JAVA PROJECT

CHATBOT

(ASSISTANT)

MADE BY: -

Chirag Agarwal

CSE-A(GROUP-1)

189301025

Introduction: -

At the most basic level, a chatbot is a computer program that simulates and processes human conversation (either written or spoken), allowing humans to interact with digital devices as if they were communicating with a real person. Chatbots can be as simple as rudimentary programs that answer a simple query with a single-line response, or as sophisticated as digital assistants that learn and evolve to deliver increasing levels of personalization as they gather and process information

Software Requirements: -

This application requires Java Development Kit (JDK) version 1.8 or above and an editor (in my case, ATOM) to write and save the code.

Explanation: -

This project is a GUI based application made using Swing framework and AWT (Abstract Window Toolkit). The GUI provides a platform to interact with the code and get answers to your questions, take part in conversations and search the web through it.

This project uses several concepts in java such as: -

1. Packages

2. Different addressing modes

3. Inheritance

4. Aggregation

5. Exception Handling

6. Sleep method (To implement Multithreading)

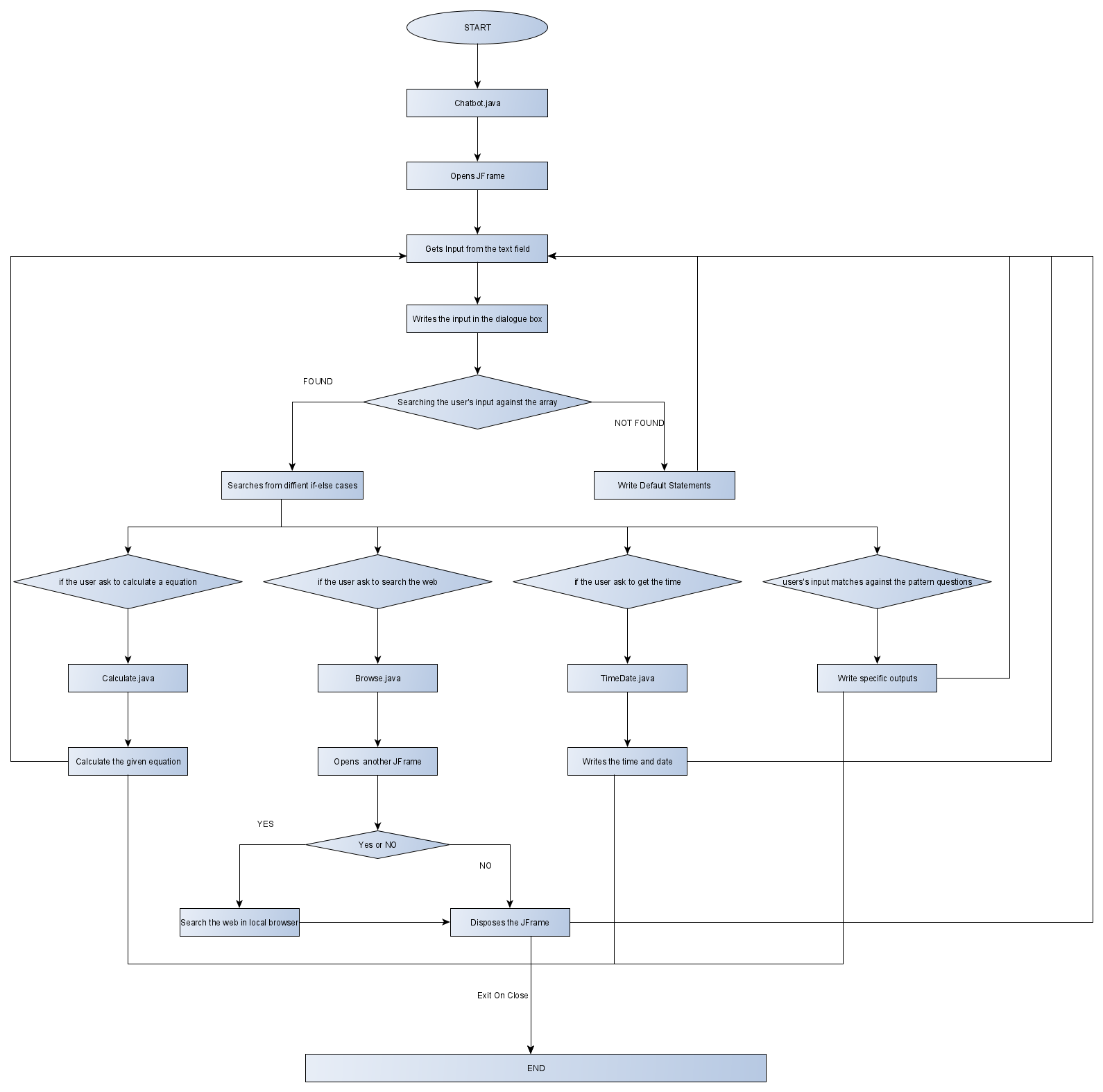
7. 2-D Jagged Array

8. Method Overriding

9. Interfaces (importing from the library)

I have used the KeyListener and ActionListener interfaces in the following code to detect activities in the text area and buttons respectively. Every class but the class Chatbot (which has the main() method) is saved in a package ‘chiragagarwal’. The code has a 2-D string jagged array which contains the limited patterns of users input and their respective responses. At the last index of the array, default outputs are stored if the inputs don’t match. The overridden method KeyPressed() gets the input when ‘Enter’ key is pressed and search the inputs against the array. The many if-else blocks display the respective outputs in the dialogue box of the GUI. Class calculates has a array list and push() – pop() functions to calculate any infix equation. Class TimeDate returns the date and time of that moment. Class Browse opens another frame which contains two buttons which decides based on user’s input whether to open the browser and search the web. Class Browse uses ActionPerformed() method of the ActionListener Class for searching web.

