing them using Java programming.

1.4 Assumptions and Scope

1.4.1 Assumptions

- User must have downloaded the lexical resources for testing in LearnITy Knowledge Engine (LKE).
- Developers must have a proper system setup like Eclipse IDE, Tomcat server, and MySQL Installer already installed in PC.

1.4.2 Scopes

- To create a module that seamlessly integrates into an existing (LKE) software. Throughout this process, we have acquired in-depth knowledge on effectively utilizing their proprietary library infrastructure and APIs. This knowledge has enabled us to develop the necessary Java classes for conducting text searches.
- To learn about Java, XML, and jQuery.
- To learn Apache tomcat server connection with Eclipse, MySQL connection with Eclipse and MySQL queries.

1.5 Concept and Problem Analysis

A semantic ontology is a representation of a top-level declarative abstraction hierarchy with enough detailed information to make sense. It can also be defined as a declarative piece of knowledge that can be reused and shared. This work aims to find a feasible solution and methods for making a module LearnITy Knowledge Engine system compatible. The module extracts text and uri-based data from ontologies. It provides the user with an overview of the ontology. The user can manipulate the ontology as desired. The module allows for a search based on the extracted data. Some of the key features of the module :

- To search from WordNet- here is a searchable lexical database that links words into semantic relations and provides our users with a list of synonyms, hyponyms, and hypernyms that are grouped into synsets for easy search.
- To search from Ontology- it is a lexical resource that helps in information retrieval as well as searching using MySQL FULLTEXT search using Natural Language mode.
- Both of this sub-module are made working and accessible by using APIs such as- OWL, WOLF and many libraries like JWI, WordnetTools.

1.6 Organization of the Thesis

- Chapter 1: This chapter contains the introduction of the thesis which includes an overview, problem statement, objectives, assumptions, scopes, concept, and problem analysis.
- Chapter 2: It covers all the literature surveys done to carry out the research work.
- Chapter 3: System architecture, proposed solution, detailed description of the Implementation, UI design, classes and their interactions, and sequence diagrams are given here.
- Chapter 4: This chapter describes the experimentations and results.
- Chapter 5: Here we have described the conclusion and future work.
- References: All the references are given here.
- Appendix Part A: The Users' manual is available here.
- Appendix Part B: Here are all the code snippets provided.

CHAPTER 2

2.0 LITERATURE SURVEY

Agnieszka Konys [1] presents a knowledge repository that contained information about a group of 22 ontology learning tools and their 65 features that extract ontologies from texts using particular learning techniques and optional interoperability with other tools in summarized form. These algorithms assist in the creation of ontologies automatically or semi-automatically based on schematic structures or data levels. During the analysis of tools, the knowledge representation is based on the results of an ontology learning algorithm, including the goal and scope of the tool as well as the types of sources used. In addition, it evaluates the tools in light of the quality of an ontology (ontology evaluation task), as well as the process of extracting output (taxonomic and non-taxonomic relations, terms, axioms, and concept discovery). A competency-based reasoning mechanism are used to validate the proposed knowledge repository. An important issue of user and expert intervention are also included in the repository.

Asim, M. N., Wasim, M., Khan, M. U. G., Mahmood, W., & Abbasi, H. M [2] have described the process of ontology learning by some ontology learning techniques categorised into three classes (linguistics, statistical, and logical). They have explored highlighting the pros and cons of ontology evaluation techniques. In addition, they have discussed the scope and application of ontology learning in a variety of industries. They also have addressed ontology learning challenges and future directions.

Jiomekong, A., Camara, G., & Tchuente, M. [3] have proposed an approach based on Hidden Markov Models (HMMs) to learn concepts, properties, axioms and rules from Java source code. They have applied it to the source code of EPICAM, an epidemiological platform developed in Java and used in Cameroon to monitor tuberculosis transmission. They have also performed an evaluation of the relevance of the terms extracted to the medical domain by arranging them with ontologies hosted on the Bioportal platform using ontology recommender tool. The result is satisfactory. The terms extracted are covered at 82.9% by many biomedical ontologies (e.g., NCIT, SNOWMEDCT, ONTOPARON).

Moldovan, D., Pop, C., Antal, M., Cioara, T., Anghel, I., & Salomie, I. [4] have presented a library for exposing ontology access and manipulation as web services for the integration of ontological knowledge bases into complex software applications. They have proposed a framework that integrates reasoning techniques and creates a web services layer that performs basic CRUD operations and inferred information retrieval. They have used third-party APIs to provide access to ontologies, such as Jena API for simple ontology access or D2RQ for accessing ontologies mapped to relational databases. They have generated an object-oriented layer in order to facilitate the flow of data between the web services and data sources.

Morad Hajji, Mohammed Qbadou, & Khalifa Mansouri [5] have proposed an approach for developing a new Java API based on Java Architecture for XML Binding by utilizing the French wordnet database. They have evaluated the API performance in the context of the browser they developed to extract semantic and lexical relations between synsets in the database, such as hypernymy trees, hyponymy trees, synonym trees, etc.

Priyatno, J., & Bijaksana, M. A. [6] have presented a WordNet to assist researchers using English databases. They have focused on grouping words of the same meaning or synonym sets. Using hierarchical clustering with an existing data set, they have found the F1 Score Result is 10.68%, the Recall is 8.53%, and the Precision is 14.28%.

Chimalakonda, S., & Nori, K. V. [7] have described an ontology-based framework for systematic modeling of different aspects of instructional design knowledge, based on domain patterns. The framework includes ontologies for modeling goals, instructional processes, and instructional material. In order to illustrate the ontology framework, they have presented instances of it for a large-scale case study of adult literacy in India. Therefore, using this framework, instruction design knowledge can be modeled for school education, technical education, and beyond.

Aditya Kalyanpur, Daniel Jimenez Pastor, Steve Battle, and Julian Padget [8] have suggested a concise and elegant approach for mapping an OWL ontology into Java language. Java interfaces and classes are created from an OWL ontology such that a Java class represents an instance of a single class in the ontology with most of its properties,

class-relationships, and restriction-definitions maintained. They have also prepared a comprehensive discussion on the HarmonIA framework(a framework that is used to automatically generate agent systems based on institution specifications) or toolset, whose OWL Ontology Creation module is the basis for their tool.

Ales Horak, & Adam Rambousek [9] have developed an automated system that extracts known facts, links them with shared ontologies, and uses this knowledge to answer questions about the document's topic. They have presented the work results of the KYOTO project database implementation which combines national WordNet semantic networks with the general SUMO ontology to offer a basis for future shared ontologies.

Luca Buoncompagni, Syed Yusha Kareem, & Fulvio Mastrogiovanni [10] have designed an API called OWLOOP. Due to OWL's lack of compatibility to the OOP paradigm, such an interface often leads to boilerplate code, which affects modularity. OWLOOP addresses both this problem and the associated computational problems. They also have provided an extension of the OWL-API to deliver a general-purpose interface between OWL axioms subject to reasoning and modular OOP object hierarchies.

CHAPTER 3

3.0 PROPOSED APPROACH

3.1 System Architecture

The system architecture of the application is shown in Figure 1.

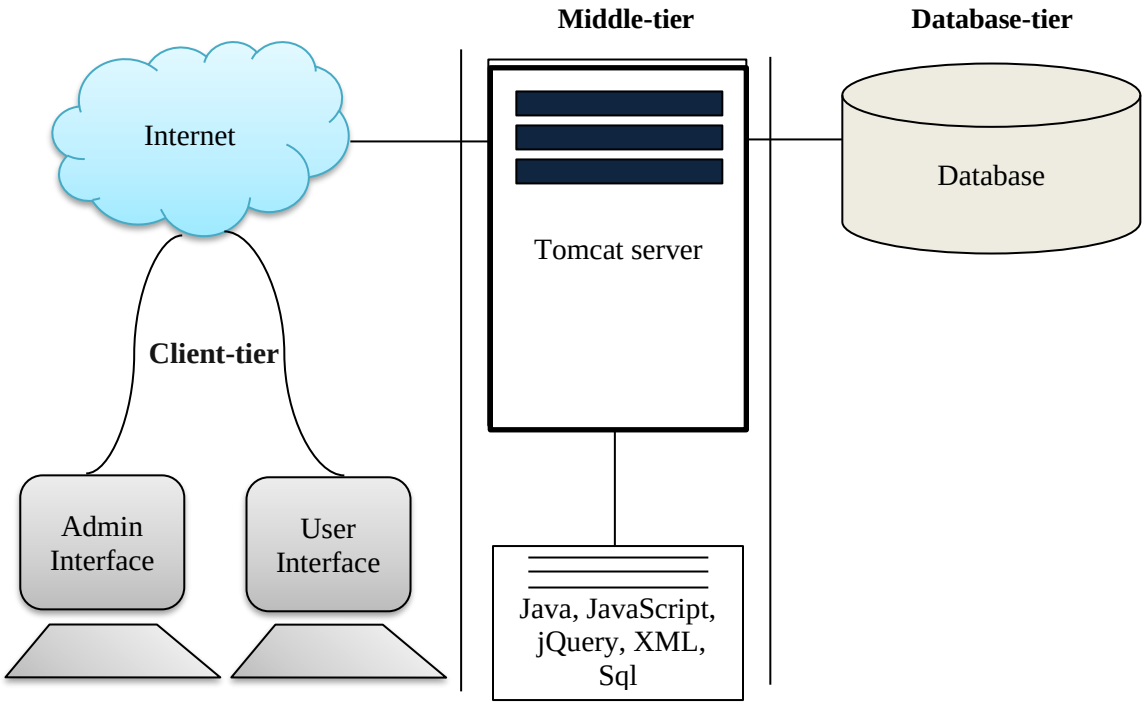

Figure 1: System Architecture of the application

The first tier of the 3-tier architecture has different types of users (e.g., Admin, clients, etc.) Users are connected to the internet and directly communicate through the middle-tier which includes the Tomcat server and it executes the code snippet and stores some result in MySQL database in the database-tier. It also fetches or receives data from the database when needed.

3.2 Proposed Solution

Intelligent document processing is the foundation of Digital Transformation, as it converts unstructured data into actionable intelligence. LearnITy Knowledge Engine (LKE) is a document understanding (document processing with the help of set of technologies using Artificial Intelligence (AI) to extract the fundamental structured information) web-based platform or tool that empowers businesses to read PDFs, Word, and Excel files, extract information, perform document validation, comparisons, categorization and search, and offer additional functionalities. We propose an idea of implementing a framework as part

of this tool, that includes a method for integrating it into LKE. This integration will enable the extraction of basic information for any type of ontology. This will give the user an overview of the ontology, they will be able to manipulate it and conduct searches based on the data according to their preferences.

Below are the modules of our framework -

- Searching from WordNet-based lexical resources –
  - ✓ It is a multilingual-based searchable lexical database that gives a list of synonyms, hyponyms, and hypernyms for a given word.
  - ✓ It has the flexibility of adding or uploading as many as different language-based wordnet and it also supports different file types (i.e., PWN, DEBVISDIC, XML).
- Searching from Ontology-based lexical resources –
  - ✓ it helps in fetching ontology details of different ontology file types along with searching based on MySQL FULLTEXT search using Natural Language mode.

3.2.1 Detailed Description of the Lexical Resource Module

In order to create this fully-featured module, the tools and services that we have used.

Table 1: Implementation Requirement for Lexical Resources

Implementation on LKE	
Programming Language	Java, XML, JavaScript, jQuery, SQL
Database	MySQL
IDE	Eclipse
Platform SDK	Latest stable Android SDK
Web hosting	LocalHost
Storage	MySQL “lkeapplication” database
Authorization	User Restricted
WordNet Definition Part Class	DBgrid, Conditionalgrid
Search	Test interface result against selected wordnet resource type and select type value option (synonym, hyponym, hypernym) from the given combo box. Ontology Search Result interface result against selected Ontology and entity type option from given

	combobox
Resource	WordNetDefinition.js,WordNetTest.js, Ontology.js, OntologyTest.js
Activity	UploadWordNet.java,WordNetTest.java, Ontology.java, OntologyTest.java
Database Activity	DataBaseLayer.java
General tools	JWI, WordnetTools, wolf API, sax parser, OWL API

After analyzing the problem statement, proposed solution, and implementation requirements, we have successfully designed and implemented the WordNet and Ontology Module. This module has been incorporated into the Lexical Resource section to enhance its functionality.

First, we visited localhost:8081/LearnITyKnowledgeEngine/coreadmin/login.html this url and logged in the LearnITy Administrator page using the right login credentials.

- Upon successful login, we proceeded to the Core Administrator dashboard.
- In the dashboard, we navigated to Interface Management.
- Our objective was to find the "LKEAdminHome" file.
- After locating it, we accessed the LKE Administration Home Page.
- Within this page, we opened the "adminmenu.js" (figure 2) file.
- We added our module and submodule, providing accurate title and URL details.
- Finally, we saved the changes to ensure they were reflected in the LKEUser interface.

```
                    {title: "Lexical Resource",isFolder: true,url:'',
                                            children: [
                                        {title: "WordNet",isFolder: true,url:'',
                                                children: [
                                                                        {title:'Definition'
url:'./PortalServlet?IID=WordNetDefinition' ,isFolder: false},
                                                                        {title:'Test',
url:'./PortalServlet?IID=WordNetTest' ,isFolder: false}
                                                                        ]},
```

Figure 2: adminmenu.js

Developers can do certain works under LearnITy Administrator. They are-

- ✓ Creating new menus.
- ✓ Adding interfaces respectively for the new added menu items.
- ✓ Updating manifest.xml (figure 3) and interfacerole.xml (figure 4) with

corresponding title, interface id, layout, content, style, and behavioral id.

- ✓ Uploading correct zip for the individual interfaces.

```
<interface id="WordNetDefinition" title="Definition" type="Interface" zip="WordNetDefinition.zip"/>
<interface id="WordNetTest" title="Test" type="Interface" zip="WordNetTest.zip"/>
<interface id="WordEmbeddingTest" title="Test" type="Interface" zip="WordEmbeddingTest.zip"/>
<interface id="Ontology" title="Load Ontology" type="Interface" zip="Ontology.zip"/>
<interface id="ImportOntology" title="Import Ontology" type="Interface" zip="ImportOntology.zip"/>
<interface id="OntologyElement" title="Ontology Element" type="Interface" zip="OntologyElement.zip"/>
```

Figure 3: manifest.xml

```
<interface id="WordNetDefinition">
    <layout id="WordNetDefinitionLayout"/>
    <content id="WordNetDefinitionContent"/>
    <style id="WordNetDefinitionStyle"/>
    <behaviour id="WordNetDefinitionBehaviour"/>
</interface>
```

Figure 4: interfacerole.xml

Developers can set their own menu module. In this research work, **WordNet** module has been considered. This module is divided into two sub-modules, they are Definition and Test.

Definition – For this, we create the "WordNetDefinition" interface first. Here, We can add new records giving mandatory details like wordnet title, file type, language, and other details like wordnet description & comment. After selecting a particular row, we can upload a Wordnet file keeping in mind it should definitely match the right file type and language already mentioned in that row while choosing the file. The Wordnet file should be in zip or Gzip form. After successful uploading wordnet table will be updated showing the wordnet file path where it has been extracted.

Test – To conduct a test, we have developed the "WordNetTest" interface. Within this interface, we can choose a particular language from WordNet (such as English or French) and input a word in the same language. By selecting a specific type (such as Synonym, Hyponym, or Hypernym), we can search for relevant results.

Working Process –

- ✓ On selection of a particular wordnet, wordnet language, and wordnet file type, a relative library or API will be called for parsing the wordnet and to retrieve search results against given input.

- ✓ Calling of the library or API will be decided based on the selected wordnet file type. Such as-
  - For WNDB or binary file format,
    - The MIT JWI library will be used to determine the case as "PWN" against English Wordnet.
  - For LMF/ XML-based file format,
    - The WordnetTools software will be employed to determine the case as "XML" in relation to English Wordnet.
    - The WOLF API will be utilized to determine the case as "DebVisDic" against French Wordnet.

Next we focus on **Ontology** module**,** which is divided into eight sub-modules. They are:

- ✓ Load Ontology
- ✓ Import Ontology
- ✓ Ontology Element
- ✓ Ontology Class
- ✓ Ontology Data Property
- ✓ Ontology Object Property
- ✓ Ontology Named Individual
- ✓ Ontology Search Result

For every sub-module, we have created a separate interface.

.**Load Ontology-** We can add different ontologies here by uploading ontology files. On successful file upload, it will load basic ontology details like ontology title, ontology iri, ontology file type, ontology file, number of classes, object properties, datatype properties, and individuals.

Import Ontology- When an individual ontology is selected from the drop-down box, the "ImportOntology" interface grid will display potential results. These results include all the imported ontology ids if the imported ontology has already been saved in the database, along with their IRI and the ontology-id of the selected ontology. However, if the selected ontology did not import any other ontologies through the "getDirectImport" relation, then no results will be shown in the grid.

Ontology Element- On the selection of an individual ontology from the drop-down box, it will provide details such as ontology element type, ontology element iri, ontology element rdfID, ontology element label, ontology element comment, and ontology-id of the selected ontology on the “OntologyElement” grid.

Ontology Class- It will deliver ontology class type, and ontology element id against selected ontology.

Ontology Data Property- It will reflect details of axiom type like data property domain, range, functional data property & ontology element id from the selected ontology.

Ontology Object Property- It will reflect details of axiom type like object property domain, object property range, reflexive, symmetric, transitive, functional, and inverse functional object property & ontology element id considering selected ontology.

Ontology Named Individual- It will give ontology element id and ontology class id for selected ontology.

Ontology Search Result- The system stores search results in the "SearchResult" grid table. These results are provided for a specific input against the selected ontology and entity types. The results are only stored if the given word matches with the information stored in the ontology element table. The matching is done based on a specific selected column.

Working Process –

- ✓ Ontology loading and extracting ontology element details from it is done by using the OWL API.
- ✓ The extracted information from the ontology element is used to load relevant details. This information is received down the line. It helps in loading applicable details for all the mentioned entity types.
- ✓ Based on the ontology element table field details, an Ontology search result is applicable utilizing **MySQL FULLTEXT** search using **Natural Language mode**.
- ✓ In the search grid, the result is displayed for a valid match input in the "ontology_element_label" field. This is done among all the fields.

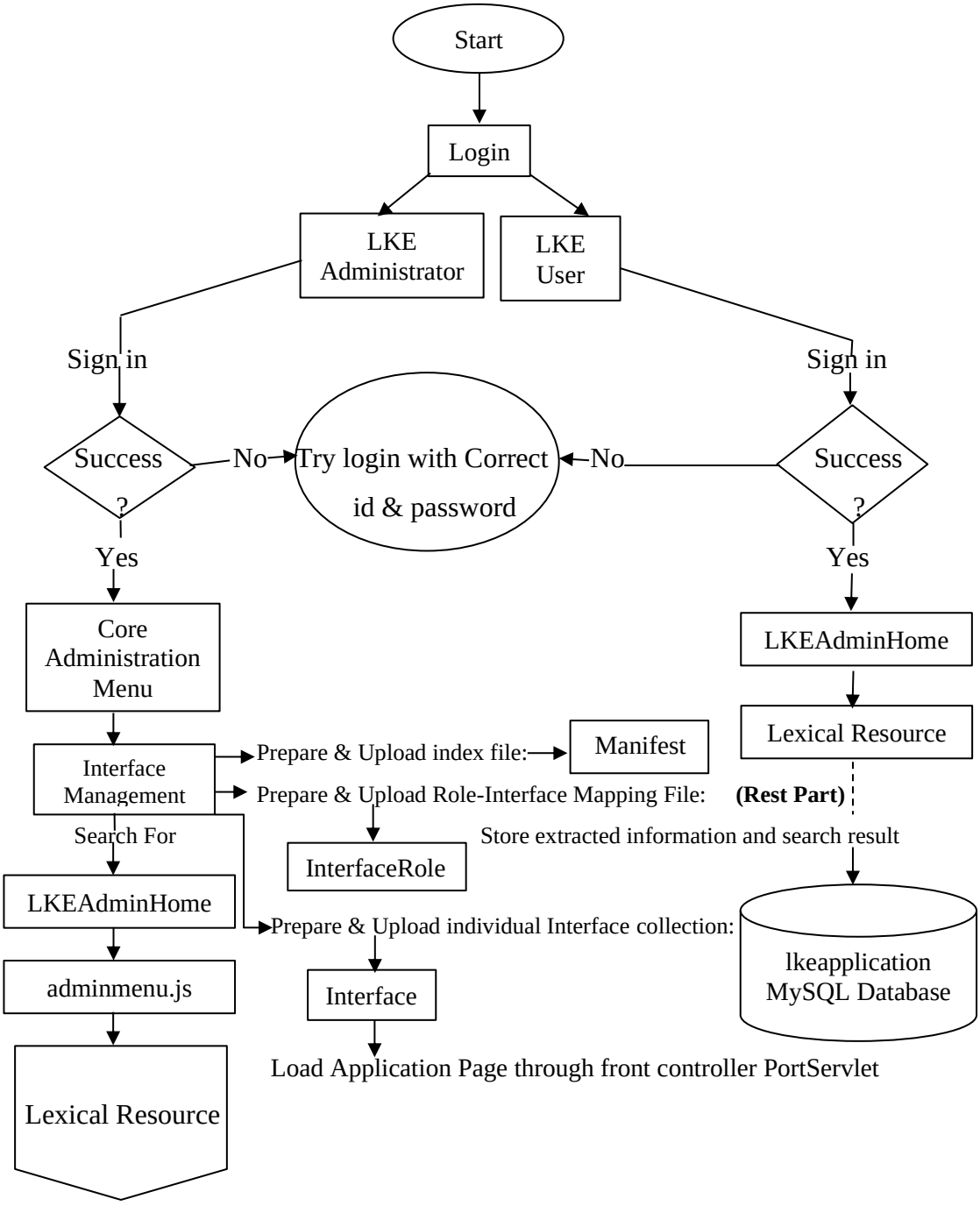

Figure 5: Flowchat of the System

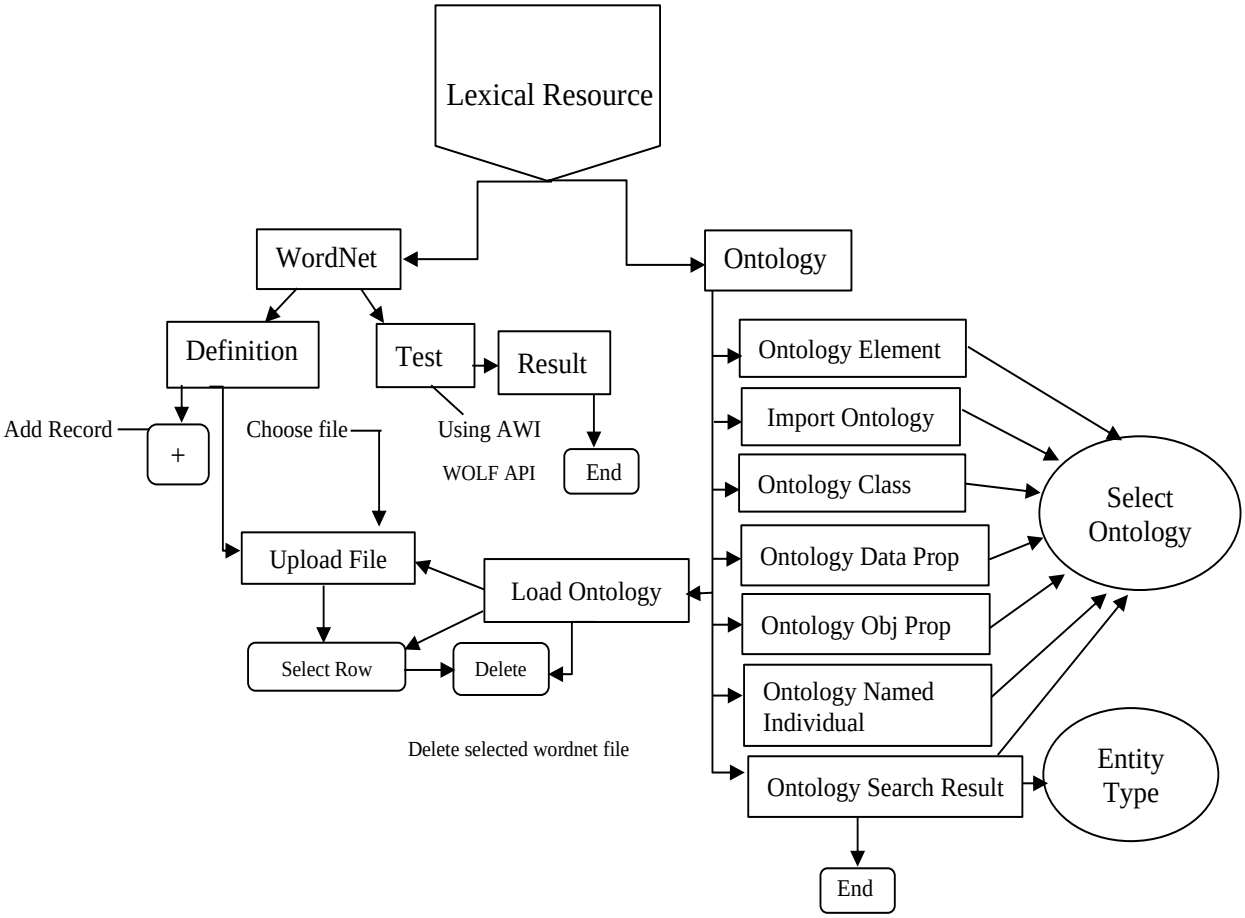

Figure 5: Flowchat of the System

3.3 Backend Detailed Design

3.3.1 Classes and their interaction

The implementation of the framework involves many classes and interfaces which are used to perform button click events present inside several user interface.xml.

