AN INDUSTRIAL TRAINING REPORT

**on**

**NLP2SQL**

**Submitted by**

**Akrati Chaturvedi**

**161500048**

**Manika Agrawal**

**161500304**

**Sajal Agrawal**

**161500475**

Department of Computer Engineering & Applications



**Institute of Engineering & Technology**

**Table of Contents**

|  |  |
| --- | --- |
| Declaration | iii |
| Certificate | iv |
| Acknowledgments | v |
| 1. **Introduction** | **1** |
| 1.1 1.1 Overview and Motivation | 1 |
| 1.2 Objective | 2 |
| 11 1.3 Summary of similar Applications | 3 |
| 1. **Company Profile** | **7** |
| 1. **Project Design** | **16** |
| 3.1 Data Flow Diagrams | 18 |
| 1.1 3.2 Use Case | 1 |
| 3.3 Sequence Diagrams | 2 |
| 3.4 ER Diagram | 2 |
| 11 3.5 Detailed Description Of Database Tables | 3 |
| 1. **Implementation and User Interface** | **27** |
|  |  |

**CERTIFICATE**

This is to certify that the above statements made by the candidate are correct to the best of my knowledge and belief.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Project Supervisor**

Mr. Pankaj Sharma

Assistant Professor

Institute of Engineering and Technology

Department of Computer Science and Applications

Date:

**DECLARATION**

We hereby declare that the work which is being presented in the Mini Project Report **“Title : NLP2SQL”,** in partial fulfillment of the requirements for Mini project LAB, is an authentic record of our own work carried under the supervision of **Mr. Pankaj Sharma, Assistant Professor, GLA University, Mathura**.

Akrati Chaturvedi

161500048

Manika Agrawal

161500304

Sajal Agrawal

161500475

**ACKNOWLEDGEMENT**

I would like to thank Mr. Pankaj Sharma, Assistant Professor at GLA University, Mathura for providing the guidance on Natural Language Processing. We work under his guidance and supervision. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us.

This project would never have seen the light of day without the help and guidance we have received.

Akrati Chaturvedi

161500048

Manika Agrawal

161500304

Sajal Agrawal

161500475

**1. INTRODUCTION**

**1.1 OVERVIEW AND MOTIVATION**

**1.2 OBJECTIVE**

The main objective of our website, Grepskill is to search the students according to their skills. It provides you the search bar where you can enter the skill name for which you want to search the students. It displays you the no. of students and also the brief description of those students who have that respective skill. Our website, Grepskill not only accept the single skill, but also search for the multiple skills. Grepskill provides you the filter page to search the students easily according to their course, department, or year. Another objective of our website is to come out with the Students of the semester, Coder of the semester, University merit holders. It also displays the topper students in their department. Grepskill updates the students of all upcoming events & provides them the platform to register themselves in their interested events online. The students can also design their resumes on out website and they can update their resumes whenever require.

**1.3 SUMMARY & SIMILAR APPLICATION**

The applications that we found similar to our project were Internshala & Linkedin.

**Internshala -** It is one of the biggest internship platform in India with 1.5 million students and 40,000 companies. The website is quite clean and user-friendly and has variety of options for students to filter the internships like category, location, work from, and duration.

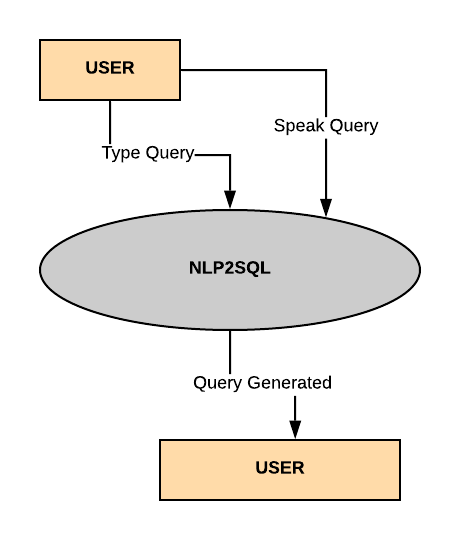
**Linkedin -** LinkedIn is a social network for professionals. Whether you're a marketing executive at a major company, a business owner who runs a small local shop or even a first year college student looking to explore future career options, LinkedIn is for anybody and everybody who's interested in taking their professional life more seriously by looking for new opportunities to grow their careers and to connect with other professionals.