FLOW CHART: -



CODE: -

1. ChatBot.java

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JTextArea;

import javax.swing.JScrollPane;

import java.awt.Color;

import java.awt.event.KeyListener;

import java.awt.event.KeyEvent;

import java.lang.Math;

import java.lang.String;

import java.lang.System;

import chiragagarwal.Calculate;

import chiragagarwal.TimeDate;

import chiragagarwal.Browse;

public class ChatBot extends JFrame implements KeyListener{

Calculate cal = new Calculate();

TimeDate td = new TimeDate();

JPanel p=new JPanel();

JTextArea dialog=new JTextArea(20,50);

JTextArea input=new JTextArea(3,50);

JScrollPane scroll=new JScrollPane(

dialog,

JScrollPane.VERTICAL\_SCROLLBAR\_AS\_NEEDED,

JScrollPane.HORIZONTAL\_SCROLLBAR\_AS\_NEEDED

);

String name = "";

String[][] chatBot={

//standard greetings

{"hi","hello","hola","ola","hey"},

{"hi","hello","hey","Hi, I am CHATBOT"},

//question greetings

{"how are you","how r you","how r u","how are u","how do u do"},

{"good","doing well"},

//standard questions

{"who are you","what is your name"},

{"My name is CHATBOT. Nice to meet you!"},

//cheeky answers

{"how much marks should i get for this project"},

{"This is an amazing project, you should get full marks!"},

{"are you alive","do you exist"},

{"I am Virtual Bot.It says it in my name."},

{"how old are you","what is your age"},

{"I was created yesterday in Chirag's room, So im still pretty young "},

{"Who owns you","who created you"},

{"I was created by Chirag Agarwal(A full time sleeper and noob programmer)"},

//help

{"help","what can you do","list the things you can do for me","how can you help me"},

{"In addition to have a conversation, I can calculate any equation, can tell the current time and date, and do a web search for you."},

