Scientific knowledge base creation from heterogeneous data

Final Year Project

2018-2022

A project submitted in partial fulfillment of the degree of

BS in Computer Science



Submitted to

Mr. Shoaib Malik

Department of Computer Science

Faculty of Computing & Artificial Intelligence (FCAI)

Air University, Islamabad

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type (Nature of project) | | | [  ] Development [  ] Research | | |
| Area of specialization | | | [ ] WebApp [ ] Mobile App  [  ] AI based [ ] Embedded System | | |
| FYP ID | | |  | | |
| **Project Group Members** | | | | | |
| Sr.# | Reg. # | Student Name | | Email ID | \*Signature |
| (i) | Hamza Ali | 181058 | | 181058@students.au.edu.pk |  |
| (ii) | Muhammad Huzaifa | 181122 | | 181122@students.au.edu.pk |  |
|  |  |  | |  |  |

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

# Plagiarism Certificate

This is to certify that, I **Hamza Ali** Son of **Saif Ali**, group leader of FYP under registration no **181058** at Computer Sciences Department, Air University. I declare that my FYP report is checked by my supervisor.

Date: May 14,2022 Name of Group Leader: Hamza Ali Signature:\_\_\_\_\_\_\_\_

Name of Supervisor: Mr. Muhammad Qasim Riaz Co-Supervisor: (Not Allocated)

Designation: Lecturer Designation: (N/A)

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HoD: Dr. Awais Ahmad

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Scientific Knowledge Base Creation from heterogeneous data**

**Change Record**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author(s)** | **Version** | **Date** | **Notes** | **Supervisor’s Signature** |
| Hamza Ali  Muhammad Huzaifa | 1.0 |  | <Original Draft> |  |
| Hamza Ali  Muhammad Huzaifa | 2.0 |  | Amendment with the document as per supervisor in FYP-2 |  |
| Hamza Ali  Muhammad Huzaifa | 3.0 |  | Edited the documents by adding progress in FYP-3 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**APPROVAL**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Supervisor** | | |  |  |
| Name: Mr. Shoaib Malik | |  | | |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| **Project Manager** | | | |  |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |  |

|  |  |
| --- | --- |
| **Chair Department** | |
|  |  |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# Dedication

I dedicate my dissertation to my family, friends, and my teachers. A special thanks to my parents who acknowledged my each decision, and helped me in achieving what I am today.

Also, I want to dedicate my work to my friends, that gave unique ideas throughout the university. I appreciate what they all have done for me when troubled. Thanks to Huzaifa, I’ve managed to polish my programming and documentation skill.

I dedicate this work to my teachers that were always available when I had any trouble with my studies, and they responded positively.

# Acknowledgments

I am thankful for my teachers that gave me an opportunity to work on this project. I would be expressing my gratitude to Sir Shoaib for helping us out with this project as a supervisor. I must be thankful to my parents for what I am now, because of him. Without their support, I couldn’t complete the project on my own.

Abstract

Solving a user’s query has been tough since the use of internet, as internet is the source of information and spams. For getting solution, one relies on the google but sometimes it gives false information via spamming tags. For that, one needs to have a search engine that would yields solution from authentic sites and scholarly articles. For that, we would be using web crawling, NLP and numerous data processing python tools for giving authentic solutions for students, teachers and any other tech seekers. Our web-app would simply input search query along with certain categories like Linux, Linux → server (sub category). After the input field, user’s queries would be entertained. In some cases, if the tools fail to gather relevant data, that queries can either be posted to our forums, which would be held till certain time, and after then (unsolved) it will be posted to stack overflow. This process will give out authentic solutions to user’s queries.

# Table of Contents

[Scientific knowledge base creation from heterogeneous data i](#__RefHeading___Toc5128_2154508051)

[Final Year Project i](#__RefHeading___Toc8030_3784747029)

[2018-2022 i](#__RefHeading___Toc8032_3784747029)

[BS in Computer Science i](#__RefHeading___Toc8034_3784747029)

[Plagiarism Certificate iii](#__RefHeading___Toc8038_3784747029)

[Dedication vi](#__RefHeading___Toc8040_3784747029)

[Acknowledgments vii](#__RefHeading___Toc8042_3784747029)

[Table of Contents ix](#__RefHeading___Toc8044_3784747029)

[List of Figures xii](#__RefHeading___Toc8046_3784747029)

[List of Tables xiii](#__RefHeading___Toc8048_3784747029)

[List of Abbreviations xiv](#__RefHeading___Toc8050_3784747029)

[Chapter 1 15](#__RefHeading___Toc8052_3784747029)

[Introduction & Background 15](#__RefHeading___Toc8054_3784747029)

[1.1. Background 16](#__RefHeading___Toc8056_3784747029)

[1.2. Motivations and Challenges 16](#__RefHeading___Toc8058_3784747029)

[1.3. Goals and Objectives 17](#__RefHeading___Toc8060_3784747029)

[1.4. Project Plan 17](#__RefHeading___Toc8062_3784747029)

[1.4.1. Work Breakdown Structure 17](#__RefHeading___Toc8064_3784747029)

[1.4.2. Roles & Responsibility Matrix 17](#__RefHeading___Toc8066_3784747029)

[1.4.3. Gantt Chart 18](#__RefHeading___Toc8068_3784747029)

[1.5. Report Outline 18](#__RefHeading___Toc8070_3784747029)

[Chapter 2 20](#__RefHeading___Toc8072_3784747029)

[Software Requirement Specifications 20](#__RefHeading___Toc8074_3784747029)

[2.1. Introduction 21](#__RefHeading___Toc8076_3784747029)

[2.1.1. Purpose 21](#__RefHeading___Toc8078_3784747029)

[2.1.2. Document Conventions 21](#__RefHeading___Toc8080_3784747029)

[2.1.3. Intended Audience and Reading Suggestions 21](#__RefHeading___Toc8082_3784747029)

[2.1.4. Product Scope 21](#__RefHeading___Toc8084_3784747029)

[2.2. Overall Description 21](#__RefHeading___Toc8088_3784747029)

[2.2.1. Product Perspective 21](#__RefHeading___Toc8090_3784747029)

[2.2.2. User Classes and Characteristics 22](#__RefHeading___Toc8092_3784747029)

[2.2.3. Operating Environment 22](#__RefHeading___Toc8094_3784747029)

[2.2.4. Design and Implementation Constraints 22](#__RefHeading___Toc8096_3784747029)

[2.3. External Interface Requirement 22](#__RefHeading___Toc8098_3784747029)

[2.3.1. User Interfaces 22](#__RefHeading___Toc8100_3784747029)

[2.3.2. Hardware Interfaces 22](#__RefHeading___Toc8102_3784747029)

[2.3.3. Communications Interfaces 23](#__RefHeading___Toc8104_3784747029)

[2.4. System Features 23](#__RefHeading___Toc8106_3784747029)

[2.4.1. System Feature 1 23](#__RefHeading___Toc8108_3784747029)

[2.4.1.1. Description 23](#__RefHeading___Toc8110_3784747029)

[2.4.1.2. Stimulus/Response Sequences 23](#__RefHeading___Toc8112_3784747029)

[2.4.1.3. Functional Requirements 23](#__RefHeading___Toc8114_3784747029)

[2.4.2. System Feature 2 23](#__RefHeading___Toc8116_3784747029)

[2.4.2.1. Description and Priority 23](#__RefHeading___Toc8118_3784747029)

[2.4.2.2. Stimulus/Response Sequences 23](#__RefHeading___Toc8120_3784747029)

[2.4.2.3. Functional Requirements 23](#__RefHeading___Toc8122_3784747029)

[2.5. Nonfunctional Requirements 24](#__RefHeading___Toc8124_3784747029)

