ALPHABOT

Prepared by

Arjun Rupavatia (17IT098)

Palak Jadhav (16IT029)

Huzefa Jambughoda (16IT030)

Smit Ladani (16IT043)

Kashyap Nirmal (16IT059)

Under the supervision of

Prof. Kamlesh Makvana

A Report Submitted to

Charotar University of Science and Technology

for Partial Fulfillment of the Requirements for the

Degree of Bachelor of Technology

in Information Technology

IT345 Software Group Project-II

Submitted at

DEPARTMENT OF INFORMATION TECHNOLOGY
Chandubhai S. Patel Institute of Technology
At: Changa, Dist: Anand – 388421
Oct 2018

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that the report entitled "**Alphabot**" is a bonafied work carried out by **Mr. Arjun Rupavatia(17IT098), Ms. Palak Jadhav(16IT029), Mr. Huzefa Jambughoda (16IT030), Mr. Smit Ladani(16IT043), Mr. Kashyap Nirmal(16IT059),** under the guidance and supervision of **Prof. Kamlesh Makvana** for the subject **IT345 Software Group Project-II** of 5th Semester of Bachelor of Technology in **Information Technology** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate **himself/herself**, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Under supervision of,

Prof. Kamlesh Makvana
Assistant Professor
Dept. of Information Technology
CSPIT, Changa, Gujarat.

Prof. Parth Shah
Head & Associate Professor
Department of Information Technology
CSPIT, Changa, Gujarat.

ACKNOWLEDGMENT

We would like to express our special thanks of gratitude to our project guide Prof. Kamlesh Makvana as well as our HOD Dr. Parth Shah who gave us the opportunity to do this wonderful project. A project while in fifth semester is really helpful to give us a vision, how our learned things are implied in real life situations. We even learnt to learn the things that are outside our syllabus. The project also helped in doing a lot of research and we came across many new things.

Secondly, we would like to thank our parents, friends and family who helped to make this project possible and work in appropriate time.

ABSTRACT

This project is a chat-based bot, named Alphabot. Alphabot responses to the queries which are fired at it. These responses are appropriate according to the conditions for which this bot is synthesized.

This is primarily an enquiry-based bot. It can be useful in places where a person cannot be assigned 24 x 7 for FAQs. With the help of this application one to one correspondence between the organization and its information and its consumer can be established. The bot can be used for its services in any kind of organization, before it would go in all set position and ready to work in the field, it would just need minor technical necessary changes to be done to turn its knowledge-base for the organization it would serve to.

The application is based on python. Python, because it is rich with libraries which are easy to work with in AI.

This report contains all the primary technical information about this bot described in its five fundamental modules.

TABLE OF CONTENT

LIST OF FIGURES

MODULES WORKING TOGETHER
FLOWCHART

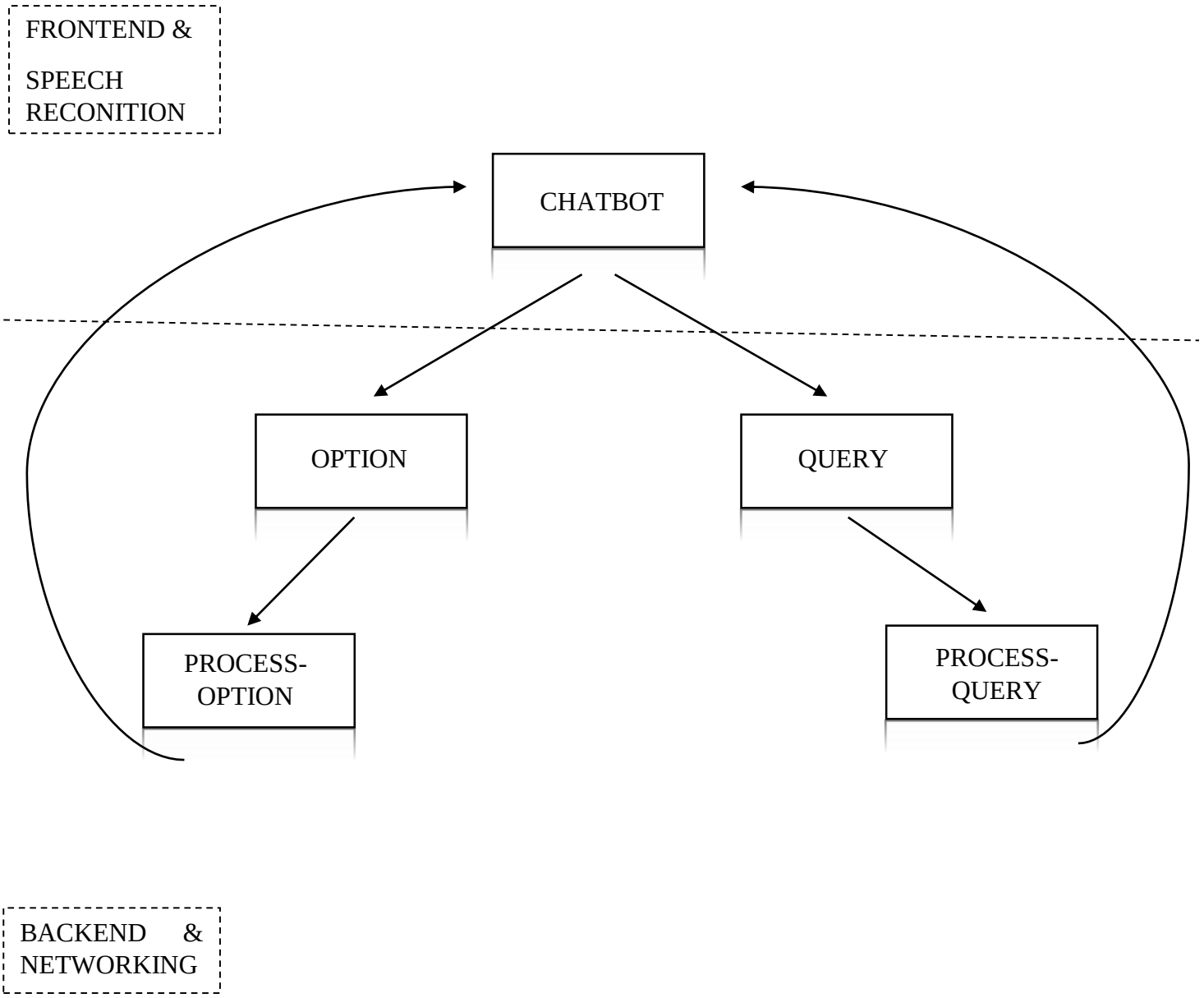

Figure 1 Flowchart-I: Working of the modules together as a system

MODULE 1: BACKEND

INTRODUCTION

The main aim of this module is to process the query or option which is given by the user. The module will take the query n pass it to the dedicated function for processing and after processing generated result is delivered to the chatbot after fetching an appropriate response from the dataset.