**3. PROJECT DESIGN**

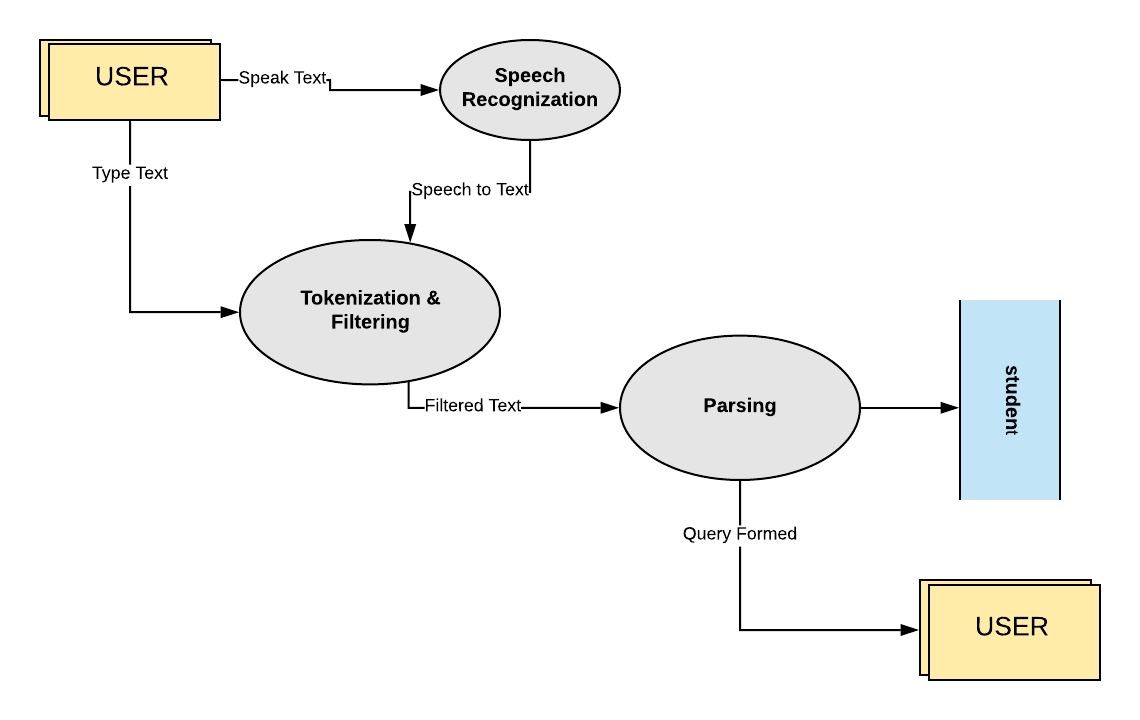
This report includes Data Flow Diagrams (DFD), Use Case Diagram, Sequence Flow Diagrams and Algorithms Flowcharts to represent the design of our project.

