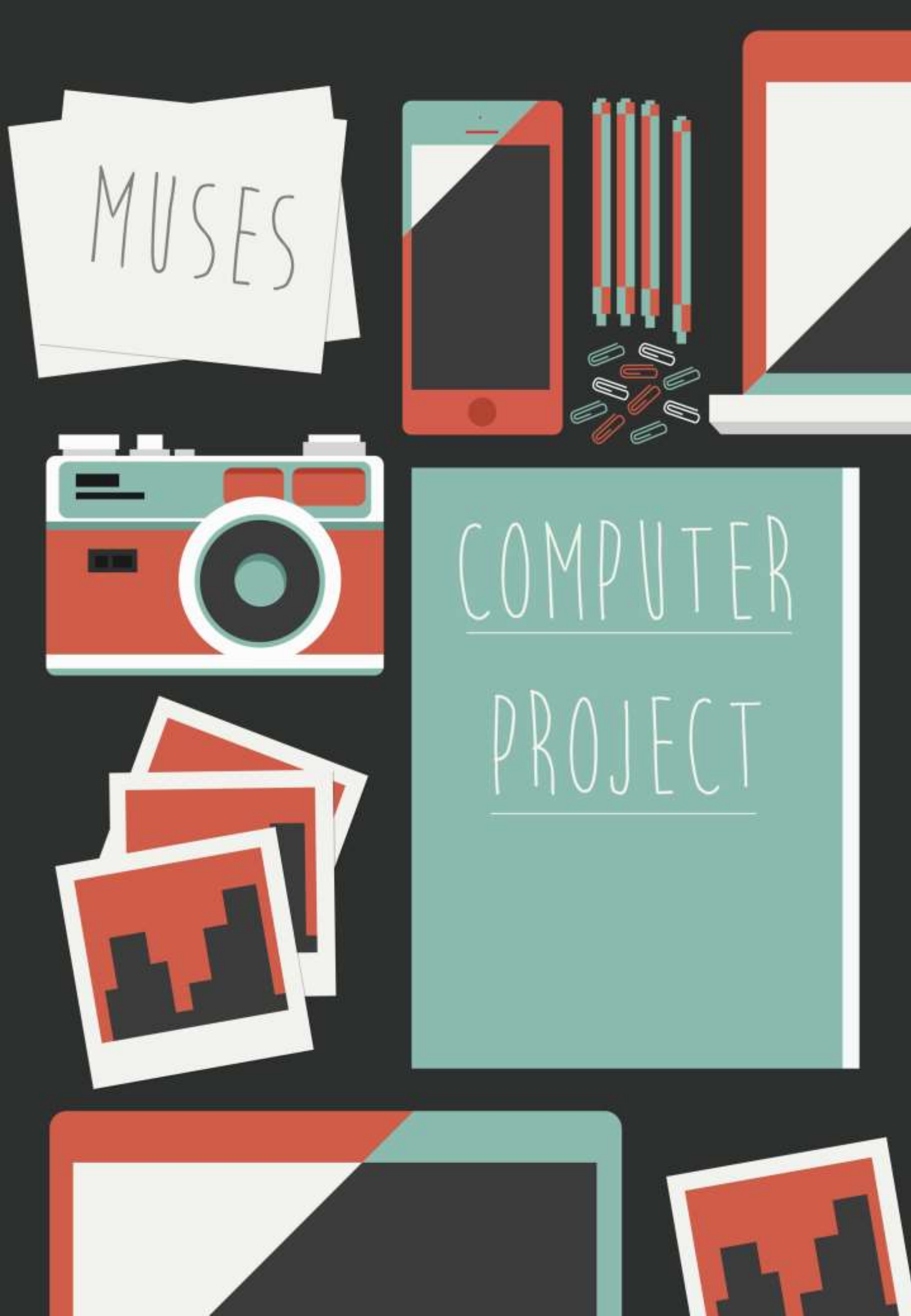
MUSES

COMPUTER PROJECT



INDEX

ACKNOWLEDGEMENTS

I WOULD LIKE TO EXPRESS MY GREATEST GRATITUDE TO EVERYONE WHO SUPPORTED ME IN COMPLETING THIS PROJECT SUCCESSFULLY.

I AM VERY GRATEFUL TO MY TEACHER MRS. REMYA, FOR HER MORAL SUPPORT AND GUIDANCE DURING THE EARLY STAGES OF CONCEPTUAL INCEPTION AND THROUGHOUT THE COURSE OF THIS PROJECT. SHE WAS VERY KIND AND PATIENT AND HELPED ME WITH ALL HER EFFORTS. I THANK HER FOR HER SINCERE SUPPORTS.

I ALSO WISH TO THANK MY PARENTS FOR THEIR ENCOURAGEMENT. THEIR UNFAILING SUPPORT AND GUIDANCE WAS THERE ALWAYS, DESPITE THEIR BUSY SCHEDULE. THEIR ADVICE AND IDEAS HELPED ME TO MAKE THIS PROJECT A SUCCESS.

I WOULD ALSO LIKE TO THANK GOD FOR THE WISDOM AND KNOWLEDGE THAT HE BESTOWED UPON ME AND FOR MAKING ALL THINGS POSSIBLE.

AIM

THE MAIN AIM IS TO CREATE A SIMPLE, INTERACTIVE AND USER FRIENDLY PROGRAM FOR ALL USERS OF "MUSES".

THIS PROGRAM ALLOW USERS TO ASK MUSES QUESTIONS AND MUSES WILL RETURN THEM WITH ANSWERS.

USER DOCUMENTATION

API'S USED

- BEAUTIFULSOUP4 (4.3.2)
- PYOWM (1.2.0)
- REQUESTS (2.3.0)
- ROTTENTOMATOES (1.1)
- WIKIPEDIA (1.3.1)
- WORDNIK (2.1.2)

MODULES USED

- IMPORT SYS
- IMPORT STRING
- IMPORT PYOWM
- IMPORT PICKLE AS P
- FROM TKINTER IMPORT *
- IMPORT RE
- IMPORT RANDOM
- FROM ROTTENTOMATOES IMPORT RT
- FROM WORDNIK IMPORT *
- IMPORT WIKIPEDIA
- IMPORT MATH AS M

PROGRAM ALGORITHM

```
START

STEP 1:  OPEN LOGIN SCREEN
                SIGN IN / SIGN UP

STEP 2:

IF SIGN UP CLICKED
        SHOW_SIGN_UP()
        BACK TO LOGIN

IF SIGN IN CLICKED
        SIGN_IN_CHECK()
        IF USER EXIST
                GOTO STEP 3
```

STEP 3:

```
    OPEN_MUSES(USERNAME)
    IF ASK BUTTON CLICKED
        GOTO STEP 4
```

STEP 4:

```
    CHECK_MUSES(QUERY)
```

ALGORITHM OF CHECK_MUSES()

```
STEP 1:
        F=STRIP_SEARCH_CODE(QUERY)
        THIS IS WHERE THE QUERY IS SPLIT INTO A LIST
         DISCARDING THE UNWANTED WORDS
STEP 2:
        IF F='MATH'
                MATH_CALCULATE(F)
        IF F CONTAINS STRING 'JOKE'
                JOKE()
```

```
IF F=='S'
        IF F CONTAINS STRING 'WORD'
                GET_MEANING()
        ELSE
                SEARCH_WIKI()
IF F=='T'
        GET_TEMP()
IF F=='M'
        MOVIE_RAT()
```