WORKING

Backend Module for handling the queries:

User has two options

1) Select appropriate option

2) Write the whole query.

- Select appropriate option

  When user select appropriate option then using jQuery AJAX will call the process_option() function in the backend and this function will take the selected option and using hierarchy of data it will make json object of other appropriate options or answer and this json object is return back to the ajax file and with the help of jQuery options or answer is shown. In this option selection process, this function will keep track of the option which is selected by the user in order to retrieve specific data from database.

- Write the whole query

  Now if user write the whole query then using Jquery AJAX will call the process_query() function in the backend and this function will take the user query and stem this query and convert in lower case and find the probability of user query which is mapped with probability of every queries of database and will find for query having maximum probability with user query and answer corresponding to this query will be return by this function. Suppose if user query doesn't match with any query of database then it will return 'Sorry! I didn't understand you…' and this answer is return back to the AJAX call.

FLOWCHART

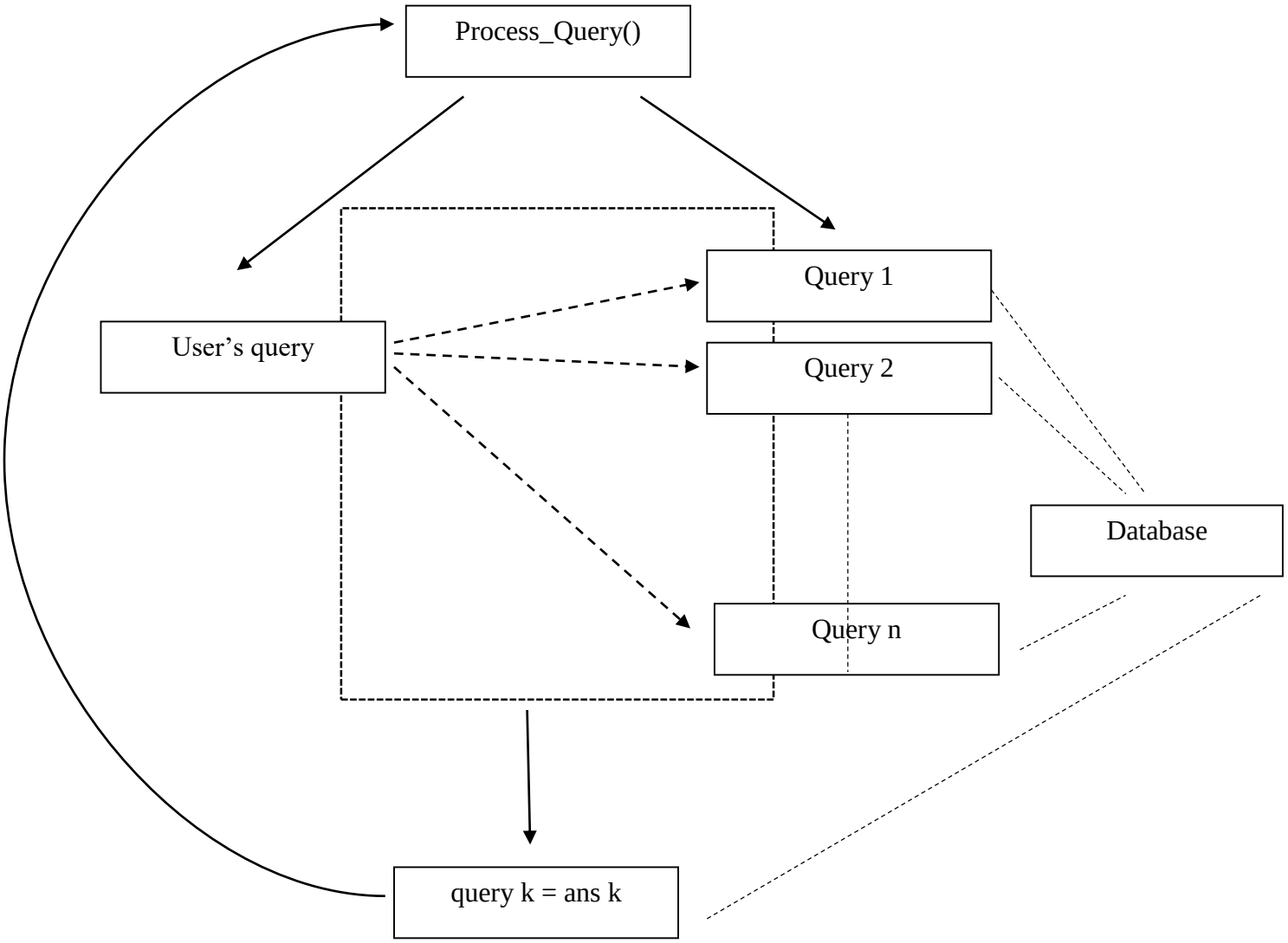

Figure 2 Flowchart-II: Query Processing

TECHNOLOGY/TOOLS USED

- Python
- HTML, CSS, jQuery, AJAX
- Flask
- NLTK
- PHP
- SQLite3

FIGURES

```
$('#toggleEffect').on('click', function(){
    var src = $('.box_head img').attr('src');
    $('.box_body').slideToggle('fast');
    if(src == 'https://maxcdn.icons8.com/windows10/PNG/16/Arrows/angle_down-16.png'){
        $('.box_head img').attr('src', 'https://maxcdn.icons8.com/windows10/PNG/16/Arrows/angle_up-16.png');
    }
    else{
        $('.box_head img').attr('src', 'https://maxcdn.icons8.com/windows10/PNG/16/Arrows/angle_down-16.png');
    }
});

$('.chat-text input').keypress(function(event) {
    if(event.keyCode == 13){
        var msg = $(this).val();
        $(this).val('');
        $('.msg-insert').append("<div class='msg-send'>"+msg+"</div>");
        $.ajax({
            url: 'http://127.0.0.1:5000/process_query',
            type: 'POST',
            data: {'data': msg},
            success: function(response){
                $('.msg-insert').append("<div class='msg-receive' id='receive-"+receiveId+"'>"+response+"</div>");
                $('.box_body').animate({
                    scrollTop: $('.box_body')[0].scrollHeight
                }, 10);
                speak(receiveId);
                receiveId++;
```