**3.1 DATA FLOW DIAGRAMS**

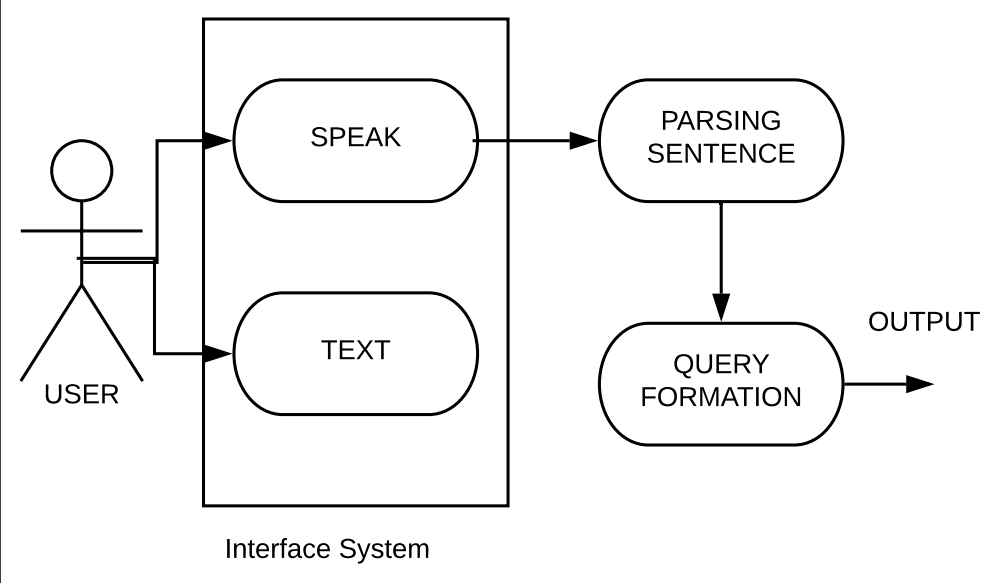
**LEVEL 0 : DATA FLOW DIAGRAM**

****

**LEVEL 1 : DATA FLOW DIAGRAM**

****

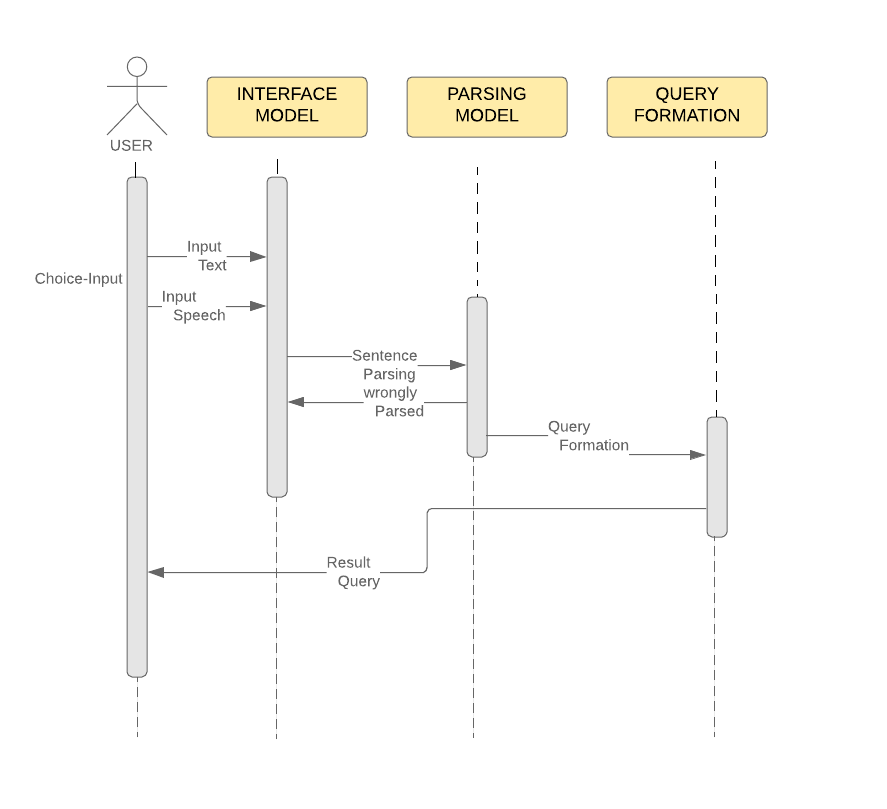
**3.2 USE CASE**



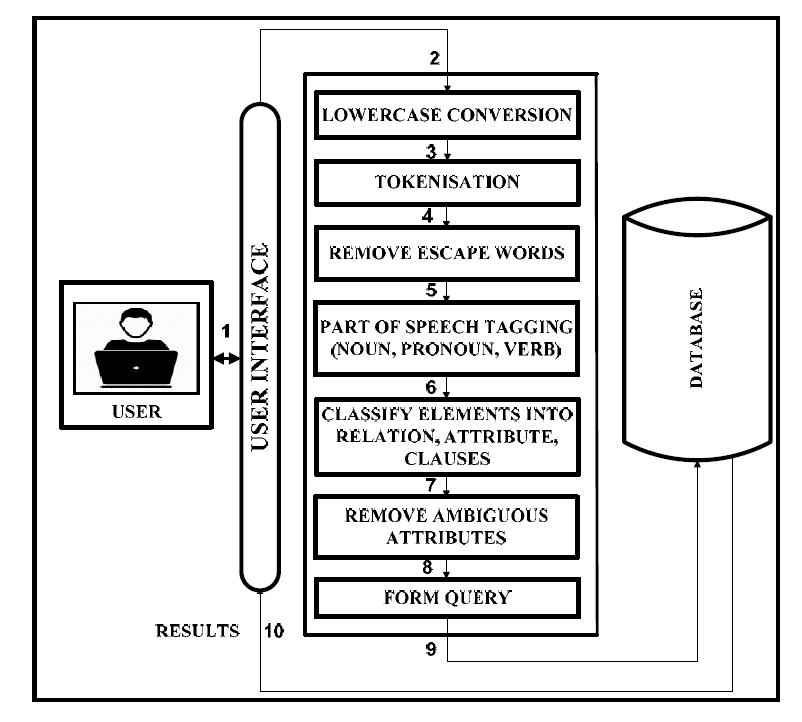
|  |  |
| --- | --- |
| **USE CASE** | **DESCRIPTION** |
| **Speak :** | User can speak the query through microphone of the system |
| **Text :** | User can write the query |
| **Parsing Sentence :** | The query as string is tokenised into words and then filtered to form SQL query |