ALGORITHM OF HISTORY FEATHER

```
STEP 1: (FOR EVERY QUERY ASKED)
        HISTORY=QUERY
        H = OPEN('HISTORY.DAT','WB')
        PICKLE.DUMP(HISTORY,H)
        H.CLOSE()
STEP 2:
        IF QUERY CONTAINS "HISTORY'
        DISPLAY_ANSWER()
        IF "DELETE ALL?" BUTTON CLICKED:
                GOTO STEP 3
STEP 3:
        DEL_HISTORY()
```

ALGORITHM OF STRIP_SEARCH_CODE()

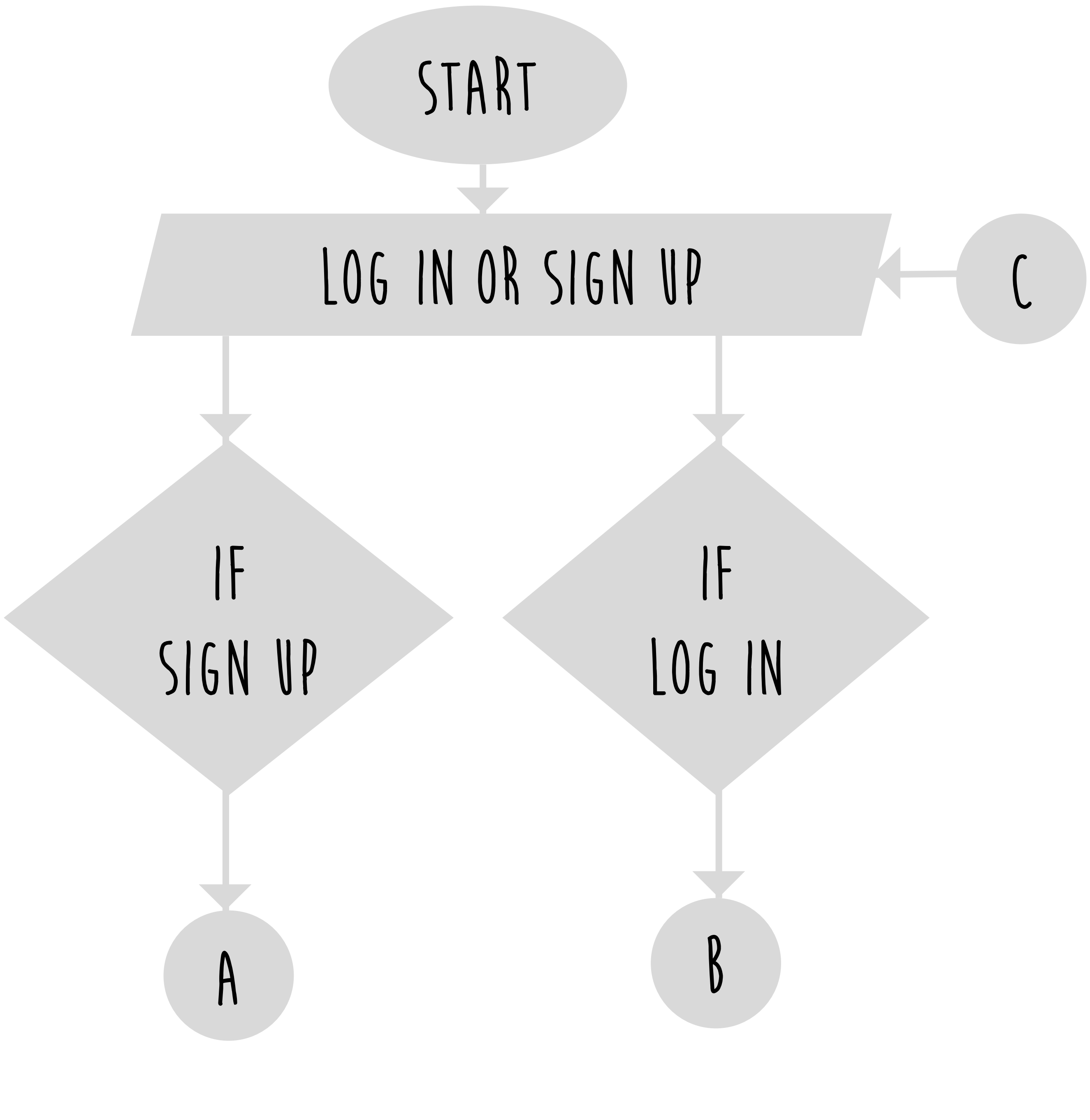
```
STEP 1:
        A=[ ]
        B=GET_CODE(STR)
        A.APPEND(B)
STEP 2:
        C=STRIP_SEARCH(STR)
        A.EXTEND(C)
STEP 3:
        RETURN A
```

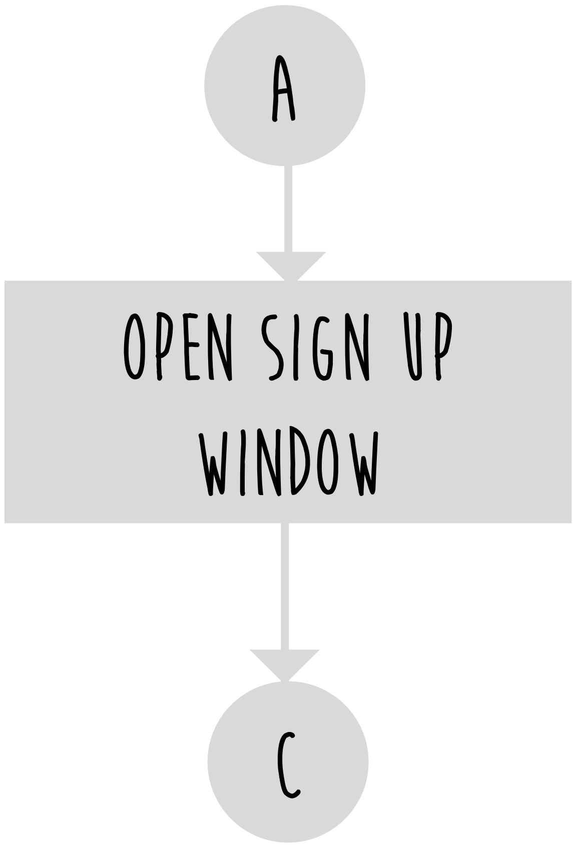
ALGORITHM OF DISPLAY_ANSWER()