[2.5.1. Security Requirements 24](#__RefHeading___Toc8126_3784747029)

[2.5.2. Usability Requirements 24](#__RefHeading___Toc8276_3784747029)

[2.5.3. Reliability Requirements 24](#__RefHeading___Toc8140_3784747029)

[2.5.4. Maintainability/Supportability Requirements 24](#__RefHeading___Toc8148_3784747029)

[2.5.5. Portability Requirements 24](#__RefHeading___Toc8150_3784747029)

[2.5.6. Efficiency Requirements 24](#__RefHeading___Toc8154_3784747029)

[Chapter 3 25](#__RefHeading___Toc8162_3784747029)

[Literature Review 25](#__RefHeading___Toc8164_3784747029)

[1.1. Introduction 26](#__RefHeading___Toc8166_3784747029)

[1.1. Why this project 26](#__RefHeading___Toc8168_3784747029)

[1.2. Working 26](#__RefHeading___Toc8170_3784747029)

[1.2.1 Web Scraping 26](#__RefHeading___Toc8172_3784747029)

[1.2.2 Getting entity pairs 27](#__RefHeading___Toc8174_3784747029)

[1.2.3 Creating knowledge graph 27](#__RefHeading___Toc8176_3784747029)

[Chapter 4 1](#__RefHeading___Toc8178_3784747029)

[Use Case Analysis 1](#__RefHeading___Toc8180_3784747029)

[4.1. Use Case Model 2](#__RefHeading___Toc8182_3784747029)

[3.1. Flow Diagram 3](#__RefHeading___Toc8184_3784747029)

[3.1. Activity Diagram 4](#__RefHeading___Toc8186_3784747029)

[3.2. Use Cases Description 5](#__RefHeading___Toc8188_3784747029)

[3.2.1. Answer a query 5](#__RefHeading___Toc8190_3784747029)

[3.2.2. Post a question 5](#__RefHeading___Toc8192_3784747029)

[Chapter 5 6](#__RefHeading___Toc8194_3784747029)

[Proposed Approach 6](#__RefHeading___Toc8196_3784747029)

[7](#__RefHeading___Toc8198_3784747029)

[5.1. Dataset Exploration 7](#__RefHeading___Toc8200_3784747029)

[Chapter 6 8](#__RefHeading___Toc8202_3784747029)

[Implementation & Results 8](#__RefHeading___Toc8204_3784747029)

[6.1 Different components of your proposed approach 9](#__RefHeading___Toc8206_3784747029)

[Data Set 9](#__RefHeading___Toc8208_3784747029)

[6.2 Implementation of proposed approach 9](#__RefHeading___Toc8210_3784747029)

[6.3 Deployment Environment 10](#__RefHeading___Toc8212_3784747029)

[6.1 Tools and Techniques 10](#__RefHeading___Toc8214_3784747029)

[6.2 Best Practices / Coding Standards 10](#__RefHeading___Toc8224_3784747029)

[6.3 Version Control 11](#__RefHeading___Toc8226_3784747029)

[Chapter 7 12](#__RefHeading___Toc8228_3784747029)

[Business Plan 12](#__RefHeading___Toc8230_3784747029)

[Chapter 7: Business Plan 13](#__RefHeading___Toc8232_3784747029)

[7.4 Business Description 13](#__RefHeading___Toc8234_3784747029)

[7.5 Market Analysis & Strategy 13](#__RefHeading___Toc8236_3784747029)

[7.6 Products/Services Description 13](#__RefHeading___Toc8238_3784747029)

[7.7 SWOT Analysis 14](#__RefHeading___Toc8240_3784747029)

[Chapter 8 15](#__RefHeading___Toc8242_3784747029)

[Conclusion & Future Work 15](#__RefHeading___Toc8244_3784747029)

[Chapter 8: Conclusion & Future Work 16](#__RefHeading___Toc8246_3784747029)

[8.3 Achievements and Improvements 16](#__RefHeading___Toc8248_3784747029)

[8.4 Future Enhancements/Recommendations 16](#__RefHeading___Toc8250_3784747029)

[Chapter 9 17](#__RefHeading___Toc8252_3784747029)

[Paper Writing 17](#__RefHeading___Toc8254_3784747029)

[9.1. Paper Writing 18](#__RefHeading___Toc8256_3784747029)

[1. Introduction 18](#__RefHeading___Toc8258_3784747029)

[2. Background 18](#__RefHeading___Toc8260_3784747029)

[3. Key Points 18](#__RefHeading___Toc8262_3784747029)

[1. *Abbreviation* 18](#__RefHeading___Toc8264_3784747029)

[2. *Glossary* 19](#__RefHeading___Toc8266_3784747029)

[4. goals and objective 19](#__RefHeading___Toc8268_3784747029)

[Appendices 21](#__RefHeading___Toc8270_3784747029)

[Appendix A: Information / Promotional Material 22](#__RefHeading___Toc8272_3784747029)

[References 23](#__RefHeading___Toc8274_3784747029)

# List of Figures

Table of Figures

[Figure 1: Gantt Chart 18](#Figure!0|sequence)

[Figure 2: Use Case Model 2](#Figure!1|sequence)

[Figure 3: Flow Diagram 3](#Figure!2|sequence)

[Figure 4: Activity Diagram 4](#Figure!3|sequence)

[Figure 5: Proposed Approach 7](#Figure!4|sequence)

[Figure 6: Output 10](#Figure!5|sequence)

[Figure 7: SWOT Analysis 14](#Figure!6|sequence)

[Figure 8: FYP Poster 22](#Figure!7|sequence)

# List of Tables

Index of Tables

[Table 1: Use Case: Answer a Query 5](#Table!0|sequence)

[Table 2: Use Case: Post a Question 5](#Table!1|sequence)

# List of Abbreviations

1.1 UML Unified Model Notation

1.2 SRE Software Requirement Engineering

2.1 SDR Software Defined Radios

3.1 KB Knowledge-base

3.2 NLP Natural language processing

3.3 UI User Interface

# Chapter 1

# Introduction & Background

Chapter 1: Introduction

Google has become the best query solver for its reputable solving algorithm, but many sites owners uses fake tags for ranking up, due to which user can’t ensure the solution to be generic. Therefore, they rely on reputed authentic sites like stack overflow, Quora and many more, but the problem is that many tech seeker don’t know where to start from. For solving their problem, we are creating a web app that would aid them in solving their queries. Our app would rely on major authentic sites and academic datasets for crawling answers, and would be shown if found, else it’ll be sent to our forums / stack overflow.

## Background

Techseeker (seniors) uses only reputed sites for their solution, be it the new technology or some help with the research, but for our juniors, they need certain guidance to distinguish between the reputed sites and unauthentic sites. For that, we are developing a knowledge base (submodule of a app) for scientific documents, which would include differnet tech terminologies, from AI, BigData, and Linux to certain small queries. A knowledge base is a self-serve customer service library that includes information about a product, service, or some specific topic. A knowledge base allows you to create self-service customer support content around recurring topics, issues, and themes. A knowledge base doesn’t simply have to be a reservoir of help center articles. It can also include functions like a frequently-asked questions (FAQ) section, a user forum, articles, white papers, how-to articles, video tutorials, case studies, and dictionaries or glossaries — essentially, anything that helps customers understand and use your products or services.

## Motivations and Challenges

Every newbie in the tech world needs help to take certain direction for the career, they need to know what will yield useful data from the rest. They would be needing a knowledge base that can extract the required information from authentic datasets and provide them the soltuion in easy UI for everyone. We would be extracting data from Wikipedia and academic datasets using Wikipedia API and devlepod the relation among the text where nodes represent entities and edges represent relationship among them. Grammatical patterns were extracted from the given text using NLP library SpaCy, and neuralcoref. After developed pattern we developed knowledge graph.