Figure 3 Backend Code Snippet-I

```
def process_options():
    conn = sqlite3.connect('Data/Options.db')
    con = conn.cursor()
    query = request.form['data']
    if query == 'About Charusat University':
        ans = json.dumps([-1, 'Charotar University of Science and Technology (CHARUSAT) has been conceived by Shri Charotar Moti Sa
    elif query == 'Charusat Address':
        ans = json.dumps([-1, 'CHARUSAT Campus, Highway 139, Off, Nadiad - Petlad Road, Changa, Gujarat 388421'])

    elif query == 'Show all Institute':
        ans = json.dumps([i[0] for i in con.execute('SELECT Name FROM Institute').fetchall()])

    elif query in [i[0] for i in con.execute('SELECT Name FROM Institute').fetchall()]:
        selectedOption['institute'] = con.execute("SELECT Institute_id FROM Institute WHERE Name='" + query + "'").fetchone()[0]
        ans = json.dumps([i[0] for i in con.execute('SELECT Name FROM Programs WHERE Institute_id=' + str(selectedOption['institute

    elif query in [i[0] for i in con.execute('SELECT Name FROM Programs WHERE Institute_id=' + str(selectedOption['institute'])).fe
        selectedOption['type'] = con.execute("SELECT Prog_id FROM Programs WHERE Name='" + query + "' AND Institute_id=" + str(selec
        ans = json.dumps([i[0] for i in con.execute('SELECT Name FROM Branch WHERE Prog_id=' + str(selectedOption['type'])).fetchall

    elif query in [i[0] for i in
        con.execute('SELECT Name FROM Branch WHERE Prog_id=' + str(selectedOption['type'])).fetchall()]:
        selectedOption['name'] = con.execute("SELECT Branch_id FROM Branch WHERE Name='" + query + "' AND Prog_id=" + str(selectedOp
        ans = json.dumps([i[0] for i in con.execute("SELECT Name FROM Information WHERE Prog_id=" + str(selectedOption['type']) + "

    else:
        ans = json.dumps([-1, con.execute("SELECT Value FROM Information WHERE Prog_id=" + str(selectedOption['type']) +" AND Bran

    return ans
```

Figure 4 Backend Code Snippet-II

```
@app.route("/process_query",methods=['GET','POST'])
def process_query():
    user_query = request.form['data']
    return process(user_query)

def process(message):
    con = sqlite3.connect('Data/replies.db')
    c = con.cursor()
    keywords = Word_Frequency.Get_Tokens(message)
    token = ' '.join(keywords)

    c.execute('SELECT Tag FROM tab1 WHERE Token=\"{0}\"'.format(token))
    out = c.fetchone()
    if not out:
        out = MyChatBot.Process_Query(message,c)  # response from
        c.execute('SELECT Tag FROM tab2 WHERE Response=\"{0}\"'.format(out))
        tag = c.fetchone()
        if tag:
            tag = tag[0]
        else:
            c.execute('SELECT max(tag) FROM tab2')
            tag = c.fetchone()[0]
        c.execute('INSERT INTO tab1 VALUES(\"{0}\",{1})'.format(token, tag))
        c.execute('INSERT INTO tab2 VALUES({0},\"{1}\")'.format(tag, out))
        return out
    else:
        c.execute('SELECT Response FROM tab2 WHERE Tag={0}'.format(out[0]))
        return c.fetchone()[0]
```

Figure 5 Backend Code Snippet-III

MODULE 2: DATASET

INTRODUCTION

This module focuses on the dataset of the chatbot.

The procedure by which the query is filtered to process is the same procedure used here; the data is collected appropriately and filtered stored into the database table. Here, filtering means that the query is made into a unique string of tokens containing the information about the query. One table to store the filtered queries - so there is no need to process every time, making the response quick - and an another table having the replies that are to be given. The primary key of the table containing replies is used to refer the query in the queries table. This reduces redundancy as there can be a single answer to multiple questions.

WORKING

For option processing we have separate database named Options and in that database we have table named institute which contains name of all the institute and unique institute id, program which contains all the information of programs which is offered by all the institute and also contains unique program id and institute id, branch which contains information about all the branch of program and contain branch id and program id, and information table which contains information about all the branches of all the program offered by all institute and contain branch id and program id with unique information id.

This hierarchy structure of database is very useful for processing options efficient and unique id given to each table is useful to keep the track of selected options by the user