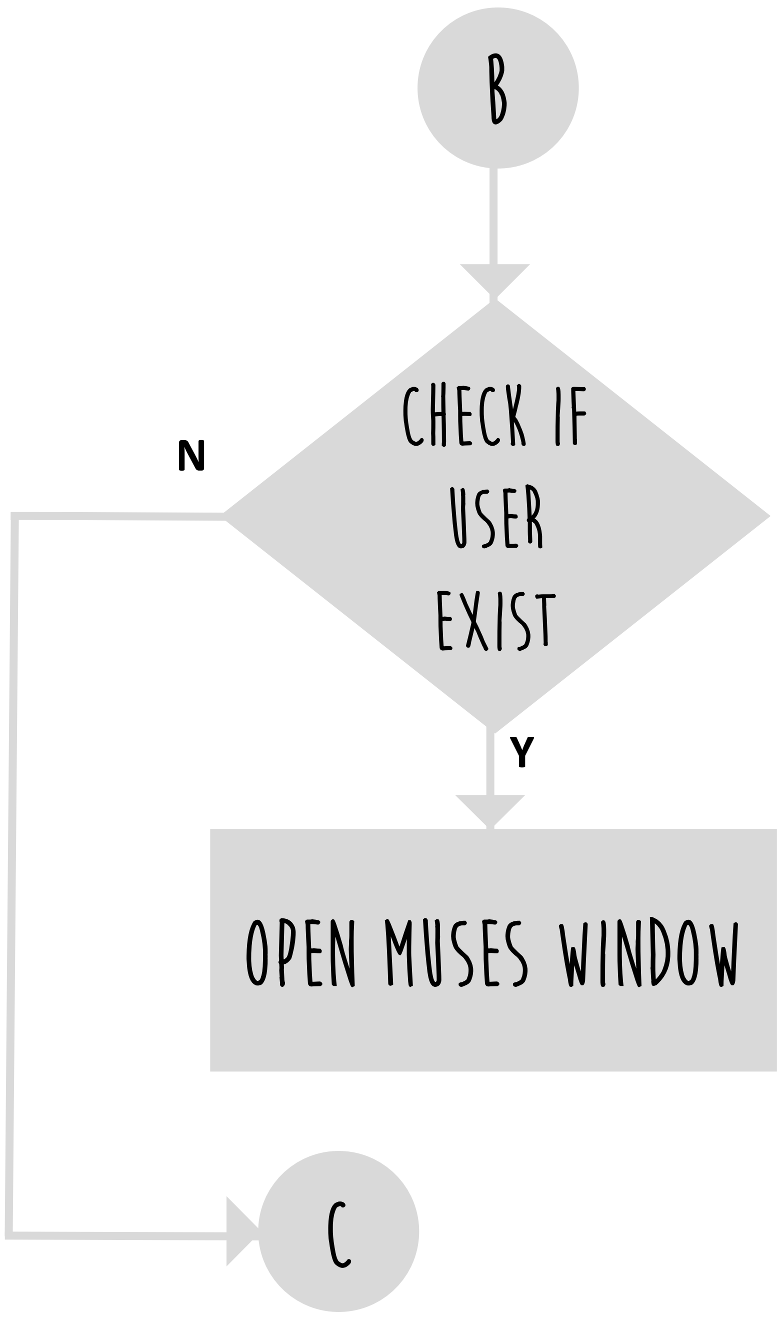
```
STEP 1:
        QUESTION = ASK_MUSES.GET()
        (GETS THR QUERY FROM MUSES WINDOW)
        ANSWER = CHECK_MUSES(QUESTION)
        RETURNS ANSWER IN A LIST
STEP 2:
        IF ANSWER[CODE]='S':
                SHOW SEARCH ANSWER BOX
        ELIF ANSWER[CDE]='T':
                SHOW TEMPERATURE ANSWER BOX
        ELIF ANSWER[CODE]='M':
                SHOW MOVIE ANSWER BOX
```

```
ELIF ANSWER CONTAIN 'HISTORY; :
        SHOW HISTORY WINDOW
ELSE:
        SHOW DEFAULT ANSWER WINDOW
```

