

**School of computer science and engineering**

**Name of the faculty : Nandini**

Programming assignment

**Course Title** : Artificial Intelligence

**Course code**: INT404

Github Link:- <https://github.com/rupeshyadav08/Artificial_Intelligence_project>

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial no** | **Name** | **Registration no** | **Roll no** |
| 1) | Rupesh yadav | 11803906 | 24 |
| 2) | P. Sai Kesav | 11805845 | 21 |
| 3) | Venkata Subba Reddy | 11801090 | 22 |
| 4) | Md Asad Ansari | 11815254 | 23 |

**METHODOLOGY SELECTED:-**

Our project is about Relationship identification system . In this project the machine will decide the family relationships . Mainly in this program we had imported the packages like pyttsx3,speech\_recognition , logpy etc …. Firstly we are going to explain the packages and why there are necessary .

1. Pyttsx3 :-

**pyttsx3** is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech.  
The pyttsx3 module supports two voices first is female and the second is male which is provided by “sapi5” for windows.

To install the pyttsx3 module, first of all, you have to open the terminal and write:-

**pip install pyttsx3**

1. Speech recognition:-

Speech recognition is the process of converting spoken words to text. Python supports many speech recognition engines and APIs, including Google Speech Engine, Google Cloud Speech API, Microsoft Bing Voice Recognition and IBM Speech to Text.

The Speech Recognition module depends on pyaudio, you have to install the pyaudio package too.

To install the pyttsx3 module, first of all, you have to open the terminal

and write:-

**pip install SpeechRecognition**

**pip install pyaudio**

1. LogPy :-

LogPy is an alternative for standard Python logging facilities, loosely based on Lisp’s log5. LogPy is based on KISS principles - therefore I wanted it to be as most transparent as possible.

The main difference when compared to stdlib’s logging is tag-based architecture. In logging, each log has assigned a certain level (be it debug, error, etc.). That’s all. LogPy, on the other hand, sports tags - you can attach short strings to each message. Tag can represent variety of things: severity level, module name, or some custom log categorization.

Secondly we are looking forward to explain about things that we wrote in prolog form :-

* father(X,Y) means X is Y’s father.
* mother(X,Y) means X is Y’s mother.
* female(X) means X is a female.
* male(X) means X is a male.
* husband(X,Y) means X is Y’s husband
* parent(X,Y) means X is Y’s parent.
* grandfather(X,Y) means X is Y’s grandfather.
* grandparent(X,Y) means X is Y’s grandparent.
* children(X,Y) means X is Y’s child.
* son(X,Y) means X is Y’s son.
* daughter(X,Y) means X is Y’s daughter.
* sibling(X,Y) means X and Y are siblings.
* parent(X,Y):-father(X,Y);mother(X,Y).
* grandparent(X,Y):-parent(X,M),parent(M,Y).
* husband(X,Y):-male(X),female(Y),children(M,X),children(M,Y).
* sibling(X,Y):-mother(M,X),mother(M,Y),father(F,X),father(F,Y),X=Y.
* grandfather(X,Y):-father(X,K),mother(K,Y).
* grandmother(X,Y):-mother(X,K),mother(K,Y).

**Some important rules of prolog writing :-**

* This symbol “ :- “ represents like **if** statement
* The symbol comma “ , “ represents like **And**
* The symbol underscore “ \_ “ represents **anonymous** variable
* The word “conde” is sometimes used it as **or** and sometimes as **and**
* Numbers can be floats or integers .
* List are defined in []

Moreover we add speech recognition to our project that’s why we are going to explain about the programming based on the speech recognition.

1)r = sr.Recognizer()

* we created a instance and that was stored in a variable r
* If you want to use audio input from microphones, [PyAudio](http://people.csail.mit.edu/hubert/pyaudio/#downloads) is also necessary.

2)with sr.Microphone() as source:

print("Speak Anything :")

audio = r.listen(source)

* This context of code is used to listen the user voice by using our microphone

1. text = r.recognize\_google(audio,language="en-US")

* Google Speech Recognition takes the language that you said in English.