Flowchart

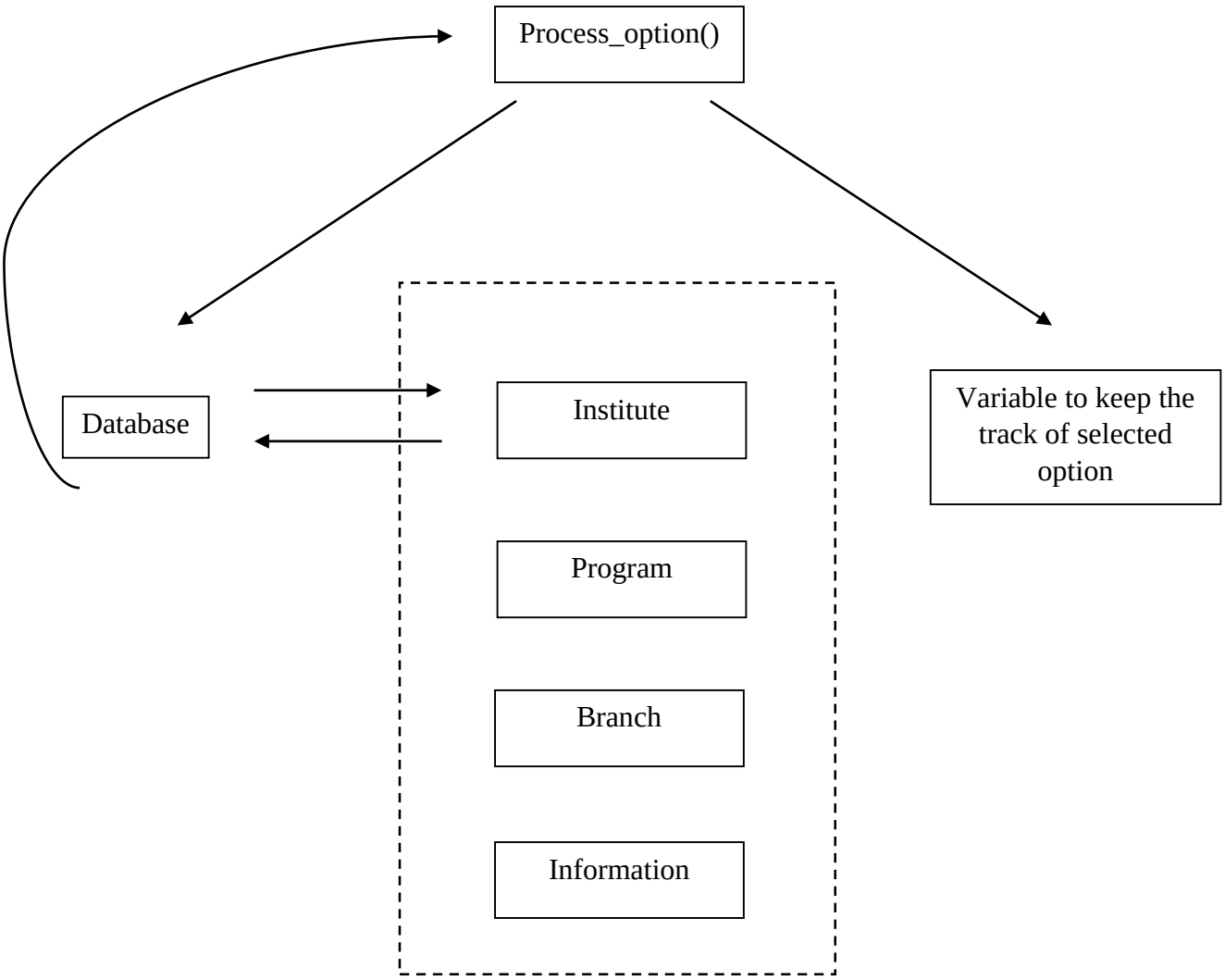

Figure 6 Flowchart-3: Database structure

TECHNOLOGY/TOOLS USED

- Python
- Sqlite3
- Pandas
- Bss
- sqlite3
- PorterStemmer
- nlt.stopwords

FIGURES

Table: questions

	Token	Tag
	Filter	Filter
1	b civil cspit engin last qualif tech	0
2	b civil cspit disciplin engin tech	1
3	b civil cspit engin tech unit	2
4	b civil cspit engin tech url websit	3
5	b cspit electr engin last qualif tech	0
6	b cspit disciplin electr engin tech	1
7	b cspit electr engin tech unit	2
8	b cspit electr engin tech url websit	4
9	b commun cspit electron engin last qualif tech	0
10	b commun cspit disciplin electron engin tech	1
11	b commun cspit electron engin tech unit	2
12	b commun cspit electron engin tech url websit	5
13	b cspit engin last mechan qualif tech	0
14	b cspit disciplin engin mechan tech	1
15	b cspit engin mechan tech unit	2
16	b cspit engin mechan tech url websit	6
17	b cspit inform last qualif tech technolog	0
18	b cspit disciplin inform tech technolog	1