| **Query Formation :** | The filtered query is passed through this algorithm and the query is formed as output query |

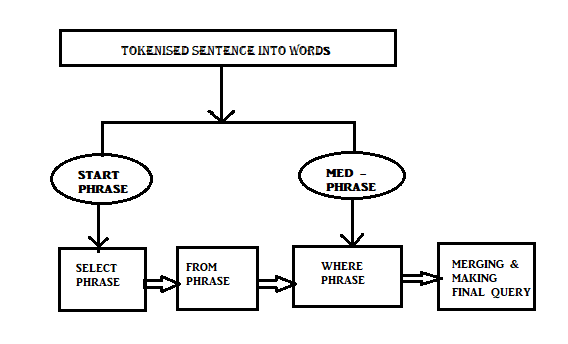
**3.3 SEQUENCE DIAGRAM**

**USER INTERACTION**

****

**3.4 ALGORITHM FLOWCHART**

****

****

**Fig : Parsing Phrase to analyse query type**

**3.5 DETAILED DESCRIPTION OF DATABASE TABLE**

**Table 1- student**

|  |  |  |
| --- | --- | --- |
| **Name of Field** | **Data Type** | **Explanation** |
| rollno | int | University rollno of the student (PK) |
| name | int | Name of the Student |
| phone | varchar | Contact No. of Student |
| section | char | Section of Student |

