Experiment No: 2

Aim: To study semantic web open source tools like Apache tinkerpop, RDFlib, Apache jena, protégé, sesame.

Theory:

The Semantic Web is a vision about an extension of the existing World Wide Web, which provides software programs with machine-interpretable metadata of the published information and data. In other words, we add further data descriptors to otherwise existing content and data on the Web. As a result, computers are able to make meaningful interpretations similar to the way humans process information to achieve their goals.

The ultimate ambition of the Semantic Web, as its founder Tim Berners-Lee sees it, is to enable computers to better manipulate information on our behalf. He further explains that, in -processable or what a machine is able to do with the data. Whereas conveys the idea of a navigable space of interconnected objects with mappings from URIs to resources.

Apache Jena:

Jena is a Java framework for building Semantic Web applications. It provides a extensive Java libraries for helping developers develop code that handles RDF, RDFS, RDFa, OWL and SPARQL in line with published W3C recommendations. Jena includes a rule-based inference engine to perform reasoning based on OWL and RDFS ontologies, and a variety of storage strategies to store RDF triples in memory or on disk.

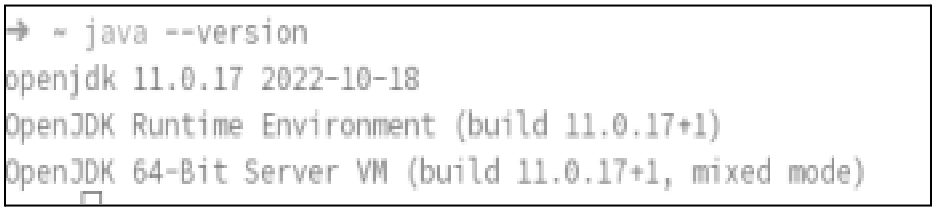
Protégé:

Protégé is a free, open source ontology editor and a knowledge management system. The Protégé meta-tool was first built by Mark Musen in 1987 and has since been developed by a team at Stanford University. The software is the most popular and widely used ontology editor in the world.

Protégé provides a graphic user interface to define ontologies. It also includes deductive classifiers to validate that models are consistent and to infer new information based on the analysis of an ontology. Like Eclipse, Protégé is a framework for which various other projects suggest plugins. This application is written in Java and makes heavy use of Swing to create the user interface.

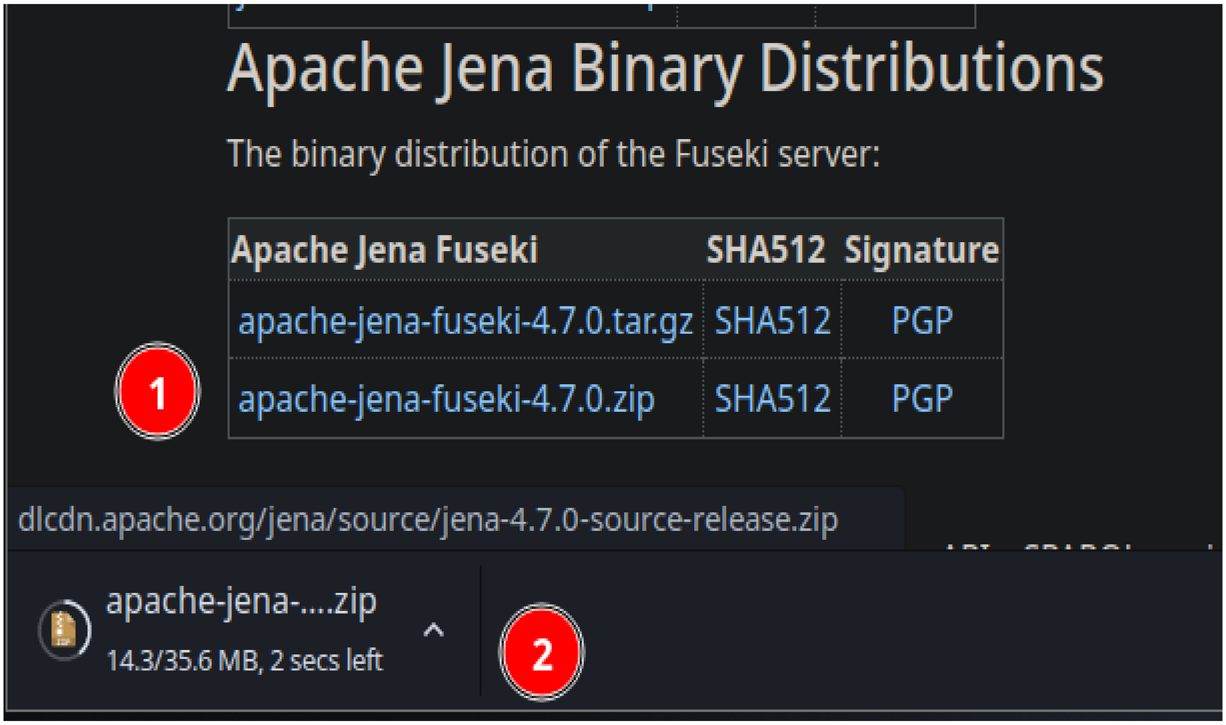
Steps for installation of Apache jena:

Step 1: Make sure that JAVA is installed in system



```
➜ ~ java --version
openjdk 11.0.17 2022-10-18
OpenJDK Runtime Environment (build 11.0.17+1)
OpenJDK 64-Bit Server VM (build 11.0.17+1, mixed mode)
```

Step 2: go to the **Apache Jena - Apache Jena Releases** and download **apache-jena-fuseki 4.7.0.zip**



Apache Jena Binary Distributions

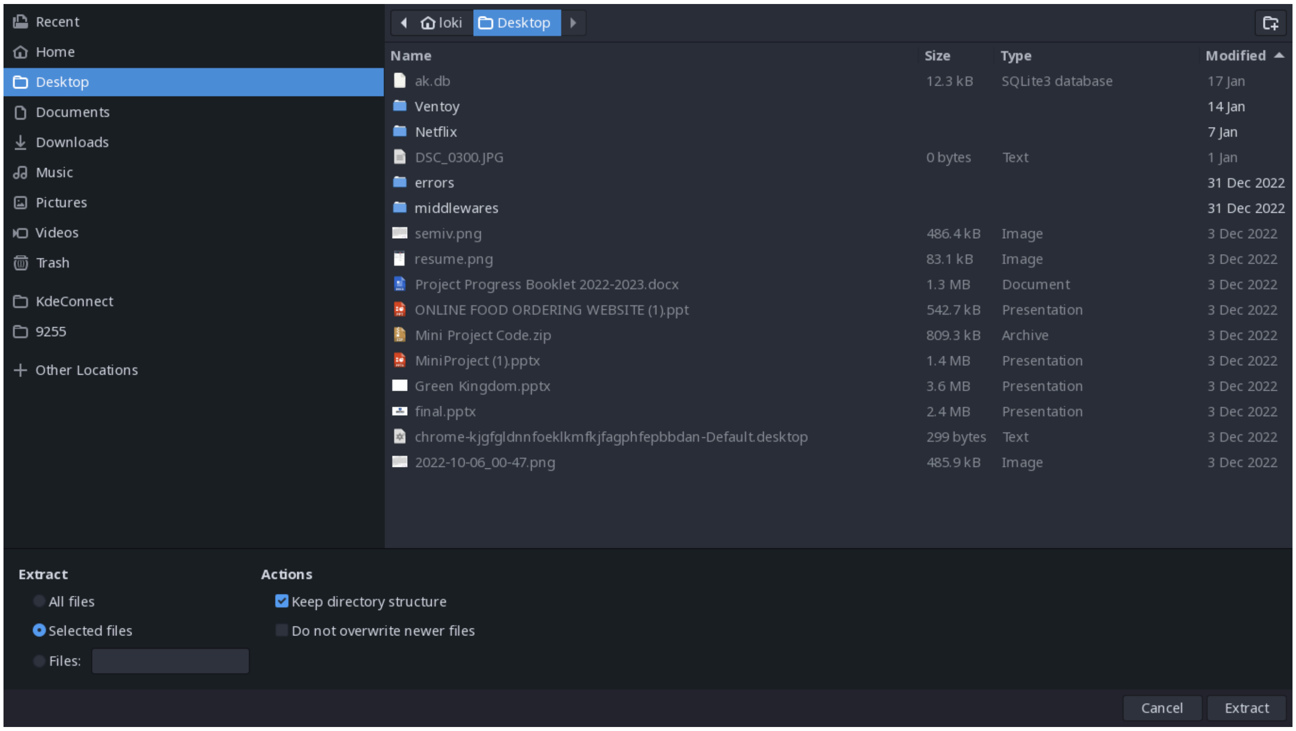
The binary distribution of the Fuseki server:

Apache Jena Fuseki	SHA512	Signature
apache-jena-fuseki-4.7.0.tar.gz	SHA512	PGP
apache-jena-fuseki-4.7.0.zip	SHA512	PGP

dlcdn.apache.org/jena/source/jena-4.7.0-source-release.zip

apache-jena-....zip
14.3/35.6 MB, 2 secs left

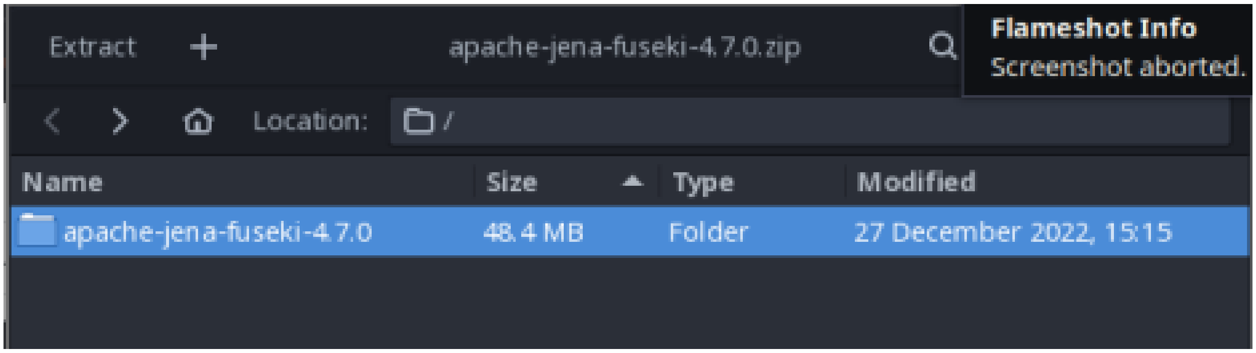
Step3 : Open Downloaded file and extract the content in some directory



Name	Size	Type	Modified
ak.db	12.3 kB	SQLite3 database	17 Jan
Ventoy			14 Jan
Netflix			7 Jan
DSC_0300.JPG	0 bytes	Text	1 Jan
errors			31 Dec 2022
middlewares			31 Dec 2022
semiv.png	486.4 kB	Image	3 Dec 2022
resume.png	83.1 kB	Image	3 Dec 2022
Project Progress Booklet 2022-2023.docx	1.3 MB	Document	3 Dec 2022
ONLINE FOOD ORDERING WEBSITE (1).ppt	542.7 kB	Presentation	3 Dec 2022
Mini Project Code.zip	809.3 kB	Archive	3 Dec 2022
MiniProject (1).pptx	1.4 MB	Presentation	3 Dec 2022
Green Kingdom.pptx	3.6 MB	Presentation	3 Dec 2022
final.pptx	2.4 MB	Presentation	3 Dec 2022
chrome-kjgfgldnnfoeklkmfkjfagphfepbbdan-Default.desktop	299 bytes	Text	3 Dec 2022
2022-10-06_00-47.png	485.9 kB	Image	3 Dec 2022

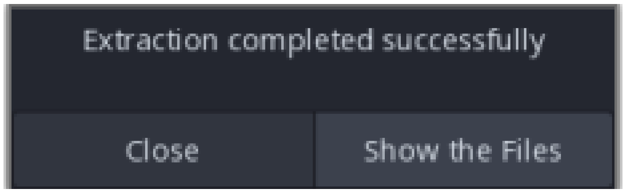
Extract
○ All files
● Selected files
○ Files:

Actions
☑ Keep directory structure
□ Do not overwrite newer files

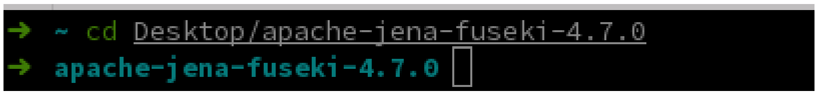


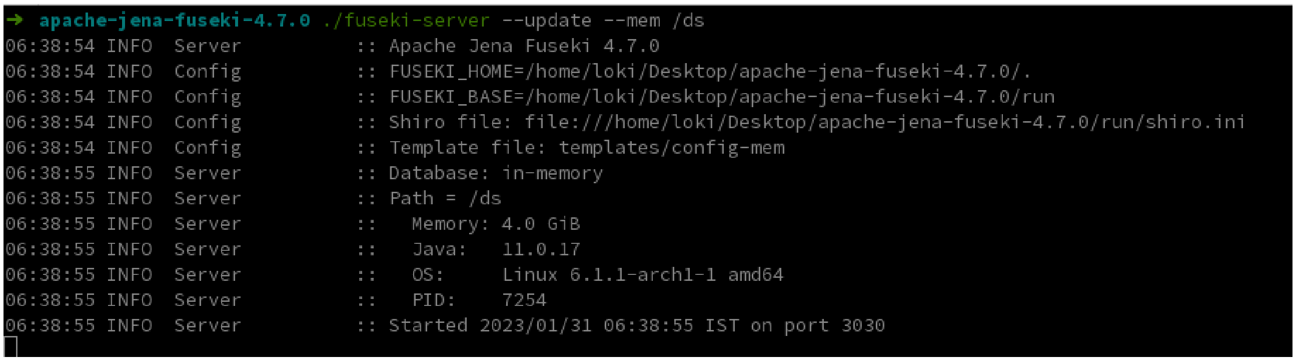
Location: /

Name	Size	Type	Modified
apache-jena-fuseki-4.7.0	48.4 MB	Folder	27 December 2022, 15:15