## Goals and Objectives

By taking input and categories from the user, our goals will be:

* Extracting the required information for specific topic from available datasets,
* Extraction of entity pairs from grammatical patterns from text using NLP library SpaCy,
* Developing relationships among text,
* Knowledge graph formation

## Project Plan

Our project is divided into 3 components, each being dependent on the others.

* FYP – 1
* FYP – 2
* FYP – 3

At each phase, we have set certain objectives or challenges to overcome in our process. Our FYP-1 was to gather required research for the project, while FYP-2 and FYP-3 revolves around the documentation and the practical part, where we programmed to obtain results from input queries via Wikipedia scraping and then from provided datasets. Also, our project was solely research and development based, so we didn’t invest a penny for our project, as majority of the tools required are open source. For the Gantt chart, visit [section](#_toc545) [1.7](#_toc13).

## Work Breakdown Structure

**PHASE 1:** Project Idea finalized + documentation

**PHASE 2:** Coding

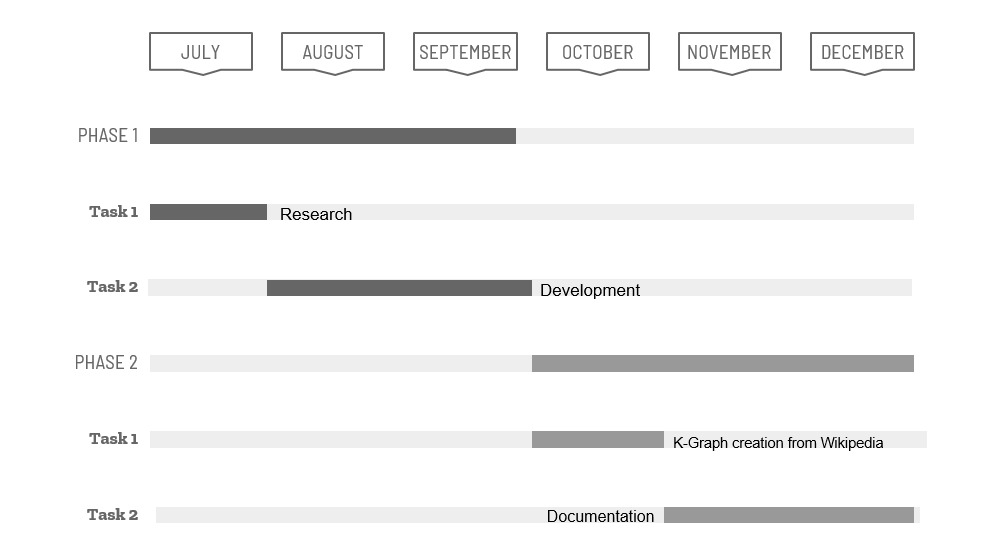
**PHASE 3:** Knowledgegraph creation from wikipedia

**PHASE 4:** Knowledgegraph creation from given Datasets

## Roles & Responsibility Matrix

|  |  |  |
| --- | --- | --- |
| Responsibility | (R1) Hamza Ali | (R2) M Huzaifa |
| Research | ✔ | ✔ |
| Coding | ✔ |  |
| Understanding | ✔ | ✔ |
| Implementation | ✔ |  |
| Documentation |  | ✔ |
| Visual representation |  | ✔ |

## Gantt Chart

Figure 1: Gantt Chart

## Report Outline

Our thesis report will be including each in-depth details of following topics.

* Literature Review
* SRS (Software Requirement Specifications) document
* Use Case Analysis
* Our Approach (proposed)
* Business Plan

# Chapter 2

# Software Requirement Specifications

**Chapter 2:** Software Requirement Specifications



## Introduction

## Purpose

Purpose is to develop a knowledge graph that can entertain the query given to it on the basis of provided datasets. It will work by developing relationship and forming entity pairs among them. If the query can’t be resolved from provided documents, then experts will answer that particular query.

## Document Conventions

The main chapter headings are labeled as Chapter no. and chapter name with font size 20 **BOLD** and font style Time New Romans. The headings inside the chapter are **BOLD** with font size 16 and labeled as 1.1, 1.2, 1.3, and the subheadings among them are labeled as 1.1.1, 1.1.2 **BOLD** and so on.

## Intended Audience and Reading Suggestions

This report is intended for developers, project managers, users, testers, and writers. Chapter defines basic overview of the project by reading chapter 1 users can study about basic functionality etc. Chapter 3 describes purpose, audience type and what will be the scope of the project. Chapter 4 describes functionality. Chapter 5 defines flow of project, diagrams to visualize the project so that developers, testers can understand the flow of project. Chapter 6 defines business model.

## Product Scope

Knowledge base will be part of platform which will respond to user query on the basis of dependencies and relations. It can be any forms like FAQs section, suggestion on search etc. The benefit of knowledge base will be that user will get answer to his solution on the spot without waiting for response from the expert. If query is unable to get resolved by knowledge base itself then expert will answer.

## Overall Description

## Product Perspective

Our goal was to scrape the data in the internet. We have used wikipedia-api for this task. For that, we have to submit the topic name and let it do its job. For this, we have used the given libraries.

***Pandas*** *is used to analyze and manipulate data.*

***Spacy*** *for using processing natural language, and making computer understand  words*

***NLTK*** *for pre-processing/ clearing unnecessary data from the dataset.*

***matplotlib*** *for plotting the relevant graph by using subject, object and their relations* 

## User Classes and Characteristics

**Getting entity pairs**

With the help of libraries: Spacy, neuralcoref we developed relations among data and finally got entity pairs.

**Knowledge graph creation**

After extraction of entity pairs the function nx.draw\_networkx draws the graph with Matplotlib with options for node positions, labeling, titles, and many other drawing features. The network\_edge\_label creates a connection between vertices and then we show it up.

## Operating Environment

It can be integrated to any software or website, but we are currently developing it for a social media platform. It does not require any special system requirements.

## Design and Implementation Constraints

Only owners can update/remove datasets on the basis of which query will be resolved. The users can only access the service. If they didn’t get the answer then they can post their question which will be answered by expert.

## External Interface Requirement

## User Interfaces

User interface will include a search bar where user will put his query and there will be several other options like selecting a category and further sub category to get more accurate answer. After search answers will be shown to the user. There will be another option to post a question for which you need an expert answer.

## Hardware Interfaces

Knowledge base does not require any special hardware to run. It utilizes a smaller amount of memory to provide you results. Initially, we are running it on our systems where our system is operating as a server to answer the user queries.

## Communications Interfaces

When the user question is answered, the user will be notified. If his query is not solved, then he’ll be notified via email when solved by the experts.

## System Features

We are using academic datasets for this project

## System Feature 1

Academic datasets

## Description

Datasets contain all the data from where query is to be resolved. The query will be entertained on the basis of relations among the data stored in datasets.

## Stimulus/Response Sequences

The user asks for some solution. It will be answered after searching from datasets. If the answer to the question is found it will be shown to the user.

## Functional Requirements

The functional requirements associated with this feature will be:

* Must be authentic
* Must be sorted

## System Feature 2

Knowledge graph creation

## Description and Priority

From the given datasets relations among the data are formed with interlinks and are stored as a graph.

## Stimulus/Response Sequences

The sentences are broken down on the basis of relations present in between them and finding relations among them.

## Functional Requirements

The functional requirements associated with this feature will be:

* Data must be in the form of sentences
* Data must be cleaned

## Nonfunctional Requirements

## Security Requirements

Datasets should not contain personal details of any individual. It shouldn’t reveal any personal detail of the organization to its users because it can may cause private information disclose to the public.