//time and date

{"what time is it","time","can you tell me the time","what date it is today","date"},

{"Current Date and Time: "},

//Online searching

{"what","how","where","when","who","why","whom","by whom","are","search"},

{"Getting Confirmation!"},

//name

{"my name is"},

{"Hi, "},

//asking name

{"what is my name","did i tell you my name?"},

{"Your name is "},

//yes

{"yes"},

{"no","NO","NO!!!!!!!"},

//calculate

{"calculate","cal","Calculate","Cal"},

{"="},

//default

{"Can't reach CHATBOT at the moment","I didn't quite get you!",

"(CHATBOT is unavailable, due to Issues)"}

};

public static void main(String[] args){

new ChatBot();

}

public ChatBot(){

super("Chat Bot");

setSize(600,420);

setResizable(false);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

dialog.setEditable(false);

input.addKeyListener(this);

p.add(scroll);

p.add(input);

p.setBackground(new Color(255,0,0));

add(p);

addText("\n-->CHATBOT:\tHow can I help you?\n");

setVisible(true);

}

public void keyPressed(KeyEvent e){

if(e.getKeyCode()==KeyEvent.VK\_ENTER){

input.setEditable(false);

//-----Take quote-----------

String quote=input.getText();

input.setText("");

addText("-->You:\t"+quote);

quote = quote.trim();

while(

quote.charAt(quote.length()-1)=='!' ||

quote.charAt(quote.length()-1)=='.' ||

quote.charAt(quote.length()-1)=='?'

){

quote=quote.substring(0,quote.length()-1);

}

quote = quote.trim();

byte response=0;

/\*

0:we're searching through chatBot[][] for matches

1:we didn't find anything

2:we did find something

\*/

//-----check for matches----

int j=0;//which group we're checking

while(response==0){

if(inArray(quote.toLowerCase(),chatBot[j\*2])){

response=2;

if((j\*2 != chatBot.length-3) && (j\*2 != chatBot.length-9) && (j\*2 != chatBot.length-11) && (j\*2 != chatBot.length-13))

{

int r=(int)Math.floor(Math.random()\*chatBot[(j\*2)+1].length);

if(j\*2 != chatBot.length-7){

addText("\n-->CHATBOT:\t"+chatBot[(j\*2)+1][r]);

}

else{

addText("\n-->CHATBOT:\t"+chatBot[(j\*2)+1][r]+ " "+ name);

}

}

else if(j\*2 == chatBot.length-3)

{

String expression[] = quote.split(" ");

int eval = cal.evaluate(expression[1]);

addText("\n-->CHATBOT:\t"+chatBot[(j\*2)+1][0] + " "+ eval);

}

else if(j\*2 == chatBot.length-9){

String expression[] = quote.split(" ");

try{

name = expression[3];

}

catch(ArrayIndexOutOfBoundsException f)

{

addText("\n-->CHATBOT:\tNo name written!");

}

addText("\n-->CHATBOT:\t"+chatBot[(j\*2)+1][0] + " "+ name);

}

else if(j\*2 == chatBot.length-13){

addText("\n-->CHATBOT:\t"+chatBot[(j\*2)+1][0] + td.getCurrentTimeDate());

}

else if(j\*2 == chatBot.length-11){

addText("\n-->CHATBOT:\t"+chatBot[(j\*2)+1][0]);

try{

Thread.sleep(2000);

}

catch(InterruptedException Ie)

{

System.out.println(Ie);

}

Browse br = new Browse();

quote = quote.replaceAll(" ","%20");

br.link = br.link + quote;

}

}

j++;

if(j\*2==chatBot.length-1 && response==0){

response=1;

}

}

//-----default--------------

if(response==1){

int r=(int)Math.floor(Math.random()\*chatBot[chatBot.length-1].length);

addText("\n-->CHATBOT:\t"+chatBot[chatBot.length-1][r]);

}

addText("\n");