Extraction completed successfully

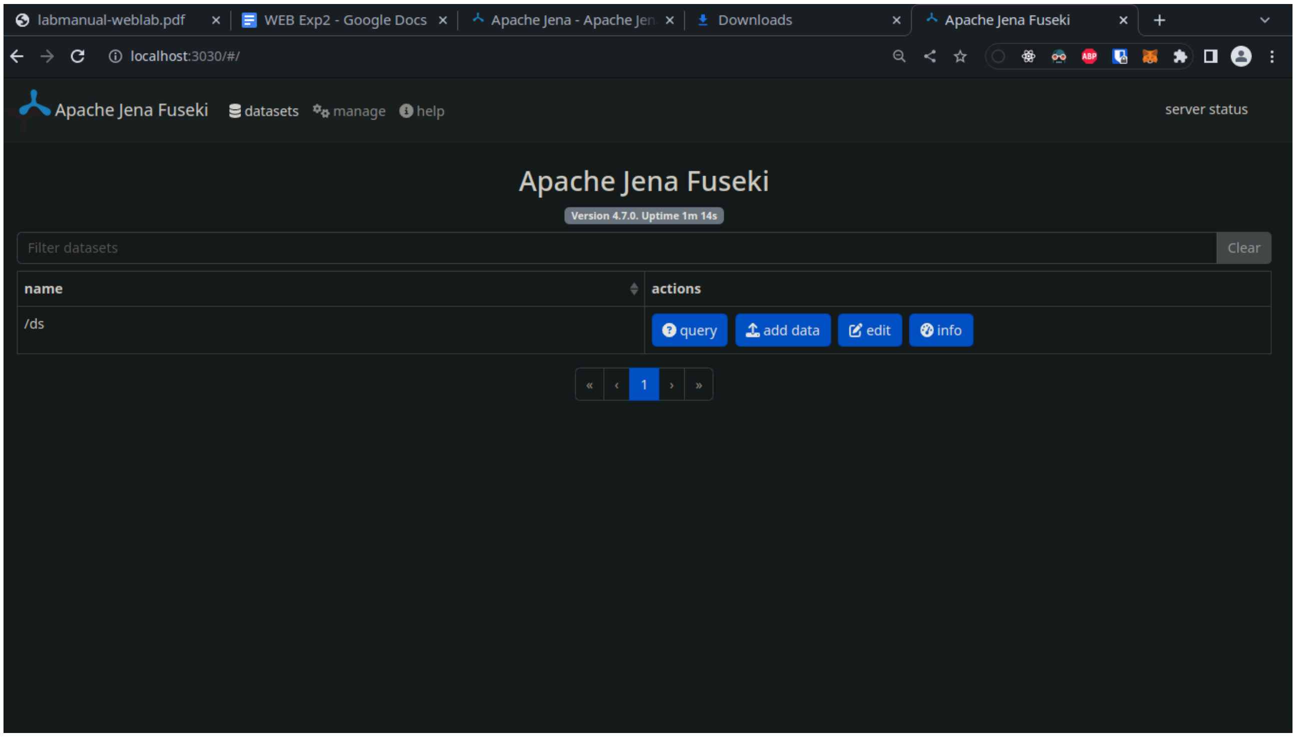
Step 4: Now navigate to directory where contents are extracted and type the following command ***./fuseki-server --update --mem /ds***


```
➜  ~ cd Desktop/apache-jena-fuseki-4.7.0
➜  apache-jena-fuseki-4.7.0
```



```
➜ apache-jena-fuseki-4.7.0 ./fuseki-server --update --mem /ds
06:38:54 INFO  Server          :: Apache Jena Fuseki 4.7.0
06:38:54 INFO  Config          :: FUSEKI_HOME=/home/loki/Desktop/apache-jena-fuseki-4.7.0/.
06:38:54 INFO  Config          :: FUSEKI_BASE=/home/loki/Desktop/apache-jena-fuseki-4.7.0/run
06:38:54 INFO  Config          :: Shiro file: file:///home/loki/Desktop/apache-jena-fuseki-4.7.0/run/shiro.ini
06:38:54 INFO  Config          :: Template file: templates/config-mem
06:38:55 INFO  Server          :: Database: in-memory
06:38:55 INFO  Server          :: Path = /ds
06:38:55 INFO  Server          ::   Memory: 4.0 GiB
06:38:55 INFO  Server          ::   Java:   11.0.17
06:38:55 INFO  Server          ::   OS:     Linux 6.1.1-arch1-1 amd64
06:38:55 INFO  Server          ::   PID:    7254
06:38:55 INFO  Server          :: Started 2023/01/31 06:38:55 IST on port 3030
```

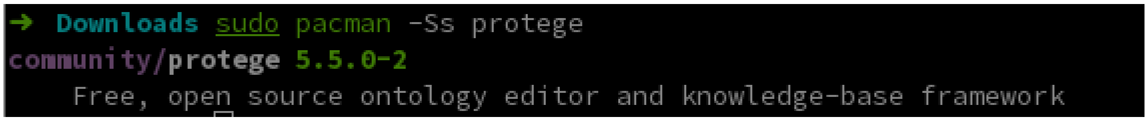
Step 5: Now open the port on the apache jena is started that is port 3030



