This project has received funding from the EU Erasmus+ Programme under grant agreement 2021-1-EL02-KA220-YOU-000028780

Co-funded by the Erasmus+ Programme of the European Union

BRIDGING THE GAP

An AI-enabled versatile skill matching tool to assist the less privileged

BRIDGING THE GAP DELIVERABLE 1.2

Authors:	International Hellenic University [IHU]
Status:	Final
Due Date:	31/12/2022
Version:	1.0
Dissemination Level:	PU

BRIDGING THE GAP Project Profile

Grant Agreement No.: 2021-1-EL02-KA220-YOU-000028780

Acronym:	BRIDGING THE GAP
Title:	Bridging the Gap – An AI-enabled versatile skill matching tool to assist the less privileged
URL:	https://bridgingthegapproject.eu/
Start Date:	14/02/2022
Duration:	24 months

Partners

INTERNATIONAL HELLENIC UNIVERSITY	DIETHNES PANEPISTIMIO ELLADOS (IHU)	Greece
MYCOMPANY	MY COMPANY PROJECTS O.E.	Greece
	UNIVERSITATEA DIN CRAIOVA	Romania
1992	Regional center for vocational training and education to CCI-Blagoevgrad	Bulgaria

Document History

Version	Date	Author (Partner)	Remarks/Changes
0.1	07/12/2022	Kalliopi Kravari (IHU)	Table of Contents
0.2	12/12/2022	Kostis Ioannis – Aris (My Company Projects O.E.), Dimitrios Sarafis (My Company Projects O.E.), Chenyang Ji (IHU), Athanasios Kravaris (IHU), Costin Badica (UCV)	1st Draft ready for internal review
0.3	27/12/2022	Kalliopi Kravari (IHU)	2nd Draft ready for quality control
1.0	31/12/2022	Periklis Chatzimisios (IHU)	FINAL VERSION TO BE SUBMITTED

Abbreviations and acronyms

Deliverable	D
Expected Outcomes	EO
International Hellenic University	IHU
Non-governmental organization	NGO

Executive Summary

BRIDGING THE GAP is a 24 month duration project funding from the European Union's Erasmus+: KA220-YOU under Grant Agreement 2021-1-EL02-KA220-YOU-000028780.

The overarching objective of the BRIDGING THE GAP project is to provide a holistic approach beyond a classical skill-matching system to a system that will bridge the gap as to who are the underprivileged and why are they underprivileged and how education and skill improvement will benefit them.

The main purpose of this document is to a report the progress of the BRIDGING THE GAP project during the Deliverable 1.2. More specifically, this deliverable reports on Project Results 1 findings that investigate and test available open-source technologies suitable for the solution area (e.g. standards, semantic tools, agent technology etc.) while produce initial illustrative demos/prototypes of the solution area. The current document reports on the investigation on mockup/prototypical solutions based on available open-source tools. The team tested most of the solutions in order to decide technologies to be used in other tasks. Furthermore, the team produced some initial prototypes, used for dissemination purposes and presented in a conference. The internal work and document were refined continuously every 3 months, as a result, the current document depicts the final updated content of this work.

Table of Contents

List of Figures & Tables

1 Introduction

1.1 Purpose of the document

The purpose of this document is to present the progress of the project during the first implementation phase regarding the implemented research activities as they are reported in Project Results 1 (Deliverable 1.2).

1.2 Intended audience

The intended audience of this document consists of the following target groups:

- BRIDGING THE GAP project partners and the Project Officer at the National Agency
- Young people, especially on the Balkan area, that are interested in skill building/matching
- Labour market actors
- Universities, course providers

1.3 Work Package Objective

The current deliverable reports on the investigation on solutions based on available open-source tools. The team tested most of the solutions in order to decide technologies to be used in other tasks. Furthermore, the team produced some initial prototypes. The internal work and document was refined continuously every 3 months, as a result, the current document depicts the final updated content of this work.

1.4 Structure of the document

In Chapter 2, this report describes the open educational and vocational resources that were examined at the first phase of the project. Some of them will be used at the next implantation phase.

In Chapter 3, this report presents the technical resources focusing on the recommended resources and their review and the finally ending up to the utilized resources that cover the needs of the project.

In chapter 4, this report presents a first review of intelligent agents and their added-value.

Chapter 5 concludes the study presented in the current report.

2 Open Educational and Vocational Resources

The findings presented in this report are based on issues concerning mainly the aggregation of open-source resources that will be utilized in the project by all stakeholders in the implementation, including data resources or technical resources. Part of the contents involve feedback by MY COMPANY PROJECTS O.E.. the leading partner of the following Project Results 2 "Tools development and Integrations with external systems & semantic intelligent bridging", on the already examined resources by partners on the project (as stated in Deliverable 1.1) and additions to these. It is worth mentioning that this work follows a bottom-up methodology investigating the problem of formalising the lifelong learning process in a dynamic and flexible way while it utilises a parallel top-down approach in applying semantics and standards upon data in order to alleviate the gap among individuals, workplaces and educational contexts for the benefit of all in a transparent way.

For purposes of better understanding the following subsections are formed based on a specific template that will allow a concrete and solid representation of the resources:

Resource Template	
Resource Title:	
Resource URI/URL:	
Language Provided:	
Notes on Content:	
Notes on Accessibility:	
Conclusion:	

2.1 Educational

Plenty of online educational resources were examined (see Figure 1). Some of the most worth mentioning are presented below.