## Usability Requirements

A knowledge base must have following usability factors:

* Effective
* Error tolerant
* Efficient
* Must be maintainable

## Reliability Requirements

We can say the Knowledge base is reliable if:

* It must respond to user query
* The knowledge graph formed is correct

## Maintainability/Supportability Requirements

Maintainability defines the time required for a solution or its component to be fixed, changed to increase performance, speedup or other qualities, or adapted to a changing environment. It will be maintainable if it is adaptable to above mentioned features.

## Portability Requirements

Portability requirements include:

* Operating systems and their versions,
* Network specifics,
* Browsers and their versions, and
* Devices and other hardware requirements.

## Efficiency Requirements

We can say the Knowledge base is reliable and efficient if:

* It must respond to user query
* The knowledge graph formed must be accurate

# Chapter 3

# Literature Review

**Chapter 3:** Literature Review

1. 1. Introduction

Getting authentic information has been quite difficult now-a-days due to large number of scams/ fake meta tags. The huge internet is useful if one knows how to use it for themselves. For a developer/ tech seeker, one would always consider relevant and up to date solutions for their queries, therefore they depend on getting that without wasting time in browsing through each links. Similarly, for newbies, they would get many links for their queries, each with relevant or fake meta tags, but they can’t distinguish between the both. In both cases, they want a platform that can provide authentic and up to date solutions to their problems. They just need to provide the queries and relevant categories to our server, and let the server handles the rest, which would crawls the web and academic datasets for the solutions. Provided the case that if the solution could not be found, then the queries (with unique tokens) would be posted to our forums where our experts will answer it for them.

* 1. Why this project

There are many sites that provide same functionality, but newbies don’t know about them. Some are (of close resemblance) open-KB that don’t have enough datasets. Stack-overflow, Quora, and many more. Many tech seekers need quick solution within a click away, that’s what we are trying to achieve in our project. We would be using help of large academic datasets with recent articles, and the internet, if solution is not provided in the datasets. In worst case, it’ll be sent to our forum for our experts to solve.

* 1. Working

We are utilizing the help of Wikipedia that is scraping each links of topic to find relevant data, and printing it out, along with different entities that it found with their respective knowledge graph. Our project is divided into 3 parts.

* + 1. Web Scraping

Our main goal was to scrape the data in the internet. We have used provided datasets for this task. For that, we have to submit the topic name and let it do its job. For this, we have used the given libraries.

***Pandas***

Pandas is used for data analysis and its manipulation. In our code, we have used it for tabular data of each list that has to be printed. Our list consists of all pages with relevant text, categories and topic. This list will be added to source that can be printed later in the code.

* + 1. Getting entity pairs

For this task, we have selected the first page of our topic, that is most relevant, As spacy and neuralcoref only works well with spacy 2.1.0, neuralcoref 4.0, Python 3.7.0, so we have to install it as per requirements, in order for it to work. Also the downside of spacy 2.1.0, we have to add filter spans function that only works with spacy above 2.1.4. After all requirements, we would create pairs for first page, let's see how it works. To begin with, we would use regex to replace next line and numbers, as we don't need it for the nlp.

Then we would use neuralcoref to replace each pronoun with the main person (resolving co-reference clusters), after that, we would create unwanted tokens that would include particle, determiners, stop words. Then, we would resolve each entity if it's not in unwanted token or stop words. We may confirm each entity along with its type after the refinement part. After refinement, the program would tokenize each entity, and at the last, we would use pandas Data frame for printing the data accordingly.

* + 1. Creating knowledge graph

By using NLP library SpaCy, we extracted entity pairs from large amount of text. The spaCy library comes particularly handy when dealing with big amounts of text. It may be used to create data extraction and natural language understanding systems, as well as to pre-process text for deep learning.

The dependencies can be mapped in a directed graph representation:

* Words are the nodes.
* The grammatical relationships are the edges.

The get\_entity\_pairs function defines entity pairs as entities/noun chunks with subject — object dependencies connected by a root verb “**from\_pandas\_edgelist (df, source='source', target='target', edge\_attr=None, create\_using=None, edge\_key=None)**. This function returns a graph from Pandas DataFrame containing an edge list. The Pandas DataFrame contain at least two columns of node names and zero or more columns of edge attributes. **layout = nx.spring\_layout (k\_graph, k=0.15, iterations=20)** This function creates a graph where k controls the distance between the nodes and iterations is the number of times simulated annealing is run. If we don’t use this function then the nodes will be overlapping each other which would be challenging to understand. The next function nx.draw\_networkx draws the graph with Matplotlib with options for node positions, labeling, titles, and many other drawing features means we can set the property of each element according to our choice. The network\_edge\_label creates a connection between vertices and then we show it up.