Here, all the class names and their detailed description are given.

- **UploadWordNet Class:**

  This class is used for Wordnet zip file extraction and saving the wordnet file inside an already mentioned store location which is “wordnetpath”. Two methods are describe in the following-

a) uploadWordNetXML -

The process of file extraction is carried out in this implementation. It involves importing the "learnityInit.util" package. Within the "UploadWordNet.java" class, the methods "unzip" and "decompressGzip" from the "FolderZiper" class (which are already written inside the "FolderZiper" class) are called to accomplish this task.

b) deleteWordNetDetails –

This method is used to delete a Wordnet file from the Wordnet definition table.

- **WordNetTest Class:**

  This class facilitates the process of conducting word searches against individual Wordnet and selected types. The method utilized for this purpose is called "**getWordNetSearchFile**". A "**SwitchCase**" based approach is adopted in this class to handle relevant file formats and search types.

- **DataBaseLayer Class:**

  This class is basically used for writing methods that implement SQL queries importing java.sql.PreparedStatement. It is an interface that extends Statement which is an object that represents a precompiled SQL statement. The methods executed here are-

  - ✓ **saveWordNetDeatails**- After the wordnet file has been successfully uploaded and extracted to a specific path, an SQL **Update** query is invoked to update the file path for the wordnet.
  - ✓ **deleteWordNetdetails**- A **Delete** query is called inside this method.

- ✓ **Triple<String, String, String> getWordnettitle**- A package named **comv2.aunwesha.lfutil** is utilized to invoke the Triple.java interface. This interface is responsible for retrieving exactly three attributes from the **Select** query that is executed in this context.

❖ **Ontology Class:**

Ontology upload and information extraction are done by this class. We used several OWL API interfaces for different purposes such as:

- ✓ **OwlOntologyManager -** it manages a set of ontologies.
- ✓ **OwlDataFactory-** Entities, class expressions, and axioms can be created using this interface.
- ✓ **OwlOntology -** In the OWL2 specification, represents an Ontology.

The "**uploadOntologyXML**" method incorporates numerous interfaces, including the one mentioned.

If there are no dependencies or foreign key constraints in other entity tables, the "deleteOntologyDetails" method can be used to delete a selected ontology.

❖ **OntologyTest Class:**

When the search button is clicked with a specific input (any meaningful word or phrase), the "**getOntologySearchResult**" method in this class is invoked. This method retrieves search results by calling the "**DatabaseLayer.searchOntology**" method against specific ontology element types.

❖ **DatabaseLayer Class:**

Several methods are written here for different SQL queries to perform and deliver stored information and search result. They are-

- ✓ **saveOntologyDetails –** An **insert** query is called here which helps in loading the basic ontology details.
- ✓ **fetchOntologyElementdetails –** Here, again an insert query is called but this time it loads all the axiom type-related data retrieving more detailed information.
- ✓ **importOntologyDeails –** All the importOntology related details are stored under ontology_import table.
- ✓ **ontologyClassDetails,ontologyNamedIndividualDetails,**

ontologyDataPropertyDetails, ontologyObjectPropertydetails - All these methods store their respective axiom type details in the individual table.

- ✓ **Triple<String, String, String> SearchOntology**- This method performs a Select query using inner join. It uses a FULLTEXT based search syntax. The search is performed against NATURAL LANGUAGE MODE. The goal is to find a match for the chosen field against the given input.

3.3.2 Database Design

The database has been designed in MySQL Workbench executing certain create table queries.

Figure 6 displays the EER Diagram of the database that has been built for our application framework.

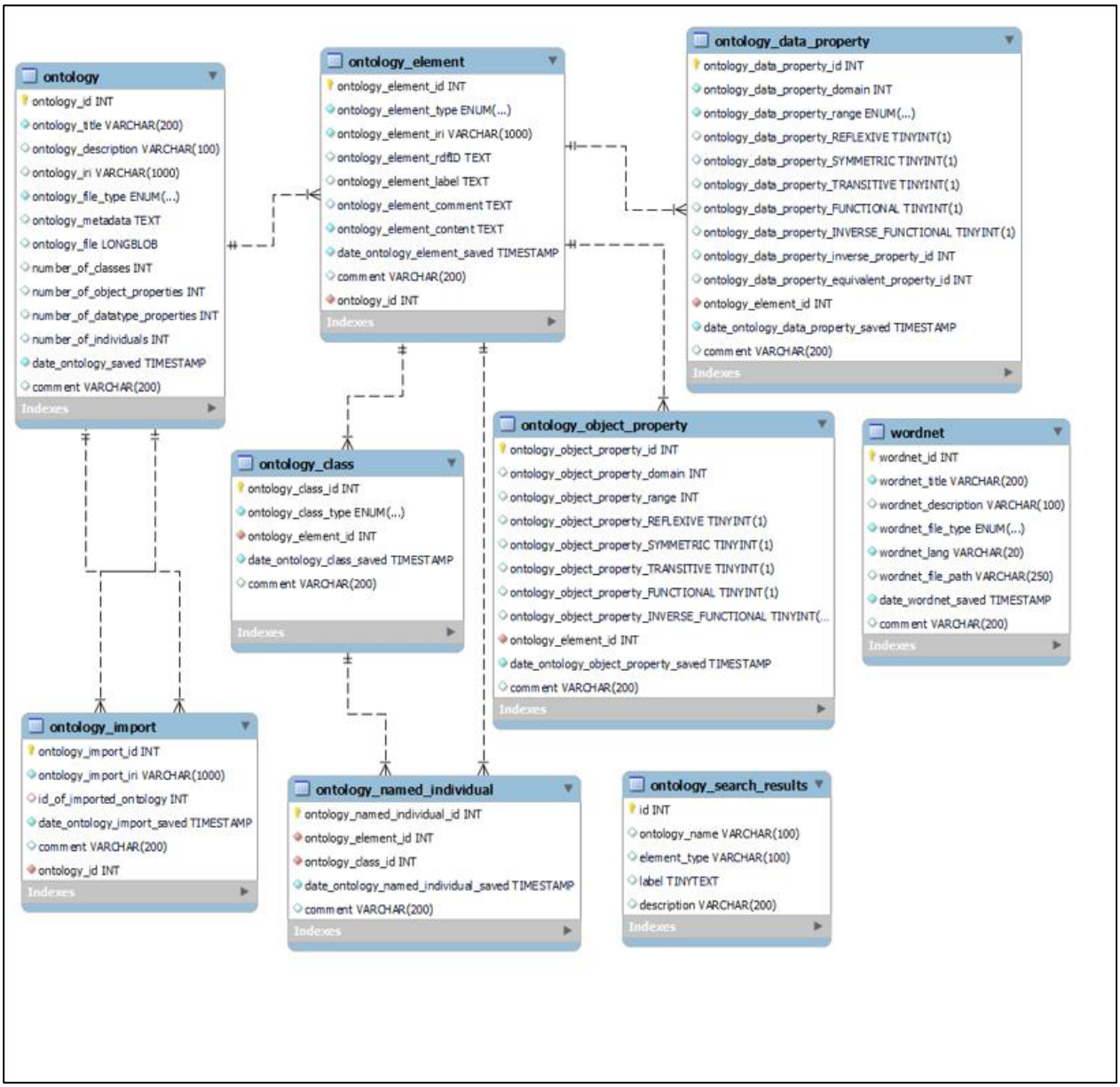

Figure 6: Database Design – EER Diagram

CHAPTER 4

4.0 EXPERIMENTATION AND RESULTS

❖ Experimental Setup

Before getting started, ensure that Eclipse IDE has been installed on your system. Import the LearnITy Knowledge Engine (LKE.zip) file as a project in the console. Afterwards, continue with the required setup by ensuring a proper connection check for the Tomcat server to ensure the project runs smoothly, and also configure the MySQL database connection with right login details. Once the experimental setup is completed, run it and proceed to the next step.

❖ Adding Module as Menu

1. Login to LearnITy Administrator and go to Interface Management under Core Administration lists. Figure 7 shows LearnITy Administrator Login page.

Figure 7: Admin Login

After successful login, Core Administration Dashboard is displayed and is shown in Figure 8.

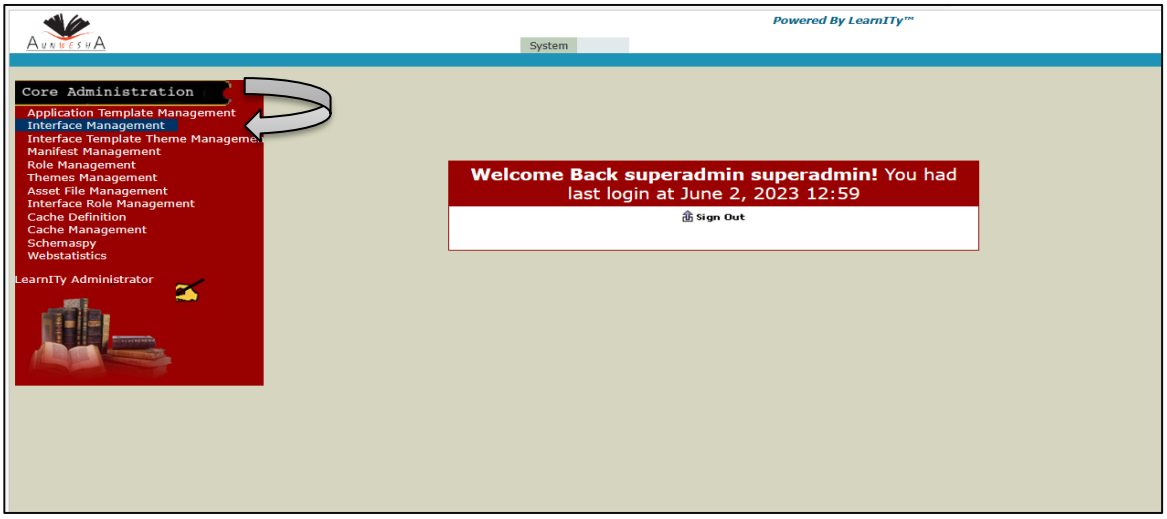

Figure 8: Core Administration Dashboard

2. Enter LKEAdminHome into the search bar and click on the go button, is shown in Figure 9.

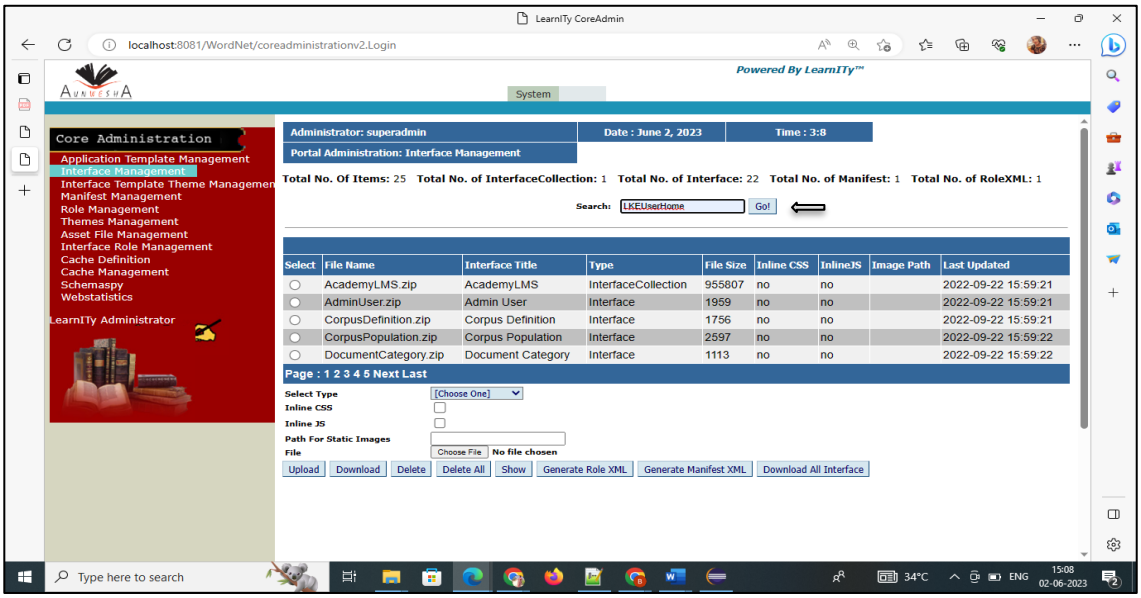

Figure 9: Interface Management

3. First, select LKEAdminHome.zip and click on the show button. It is displayed in Figure 10.

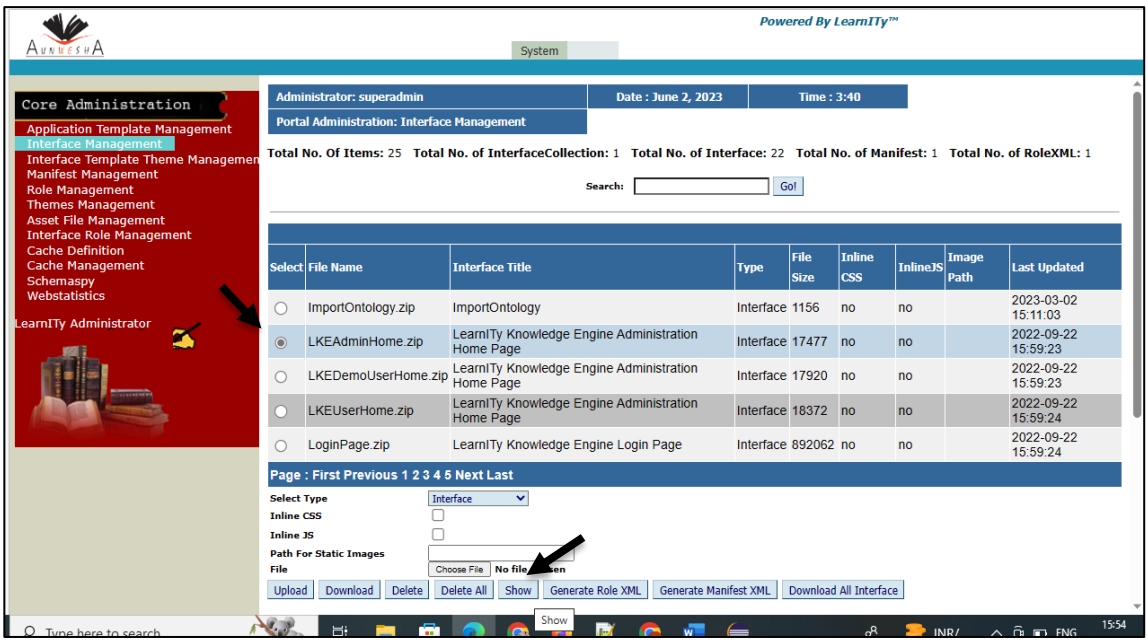

Figure 10: Interface Management

A page will open. Next, on this page, select adminmenu.js and click on the view button. Figure 11 shows this.

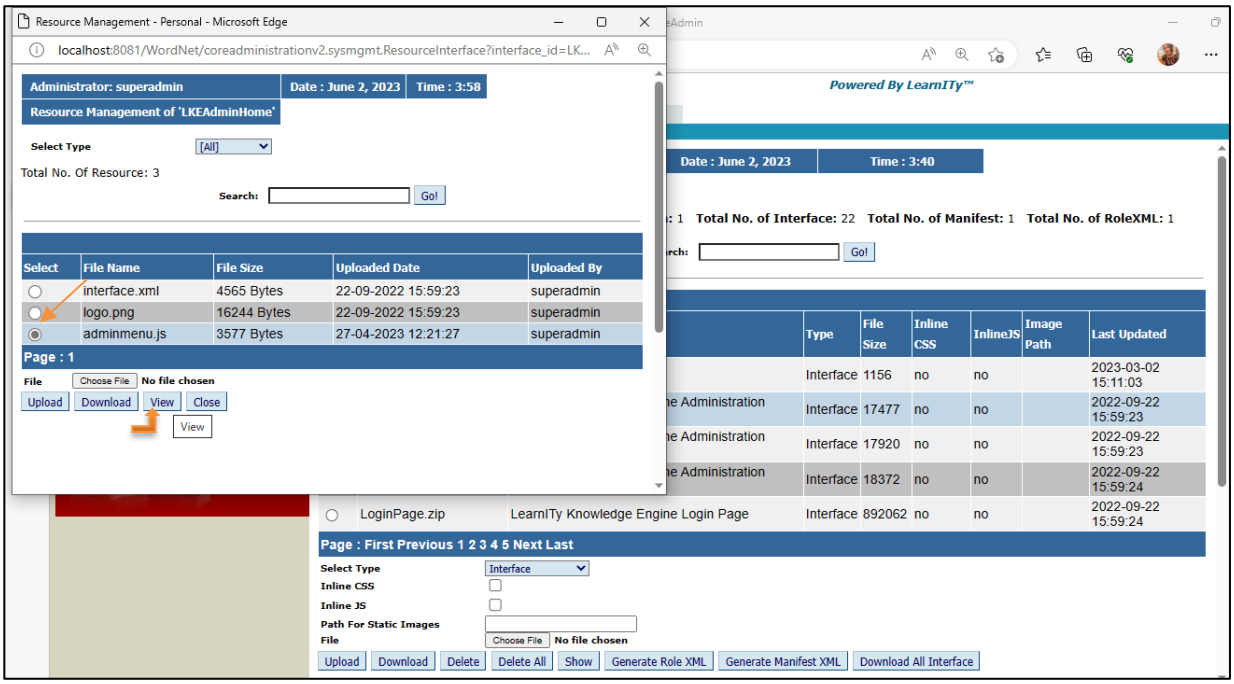

Figure 11: LKEAdminHome

Another page will be open here. Finally, in this page, write relative code for adding new capabilities or module and sub-modules. Refer to Figure 12 for visual representation.

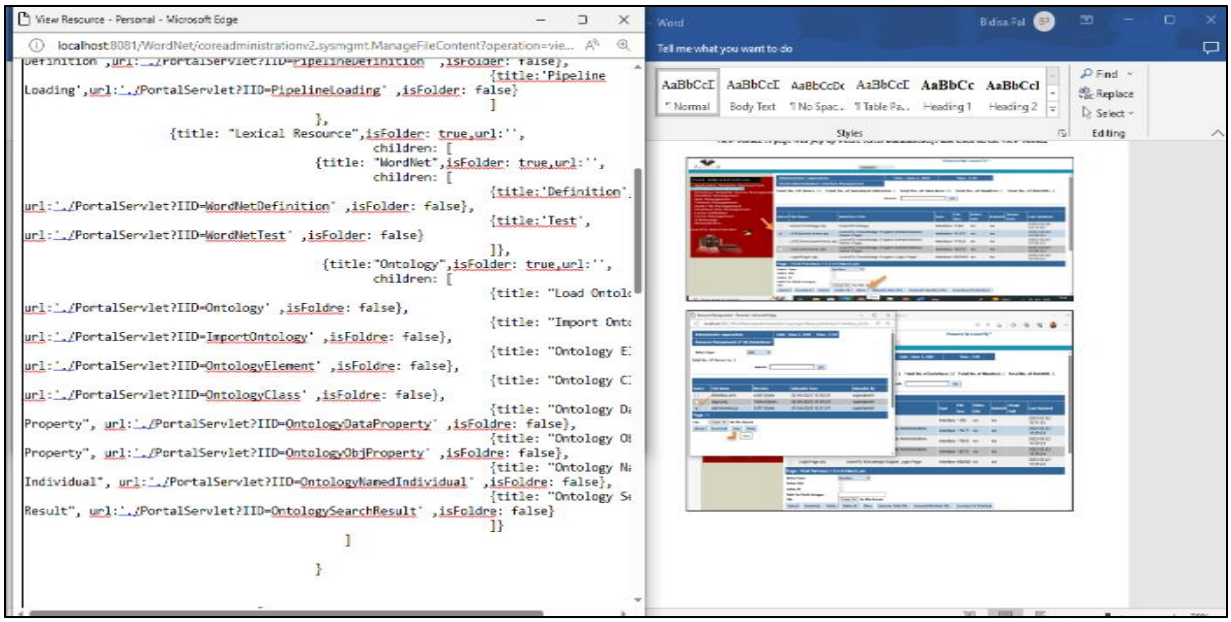

Figure 12: adminmenu.js

Then Save it by hitting the Save button. On correct code insertion, it will show the message as Resource Inserted. Click the Close button as shown in Figure 13.

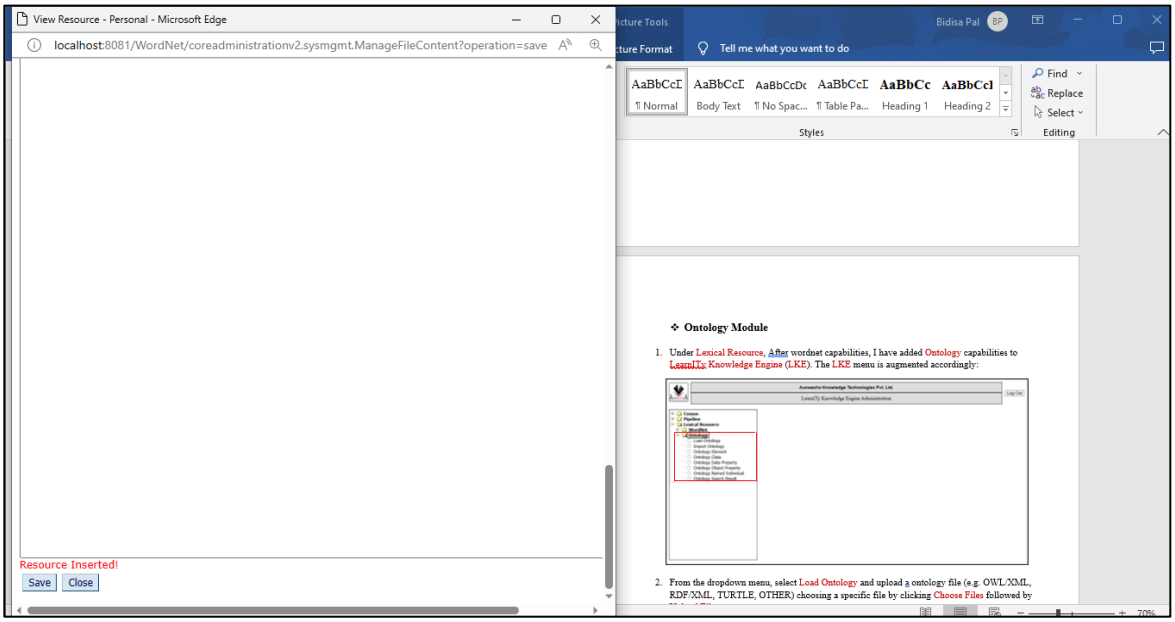

Figure 13: adminmenu.js

❖ Wordnet Module

1. After Adding Lexical Resource, I have added WordNet capabilities to LearnITy Knowledge Engine (LKE) as shown in Figure 14. The LKE menu is augmented accordingly:

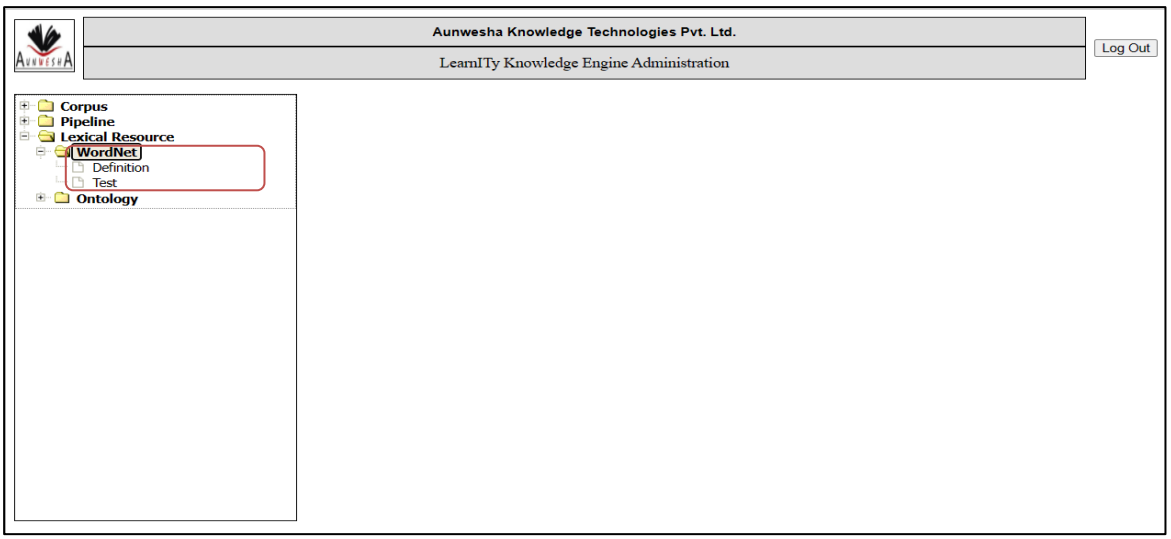

Figure 14: WordNet User Interface

2. From the dropdown menu, select Definition and create or add a new row by clicking on '+'. Figure 15 is referred to this context.