FLOWCHART




A
OPEN SIGN UP WINDOW
C


B
CHECK IF USER EXIST
N
Y
OPEN MUSES WINDOW
C

Program Code

```
# -*- coding: utf-8 -*-
import sys
import string
import pyowm
import pickle as p
from Tkinter import *
import re
import random
from rottentomatoes import RT
from wordnik import *
import wikipedia
import math as m

pyowm_connect = pyowm.OWM('d96fd08f487348c0277372130a6ab12e')


#GET THE HISTORY
history = []
h = open('history.dat','rb')
history = p.load(h)
h.close()
print history

#GET ALL THE USERS DETAILS
#'''
all_users = {}
f = open('db.dat','rb')
all_users = p.load(f)
f.close()
print all_users

#Joke Function
def joke():
    f=open("jokes.txt",'r')
    l=f.readlines()
    r=random.randrange(0,len(l))
    return l[r]

#Strip the Search string
def strip_search(str):
    L=['a', 'abaft', 'aboard', 'about', 'above', 'absent', 'across', 'afore', 'after',
'against', 'along',
       'alongside', 'amid', 'amidst', 'among', 'amongst', 'an', 'anenst', 'apropos',
'apud', 'around', 'as', 'aside',
       'astride', 'at', 'athwart', 'atop', 'barring', 'before', 'behind', 'below',
'beneath', 'beside', 'besides',
       'between', 'beyond', 'but', 'by', 'circa', 'concerning', 'despite', 'down',
'during', 'except', 'excluding',
       'failing', 'following', 'for', 'forenenst', 'from', 'given', 'in', 'including',
'inside', 'into', 'lest', 'like',
       'mid', 'midst', 'minus', 'modulo', 'near', 'next', 'notwithstanding', 'of', 'off',
'on', 'onto', 'opposite', 'out',
       'outside', 'over', 'pace', 'past', 'per', 'plus', 'pro', 'qua', 'regarding',
'round', 'sans', 'save', 'since', 'than',
       'through', 'throughout', 'till', 'times', 'to', 'toward', 'towards', 'under',
'underneath', 'unlike', 'until', 'unto',
       'up', 'upon', 'versus', 'via', 'vice', 'with', 'within', 'without',
'worth','is','when','what','whats', 'why' ,
       'which' ,'where' ,'how','the','it','will','do','i','and']
    L1=str.split(" ")
    f=[]
```

```
for i in L1:
        if i.lower() not in L:
            f.append(i.strip(string.punctuation))

    return f

#Get the string code
def get_code(str):
    t=['temp','weather','climate'] #Weather
    m=['rating','movie','film',] #Movie
    s=['who','info','meaning'] #Search
    for i in t:
        if re.search(i,str,re.IGNORECASE):
            return "t"
    for i in m:
        if re.search(i,str,re.IGNORECASE):
            return "m"
    for i in s:
        if re.search(i,str,re.IGNORECASE):
            return "s"
    if re.search('[+-\/*]',str,re.IGNORECASE) or re.search('add',str,re.IGNORECASE)or
re.search('sum',str,re.IGNORECASE) or re.search('subract',str,re.IGNORECASE)or
re.search('difference',str,re.IGNORECASE) or re.search('divide',str,re.IGNORECASE)or
re.search('product',str,re.IGNORECASE)or re.search('quotient',str,re.IGNORECASE) or
re.search('multiply',str,re.IGNORECASE) :
        return "math"
    else:
        return "Error: Could not find the code."

strip_list=[]
#Code and Strip search Together
def strip_search_code(str):
    f=[]
    f.append(get_code(str))
    f.extend(strip_search(str))
    print f
    global strip_list
    strip_list.extend(f)
    return f

#Movie Ratings
def movie_rat(f):
    movie=f[2]
    if len(f)>3:
        for i in range(3,len(f)):
            movie+=' '+f[i]

    rt = RT('53uhmdfpu5sybbb5y529skkh')
    print rt.search(movie,page_limit=1)[0]['ratings']
    print 'Movie:',movie  #Testing purpose only
    #D=RT('53uhmdfpu5sybbb5y529skkh').search(movie,page_limit=1) #amitbj96
    #return D[0]['ratings']

#Wikipeda
def search_wiki(f):
    search=f[2]
    if len(f)>3:
        for i in range(3,len(f)):
            search+=' '+f[i]
    print "Search:",search #Testing purpose only
    #print wikipedia.search(search)
    #print wikipedia.page(search).content
    return wikipedia.summary(search, sentences=2)
```

```
#Weather
def get_temp(f):
    city=f[2]
    if len(f)>3:
        for i in range(3,len(f)):
            city+=' '+f[i]
    print "City/Country:",city #Testing purpose only
    pyowm_connect = pyowm.OWM('d96fd08f487348c0277372130a6ab12e')
    return pyowm_connect.weather_at(f[2]).get_weather().get_temperature('celsius')

def get_weather_status(f):
    city=f[2]
    if len(f)>3:
        for i in range(3,len(f)):
            city+=' '+f[i]
    print "City/Country:",city #Testing purpose only
    pyowm_connect = pyowm.OWM('d96fd08f487348c0277372130a6ab12e')
    return pyowm_connect.weather_at(f[2]).get_weather().get_status()

#Dictionry
def get_meaning(f):
    search=f[3]
    if len(f)>3:
        for i in range(4,len(f)):
            search+=' '+f[i]
    print "Word:",search #Testing purpose only
    apiUrl = 'http://api.wordnik.com/v4'
    apiKey = '16d82911721b5989a4454e8b04200cc16d34cdf4c33b588f9' #theamit
    client = swagger.ApiClient(apiKey, apiUrl)
    wordApi = WordApi.WordApi(client)
    #example = wordApi.getTopExample('irony')
    #print example.text
    definitions = wordApi.getDefinitions(search)
    return definitions[0].text


#Offline basic Math Calculator
def math_calculate(str):
    sign=''
    for i in str:
        if re.search('[+-/*]',i,re.IGNORECASE):
            if re.search('[+]',i):
                sign='+'
            elif re.search('[-]',i):
                sign='-'
            elif re.search('[/]',i):
                sign='/'
            elif re.search('[*]',i):
                sign='*'
            if len(i)!=1:
                number_list= i.split(sign)
                if sign=='+':
                    return int(float(number_list[0])+float(number_list[1]))
                elif sign=='-':
                    return int(float(number_list[0])-float(number_list[1]))
                elif sign=='*':
                    return int(float(number_list[0])*float(number_list[1]))
                elif sign=='/':
                    return int(float(number_list[0])/float(number_list[1]))
                break
    else:
        for i in str:
```

```
if re.search('add',i,re.IGNORECASE)or re.search('sum',i,re.IGNORECASE):
                sign='+'
                break
            elif re.search('subract',i,re.IGNORECASE)or
re.search('difference',i,re.IGNORECASE):
                sign='-'
                break
            elif re.search('multiply',i,re.IGNORECASE)or
re.search('prodict',i,re.IGNORECASE):
                sign='*'
                break
            elif re.search('divide',i,re.IGNORECASE)or
re.search('quotient',i,re.IGNORECASE):
                sign='/'
                break

        if sign=='+':
            return int(float(str[-2])+float(str[-1]))
        elif sign=='-':
            return int(float(str[-1])-float(str[-2]))
        elif sign=='*':
            return int(float(str[-2])*float(str[-1]))
        elif sign=='/':
            return int(float(str[-2])/float(str[-1]))

#Checking...
def check_muses(str):
    print "FUNCTION CHECK ENTERED"
    f=strip_search_code(str)
    if f[0]=='math':
        print math_calculate(f)
        return math_calculate(f)
    elif re.search('joke',str,re.IGNORECASE):
        print joke()
        return joke()

    elif re.search('Who am i',str,re.IGNORECASE):
        print open("taken.txt",'r').read()

    elif f[0]=='s':
        if re.search('word',str,re.IGNORECASE):
            print get_meaning(f)
            return get_meaning(f)
        else:
            print search_wiki(f)
            return search_wiki(f)

    elif f[0]=='t':
        print get_temp(f),get_weather_status(f)
        return [get_temp(f),get_weather_status(f)]

    elif f[0]=='m':
        print 'Rat is running...'
        print movie_rat(f)
        print
        print
        return movie_rat(f)

    else:
        print "No API/ No function of find the answes of the qery"


no_of_qs = 0 #To delete the 2 and show the next one
```

```
answer_frames = []
def del_history():
    print 'Deleting history'
    print history
    history_win.destroy()
    del history[:]
    h = open('history.dat','wb')
    p.dump(history,h)
    h.close()
    print history

def display_answer():
    global no_of_qs
    global answer_frames
    no_of_qs += 1

    question = ask_muses.get()
    if(question!=''):
        history.append(question)
        h = open('history.dat','wb')
        p.dump(history,h)
        print history
        print 'Histor-ised it'
        h.close()
        print "RAW:",question
        main_intro_frame.destroy()
        answer = check_muses(question)
        print 'ANSWER:',answer
        global strip_list
        details_of_question = strip_list
        strip_list = []
        print "details_of_question: ",details_of_question

        if(no_of_qs>=2):
            for i in range(len(answer_frames)):
                answer_frames[i].destroy()
            no_of_qs = 0
            answer_frames = []

        answer_frame = Frame(muses)
        answer_frame.config(width=500,height=100)
        answer_frame.config(bg='#26394C')
        answer_frame.pack()

        answer_frames.append(answer_frame)
        print 'THE ANSWER FRAMES',answer_frames

        if(details_of_question[0]=='s'):
            question_label = Label(answer_frame,text='Q:'+question)
            question_label.config(bg='#202020',fg='#ffffff')
            question_label.config(font=('Moon Flower',30))
            question_label.config(height=2,width=100)
            question_label.pack()

            answer_label = Label(answer_frame,text='A:'+answer,wraplength=500 )
            answer_label.config(bg='#2B4157',fg='#ffffff')
            answer_label.config(font=('Moon Flower',20))
            answer_label.config(height=7,width=100)
            answer_label.pack()
        elif(details_of_question[0]=='t'):
            temp_colors = ['#87CCD3','#EDD872','#F9A61A','#D54F2A']
            temp_color = ''
            question_label = Label(answer_frame,text=question)
```