}

}

public void keyReleased(KeyEvent e){

if(e.getKeyCode()==KeyEvent.VK\_ENTER){

input.setEditable(true);

}

}

public void keyTyped(KeyEvent e){}

public void addText(String str){

dialog.setText(dialog.getText()+str);

}

public boolean inArray(String in,String[] str){

boolean match=false;

for(int i=0;i<str.length;i++){

if(str[i].matches(in) || in.startsWith(str[i])){

match=true;

}

}

return match;

}

}

1. Inside Package chiragagarwal-
   * 1. TimeDate.java

package chiragagarwal;

import java.util.Date;

import java.text.SimpleDateFormat;

import java.util.TimeZone;

public class TimeDate{

public String getCurrentTimeDate() {

SimpleDateFormat dateTime = new SimpleDateFormat("yyyy-MMM-dd hh:mm:ss aa");

dateTime.setTimeZone(TimeZone.getDefault());

String TimeDate = dateTime.format(new Date());

return TimeDate;

}

}

* + 1. Calculate.java

package chiragagarwal;

import java.util.Stack;

public class Calculate {

public int evaluate(String expression){

//Stack for Numbers

Stack<Integer> numbers = new Stack<>();

//Stack for operators

Stack<Character> operations = new Stack<>();

for(int i=0; i<expression.length();i++) {

char c = expression.charAt(i);

//check if it is number

if(Character.isDigit(c)){

//Entry is Digit, it could be greater than one digit number

int num = 0;

while (Character.isDigit(c)) {

num = num\*10 + (c-'0');

i++;

if(i < expression.length())

c = expression.charAt(i);

else

break;

}

i--;

//push it into stack

numbers.push(num);

}else if(c=='('){

//push it to operators stack

operations.push(c);

}

//Closed brace, evaluate the entire brace

else if(c==')') {

while(operations.peek()!='('){

int output = performOperation(numbers, operations);

//push it back to stack

numbers.push(output);

}

operations.pop();

}

// current character is operator

else if(isOperator(c)){

//1. If current operator has higher precedence than operator on top of the stack,

//the current operator can be placed in stack

// 2. else keep popping operator from stack and perform the operation in numbers stack till

//either stack is not empty or current operator has higher precedence than operator on top of the stack

while(!operations.isEmpty() && precedence(c)<precedence(operations.peek())){

int output = performOperation(numbers, operations);

//push it back to stack

numbers.push(output);

}

//now push the current operator to stack

operations.push(c);

}

}

//If here means entire expression has been processed,

//Perform the remaining operations in stack to the numbers stack

while(!operations.isEmpty()){

int output = performOperation(numbers, operations);

//push it back to stack

numbers.push(output);

}

return numbers.pop();

}

static int precedence(char c){

switch (c){

case '+':

case '-':

return 1;

case '\*':

case '/':

return 2;

case '^':

return 3;

}

return -1;

}

public int performOperation(Stack<Integer> numbers, Stack<Character> operations) {

int a = numbers.pop();

int b = numbers.pop();

char operation = operations.pop();

switch (operation) {

case '+':

return a + b;

case '-':

return b - a;

case '\*':

return a \* b;

case '/':

if (a == 0)

throw new

UnsupportedOperationException("Cannot divide by zero");

return b / a;

}

return 0;

}

public boolean isOperator(char c){

return (c=='+'||c=='-'||c=='/'||c=='\*'||c=='^');

}

}

* + 1. Browse.java

package chiragagarwal;

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

import java.net.\*;

public class Browse {

JFrame f = new JFrame("Confirmation");

JPanel pan=new JPanel();

JButton b1 = new JButton("Yes");

JButton b2 = new JButton("No");

JLabel l = new JLabel("Do you want me to search in the web browser??");

public String link = "https://www.google.com/search?q=";

public Browse()

{

f.setSize(600, 200);

l.setBounds(20,20,160,20);

b1.setBounds (40,60,120,40);

b2.setBounds(40,120,120,40);

b1.addActionListener( new ActionListener(){

public void actionPerformed(ActionEvent e)