**4. IMPLEMENTATION AND USER INTERFACE**

This project was implemented as follows -

**ALGORITHM**

* Accept the input from the user either in the form of speech and convert it to text or directly in the form of text.
* If you take the input in speech, then convert it to text by Speech Recognition using Android.
* Split the input query and store it in a list, i.e. tokenize the input sentence.
* Find all the attributes of all the tables.
* Examine the query and find the table present in the query and the attributes present in the query.
* Find the attributes which belong to table present in the query.
* Find the attribute which do not belong to the table in the query (if any).
* Now find the tables which will contain the pair of ((attribute which do not belong to the table in the query), (other attributes present in the table in the query)).
* Select any one table. Thus we will obtain the tables required for natural join.
* For a natural join query, find out the common attribute of the 2 tables and form the inner query.
* Then form the outer query according to the different conditions.. Merge both of them and generate the final query. For a simple query, generate the final query by checking the different conditions accordingly.
* If there are 2 tables, then perform a natural join on the 2 tables with appropriate attributes of the tables.
* Obtain the conditions of the where clause (single condition or multiple condition by finding the ―and‖ word in the input query), aggregate function (checking whether any aggregate function (like sum, avg, count , etc) present in the query) and the relational operators between the conditions from the list of attributes. Add these to the final query.
* Print the final query.
* Write the generated SQL query in a ―log file‖ along with an index number and time of the query using the system clock, so that we can retrieve the query generated at a particular time giving the time as input.

**NATURAL LANGUAGE PROCESSING (NLP)**

Natural language processing is a field of computer science concerned with the interactions between computers and human (natural) languages.

These include spoken language systems that integrate speech and natural language.

The goal of NLP is to enable communication between people and computers without resorting to memorization of complex commands and procedures.

**NLTK [Natural Language Toolkit]**

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, part-of-speech tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

**SPEECH RECOGNITION LIBRARIES**

1. SpeechRecognition [Used to recognize the speech and convert it into text]
2. Pyaudio [[PyAudio](http://people.csail.mit.edu/hubert/pyaudio/#downloads) is required if and only if you want to use microphone input]

**VISUAL STUDIO**

Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.

**GITHUB**

Github is an open source code repository and collaborative development platform.It offers a location for online code storage and collaborative development of massive software projects. The repository includes version control to enable hosting different development chains and versions, allowing users to inspect previous code and roll back to it in the event of unforeseen problems.

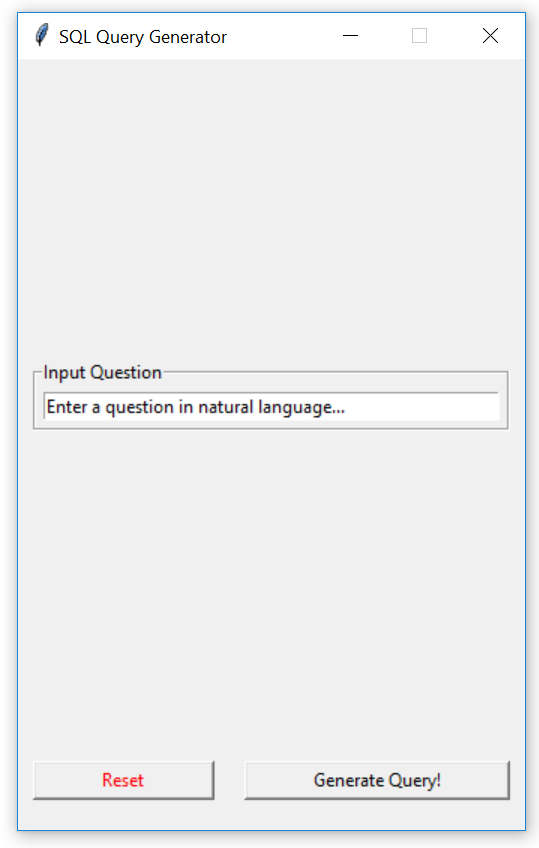
Our github link for the project [**https://github.com/3musk/nlprepo1.git**](https://github.com/3musk/nlprepo1.git)