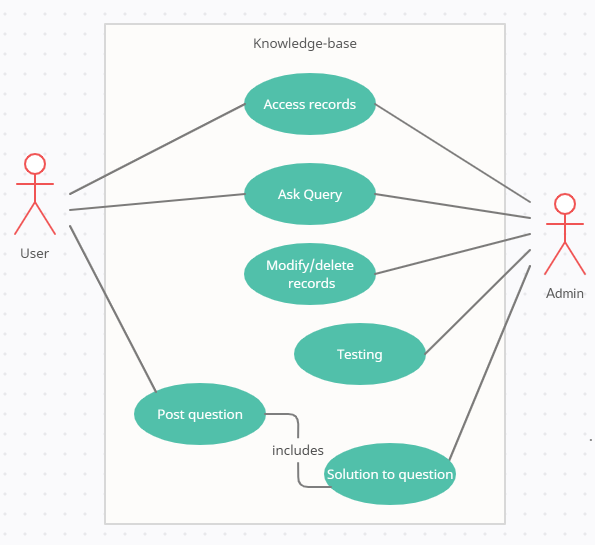
# Chapter 4

# Use Case Analysis

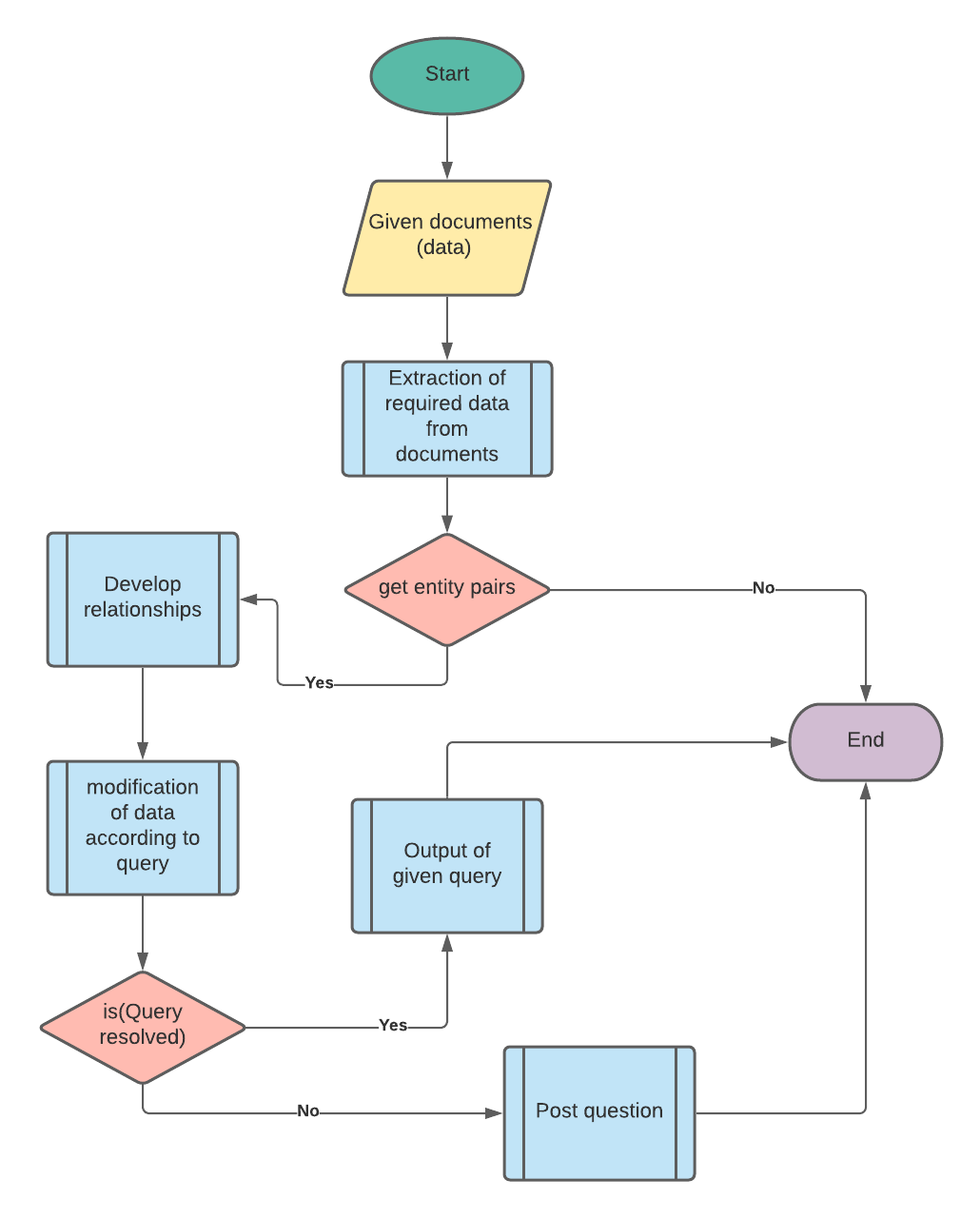
**Chapter 4:** Use Case Analysis



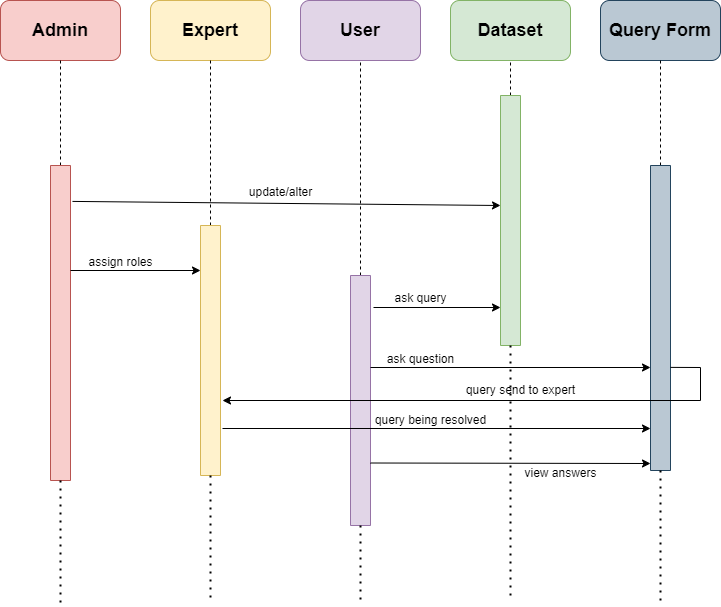
## Use Case Model

Figure 2: Use Case Model

## Flow Diagram

Figure 3: Flow Diagram

## Activity Diagram

Figure 4: Activity Diagram

## Use Cases Description

* + 1. Answer a query

|  |  |
| --- | --- |
| **Title** | Answer a query |
| **Requirement** | To answer a user query |
| **Rational** | Get answer |
| **Restriction or Risk** | Checking for answer |
| **Dependency** | Pc, datasets, internet |
| **Priority** | Safety, timing |
| **Output** | Answer to the query |

Table 1: Use Case: Answer a Query

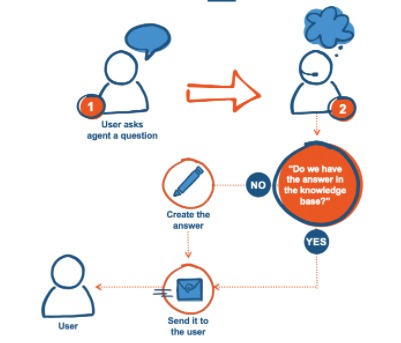
* + 1. Post a question

|  |  |
| --- | --- |
| **Title** | Post a question |
| **Requirement** | To get answer a question from expert |
| **Rational** | Get answer |
| **Restriction or Risk** | May be time taking |
| **Dependency** | Knowledge of expert |
| **Priority** | timing |
| **Output** | Answer to the given question |

Table 2: Use Case: Post a Question

# Chapter 5

# Proposed Approach

Figure 5: Proposed Approach



## Dataset Exploration

In In this step, we explored our data-set and found the following limitations:

* Tackling of large datasets
* Web Crawling (request approval)
* Faster execution with higher accuracy

**Data Augmentation and Pre-Processing**

In this step, we solved issues in our dataset using data augmentation and different data pre-processing techniques.

# Chapter 6

# Implementation & Results

**Chapter 6:** Implementation & Results

1. Different components of your proposed approach

Data Set

In this step, we explored our data-set and found the following limitations:

* Tackling of large datasets
* Web Crawling (request approval)
* Removing the unwanted data
* Faster execution with higher accuracy

**Getting entity pairs**

With the help of libraries: Spacy, neuralcoref we developed relations among data and finally got entity pairs

**Knowledge graph creation**

After extraction of entity pairs the function nx.draw\_networkx draws the graph with Matplotlib with options for node positions, labeling, titles, and many other drawing features. The network\_edge\_label creates a connection between vertices and then we show it up

1. Implementation of proposed approach
   1. **Data cleaning**

First, we need to clean the data. For that, we would

1. Create token and replace unnecessary punctuations with null
2. Eradicate stopwords tokens from the rest.
   1. **Finding entities**

After data cleaning, we would get the different entities from cleaned data. Here, we would getting the subject and the object