{

try{

f.dispose();

URI u = new URI(link);

Desktop d = Desktop.getDesktop();

d.browse(u);

}

catch(Exception f){

System.out.println(f);

}

}

});

b2.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

f.dispose();

}

});

pan.add(l);

pan.add(b1);

pan.add(b2);

f.add(pan);

pan.setBackground(new Color(255,0,0));

f.setVisible(true);

}

public void actionPerformed(ActionEvent e)

{

try{

f.dispose();

URI u = new URI(link);

Desktop d = Desktop.getDesktop();

d.browse(u);

}

catch(Exception f){

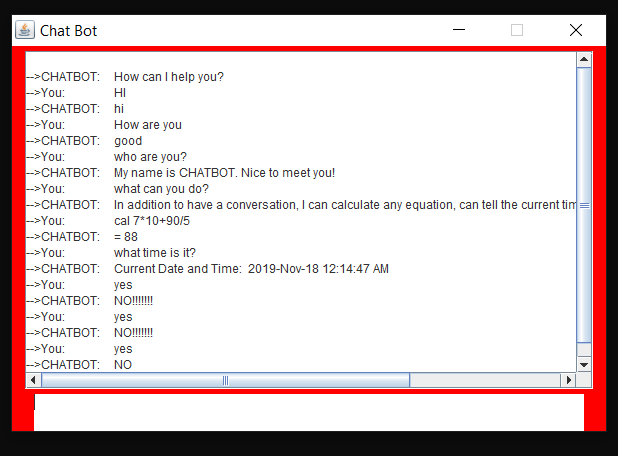
System.out.println(f);