Apache jena is server up and running

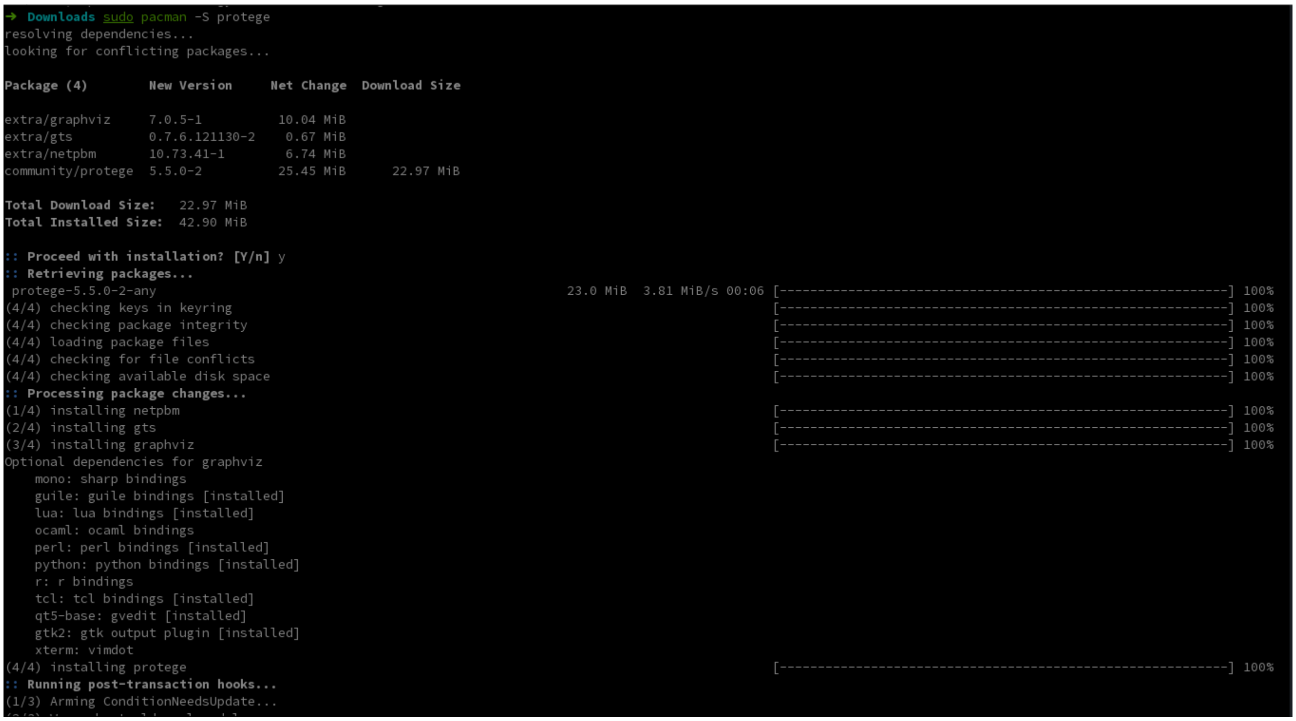
Steps for installation of Protege:

Step 1: On linux based system use package manager to search Protege



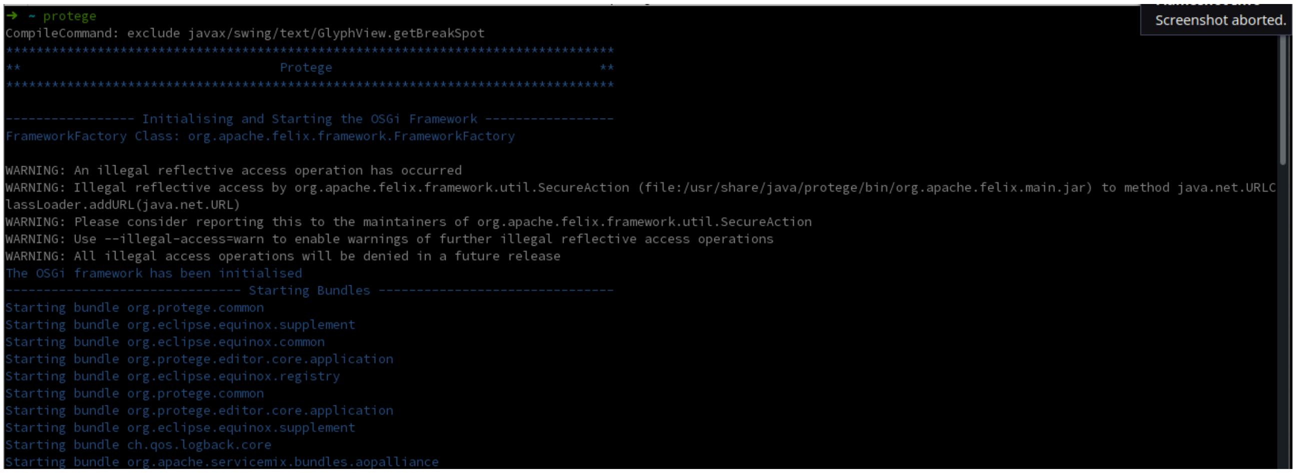
```
➜ Downloads sudo pacman -Ss protege
community/protege 5.5.0-2
    Free, open source ontology editor and knowledge-base framework
```