Figure 7 A Snap of dataset -I

Figure 8 A Snap of dataset -II

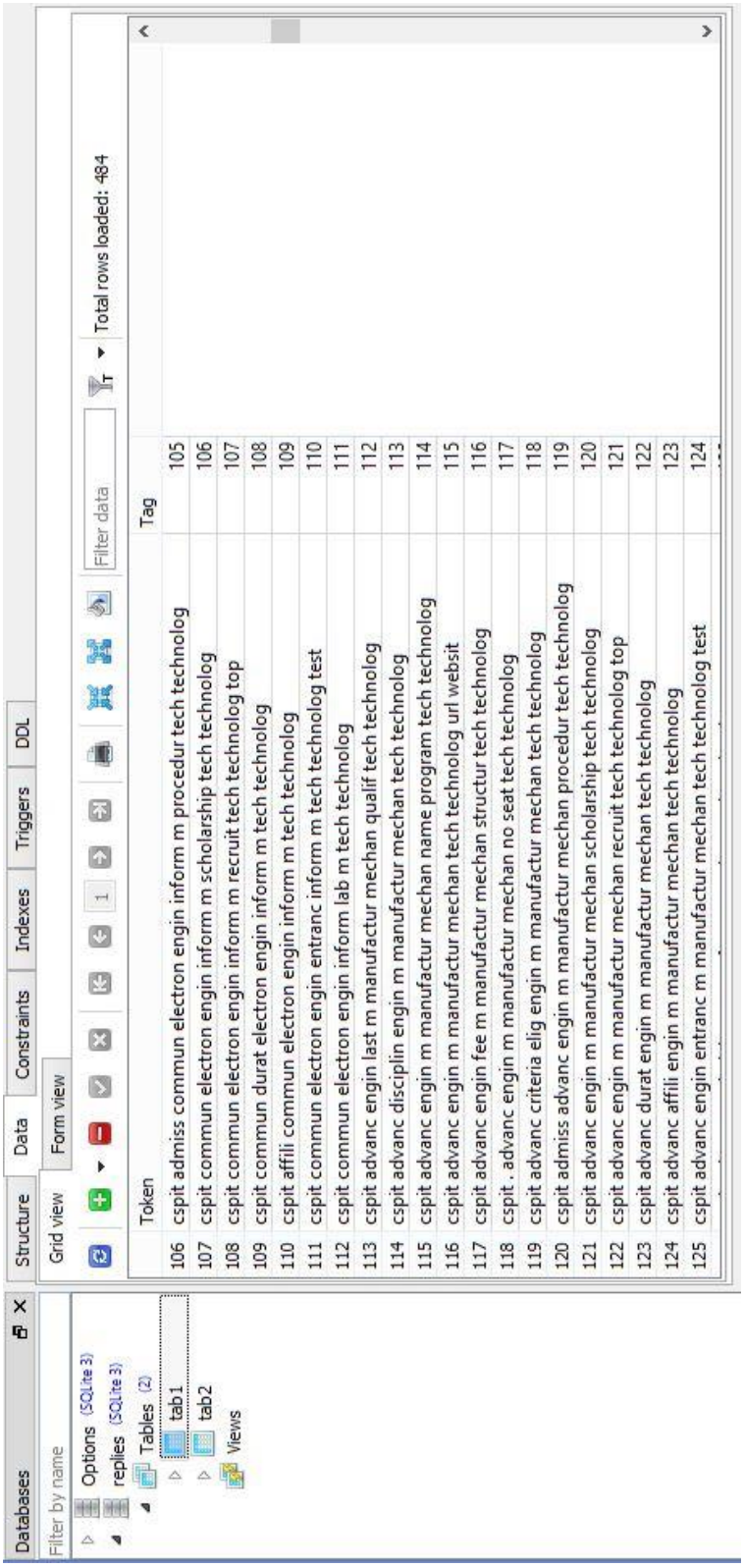

Figure 9 A Snap of dataset -III

MODULE 3: NETWORK

INTRODUCTION

Presently, the educational institutes are using the service of Chatbot. It may serve various purposes. One of the many important use cases may be **marketing**. There may be various ways to serve these purpose. One of the way we have recognized to achieve this purpose is by using the **Live IP**. One of the thing that we can recognize using the requested IP is region. So it can be used to serve the purpose of the promotion of the Institutes.

WORKING

- Based on the Live IP of the user request, we can keep the track of the useful information about user on the need basis.
- Here, we have used an API that can provide the information from the requested IP in **JSON** format.
- The information that can be retrieved are likely IP, Region, Latitude, Longitude, Proxy, Device and many more.
- From this retrieved information we can keep the record of the information of the regions that are seeking the request for information about the educational institute.
- Thus, this record can be used by various educational institutes for their promotions i.e. **Admission Seminars** etc.

TOOLS/TECHNOLOGIES USED:

- HTML
- CSS
- AJAX
- PHP
- MYSQL
- JSON
- WAMP

FIGURES

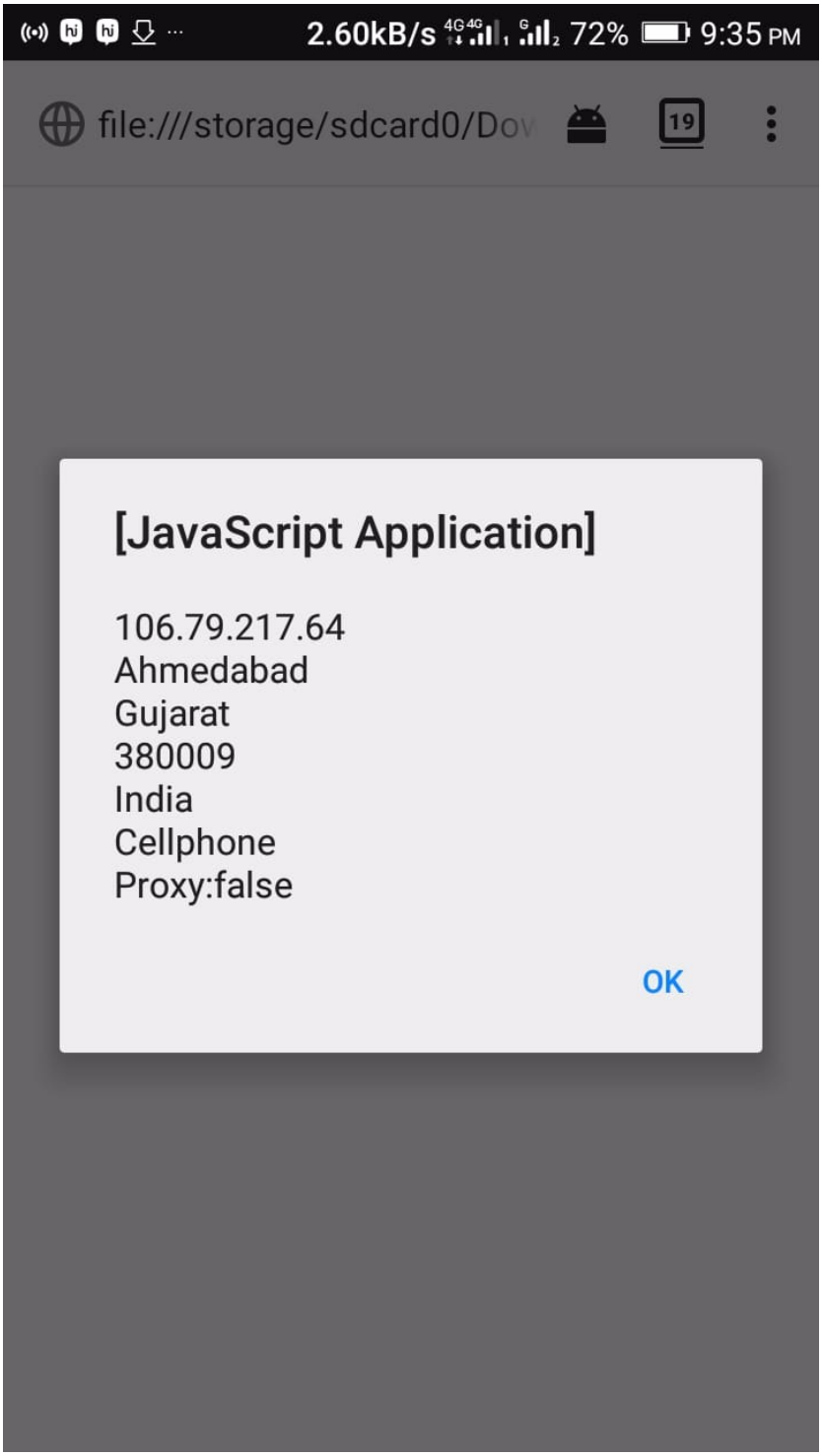

Figure 10 Result of Network module's process

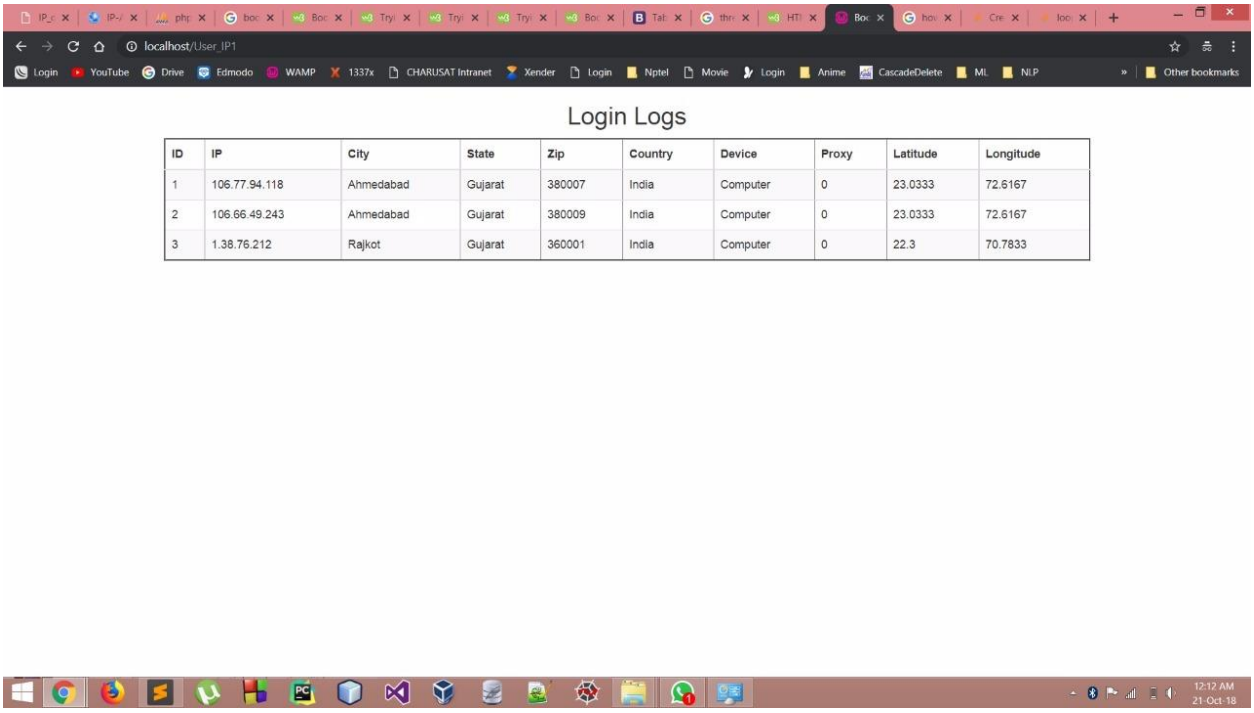

Figure 11 Log UI of Network module at admin panel

Login Logs

ID	IP	City	State	Zip	Country	Device	Proxy	Latitude	Longitude
1	106.77.94.118	Ahmedabad	Gujarat	380007	India	Computer	0	23.0333	72.6167
2	106.66.49.243	Ahmedabad	Gujarat	380009	India	Computer	0	23.0333	72.6167
3	1.38.76.212	Rajkot	Gujarat	360001	India	Computer	0	22.3	70.7833