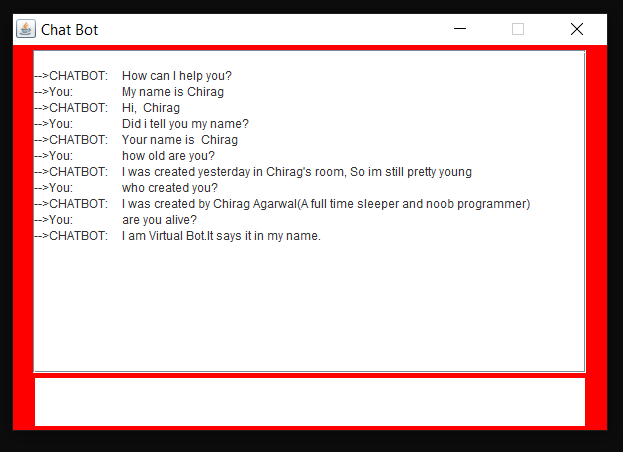
}

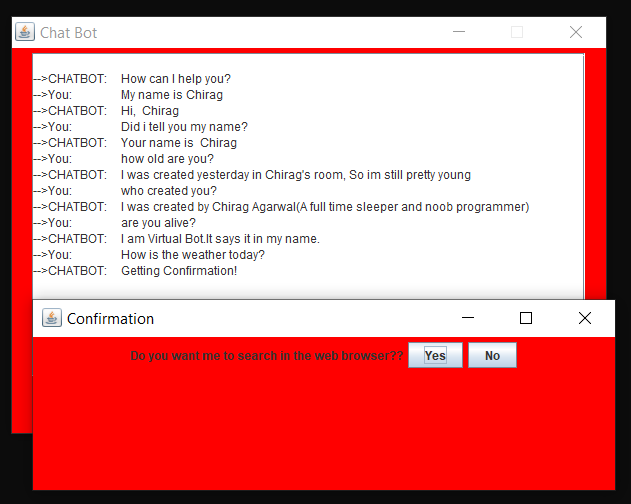
}

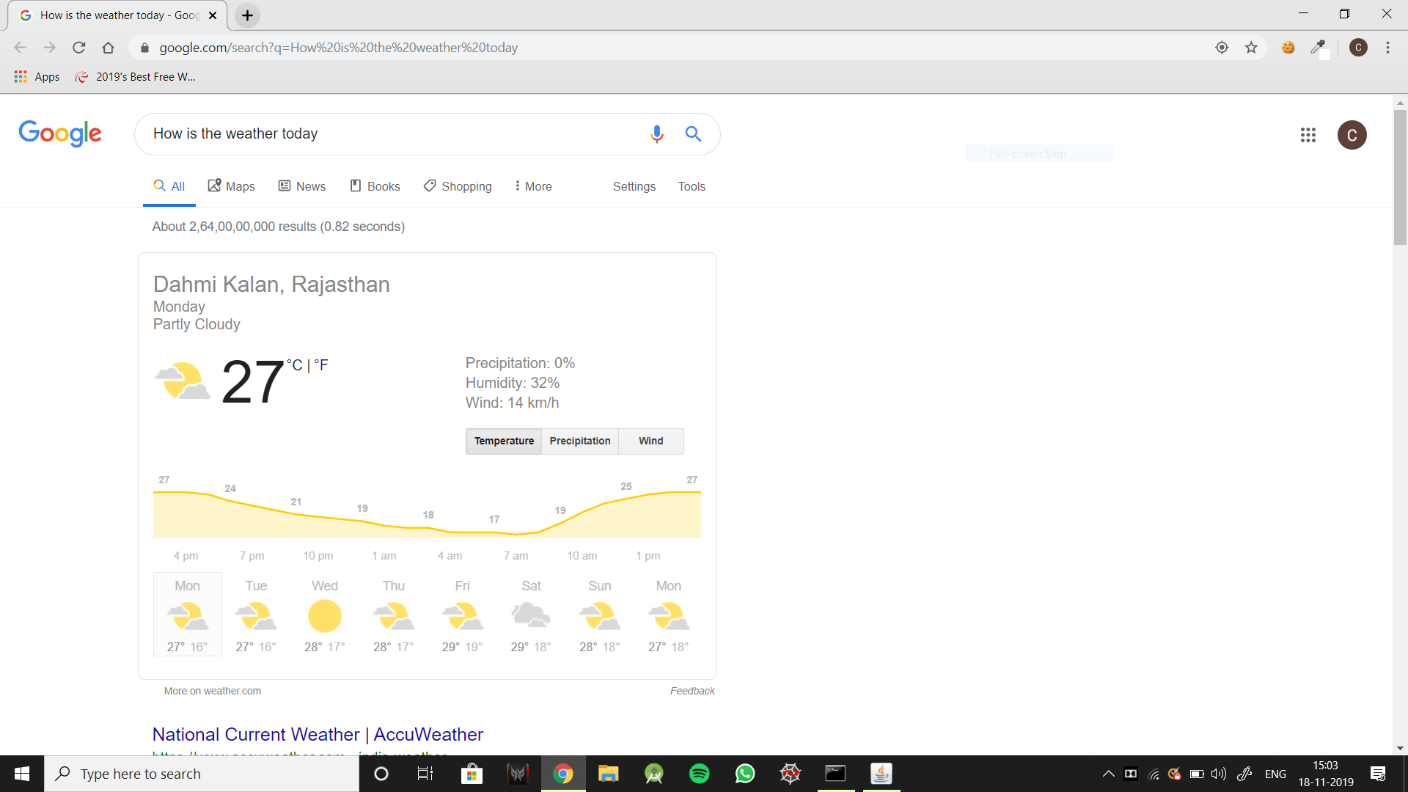
}

OUTPUT: -









CONCLUSION: -

This Project can therefore have a proper conversation with a person, can return time and date, can search web for your queries and can calculate any infix equation for you.

As a result of not using any deep learning algorithms, this project has some limitations. It can answer only certain set of questions and carry out only certain operations.