Site	Type	Content	Area	Certificate	Duration	Cost	Job relevant info	Language	Accessibility	Requirement
opencourses.gr	Education	Study materials		No	No specific	free	No	En / Gr		High
https://www.eduguide.gr/	Education	Diploma programmes		Yes	Specific	Based	News / Advice	GR		High
https://www.coursera.org/	Education	Skill certificates / Diploma programmes		Yes	Specific	Based	Career advice	En		Medium low - High
https://www.udemy.com/	Education	Lesson certificate		Yes(?)	No specific	Based	No	En		Medium low - High
http://www.gsae.edu.gr/	Education	Skill certificates / Adult education		Yes	Specific	(?)		GR		Low
https://online.stanford.edu/	Education	Diploma programmes		Yes	Specific	Based	No	En		Medium low - High
https://ocw.mit.edu/	Education	Study materials		No	No specific	Free	No	En		High
https://oyc.yale.edu/	Education	Study materials		No	No specific	Free	No	En		High
https://www.edx.org/	Education	Skill certificates / Diploma programmes		Yes	Specific	Based	No	En		Medium low - High
https://docs.microsoft.com/zh-cn/learn/	Education	Skill certificate		Yes	No	Based	No	En		Medium high - High
https://www.linkedin.com/learning	Education	Lesson certificate		Yes(?)	No	Based	No	En		Medium high - High
online?source=AdW-	Education	Diploma programmes		Yes	Specific	Based	No	En		Medium high - High
https://elearning.upt.ro/	Education	Diploma programmes		Yes	Specific	Based	No	Ro		Medium high - High
https://www.link-academy.com/	Education	Skill certificates / Diploma programmes		Yes	Specific	Based	Career advice	Ro		Medium heigh
https://www.telacad.ro/	Education	Skill certificates		Yes	Specific	Based	Career advice	Ro		Medium - Heigh
https://scoalainformala.ro//	Education	Skill certificates		Yes	Specific	Based	Career advice	Ro		Medium
https://scoalainformala.ro/	Education	Lesson certificate		Yes	Specific	Based	Career advice	RO		Low
https://mooc.ro	Education	Study materials		No	No specific	Free	No	RO		Low
https://andrei.clubcisco.ro	Education	Study materials		No	No specific	Free	No	RO		Medium low - High
https://anis.ro	Education	Skill certificates		Yes	Specific	Based	Career advice	RO		Low
https://www.itschool.ro	Education	Lesson certificate		Yes	Specific	Based	Career advice	RO		Low
https://codecool.com/ro/	Education	Skill certificates		Yes	Specific	Based	Career advice	RO/EN/DE/PL/HU		Low
https://unicampus.ro	Education	Study materials		No	No specific	Free	No	RO		Medium high - High
https://wellcode.ro	Education	Skill certificates		Yes	Specific	Free/Based	News/Advice	RO		Low
https://wellcode.com	Education	Skill certificates		Yes	Specific	Free/Based	News/Advice	EN		Low
https://www.udacity.com	Education	Skill certificates		No	Specific	Based	Career advice	EN		Medium low - High

https://www.freecodecamp.org	Education	Skill certificates		Yes	No specific	Free	No	EN		Low
https://www.khanacademy.org	Education	Study materials		No	No specific	Free	No	EN		Low
https://www.codecademy.com	Education	Lesson certificate		Yes	No specific	Free	No	EN		Low
https://www.open.edu/openlearn/	Education	Lesson certificate		Yes	No specific	Free	No	EN		Medium low - High
https://www.sophia.org	Education	Study materials		No	Specific	Free	No	EN		Low
https://alison.com	Education	Skill certificates / Diploma programmes		Yes	Specific	Free/Based	News/Advice	EN		Medium high - High
https://www.w3schools.com	Education	Skill certificates / Study materials		Yes	Specific	Free/Based	No	EN		Low
https://www.invata-programare.ro/	Education	Study materials	Programming	No	No specific	Free	No	RO		
https://www.pythonisti.ro/	Education	Study materials	Programming	No	No specific	Free/Based	No	RO		
https://codeberryschool.com/ro/	Education	Courses	Programming	No	No specific	Based	Career advice	RO		
https://codecool.com/ro/	Education	Courses	Programming	No	Specific	Based	Career advice	RO		
https://facemsoft.ro/en/	Education	Video courses/code snippets	Programming	No	No specific	Free	No	RO		
https://www.sqsgroup.ro/ro-ro/certification	Education	Lesson certificate		No	No specific	Free	No	RO/EN/Many others		
https://www.tuv-austria.ro/	Education	Lesson certificate		No	No specific	Based	No	RO		
https://academicearth.org/	Education	Study materials		No	No specific	Free	No	EN		
https://htmldog.com/	Education	Beginer study materials	Web Dev	No	No specific	Free	No	EN		
https://github.com/	Education	Coding resources		No	No specific	Based	No	EN		
https://unplugthetv.com/	Education	Video resources		No	No specific	Free	No	EN		
https://www.codecademy.com/	Education	Study materials	Programming	No	No specific	Based	No	EN		
https://learncodethehardway.org/	Education	Study materials	Programming	No	No specific	Based	No	EN		
https://rubymonk.com/	Education	Study materials	Programming	No	No specific	Free	No	EN		
age/	Education	Courses		No	No specific	Free	No	EN		

Figure 1: List of educational resources

2.1.1 OpenCourses

Resource Title:	OpenCourses
Resource URI/URL:	opencourses.gr
Language Provided:	GR/EN
Notes on Content:	Wide range of subjects, for beginners and intermediate learners. Academic context.
Notes on Accessibility:	Mainly developed in Greek, the majority of subjects are well-structured also in English. Page needs to be web-crawled in order to retrieve the info on the subjects, as there is no access provided by some form of API.
Conclusion:	Content is highly valuable for the project, and is most likely to be utilized in the implementation.

2.1.2 eduguide.gr

Resource Title:	eduguide.gr
Resource URI/URL:	eduguide.gr
Language Provided:	GR
Notes on Content:	Search engine for MSc's and PhD's. European and international studies.
Notes on Accessibility:	Developed in Greek. Curriculum page of MSc's does not provide info on the content of each subject included and provided.
Conclusion:	This website does not provide any substantial educational or vocational information. It is also only developed in Greek, making any info extracted unusable for international users. We advise against in its utilization in the implementation.

2.1.3 Coursera

Resource Title:	Coursera
Resource URI/URL:	www.coursera.org
Language Provided:	EN
Notes on Content:	Probably the most popular provider of online courses in the market. Wide range of subjects, descriptions and curricula very well-structured, tags provided on prospective acquired skills are a plus.

Notes on Accessibility:	This website provides great content that could be utilized in our project. However, it is inaccessible by API, and way too extensive to be web-crawled.
Conclusion:	Due to its popularity, some small extent of its content can be utilized in order to provide proof of scalability-expandability of our project-software via automated web procedures. We advise that these cases will be cherry-picked in our favor.

2.1.4 Udemy

Resource Title:	Udemy
Resource URI/URL:	udemy.com
Language Provided:	EN
Notes on Content:	One of the most popular provider of online courses in the market. Wide range of subjects, descriptions and curricula very well-structured.
Notes on Accessibility:	This website provides great content that could be utilized in our project. It is accessible via API, provided for free upon application[1].
Conclusion:	Even if accessible by API, there is a restriction on API requests. As so, we should avoid basing our implementation on calling it on demand for every user of our software. However, it could be utilized to some extent by preloading some of the most popular courses provided, in order to prove some utility on it also. In further, production-grade implementations of our project, a software module could be developed that crawls and loads this website's content into our system periodically. We advise against doing it now, as the process seems to be resource and computational heavy.

