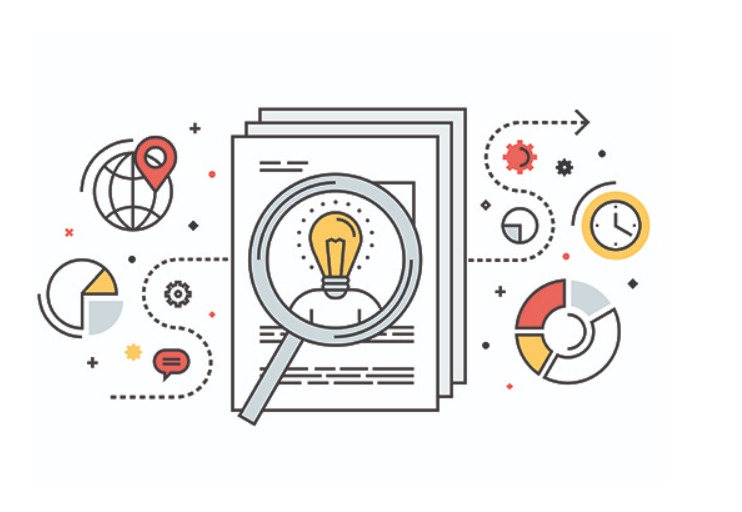
**THE HINDU COLONY CHELLAMMAL VIDYALAYA SR.SEC.SCHOOL, NANGANALLUR, CHENNAI-61**

****

**COMPUTER SCIENCE PROJECT**

 **TOPIC: JOB ASSISTANT**

**DONE BY:**

* **KRISHA.R**
* **NIKITHA.S**
* **RESHMA.M.U.**

**The Hindu Colony Chellammal**

**Vidyalaya Senior Secondary School,**

**NANGANALLUR, CHENNAI-61**

XII-Std

COMPUTER SCIENCE

Name…………………………Class……………… Register No………………Year…………………

***This is certified to be the bonafide work of the student in this Laboratory***

Principal Teacher InCharge

Submitted for the practical examination held on………………………….

at THE HINDU COLONY CHELLAMMAL VIDYALAYA SENIOR SECONDARY SCHOOL, NANGANALLUR, CHENNAI-61.

**Internal Examiner**

**INTERNAL ASSESSMENT External Examiner**

**CONTENT**

1. **INTRODUCTION**
2. **HARDWARE AND SOFTWARE USED**
3. **DATA DICTIONARY**
4. **PROCESS LOGIC**
5. **ENTITY RELATIONSHIP DIAGRAM**
6. **PROCESS LOGIC FLOW CHART**
7. **MAIN PROGRAM EXECUTION**
8. **MENU DESCRIPTION**
9. **SOURCE CODE**
10. **INPUT AND OUTPUT SCREENSHOT**
11. **FUTURESCOPE OF THE PROJECT**
12. **BIBLIOGRAPHY**

**INTRODUCTION**

No matter what age we are, the one question which lingers our minds is ‘what will I do to earn?’, ‘What type of job will I do in future?’. The available paid jobs are just too much to count. The job that we do not only decide the lifestyle we live, but also our mental wellbeing. Many factors must be looked into while deciding on a job excluding the money factor.

Work is important to most of us on many levels. Doing a job, we enjoy and finding satisfaction can provide a meaningful focus for our lives, as well as bringing in an income. Our standard of living hinges on the money we make, while employment often contributes to our self-image and self-esteem.

Work-related problems can affect our physical, emotional and mental health. Common issues include job dissatisfaction, working hours etc. Thus, while selecting the job it is important that one is well aware of the perks that come along with the job. This project is aimed to provide user with the idea about the jobs available in a selected State. With just few clicks the user will get all basic information about the job they are looking forward!

**HARDWARE AND SOFTWARE USED**

**HARDWARE USED**

* Processor: AMD Ryzen 3 3250U with Radeon Graphics
* Installed memory (RAM): 8.00 GB
* System type: 64-bit Operating System, x64-based processor

**SOFTWARE USED**

* Python (3.9 64-bit)
* Snipping tool
* Word processor (Microsoft office student and home 2019 package)