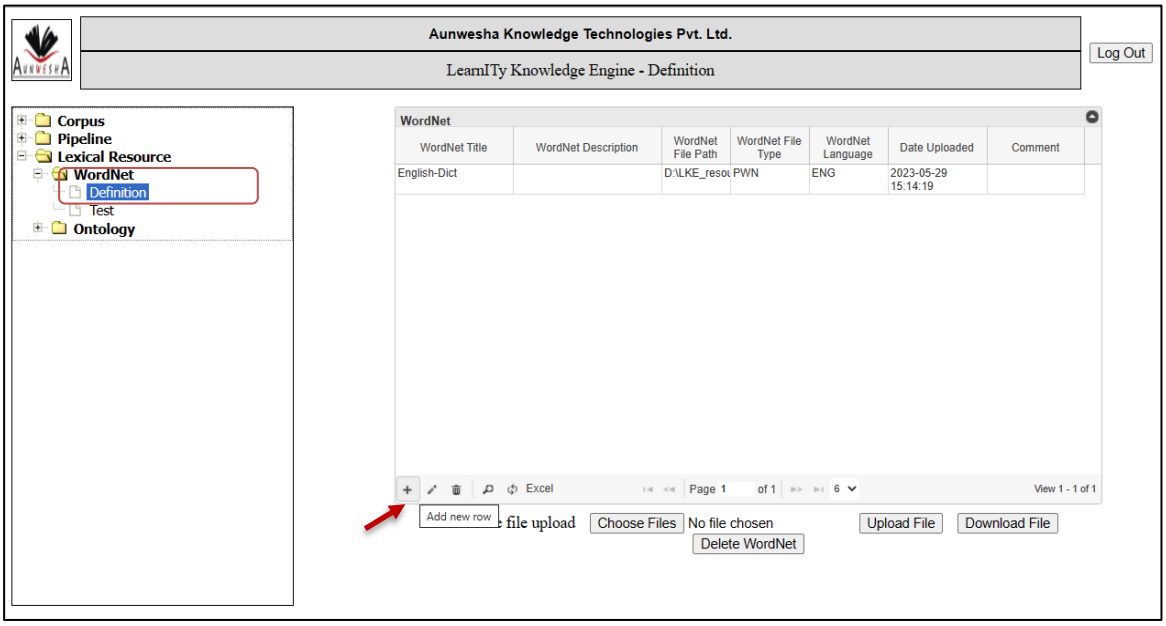

Figure 15: Definition Interface

Give the WordNet title and choose other mandatory fields (i.e., WordNet file type) then click on the submit button as shown in Figure 16.

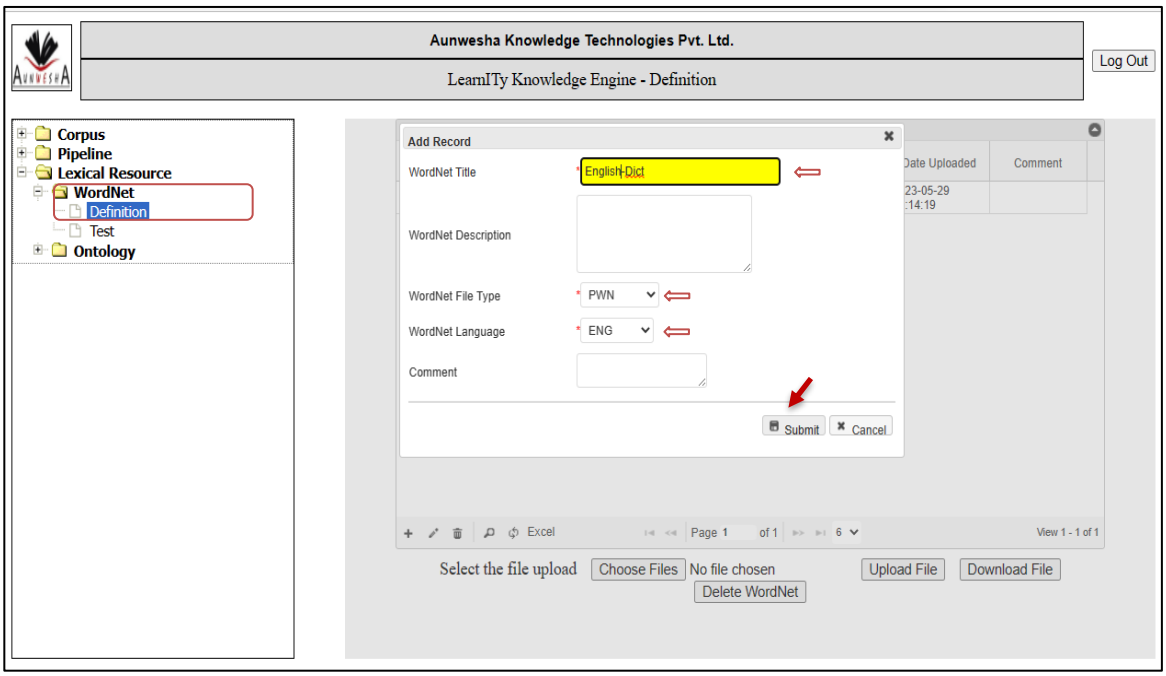

Figure 16: Add Record

3. Select the row just created and choose the wordnet file to upload (e.g. Dict or XML-based zip or xml.gz file) hitting by Choose Files button and then hit the Upload File button. On successful upload, the WordNet file path will be displayed under the WordNet File Path column. Figure 17 shows the Definition interface.

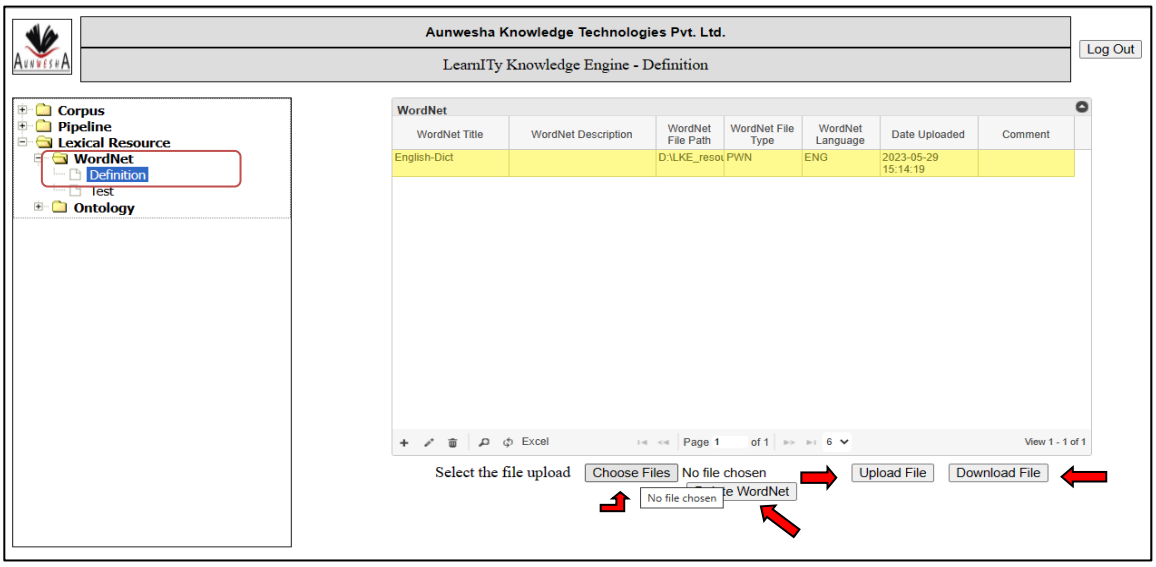

Figure 17: Definition Interface

Note: By selecting a particular row, one can delete or download the uploaded wordnet file by clicking on the Delete WordNet or Download File button.

4. Proceed for the Test and select specific wordnet language type and type from the relative drop-down list and then search for a word in that specific selected language. Click on the Search button. On successful search, it will show the result in text area of result part. One can clear the result by hitting Clear Result button. Test interface is shown in Figure 18.

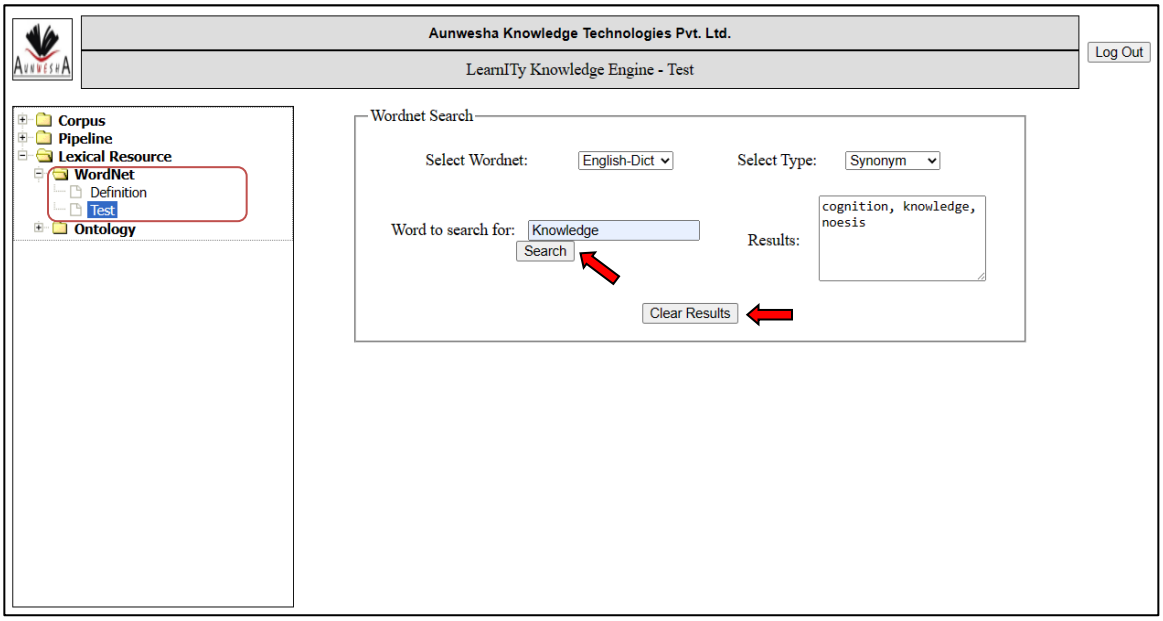

Figure 18: Test Interface

❖ Ontology Module

1. Under Lexical Resource, after wordnet capabilities, I have added Ontology capabilities to LearnITy Knowledge Engine (LKE). The LKE menu is augmented accordingly. Ontology screen is displayed in Figure 19.

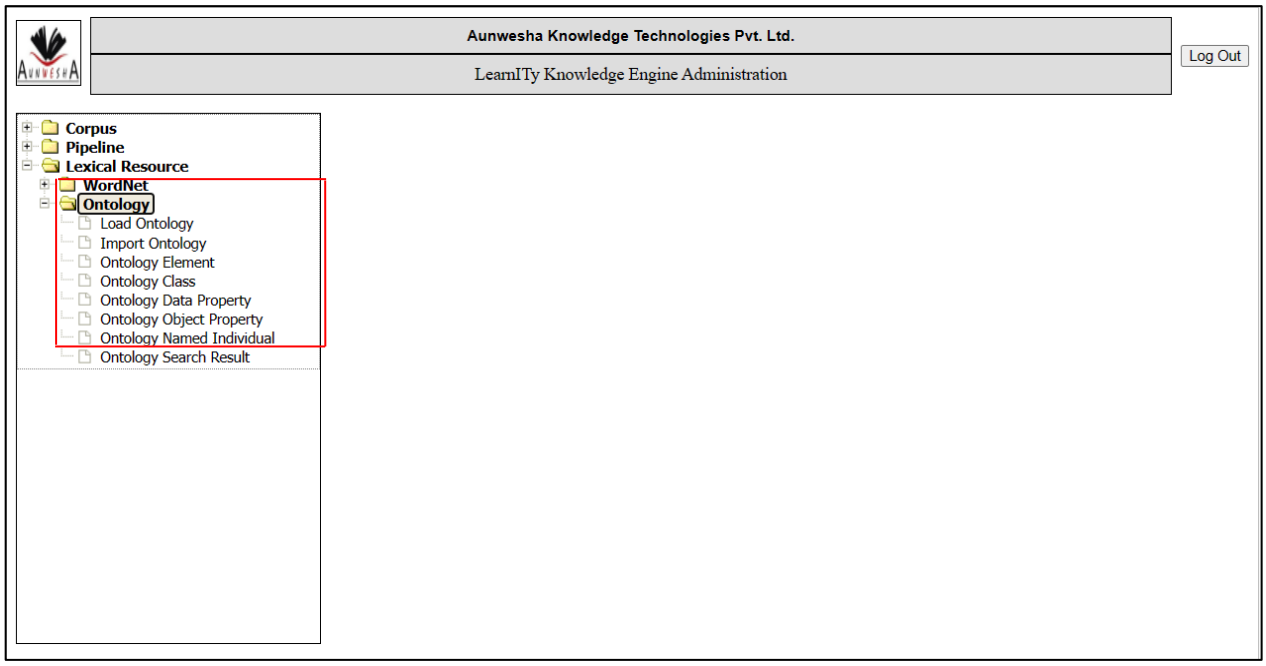

Figure 19: Ontology Screen

2. From the dropdown menu, select Load Ontology and upload a ontology file (e.g., OWL/XML, RDF/XML, TURTLE, OTHER) choosing a specific file by clicking Choose Files followed by Upload File as shown in Figure 20.

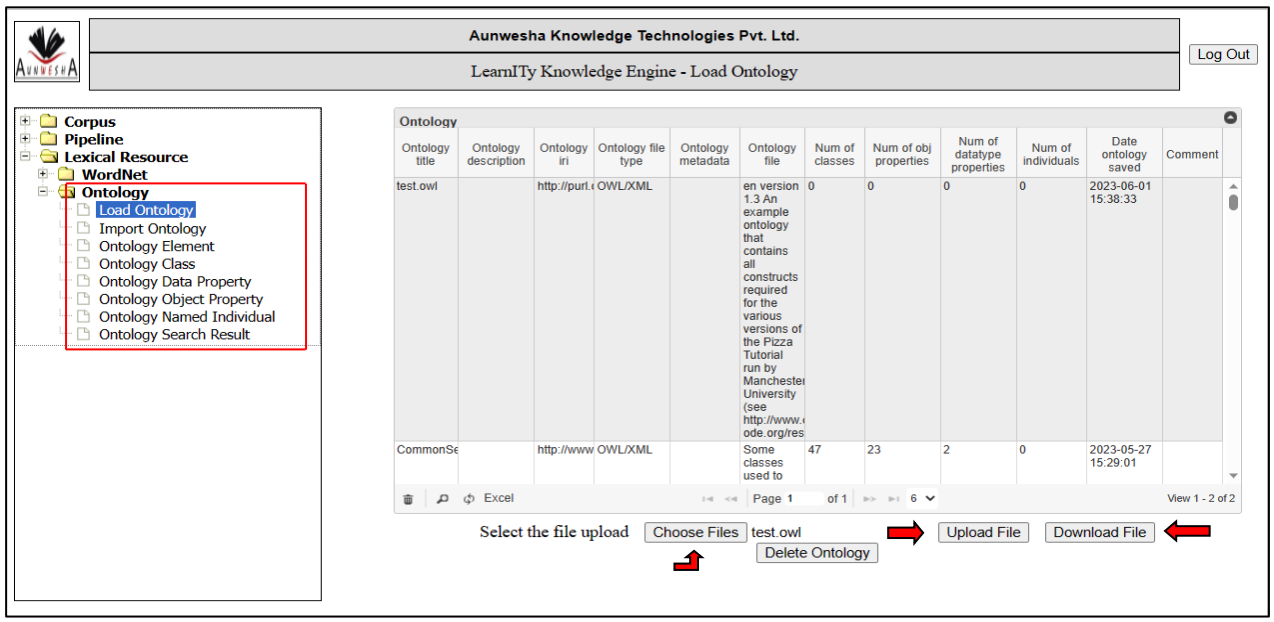

Figure 20: Load Ontology Screen

Note: One can download the loaded ontology file by clicking on Download File button.

3. Proceed to Import Ontology and select a ontology from the drop-down box as shown in Figure 21. The chosen ontology will then retrieve the set of loaded ontologies it is related to through the "directlyImports" relation, if any exists.

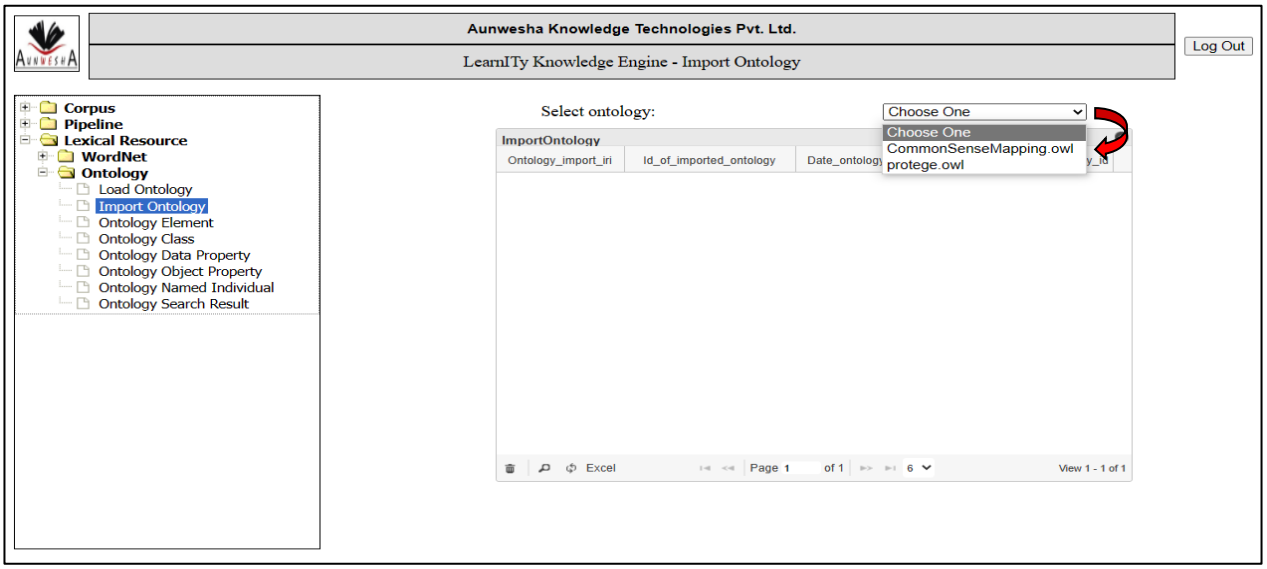

Figure 21: Import Ontology Screen

4. To access the ontology elements, go to Ontology Element and choose an ontology from the drop-down menu. Upon selecting the ontology, it will display a comprehensive set of entities or element types (such as class, datatype, object property, etc.) along with other relevant details that exists. This feature allows you to view all the pertinent information for each element type in the chosen ontology. Figure 22 shows the Ontology Element page.

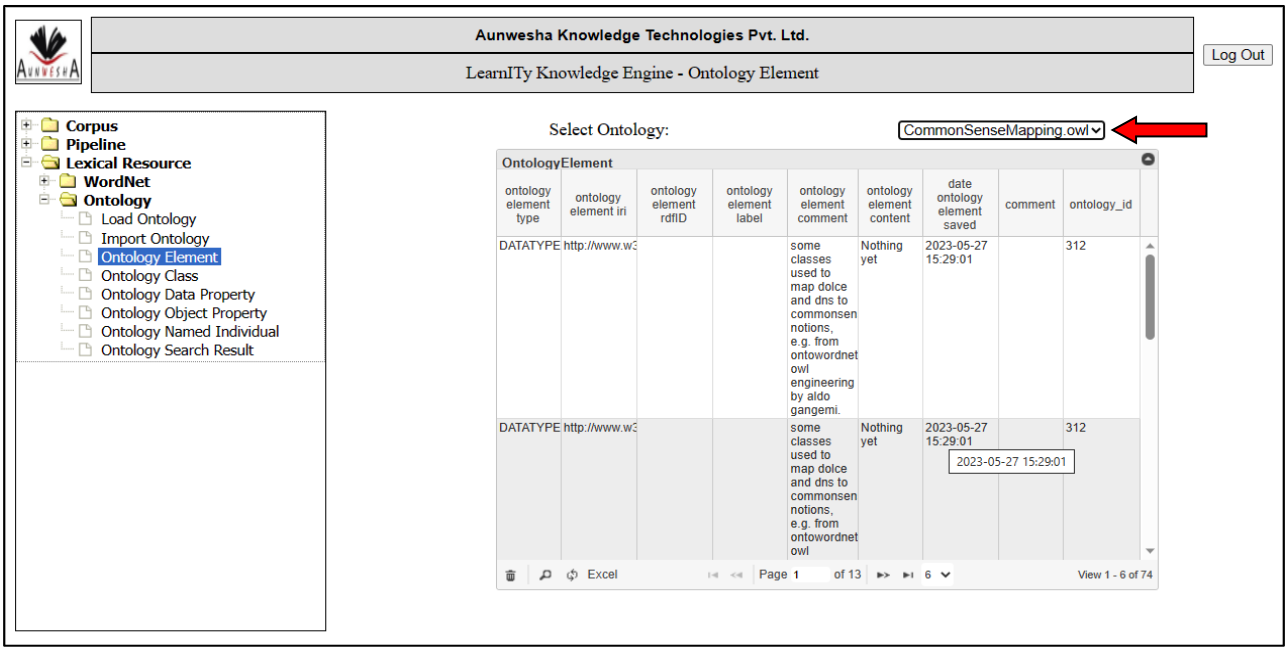

Figure 22: Ontology Element Screen

5. To access the ontology classes, navigate to Ontology Class and select a ontology from the drop down box, for the chosen ontology it gets the ontology_class_type in general as SIMPLE and corresponding ontology_element_id.

Note: Class type 'Enumerated' and 'Restriction' are not able to fetch or test for the used ontologies so all classes are categorized as 'simple' in general.

Figure 23 displays the same.

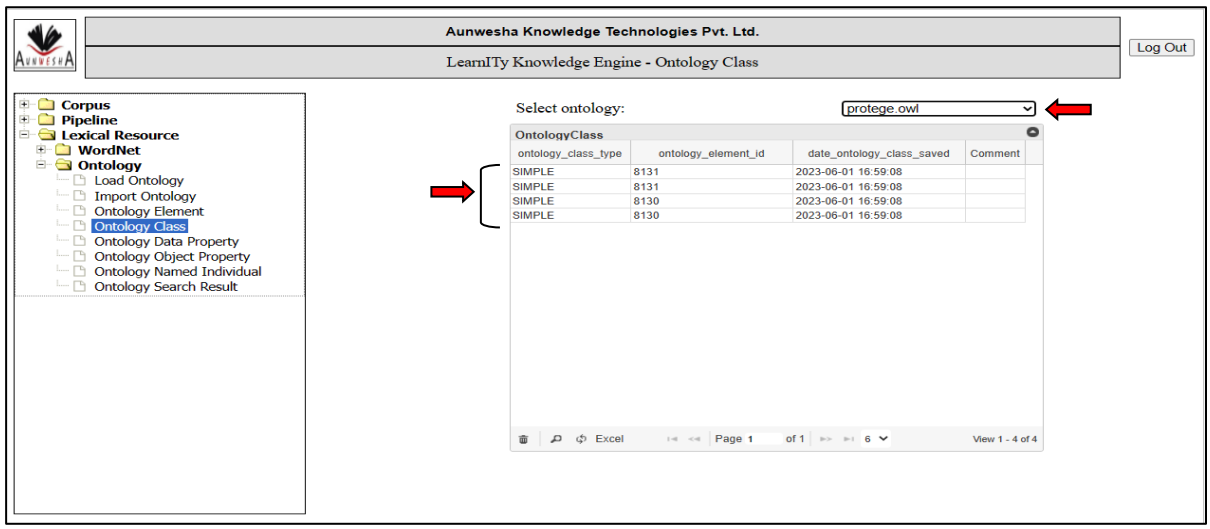

Figure 23: Ontology Class Screen

6. Proceed to Ontology Data Property and select a ontology from the drop down box. For the chosen ontology it gets the ontology data property domain, range and many other relevant details if exist. Figure 24 shows the Ontology Data Property screen.