**USER INTERFACE**

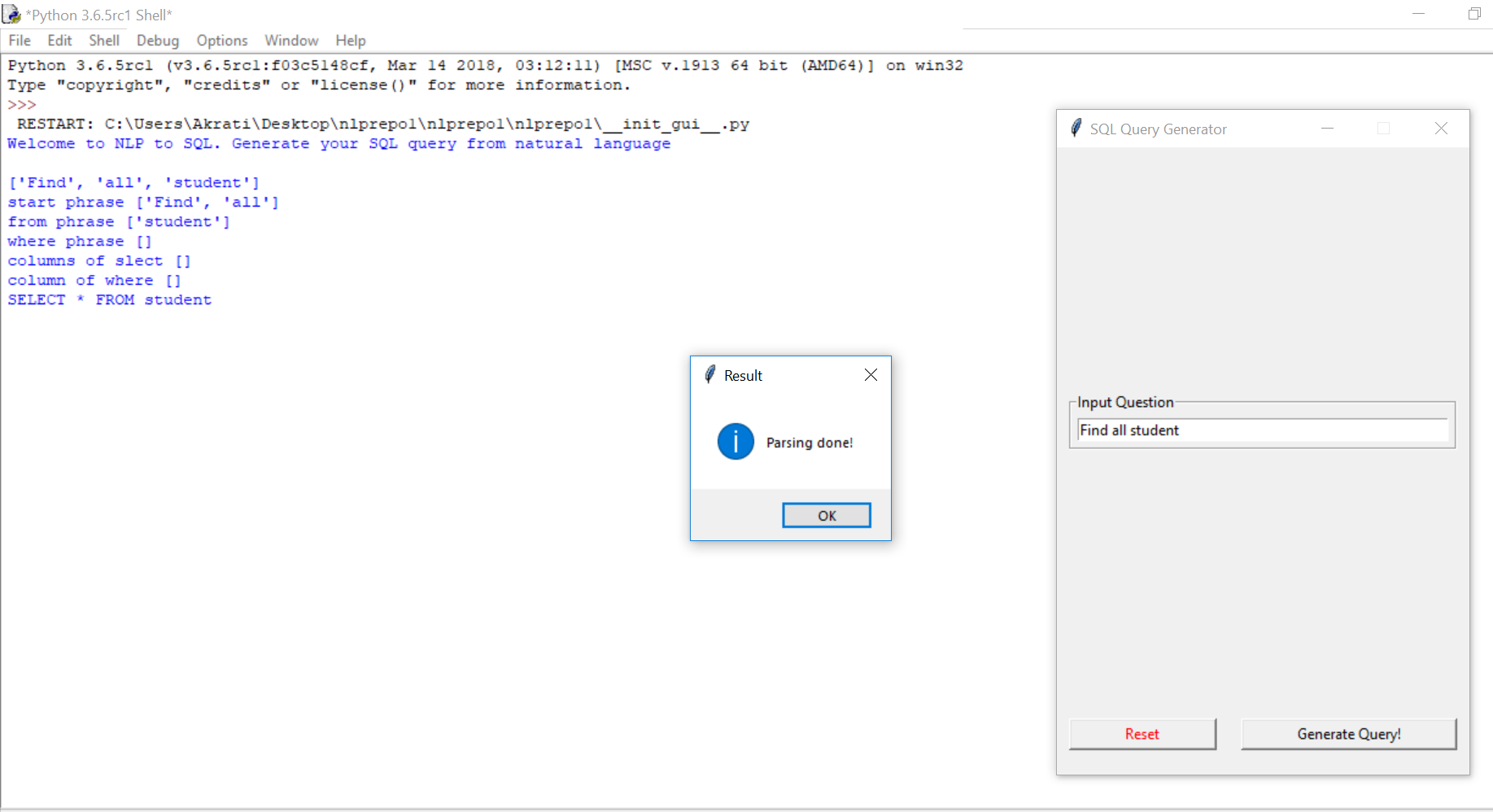
**Using Command Prompt**

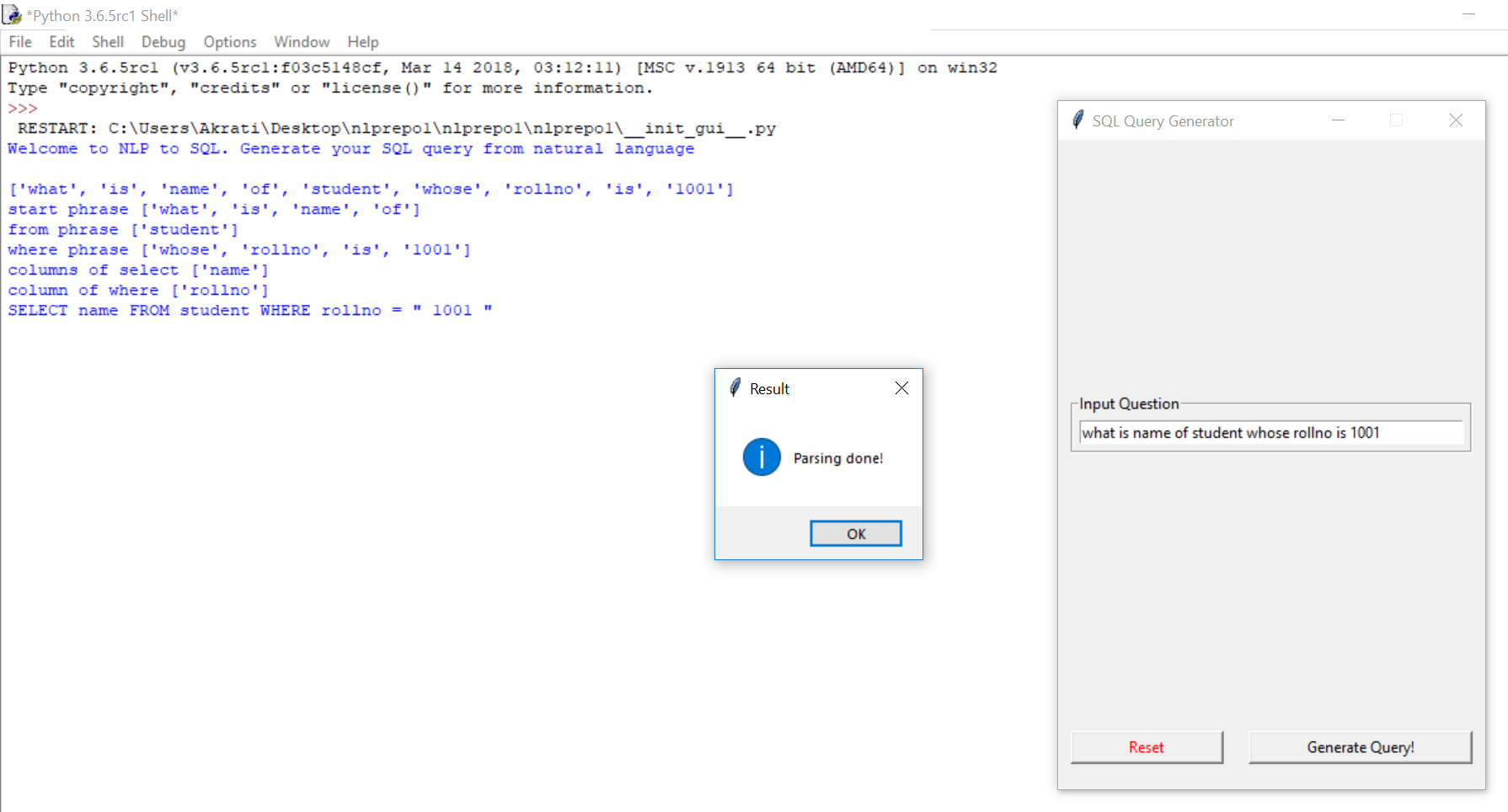
****

**Using GUI**

****

**Parsing Done (Query Formed)**

****

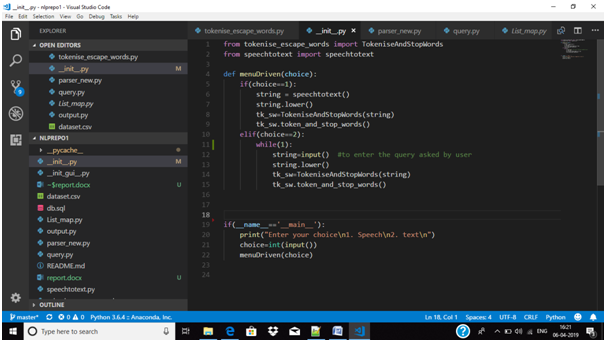
****

**5. Bibliography/References**

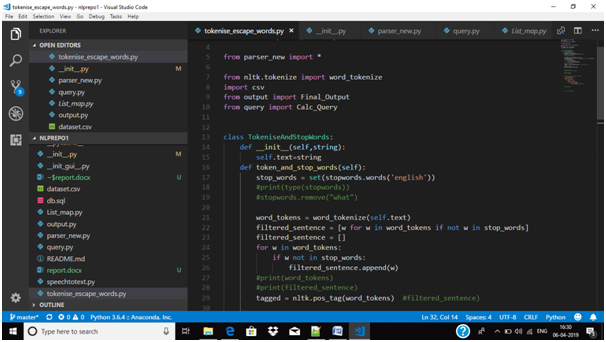
* Hendrix, G.G., Sacerdoti, E.D., Sagalowicz, D., Slocum, J. ―Developing a natural language interface to complex data‖, in ACM Transactions on database systems,1978,pp.105147.
* Huangi,Guiang Zangi, Phillip C-Y Sheu ―A Natural Language database Interface based on probabilistic context free grammar‖, IEEE International workshop on Semantic Computing and Systems 2008