2.1.5 General Secretariat for Professional, Vocational and Lifelong Learning

Resource Title:	General Secretariat for Professional, Vocational and Lifelong Learning
Resource URI/URL:	www.gsae.edu.gr
Language Provided:	GR
Notes on Content:	Informational portal of the Greek General Secretariat for Professional, Vocational and Lifelong Learning. Provides updates on several platforms and opportunities concerning Vocational Training and Lifelong Learning.
Notes on Accessibility:	This website provides virtually no information that can be utilized in our project. Slim possibility of partial incorporation of a small part of the info provided, needs to be handpicked and manually handled.
Conclusion:	We advise against any further research on it at this stage of the project.

2.1.6 Stanford Online

Resource Title:	Stanford Online
Resource URI/URL:	online.stanford.edu
Language Provided:	EN

[1] https://www.udemy.com/developers/affiliate/

Notes on Content:	Wide range of open access (however not entirely free) subjects provided by the Stanford University. Very high quality of provided education, highly esteemed prospective acquired qualifications. Some courses are also provided by Coursera, some require a fee, and most of them have prerequisite knowledge in order to be able to be attended.
Notes on Accessibility:	There is an available API provided for standardized access to the courses' information. Gaining access to it is feasible, yet complicated. We provide some useful links: • Developer APIs[2] • API Devs - Communities of Practice[3] • Digital Certificate Acquisition[4] • API Devs access request webpage[5]
Conclusion:	We highly advise further research and consideration on whether incorporating courses with such financial and educational prerequisites serves our project's purpose. Their integration and utilization seems feasible.

2.1.7 MIT OpenCourseWare

Resource Title:	MIT OpenCourseWare
Resource URI/URL:	ocw.mit.edu
Language Provided:	EN
Notes on Content:	Wide range of open access subjects provided by the MIT University. Very high quality of provided education, highly esteemed prospective acquired qualifications. Majority of courses are purposed towards prospective students with a certain degree of existing (academic) knowledge on them. Very well-structured syllabus description and provided material. The majority of courses (if not in their entirety) are free.
Notes on Accessibility:	There is an available API provided for standardized access to the courses' information. Gaining access to it is feasible, yet it requires an application[6].
Conclusion:	We advise further research and consideration on whether incorporating courses with such educational prerequisites serves our project's purpose. Their integration and utilization seem feasible.

2.1.8 Open Yale Courses

Resource Title:	Open Yale Courses
Resource URI/URL:	oyc.yale.edu
Language Provided:	EN
Notes on Content:	Wide range of open access subjects provided by the Yale University. Very high quality of provided education, highly esteemed prospective acquired qualifications.

[2] https://uit.stanford.edu/developers/apis
[3] https://cop.stanford.edu/community/api-developers
[4] https://uit.stanford.edu/service/registry/certificates#panel2e
[5] https://mailman.stanford.edu/mailman/listinfo/api_developers
[6] https://ist.mit.edu/apis

	Majority of courses are purposed towards prospective students with a certain degree of existing (academic) knowledge on them. Very well-structured syllabus description and provided material. The majority of courses (if not in their entirety) are free.
Notes on Accessibility:	There is an available API provided for standardized access to the courses' information. Gaining access to it is feasible, yet it requires an application[7].
Conclusion:	We advise further research and consideration on whether incorporating courses with such educational prerequisites serves our project's purpose. Their integration and utilization seem feasible.

2.1.9 edX Courses

Resource Title:	edX Courses
Resource URI/URL:	www.edx.org
Language Provided:	EN
Notes on Content:	Portal that aggregates open (free) courses from top universities all over the world. Educational prerequisites apply and depend on the original provider of a course.
Notes on Accessibility:	This website constitutes a mediator between major universities providing free open courses, and prospective students. It does not provide any exclusive content itself. Conveniently, it possesses an exclusive API[8] for standardized access to its contents.
Conclusion:	Should we agree to utilize high-level academic content by any major university (such as the three aforementioned, or any included in edX Courses services) this website's interconnection to them could prove useful as a medium of navigation. Some technical restrictions (API calls per minute) could still apply. However, these restrictions apply in almost any similar case.

2.1.10 Microsoft Learn

Resource Title:	Microsoft Learn
Resource URI/URL:	docs.microsoft.com/en-us/learn/
Language Provided:	EN
Notes on Content:	Microsoft learning platform providing free courses, mainly involving their products. Considering the market value of them, this should theoretically be one of the most valuable available resources to our projects, especially with an IT - CS target market/group.
Notes on Accessibility:	Courses' syllabus and description are not especially well-structured, and the content is more product-oriented. An API for standardized access exists[9]. Technical restrictions could apply.
Conclusion:	We suggest utilizing the API for frequent dynamic updating of certain, selected courses. The selection should include the most popular Microsoft products available

[7] https://developers.yale.edu/
[8] https://courses.edx.org/api-docs/?format=openapi
[9] https://docs.microsoft.com/en-us/learn/support/catalog-api#catalog-api-program

	and interconnected with the current IT market (MS Windows, MS Office, MS Azure, MS PowerBI, etc.).

2.1.11 LinkedIn Learning

Resource Title:	LinkedIn Learning
Resource URI/URL:	linkedin.com/learning
Language Provided:	EN
Notes on Content:	LinkedIn's learning platform. Focused, quality content to acquire soft skills, technical skills and interview preparation skills. Require monthly subscription.
Notes on Accessibility:	This resource can be accessed in a standardized way via the relative API[10]. However, it might prove to be very difficult and time-consuming to be granted access to it by LinkedIn. Furthermore, if we chose to manually access the courses to cherry-pick integrate them to our implementation, a LinkedIn Premium account is required.
Conclusion:	Upon consideration, we advise against the utilization of this resource in this stage of the project. Should the project reach production level development stages in order to be used commercially, then this feature should be a must inclusion.

2.1.12 wellcode

Resource Title:	wellcode
Resource URI/URL:	https://wellcode.com/
Language Provided:	EN
Notes on Content:	Wellcode offers a course catalog of top programming languages and supports online education to specific combination of programming languages, job skills, and the critical thinking capability candidates needs to technology jobs.
Notes on Accessibility:	This website provides virtually no information that can be utilized in our project. Slim possibility of partial incorporation of a small part of the info provided, needs to be handpicked and manually handled.
Conclusion:	We advise against any further research on it at this stage of the project.

2.1.13 https://academicearth.org/

Resource Title:	academicearth
Resource URI/URL:	https://academicearth.org/
Language Provided:	EN
Notes on Content:	Academic Earth provides collection of free online college courses from the world's top universities.
Notes on Accessibility:	The curated lists of online courses are hand selected by their staff. Slim possibility of partial incorporation of a small part of the info provided, needs to be handpicked and manually handled.

[10] https://docs.microsoft.com/en-us/linkedin/learning/overview/

Conclusion:	We advise against any further research on it at this stage of the project.