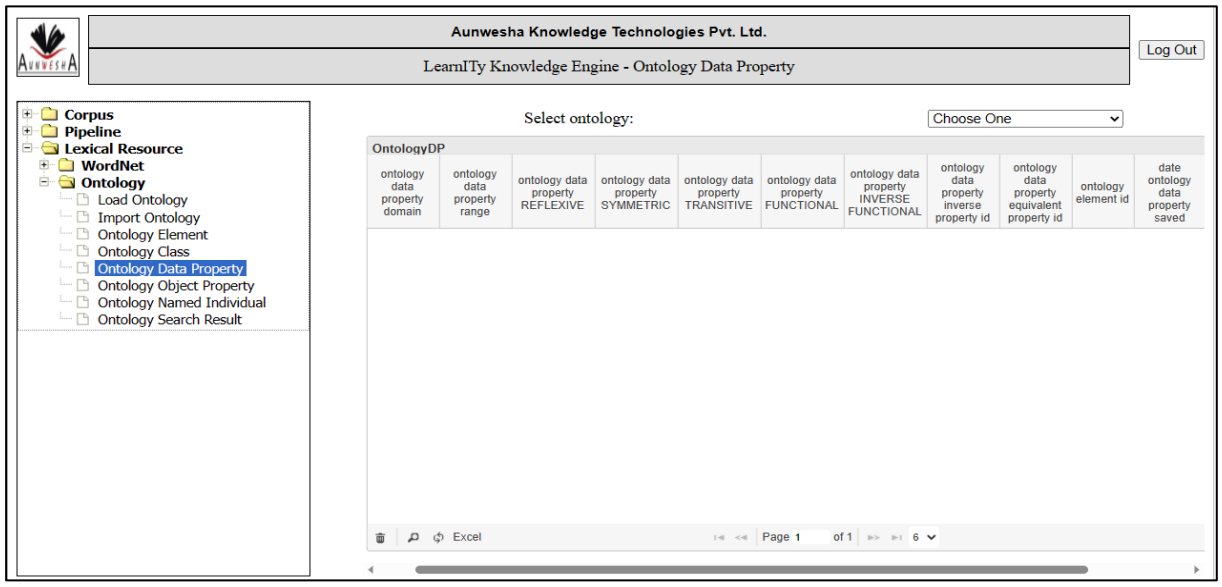

Figure 24: Ontology Data Property Screen

7. Likewise, proceed for other two element type Ontology Object Property, Ontology Named Individual and get the applicable result accordingly as depicted in Figures 25 and 26.

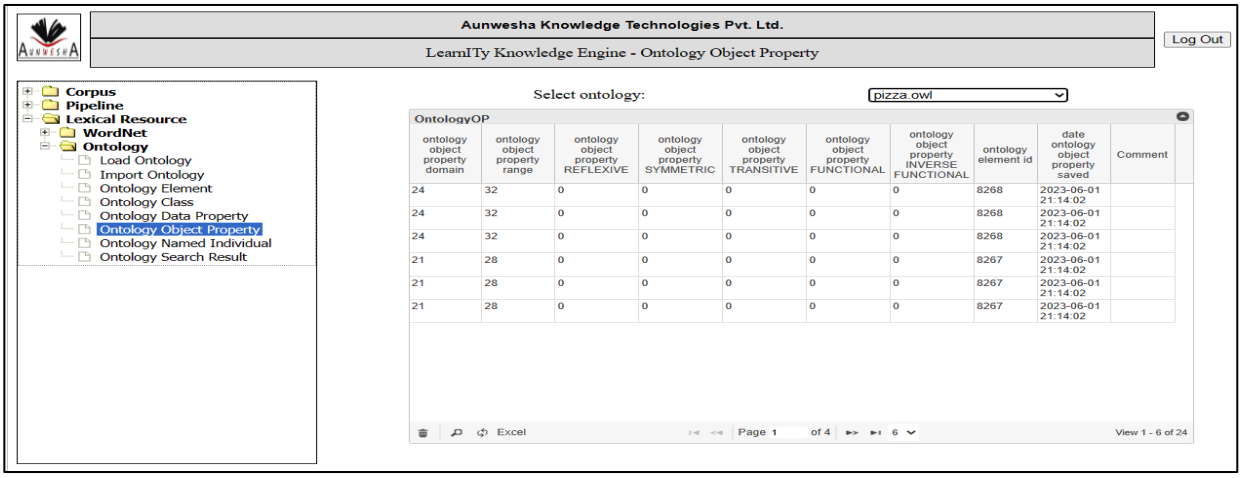

Figure 25: Ontology Object Property Screen

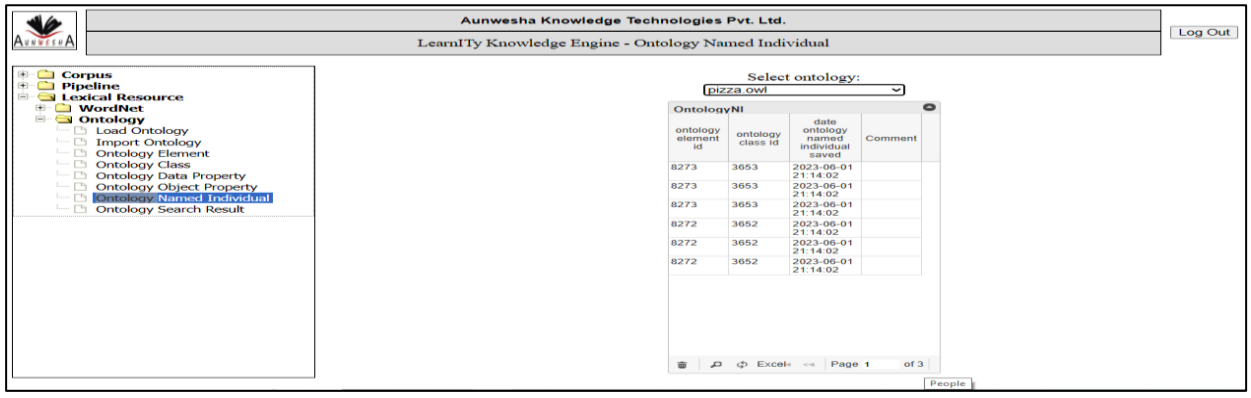

Figure 26: Ontology Named Individual Screen

8. Proceed for the Ontology Search Result and select specific ontology and ontology element type or entity type from the relative drop-down list. Figure 27 displays the Ontology Search Result screen.

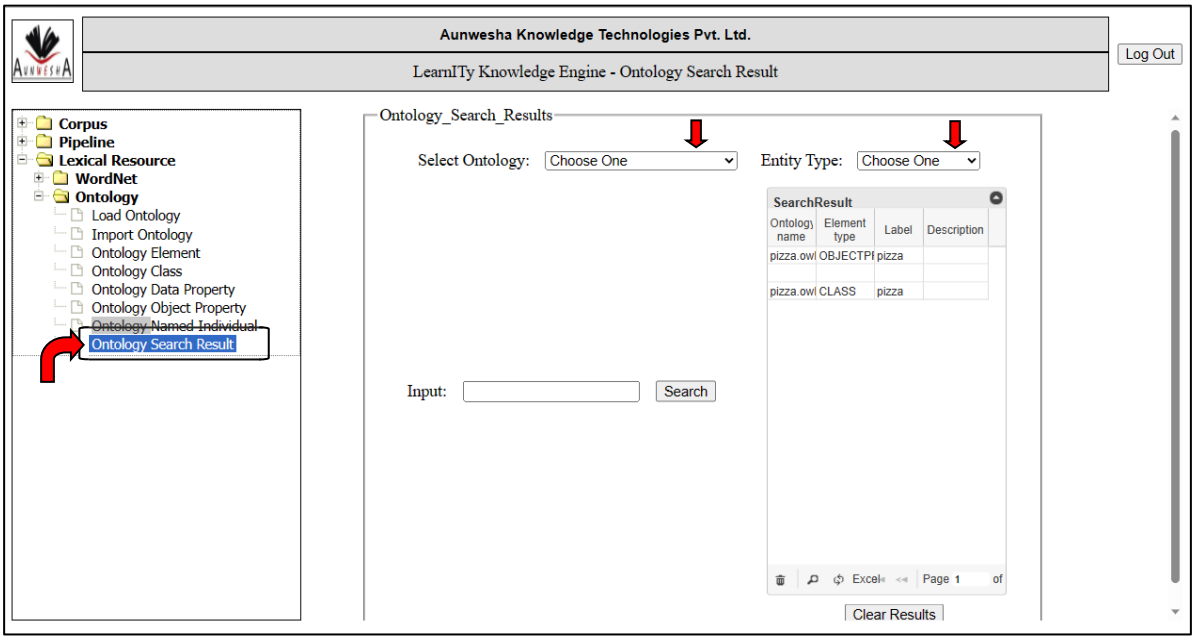

Figure 27: Ontology Search Result Screen

9. To perform a search, enter a comparable word in the input field. If there is a successful match with the ontology element label from Ontology Element table, the result will be displayed and stored in the Ontology search results table. Otherwise, it will be stored as a null value. Here, the search is conducted using **MySQL FULLTEXT** search in **Natural Language mode**.

   Note: One can search matching against many other or more field present in ontology_element table. Here, result has been shown against only ontology_element_lable attribute.

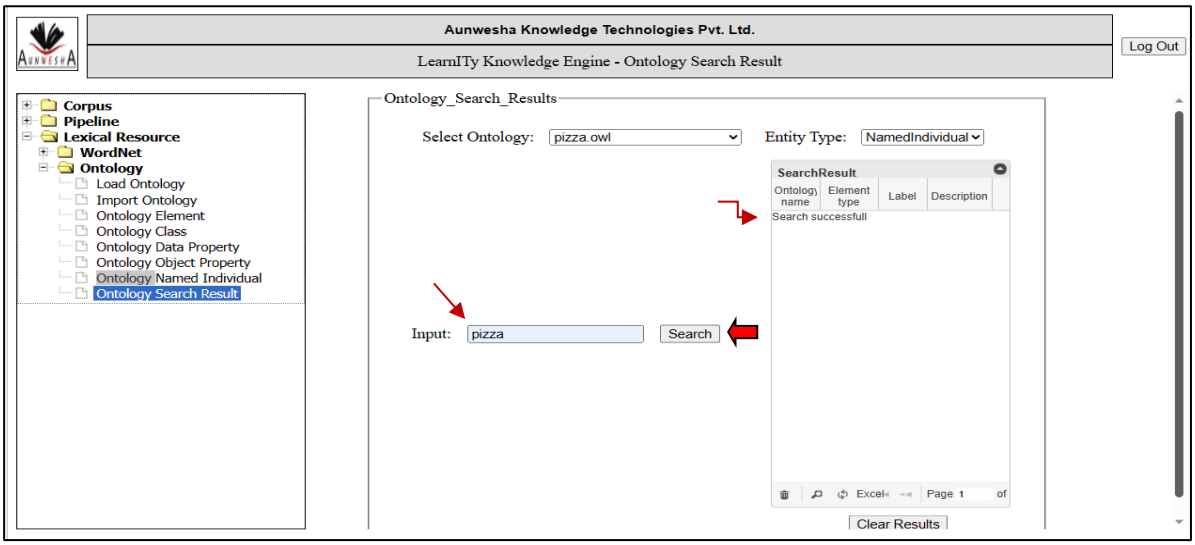

Figure 28: Ontology Search Result Screen

10. Work on few remaining part is under progress satisfactorily with ongoing testing of additional ontology files.

CHAPTER 5

CONCLUSION & FUTURE WORK

5.1 Conclusion

In this work, we proposed an approach for designing and developing a search algorithm on Lexical Resources. We experimented with this approach by extracting data from lexical resources and then searching a text based on matching. We used ontology downloading from Archivo(ontology archive). We presented the possibility to generate queries like (SELECT, INSERT, DELETE, and MODIFY) by selecting the relevant fields to include the queries based on extracted data from Wordnet and Ontology files. Moreover, we discussed the possibility of accessing the ontology via web services.

5.2 Future work

Our future work consists of experimenting like,

1. Including reasoning functionality for OWL APIs and enhancing it by integrating reasoning techniques, selecting different ontology reasoners like (HermiT, Pellet) and exploring new ways to describe reasoning rules.
2. Continuing the work of the test interface that has been created for comparing two synsets or words using different algorithms. It is, therefore, possible to develop a wordnet similarity checker on the same screen as the search.
3. Using the download button which is in progress will enable the extracted files to be downloaded in the same file format as it was before even after manipulating the extracted data.
4. Adding more different language-based WordNet files can provide access to more different language-based searches in one place. It is also possible to create a local language-based wordnet and add it here for searching.

REFERENCES

[1] Agnieszka Konys, “Knowledge Repository of Ontology Learning Tools from Text”, in 23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Procedia Computer Science, 2019.

[2] Asim, M. N., Wasim, M., Khan, M. U. G., Mahmood, W., & Abbasi, H. M., “A survey of ontology learning techniques and applications”, 2018.

[3] Jiomekong, A., Camara, G., & Tchuente, M., “Extracting ontological knowledge from Java source code using Hidden Markov Models”, Open Computer Science, pp: 9(1):181–199, (2019).

[4] Moldovan, D., Pop, C., Antal, M., Cioara, T., Anghel, I., & Salomie, I., “ *SWAG*: Semantic web application generator - a library for using ontologies as web services”, IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP), 2016.

[5] Morad Hajji, Mohammed Qbadou, & Khalifa Mansouri, “JWOLF:Java French Wordnet Library”, International Journal of Advanced Computer Science and Applications(IJACSA), Laboratory SSDIA, ENSET Mohammedia Hassan II University of Casablanca Mohammedia, Morocco, Vol. 10, 2019.

[6] Priyatno, J., & Bijaksana, M. A. , “Clustering Synonym Sets in English WordNet”, 7th International Conference on Information and Communication Technology (ICoICT), School of Computing, Telkom University, Indonesia, 2019.

[7] Chimalakonda, S., & Nori, K. V. , “An ontology based modeling framework for design of educational technologies”, Smart Learning Environments, Department of Computer Science & Engineering, IIT Tirupati, Tirupati, India7, 2020.

[8] Aditya Kalyanpur, Daniel Jimenez Pastor, Steve Battle, Julian Padget, “Automatic Mapping of OWL Ontologies into Java”, May 2004.

[9] Ales Horak, & Adam Rambousek, “Using Wordnets and Ontologies for Text-Meaning Assignment Implementation Details of the KYOTO Project First Phase”, In Proceedings of the 4th International Conference on Software and Data Technologies, pages 303-307, Faculty of Informatics, Masaryk University, Botanick´a 68a, 602 00 Brno, Czech Republic, 2009.

[10] Luca Buoncompagni, Syed Yusha Kareem, & Fulvio Mastrogiovanni, “OWLOOP: A modular API to describe OWL axioms in OOP objects hierarchies ”, Publisher: Elsevier, Publication: SoftwareX, Volume 17, January 2022.

Appendix part A

❖ Users' Manual

✓ System Requirements

Personal computer or laptop having the following features.

- Intel core i3
- 8 GB RAM
- Internet Connection
- Windows operating system (Windows 10 Pro)
- 15.6" HD display, Built in DVD Drive, Wireless and Bluetooth
- Connection and HDMI
- For accelerated emulator: 64-bit operating system, x64-based processor.

✓ Software Requirements

- Web Server: Tomcat 8.5
- Database: MySQL (8.0.29)
- Eclipse IDE (4.24.0)
- Browser: Windows internet explorer and Google Chrome.
- Notepad++ (8.4.2) and
- Microsoft word 2010.
- Java JDK (18)

✓ Programming Requirements

- Java
- JavaScript
- XML
- jQuery
- MySQL

✓ Download Speeds

- Internet speed is measured in Mbps.
- 3-5Mbps is recommended for the download speed.

Appendix part B

❖ Code Snippets for Wordnet:

✓ WordNetDefinition(interface.xml)

```
<interface
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://learnityframework.org/"
    xsi:schemaLocation="http://learnityframework.org/
    http://localhost:8181/learnitymsme/xsd/Interface.xsd"
    id="WordNetDefinition" title="WordNet Definition" type="Interface">

    <configuration checkrole="false"></configuration>

    <structure>
        <part id="BodyDiv" class="label"></part>
        <part id="WordNetDefinitionGrid" class="DBgrid" caption="WordNet"
    sortname="wordnet_id" sortorder="desc" autoencode="false"
    generateExcel="true">
            <column head="Wordnet_id" name="wordnet_id" index="1"
    width="0" editable="true" hidden="true" key="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="WordNet Title" name="wordnet_title" index="2"
    width="120" editable="true" hidden="false" key="false" required="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="WordNet Description"
    name="wordnet_description" index="3" width="150" editable="true"
    hidden="false" key="false" required="false">
                <edit type="textarea" cols="28" rows="5"></edit>
            </column>
```

<column head="WordNet File Path" name="wordnet_file_path" index="4" width="70" editable="false" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>
<column head="WordNet File Type" name="wordnet_file_type" index="5" width="80" editable="true" hidden="false" key="false" required="true">
<edit type="select" editdomaintype="fixed" value="PWN:PWN;XML:XML;RDF:RDF;DebVisDic:DebVisDic;OTHER:OTHER" size="30"></edit>
</column>
<column head="WordNet Language" name="wordnet_lang" index="6" width="80" editable="true" hidden="false" key="false" required="true">
<edit type="select" editdomaintype="fixed" value="ENG:ENG;BENG:BENG;FRENCH:FRENCH;OTHER:OTHER" size="30"></edit>
</column>
<column head="Date Uploaded" name="date_wordnet_saved" index="7" width="100" editable="false" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>
<column head="Comment" name="comment" index="8" width="100" editable="true" hidden="false" key="false" required="false">
<edit type="textarea" size="30"></edit>
</column>

<loadquery

```
                value="select WordNet_id, wordnet_title,
wordnet_description, wordnet_file_path, wordnet_file_type, wordnet_lang,
date_wordnet_saved, comment from wordnet"></loadquery>

        <delete>
            <queries>
                <query id="1"
                    sql="[CDATA[delete FROM wordnet where
wordnet_id=%wordnet_id%]]"
                    parameter="wordnet_id"></query>
            </queries>
        </delete>

        <add>
            <queries>
                <query id="1"
                    sql="[CDATA[insert into wordnet
(wordnet_title,wordnet_description,wordnet_file_type,wordnet_lang,comment)
values
(%wordnet_title%,%wordnet_description%,%wordnet_file_type%,%wordnet_la
ng%,%comment%)]]"

    parameter="wordnet_title,wordnet_description,wordnet_file_type,wordnet
_lang,comment"></query>
            </queries>
        </add>
        <modify>
            <queries>
                <query id="1"
                    sql="[CDATA[update wordnet set
wordnet_title=%wordnet_title%,
```

```
wordnet_description=%wordnet_description%,wordnet_file_type=%wordnet_fil
e_type%, wordnet_lang=%wordnet_lang%, comment=%comment% where
wordnet_id=%wordnet_id%]]"

        parameter="wordnet_id,wordnet_title,wordnet_description,wordnet_file_ty
pe,wordnet_lang,comment"></query>
                        </queries>
                </modify>
        </part>
        <part id="UploadFileRowContainer" class="label"></part>
        <part id="WordNetFileLabel" class="label"></part>
        <part id="WordNetFile" class="inputfile"></part>
        <part id="UploadButton" class="button"></part>
        <part id="DownloadButton" class="button"></part>
        <part id="DeleteButton" class="button"></part>
        <part id="WordNetDefinitionAjaxComp" class="ajaxcomponent"></part>
</structure>

<layout id="WordNetDefinitionLayout">
        <part id="BodyDiv"></part>
        <part id="WordNetDefinitionGrid" height="300"
parent_id="BodyDiv"></part>

        <part id="UploadFileRowContainer" parent_id="BodyDiv"></part>
        <part id="WordNetFileLabel"
parent_id="UploadFileRowContainer"></part>
        <part id="WordNetFile" parent_id="UploadFileRowContainer"></part>
        <part id="UploadButton" parent_id="UploadFileRowContainer"></part>
        <part id="DownloadButton"
parent_id="UploadFileRowContainer"></part>
        <part id="DeleteButton" parent_id="UploadFileRowContainer"></part>
```

```
        <part id="WordNetDefinitionAjaxComp" parent_id="BodyDiv"></part>

</layout>
<content id="WordNetDefinitionContent">
   <part id="WordNetFileLabel" value="Select the file upload"
valuetype="inline"></part>
        <part id="UploadButton" value="Upload File" valuetype="inline"></part>
        <part id="DownloadButton" value="Download File"
valuetype="inline"></part>
        <part id="DeleteButton" value="Delete WordNet"
valuetype="inline"></part>
</content>

<style id="WordNetDefinitionStyle">
   <part id="BodyDiv" valuetype="reference"
value="FlexColumnContainer"></part>
        <part id="BodyDiv" valuetype="reference"
value="FlexColumnContainer"></part>
        <part id="UploadFileRowContainer" valuetype="reference"
value="FlexRowContainer"></part>
        <part id="UploadFileRowContainer" valuetype="inline"
value="[CDATA[width:80%;]]"></part>
</style>

<behaviour id="WordNetDefinitionBehaviour">
                <part id="UploadButton">
                <event name="onclick" type="simple"
function="UploadButton_onclick"
                        valuetype="jsevent"
resourceid="WordNetDefinitionjs"></event>
        </part>
```

```
        <part id="DownloadButton">
                <event name="onclick" type="simple"
function="DownloadButton_onclick"
                        valuetype="jsevent"
resourceid="WordNetDefinitionjs"></event>
        </part>
        <part id="DeleteButton">
                <event name="onclick" type="simple"
function="DeleteButton_onclick"
                        valuetype="jsevent"
resourceid="WordNetDefinitionjs"></event>
        </part>
        <part id="WordnetDefinitionAjaxComp">
                <event name="ajax" type="simple" function="dummy"
javaclass="UploadWordNet"></event>
        </part>
</behaviour>

<resource>
        <resourceitem id="WordNetDefinitionjs"
                href="WordNetDefinition.js" valuetype="js"></resourceitem>
</resource>
</interface>
```

- ✓ **WordNetDefinition(WordNetDefinition.js)**

```
function UploadButton_onclick(){
    var uploadFile=getValue('WordNetFile');
    var wordnet_id=getIdOfSelectedRow('WordNetDefinitionGrid');
    $.blockUI({message: 'Starting upload of file'});
    UploadWordNet.uploadWordNetXML(uploadFile,wordnet_id,function(data)
        {
            $.unblockUI();
            setReloadGrid('WordNetDefinitionGrid');
            $.blockUI({message: data});
            setTimeout($.unblockUI, 1000);
        });
}
function DownloadButton_onclick(){
    var wordnet_id=getIdOfSelectedRow('WordNetDefinitionGrid');
    if(wordnet_id!=0){
        $.blockUI({message: 'Starting download of WordNet file'});
        UploadWordNet.downloadWordNetXML(wordnet_id,function(data) {
            $.unblockUI();
            if(data)
            {
                window.open(data);
            }
        });
    }
    else {
        alert("Please Select a wordnet from the grid");
        return;
    }
}
```

```
function DeleteButton_onclick(){
    var wordnet_id=getIdOfSelectedRow('WordNetDefinitionGrid');
    if(wordnet_id!=0){
        $.blockUI({message: 'Starting download of WordNet file'});
        UploadWordNet.deleteWordNetDetails(wordnet_id,function(data) {
            $.unblockUI();
            setReloadGrid('WordNetDefinitionGrid');
            $.blockUI({message: data});
            setTimeout($.unblockUI, 1000);
        });
    }
    else {
        alert("Please Select a wordnet from the grid");
        return;
    }}
```

- ✓ **WordNetDefinition Java Code(UploadWordNet.java)**

```
package wordnet;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintStream;
import java.util.zip.*;

import javax.activation.MimetypesFileTypeMap;
import javax.xml.parsers.ParserConfigurationException;

import org.apache.xerces.parsers.DOMParser;
import org.directwebremoting.io.FileTransfer;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
```

```
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

import comv2.aunwesha.lfutil.Triple;

import org.w3c.dom.CharacterData;

import interfaceenginev2.FileDownloaderPojo;
import interfaceenginev2.FileUploaderPojo;
import learnityInit.Host;
import learnityInit.util.FolderZiper;

public class UploadWordNet
{
    public static String uploadWordNetXML(FileTransfer file, String wordnet_id) throws
    ParserConfigurationException, SAXException, IOException
    {
            String html = "Error in uploading wordnet file";
            String dirName= "";
            String strLocation = Host.getWordNetPath();
            dirName=strLocation;
            File ufile=null;
            if(dirName == null) {
                    html="Please supply uploadDir parameter";
                    return html;
            }

            ufile = new File(dirName);
            if(!ufile.exists())
                    ufile.mkdir();

            FileOutputStream out;
            PrintStream p;
```

```
        FileUploaderPojo fu = new FileUploaderPojo(file);

        InputStream fileInputStream = fu.getInputStream();
        String fileName = fu.getFilename();
        String fileType = fu.getMimeType();
        long fileSize = fu.getFileSize();
        dirName = dirName+File.separator+fileName;

        try {
                out = new FileOutputStream(dirName);

                int length = 0;
                byte b[]= new byte[1024];

                while ((fileInputStream != null) && ((length = fileInputStream.read(b))
!= -1))
                {

                        out.write(b,0,length);
                }

                p = new PrintStream( out );

                p.close();

                String zipFileLocation = dirName;
                System.out.println(zipFileLocation);
                String zipType = fileType;
                String filepath = FolderZiper.decompressGzip(zipFileLocation,
Host.getWordNetPath(),zipType);

                try {
                        fileType =
MimetypesFileTypeMap.getDefaultFileTypeMap().getContentType(filepath);
```

```
                } catch (final Exception exception) {
                   return "<EXCEPTION: " + exception.getMessage() + ">";
                }
                String id = wordnet_id;
                boolean flag = DataBaseLayer.saveWordNetDetails(id,filepath);

                        html = "WordNet uploaded succesfully";

        } catch(Exception e) {
                e.printStackTrace();
        }

        return html;
    }
public static String deleteWordNetDetails(String wordnet_id) {
    String html = "Error in deleting wordnet";
    boolean flag = DataBaseLayer.deleteWordNetDetails(wordnet_id);
    if(flag==true)
        html = "WordNet deleted successfully.";
    return html;
    }}
```