* <https://pypi.org/project/SpeechRecognition/>
* <https://stackoverflow.com/>

**6. Appendices**

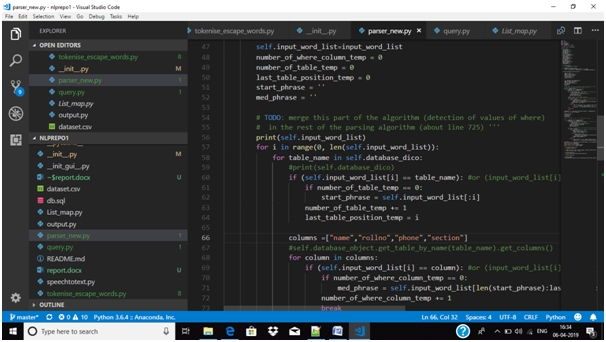
**CODE SNIPPETS**

****

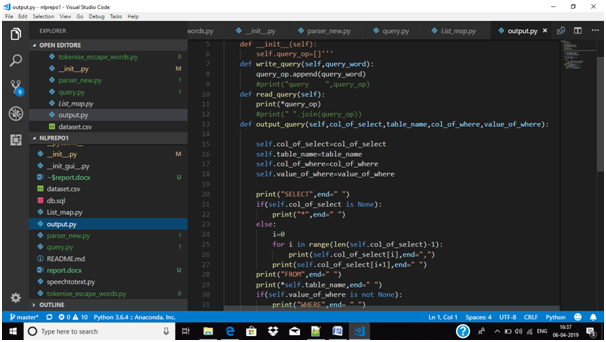
This is the initial file “\_\_init\_\_.py” which starts with asking the choice to enter query by typing or by speaking. It takes the query as string then passes on to the other module or class where it gets its further computation.

****

In this module (class) the sentence gets tokenized and done part-of-speech tagging. After which the escape words are taken out from the tokenized list and the filtered sentence is then passes to parsing to convert it into desired query.

****

In this code class , the tokenized sentence is broken into two parts the start phrase and the med phrase where the start phrase is the SELECT phrase and then concatenated the FROM phrase(which has the table name, from which the data to be retrieved) and then the med- phrase contains the WHERE phrase. This broken query is passed onto the output file.

****

This is the output file of the last module which generates the output for the entered sentence. This takes the broken query from parsing file and joins it to make the desired or the final query.