4) except:

print("Sorry could not recognize your voice")

* These statements will evoke when there is any exceptions in the

recognition.

* If the background sound or noise is greater than the original voice then the exception will come into action.

**Project code:-**

import pyttsx3

import speech\_recognition as sr

from logpy import Relation,facts,run,var,conde,fact

class color:

Purple = '\033[95m'

Cyan = '\033[96m'

Darkcyan = '\033[36m'

Blue = '\033[94m'

Green = '\033[92m'

Yellow = '\033[93m'

Red = '\033[91m'

Bold = '\033[1m'

Underline = '\033[4m'

End = '\033[0m'

def split\_line(text,list2):

words = text.split()

for word in words:

list2.append(word)

print(color.Bold+"---------------------------------Welcome to Relation Predicting Program------------------------------"+color.End+"\n\n")

print("-----------------------------------------------------------------------------------------------------")

print(color.Red+"1)JOHN\n2)WILLIAM\n3)DAVID\n4)ADAM\n5)RAHUL\n6)BEN\n7)NEIL\n8)PETER\n"+color.End)

print(color.Green+"9)MEGAN\n10)EMMA\n11)NISHA\n12)JULIE\n13)OLIVIA\n14)RICHA\n15)LILY\n16)SONI"+color.End)

print(color.Bold+color.Blue+"\nAsk About The Realtive Of Above Mentioned Name\n"+color.End)

r = sr.Recognizer()

with sr.Microphone() as source:

print("Speak Anything :")

audio = r.listen(source)

try:

text = r.recognize\_google(audio,language="en-US")

print("You said : {}".format(text))

except:

print("Sorry could not recognize your voice")

engine = pyttsx3.init()

engine.say("Sorry could not recognize your voice")

engine.runAndWait()

text=text.lower()

list2=[]

name=""

Relat=""

split\_line(text,list2)

NameList=['john','megan',

'william','emma',

'david','olivia',

'adam','lily',

'peter','neil',

'richa','julie',

'nisha','soni',

'ben','rahul']

Realtionship=['parent','father','mother','sister','brother',

'sibling','partner','wife','husband','grandparent',

'grandmother','grandfather','uncle']

for i in list2:

if i in NameList:

name=i

for i in list2:

if i in Realtionship:

Relat=i

def parents(x,y):

return conde([Father(x,y)],[Mother(x,y)])

def grandparents(x,y):

z=var()

return conde((parents(x,z),parents(z,y)))

def grandfather(x,y):

z=var()

return conde((grandparents(x,y),Male(x)))

def grandmother(x,y):

z=var()

return conde((grandparents(x,y),Female(x)))

def uncle(m,y):

z=var()

x=var()

return conde((Father(x,z),Father(x,m),Father(z,y),Male(m)))

def sibling(x,y):

z=var()

return conde((parents(z,x),parents(z,y)))

def brother(x,y):

z=var()

return conde((parents(z,x),parents(z,y),Male(x)))

def sister(x,y):

z=var()

return conde((parents(z,x),parents(z,y),Female(x)))

def aunty(x,y):

z=var()

return conde((uncle(z,y),Couple(z,x),Female(x)))

def children(x,y):

return ((parents(x,y)))

def son(x,y):

return((parents(x,y),Male(y)))

def daughter(x,y):

return((parents(x,y),Female(y)))

Father=Relation()

Mother=Relation()

Male=Relation()

Female=Relation()