- ✓ **WordNetTest Interface(interface.xml)**

```
<interface
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://learnityframework.org/"
    xsi:schemaLocation="http://learnityframework.org/
    http://localhost:8181/learnitymsme/xsd/Interface.xsd"
    id="WordNetTest" title="Test" type="Interface">

    <configuration checkrole="false"></configuration>

<structure>
        <part id="BodyDiv" class="label"></part>
```

```
        <part id="WordnetForm" class="form"></part>
        <part id="WordnetFormFieldset" class="fieldset"></part>
        <part id="WordnetFormLegend" class="legend"></part>

    <part id="WordNetSelectorRowContainer" class="label"></part>
        <part id="WordNetTitle" class="label"></part>
        <part id="ChooseLanguage" class="combo"></part>

        <part id="DisplayOptionRowContainer" class="label"></part>
        <part id="DisplayOptionLabel" class="label"></part>
        <part id="SelectTypeValueOption" class="combo"></part>

        <part id="WordInputRowContainer" class="label"></part>
        <part id="WordSearchValueLabel" class="label"></part>
        <part id="WordSearchValue" class="inputtext"></part>
        <part id="SearchButtonRowContainer" class="label"></part>
        <part id="SearchButton" class="button"></part>
        <part id="ViewResultRowContainer" class="label"></part>
        <part id="ResultLabel" class="label"></part>
        <part id="ResultValue" class="textarea" rows="5" cols="20"></part>

        <part id="WordSynsetRowContainer" class="label"></part>
        <part id="WordSynsetValueLabel" class="label"></part>
        <part id="WordValue1" class="inputtext"></part>
        <part id="WordSynsetRowContainer1" class="label"></part>
        <part id="WordSynsetValueLabel1" class="label"></part>
        <part id="WordValue2" class="inputtext"></part>
        <part id="SearchSimilarityButtonRowContainer" class="label"></part>
        <part id="GetSimilarityButton" class="button"></part>

        <part id="WordNetSimilarityMethodRowContainer" class="label"></part>
        <part id="WordNetSimilarity" class="label"></part>
        <part id="ChooseMethod" class="combo"></part>
```

```
        <part id="ViewResultRowContainer1" class="label"></part>
        <part id="ResultLabel1" class="label"></part>
        <part id="ResultValue1" class="textarea" rows="5" cols="20"></part>

        <part id="ClearButtonRowContainer" class="label"></part>
        <part id="ClearResultsButton" class="button"></part>
        <part id="WordNetTestAjaxComp" class="ajaxcomponent"></part>
</structure>
<layout id="WordNetTestLayout">
        <part id="BodyDiv"></part>
        <part id="WordnetForm" parent_id="BodyDiv"></part>
        <part id="WordnetFormFieldset" parent_id="WordnetForm"></part>
        <part id="WordnetFormLegend" parent_id="WordnetFormFieldset"></part>

        <part id="WordNetSelectorRowContainer"
parent_id="WordnetFormFieldset"></part>
        <part id="WordNetTitle" parent_id="WordNetSelectorRowContainer"></part>
        <part id="ChooseLanguage"
parent_id="WordNetSelectorRowContainer"></part>
        <part id="DisplayOptionRowContainer"
parent_id="WordNetSelectorRowContainer"></part>
        <part id="DisplayOptionLabel"
parent_id="DisplayOptionRowContainer"></part>
        <part id="SelectTypeValueOption"
parent_id="DisplayOptionRowContainer"></part>
        <part id="WordInputRowContainer" parent_id="WordnetFormFieldset"></part>
        <part id="WordSearchValueLabel"
parent_id="WordInputRowContainer"></part>
        <part id="WordSearchValue" parent_id="WordInputRowContainer"></part>
        <part id="SearchButtonRowContainer"
parent_id="WordInputRowContainer"></part>
        <part id="SearchButton" parent_id="SearchButtonRowContainer"></part>

        <part id="ViewResultRowContainer" parent_id="WordnetFormFieldset"></part>
```

```
        <part id="ResultLabel" parent_id="ViewResultRowContainer"></part>
        <part id="ResultValue" parent_id="ViewResultRowContainer"></part>

        <part id="WordSynsetRowContainer"
parent_id="WordnetFormFieldset"></part>
        <part id="WordSynsetValueLabel"
parent_id="WordSynsetRowContainer"></part>
        <part id="WordValue1" parent_id="WordSynsetRowContainer"></part>
        <part id="WordSynsetRowContainer1"
parent_id="WordnetFormFieldset"></part>
        <part id="WordSynsetValueLabel1"
parent_id="WordSynsetRowContainer1"></part>
        <part id="WordValue2" parent_id="WordSynsetRowContainer1"></part>
        <part id="SearchSimilarityButtonRowContainer"
parent_id="WordnetFormFieldset"></part>
        <part id="GetSimilarityButton"
parent_id="SearchSimilarityButtonRowContainer"></part>
        <part id="WordNetSimilarityMethodRowContainer"
parent_id="SearchSimilarityButtonRowContainer"></part>
        <part id="WordNetSimilarity"
parent_id="WordNetSimilarityMethodRowContainer"></part>
        <part id="ChooseMethod"
parent_id="WordNetSimilarityMethodRowContainer"></part>

        <part id="ViewResultRowContainer1"
parent_id="WordnetFormFieldset"></part>
        <part id="ResultLabel1" parent_id="ViewResultRowContainer1"></part>
        <part id="ResultValue1" parent_id="ViewResultRowContainer1"></part>

        <part id="ClearButtonRowContainer"
parent_id="WordnetFormFieldset"></part>
        <part id="ClearResultsButton" parent_id="ClearButtonRowContainer"></part>
        <part id="WordNetTestAjaxComp" parent_id="BodyDiv"></part>
```

```
</layout>
<content id="WordNetTestContent">

<part id="WordnetFormLegend" value="Wordnet Search" valuetype="inline"></part>
<part id="WordNetTitle" value="Select Wordnet: " valuetype="inline"></part>
        <part id="ChooseLanguage" value="select wordnet_id, wordnet_title from
WORDNET" valuetype="query"></part>

        <part id="WordSearchValueLabel" value="Word to search for: "
valuetype="inline"></part>
        <part id="SearchButton" value="Search" valuetype="inline"></part>
        <part id="DisplayOptionLabel" value="Select Type: "
valuetype="inline"></part>
        <!--<part id="ChangeButton" value="Change" valuetype="inline"></part>-->
        <part id="SelectTypeValueOption" value="searchtype" valuetype="fixed">
                <combooption name="Synonym" value="Synonym"></combooption>
                <combooption name="Hyponym" value="Hyponym"></combooption>
                <combooption name="Hypernym" value="Hypernym"></combooption>

        </part>
 <part id="ResultLabel" value="Results:" valuetype="inline"></part>

        <part id="WordSynsetValueLabel" value="Word Synset 1: "
valuetype="inline"></part>
        <part id="WordSynsetValueLabel1" value="Word Synset 2: "
valuetype="inline"></part>
        <part id="GetSimilarityButton" value="Get Similarity"
valuetype="inline"></part>
        <part id="WordNetSimilarity" value="Methods" valuetype="inline"></part>
        <part id="ResultLabel1" value="Results:" valuetype="inline"></part>
        <part id="ChooseMethod" value="searchtype" valuetype="fixed">
                <combooption name="WuPalmer" value="WuPalmer"></combooption>
                <combooption name="Edge-based" value="Edge-
based"></combooption>
```

```
                <combooption name="wup_similarity"
value="wup_similarity"></combooption>

        </part>
        <part id="ClearResultsButton" value="Clear Results" valuetype="inline"></part>

</content>
<style id="WordNetTestStyle">
   <part id="BodyDiv"  valuetype="reference" value="FlexColumnContainer"></part>
        <part id="WordnetFormFieldset"  valuetype="reference"
value="FlexRowContainer"></part>
        <part id="WordnetFormFieldset"  valuetype="inline"
value="[CDATA[width:85%;]]"></part>
        <part id="WordNetSelectorRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
        <part id="WordNetSelectorRowContainer"  valuetype="inline"
value="[CDATA[width:90%;]]"></part>
        <part id="WordInputRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
        <part id="WordInputRowContainer"  valuetype="inline"
value="[CDATA[width:50%;]]"></part>
        <part id="DisplayOptionRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
        <part id="DisplayOptionRowContainer"  valuetype="inline"
value="[CDATA[width:40%;]]"></part>
        <part id="ViewResultRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
        <part id="ViewResultRowContainer"  valuetype="inline"
value="[CDATA[width:40%;]]"></part>
        <part id="WordSynsetRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
        <part id="WordSynsetRowContainer"  valuetype="inline"
value="[CDATA[width:80%;]]"></part>
```

```
        <part id="WordSynsetRowContainer1"  valuetype="reference"
value="FlexRowContainer"></part>
        <part id="WordSynsetRowContainer1"  valuetype="inline"
value="[CDATA[width:80%;]]"></part>
        <part id="SearchSimilarityButtonRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
        <part id="SearchSimilarityButtonRowContainer"  valuetype="inline"
value="[CDATA[width:90%;]]"></part>
        <part id="WordNetSimilarityMethodRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
        <part id="WordNetSimilarityMethodRowContainer"  valuetype="inline"
value="[CDATA[width:40%;]]"></part>
        <part id="ViewResultRowContainer1"  valuetype="reference"
value="FlexRowContainer"></part>
        <part id="ViewResultRowContainer1"  valuetype="inline"
value="[CDATA[width:40%;]]"></part>

        <part id="ClearButtonRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
        <part id="ClearButtonRowContainer"  valuetype="inline"
value="[CDATA[width:30%;]]"></part>
</style>

<behaviour id="WordNetTestBehaviour">
        <part id="SearchButton">
                <event name="onclick" type="simple" function="SearchButton_onclick"
                        valuetype="jsevent" resourceid="WordNetTestjs"></event>
        </part>
        <part id="GetSimilarityButton">
                <event name="onclick" type="simple"
function="GetSimilarityButton_onclick"
                        valuetype="jsevent" resourceid="WordNetTestjs"></event>
        </part>
        <part id="ClearResultsButton">
```

```
            <event name="onclick" type="simple"
function="ClearResultsButton_onclick"
                valuetype="jsevent" resourceid="WordNetTestjs"></event>
        </part>
        <part id="WordNetTestAjaxComp">
            <event name="ajax" type="simple" function="dummy"
javaclass="WordNetTest"></event>
        </part>
</behaviour>

<resource>
        <resourceitem id="WordNetTestjs" href="WordNetTest.js"
valuetype="js"></resourceitem>
</resource>
</interface>
```

- ✓ **WordNetTest(WordNetTest.js)**

```
function SearchButton_onclick(){
    var wordnet_title=getValue('ChooseLanguage');
    var word = getValue('WordSearchValue');
    if(wordnet_title!=0){
        var search_type=getValue('SelectTypeValueOption');
        if(search_type!=0){
            $.blockUI({message: 'finding chosen search_type'});

        WordNetTest.getWordNetSearchFile(wordnet_title,word,search_type,function(da
    ta) {
                $.unblockUI();
                setValue('ResultValue',data);
            })
        }
        else{
            alert("Please choose select type from the dropdown above");
        }
```

```
    }
    else{
        alert("Please Select an wordnet from the dropdown above");
    }
}
function ClearResultsButton_onclick(){
    setValue('ResultValue','');
    }
```

✓ Backend Code (WordNetTest.java)

```
package wordnet;
import java.io.BufferedReader;
import java.io.EOFException;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.net.URL;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Iterator;
import java.util.LinkedHashSet;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import java.util.ResourceBundle;
import java.util.Map.Entry;
import javax.xml.bind.JAXBContext;
import javax.xml.bind.JAXBException;
import javax.xml.bind.Unmarshaller;
import bengali.dictionary.DictionaryBen;
import org.apache.logging.log4j.Logger;
import org.apache.logging.log4j.LogManager;
import java.util.Set;
```

```
import java.util.TreeMap;
import comv2.aunwesha.lfutil.Pair;
import comv2.aunwesha.lfutil.Pentuple;
import comv2.aunwesha.lfutil.Quadruple;
import comv2.aunwesha.lfutil.Triple;
import edu.mit.jwi.Dictionary;
import edu.mit.jwi.IDictionary;
import edu.mit.jwi.data.parse.SenseKeyParser;
import edu.mit.jwi.item.IIndexWord;
import edu.mit.jwi.item.ISenseKey;
import edu.mit.jwi.item.ISynset;
import edu.mit.jwi.item.ISynsetID;
import edu.mit.jwi.item.IWord;
import edu.mit.jwi.item.IWordID;
import edu.mit.jwi.item.POS;
import edu.mit.jwi.item.Pointer;
import learnityInit.Host;
import net.sf.json.JSONObject;
import vu.wntools.lmf.Synset;
import vu.wntools.wordnet.WordNetLmfData;
import vu.wntools.wordnet.WordnetLmfDataSaxParser;
import vu.wntools.wordnet.WordnetLmfSaxParser;
import wolfapi.WnQuery;
import wolfapi.WNModel.LITERAL;
import wolfapi.WNModel.SYNSET;
import wolfapi.WNModel.WN;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.nio.file.Paths;

public class WordNetTest
{
    static Logger strLogger = LogManager.getLogger(WordNetTest.class);
```

```
public IDictionary dictionary = null;
public String getWordNetSearchFile(String wordnet_title,String word,String
search_type) {
        long startTime = System.nanoTime();
        //String strLocation = Host.getWordNetPath();
        //String path = strLocation + File.separator + "dict";
        String html = "Error in word Search!";
        String flag = null;
        Map<String,LinkedHashSet<String>> synonymMap = new
TreeMap<String,LinkedHashSet<String>>(String.CASE_INSENSITIVE_ORDER);
        Map<String,LinkedHashSet<String>> hypernymMap = new
TreeMap<String,LinkedHashSet<String>>(String.CASE_INSENSITIVE_ORDER);
        Map<String,LinkedHashSet<String>> hyponymMap = new
TreeMap<String,LinkedHashSet<String>>(String.CASE_INSENSITIVE_ORDER);
        LinkedHashSet<String> valSynonym = new LinkedHashSet<String>();
    LinkedHashSet<String> valHypernym = new LinkedHashSet<String>();
    LinkedHashSet<String> valHyponym = new LinkedHashSet<String>();
        LinkedHashSet<String> synonym_text = new LinkedHashSet<String>();
    LinkedHashSet<String> hypernym_text = new LinkedHashSet<String>();
    LinkedHashSet<String> hyponym_text = new LinkedHashSet<String>();
    HashMap<String, String> mapwords_beng;
        try {
                        String wordnet_fileType = null;
                        Triple<String,String,String>  listNameWordNetTriple =
wordnet.DataBaseLayer.getWordnetTitle(wordnet_title);
                        wordnet_fileType = listNameWordNetTriple.getSecond();
                        System.out.println(wordnet_fileType);
                        switch(wordnet_fileType) {
                        case"PWN":
                                String getPath = listNameWordNetTriple.getFirst();
                                URL url = new URL("file", null, getPath);
                                System.out.println(url);
                           // construct the dictionary object and open it
                           dictionary = new Dictionary(url);
```

```
                dictionary.open();

                        IIndexWord idxWord = dictionary . getIndexWord
(word, POS . NOUN ) ;
                IWordID wordID = idxWord . getWordIDs () . get (0) ; // 1st
meaning
                IWord word1 = dictionary . getWord ( wordID ) ;
                ISynset synset = word1 . getSynset () ;
                switch(search_type) {
                case"Synonym":
                        // iterate over words associated with the synset
                        for( IWord w : synset.getWords ()) {
                      System.out.println ( w.getLemma () ) ;
                        synonym_text.add(w.getLemma().toLowerCase());
                                valSynonym.add(w.getLemma());
                                synonymMap.put(word, valSynonym);
                                String strTokenSyn = "";
                           for (String strSynonym : synonym_text) {
                                        strTokenSyn += strSynonym+", ";
                                        }
                           strTokenSyn = strTokenSyn.replaceAll(", $", "");
                                flag=strTokenSyn;
                }
                        break;
                case"Hyponym":
                         List<ISynsetID> hyponyms =
synset.getRelatedSynsets(Pointer.HYPONYM);
                                List<IWord> hypowords;
                                for (ISynsetID sid : hyponyms) {
                                        hypowords =
dictionary.getSynset(sid).getWords();
                                  for (Iterator<IWord> i =
hypowords.iterator(); i.hasNext();) {
                                        String hyponym = i.next().getLemma();
```

```
hyponym_text.add(hyponym.toLowerCase());
                                        valHyponym.add(hyponym);
                                    }
                                   }
                            hyponymMap.put(word, valHyponym);
                            String strTokenHypo = "";
                            for (String strHyponym : hyponym_text) {
                                        strTokenHypo += strHyponym+", ";
                                        }
                            strTokenHypo = strTokenHypo.replaceAll(", $", "");
                            flag=strTokenHypo;
                            break;
                        case"Hypernym":
                                    // get the hypernyms
                                    List < ISynsetID > hypernyms =
                                    synset.getRelatedSynsets ( Pointer.HYPERNYM
) ;

                                    // print out each hypernyms id and synonyms
                                    List < IWord > words ;
                                    for( ISynsetID sid : hypernyms ) {
                                    words = dictionary . getSynset ( sid ) . getWords
() ;
                                    System . out . print ( sid + " {") ;
                                    for( Iterator < IWord > i = words . iterator () ; i .
hasNext () ;) {
                                    System . out . print ( i . next () . getLemma () ) ;
                                    if( i . hasNext () )
                                    System.out.print (", ") ;
                                     String hypernym = i.next().getLemma();
                                      //hypernym_text += hypernym+",";
                                      hypernym_text.add(hypernym.toLowerCase());
```

```
                                        //hypernym_total_text += "{
"+corpusFileSentenceTokenText+" : "+hypernym+" } , ";
                                         valHypernym.add(hypernym);
                                        }
                                        System . out . println ("}") ;
                                        }
                                        hypernymMap.put(word, valHypernym);
                                        String strTokenHyper = "";
                                        for (String strHypernym : hypernym_text) {
                                                strTokenHyper += strHypernym+", ";
                                                }
                                        strTokenHyper = strTokenHyper.replaceAll(",
$", "");
                                        flag=strTokenHyper;
                                 break;
                         }
                         break;
                        case"DebVisDic":
                                String getPath1 = listNameWordNetTriple.getFirst();
                                URL url1 = new URL("file", null, getPath1);
                                WnQuery wq = new WnQuery();
                                wq.getWnFromXml();
                                Set<SYNSET> res = wq.querySynonym(word,"n");
                                String strTokensSyn = "";
                                for (SYNSET syns : res) {
                                        List<LITERAL> literal
=syns.getSYNONYM().getLITERAL();
                                        for (LITERAL lit: literal) {
                                                strTokensSyn += lit.getvalue()+", ";
                                        }
                                        //strTokensSyn += syns.toString()+", ";
                                }
                                WnQuery.printSynsetCollection(res);
                                strTokensSyn = strTokensSyn.replaceAll(", $", "");
```

```
                    flag =strTokensSyn;
                    break;
                case"XML":
                    String getPath2 = listNameWordNetTriple.getFirst();
                    URL url2 = new URL("file", null, getPath2);
                    System.out.println(url2);
                    String strTokensSyn1 = "";

                    WordNetLmfData fileXML = new WordNetLmfData();

                    WordnetLmfSaxParser Parser = new
WordnetLmfSaxParser();
                Parser.parseFile(getPath2);
                Parser.wordnetData.buildSynsetIndex();

                if (Parser.wordnetData.entryToSynsets.containsKey(word)) {
                for (int i = 0; i < fileXML.getSynsets().size(); i++) {
                   Synset sense = fileXML.getSynsets().get(i);
                   String synsetID= sense.getSynsetId();
                   ArrayList<String> synonyms =
Parser.wordnetData.getSynonyms(synsetID);
                        for (int j = 0; j < synonyms.size(); j++) {
                    String synonym = synonyms.get(j);
                    ArrayList<String> synsetIds =
Parser.wordnetData.lemmaToSynsets.get(synonym);
                    if (synsetIds!=null) {
                       for (int k = 0; k < synsetIds.size(); k++) {
                          String id = synsetIds.get(k);
                                //  System.out.println("id = " + id);
                          if (id.startsWith("oewn")) {
                             strTokensSyn1 += id+", ";
                          flag = strTokensSyn1;
                          break;
                                }}}}
```

```
                    }}
                flag=strTokensSyn1;
                        break;
                case"Bengali-Dict":
                        String getPath11 = listNameWordNetTriple.getFirst();
                        File file1 = new
File("D:/LKE_Resources/wordnetpath/BengaliDictionary/BengaliDictionary_17.txt");
                        String fileName = file1.getPath();
                        //URL url11 = new URL("file", null, fileName);
                        System.out.println(fileName);
                        try {
                                DictionaryBen dictionary = new
DictionaryBen(fileName);
                                flag = dictionary.search(word);
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                        break;
                }
            }catch(IOException e){

        }
        return flag;
}
}
```

- ✓ **Wordnet Database Code (DataBaseLayer.java)**

```
package wordnet;
import java.io.ByteArrayOutputStream;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
```

```
import javax.sql.DataSource;
import comv2.aunwesha.lfutil.Triple;
//import comv2.aunwesha.lfutil.Triple;
import comv2.aunwesha.param.CoreAdminInitHostInfo;

public class DataBaseLayer
{
    // database connection settings
    public static DataSource ds1 = CoreAdminInitHostInfo.ds1;

   public static boolean saveWordNetDetails(String id,String filepath)
   {
    boolean flag = false;
     try(Connection oConn = ds1.getConnection())
     {
                    String sql = "Update WORDNET set wordnet_file_path=? where
    wordnet_id=?" ;
         PreparedStatement statement = oConn.prepareStatement(sql);
         statement.setString(1, filepath);
        statement.setString(2, id);
        statement.executeUpdate();
        statement.close();
        flag = true;
     }
     catch(SQLException e)
     {
          e.printStackTrace();
     }
     catch (Exception e)
     {
                    e.printStackTrace();
          }
     return flag;
   }
```

```
public static boolean deleteWordNetDetails(String wordnet_id) {

        boolean flag = false;

        try(Connection oConn = ds1.getConnection()) {

                String sql1 = "delete from WORDNET where wordnet_id = ?";
                PreparedStatement pstmt = oConn.prepareStatement(sql1);
                pstmt.setString(1,wordnet_id);
                pstmt.executeUpdate();

                pstmt.close();
                flag = true;
        }
        catch (SQLException e) {
                        System.out.println("==SQLException===");
                        e.printStackTrace();
        }
        catch (Exception ex) {
                        System.out.println("==Exception===");
                        ex.printStackTrace();
        }
        return flag;
  }
public String getSynonyms(String resourcetype, String language)
  {
  boolean flag = false;

        try(Connection oConn = ds1.getConnection()) {
                String sql1 = "Select wordnet_id,
  wordnet_title,wordnet_file_type,wordnet_lang from WORDNET where
  wordnet_file_type = ? and wordnet_lang= ?";

        }catch(SQLException e) {}
```

```
        return getSynonyms(resourcetype,language);
    }
public synchronized static Triple<String,String,String>  getWordnetTitle(String
    wordnet_id) {

        Triple<String,String,String>  listNameWordNetTriple = new
    Triple<String,String,String>();

        try(Connection oConn = ds1.getConnection()) {

                String sql = "Select wordnet_file_path,wordnet_file_type,wordnet_title
    from WORDNET where wordnet_id = ?";
                System.out.println("Select
    wordnet_file_path,wordnet_file_type,wordnet_title from WORDNET where
    wordnet_id='"+wordnet_id+"'");

                PreparedStatement pstmt = oConn.prepareStatement(sql);
                pstmt.setString(1,wordnet_id);
                ResultSet resultSet = pstmt.executeQuery();
                while (resultSet.next()) {
                        listNameWordNetTriple.setFirst(resultSet.getString(1));
                        listNameWordNetTriple.setSecond(resultSet.getString(2));
                        listNameWordNetTriple.setThird(resultSet.getString(3));
    }
                resultSet.close();
                pstmt.close();
        }
        catch (SQLException e) {
                e.printStackTrace();
        }
        catch (Exception ex) {
                ex.printStackTrace();
        }
```

```
        return listNameWordNetTriple;
    }}
```