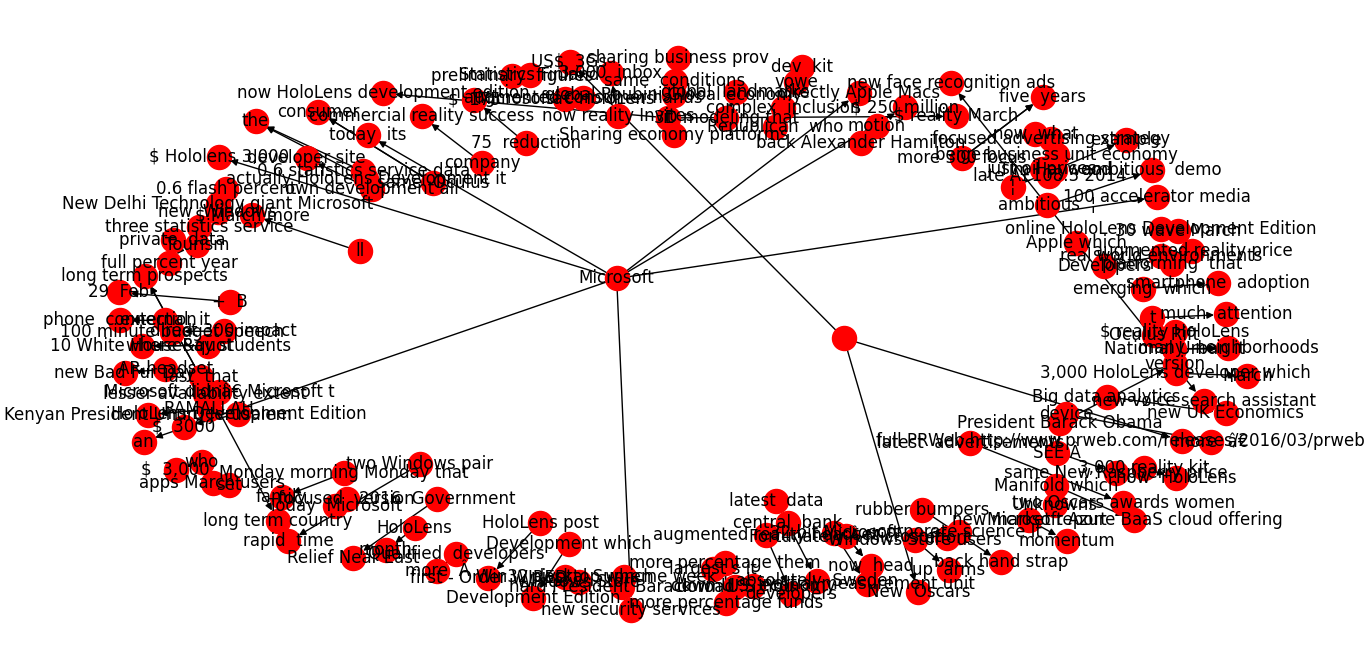
* 1. **Finding relations**

After finding the entities successfully, we would need to find the relation between those entities. For that. we would use matcher class, that would match the vocabulary and find the possible matches for those entities.

* 1. **Create the Knowledge Graph**

Next, we would be plotting the graph. We would need to create a list of source and target from entities, and edge from relation. For plotting, we would create a dataframe (using pandas library) of source and target.

* 1. **Output**

Figure 6: Output

## Deployment Environment

1. Operating system: Pop OS GNU Linux
2. System: HP EliteBook Revolve 810 G3 A30
3. Processor i5, 5th gen
4. Ram: 8 GB
5. Software used: Jupyter Notebook integrated in Vs Code

## Tools and Techniques

We coded this project on Vs Code with Python 3.7.0 and used following libraries

* Neuralcoref 4.0
* spacy 2.1.0
* networkx

## Best Practices / Coding Standards

We added comments on each portion of code so that the others can easily understand it.

## Version Control

Our main focus is to gather relevant data from the datasets that would be used to plot the knowledge graphs and then apply queries as per user’s request. Implementation of the knowledge graph is completed but the queries.

* Data cleaning
* Finding entities
* Finding relations
* Create the Knowledge Graph

# Chapter 7

# Business Plan

**Chapter 7:** Business Plan

This project aims to provide academic knowledge to not only student, teach but all tech seekers along the globe. It would benefit every individual that wants to learn about new technology or any programming language. In short, if one wants to learn about latest tech, or solving errors in their code, they can just switch to our web-app and find their solution right away.

2. 2. Business Description

As for the business, we are aiming it to be an **open-source** where each experts in the tech-world can integrate with the code and make it bug free and more efficient.

## Market Analysis & Strategy

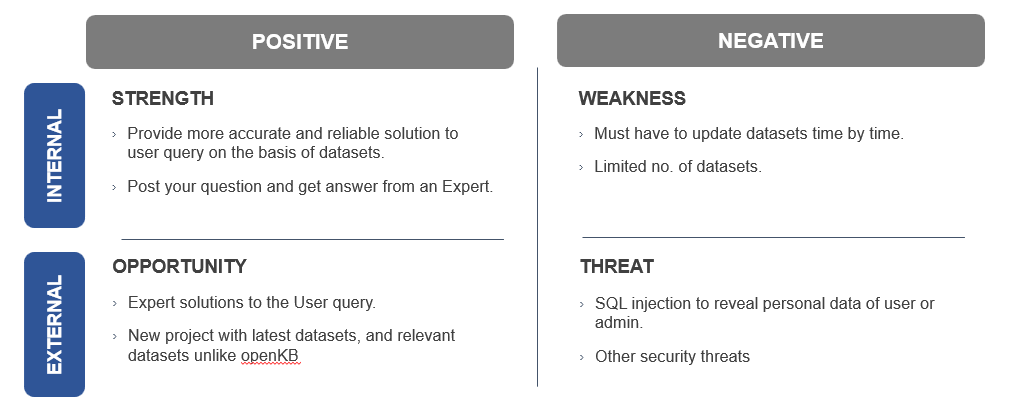
There are many sites as described that proficient developers are using for their day-to-day work, but for starters, they need to research a lot to find trust-worthy sites. For that, we need to help them in finding solutions to their queries faster and with easy access. Other developers can contribute to the project to enhance functioning of the project.

## Products/Services Description

We are aiming to provide following services,

* Giving solution to user’s queries
* If found, user is notified with the solution
* If not found, user will be notified via email when his query would be solved via experts.

## SWOT Analysis

Figure 7: SWOT Analysis

# Chapter 8

# Conclusion & Future Work

**Chapter 8:** Conclusion & Future Work

1. 1. Achievements and Improvements

After importing dataset, we have created entity pairs among data on the basis of relations present among them. After that we got a knowledge graph of the data. It will be integrated to the app and will answer the queries from the provided datasets. Backend is completed and it just need to be linked with the front end to resolve queries.

* 1. Future Enhancements/Recommendations

In light of the above FYP, it is quoted that "Knowledge is Power", therefore every individual that has access to the internet and hunger for seeking information should be provided with ease. For future researchers, we would help them with our API. They can create a knowledge graph of any datasets they like, and our API would handle the rest. We’re focusing our datasets to be academic relevant as described, but it will act as a template for other programmers, so they create programs with datasets of different scope, be it medical or (any data related) scope.

# Chapter 9

# Paper Writing

**Chapter 9:** Paper Writing



## Paper Writing

Scientific Knowledge Base Creation via Intention Mining

Hamza Ali

Bachelor in Computer Science

Air University

Islamabad, Pakistan

[181058@students.au.edu.pk](mailto:181058@students.au.edu.pk)

Muhammad Huzaifa

Bachelor in Computer Science

Air University

Islamabad, Pakistan

[181122@students.au.edu.pk](mailto:181122@students.au.edu.pk)

**Solving a user’s query has been tough since the use of the internet, as the internet is the source of information and spams. For getting a solution, one relies on google but sometimes it gives false information via spamming tags. For that, one needs to have a search engine that would yield solutions from authentic sites and scholarly articles. For that, we would be using web crawling, NLP and numerous data processing python tools for giving authentic solutions for students, teachers and any other tech seekers. Our web-app would simply input search query along with certain categories like Linux, Linux → server (sub category). After the input field, user’s queries would be entertained. In some cases, if the tools fail to gather relevant data, queries can either be posted to our forums, which would be held till a certain time, and after then  (unsolved) it will be posted to stack overflow. This process will give out authentic solutions to user’s queries.**

***Keywords— knowledge base, knowledge graphs, NLP(****key words****)***

# Introduction

Getting authentic information has been quite difficult now-a-days due to a large number of scams/ fake meta tags. The huge internet is useful if one knows how to use it for themselves. For a developer/ tech seeker, one would always consider relevant and up to date solutions for their queries, therefore they depend on getting that without wasting time in browsing through each link. Similarly, for newbies, they would get many links for their queries, each with relevant or fake meta tags, but they can’t distinguish between the both. In both cases, they want a platform that can provide authentic and up to date solutions to their problems. They just need to provide the queries and relevant categories to our server, and let the server handle the rest, which would crawl the web and academic datasets for the solutions. Provided the case that if the solution could not be found, then the queries (with unique tokens) would be posted to our forums where our experts will answer it for them.