2.2 Vocational Resources

Another group of resources that were examined is that of vocational resources (see Figure 2). Some of the most worth mentioning are presented below, following the same template.

Site	Type	Content	Accessibility	User experienc	Region covered	Language
https://www.linkedin.com/	Web platform	1 - 5 - 6 - 7 - 9			Worldwide	En
https://www.glassdoor.com/	Web platform	1 - 2 - 4 - 5 - 7			Worldwide	En
https://ec.europa.eu/eures	Web platform	1 - 3 - 5			Europe	En
https://www.monster.com/	Web platform	1 - 2 - 3 - 5 - 8			Limited countries	Multiple
https://euraxess.ec.europa.eu/	Web platform	1 - 2 - 3			Worldwide	En
https://eurojobs.com/	Web platform	1 - 5			worldwide	En
https://aquent.com/	Staffing agency					En
https://www.ejobs.ro	Web platform	1 - 2 - 5 - 7			Europe	RO/EN
https://jobzz.ro	Web platform	1 - 7			Romania	RO
https://ro.jooble.org	Web platform	1 - 2 - 7			Worldwide	Multiple
https://www.hipo.ro	Web platform	1 - 2 - 5 - 7			Romania	RO
https://www.undelucram.ro	Web platform	1 - 2 - 5 - 7			Romania	RO
https://www.romjob.ro	Web platform	1			Romania	RO
https://devjob.ro	Web platform	1 - 4 - 7			Romania	RO
https://jobby.ro	Web platform	1 - 7			Romania	RO
https://www.careerbuilder.com	Web platform	1 - 2 - 5 - 7			Worldwide	EN
https://www.bestjobs.eu/en/	Web platform	1 - 2 - 7			Worldwide	EN
https://www.flexjobs.com/	Web platform	1 - 2 - 7 - 8			Worldwide	EN
https://www.theladders.com/apply4me	Web platform	1 - 2 - 5 - 7			Worldwide	EN
https://angel.co/	Web platform	1 - 6			Worldwide	EN
https://getwork.com/	Web platform	1			Worldwide	EN
https://www.myjob.ro/	Web platform	1 - 2 - 5 - 8			Romania	RO
https://www.jobber.ro/	Web platform	1			Romania	RO
http://posturi.gov.ro/	Web platform	1			Romania	RO - goverment jo
http://salesconsulting.ro/	Web platform	1			Romania	RO

Figure 2: List of vocational resources

2.2.1 LinkedIn

Resource Title:	LinkedIn
Resource URI/URL:	linkedin.com
Language Provided:	EN
Notes on Content:	One of the most popular stakeholders in the job search market. Reliable resource for Job Vacancies retrieval.
Notes on Accessibility:	Despite the existence of an API[11] for standardized automated access to the job vacancies available, gaining access to it could prove to be a difficult and time-consuming task. Manual composition of a moderate-sized dataset seems feasible as a task.
Conclusion:	We advise against the reliance of the project on the automated info retrieval via crawlers and API calls from the website. We strongly advise in favor of a dataset creation, based on job vacancies of a determined range of occupations.

2.2.2 GlassDoor

Resource Title:	GlassDoor

[11]https://docs.microsoft.com/en-us/linkedin/consumer/integrations/self-serve/sign-in-with-linkedin?context=linkedin%2Fconsumer%2Fcontext

Resource URI/URL:	glassdoor.com
Language Provided:	EN
Notes on Content:	Glassdoor is another job searching platform, with the additional feature of giving its members the opportunity to post their experiences and insights on a company they have worked for other perspective job-seekers to read. This feature serves no obvious purpose to our project, reducing this resource to an ordinary job-seeking website.
Notes on Accessibility:	Job searching on Glassdoor requires just a free account, so manual job vacancy description aggregation is feasible. A web crawler could be implemented in such case. Furthermore, Glassdoor provides an API[12] for standardized access to its contents, which, however, requires one to apply and be accepted into their partners program. At this moment, we have no further information on the matter, considering the requirements to be admitted and how long that would take.
Conclusion:	If the process of acquiring access to Glassdoor's API would be an easy task, we could take it into consideration. In any other case, it provides nothing more to the project as a resource provided that LinkedIn already has the same, and many more, job vacancies posted.

2.2.3 EURES

Resource Title:	EURES
Resource URI/URL:	ec.europa.eu/eures
Language Provided:	Mainly EN, JVs available in several other European languages
Notes on Content:	European Commission Portal for jobseekers in the Europe. With over 3 million jobs posted and already annotated by Occupation terminology provided, it constitutes an indispensable resource to our project.
Notes on Accessibility:	This resource is supposed to be accessed via API. However, its stability and availability is uncertain. There are records of it being utilized[13]. Further research will be conducted.
Conclusion:	If we will be ultimately able to access the information provided of this resource, its incorporation to the project is a must.

2.2.4 MONSTER

Resource Title:	MONSTER
Resource URI/URL:	monster.com
Language Provided:	EN
Notes on Content:	Job posting - seeking website. Mainly focused on the US job market search.
Notes on Accessibility:	Nothing to mention.
Conclusion:	Since this website is mainly focused on the US job market, we advise against any further investigation into it. We can acquire more than enough data from other

[12] https://www.glassdoor.com/developer/index.htm
[13] https://github.com/navikt/pam-eures-stilling-eksport

	resource to create a proof-of-concept stage implementation, and due to its target market being outside EU in its majority, it is inconvenient for our final implementation goal.

2.2.5 EURAXESS

Resource Title:	EURAXESS
Resource URI/URL:	euraxess.ec.europa.eu
Language Provided:	EN
Notes on Content:	Job Search Engine for academic researchers, prospective PhD students, post-doc searchers, etc.
Notes on Accessibility:	Nothing to mention.
Conclusion:	Upon investigation and consideration on the nature of the resource and its contents, we advise against its utilization in our project, mainly because it constitutes a tool focused on a ”niche” (in relative terms and volumes) market in comparison to ours. We believe that there is no point in this stage of the project to try and integrate such features into our implementation, especially when EURAXESS is one of the most popular tools in its domain. Should we decide in the future to extend our target market into academic researchers and prospective PhD students, then at that point we should re-investigate and reconsider this particular resource.

2.2.6 Eurojobs

Resource Title:	Eurojobs
Resource URI/URL:	eurojobs.com
Language Provided:	EN
Notes on Content:	Job posting - seeking site for job vacancies all over Europe.
Notes on Accessibility:	While not providing access to its contents via an API or any other standardized method, it contains a large amount of job postings from all over Europe.
Conclusion:	We are in favour of this resource as supplementary to the rest of our resources in case we manually compile a JV dataset.