Step 2: Now that the package is available then install it

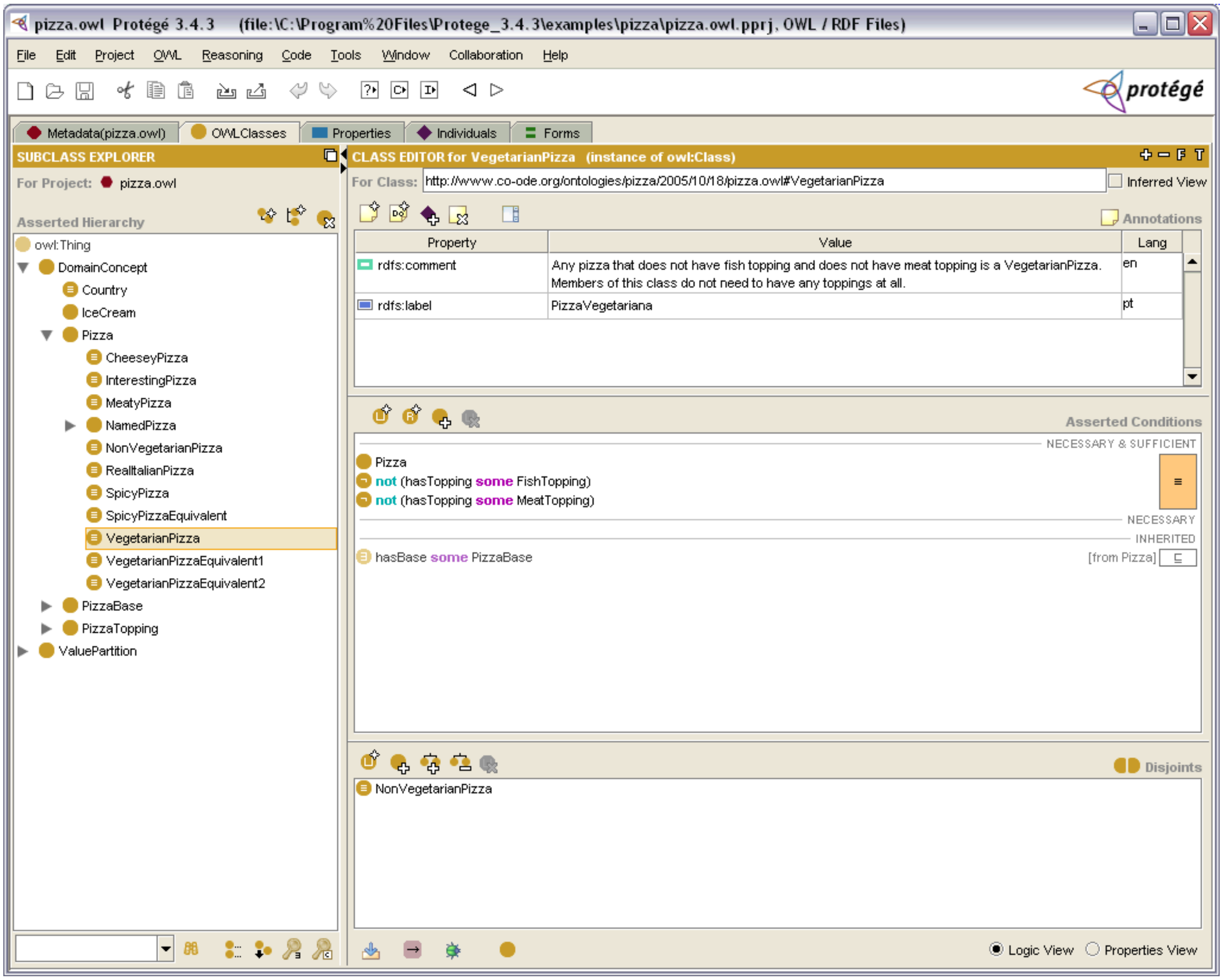


```
➜ Downloads sudo pacman -S protege
resolving dependencies...
looking for conflicting packages...

Package (4)        New Version     Net Change  Download Size

extra/graphviz     7.0.5-1          10.04 MiB
extra/gts          0.7.6.121130-2    0.67 MiB
extra/netpbm       10.73.41-1        6.74 MiB
community/protege  5.5.0-2          25.45 MiB      22.97 MiB

Total Download Size:   22.97 MiB
Total Installed Size:  42.90 MiB

:: Proceed with installation? [Y/n] y
:: Retrieving packages...
 protege-5.5.0-2-any                                                      23.0 MiB  3.81 MiB/s 00:06 [----------------------------------------------------------] 100%
(4/4) checking keys in keyring                                                                       [----------------------------------------------------------] 100%
(4/4) checking package integrity                                                                     [----------------------------------------------------------] 100%
(4/4) loading package files                                                                          [----------------------------------------------------------] 100%
(4/4) checking for file conflicts                                                                    [----------------------------------------------------------] 100%
(4/4) checking available disk space                                                                  [----------------------------------------------------------] 100%
:: Processing package changes...
(1/4) installing netpbm                                                                              [----------------------------------------------------------] 100%
(2/4) installing gts                                                                                 [----------------------------------------------------------] 100%
(3/4) installing graphviz                                                                            [----------------------------------------------------------] 100%
Optional dependencies for graphviz
    mono: sharp bindings
    guile: guile bindings [installed]
    lua: lua bindings [installed]
    ocaml: ocaml bindings
    perl: perl bindings [installed]
    python: python bindings [installed]
    r: r bindings
    tcl: tcl bindings [installed]
    qt5-base: gvedit [installed]
    gtk2: gtk output plugin [installed]
    xterm: vimdot
(4/4) installing protege                                                                             [----------------------------------------------------------] 100%
:: Running post-transaction hooks...
(1/3) Arming ConditionNeedsUpdate...
```