❖ Code Snippets for Ontology:

✓ Ontology Interface(interface.xml)

```
<interface
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://learnityframework.org/"
    xsi:schemaLocation="http://learnityframework.org/
    http://localhost:8181/learnitymsme/xsd/Interface.xsd"
    id="Ontology" title="LoadOntology" type="Interface">
    <configuration checkrole="false"></configuration>

    <structure>
        <part id="BodyDiv" class="label"></part>
        <part id="OntologyGrid" class="DBgrid" caption="Ontology"
    sortname="ontology_id" sortorder="desc" autoencode="false" generateExcel="true">
            <column head="Ontology_id" name="ontology_id" index="1" width="0"
    editable="true" hidden="true" key="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="Ontology title" name="ontology_title" index="2"
    width="60" editable="true" hidden="false" key="false" required="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="Ontology description" name="ontology_description"
    index="3" width="70" editable="true" hidden="false" key="false" required="false">
                <edit type="textarea" cols="28" rows="5"></edit>
            </column>
            <column head="Ontology iri" name="ontology_iri" index="4"
    width="55" editable="true" hidden="false" key="false" required="false">
                <edit type="text" size="30"></edit>
            </column>
            <column head="Ontology file type" name="ontology_file_type"
    index="5" width="70" editable="true" hidden="false" key="false" required="true">
```

<edit type="select" editdomaintype="fixed" value="SADL:SADL;OWL/XML:OWL/XML;RDF/XML:RDF/XML;TURTLE:TURTLE;OTHER:OTHER" size="30"></edit>
</column>
<column head="Ontology metadata" name="ontology_metadata" index="6" width="65" editable="true" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>
<column head="Ontology file" name="ontology_file" index="7" width="60" editable="true" hidden="false" key="false" required="false">
<edit type="file" size="30"></edit>
</column>
<column head="Num of classes" name="number_of_classes" index="8" width="55" editable="true" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>
<column head="Num of obj properties" name="number_of_object_properties" index="9" width="70" editable="true" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>
<column head="Num of datatype properties" name="number_of_datatype_properties" index="10" width="70" editable="true" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>
<column head="Num of individuals" name="number_of_individuals" index="11" width="65" editable="true" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>
<column head="Date ontology saved" name="date_ontology_saved" index="12" width="70" editable="false" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>

```
<column head="Comment" name="comment" index="13" width="60"
editable="true" hidden="false" key="false" required="false">
<edit type="textarea" size="30"></edit>
</column>
<loadquery
value="select
ontology_id,ontology_title,ontology_description,ontology_iri,ontology_file_type,ontolo
gy_metadata,ontology_file,number_of_classes,number_of_object_properties,number_of
_datatype_properties,number_of_individuals,date_ontology_saved,comment from
ontology"></loadquery>
<delete>
<queries>
<query id="1"
sql="[CDATA[delete FROM ontology where
ontology_id=%ontology_id%]]"
parameter="ontology_id"></query>
</queries>
</delete>
<add>
<queries>
<query id="1"
sql="[CDATA[insert into ontology
(ontology_title,ontology_file,ontology_file_type,comment) values
(%ontology_title%,%ontology_file%,%ontology_file_type%,%comment%)]]"

parameter="ontology_title,ontology_file,ontology_file_type,comment"></query>
</queries>
</add>
<modify>
<queries>
<query id="1"
sql="[CDATA[update ontology set
ontology_title=%ontology_title%,
```

```
ontology_description=%ontology_description%,ontology_file_type=%ontology_file_ty
pe%,comment=%comment% where ontology_id=%ontology_id%]]"

        parameter="ontology_title,ontology_description,ontology_file_type,comment"></
query>
                        </queries>
                </modify>
        </part>

        <part id="UploadFileRowContainer" class="label"></part>
        <part id="OntologyFileLabel" class="label"></part>
        <part id="OntologyFile" class="inputfile"></part>
        <part id="UploadButton" class="button"></part>
        <part id="DownloadButton" class="button"></part>
        <part id="DeleteButton" class="button"></part>
        <part id="OntologyAjaxComp" class="ajaxcomponent"></part>
</structure>
<layout id="OntologyLayout">
        <part id="BodyDiv"></part>
        <part id="OntologyGrid" height="300" parent_id="BodyDiv"></part>

        <part id="UploadFileRowContainer" parent_id="BodyDiv"></part>
        <part id="OntologyFileLabel" parent_id="UploadFileRowContainer"></part>
        <part id="OntologyFile" parent_id="UploadFileRowContainer"></part>
        <part id="UploadButton" parent_id="UploadFileRowContainer"></part>
        <part id="DownloadButton" parent_id="UploadFileRowContainer"></part>
        <part id="DeleteButton" parent_id="UploadFileRowContainer"></part>
        <part id="OntologyAjaxComp" parent_id="BodyDiv"></part>
</layout>
<content id="OntologyContent">
   <part id="OntologyFileLabel" value="Select the file upload"
valuetype="inline"></part>
        <part id="UploadButton" value="Upload File" valuetype="inline"></part>
        <part id="DownloadButton" value="Download File" valuetype="inline"></part>
```

```
        <part id="DeleteButton" value="Delete Ontology" valuetype="inline"></part>
</content>
<style id="OntologyStyle">
   <part id="BodyDiv" valuetype="reference" value="FlexColumnContainer"></part>
        <part id="BodyDiv" valuetype="reference"
value="FlexColumnContainer"></part>
        <part id="UploadFileRowContainer" valuetype="reference"
value="FlexRowContainer"></part>
        <part id="UploadFileRowContainer" valuetype="inline"
value="[CDATA[width:80%;]]"></part>
</style>
<behaviour id="OntologyBehaviour">
                <part id="UploadButton">
                <event name="onclick" type="simple" function="UploadButton_onclick"
                        valuetype="jsevent" resourceid="Ontologyjs"></event>
        </part>
        <part id="DownloadButton">
                <event name="onclick" type="simple"
function="DownloadButton_onclick"
                        valuetype="jsevent" resourceid="Ontologyjs"></event>
        </part>
        <part id="DeleteButton">
                <event name="onclick" type="simple" function="DeleteButton_onclick"
                        valuetype="jsevent" resourceid="Ontologyjs"></event>
        </part>
        <part id="OntologyAjaxComp">
                <event name="ajax" type="simple" function="dummy"
javaclass="Ontology"></event>
        </part>
</behaviour>
<resource>
        <resourceitem id="Ontologyjs"
                href="Ontology.js" valuetype="js"></resourceitem>
</resource>
```

```
</interface>
```

✓ **Ontology.js**

```
function UploadButton_onclick(){
    var uploadFile=getValue('OntologyFile');
    var ontology_id=getIdOfSelectedRow('OntologyGrid');
    $.blockUI({message: 'Starting upload of file'});
    Ontology.uploadOntologyXML(uploadFile,ontology_id,function(data)
        {
            $.unblockUI();
            setReloadGrid('OntologyGrid');
            $.blockUI({message: data});
            setTimeout($.unblockUI, 1000);
        });
}
function DownloadButton_onclick(){
    var ontology_id=getIdOfSelectedRow('OntologyGrid');
    if(ontology_id!=0){
        $.blockUI({message: 'Starting download of Ontology file'});
        Ontology.downloadOntologyXML(ontology_id,function(data) {
            $.unblockUI();
            if(data)
            {
                window.open(data);
            }
        });
    }
    else {
        alert("Please Select a ontology from the grid");
        return;
    }
}
function DeleteButton_onclick(){
    var ontology_id=getIdOfSelectedRow('OntologyGrid');
```

```
if(ontology_id!=0){
	$.blockUI({message: 'Starting deleting of Ontology file'});
	Ontology.deleteOntologyDetails(ontology_id,function(data) {
		$.unblockUI();
		setReloadGrid('OntologyGrid');
		$.blockUI({message: data});
		setTimeout($.unblockUI, 1000);
	});
}
else {
	alert("Please Select a ontology from the grid");
	return;
}
}
```

- ✓ **OntologyElement Interface(interface.xml)**

```
<interface
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://learnityframework.org/"
    xsi:schemaLocation="http://learnityframework.org/
    http://localhost:8181/learnitymsme/xsd/Interface.xsd"
    id="OntologyElement" title="OntologyElement" type="Interface">

    <configuration checkrole="false"></configuration>

    <structure>
        <part id="BodyDiv" class="label"></part>
        <part id="OntologyElementSelectorRowContainer" class="label"></part>
        <part id="OntologyElementTitle" class="label"></part>

        <part id="OntologyElementGrid" class="Conditionalgrid"
    caption="OntologyElement" sortname="ontology_element_id" sortorder="desc"
    autoencode="false" generateExcel="true">
            <selector id="ontology_id" class="combo"
```

```
                contentid="OntologyElementContent"
                behaviourid="OntologyElementBehaviour" domaintype="query"
                value="select ontology_id,ontology_title from ONTOLOGY"
                gridrefresh="true">
            </selector>
            <column head="ontology element id" name="ontology_element_id"
index="1" width="0" editable="true" hidden="true" key="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology element type" name="ontology_element_type"
index="2" width="60" editable="true" hidden="false" key="false" required="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology element iri" name="ontology_element_iri"
index="3" width="70" editable="true" hidden="false" key="false" required="false">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology element rdfID"
name="ontology_element_rdfID" index="4" width="70" editable="true" hidden="false"
key="false" required="false">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology element label" name="ontology_element_label"
index="5" width="70" editable="true" hidden="false" key="false" required="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology element comment"
name="ontology_element_comment" index="6" width="65" editable="true"
hidden="false" key="false" required="false">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology element content"
name="ontology_element_content" index="7" width="60" editable="true"
hidden="false" key="false" required="false">
```

```
                    <edit type="text" size="30"></edit>
                </column>
                <column head="date ontology element saved"
name="date_ontology_element_saved" index="8" width="70" editable="false"
hidden="false" key="false" required="false">
                    <edit type="text" size="30"></edit>
                </column>
                <column head="comment" name="comment" index="9" width="60"
editable="true" hidden="false" key="false" required="false">
                    <edit type="textarea" size="30"></edit>
                </column>
                <column head="ontology_id" name="ontology_id" index="10"
width="70" editable="true" hidden="false" key="false" required="false">
                    <edit type="text" size="30"></edit>
                </column>

                <loadquery
                    value="select
OM.ontology_element_id,OM.ontology_element_type,OM.ontology_element_iri,OM.o
ntology_element_rdfID,OM.ontology_element_label,OM.ontology_element_comment,
OM.ontology_element_content,OM.date_ontology_element_saved,OM.comment,OM.o
ntology_id from ontology_element OM where
OM.ontology_id=%ontology_id%"></loadquery>

                <delete>
                    <queries>
                        <query id="1"
                            sql="[CDATA[delete FROM ontology_element
where ontology_element_id=%ontology_element_id%]]"
                            parameter="ontology_element_id"></query>
                    </queries>
                </delete>
        </part>
</structure>
```

```
<layout id="OntologyElementLayout">
        <part id="BodyDiv"></part>
        <part id="OntologyElementSelectorRowContainer"
parent_id="BodyDiv"></part>
        <part id="OntologyElementTitle"
parent_id="OntologyElementSelectorRowContainer"></part>
        <part id="ontology_id"
parent_id="OntologyElementSelectorRowContainer"></part>
        <part id="OntologyElementGrid" height="300" parent_id="BodyDiv"></part>
</layout>
<content id="OntologyElementContent">
   <part id="OntologyElementTitle" value="Select Ontology: "
valuetype="inline"></part>
</content>
<style id="OntologyElementStyle">
   <part id="BodyDiv" valuetype="reference" value="FlexColumnContainer"></part>
        <part id="OntologyElementSelectorRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
        <part id="OntologyElementSelectorRowContainer"  valuetype="inline"
value="[CDATA[width:90%;]]"></part>
</style>
<behaviour id="OntologyElementBehaviour">
</behaviour>
</interface>
```

- ✓ **OntologyClass Interface(interface.xml)**

```
<interface
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://learnityframework.org/"
    xsi:schemaLocation="http://learnityframework.org/
    http://localhost:8181/learnitymsme/xsd/Interface.xsd"
    id="OntologyClass" title="OntologyClass" type="Interface">

    <configuration checkrole="false"></configuration>
```

```
<structure>
	<part id="BodyDiv" class="label"></part>
	<part id="OntologySelectorRowContainer" class="label"></part>
	<part id="OntologyTitleLabel" class="label"></part>
	<part id="OntologyClassGrid" class="Conditionalgrid" caption="OntologyClass"
sortname="ontology_class_id" sortorder="desc" autoencode="false"
generateExcel="true">

		<selector id="ontology_id" class="combo"
			contentid="OntologyClassContent"
			behaviourid="OntologyClassBehaviour" domaintype="query"
			value="select ontology_id,ontology_title from ONTOLOGY"
			gridrefresh="true">
		</selector>
		<column head="Ontology_class_id" name="ontology_class_id"
index="1" width="0" editable="true" hidden="true" key="true">
			<edit type="text" size="30"></edit>
		</column>
		<column head="ontology_class_type" name="ontology_class_type"
index="2" width="120" editable="true" hidden="false" key="false" required="true">
			<edit type="text" size="30"></edit>
		</column>
		<column head="ontology_element_id" name="ontology_element_id"
index="3" width="160" editable="true" hidden="false" key="false" required="false">
			<edit type="text" size="30"></edit>
		</column>

		<column head="date_ontology_class_saved"
name="date_ontology_class_saved" index="4" width="170" editable="false"
hidden="false" key="false" required="false">
			<edit type="text" size="30"></edit>
		</column>
```

```
                <column head="Comment" name="comment" index="5" width="60"
editable="true" hidden="false" key="false" required="false">
                        <edit type="textarea" size="30"></edit>
                </column>


                <loadquery
                        value="select
OC.ontology_class_id,OC.ontology_class_type,OC.ontology_element_id,OC.date_ontol
ogy_class_saved,OC.comment from ONTOLOGY_CLASS OC, ONTOLOGY O,
ONTOLOGY_ELEMENT E where OC.ontology_element_id = E.ontology_element_id
and E.ontology_id=%ontology_id% "></loadquery>

                <delete>
                        <queries>
                                <query id="1"
                                        sql="[CDATA[delete FROM ontology_class
where ontology_class_id=%ontology_class_id%]]"
                                        parameter="ontology_class_id"></query>
                        </queries>
                </delete>
        </part>
</structure>
<layout id="OntologyClassLayout">
        <part id="BodyDiv"></part>
        <part id="OntologySelectorRowContainer" parent_id="BodyDiv"></part>
        <part id="OntologyTitleLabel"
parent_id="OntologySelectorRowContainer"></part>
        <part id="ontology_id" parent_id="OntologySelectorRowContainer"></part>
        <part id="OntologyClassGrid" height="300" parent_id="BodyDiv"></part>

</layout>
<content id="OntologyClassContent">
   <part id="OntologyTitleLabel" value="Select ontology:" valuetype="inline"></part>
```

```
</content>
<style id="OntologyClassStyle">
   <part id="BodyDiv" valuetype="reference" value="FlexColumnContainer"></part>

        <part id="OntologySelectorRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
        <part id="OntologySelectorRowContainer"  valuetype="inline"
value="[CDATA[width:90%;]]"></part>
</style>
<behaviour id="OntologyClassBehaviour">
</behaviour>
</interface>
```

- ✓ **OntologyDataProperty Interface(interface.xml)**

```
<interface
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://learnityframework.org/"
   xsi:schemaLocation="http://learnityframework.org/
   http://localhost:8181/learnitymsme/xsd/Interface.xsd"
   id="OntologyDataProperty" title="ontology Data Property" type="Interface">

   <configuration checkrole="false"></configuration>
   <structure>
        <part id="BodyDiv" class="label"></part>
        <part id="OntologySelectorRowContainer" class="label"></part>
        <part id="OntologyTitleLabel" class="label"></part>
        <part id="OntologyDPGrid" class="Conditionalgrid" caption="OntologyDP"
   sortname="ontology_data_property_id" sortorder="desc" autoencode="false"
   generateExcel="true">

              <selector id="ontology_id" class="combo"
                   contentid="OntologyDPContent"
                   behaviourid="OntologyDPBehaviour" domaintype="query"
                   value="select ontology_id,ontology_title from ONTOLOGY"
```

```
                gridrefresh="true">
            </selector>
            <column head="ontology_data_property_id"
name="ontology_data_property_id" index="1" width="0" editable="true" hidden="true"
key="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology data property domain"
name="ontology_data_property_domain" index="2" width="70" editable="true"
hidden="false" key="false" required="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology data property range"
name="ontology_data_property_range" index="3" width="70" editable="true"
hidden="false" key="false" required="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology data property REFLEXIVE"
name="ontology_data_property_REFLEXIVE" index="4" width="80" editable="true"
hidden="false" key="false" required="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology data property SYMMETRIC"
name="ontology_data_property_SYMMETRIC" index="5" width="80" editable="true"
hidden="false" key="false" required="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology data property TRANSITIVE"
name="ontology_data_property_TRANSITIVE" index="6" width="80" editable="true"
hidden="false" key="false" required="true">
                <edit type="text" size="30"></edit>
            </column>
```

<column head="ontology data property FUNCTIONAL" name="ontology_data_property_FUNCTIONAL" index="7" width="80" editable="true" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>
<column head="ontology data property INVERSE FUNCTIONAL" name="ontology_data_property_INVERSE_FUNCTIONAL" index="8" width="80" editable="true" hidden="false" key="false" required="true">
<edit type="text" size="30"></edit>
</column>
<column head="ontology data property inverse property id" name="ontology_data_property_inverse_property_id" index="9" width="70" editable="true" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>
<column head="ontology data property equivalent property id" name="ontology_data_property_equivalent_property_id" index="10" width="70" editable="true" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>
<column head="ontology element id" name="ontology_element_id" index="11" width="60" editable="true" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>

<column head="date ontology data property saved" name="date_ontology_data_property_saved" index="12" width="70" editable="false" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>

<column head="Comment" name="comment" index="13" width="60" editable="true" hidden="false" key="false" required="false">
<edit type="textarea" size="30"></edit>

```
            </column>
            <loadquery
                value="select
DP.ontology_data_property_id,DP.ontology_data_property_domain,DP.ontology_data_
property_range,DP.ontology_data_property_REFLEXIVE,DP.ontology_data_property_
SYMMETRIC,DP.ontology_data_property_TRANSITIVE,DP.ontology_data_property_
FUNCTIONAL,DP.ontology_data_property_INVERSE_FUNCTIONAL,DP.ontology_
data_property_inverse_property_id,DP.ontology_data_property_equivalent_property_id,
DP.ontology_element_id,DP.date_ontology_data_property_saved,DP.comment from
ONTOLOGY_DATA_PROPERTY DP, ONTOLOGY O, ONTOLOGY_ELEMENT E
where DP.ontology_element_id = E.ontology_element_id and
E.ontology_id=%ontology_id% "></loadquery>

            <delete>
                <queries>
                    <query id="1"
                        sql="[CDATA[delete FROM
ontology_data_property where
ontology_data_property_id=%ontology_data_property_id%]]"

    parameter="ontology_data_property_id"></query>
                </queries>
            </delete>
    </part>
</structure>

<layout id="OntologyDPLayout">
    <part id="BodyDiv"></part>
    <part id="OntologySelectorRowContainer" parent_id="BodyDiv"></part>
    <part id="OntologyTitleLabel"
parent_id="OntologySelectorRowContainer"></part>
    <part id="ontology_id" parent_id="OntologySelectorRowContainer"></part>
    <part id="OntologyDPGrid" height="300" parent_id="BodyDiv"></part>
</layout>
```

```
<content id="OntologyDPContent">
   <part id="OntologyTitleLabel" value="Select ontology:" valuetype="inline"></part>
</content>
<style id="OntologyDPStyle">
   <part id="BodyDiv" valuetype="reference" value="FlexColumnContainer"></part>
   <part id="BodyDiv"  valuetype="inline" value="[CDATA[width:120%;]]"></part>
        <part id="OntologySelectorRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
        <part id="OntologySelectorRowContainer"  valuetype="inline"
value="[CDATA[width:90%;]]"></part>
</style>
<behaviour id="OntologyDPBehaviour">
</behaviour>
</interface>
```

- ✓ **OntologyObjectProperty Interface(interface.xml)**

```
<interface
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://learnityframework.org/"
   xsi:schemaLocation="http://learnityframework.org/
   http://localhost:8181/learnitymsme/xsd/Interface.xsd"
   id="OntologyObjProperty" title="ontology Object Property" type="Interface">

   <configuration checkrole="false"></configuration>

   <structure>
        <part id="BodyDiv" class="label"></part>
        <part id="OntologySelectorRowContainer" class="label"></part>
        <part id="OntologyTitleLabel" class="label"></part>
        <part id="OntologyOPGrid" class="Conditionalgrid" caption="OntologyOP"
   sortname="ontology_object_property_id" sortorder="desc" autoencode="false"
   generateExcel="true">

                <selector id="ontology_id" class="combo"
                        contentid="OntologyOPContent"
```

```
                behaviourid="OntologyOPBehaviour" domaintype="query"
                value="select ontology_id,ontology_title from ONTOLOGY"
                gridrefresh="true">
            </selector>
            <column head="ontology_object_property_id"
name="ontology_object_property_id" index="1" width="0" editable="true"
hidden="true" key="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology object property domain"
name="ontology_object_property_domain" index="2" width="70" editable="true"
hidden="false" key="false" required="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology object property range"
name="ontology_object_property_range" index="3" width="70" editable="true"
hidden="false" key="false" required="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology object property REFLEXIVE"
name="ontology_object_property_REFLEXIVE" index="4" width="80" editable="true"
hidden="false" key="false" required="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology object property SYMMETRIC"
name="ontology_object_property_SYMMETRIC" index="5" width="80"
editable="true" hidden="false" key="false" required="true">
                <edit type="text" size="30"></edit>
            </column>
            <column head="ontology object property TRANSITIVE"
name="ontology_object_property_TRANSITIVE" index="6" width="80"
editable="true" hidden="false" key="false" required="true">
                <edit type="text" size="30"></edit>
            </column>
```

<column head="ontology object property FUNCTIONAL" name="ontology_object_property_FUNCTIONAL" index="7" width="80" editable="true" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>
<column head="ontology object property INVERSE FUNCTIONAL" name="ontology_object_property_INVERSE_FUNCTIONAL" index="8" width="80" editable="true" hidden="false" key="false" required="true">
<edit type="text" size="30"></edit>
</column>
<column head="ontology element id" name="ontology_element_id" index="9" width="60" editable="true" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>
<column head="date ontology object property saved" name="date_ontology_object_property_saved" index="10" width="70" editable="false" hidden="false" key="false" required="false">
<edit type="text" size="30"></edit>
</column>
<column head="Comment" name="comment" index="11" width="60" editable="true" hidden="false" key="false" required="false">
<edit type="textarea" size="30"></edit>
</column>

<loadquery
value="select OP.ontology_object_property_id,OP.ontology_object_property_domain,OP.ontology_object_property_range,OP.ontology_object_property_REFLEXIVE,OP.ontology_object_property_SYMMETRIC,OP.ontology_object_property_TRANSITIVE,OP.ontology_object_property_FUNCTIONAL,OP.ontology_object_property_INVERSE_FUNCTIONAL,OP.ontology_element_id,OP.date_ontology_object_property_saved,OP.comment from ONTOLOGY_OBJECT_PROPERTY OP, ONTOLOGY O,

```
ONTOLOGY_ELEMENT E where OP.ontology_element_id = E.ontology_element_id
and E.ontology_id=%ontology_id% "></loadquery>

                <delete>
                        <queries>
                                <query id="1"
                                        sql="[CDATA[delete FROM
ontology_object_property where
ontology_object_property_id=%ontology_object_property_id%]]"

        parameter="ontology_object_property_id"></query>
                        </queries>
                </delete>
        </part>
</structure>

<layout id="OntologyOPLayout">
        <part id="BodyDiv"></part>
        <part id="OntologySelectorRowContainer" parent_id="BodyDiv"></part>
        <part id="OntologyTitleLabel"
parent_id="OntologySelectorRowContainer"></part>
        <part id="ontology_id" parent_id="OntologySelectorRowContainer"></part>
        <part id="OntologyOPGrid" height="300" parent_id="BodyDiv"></part>

</layout>

<content id="OntologyOPContent">
   <part id="OntologyTitleLabel" value="Select ontology:" valuetype="inline"></part>

</content>

<style id="OntologyOPStyle">
   <part id="BodyDiv" valuetype="reference" value="FlexColumnContainer"></part>
   <part id="BodyDiv"  valuetype="inline" value="[CDATA[width:100%;]]"></part>
```

```
        <part id="OntologySelectorRowContainer"  valuetype="reference"
    value="FlexRowContainer"></part>
        <part id="OntologySelectorRowContainer"  valuetype="inline"
    value="[CDATA[width:90%;]]"></part>
    </style>

    <behaviour id="OntologyOPBehaviour">
    </behaviour>
    </interface>
```