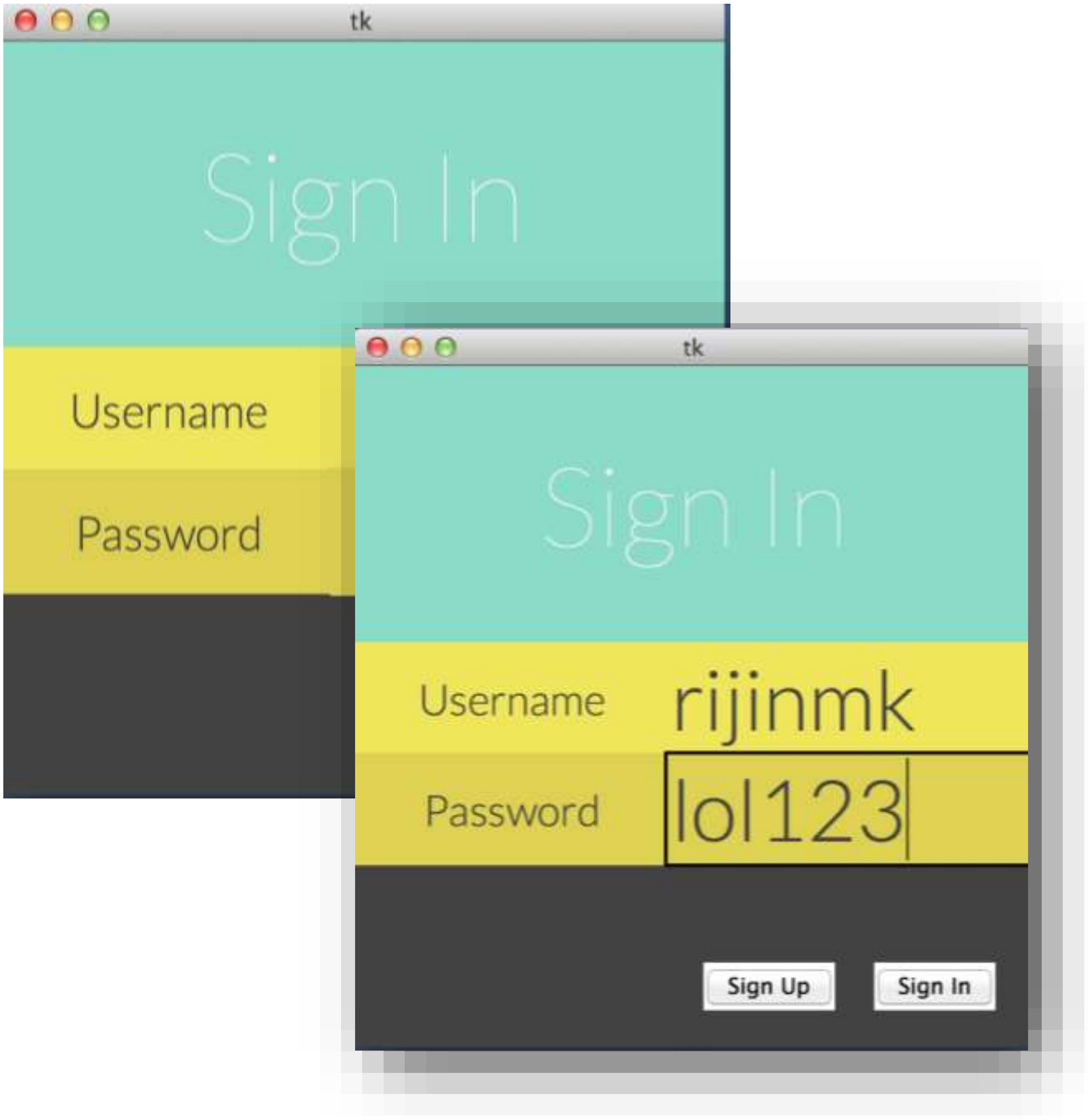
```
question_label.config(bg='#202020',fg='#ffffff')
            question_label.config(font=('Moon Flower',30))
            question_label.config(height=2,width=100)
            question_label.pack()

            if(int(answer[0]['temp']) < 10):
                temp_color = temp_colors[0]
            elif(int(answer[0]['temp']) > 10 and int(answer[0]['temp']) < 30 ):
                temp_color = temp_colors[1]
            elif(int(answer[0]['temp']) > 30 and int(answer[0]['temp']) < 40 ):
                temp_color = temp_colors[2]
            elif(int(answer[0]['temp']) > 40):
                temp_color = temp_colors[3]

            color_indi = Label(answer_frame,text='hello')
            color_indi.config(bg=temp_color,fg=temp_color)
            color_indi.config(font=('Moon Flower',2))
            color_indi.config(height=1,width=1000)
            color_indi.pack()

            temp_label = Label(answer_frame,text='The temprature is ' +
str(int(answer[0]['temp'])) + '°C')
            temp_label.config(bg='#2B4157',fg='#ffffff')
            temp_label.config(font=('Moon Flower',40))
            temp_label.config(height=2,width=100)
            temp_label.pack()

            status_label = Label(answer_frame,text='Weather status: ' +
str(answer[1].encode('ascii')))
            status_label.config(bg='#465057',fg='#ffffff')
            status_label.config(font=('Moon Flower',20))
            status_label.config(height=2,width=100)
            status_label.pack()

        elif(details_of_question[0]=='m'):
            question_label = Label(answer_frame,text=question)
            question_label.config(bg='#202020',fg='#ffffff')
            question_label.config(font=('Moon Flower',30))
            question_label.config(height=2,width=100)
            question_label.pack()

            answser_label = Label(answer_frame,text='The audience score for the movie
'+details_of_question[2] + ' is: ' )
            answser_label.config(bg='#26394C',fg='#ffffff')
            answser_label.config(font=('Moon Flower',30))
            answser_label.config(height=2,width=100)
            answser_label.pack()
        elif(details_of_question[-1].lower()=='history'):
            question_label = Label(answer_frame,text='Opening MUSES History window')
            question_label.config(bg='#7BE19A',fg='#ffffff')
            question_label.config(font=('Moon Flower',30))
            question_label.config(height=2,width=100)
            question_label.pack()

            history_win = Tk()
            history_win.geometry('500x700')
            history_win.config(bg='#333333')
            global history_win

            history_label = Label(history_win,text='Your history on muses')
            history_label.config(bg='#63A599',fg='#ffffff')
            history_label.config(font=('Moon Flower',60))
            history_label.config(height=2,width=100)
history_label.pack()

            questions_asked = [i for i in history]


            history_no = Label(history_win,text='You have asked ' +
str(len(questions_asked)) + ' question(s)!')
            history_no.config(bg='#67AB9E',fg='#ffffff')
            history_no.config(font=('Moon Flower',20))
            history_no.config(height=2,width=100)
            history_no.pack()


            question_labels = [Label(history_win,text=questions_asked[i]) for i in
range(len(history))]
            for i in range(len(history)):
                if(i%2==0):
                    question_labels[i].config(bg='#202020',fg='#ffffff')
                else:
                    question_labels[i].config(bg='#333333',fg='#ffffff')
                question_labels[i].config(font=('Moon Flower',20))
                question_labels[i].config(height=2,width=100)
            for i in range(len(history)):
                question_labels[i].pack()

            delete_history = Button(history_win,text='Delete all?',command=del_history)
            delete_history.place(x=400,y=150)

            history_win.mainloop()
        else:
            question_label = Label(answer_frame,text=question)
            question_label.config(bg='#202020',fg='#ffffff')
            question_label.config(font=('Moon Flower',30))
            question_label.config(height=2,width=100)
            question_label.pack()

            answser_label = Label(answer_frame,text=answer,wraplength=300)
            answser_label.config(bg='#26394C',fg='#ffffff')
            answser_label.config(font=('Moon Flower',20))
            answser_label.config(height=5,width=100)
            answser_label.pack()



def open_muses(x):

    details = all_users[x] #List of Information

    muses = Tk()
    muses.geometry('500x700')
    muses.config(bg='#1F2F3F')
    global muses

    muses_title = Label(muses,text='MUSES')
    muses_title.config(bg='#63A599',fg='#ffffff')
    muses_title.config(font=('Moon Flower',60))
    muses_title.config(height=2,width=100)
    muses_title.pack()

    muses_version = Label(muses,text='1.0v')
    muses_version.config(bg='#50857B',fg='#ffffff')
    muses_version.config(font=('Moon Flower',20))
    muses_version.config(height=1,width=100)
muses_version.pack()

    ask_muses = Entry(muses)
    ask_muses.config(width=100)
    ask_muses.config(font=('Moon Flower',50))
    ask_muses.config(bg='#465057',fg='#eeeeee')
    ask_muses.config(highlightbackground='#465057')
    ask_muses.config(bd=0)
    ask_muses.pack()
    global ask_muses

    ask_muses_button = Button(muses,text='Ask',command=display_answer)
    ask_muses_button.place(x=400,y=190)

    main_intro_frame = Frame(muses,width=500,height=462)
    main_intro_frame.config(bg='#eeeeee')
    main_intro_frame.pack()
    global main_intro_frame

    inst = '#3B5978'

    inst_title = Label(main_intro_frame,text='What can you ask to muses:')
    inst_title.config(bg='#24374A',fg='#ffffff')
    inst_title.config(font=('Moon Flower',40))
    inst_title.config(height=2,width=100)
    inst_title.pack()

    inst_1 = Label(main_intro_frame,text='• Ratings of James Bond')
    inst_1.config(bg=inst,fg='#ffffff')
    inst_1.config(font=('Moon Flower',40))
    inst_1.config(height=2,width=100)
    inst_1.pack()

    inst_2 = Label(main_intro_frame,text='• Who is Obama')
    inst_2.config(bg=inst,fg='#ffffff')
    inst_2.config(font=('Moon Flower',40))
    inst_2.config(height=2,width=100)
    inst_2.pack()

    inst_3 = Label(main_intro_frame,text='• Whats the weather in New Delhi')
    inst_3.config(bg=inst,fg='#ffffff')
    inst_3.config(font=('Moon Flower',40))
    inst_3.config(height=2,width=100)
    inst_3.pack()

    inst_4 = Label(main_intro_frame,text='• Meaning of the word network')
    inst_4.config(bg=inst,fg='#ffffff')
    inst_4.config(font=('Moon Flower',40))
    inst_4.config(height=2,width=100)
    inst_4.pack()

    muses.mainloop()

def pretty_details(x):
    print
    print
    print '------ Congrats -------------------'
    print '| Firstname: ' + x[0] + '\t\t |'
    print '| Username: '+ x[1] + '\t\t |'
    print '| Password: ' + x[2] + '\t\t |'
    print '| Email: ' + x[3] + '\t\t |'