2.2.7 Aquent

Resource Title:	Aquent
Resource URI/URL:	aquent.com
Language Provided:	EN
Notes on Content:	Aquent is a staffing agency, a recruiting and HR contractor. Apart from that, they maintain a job posting webpage that primarily focuses on the US job market.
Notes on Accessibility:	Nothing to mention.
Conclusion:	Since this website is mainly focused on the US job market, we advise against any further investigation into it. We can acquire more than enough data from other resource to create a proof-of-concept stage implementation, and due to its target

	market being outside EU in its majority, it is inconvenient for our final implementation goal.

3 Technical Resources

In this section, we present the range of (open source) resources we considered for the initial version of our deliverable software module, as well as the range of resources we used in our initial - proof-of-concept implementation.

3.1 Considered Resources

3.1.1 ESCO Framework

ESCO[14] is the multilingual classification of European Skills, Competences, and Occupations. The ESCO classification identifies and categorizes skills, competences, qualifications and occupations relevant for the EU labour market and education and training. It systematically shows the relationships between the different concepts. ESCO is published as Linked Open Data (LOD) so it can be easily reused and linked to other data sources. The classification is composed of modules that contain elements such as occupations, knowledge, skills and competences, qualifications, and the International Standard Classification of Occupations (ISCO) hierarchy. It is published in 26 European Languages (including EN, RO, BG and GR) as well as in Arabic. The aforementioned classification is provided through an API[15], either Local[16] or Online[17]. Furthermore, the overall Dataset is provided in either CSV formats, or RDF Turtle (TTL[18]) Graph format. Furthermore, a RESTful API[19] exists that exposes ESCO via HTTP requests.

3.1.2 RDF

RDF[20] is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed. RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a "triple"). Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications. This linking structure forms a directed, labelled graph, where the edges represent the named link between two resources, represented by the graph nodes. This graph view is the easiest possible mental model for RDF and is often used in easy-to-understand visual explanations. RDF could prove to be a highly effective tool in our effort, as our task is to essentially interlink several online resources that share a certain relation (skills - occupations - CVs - Jobs - Courses).

3.1.3 Python

Python[21] is a high-level, interpreted programming language, with a very short instruction set. Its full potential relies on the utilization of several user-created libraries and add-ons, each specialized for a specific purpose. Python excels on the subject of data wrangling and handling by utilizing both internal

[14] https://esco.ec.europa.eu/en/about-esco
[15] https://esco.ec.europa.eu/en/use-esco/use-esco-services-api
[16] https://esco.ec.europa.eu/en/use-esco/use-esco-services-api/esco-local-api
[17] https://esco.ec.europa.eu/en/use-esco/use-esco-services-api/esco-web-service-api
[18] https://en.wikipedia.org/wiki/Turtle (syntax)
[19] https://ec.europa.eu/esco/api/doc/esco api doc.html
[20] Retrieved from https://www.w3.org/RDF/
[21] https://www.python.org/

data structures, such as lists[22] and dictionaries[23], as well as data structures provided by external libraries, such as Dataframes[24] from Pandas[25] and arrays[26] from Numpy[27]. Python is a popular, very well documented, highly operational open source tool that could certainly prove to be an asset in our effort during data handling and manipulation.

3.1.4 Apache Jena

Apache Jena[28] is primarily a Java implemented framework for Semantic Web Applications. It provides an environment for Linked Open Data handling, a lightweight, scalable Triples Data Base (TDB) with a SPARQL[29] query layer[30], and a SPARQL Server[31]. Even though Python could provide ways of Linked Open Data handling via the "RDFLib" library[32], utilizing a native framework for RDF Turtle Graph handling and a server with scalability and workload handling could prove to be an integral asset to the performance and optimization of our implementation.

3.1.5 Markup Language formatting: CVs, JVs, UoLs

One way of formatting data on the (Semantic) Web is via the usage of Markup Languages. Two of the most popular applications of Markup Languages are HTML (HyperText Markup Language) and XML (Extensible Markup Language). XML is a markup language similar to HTML, but without predefined tags to use. Instead, one defines their own tags designed specifically for their needs. This is a powerful way to store data in a format that can be stored, searched, and shared. Most importantly, since the fundamental format of XML is standardized, if you share or transmit XML across systems or platforms, either locally or over the internet, the recipient can still parse the data due to the standardized XML syntax. RDF[33] is also a form of XML. In the scope of our needs, XML comes in several Schemas[34], some of which service HR related data. The primary of those Schemas is the HR-XML Schema[35], which channels HR related data through the Web, structured by distinctive tags. Such data could be Job Openings or even, in our case, Course descriptions. Furthermore, Europass[36] - a service by European Union with the purpose of CV creation - supported a special form - format, namely "EuropassXML[37]". The EuropassXML Schema utilizes specialized structure and tags to standardize the info provided by a Europass CV. Since our project is focused on servicing European Citizens under a common framework and into the job seeking and lifelong learning market, such technologies constitute a major asset for the facilitation and standardization of our implementation. It should be hereby noted that utilizing such implementations would limit our potential of expanding and scaling to different types of general input data to the system.

[22] https://docs.python.org/3/tutorial/datastructures.html#more-on-lists
[23] https://docs.python.org/3/tutorial/datastructures.html#dictionaries
[24] https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html
[25] https://pandas.pydata.org/
[26] https://numpy.org/doc/stable/reference/arrays.html
[27] https://numpy.org/doc/stable/index.html
[28] https://www.w3.org/2001/sw/wiki/Apache Jena
[29] https://www.w3.org/TR/rdf-sparql-query/
[30] https://jena.apache.org/documentation/tdb/index.html
[31] https://jena.apache.org/documentation/fuseki2/
[32] https://rdflib.readthedocs.io/en/stable/
[33] https://www.w3.org/TR/rdf-concepts/
[34] https://www.w3.org/XML/Schema
[35] https://schemas.liquid-technologies.com/hr-xml/2007-04-15/
[36] https://europa.eu/europass/en
[37] https://europa.eu/europass/system/files/2020-07/europass-xml-schema-docv3.4.00.pdf

3.1.6 NLP - based Data Retrieval

Another feasible, more scalable, yet more complex way of extracting information from CVs, JVs and UoLs is Natural Language Processing (NLP[38]) and its sub-fields Natural Language Understanding (NLU[39]) and Natural Entity Recognition (NER[40]). NLP broadly involves the automatic manipulation of natural language, like speech and text, by software. The study of natural language processing has been around for more than 50 years and grew out of the field of linguistics with the rise of computers. Its first subfield, NLU, involves transforming human language into a machine-readable format. Some of its immediate applications involve Sentiment Analysis, Topic Detection and Keyword Extraction. The second subfield mentioned, namely NER, is the task of identifying and categorizing key information (entities) in text. An entity can be any word or series of words that consistently refers to the same thing. Every detected entity is classified into a predetermined category. For example, in the sentence "I hear Berlin is wonderful in the winter" an NER machine learning (ML) model might detect the words "Berlin" and "winter" in a text and classify them as a "place" and "time" respectively. Amongst the aforementioned concepts exists a more technologically advanced solution to our problem that would enable us to extract the skills owned, required or acquired, depending on the stakeholder. However, such a solution is way more complex than a conventional, heuristic one, and should not be prioritized over it.