# Background

Tech-seeker (seniors) use only reputed sites for their solution, be it the new technology or some help with the research, but for our juniors, they need certain guidance to distinguish between the reputed sites and unauthentic sites. For that, we are developing a knowledge base for scientific documents, which would include different tech terminologies, from AI, BigData, and Linux to certain small queries. A knowledge base is a self-serve customer service library that includes information about a product, service, or some specific topic. A knowledge base allows you to create self-service customer support content around recurring topics, issues, and themes. A knowledge base doesn’t simply have to be a reservoir of help center articles. It can also include functions like a frequently-asked questions (FAQ) section, a user forum, articles, white papers, how-to articles, video tutorials, case studies, and dictionaries or glossaries — essentially, anything that helps customers understand and use your products or services.

# Key Points

## *Abbreviation*

* KB → Knowledge Base
* NLP → Natural Language Processing
* UI → User Interface
* FAQ → Frequently Asked Questions
* FYP → Final Year Project

## *Glossary*

* Knowledge Base → A self-serve online library of information about a particular product, service, department, or topic.
* Knowledge Graph → A graph representing subject, action and object
* Stop words → common words that are unnecessary in building graphs like as, before, did, doesn’t, etc. For more info, see [stopwords](https://countwordsfree.com/stopword).

# goals and objective

By taking input and categories from the user, our goals will be:

* Extracting the required information for specific topic from available datasets,
* Extraction of entity pairs from grammatical patterns from text using NLP library SpaCy,
* Developing relationships among text,
* Knowledge graph formation

1. Working

Our main focus is to gather relevant data from the datasets that would be used to plot the knowledge graphs and then apply queries as per user’s request. Implementation of the knowledge graph is completed but the queries.

1. Data cleaning

First, we need to clean the data. For that, we would

1. Create token and replace unnecessary punctuations with null
2. Eradicate stopwords tokens from the rest.
3. Finding entities

After data cleaning, we would get the different entities from cleaned data. Here, we would getting the subject and the object

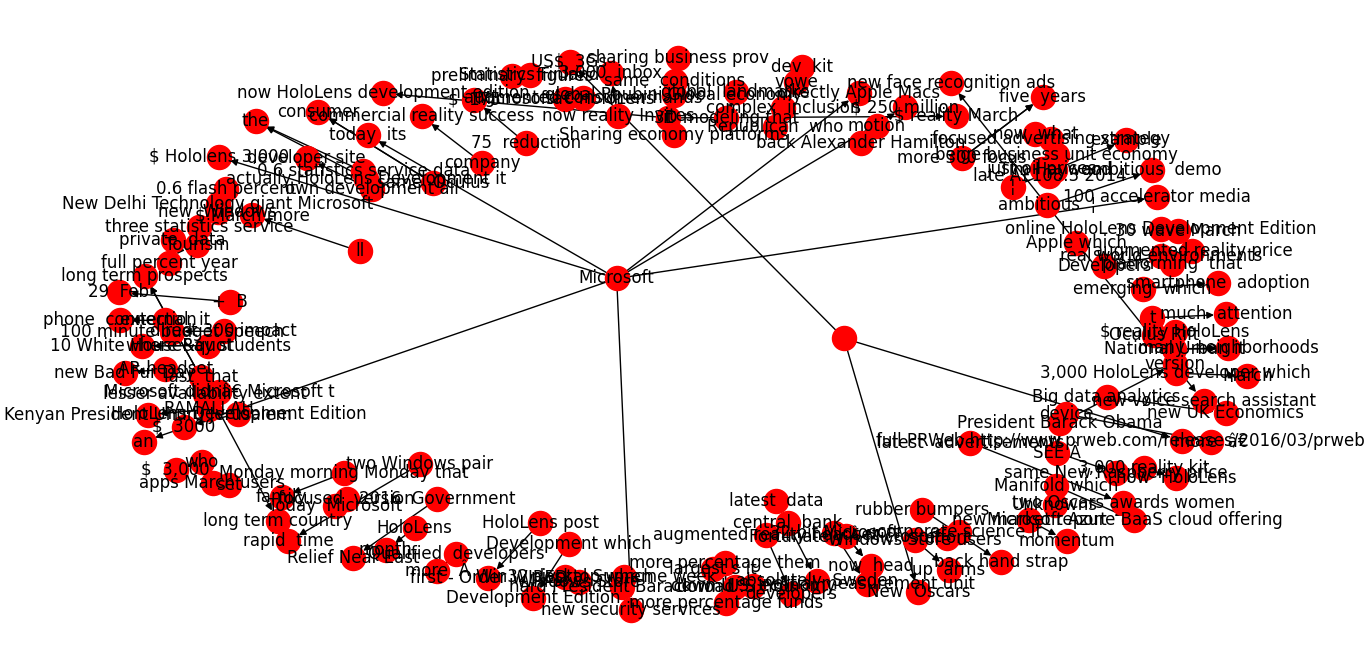
1. Finding relations

After finding the entities successfully, we would need to find the relation between those entities. For that. we would use matcher class, that would match the vocabulary and find the possible matches for those entities.

1. Create the Knowledge Graph

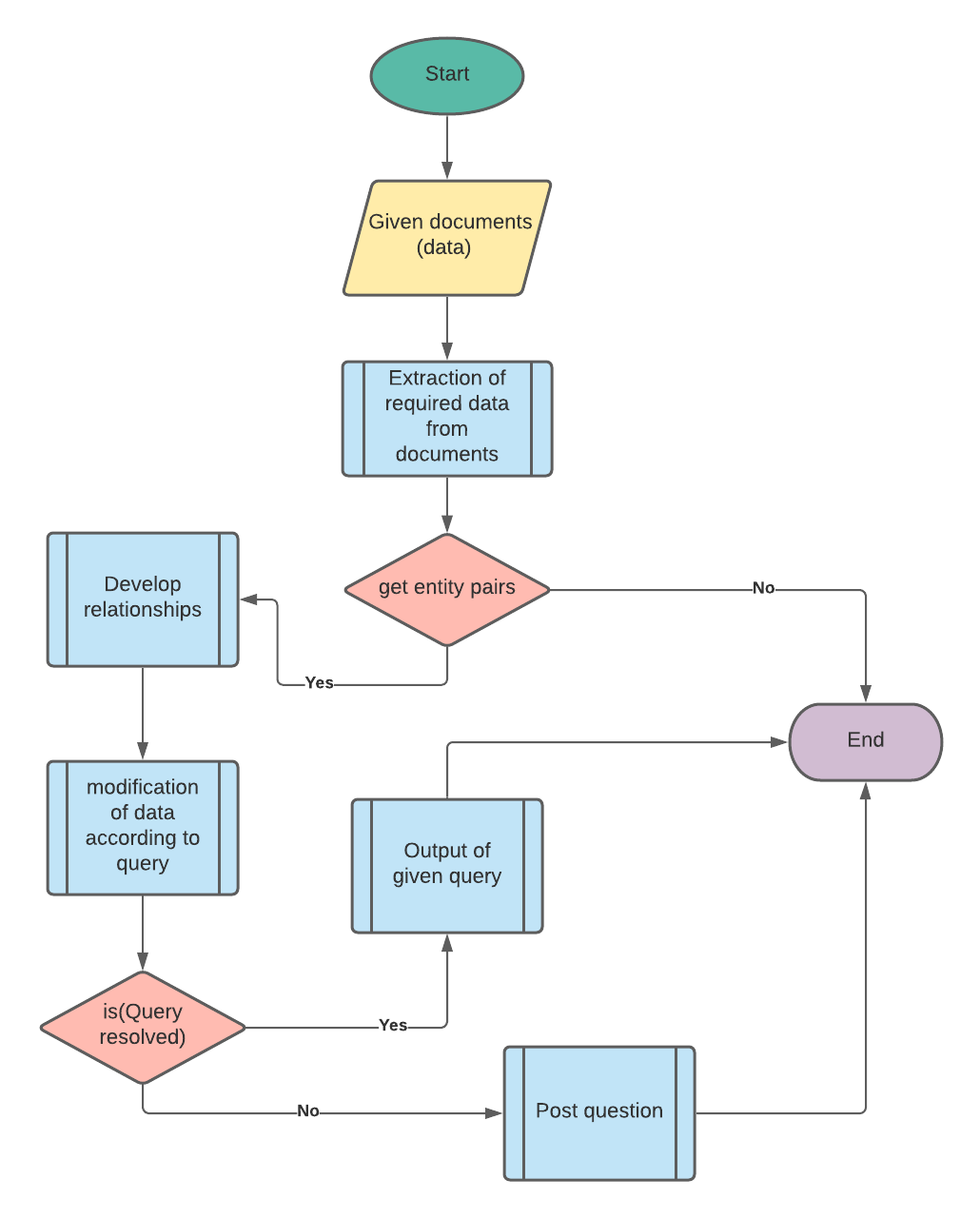
Next, we would be plotting the graph.  We would need to create a list of source and target from entities, and edge from relation. For plotting, we would create a data frame (using pandas library) of source and target.