    print '| Country: ' + x[4] + '\t\t\t |'
    print '-----------------------------------'
```

```
print
    print

def show_sign_up():
    sign_up.deiconify()
    print 'Sign up button clicked'

def hide_sign_up():
    sign_up.withdraw()

def sign_in_check():
    all_usernames = all_users.keys()
    username = sign_in_username_entry.get()
    password = sign_in_password_entry.get()
    if(username in all_usernames):
        if(all_users[username][1] == password):
            sign_in.destroy()
            open_muses(username)
            print 'Successfully logged in'
        else:
            print
            print 'Wrong username or password'
    else:
        print 'Wrong username or password'

def check(x):
    usernames = all_users.keys()
    c = 1
    for i in x:
        if(i==''):
            c = 0
    if(c == 0):
        print
        print 'Some feilds are empty'
        return 0
    elif(x[1] in usernames):
        print
        print 'Username already exists'
        return 0
    else:
        return 1

def register():
    firstname = sign_up_fn.get()
    username = sign_up_username.get()
    password = sign_up_password.get()
    email = sign_up_email.get()
    country = sign_up_country.get()
    if(check([firstname,username,password,email,country])):
        all_users[username] = [firstname,password,email,country]
        pretty_details([firstname,username,password,email,country])
        f = open('db.dat','wb')
        p.dump(all_users,f)
        f.close()
        hide_sign_up()
        print 'Congrats, You are registered'

    print 'Register button clicked'
```

```
sign_in = Tk()
sign_in.geometry('400x400')
sign_in.config(bg='#333333')

#SIGN IN:---------> String variables
sign_in_username_entry = StringVar()
sign_in_password_entry = StringVar()

#Sign in title
sign_in_label = Label(sign_in,text='Sign In')
sign_in_label.config(bg='#78D6BD',fg='#eeeeee')
sign_in_label.config(width=100,height=2)
sign_in_label.config(font=('Lato Hairline',65))
sign_in_label.pack()

#Sign in user name label
sign_in_username_label = Label(sign_in,text='Username')
sign_in_username_label.config(bg='#EBE53E',fg='#333333')
sign_in_username_label.config(font=('Lato Light',25))
sign_in_username_label.config(width=12,height=2)
sign_in_username_label.place(x=0,y=162)

#Sign in username entry
sign_in_username = Entry(sign_in,textvariable=sign_in_username_entry)
sign_in_username.config(width=8)
sign_in_username.config(font=('Lato Light',49))
sign_in_username.config(bg='#EBE53E',fg='#333333')
sign_in_username.config(highlightbackground='#EBE53E')
sign_in_username.config(bd=0)
sign_in_username.place(x=182,y=162)

#Sign in password label
sign_in_password_label = Label(sign_in,text='Password')
sign_in_password_label.config(bg='#D8CD36',fg='#333333')
sign_in_password_label.config(font=('Lato Light',25))
sign_in_password_label.config(width=12,height=2)
sign_in_password_label.place(x=0,y=227)

#Sign in password entry
sign_in_password = Entry(sign_in,textvariable=sign_in_password_entry)
sign_in_password.config(width=8)
sign_in_password.config(font=('Lato Light',50))
sign_in_password.config(bg='#D8CD36',fg='#333333')
sign_in_password.config(highlightbackground='#D8CD36')
sign_in_password.config(bd=0)
sign_in_password.place(x=182,y=226)

sign_in_button = Button(sign_in,text='Sign In',command=sign_in_check)
sign_in_button.place(x=305,y=350)

sign_up_button = Button(sign_in,text='Sign Up',command=show_sign_up)
sign_up_button.place(x=205,y=350)


sign_up = Tk()
sign_up.geometry('400x600')
sign_up.config(bg='#333333')

#HIDE SIGN UP BOX
sign_up.withdraw()
```

```python
#SIGN UP:---------> String variables
x = StringVar()

#Sign up title
sign_up_label = Label(sign_up,text='Sign Up')
sign_up_label.config(bg='#EF5F55',fg='#eeeeee')
sign_up_label.config(width=100,height=2)
sign_up_label.config(font=('Lato Hairline',65))
sign_up_label.pack()

#Sign up first name label
sign_up_fn_label = Label(sign_up,text='First name')
sign_up_fn_label.config(bg='#B39CBE',fg='#eeeeee')
sign_up_fn_label.config(font=('Lato Light',25))
sign_up_fn_label.config(width=12,height=2)
sign_up_fn_label.place(x=0,y=162)

#Sign up First name
sign_up_fn = Entry(sign_up,textvariable=x)
sign_up_fn.config(width=8)
sign_up_fn.config(font=('Lato Light',50))
sign_up_fn.config(bg='#8F6DA1',fg='#333333')
sign_up_fn.config(highlightbackground='#8F6DA1')
sign_up_fn.config(bd=0)
sign_up_fn.place(x=182,y=162)

#Sign up username label
sign_up_username_label = Label(sign_up,text='Username')
sign_up_username_label.config(bg='#8F6DA1',fg='#eeeeee')
sign_up_username_label.config(font=('Lato Light',25))
sign_up_username_label.config(width=12,height=2)
sign_up_username_label.place(x=0,y=228)

#Sign up username
sign_up_username = Entry(sign_up)
sign_up_username.config(width=8)
sign_up_username.config(font=('Lato Light',50))
sign_up_username.config(bg='#B39CBE',fg='#333333')
sign_up_username.config(highlightbackground='#B39CBE')
sign_up_username.config(bd=0)
sign_up_username.place(x=182,y=228)

#Sign up password label
sign_up_password_label = Label(sign_up,text='Password')
sign_up_password_label.config(bg='#B39CBE',fg='#eeeeee')
sign_up_password_label.config(font=('Lato Light',25))
sign_up_password_label.config(width=12,height=2)
sign_up_password_label.place(x=0,y=294)

#Sign up password
sign_up_password = Entry(sign_up)
sign_up_password.config(width=8)
sign_up_password.config(font=('Lato Light',50))
sign_up_password.config(bg='#8F6DA1',fg='#333333')
sign_up_password.config(highlightbackground='#8F6DA1')
sign_up_password.config(bd=0)
sign_up_password.place(x=182,y=294)

#Sign up email label
sign_up_password_label = Label(sign_up,text='Email')
sign_up_password_label.config(bg='#8F6DA1',fg='#eeeeee')
sign_up_password_label.config(font=('Lato Light',25))
sign_up_password_label.config(width=12,height=2)
sign_up_password_label.place(x=0,y=360)

#Sign up email
sign_up_email= Entry(sign_up)
sign_up_email.config(width=8)
sign_up_email.config(font=('Lato Light',50))
sign_up_email.config(bg='#B39CBE',fg='#333333')
sign_up_email.config(highlightbackground='#B39CBE')
sign_up_email.config(bd=0)
sign_up_email.place(x=182,y=360)

#Sign up country label
sign_up_password_label = Label(sign_up,text='Country')
sign_up_password_label.config(bg='#B39CBE',fg='#eeeeee')
sign_up_password_label.config(font=('Lato Light',25))
sign_up_password_label.config(width=12,height=2)
sign_up_password_label.place(x=0,y=426)

#Sign up country
sign_up_country= Entry(sign_up)
sign_up_country.config(width=8)
sign_up_country.config(font=('Lato Light',50))
sign_up_country.config(bg='#8F6DA1',fg='#333333')
sign_up_country.config(highlightbackground='#8F6DA1')
sign_up_country.config(bd=0)
sign_up_country.place(x=182,y=426)

register_button = Button(sign_up,text='Register',command=register)
register_button.place(x=305,y=550)

cancel_button = Button(sign_up,text='Cancel',command=hide_sign_up)
cancel_button.place(x=205,y=550)

#<------------------------------------------------------------->
#End the boxes
sign_up.mainloop()
sign_in.mainloop()
#<------------------------------------------------------------->
```