3.2 Utilized Resources

3.2.1 ESCO Framework

Upon conclusion of our initial research, we decided that the utilization of the Skills and Occupations Dataset provided by ESCO in the form of an RDF Turtle Graph was indispensable to our implementation. Supplementarily, we decided to also utilize the online API provided in order to translate the URIs contained in the graph that represented Skills or Occupations into human-readable text (strings). However, as effective as this was of a method, it proved to be, on one hand, computationally heavy. Loading the graph on RAM each time we wanted to run a couple of queries or a test resulted in a 2-3 minute process, making development a tedious task. On the other hand, there was a hardware limitation on how many queries a single run could execute, as also as a bandwidth limitation of ESCO's online API on how many requests could handle in a second. The latter resulted in some more complex processes being unable to complete due to IP blocking. As a solution to the graph loading and querying problem, we implemented a Triples Data Base (TDB) in Apache Jena (more info on it will be provided in a forthcoming paragraph of this section), which resulted in reducing dead times, from 2-3 minutes to 1-2 seconds for the same queries. As a solution to the online API and bandwidth throttling, we decided to transfer the overall system to a localized solution: we created Pandas Dataframes by the data provided by ESCO, containing Skills or Occupations and their respective URLs. Upon loading, we were able to match the human-readable named entities (i.e. skills or occupations) to their URIs, and vice versa, through simple processes onto the Dataframes. In the final implementation of our prototype, we discontinued the extensive usage of those Dataframes: they are used upon initial loading of the data, after which the operationally essential data are loaded into Python lists, and the Dataframes are

[38] https://en.wikipedia.org/wiki/Natural language processing
[39] https://en.wikipedia.org/wiki/Natural-language understanding
[40] ttps://en.wikipedia.org/wiki/Named-entity recognition

dropped along with the excess. In such way, we manage to reduce memory usage and increase computational speed.

3.2.2 Semantic Web Ontologies

To determine the relations between every 'data stakeholder' involved in the design, the following Semantic Web ontologies were used:

- ESCO: This is the original semantic ontology provided by the ESCO classification. It describes the relations between Skills and Occupations. Documentation can be found here[41].
- RDF & RDFS: RDF (Resource Description Framework) is a framework describing general and abstract concepts on the Web. Its Schema (RDF-S) provides further definition of relations between concepts. These two constitute the archetype and the basis upon which the rest of the Semantic Web Ontologies are created. Further documentation on RDF can be found here[42], while further documentation on RDFS can be found here[43].
- FOAF: The FOAF (Friend-of-a-friend) Ontology constitutes a vocabulary/ dictionary focused on describing relations between human-oriented entities. It is heavily used in social, representational and information networks. "foaf:Person foaf:name ?name."is a simple example of a triple from a query that retrieves every name of and RDF graph that relates to an entry. Further documentation on the FOAF ontology can be found here[44].
- SARO: The SARO (Skills and Recruitment Ontology) is a domain Ontology representing Occupations, Skills and recruitment. It is modelled by considering several similar context models, but is mainly inspired by the European Skills, Competences, Qualifications and Occupations ontology (ESCO) and Schema.org. The ontology is structured along four dimensions: job posts, skills, qualifications and users. Documentation for SARO can be found here[45].

3.2.3 Information Retrieval - Annotation

Regarding the process of designing and implementing the annotators, our initial approach involved primarily the EuropassXML Schema, and, by extension, the HR-XML Schema. Upon further research, the approach proved to be unfeasible, as the support and development of the EuropassXML schema has stopped for some years now. Even if we could implement the concept utilizing the Schema, it would prove problematic in the future. Due to those circumstances, we moved towards an NLU-NER solution. As stated in the previous section of this report, such solutions are, on the one hand, way more complex than any similar heuristic ones. On the other hand, it widened the range of available resources to us: now, every typed CV, JV and UoL can theoretically be incorporated and utilized in our project. The annotator(s) resulting from this approach can be considered universal: the same software module can be used to extract information from any CV, JV and UoL and annotate it accordingly. The implementation is based on the Natural Language Toolkit[46] (NLTK), the leading platform for building software concerning human language data processing. Through NTLK, we tokenize the text included in all the files in our disposal, and we 'map' them to entities related to the Skills and Occupations context.

[41] https://ec.europa.eu/esco/lod/static/model.html
[42] https://www.w3.org/TR/rdf-concepts/
[43] https://www.w3.org/TR/rdf-schema/
[44] http://xmlns.com/foaf/spec/
[45] https://elisasibarani.github.io/SARO/
[46] https://www.nltk.org/

The output of the process is a list of skills or occupations inferred from the initial resource, depending on the occasion. Our approach yielded adequate results in the context of an operational, proof-of-concept prototype implementation. There is still a lot of room for improvement; a process which we will be focused on in the near future. Our priorities revolve around improving the entity extraction process and implementing online operations, deprecating the need for local files.

3.2.4 RDF Graph

As stated before, the RDF Turtle Graph provided by ESCO and containing the Skills and Occupations classification was the basis of our general implementation. Due to the nature and structure of Linked Open Data and the Semantic Web, we decided that it would be optimal if we integrated the data extracted from the CVs, JVs and UoLs on the same graph, since they would be interconnected to the rest of the entities included. As of the latest iteration of our prototype, every resource (CV, JV, UoL) available to us was assigned a 'file' URI[47] in order for it to be transformed into a Semantic Web entity. Then, we consider each one of these URIs representative of a physical entity: CVs represent People and are interpreted in the graph as "User"; JVs represent Job Postings and are interpreted in the graph as "JobPosting"; UoLs represent Courses and are interpreted in the graph as 'Training Resource'. All the aforementioned concepts are provided by the SARO Ontology. Ultimately, each one of these entities in the graph are then connected to a name, their md5 hash of the original file that serves as a unique ID, the skills associated with them and, in the cases of JVs and UoLs, the title of the posting/course and their respective publishers. All the required entities and their respective relations were provided by the RDFS, FOAF and SARO ontologies.