Figure 12 Enlarged Log-table of Network module

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<body>
<?php
    $login=mysqli_connect('localhost','root','','user_ip_details');
    if($login==null)
        echo'Error connecting to teh Database.';
    $sql="select * from user_details";
    if(mysqli_query($login,$sql))
    { $result=mysqli_query($login,$sql); ?>
        <div class="container">
            <center><h2>Login Logs</h2></center>
        </div>
        <div class="container">
            <table class='table table-striped' border='2' padding='5px'>
                <thead>
                    <tr>
```

Figure 13 Code Snippet of Network-UI

MODULE 4: FRONTEND

INTRODUCTION

UI platform for our application. Web-based UI working jQuery, AJAX and JSON

WORKING & TECHNOLOGY/TOOLS USED

Why jQuery?

There are lots of other JavaScript frameworks out there, but jQuery seems to be the most popular, and also the most extendable among all others.

It can really simplify javaScript programming for your web application.

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. And as we have used much of HTML and Ajax concepts in our application it is more relevant to use jQuery.

Why AJAX?

Our application of chatbot is a pop-up type application which is to be executed along with the running web page and not as a new page action, i.e. without requesting for a redirected page, we need to run our application in parallel, and AJAX is best suited for serving the purpose.

Why use JSON?

Since the JSON format is text only, it can easily be sent to and from a server, and used as a data format by any programming language.

JavaScript has a built in function to convert a string, written in JSON format, into native JavaScript objects:

JSON.parse()

So, if you receive data from a server, in JSON format, you can use it like any other JavaScript object.

And as our project is a on server application, and as we have used javaScript and we need to communicate between our python code on server side and javaScript programs on client side we need to exchange information between both which can be achieved by using JSON.

FIGURES

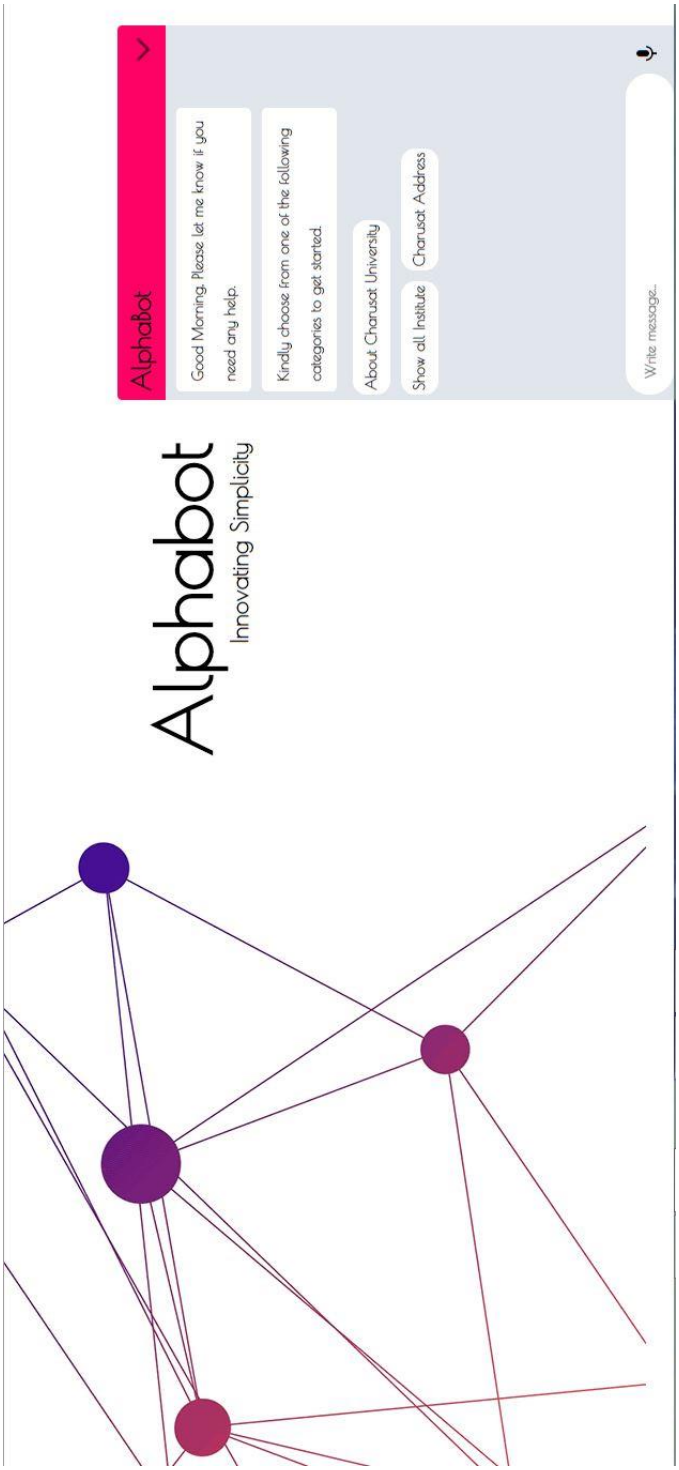

Figure 14 Application UI-I

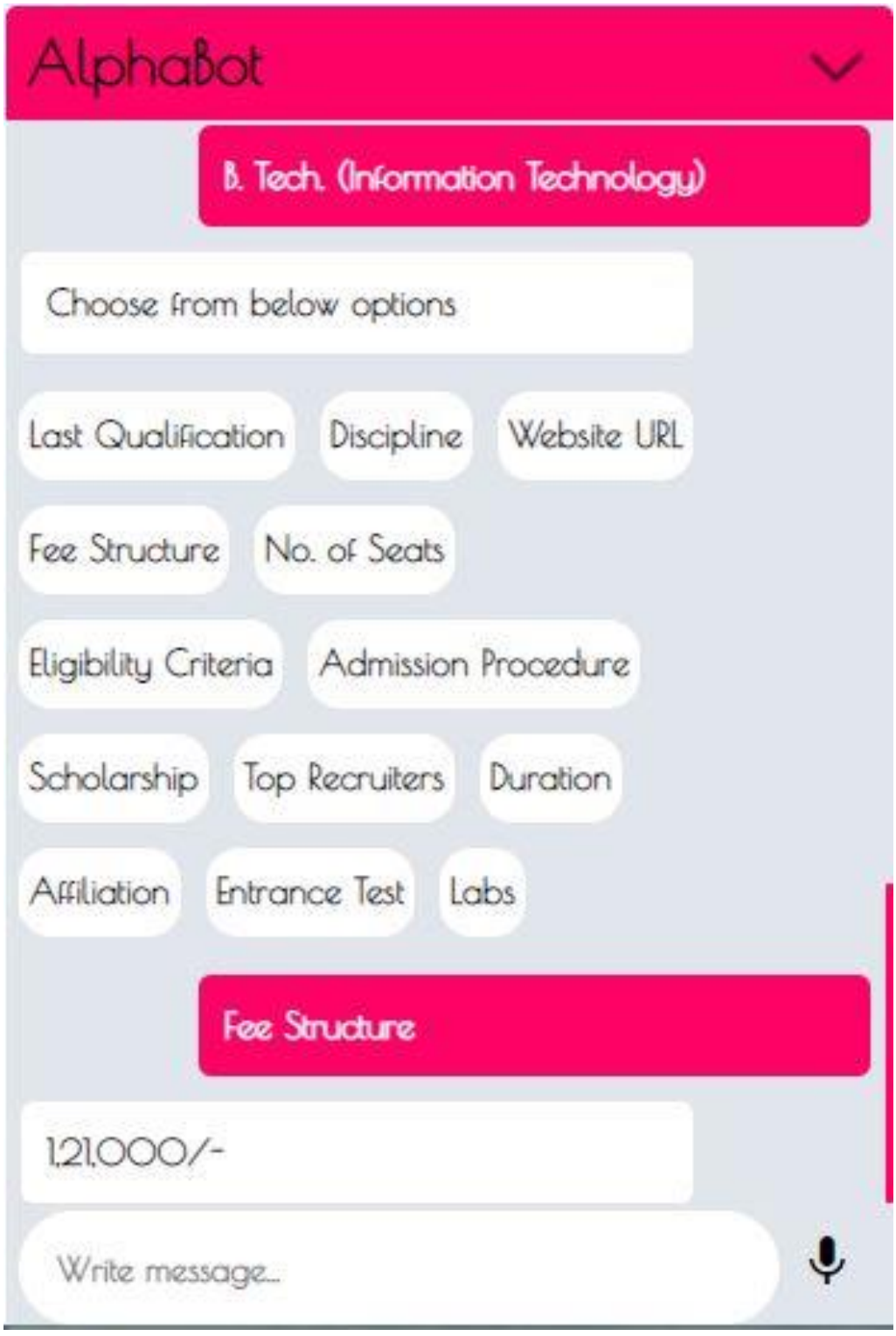

Figure 15 Application pop-up chat-box with response to a query-I

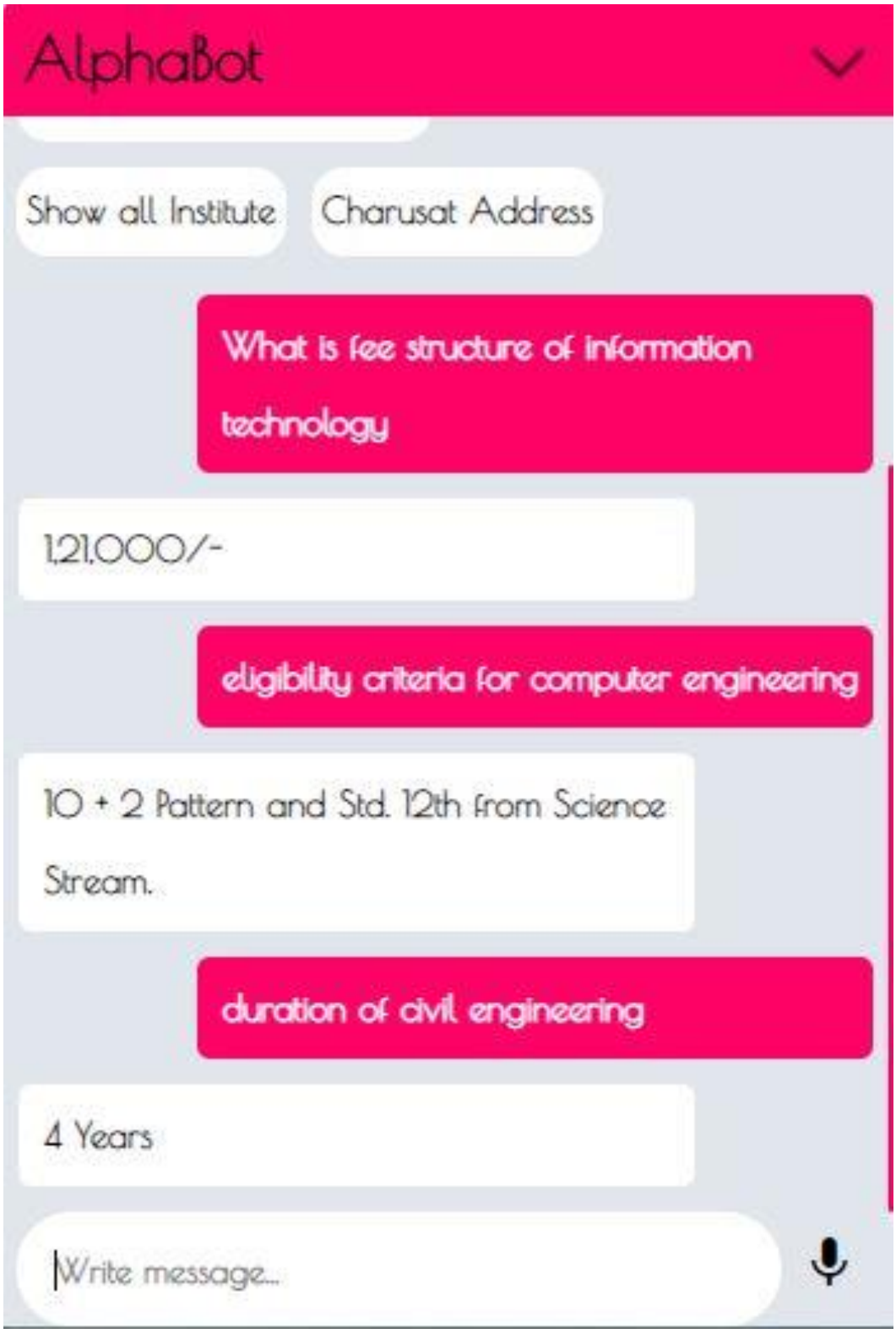

Figure 16 Application pop-up chat-box with response to a query-II

FUTURE EXTENTIONS

To make the web application responsive to all interfaces including mobile phone supportive.

To make an Admin panel interface for keeping the logs of sessions with users and to study and analyze the queries and provide possible improvements to datasets and also to manage and carry our surveys.

MODULE 5: SPEECH RECOGNITION

INTRODUCTION