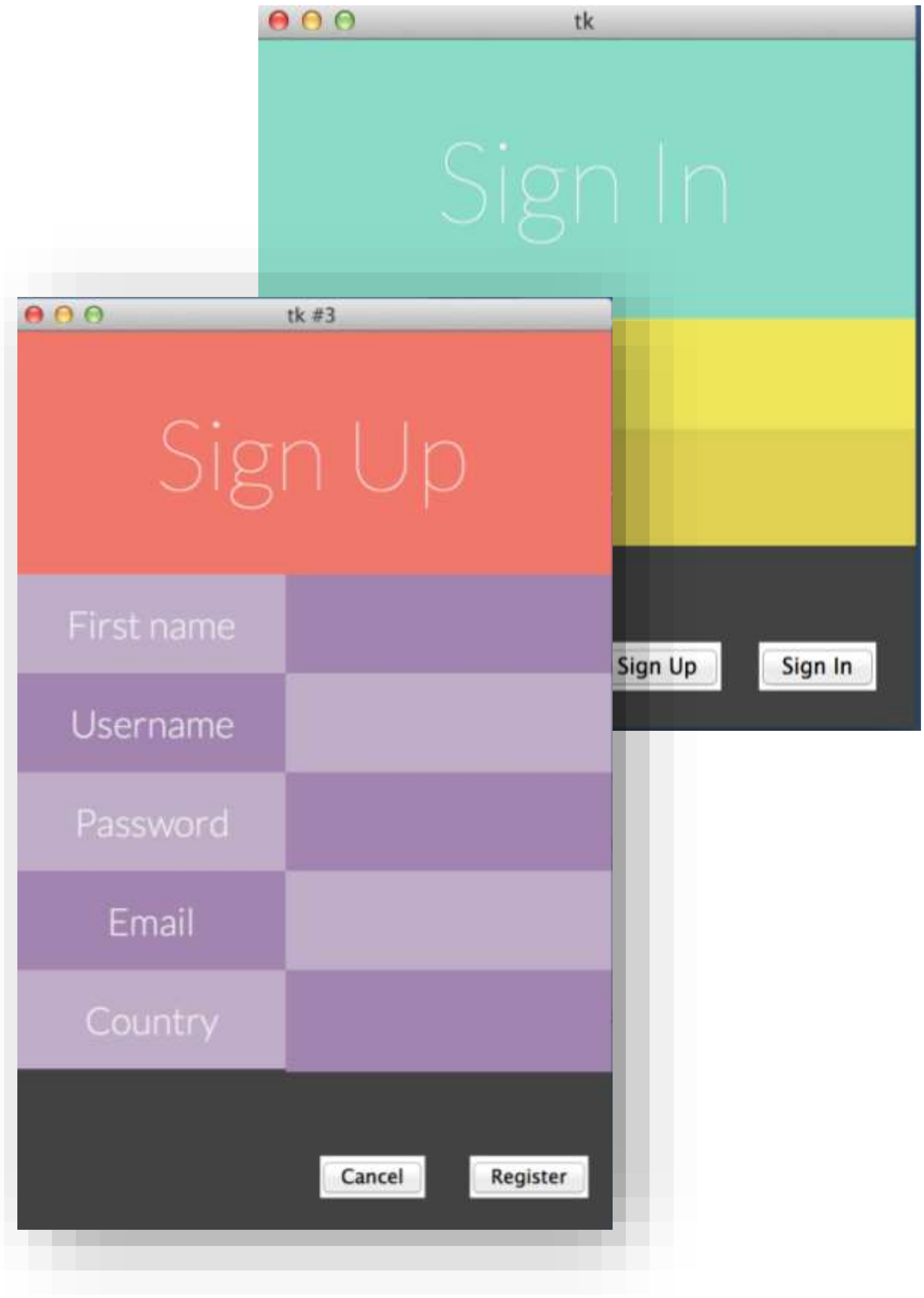
SCREEN SHOTS



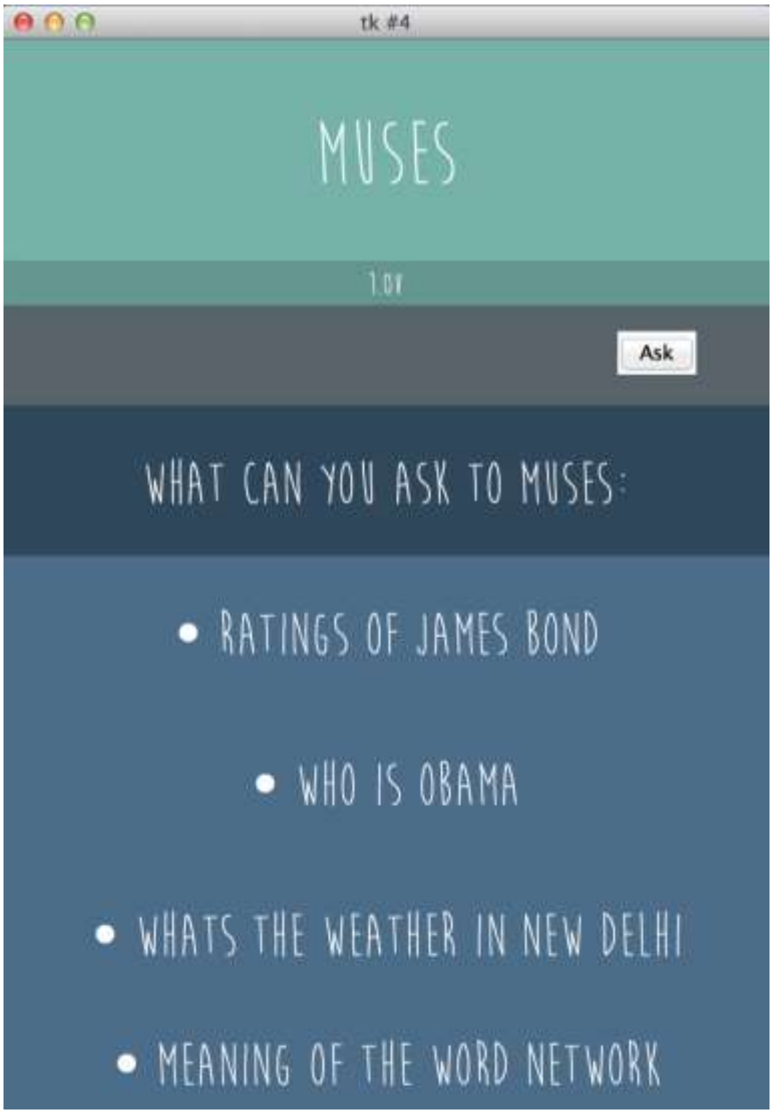
Username rijinmk

Password lol123

Sign up Window



Screen after Logging in


tk #4

MUSES

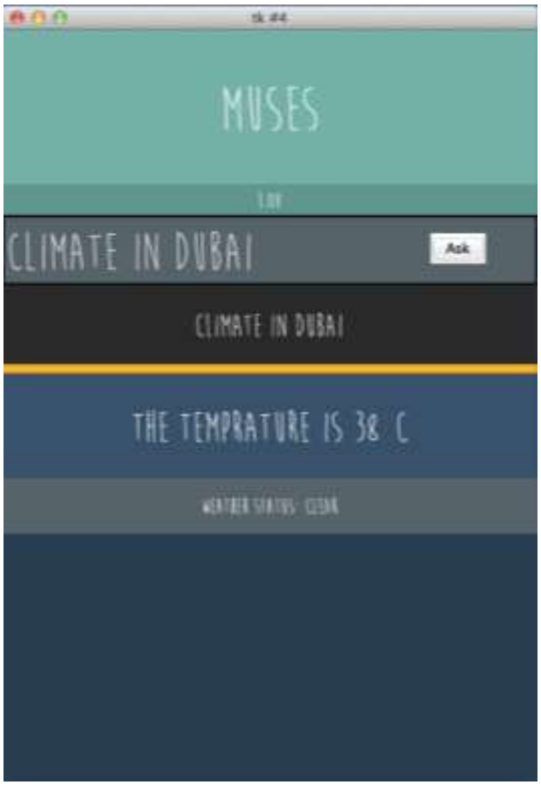
1.0V

Ask

WHAT CAN YOU ASK TO MUSES:

- RATINGS OF JAMES BOND
- WHO IS OBAMA
- WHATS THE WEATHER IN NEW DELHI
- MEANING OF THE WORD NETWORK

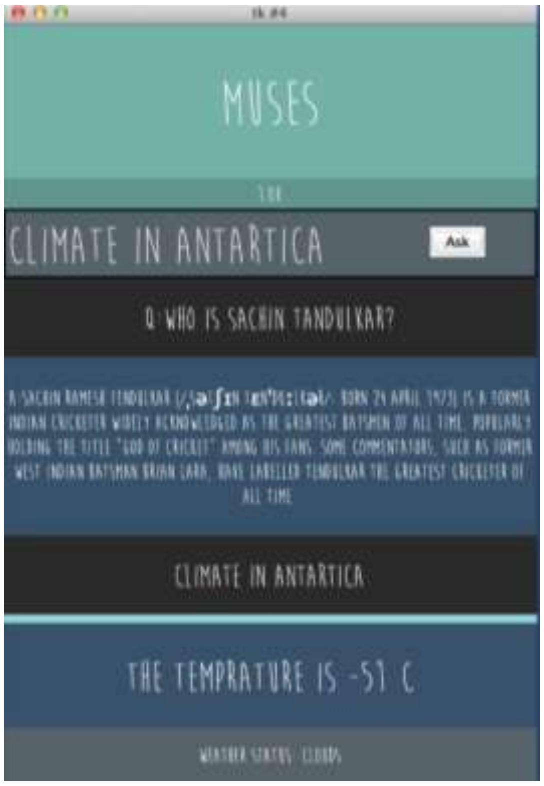
Examples of How you can use Muses



CLIMATE IN DUBAI

THE TEMPRATURE IS 38 C

WEATHER STATUS: CLEAR



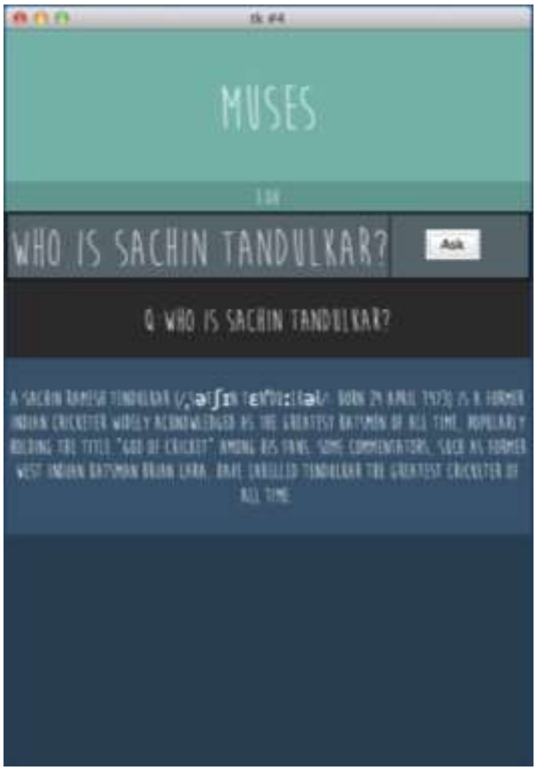
Q: WHO IS SACHIN TANDULKAR?