- ✓ **OntologyNamedIndividual Interface(interface.xml)**

```
<interface
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://learnityframework.org/"
  xsi:schemaLocation="http://learnityframework.org/
  http://localhost:8181/learnitymsme/xsd/Interface.xsd"
  id="OntologyNamedIndividual" title="Ontology named individual" type="Interface">

  <configuration checkrole="false"></configuration>

  <structure>
    <part id="BodyDiv" class="label"></part>
    <part id="OntologySelectorRowContainer" class="label"></part>
    <part id="OntologyTitleLabel" class="label"></part>
    <part id="OntologyNIGrid" class="Conditionalgrid" caption="OntologyNI"
  sortname="ontology_named_individual_id" sortorder="desc" autoencode="false"
  generateExcel="true">

        <selector id="ontology_id" class="combo"
            contentid="OntologyNIContent"
            behaviourid="OntologyNIBehaviour" domaintype="query"
            value="select ontology_id,ontology_title from ONTOLOGY"
            gridrefresh="true">
```

```
		</selector>
		<column head="ontology_named_individual_id"
name="ontology_named_individual_id" index="1" width="0" editable="true" hidden="true"
key="true">
			<edit type="text" size="30"></edit>
		</column>
		<column head="ontology element id" name="ontology_element_id" index="2"
width="60" editable="true" hidden="false" key="false" required="false">
			<edit type="text" size="30"></edit>
		</column>
	<column head="ontology class id" name="ontology_class_id" index="3" width="60"
editable="true" hidden="false" key="false" required="false">
			<edit type="text" size="30"></edit>
		</column>
		<column head="date ontology named individual saved"
name="date_ontology_named_individual_saved" index="4" width="70" editable="false"
hidden="false" key="false" required="false">
			<edit type="text" size="30"></edit>
		</column>
		<column head="Comment" name="comment" index="5" width="60"
editable="true" hidden="false" key="false" required="false">
			<edit type="textarea" size="30"></edit>
		</column>


		<loadquery
			value="select
NI.ontology_named_individual_id,NI.ontology_element_id,NI.ontology_class_id,NI.date_o
ntology_named_individual_saved,NI.comment from
ONTOLOGY_NAMED_INDIVIDUAL NI, ONTOLOGY O, ONTOLOGY_ELEMENT E,
ONTOLOGY_CLASS C where NI.ontology_class_id = C.ontology_class_id and
NI.ontology_element_id = E.ontology_element_id and E.ontology_id=%ontology_id%
"></loadquery>
```

```
            <delete>
                <queries>
                    <query id="1"
                        sql="[CDATA[delete FROM
ontology_named_individual where
ontology_named_individual_id=%ontology_named_individual_id%]]"
                        parameter="ontology_named_individual_id"></query>
                </queries>
            </delete>
    </part>
</structure>

<layout id="OntologyNILayout">
    <part id="BodyDiv"></part>
    <part id="OntologySelectorRowContainer" parent_id="BodyDiv"></part>
    <part id="OntologyTitleLabel" parent_id="OntologySelectorRowContainer"></part>
    <part id="ontology_id" parent_id="OntologySelectorRowContainer"></part>
    <part id="OntologyNIGrid" height="300" parent_id="BodyDiv"></part>
</layout>
<content id="OntologyNIContent">
    <part id="OntologyTitleLabel" value="Select ontology:" valuetype="inline"></part>
</content>
<style id="OntologyNIStyle">
    <part id="BodyDiv" valuetype="reference" value="FlexColumnContainer"></part>
    <part id="BodyDiv"  valuetype="inline" value="[CDATA[width:100%;]]"></part>
    <part id="OntologySelectorRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
    <part id="OntologySelectorRowContainer"  valuetype="inline"
value="[CDATA[width:30%;]]"></part>
</style>

<behaviour id="OntologyNIBehaviour">
</behaviour>
</interface>
```

✓ **Backend Code (Ontology.java)**

```
package ontology;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintStream;
import java.util.Optional;
import java.util.Set;
import java.util.stream.Collectors;
import java.util.zip.*;
import javax.activation.MimetypesFileTypeMap;
import javax.print.DocFlavor.URL;
import javax.xml.parsers.ParserConfigurationException;
import org.apache.xerces.parsers.DOMParser;
import org.directwebremoting.io.FileTransfer;
import org.semanticweb.owlapi.apibinding.OWLManager;
import org.semanticweb.owlapi.io.UnparsableOntologyException;
import org.semanticweb.owlapi.model.AxiomType;
import org.semanticweb.owlapi.model.ClassExpressionType;
import org.semanticweb.owlapi.model.HasDomain;
import org.semanticweb.owlapi.model.HasIRI;
import org.semanticweb.owlapi.model.IRI;
import org.semanticweb.owlapi.model.OWLAnnotation;
import org.semanticweb.owlapi.model.OWLAnnotationProperty;
import org.semanticweb.owlapi.model.OWLAnnotationPropertyDomainAxiom;
import org.semanticweb.owlapi.model.OWLAxiom;
import org.semanticweb.owlapi.model.OWLClass;
import org.semanticweb.owlapi.model.OWLClassExpression;
import org.semanticweb.owlapi.model.OWLDataProperty;
import org.semanticweb.owlapi.model.OWLDataPropertyDomainAxiom;
import org.semanticweb.owlapi.model.OWLDataPropertyRangeAxiom;
```

```
import org.semanticweb.owlapi.model.OWLDatatype;
import org.semanticweb.owlapi.model.OWLDifferentIndividualsAxiom;
import org.semanticweb.owlapi.model.OWLDocumentFormat;
import org.semanticweb.owlapi.model.OWLEntity;
import org.semanticweb.owlapi.model.OWLFunctionalDataPropertyAxiom;
import org.semanticweb.owlapi.model.OWLFunctionalObjectPropertyAxiom;
import org.semanticweb.owlapi.model.OWLImportsDeclaration;
import org.semanticweb.owlapi.model.OWLInverseFunctionalObjectPropertyAxiom;
import org.semanticweb.owlapi.model.OWLLiteral;
import org.semanticweb.owlapi.model.OWLNamedIndividual;
import org.semanticweb.owlapi.model.OWLObjectCardinalityRestriction;
import org.semanticweb.owlapi.model.OWLObjectExactCardinality;
import org.semanticweb.owlapi.model.OWLObjectMaxCardinality;
import org.semanticweb.owlapi.model.OWLObjectMinCardinality;
import org.semanticweb.owlapi.model.OWLObjectProperty;
import org.semanticweb.owlapi.model.OWLObjectPropertyDomainAxiom;
import org.semanticweb.owlapi.model.OWLObjectPropertyRangeAxiom;
import org.semanticweb.owlapi.model.OWLObjectSomeValuesFrom;
import org.semanticweb.owlapi.model.OWLObjectVisitor;
import org.semanticweb.owlapi.model.OWLOntology;
import org.semanticweb.owlapi.model.OWLOntologyCreationException;
import org.semanticweb.owlapi.model.OWLOntologyManager;
import org.semanticweb.owlapi.model.OWLQuantifiedObjectRestriction;
import org.semanticweb.owlapi.model.OWLReflexiveObjectPropertyAxiom;
import org.semanticweb.owlapi.model.OWLSubClassOfAxiom;
import org.semanticweb.owlapi.model.OWLSymmetricObjectPropertyAxiom;
import org.semanticweb.owlapi.model.OWLTransitiveObjectPropertyAxiom;
import org.semanticweb.owlapi.model.UnloadableImportException;
import org.semanticweb.owlapi.model.parameters.Imports;
import org.semanticweb.owlapi.rdf.rdfxml.renderer.OWLOntologyXMLNamespaceManager;
import org.semanticweb.owlapi.reasoner.NodeSet;
import org.semanticweb.owlapi.reasoner.OWLReasoner;
import org.semanticweb.owlapi.reasoner.OWLReasonerFactory;
import org.semanticweb.owlapi.vocab.OWLRDFVocabulary;
```

```
import org.semanticweb.owlapi.model.OWLDataFactory;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;
import com.clarkparsia.pellet.owlapi.PelletReasonerFactory;
//import com.github.andrewoma.dexx.collection.Set;
import comv2.aunwesha.lfutil.Triple;
import org.w3c.dom.CharacterData;
import interfaceenginev2.FileDownloaderPojo;
import interfaceenginev2.FileUploaderPojo;
import learnityInit.Host;
import learnityInit.util.FolderZiper;
import wordnet.DataBaseLayer;

public class Ontology
{
    public static String uploadOntologyXML(FileTransfer file, String ontology_given_id)
    throws OWLOntologyCreationException, FileNotFoundException, SAXException,
    IOException
    {
        String html = "Error in uploading ontology file";
        String dirName= "";
        String strLocation = Host.getWordNetPath();
        int
    class_count=0,obj_prop_cnt=0,data_prop_cnt=0,indiv_cnt=0,op_domain=0,op_range=0,op_
    reflex=0,op_sym=0,op_trans=0,op_func=0,op_inverse_func=0;
        int
    dp_domain=0,dp_reflex=0,dp_sym=0,dp_trans=0,dp_func=0,inverse_func=0,inverse_prop_
    id=0,equiv_prop_id=0;
        String elmntType = "",elmntIri = "",rdfId ="",elmntLabel ="",elmntComment
    ="",content="Nothing yet",classType="",dp_range="";
        dirName=strLocation;
```

```
//System.out.println(dirName);
File ufile=null;
if(dirName == null) {
	html="Please supply uploadDir parameter";
	return html;
}
ufile = new File(dirName);
if(!ufile.exists())
	ufile.mkdir();

//UnzipUtility unzipper = new UnzipUtility();
FileOutputStream out;
PrintStream p;
FileUploaderPojo fu = new FileUploaderPojo(file);
InputStream fileInputStream = fu.getInputStream();
String fileName = fu.getFilename();
String fileType = fu.getMimeType();
long fileSize = fu.getFileSize();
dirName = dirName+File.separator+fileName;

try {
	out = new FileOutputStream(dirName);
	int length = 0;
	byte b[]= new byte[1024];
	while ((fileInputStream != null) && ((length = fileInputStream.read(b)) != -1))
	{
		out.write(b,0,length);
	}
	p = new PrintStream( out );

	p.close();
	String filepath = Host.getWordNetPath();
	try {
```

```
                fileType =
MimetypesFileTypeMap.getDefaultFileTypeMap().getContentType(filepath);
            } catch (final Exception exception) {
               return "<EXCEPTION: " + exception.getMessage() + ">";
            }
            if(fileName.contains(".owl")) {
                String file_Type = "OWL/XML";
                fileType = file_Type;
            }
            else if(fileName.contains(".owl.xml")) {
                String file_Type = "OWL/XML";
                fileType = file_Type;
            }
            else if(fileName.contains(".owl.rdf")) {
                String file_Type = "RDF/XML";
                fileType = file_Type;
            }
            //System.out.println(fileType);
            String id = ontology_given_id;
            File inFile=new File(dirName);
            InputStream inStream= new FileInputStream(inFile);

        // load ontology
        OWLOntologyManager manager = OWLManager.createOWLOntologyManager();
        OWLDataFactory factory = manager.getOWLDataFactory();

        File inputfile = new File(dirName);
        OWLOntology ontology = manager.loadOntologyFromOntologyDocument(inputfile);
        IRI ontology_iri = ontology.getOntologyID().getOntologyIRI().get();
        String iri = ontology_iri.getIRIString();

        Set<OWLEntity> setOfOwlEntity = ontology.getSignature();
        Set<OWLNamedIndividual> individuals = (Set<OWLNamedIndividual>)
ontology.getIndividualsInSignature();
```

```
      Set<OWLClass> classes = (Set<OWLClass>) ontology.getClassesInSignature();;
    Set<OWLObjectProperty> prop = (Set<OWLObjectProperty>)
ontology.getObjectPropertiesInSignature();
    Set<OWLDataProperty> dataProp= (Set<OWLDataProperty>)
ontology.getDataPropertiesInSignature();
    Set<OWLDatatype> datatype= (Set<OWLDatatype>)
ontology.getDatatypesInSignature();
    Set<OWLAnnotationProperty> annoProp= (Set<OWLAnnotationProperty>)
ontology.getAnnotationPropertiesInSignature();
    indiv_cnt = individuals.size();
    class_count = classes.size();
    obj_prop_cnt = prop.size();
    data_prop_cnt = dataProp.size();
    Set<OWLClassExpression> classtype = (Set<OWLClassExpression>)
ontology.getNestedClassExpressions();
    Set<OWLOntology> imp = (Set<OWLOntology>) ontology.getDirectImports();
   // boolean flag =
DatabaseLayer.saveOntologyDetails(fileName,iri,fileType,inStream,class_count,obj_prop_c
nt,data_prop_cnt,indiv_cnt);
    int flag =
DatabaseLayer.saveOntologyDetails(fileName,iri,fileType,inStream,class_count,obj_prop_c
nt,data_prop_cnt,indiv_cnt);
      for(OWLEntity oe:setOfOwlEntity) {
            String eType = oe.getEntityType().getName();
            //System.out.println(eType);
            switch(eType) {
            case"Class":
                  for (OWLClass cls : classes) {
              elmntIri = cls.getIRI().getIRIString();
                  }
            for (OWLAnnotation ann: ontology.getAnnotations()){
              if(ann.getProperty().equals(factory.getRDFSLabel())){
                  OWLLiteral val = (OWLLiteral) ann.getValue();
                  elmntLabel = val.getLiteral().toLowerCase();
```

```
            }
            if(ann.getProperty().equals(factory.getRDFSComment())){
                    OWLLiteral val = (OWLLiteral) ann.getValue();
                    elmntComment = val.getLiteral().toLowerCase();
            }
          }

            elmntType="CLASS";
            int flag1=
DatabaseLayer.fetchOntologyElementDetails(elmntType,elmntIri,rdfId,elmntLabel,elmntCo
mment,content,flag);
            classType="simple";
            int flag7= DatabaseLayer.OntologyClassDetails(classType, flag1);
                    break;
         case"NamedIndividual":
                    for (OWLNamedIndividual i : individuals) {
             elmntIri = i.getIRI().getIRIString();
                    }
        for (OWLAnnotation ann: ontology.getAnnotations()){
            if(ann.getProperty().equals(factory.getRDFSLabel())){
                    OWLLiteral val = (OWLLiteral) ann.getValue();
                    elmntLabel = val.getLiteral().toLowerCase();
            }
            if(ann.getProperty().equals(factory.getRDFSComment())){
                    OWLLiteral val = (OWLLiteral) ann.getValue();
                    elmntComment = val.getLiteral().toLowerCase();
            }
          }

             elmntType="NAMEDINDIVIDUAL";
              int flag2=
DatabaseLayer.fetchOntologyElementDetails(elmntType,elmntIri,rdfId,elmntLabel,elmntCo
mment,content,flag);
              int classid = DatabaseLayer.OntologyClassDetails(classType, flag2);
```

```
                boolean flag8=
DatabaseLayer.OntologyNamedIndividualDetails(flag2,classid);
                        break;
                case"ObjectProperty":
                        for(OWLObjectProperty oop : prop){
                 elmntIri = oop.getIRI().getIRIString();
                        }
                        for (OWLAnnotation ann: ontology.getAnnotations()){
                  if(ann.getProperty().equals(factory.getRDFSLabel())){
                        OWLLiteral val = (OWLLiteral) ann.getValue();
                        elmntLabel = val.getLiteral().toLowerCase();
                  }
                  if(ann.getProperty().equals(factory.getRDFSComment())){
                        OWLLiteral val = (OWLLiteral) ann.getValue();
                        elmntComment = val.getLiteral().toLowerCase();
                  }}
                elmntType="OBJECTPROPERTY";
                int flag3=
DatabaseLayer.fetchOntologyElementDetails(elmntType,elmntIri,rdfId,elmntLabel,elmntCo
mment,content,flag);
                for (OWLClass cls : classes) {
                   for (OWLObjectPropertyDomainAxiom op :
ontology.getAxioms(AxiomType.OBJECT_PROPERTY_DOMAIN)) {
                        if (op.getDomain().equals(cls)) {
                        op_domain++;
                        }
                   }
                   for(OWLObjectPropertyRangeAxiom opr :
ontology.getAxioms(AxiomType.OBJECT_PROPERTY_RANGE)) {
                        if(opr.getRange().equals(cls)) {
                                op_range++;
                        }
                   }
```

```
                for(OWLReflexiveObjectPropertyAxiom  ref :
ontology.getAxioms(AxiomType.REFLEXIVE_OBJECT_PROPERTY)) {
                        if(ref.getAxiomType().equals(cls)) {
                                op_reflex++;
                        }
                }
                for(OWLSymmetricObjectPropertyAxiom
sym:ontology.getAxioms(AxiomType.SYMMETRIC_OBJECT_PROPERTY)) {
                        if(sym.getAxiomType().equals(cls)) {
                                op_sym++;
                        }
                }
                for(OWLTransitiveObjectPropertyAxiom
trans:ontology.getAxioms(AxiomType.TRANSITIVE_OBJECT_PROPERTY)) {
                        if(trans.getAxiomType().equals(cls)) {
                                op_trans++;
                        }
                }
                for(OWLFunctionalObjectPropertyAxiom
func:ontology.getAxioms(AxiomType.FUNCTIONAL_OBJECT_PROPERTY)) {
                        if(func.getAxiomType().equals(cls)) {
                                op_func++;
                        }
                }
                for(OWLInverseFunctionalObjectPropertyAxiom
inv:ontology.getAxioms(AxiomType.INVERSE_FUNCTIONAL_OBJECT_PROPERTY))
{
                        if(inv.getAxiomType().equals(cls)) {
                                op_inverse_func++;
                        }}}
             boolean opdetails = DatabaseLayer.OntologyObjectPropertyDetails(op_domain,
op_range, op_reflex, op_sym, op_trans, op_func, op_inverse_func, flag3);
                        break;
               case"DataProperty":
```

```
                for(OWLDataProperty odp : dataProp){
            elmntIri = odp.getIRI().getIRIString();
                }
            for (OWLAnnotation ann: ontology.getAnnotations()){
              if(ann.getProperty().equals(factory.getRDFSLabel())){
                OWLLiteral val = (OWLLiteral) ann.getValue();
                elmntLabel = val.getLiteral().toLowerCase();
              }
              if(ann.getProperty().equals(factory.getRDFSComment())){
                OWLLiteral val = (OWLLiteral) ann.getValue();
                elmntComment = val.getLiteral().toLowerCase();
              }
            }

            elmntType="DATAPROPERTY";
            int flag4=
DatabaseLayer.fetchOntologyElementDetails(elmntType,elmntIri,rdfId,elmntLabel,elmntCo
mment,content,flag);
            for (OWLClass cls : classes) {
              for (OWLDataPropertyDomainAxiom dp :
ontology.getAxioms(AxiomType.DATA_PROPERTY_DOMAIN)) {
                if (dp.getDomain().equals(cls)) {
                dp_domain++;
                }
              }
              for(OWLDataPropertyRangeAxiom dpr :
ontology.getAxioms(AxiomType.DATA_PROPERTY_RANGE)) {
                if(dpr.getRange().equals(datatype)) {
                  op_range++;
                }

              }
```

```
                for(OWLFunctionalDataPropertyAxiom
func:ontology.getAxioms(AxiomType.FUNCTIONAL_DATA_PROPERTY)) {
                        if(func.getAxiomType().equals(cls)) {
                                dp_func++;
                        }}}
                        break;
                case"AnnotationProperty":

                        for(OWLAnnotationProperty oap : annoProp){
                 elmntIri = oap.getIRI().getIRIString();
                        }
                for (OWLAnnotation ann: ontology.getAnnotations()){
                  if(ann.getProperty().equals(factory.getRDFSLabel())){
                        OWLLiteral val = (OWLLiteral) ann.getValue();
                        elmntLabel = val.getLiteral().toLowerCase();
                  }
                  if(ann.getProperty().equals(factory.getRDFSComment())){
                        OWLLiteral val = (OWLLiteral) ann.getValue();
                        elmntComment = val.getLiteral().toLowerCase();
                  }
                }

                 elmntType="ANNOTATIONPROPERTY";
                 int flag5=
DatabaseLayer.fetchOntologyElementDetails(elmntType,elmntIri,rdfId,elmntLabel,elmntCo
mment,content,flag);
                        break;
                case"Datatype":

                        for(OWLDatatype d: datatype) {
                        elmntIri = d.getIRI().getIRIString();
                        }
                for (OWLAnnotation ann: ontology.getAnnotations()){
                  if(ann.getProperty().equals(factory.getRDFSLabel())){
```

```
                    OWLLiteral val = (OWLLiteral) ann.getValue();
                    elmntLabel = val.getLiteral().toLowerCase();
                }
                if(ann.getProperty().equals(factory.getRDFSComment())){
                    OWLLiteral val = (OWLLiteral) ann.getValue();
                    elmntComment = val.getLiteral().toLowerCase();
                }}
                    elmntType="DATATYPE";
                    int flag6=
    DatabaseLayer.fetchOntologyElementDetails(elmntType,elmntIri,rdfId,elmntLabel,elmntCo
    mment,content,flag);
                    break;
}}
            String imported_onto_id= ontology_given_id;
          //  boolean flag1= DatabaseLayer.importOntologyDetails(iri, imported_onto_id, flag);


           OWLDocumentFormat format = manager.getOntologyFormat(ontology);
           html = "Ontology uploaded succesfully";
      }catch(UnloadableImportException u) {
            u.printStackTrace();
      } catch(Exception e) {
            e.printStackTrace();
      }
      return html;
    }
      // resolve maven specific path for resources
      public static String getRelativeResourcePath( String filepath ) throws
    FileNotFoundException {
        if( filepath == null || filepath.equals("") ) throw new IllegalArgumentException(
    filepath );

        java.net.URL url = Ontology.class.getClassLoader().getResource( filepath );
```

```
            if( url == null ) throw new FileNotFoundException( filepath );

            return url.getPath();
        }
public static String deleteOntologyDetails(String ontology_id) {
    String html = "Error in deleting ontology";
    //String id=wordnet_id;
    boolean flag = DatabaseLayer.deleteOntologyDetails(ontology_id);
    if(flag==true)
       html = "Ontology deleted successfully.";
    return html;
}
    }
```