3.2.5 Apache Jena Fuseki - SPARQL

As was stated in the first paragraph of this section, the final integrated graph requires a specific framework for it to be loaded and queried. In the initial iterations of our prototype, we utilized the Python library "RDFLib" to load the original graph, as provided by ESCO, and query it with SPARQL queries. However, RDFLib is design to load and handle small scale RDF graphs (i 10k triples) for development and testing purposes. Our graph, consisting of approximately 980k triples, needed a significant amount of time to be loaded, parsed and queried, thus hindering the development process. Moreover, having in mind that this solution wouldn't serve us in the deployment and production stage of the project, we quickly decided to set up an Apache Jena Fuseki Server, consisting of a Triples Database (TDB) and a SPARQL layer to handle queries. The benefits of this transition are also detailed in the first paragraph of this section. The Apache JTDB was set up on a dedicated server provided by My-Company Projects. Querying and updating the TDB is available in two ways: through a GUI provided by Apache Jena - serving mainly testing purposes - and via specific endpoints, established in certain ports of the server's IP, enabling automated, machine-scripted querying/updating.

3.2.6 Python

With all its components described to this point of this section, we proceed to describe the application developed to incorporate and support all the aforementioned. The app was developed in Python, mainly due to its capabilities of data handling and processing. The app primarily includes the annotators' implementation. Due to NLTK being native to Python, the app initially began as the annotators' submodule. Upon completion and end of operation, the annotators yield a set of skills related to the

[47] https://en.wikipedia.org/wiki/File URI scheme

original input (CV, JV, UoL). Handling this set of skills and correlating it to the ESCO API and Dataset was done via Python lists and Dataframes, as was described in the first paragraph of this section. The next part involved handling the ESCO graph and incorporating the aforementioned data to it. Since discontinuation of the RDFLib utilization and our transition into the Apache Jena TDB, we implemented an interface in Python for query and results handling, based on the SPARQLWrapper library. The results yielded by the queries were transformed into useful data through lists and dictionaries, and served as input to other operations. Moreover, in a couple of occasions where we needed more complex 'queries' (such as "Top 5 Occupations for a Candidate based on Skills in possession), the essential data for these procedures are yielded by the graph, while the calculations and operations are conducted in the Python back-end. The way that the rest ESCO-provided data and API is handled has already been described in the first paragraph of this section. Finally, we decided that the only way the TDB could be accessed would be through the Python app/back-end. While we had already implemented a way for the app and the TDB to interact and communicate, another one was needed in order for the app being accessible by its users. For this purpose, a RESTful API was implemented through the Flask Python library, providing 47 unique endpoints at this stage, and ways for more to be added. Documentation for the API could be provided on short notice.

3.3 Deployment

As a final step for our first stage of the development of the prototype, the API described in the previous paragraph was deployed on the same server that hosts the Apache Jena TDB. As it was developed in Python, its deployment was facilitated via the Gunicorn[48] framework/WSGI Server. Its availability is handled by a proxy established via NGINX[49] Web Server. This tech stack enables optimal workload distribution and handling, along with wider bandwidth for API requests.

[48] https://gunicorn.org/
[49] https://www.nginx.com/

4 Intelligent Agents

The aim of this project, as already mentioned, is to develop and offer a smart approach with innovative characteristics that will combine established technologies with symbolic Artificial Intelligence, including Intelligent Agents, in order to support of the less privileged. Hence, our intention was to focus on areas that have above the EU average percentage of less privileged. Southeastern Europe is such an area for a variety of reasons, including among other extensive economic crisis and social impact. In this context, the project will utilize an artificial Intelligence-based module as described in the following Projects Results 3. This section presents the basic notions of this domain, revealing its added value.

More specifically, the goal is to create a human-like artificial intelligence-enabled module without the need of supervision that will represent and bring together candidates and stakeholders along with agents' autonomy, reactiveness, and proactiveness. Intelligent Agents (IAs) are a well-studied technology that offers plenty of properties (Table 1), especially autonomy, reactivity, proactivity and communication ability. The notion of autonomy means that an agent exercises exclusive control over its own actions and state. Reactivity means sensing or perceiving change in their environment and responding while all agents have the ability to communicate with other entities, such as human users, other agents, or objects. Furthermore, agents have the ability to plan and set goals, to maintain beliefs, to reason about their actions and others, including humans, and learn from past experience. In the module, each agent has its own roles and responsibilities, yet all agents acting on the environment form a dynamic and meaningful community.

Table 1. Intelligent agents' properties.

Autonomy	Migration
Adaptability	Learning
Social ability (Collaboration/ Coordination/Interaction)	Reactivity
Persistence (execution)	Proactivity
Communication ability	Mobility

Although designing and building agents is not trivial, information systems based on intelligent agent technology are inevitable when strategic advantage, in cases such as innovating skill building, is needed. What is even more important for bridging artificial intelligence, skill matching and building, and various stakeholders is the ability of agents to think and, therefore, act like humans. In other words, agents have the ability to plan and set goals, to maintain beliefs, to reason about their actions and others, including humans, and learn from past experience and machine learning techniques. We can think of a multi-agent system as a virtual social community, which is the reflection of a real human community. Hence, agents equipped with rules and facts are goal-oriented and suitable for changing (dynamic, unpredictable, unreliable) environments. For instance, if they know that a specific programming language is top-rated by job providers, they will check all alternatives in order to find potential cases to propose, e.g. courses. The cases to be simulated will be composed by a set of sequences, events or tasks that may exist in the environment, which will be configurable through the module. These tasks will be interrelated, since they depend on the order or time in which the events

occur. The complexity lies in the fact that these models allow the simulation of changing and complex environments, so the use of agent technology is ideal.

Sometimes, each agent is equipped with a specific logic. For instance, defeasible logic (DL) that was introduced by Nute [1]. DL is a simple and efficient rule based nonmonotonic formalism that deals with incomplete and conflicting information. More specifically, DL has the notion of rules that can be defeated; hence it derives plausible conclusions from partial and sometimes conflicting information. These conclusions, despite being supported by the currently available information, could nonetheless be rejected in the light of new, or more refined, information. DL in contrast with traditional deductive logic, allows the addition of further propositions to make an existing belief false, making it non-monotonic, pretty much like the way human brain acts. Hence DL (being a non-monotonic logic) is capable of modelling the way intelligent agents (like humans), draw reasonable conclusions from inconclusive information, leading to more realistic conclusions and assessments similar to human reasoning. DL can be and it was adapted in a variety of applications such as decision making and negotiation cases [2]. For instance, DL was successfully applied for knowledge representation and reasoning about which task to perform next. It can be used even in order to provide a formalism for specifying authorization policies of a dynamic system. Furthermore, compared to more mainstream non-monotonic approaches, the main advantages of defeasible reasoning are enhanced representational capabilities and low computational complexity [3].