A: SACHIN RAMESH TENDULKAR (/ˌsətʃɪn tɛnˈduːlkər/; BORN 24 APRIL 1973) IS A FORMER INDIAN CRICKETER WIDELY ACKNOWLEDGED AS THE GREATEST BATSMEN OF ALL TIME. POPULARLY HOLDING THE TITLE "GOD OF CRICKET" AMONG HIS FANS. SOME COMMENTATORS, SUCH AS FORMER WEST INDIAN BATSMAN BRIAN LARA, HAVE LABELLED TENDULKAR THE GREATEST CRICKETER OF ALL TIME

CLIMATE IN ANTARTICA

THE TEMPRATURE IS -51 C

WEATHER STATUS: CLOUDS



Q: WHO IS SACHIN TANDULKAR?

A: SACHIN RAMESH TENDULKAR (/ˌsətʃɪn tɛnˈduːlkər/; BORN 24 APRIL 1973) IS A FORMER INDIAN CRICKETER WIDELY ACKNOWLEDGED AS THE GREATEST BATSMEN OF ALL TIME. POPULARLY HOLDING THE TITLE "GOD OF CRICKET" AMONG HIS FANS. SOME COMMENTATORS, SUCH AS FORMER WEST INDIAN BATSMAN BRIAN LARA, HAVE LABELLED TENDULKAR THE GREATEST CRICKETER OF ALL TIME