- ✓ **OntologySearchResults Interface(interface.xml)**

```
<interface
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://learnityframework.org/"
    xsi:schemaLocation="http://learnityframework.org/
    http://localhost:8181/learnitymsme/xsd/Interface.xsd"
    id="OntologySearchResult" title="Ontology_Search_Results" type="Interface">

    <configuration checkrole="false"></configuration>

<structure>
        <part id="BodyDiv" class="label"></part>
        <part id="OntologyForm" class="form"></part>
        <part id="OntologyFormFieldset" class="fieldset"></part>
        <part id="OntologyFormLegend" class="legend"></part>

       <part id="OntologySelectorRowContainer" class="label"></part>
        <part id="OntologyTitle" class="label"></part>
        <part id="ChooseOntology" class="combo"></part>
```

```
   <part id="DisplayOptionRowContainer" class="label"></part>
   <part id="DisplayOptionLabel" class="label"></part>
   <part id="SelectTypeValueOption" class="combo"></part>

   <part id="OntologyInputRowContainer" class="label"></part>
   <part id="OntologySearchValueLabel" class="label"></part>
   <part id="OntologySearchValue" class="inputtext"></part>
   <part id="SearchButtonRowContainer" class="label"></part>
   <part id="SearchButton" class="button"></part>

  <part id="OntologySRGrid" class="DBgrid" caption="SearchResult" sortname="id"
sortorder="desc" autoencode="false" generateExcel="true">
      <column head="id" name="id" index="1" width="0" editable="true" hidden="true"
key="true">
                  <edit type="text" size="30"></edit>
            </column>
      <column head="Ontology name" name="ontology_name" index="2" width="50"
editable="true" hidden="false" key="false" required="true">
                  <edit type="text" size="30"></edit>
            </column>
            <column head="Element type" name="element_type" index="3" width="60"
editable="true" hidden="false" key="false" required="true">
                  <edit type="text" size="30"></edit>
            </column>
            <column head="Label" name="label" index="4" width="50" editable="true"
hidden="false" key="false" required="false">
                  <edit type="text" size="30"></edit>
            </column>
            <column head="Description" name="description" index="5" width="70"
editable="true" hidden="false" key="false" required="false">
                  <edit type="textarea" cols="28" rows="5"></edit>
            </column>
            <loadquery
```

```
                value="select id,ontology_name,element_type,label,description from
ontology_search_results"></loadquery>

        <delete>
            <queries>
                <query id="1"
                    sql="[CDATA[delete FROM ontology_search_results]]"
                    parameter="ontology_name"></query>
            </queries>
        </delete>
        </part>

   <part id="ClearButtonRowContainer" class="label"></part>
   <part id="ClearResultsButton" class="button"></part>
   <part id="OntologyTestAjaxComp" class="ajaxcomponent"></part>
</structure>
<layout id="OntologySRLayout">
   <part id="BodyDiv"></part>
   <part id="OntologyForm" parent_id="BodyDiv"></part>
   <part id="OntologyFormFieldset" parent_id="OntologyForm"></part>
   <part id="OntologyFormLegend" parent_id="OntologyFormFieldset"></part>

   <part id="OntologySelectorRowContainer" parent_id="OntologyFormFieldset"></part>
   <part id="OntologyTitle" parent_id="OntologySelectorRowContainer"></part>
   <part id="ChooseOntology" parent_id="OntologySelectorRowContainer"></part>
   <part id="DisplayOptionRowContainer"
parent_id="OntologySelectorRowContainer"></part>
   <part id="DisplayOptionLabel" parent_id="DisplayOptionRowContainer"></part>
   <part id="SelectTypeValueOption" parent_id="DisplayOptionRowContainer"></part>

   <part id="OntologyInputRowContainer" parent_id="OntologyFormFieldset"></part>
   <part id="OntologySearchValueLabel"
parent_id="OntologyInputRowContainer"></part>
   <part id="OntologySearchValue" parent_id="OntologyInputRowContainer"></part>
```

```
   <part id="SearchButtonRowContainer"
parent_id="OntologyInputRowContainer"></part>
   <part id="SearchButton" parent_id="SearchButtonRowContainer"></part>

  <part id="OntologySRGrid" height="300" parent_id="OntologyFormFieldset"></part>

   <part id="ClearButtonRowContainer" parent_id="OntologyFormFieldset"></part>
   <part id="ClearResultsButton" parent_id="ClearButtonRowContainer"></part>
   <part id="OntologyTestAjaxComp" parent_id="BodyDiv"></part>

</layout>
<content id="OntologySRContent">

<part id="OntologyFormLegend" value="Ontology_Search_Results"
valuetype="inline"></part>
<part id="OntologyTitle" value="Select Ontology: " valuetype="inline"></part>
   <part id="ChooseOntology" value="select ontology_id,ontology_title from
ONTOLOGY" valuetype="query"></part>

   <part id="OntologySearchValueLabel" value="Input: " valuetype="inline"></part>
   <part id="SearchButton" value="Search" valuetype="inline"></part>
   <part id="DisplayOptionLabel" value="Entity Type: " valuetype="inline"></part>

   <part id="SelectTypeValueOption" value="searchtype" valuetype="fixed">
            <combooption name="Class" value="Class"></combooption>
            <combooption name="NamedIndividual"
value="NamedIndividual"></combooption>
            <combooption name="ObjectProperty" value="ObjectProperty"></combooption>
            <combooption name="DataProperty" value="DataProperty"></combooption>
            <combooption name="DataType" value="DataType"></combooption>

   </part>

  <part id="ResultLabel" value="Results:" valuetype="inline"></part>
```

```
   <part id="ClearResultsButton" value="Clear Results" valuetype="inline"></part>

</content>
<style id="OntologySRStyle">
   <part id="BodyDiv"  valuetype="reference" value="FlexColumnContainer"></part>
   <part id="OntologyFormFieldset"  valuetype="reference"
value="FlexRowContainer"></part>
   <part id="OntologyFormFieldset"  valuetype="inline"
value="[CDATA[width:85%;]]"></part>
   <part id="OntologySelectorRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
   <part id="OntologySelectorRowContainer"  valuetype="inline"
value="[CDATA[width:90%;]]"></part>
   <part id="OntologyInputRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
   <part id="OntologyInputRowContainer"  valuetype="inline"
value="[CDATA[width:50%;]]"></part>
   <part id="DisplayOptionRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
   <part id="DisplayOptionRowContainer"  valuetype="inline"
value="[CDATA[width:40%;]]"></part>
   <part id="ViewResultRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
   <part id="ViewResultRowContainer"  valuetype="inline"
value="[CDATA[width:40%;]]"></part>

   <part id="ClearButtonRowContainer"  valuetype="reference"
value="FlexRowContainer"></part>
   <part id="ClearButtonRowContainer"  valuetype="inline"
value="[CDATA[width:30%;]]"></part>
</style>

<behaviour id="OntologySRBehaviour">
```

```
    <part id="SearchButton">
        <event name="onclick" type="simple" function="SearchButton_onclick"
            valuetype="jsevent" resourceid="OntologyTestjs"></event>
    </part>

    <part id="ClearResultsButton">
        <event name="onclick" type="simple" function="ClearResultsButton_onclick"
            valuetype="jsevent" resourceid="OntologyTestjs"></event>
    </part>
    <part id="OntologyTestAjaxComp">
        <event name="ajax" type="simple" function="dummy"
javaclass="OntologyTest"></event>
    </part>
</behaviour>

<resource>
    <resourceitem id="OntologyTestjs" href="OntologyTest.js"
valuetype="js"></resourceitem>
</resource>
</interface>
```

- ✓ **OntologyTest.js**

```
function SearchButton_onclick(){
    var ont_name=getValue('ChooseOntology');
    var input = getValue('OntologySearchValue');
    if(ont_name!=0){
        var elmt_type=getValue('SelectTypeValueOption');
        if(elmt_type!=0){
            $.blockUI({message: 'finding chosen search_type'});
            OntologyTest.getOntologySearchResult(ont_name,input,elmt_type,function(data)
    {
                $.unblockUI();
                setValue('OntologySRGrid',data);
            })
```

```
        }
        else{
                alert("Please choose select type from the dropdown above");
        }
    }
    else{
        alert("Please Select an ontology from the dropdown above");
    }
}

function ClearResultsButton_onclick(){
    //setValue('OntologySRGrid','');
        var id=getIdOfSelectedRow('OntologySRGrid');
    if(id!=0){
        $.blockUI({message: 'Starting deleting of result file'});
        OntologyTest.deleteOntologyDetails(id,function(data) {
                $.unblockUI();
                setReloadGrid('OntologySRGrid');
                $.blockUI({message: data});
                setTimeout($.unblockUI, 1000);
        });
    }
    else {
        alert("Please Select a ontology from the grid");
        return;
    }
    }
```

- ✓ **Backend Code (OntologyTest.java)**

```
package ontology;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
```

```
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintStream;
import java.util.Optional;
import java.util.Set;
import java.util.stream.Collectors;
import java.util.zip.*;

import javax.activation.MimetypesFileTypeMap;
import javax.print.DocFlavor.URL;
import javax.xml.parsers.ParserConfigurationException;
import org.apache.xerces.parsers.DOMParser;
import org.directwebremoting.io.FileTransfer;
import org.semanticweb.owlapi.apibinding.OWLManager;
import org.semanticweb.owlapi.io.UnparsableOntologyException;
import org.semanticweb.owlapi.model.AxiomType;
import org.semanticweb.owlapi.model.HasDomain;
import org.semanticweb.owlapi.model.HasIRI;
import org.semanticweb.owlapi.model.IRI;
import org.semanticweb.owlapi.model.OWLAnnotation;
import org.semanticweb.owlapi.model.OWLAnnotationProperty;
import org.semanticweb.owlapi.model.OWLAnnotationPropertyDomainAxiom;
import org.semanticweb.owlapi.model.OWLAxiom;
import org.semanticweb.owlapi.model.OWLClass;
import org.semanticweb.owlapi.model.OWLClassExpression;
import org.semanticweb.owlapi.model.OWLDataProperty;
import org.semanticweb.owlapi.model.OWLDataPropertyDomainAxiom;
import org.semanticweb.owlapi.model.OWLDataPropertyRangeAxiom;
import org.semanticweb.owlapi.model.OWLDatatype;
import org.semanticweb.owlapi.model.OWLDifferentIndividualsAxiom;
import org.semanticweb.owlapi.model.OWLDocumentFormat;
import org.semanticweb.owlapi.model.OWLEntity;
import org.semanticweb.owlapi.model.OWLFunctionalDataPropertyAxiom;
import org.semanticweb.owlapi.model.OWLFunctionalObjectPropertyAxiom;
```

```
import org.semanticweb.owlapi.model.OWLImportsDeclaration;
import org.semanticweb.owlapi.model.OWLInverseFunctionalObjectPropertyAxiom;
import org.semanticweb.owlapi.model.OWLLiteral;
import org.semanticweb.owlapi.model.OWLNamedIndividual;
import org.semanticweb.owlapi.model.OWLObjectCardinalityRestriction;
import org.semanticweb.owlapi.model.OWLObjectExactCardinality;
import org.semanticweb.owlapi.model.OWLObjectMaxCardinality;
import org.semanticweb.owlapi.model.OWLObjectMinCardinality;
import org.semanticweb.owlapi.model.OWLObjectProperty;
import org.semanticweb.owlapi.model.OWLObjectPropertyDomainAxiom;
import org.semanticweb.owlapi.model.OWLObjectPropertyRangeAxiom;
import org.semanticweb.owlapi.model.OWLObjectSomeValuesFrom;
import org.semanticweb.owlapi.model.OWLObjectVisitor;
import org.semanticweb.owlapi.model.OWLOntology;
import org.semanticweb.owlapi.model.OWLOntologyCreationException;
import org.semanticweb.owlapi.model.OWLOntologyManager;
import org.semanticweb.owlapi.model.OWLQuantifiedObjectRestriction;
import org.semanticweb.owlapi.model.OWLReflexiveObjectPropertyAxiom;
import org.semanticweb.owlapi.model.OWLSubClassOfAxiom;
import org.semanticweb.owlapi.model.OWLSymmetricObjectPropertyAxiom;
import org.semanticweb.owlapi.model.OWLTransitiveObjectPropertyAxiom;
import org.semanticweb.owlapi.model.UnloadableImportException;
import org.semanticweb.owlapi.model.parameters.Imports;
import org.semanticweb.owlapi.rdf.rdfxml.renderer.OWLOntologyXMLNamespaceManager;
import org.semanticweb.owlapi.reasoner.NodeSet;
import org.semanticweb.owlapi.reasoner.OWLReasoner;
import org.semanticweb.owlapi.reasoner.OWLReasonerFactory;
import org.semanticweb.owlapi.vocab.OWLRDFVocabulary;
import org.semanticweb.owlapi.model.OWLDataFactory;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;
```

```
import com.clarkparsia.pellet.owlapi.PelletReasonerFactory;

//import com.github.andrewoma.dexx.collection.Set;

import comv2.aunwesha.lfutil.Triple;
import org.w3c.dom.CharacterData;
import interfaceenginev2.FileDownloaderPojo;
import interfaceenginev2.FileUploaderPojo;
import learnityInit.Host;
import learnityInit.util.FolderZiper;
import wordnet.DataBaseLayer;

public class OntologyTest
{
    public static String getOntologySearchResult(String ont_name,String input,String
    elmt_type) throws OWLOntologyCreationException, FileNotFoundException,
    SAXException, IOException
    {
       String html = "Error in ontology Search!";


       try {
                String name=null,type=null,label=null;

                 switch(elmt_type) {
            case"Class":
                Triple<String, String, String> searchRes
    =DatabaseLayer.SearchOntology(ont_name,elmt_type,input);
                                    name = searchRes.getFirst();
                                    type = searchRes.getSecond();
                                    label =searchRes.getThird();
                                    System.out.println(name);
                                    System.out.println(type);
                                    System.out.println(label);
```

```
                    boolean flag = DatabaseLayer.SearchResults(name,type,label);
            break;
        case"NamedIndividual":
            Triple<String, String, String> searchRes1
=DatabaseLayer.SearchOntology(ont_name,elmt_type,input);
                    name = searchRes1.getFirst();
                    type = searchRes1.getSecond();
                    label =searchRes1.getThird();
                    System.out.println(name);
                    System.out.println(type);
                    System.out.println(label);
                    boolean flag1 = DatabaseLayer.SearchResults(name,type,label);
            break;
        case"ObjectProperty":
            Triple<String, String, String> searchRes2
=DatabaseLayer.SearchOntology(ont_name,elmt_type,input);
                    name = searchRes2.getFirst();
                    type = searchRes2.getSecond();
                    label =searchRes2.getThird();
                    System.out.println(name);
                    System.out.println(type);
                    System.out.println(label);
                    boolean flag2 = DatabaseLayer.SearchResults(name,type,label);
            break;
        case"DataProperty":
            Triple<String, String, String> searchRes3
=DatabaseLayer.SearchOntology(ont_name,elmt_type,input);
                    name = searchRes3.getFirst();
                    type = searchRes3.getSecond();
                    label =searchRes3.getThird();
                    System.out.println(name);
                    System.out.println(type);
                    System.out.println(label);
                    boolean flag3 = DatabaseLayer.SearchResults(name,type,label);
```

```
                break;
            case"DataType":
                Triple<String, String, String> searchRes4
    =DatabaseLayer.SearchOntology(ont_name,elmt_type,input);
                                name = searchRes4.getFirst();
                                type = searchRes4.getSecond();
                                label =searchRes4.getThird();
                                System.out.println(name);
                                System.out.println(type);
                                System.out.println(label);
                                boolean flag4 = DatabaseLayer.SearchResults(name,type,label);
                break;
            }
     html = "Search successfull";

      }catch(UnloadableImportException u) {
                u.printStackTrace();
      } catch(Exception e) {
                e.printStackTrace();
      }
      return html;
    }

public static String deleteOntologyDetails(String id) {
    String html = "Error in deleting ontology";
    //String id=wordnet_id;
    boolean flag = DatabaseLayer.deleteSearchResultDetails(id);
    if(flag==true)
      html = "Selected search result deleted successfully.";
    return html;
}
    }
```

✓ **Ontology Database Code (DatabaseLayer.java)**

```
package ontology;

import java.io.ByteArrayOutputStream;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.sql.DataSource;
import org.apache.jena.iri.IRI;
import org.apache.tomcat.jni.File;
import comv2.aunwesha.lfutil.Triple;
//import comv2.aunwesha.lfutil.Triple;
import comv2.aunwesha.param.CoreAdminInitHostInfo;

public class DatabaseLayer
{
        // database connection settings
        public static DataSource ds1 = CoreAdminInitHostInfo.ds1;

   public static int saveOntologyDetails(String fileName, String iri, String fileType, InputStream
is, int class_count,int obj_prop_cnt,int data_prop_cnt, int indiv_cnt)
   {
        boolean flag = false;
        int ontology_id = 0;
      try(Connection oConn = ds1.getConnection())
      {
                        String sql = "Insert into ONTOLOGY
(ontology_title,ontology_iri,ontology_file_type,ontology_file,number_of_classes,number_of_ob
ject_properties,number_of_datatype_properties,number_of_individuals) values(?,?,?,?,?,?,?,?)" ;
            // PreparedStatement statement = oConn.prepareStatement(sql);
```

```
            PreparedStatement statement = oConn.prepareStatement(sql,
Statement.RETURN_GENERATED_KEYS);
            //statement.setString(1, id);
            statement.setString(1, fileName);
            statement.setString(2, iri);
            statement.setString(3, fileType);
          statement.setBlob(4, is);
          statement.setInt(5, class_count);
          statement.setInt(6, obj_prop_cnt);
          statement.setInt(7, data_prop_cnt);
          statement.setInt(8, indiv_cnt);
          statement.executeUpdate();
          ResultSet rs = statement.getGeneratedKeys();
          while(rs.next())
          {
         ontology_id = rs.getInt(1);
          }
          statement.close();
          flag = true;
       }
          catch(SQLException e)
       {
         e.printStackTrace();
       }
       catch (Exception e)
       {
                        e.printStackTrace();
                }

       return ontology_id;
    }

    public static int fetchOntologyElementDetails(String elmntType, String elmntIri, String rdfId,
String elmntLabel, String elmntComment,String content, int ontology_id)
```

```
{
    boolean flag = false;
    int ontology_element_id = 0;
   try(Connection oConn = ds1.getConnection())
   {
                        String sql = "Insert into ONTOLOGY_ELEMENT
(ontology_element_type,ontology_element_iri,ontology_element_rdfID,ontology_element_label
,ontology_element_comment,ontology_element_content,ontology_id) values(?,?,?,?,?,?,?)" ;
            // PreparedStatement statement = oConn.prepareStatement(sql);
             PreparedStatement statement = oConn.prepareStatement(sql,
Statement.RETURN_GENERATED_KEYS);
            // statement.setString(1, elmnt_id);
             statement.setString(1, elmntType);
             statement.setString(2, elmntIri);
             statement.setString(3, rdfId);
      statement.setString(4, elmntLabel);
      statement.setString(5, elmntComment);
      statement.setString(6, content);
      statement.setInt(7, ontology_id);
      statement.executeUpdate();
      ResultSet rs = statement.getGeneratedKeys();
      while(rs.next())
      {
     ontology_element_id = rs.getInt(1);
      }
      statement.close();
      flag = true;
   }
      catch(SQLException e)
   {
     e.printStackTrace();
   }
   catch (Exception e)
   {
```

```
                    e.printStackTrace();
                }
    return ontology_element_id;
  }
    public static boolean importOntologyDetails(String iri, String imported_onto_id,int
ontology_id)
  {
     boolean flag = false;
    try(Connection oConn = ds1.getConnection())
    {
                    String sql = "Insert into ONTOLOGY_IMPORT
(ontology_import_iri,id_of_imported_ontology,ontology_id) values(?,?,?)" ;
                    PreparedStatement statement = oConn.prepareStatement(sql);
                statement.setString(1, iri);
            statement.setString(2, imported_onto_id);
            statement.setInt(3, ontology_id);
       statement.execute();
       statement.close();
       flag = true;
    }
    catch(SQLException e)
    {
      e.printStackTrace();
    }
    catch (Exception e)
    {
                    e.printStackTrace();
                }
    return flag;
  }

  public static int OntologyClassDetails(String classType,int ontology_element_id)
  {
     boolean flag = false;
```

```
        int ontology_class_id=0;
      try(Connection oConn = ds1.getConnection())
      {
                              String sql = "Insert into ONTOLOGY_CLASS
(ontology_class_type,ontology_element_id) values(?,?)" ;
                              PreparedStatement statement =
oConn.prepareStatement(sql,Statement.RETURN_GENERATED_KEYS);

                  statement.setString(1, classType);
                  statement.setInt(2, ontology_element_id);
                  statement.executeUpdate();
          ResultSet rs = statement.getGeneratedKeys();
          while(rs.next())
          {
        ontology_class_id = rs.getInt(1);
          }
          statement.close();
          flag = true;
      }
        catch(SQLException e)
      {
        e.printStackTrace();
      }
      catch (Exception e)
      {
                              e.printStackTrace();
                  }
      return ontology_class_id;
   }
   public static boolean OntologyNamedIndividualDetails(int ontology_element_id,int
ontology_class_id)
   {
        boolean flag = false;
      try(Connection oConn = ds1.getConnection())
```

```
    {
                    String sql = "Insert into ONTOLOGY_NAMED_INDIVIDUAL
(ontology_element_id,ontology_class_id) values(?,?)" ;
                    PreparedStatement statement = oConn.prepareStatement(sql);
          statement.setInt(1, ontology_element_id);
          statement.setInt(2, ontology_class_id);
       statement.execute();
       statement.close();
       flag = true;
    }

    catch(SQLException e)
    {
      e.printStackTrace();
    }
    catch (Exception e)
    {
                    e.printStackTrace();
            }
    return flag;
  }

  public static boolean OntologyDataPropertyDetails(int dp_domain,String dp_range,int
dp_reflex, int dp_sym,int dp_trans,int dp_func,int inverse_func,int inverse_prop_id,int
equiv_prop_id,int ontology_element_id)
  {
      boolean flag = false;
    try(Connection oConn = ds1.getConnection())
    {
                    String sql = "Insert into ONTOLOGY_DATA_PROPERTY
(ontology_data_property_domain,ontology_data_property_range,ontology_data_property_REFL
EXIVE,ontology_data_property_SYMMETRIC,ontology_data_property_TRANSITIVE,ontolog
y_data_property_FUNCTIONAL,ontology_data_property_INVERSE_FUNCTIONAL,ontology
```

```
_data_property_inverse_property_id,ontology_data_property_equivalent_property_id,ontology_
element_id) values(?,?,?,?,?,?,?,?,?,?)" ;
                        PreparedStatement statement = oConn.prepareStatement(sql);

            statement.setInt(1, dp_domain);
            statement.setString(2, dp_range);
            statement.setInt(3, dp_reflex);
            statement.setInt(4, dp_sym);
            statement.setInt(5, dp_trans);
            statement.setInt(6, dp_func);
            statement.setInt(7, inverse_func);
            statement.setInt(8, inverse_prop_id);
            statement.setInt(9, equiv_prop_id);
            statement.setInt(10,ontology_element_id);
        statement.execute();
        statement.close();
        flag = true;
    }

    catch(SQLException e)
    {
      e.printStackTrace();
    }
    catch (Exception e)
    {
                        e.printStackTrace();
                }
    return flag;
  }

  public static boolean OntologyObjectPropertyDetails(int op_domain,int op_range,int
op_reflex, int op_sym,int op_trans,int op_func,int op_inverse_func,int ontology_element_id)
  {
      boolean flag = false;
```

```
    try(Connection oConn = ds1.getConnection())
    {
                        String sql = "Insert into ONTOLOGY_OBJECT_PROPERTY
(ontology_object_property_domain,ontology_object_property_range,ontology_object_property_
REFLEXIVE,ontology_object_property_SYMMETRIC,ontology_object_property_TRANSITI
VE,ontology_object_property_FUNCTIONAL,ontology_object_property_INVERSE_FUNCTI
ONAL,ontology_element_id) values(?,?,?,?,?,?,?,?)" ;
                        PreparedStatement statement = oConn.prepareStatement(sql);

            statement.setInt(1, op_domain);
            statement.setInt(2, op_range);
            statement.setInt(3, op_reflex);
            statement.setInt(4, op_sym);
            statement.setInt(5, op_trans);
            statement.setInt(6, op_func);
            statement.setInt(7, op_inverse_func);
            statement.setInt(8,ontology_element_id);
       statement.execute();
       statement.close();
       flag = true;
    }

    catch(SQLException e)
    {
      e.printStackTrace();
    }
    catch (Exception e)
    {
                        e.printStackTrace();
                }
    return flag;
  }
```

```
   public synchronized static Triple<String,String,String> SearchOntology(String ont_name,
String elmt_type,String input)
   {

      Triple<String,String,String>  searchRes = new Triple<String,String,String>();
      boolean flag = false;
     try(Connection oConn = ds1.getConnection())
     {
      String sql1="";

      sql1="SELECT b.ontology_title, a.ontology_element_type, a.ontology_element_label
FROM ONTOLOGY_ELEMENT a,ONTOLOGY b WHERE b.ontology_id=? AND
a.ontology_element_type=? AND MATCH (a.ontology_element_label) AGAINST (? IN
NATURAL LANGUAGE MODE)";
      PreparedStatement s = oConn.prepareStatement(sql1);

      s.setString(1, ont_name);
      s.setString(2,elmt_type);
                        s.setString(3,input);
                        ResultSet rs= s.executeQuery();
                        while(rs.next()) {
                        searchRes.setFirst(rs.getString(1));
                        searchRes.setSecond(rs.getString(2));
                        searchRes.setThird(rs.getString(3));
                        }
                        s.close();

       flag = true;
     }

     catch(SQLException e)
     {
      e.printStackTrace();
     }
```

```
    catch (Exception e)
    {
                        e.printStackTrace();
                }
    return searchRes;
  }


  public static boolean SearchResults(String name, String type,String label)
  {


     boolean flag = false;
    try(Connection oConn = ds1.getConnection())
    {

                        String sql = "Insert into ONTOLOGY_SEARCH_RESULTS
(ontology_name,element_type,label) values(?,?,?)" ;
                        PreparedStatement stmt = oConn.prepareStatement(sql);
                        stmt.setString(1,name);
                        stmt.setString(2,type);
                        stmt.setString(3,label);
                        //stmt.setString(4,input);
                        stmt.execute();
                        stmt.close();


       flag = true;
    }

    catch(SQLException e)
    {
      e.printStackTrace();
    }
    catch (Exception e)
```

```
    {
                e.printStackTrace();
        }
    return flag;
}
public static boolean deleteOntologyDetails(String ontology_id) {

        boolean flag = false;

        try(Connection oConn = ds1.getConnection()) {

                String sql1 = "delete from ONTOLOGY where ontology_id = ?";

                PreparedStatement pstmt = oConn.prepareStatement(sql1);
                pstmt.setString(1,ontology_id);
                pstmt.executeUpdate();

                pstmt.close();
                flag = true;
        }
        catch (SQLException e) {
                        System.out.println("==SQLException===");
                        e.printStackTrace();
        }
        catch (Exception ex) {
                        System.out.println("==Exception===");
                        ex.printStackTrace();
        }

        return flag;
    }

public static boolean deleteSearchResultDetails(String id) {
```

```
		boolean flag = false;

		try(Connection oConn = ds1.getConnection()) {

			String sql1 = "delete from ONTOLOGY_SEARCH_RESULTS where
id=?";

			PreparedStatement pstmt = oConn.prepareStatement(sql1);
			pstmt.setString(1,id);
			pstmt.executeUpdate();

			pstmt.close();
			flag = true;
		}
		catch (SQLException e) {
				System.out.println("==SQLException===");
				e.printStackTrace();
		}
		catch (Exception ex) {
				System.out.println("==Exception===");
				ex.printStackTrace();
}
		return flag;
	}
```