Couple=Relation()

facts(Couple,('john','megan'),

('william','emma'),

('david','olivia'),

('adam','lily'))

facts(Father,('john','william'),

('john','david'),

('john','adam'),

('william','rahul'),

('william','krish'),

('william','soni'),

('david','ben'),

('david','nisha'),

('david','julie'),

('david','neil'),

('david','peter'),

('adam','richa'),)

facts(Mother,('megan','william'),

('megan','david'),

('megan','adam'),

('emma','chris'),

('emma','krish'),

('emma','stephaine'),

('olivia','wayne'),

('olivia','nisha'),

('olivia','julie'),

( 'olivia','neil'),

('olivia','peter'),

('lily','sophia'),

('lily','richa'))

fact(Male,'john')

fact(Male,'william')

fact(Male,'david')

fact(Male,'adam')

fact(Male,'rahul')

fact(Male,'ben')

fact(Male,'neil')

fact(Male,'peter')

fact(Female,'megan')

fact(Female,'emma')

fact(Female,'nisha')

fact(Female,'julie')

fact(Female,'olivia')

fact(Female,'richa')

fact(Female,'lily')

fact(Female,'soni')

x=var()

y=var()

z=var()

if name=="":

quit()

FakeName=name.upper()

name=name.lower()

if name in NameList:

Relat=Relat.lower()

if Relat=='parent' or Relat=='parents' or Relat=="father" or Relat=="mother" :

option=Relat

if option=='parent'or option=='parents':

out=(run(0,x,parents(x,name)))

elif option=='mother':

out=(run(0,x,Mother(x,name)))

else:

out=(run(1,x,Father(x,name)))

list1=list(out)

if len(out)==0:

print(color.Cyan+color.Bold+color.Underline+FakeName+color.End,color.Green+"Parent Details Not In DataBase---SORRY"+color.End)

elif Relat=='sibling' or Relat=='sister' or Relat=='brother' or Relat=='siblings':

option=Relat

if option=='sibling' or option=='siblings':

out=(run(0,x,sibling(x,name)))

elif option=='sister':

out=(run(0,x,sister(x,name)))

else:

out=(run(0,x,brother(x,name)))

list1=list(out)

if name in list1:

list1.remove(name)

if len(list1)==0:

print(color.Cyan+color.Bold+color.Underline+FakeName+color.End,color.Green+"Does not have Entered type sibling ---SORRY"+color.End)

elif Relat=='grandparent' or Relat=='grandmother' or Relat=='grandfather' or Relat=='grandparents':

option=Relat

if option=='grandparent':

out=(run(0,x,grandparents(x,name)))

elif option=='grandfather':

out=(run(1,x,grandfather(x,name)))

else:

out=(run(1,x,grandmother(x,name)))

list1=list(out)

if name in list1:

list1.remove(name)

if len(list1)==0:

print(color.Cyan+color.Bold+color.Underline+FakeName+color.End,color.Green+"Does not have GrandParent data ---SORRY"+color.End)

elif Relat=='uncle':

out1=run(1,x,Father(x,name))

out=run(0,x,uncle(x,name))

a=out1[0]

list1=list(out)

if a in list1:

list1.remove(a)

elif Relat=='aunty':

out=(run(0,x,aunty(x,name)))

out1=run(0,x,Mother(x,name))

list1=list(out)

a=out1[0]

if a in list1:

list1.remove(a)

elif Relat=='children' or Relat=="child":

print(run(0,x,children(name,x)))

elif Relat=='couple' or Relat=='husband' or Relat=='wife':

print(run(1,x,Couple(x,name)))

else:

print(color.Red+"The Relation You Is Wrong Or May Not Be In Database"+color.End)

else:

print(color.Blue+"The Name"+color.End,color.Red+color.Underline+FakeName+color.End,color.Blue+"You Have Entered Is Not In The DataBase"+color.End)

engine = pyttsx3.init()

engine.say("You Have Entered Is Not In The DataBase")

engine.runAndWait()

ans=""

for i in list1:

ans=ans+" "+i

print(color.Cyan+""+ans+color.End)

engine = pyttsx3.init()

engine.say(ans+"...."+"is... "+"....."+Relat+"....."+"of..."+name)

engine.runAndWait() **TEST CASES :-**

**NOTE:-** GIVE MICROPHONE PERMISSION BEFORE RUNNING THE PROGRAM.

**VIDEO :-**

**Contribution:-**

Everyone have contributed in this project , We have done this project sitting together.