This module is an extension or a plug-in to the already existing and working model. The board can work by itself along with its other modules. Though this module is not a necessity but it is a significant enhancement to the bot for its accessibility.

The speech recognition can help the people who are not capable of typing or cannot have the accessibility to the screen temporarily or permanently for example a blind person or a person who is engaged in some work which are keeping his hands busy.

Thus, the text to speech and speech to text conversion is an added enhancement to the application.

WORKING

As the user chooses to opt for speech recognition by pressing the mic icon on the UI, the pop-up, this module starts working.

For asking the query the user presses the mic icon and speaks the query to the application. This module converts the speech to text and sends the data to backend module which then processes it like a typed text further.

Now as the response to the query returns from backend, it is converted by this model again from text to speech

TECHNOLOGY/TOOLS USED

- Python and its libraries
    - Pyaudio
    - Pyttsx3
    - Speech Recognition
    - Google-Speech-API

FUTURE EXTENTIONS

- Once the user comes across our site we are bound to add the information to our database to make the user recognition easier and for the other reasons too.
- But at the same time the IP we have used is Live which means it is dynamic which is bound to keep on changing.
- So we are planning to overcome this problem by using the functionality of cookies.
- Generally, the users may not be bound to follow any set of rules in querying to the Chatbot. So we may come across the situation where we may need to determine the similarity of the words used in the query and the data present in the database.
- So, based on the **Word Net** library available in **Python** we plan to add few of the functionalities that we may integrate in near future into our present model.
- We will implement Inverse Document Frequency in order to reduce the computation of finding probability of all the queries of database.
- We will also add the module which will correct the spelling mistakes in the user query so the user will get appropriate data.
- We will keep the track of each and every user's conversation.
- We will also add special mechanism to solve answered queries of user.

REFERENCES

- https://www.tutorialspoint.com/artificial_intelligence_with_python/artificial_intelligence_with_python_speech_recognition.htm
- https://www.lfd.uci.edu/~gohlke/pythonlibs/#pyaudio
- https://realpython.com/python-speech-recognition/
- https://stackoverflow.com/questions/12239080/getting-started-with-speech-recognition-and-python
- https://pyttsx3.readthedocs.io/en/latest/
- http://portaudio.com/docs/v19-doxydocs/tutorial_start.html