Step 3: now open terminal and run protege command



```
➜ ~ protege
CompileCommand: exclude javax/swing/text/GlyphView.getBreakSpot
********************************************************************************
**                                  Protege                                   **
********************************************************************************

----------------- Initialising and Starting the OSGi Framework -----------------
FrameworkFactory Class: org.apache.felix.framework.FrameworkFactory

WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.felix.framework.util.SecureAction (file:/usr/share/java/protege/bin/org.apache.felix.main.jar) to method java.net.URLC
lassLoader.addURL(java.net.URL)
WARNING: Please consider reporting this to the maintainers of org.apache.felix.framework.util.SecureAction
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
The OSGi framework has been initialised
------------------------------- Starting Bundles -------------------------------
Starting bundle org.protege.common
Starting bundle org.eclipse.equinox.supplement
Starting bundle org.eclipse.equinox.common
Starting bundle org.protege.editor.core.application
Starting bundle org.eclipse.equinox.registry
Starting bundle org.protege.common
Starting bundle org.protege.editor.core.application
Starting bundle org.eclipse.equinox.supplement
Starting bundle ch.qos.logback.core
Starting bundle org.apache.servicemix.bundles.aopalliance
```



pizza.owl Protégé 3.4.3 (file:\C:\Program%20Files\Protege_3.4.3\examples\pizza\pizza.owl.pprj, OWL / RDF Files)

For Project: pizza.owl

CLASS EDITOR for VegetarianPizza (instance of owl:Class)

For Class: http://www.co-ode.org/ontologies/pizza/2005/10/18/pizza.owl#VegetarianPizza ☐ Inferred View

Property	Value	Lang
rdfs:comment	Any pizza that does not have fish topping and does not have meat topping is a VegetarianPizza. Members of this class do not need to have any toppings at all.	en
rdfs:label	PizzaVegetariana	pt

Pizza
not (hasTopping some FishTopping)
not (hasTopping some MeatTopping)

hasBase some PizzaBase [from Pizza]

NonVegetarianPizza

Conclusion: Semantic web technologies like Apache TinkerPop, RDFLib, and Protege enable machines to understand and interpret linked data efficiently. These tools facilitate knowledge representation, data integration, and reasoning, which are essential for applications in artificial intelligence and data science. Learning these technologies enhances the ability to work with structured web data and ontologies.