More specifically, a defeasible theory D consists of three basic parts; namely a set of facts (F), a set of rules (R) and a superiority relationship (>). Hence, D can be represented by the triple (F, R, >). Additionally, the set of rules R consists of three distinct types of rules; namely strict rules, defeasible rules and defeaters. Strict rules are denoted by $A \rightarrow p$ and are interpreted in the typical sense that whenever the premises are indisputable, so is the conclusion. An example of a strict rule could be "Python is language" which written formally is r_1: python(X) $\rightarrow$ language(X). On the other hand, defeasible rules are rules that can be defeated by contrary evidence and are represented as $A \Rightarrow p$. An example of a defeasible rule could be "Any language is considered to be useful" which written formally is r_2: language(X) $\Rightarrow$ useful(X). Finally, defeaters are rules that do not actively support conclusions, but can only prevent some of them. They are represented as $A \in p$. In other words, these rules can be used to defeat defeasible rules by producing evidence to the contrary. An example of such a rule could be "If a language exists, but its importance is under a threshold, then it might be not useful" which written formally is r_3: language(X), importance(X,Y), Y>threshold $\in \neg$useful(X). Hence, this defeater can defeat, for instance, rule r_4: language(X) $\Rightarrow$ useful(X).

As far as it concerns the superiority relationship among the rule set R, it is an acyclic relation > on R. An example of such a relationship could be the following: given the defeasible rules r_2 and r_4, no conclusive decision can be made about whether the language is useful or not since rules r_2 and r_4 contradict each other. Yet, if superiority relation > with $r_3 > r_4$ exists then r_3 overrides r_4 and the conclusion that the language is not useful can be derived. In this case rule r_3 is called superior to r_4 and r_4 inferior to r_3. Another important element of defeasible reasoning is the notion of conflicting literals, where literals are considered to be conflicting and at most one of a certain set should be derived. An example of such a case is a negotiation about the amount of pesticides that should be used, where a proposal should be made by the potential stakeholder. The proposal can be determined by several rules, whose conditions may or may not be mutually exclusive. All rules have proposal(X) in their head, since a proposal is usually a positive literal. However, only one offer should be made. Therefore, only

one of the rules should prevail, based on superiority relations among them. In this case, the conflict set is: $C(proposal(x,y)) = \{\neg\ proposal\ (x,y)\} \cup \{\ proposal\ (x,z) \mid z \neq y\}$.

Of course, intelligent agents are part of the Semantic Web, hence they can be combined with all aforementioned resources. More research will be conducted during Project Results 3 where this technology will be integrated to the proposed system.

5 Conclusions and recommendations

In this phase, we reviewed each recommended resource in respect to its capabilities to aid our project in its operational goals. Amongst the initial recommendations, there are feasible solutions that can support possible implementations of features to our project in several scales. Moreover, we would like to further propose the examination of the websites indeed.com and kariera.gr: two Greece-based major job-searching websites that include international job vacancies. Although they virtually offer nothing more to the table than LinkedIn, they still constitute solid alternatives for supplementary data, if needed. Furthermore, we examined a range of technical resources and proceeded from considered resources to the utilized ones, such as the ESCO Framework and the Semantic Web Ontologies. As a final step for this first stage of the development of the prototype, the API described in the previous paragraph was deployed on the same server that hosts the Apache Jena TDB. Furthermore, a first review on intelligent agents and their usability was conducted, although they will be integrated to the system during the Project Results 3 phase.

References

1 Nute, D.: Defeasible Logic. Springer-Verlag: Web Knowledge Management and Decision Support, 14th Int. Conf. on Applications of Prolog, 151-169 (2003).

2 Kravari K., Bassiliades N.: StoRM: A social agent-based trust model for the internet of things adopting microservice architecture. SIMPAT Journal 94: 286-302 (2019).

3 Maher, M.J. Propositional defeasible logic has linear complexity. Theory and Practice of Logic Programming 1(6):691–711 (2001).

Online sources:

https://www.udemy.com/developers/affiliate/

https://uit.stanford.edu/developers/apis

https://cop.stanford.edu/community/api-developers

https://uit.stanford.edu/service/registry/certificates#panel2e

https://mailman.stanford.edu/mailman/listinfo/api_developers

https://ist.mit.edu/apis

https://developers.yale.edu/

https://courses.edx.org/api-docs/?format=openapi

https://docs.microsoft.com/en-us/learn/support/catalog-api#catalog-api-program

https://docs.microsoft.com/en-us/linkedin/learning/overview/

https://www.glassdoor.com/developer/index.htm

https://github.com/navikt/pam-eures-stilling-eksport

https://esco.ec.europa.eu/en/about-esco

https://esco.ec.europa.eu/en/use-esco/use-esco-services-api

https://esco.ec.europa.eu/en/use-esco/use-esco-services-api/esco-local-api

https://esco.ec.europa.eu/en/use-esco/use-esco-services-api/esco-web-service-api

https://en.wikipedia.org/wiki/Turtle (syntax)

https://ec.europa.eu/esco/api/doc/esco api doc.html

Retrieved from https://www.w3.org/RDF/

https://www.python.org/

https://docs.python.org/3/tutorial/datastructures.html#more-on-lists

https://docs.python.org/3/tutorial/datastructures.html#dictionaries

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html

https://pandas.pydata.org/

https://numpy.org/doc/stable/reference/arrays.html

https://numpy.org/doc/stable/index.html

https://www.w3.org/2001/sw/wiki/Apache Jena

https://www.w3.org/TR/rdf-sparql-query/

https://jena.apache.org/documentation/tdb/index.html

https://jena.apache.org/documentation/fuseki2/
https://rdflib.readthedocs.io/en/stable/
https://www.w3.org/TR/rdf-concepts/
https://www.w3.org/XML/Schema
https://schemas.liquid-technologies.com/hr-xml/2007-04-15/
https://europa.eu/europass/en
https://europa.eu/europass/system/files/2020-07/europass-xml-schema-docv3.4.00.pdf
https://en.wikipedia.org/wiki/Natural language processing
https://en.wikipedia.org/wiki/Natural-language understanding
ttps://en.wikipedia.org/wiki/Named-entity recognition
https://ec.europa.eu/esco/lod/static/model.html
https://www.w3.org/TR/rdf-concepts/
https://www.w3.org/TR/rdf-schema/
http://xmlns.com/foaf/spec/
https://elisasibarani.github.io/SARO/
https://www.nltk.org/
https://en.wikipedia.org/wiki/File URI scheme
https://gunicorn.org/
https://www.nginx.com/