1. Output



1. Flow Diagram

Here’s the visualization of the project. After the necessary knowledge graph, we would use query [4] entered by the user. If that query can be solved by the datasets, then it would be shown, else the question would be posted to a forum where users have to wait for the solution and would be notified if the solution is found.



1. Proposed Approach
   1. Data Set Exploration

in our dataset, we have found the following limitations

* Tackling of large
* Web Crawling (request approval)
* Faster execution with higher accuracy

1. Conclusion

Creation of knowledge base will not only benefit every student, or teacher but every tech geek around the internet. Searching out for any solution from installation and new technology is troublesome when one doesn’t know what the internet has to offer on the back of each website. There is a dire need for a platform that would provide solutions to all academic problems from its datasets and crawling features in a short span. For this, we will be creating a simple web-app with an easy UI that would only need user’s input and categories for optimizing search for web and datasets. Our datasets would only be the combination of recent tech-related articles, where users may search for different developing languages. For crawling, we will be utilizing the authentic and trust-worthy sites rated by many developers. Both of these features will be used to enhance the solution quality to each queries. There is a possibility where the internet and datasets fail to give out quality solutions, in that case, that query would be forwarded to the experts. In short, the user will need to input the query and it’ll be entertained with faster and accurate execution.

1. References

[1] Rabah A. Al-Zaidy C. Lee Giles Automatic Knowledge Base Construction From Scholarly Documents

[2] Deepanshi Dhingra <https://www.analyticsvidhya.com/blog/2021/06/text-preprocessing-in-nlp-with-python-codes/>

[3] Aravind CR <https://neptune.ai/blog/web-scraping-and-knowledge-graphs-machine-learning>

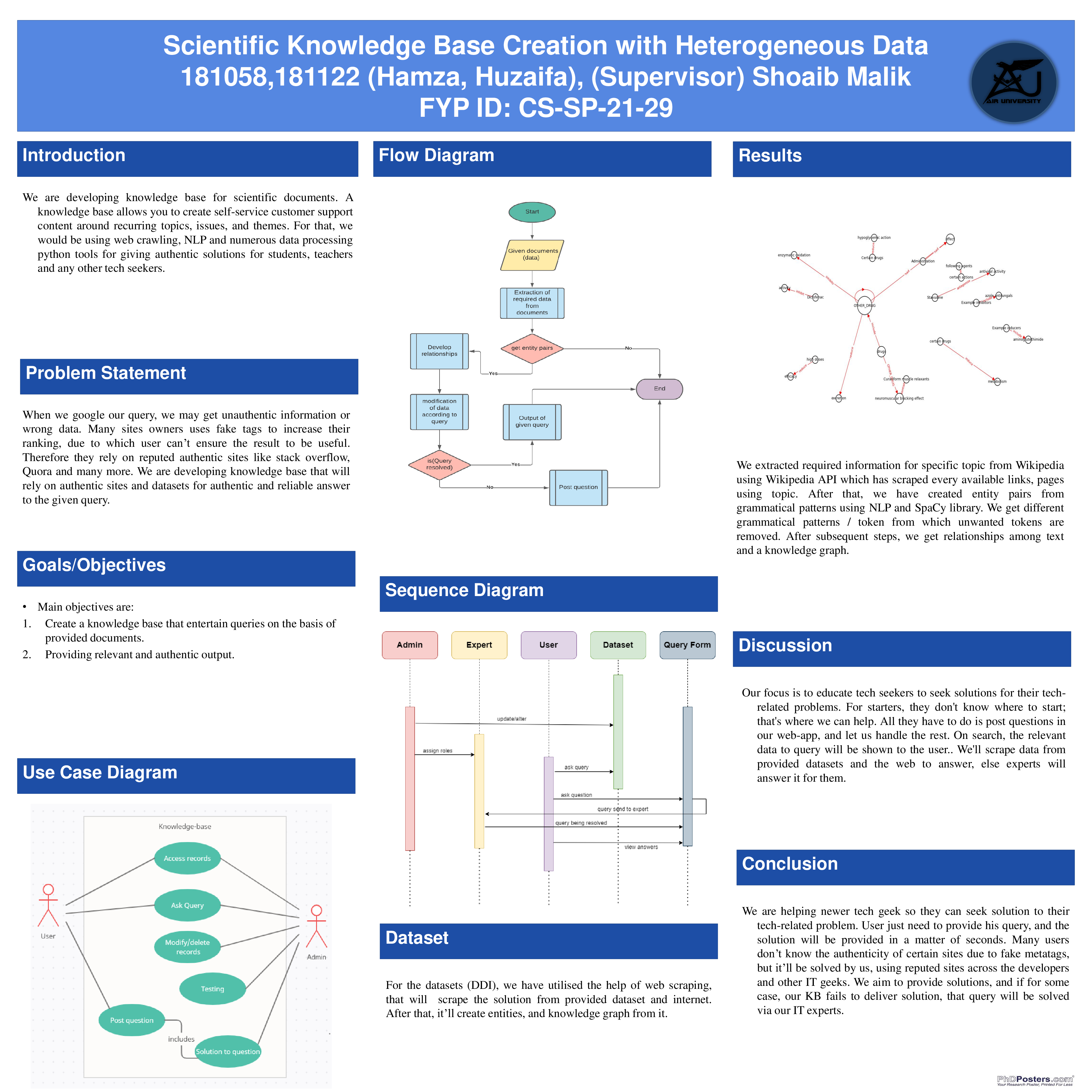
[4] G. Xiao, L. Ding, B. Cogrel, & D. Calvanese. Virtual knowledge graphs: An overview of systems and use cases. Data Intelligence 1(2019), 201-223. doi: 10.1162/dint\_a\_00

# Appendices

# Appendix A: Information / Promotional Material

This given poster would highlight the general idea of our project.

* 1. **Poster**

Figure 8: FYP Poster

## References

[1] Rabah A. Al-Zaidy C. Lee Giles Automatic Knowledge Base Construction From Scholarly Documents

[2] Deepanshi Dhingra <https://www.analyticsvidhya.com/blog/2021/06/text-preprocessing-in-nlp-with-python-codes/>

[3] Aravind CR <https://neptune.ai/blog/web-scraping-and-knowledge-graphs-machine-learning>

[4] G. Xiao, L. Ding, B. Cogrel, & D. Calvanese. Virtual knowledge graphs: An overview of systems and use cases. Data Intelligence 1(2019), 201-223. doi: 10.1162/dint\_a\_00