**DATA DICTIONARY**

* **LITERALS:**
* **a = stores age for checking**
* **destry = a frame which quits from the program**
* **b = a button to quit application**
* **x = stores the user’s choice of getting redirected**
* **l = stores the job link for each sector**
* **frame1 = creating a new frame**
* **fedbac = stores the feedback pushed in by the user**
* **frame11 = creating a new frame**
* **fblabel = labelling feedback about us**
* **fbEntry = allowing input for feedback**
* **but = a button to write the feedback in a text file**
* **sec = stores sector value given by the user**
* **frame10 = creating a new frame**
* **ll = stores the details about the particular sector chosen by the user**
* **buton = a button to move to the part of execution where the user gives in the feedback**
* **f = opens States text file**
* **s = reads States text file in the form of a list**
* **st = stores the value of the state chosen by the user**
* **frame9 = creating a new frame**
* **statelabel = Label the state chosen by the user along with the main sectors available**
* **sectorLabel = Label : ’’Enter the sector chosen’’**
* **sectorEntry = entry for sector name**
* **button = a button when clicked, move to the part of execution where it asks for sector name**
* **frame8 = creating a new frame**
* **StateLabel = Label : ‘Enter your state:’**
* **StateEntry = Entry for state name**
* **ok = a button tom move to the next part of execution where it displays details about the sector chosen**
* **eid = stores the email-id of the user**
* **LD = opens logindetaile text file**
* **id = reads the logindetaile text file in the form of a single string**
* **frame7 = creating a new frame**
* **Login\_success = Label : ‘Successfully logged in’**
* **okk = a button to move to next part of execution where it asks for the state name**
* **frame6 = creating a new frame**
* **backLabel = Label : Welcome back message when logged in**
* **emailLabel = Label : ‘Enter the email-id’**
* **emailEntry = entry for email-id**
* **ko = a button when clicked, checks the presence of that email-id in logindetaile text file or not**
* **logindetail = stores the age, name, random id and email id of the user**
* **name = stores the name of the user**
* **frame5 = creating a new frame**
* **nameLabel = Label : ‘Your name’**
* **nameEntry = entry for the user’s name**
* **e-idLabel = Label : ‘Your email-id’**
* **eidEntry = entry for email-id**
* **idf = a random number ranging from 0-1**
* **ids = a random integer ranging from 0-100**
* **ID = this is the sum of idf and ids\*1000**
* **idLabel = displays the random id**
* **b = a button when clicked, stores the details of the user in a text file**
* **frame3 = creates a new frame**
* **r\_or\_l = Label : ‘Register or Login’**
* **login = a button when clicked, gets into login part**
* **register = a button when clicked, gets into registration part**
* **frame4 = creates a new frame**
* **exit = asks the user’s input to exit from this program or not**
* **yes = a button which quits the program**
* **no = a button which proceed toexecute the upcoming parts**
* **frame2 = creates a new frame**
* **ageLabel = Label : ‘Please enter your age’**
* **ageEntry = Entry for the age if the user**
* **agebutton = a button when clicked, stores the age in a text file**
* **frame = creates a new frame**
* **MODULES:**
* **tkinter : to work basic graphics**
* **random : to generate a user id**
* **webbrowser : working with google links**
* **FUNCTIONS:**
* **display() : checks whether the user is above 18 or not**
* **quit() : quits from the program**
* **regorlog() : asks for the user to either register or login**
* **login() : opens a text file and asks for the email – id**
* **check\_presence() : checks whether the email – id is valid or not**
* **state\_find() : checks whether the entered state is valid or not**
* **city() : checks whether the entered sector is valid or not**
* **sector() : displays the details about the sector of that particular state**
* **criteria() : redirects the user to the selected sector job link**
* **feedback (): to get the feedback from user and store it**
* **close (): function under io module for working with file**
* **read (): function under io module for working with file**
* **write (): function under io module for working with file**
* **append (): function under io module for working with file**

**ENTITY RELATIONSHIP DIAGRAM**

**PERSONAL**

**DETAILS**

**COMPANY**

**DETAILS**

**TEXT FILE**

**JOB ASSISTANT**

**(SOFTWARE)**

**USER**

**COMPANY**

**JOB**

**REGISTER**

**TIE-UP**

**APPLY**

**EMPLOYMENT**

The process logic of the project is as follows:

* At first, combo of sentences about our project will be displayed which ensures the user to take solace in this.
* The first part of the program deals with checking the age of the user which is equal or above 18.
* The second part of the program assist the user in registering or logging in with the program (only if the age is equal or above 18).
* If the user is not registering with the program, he/she will be able to use the program only once.
* After registering, the details will be stored in a text file.
* The third part succours the user to find the sectors available in his/her state to surf for a job.
* Once the user types in the state, the main sectors available in the state are displayed.
* The user types in the name of the sector to get to know about the sector and list of jobs available related to that particular sector.
* The last part is about giving a feedback as everyone wishes to receive a feedback about the work they do with vigilance.
* The feedback will be stored in another file after the user pushes it in.
* Then the user will be redirected to a particular site where list of jobs related to that sector will be displayed.
* Then the user can exit from the program.

PROCESS LOGIC FLOW CHART

MAIN PROGRAM EXECUTION

**MENU DESCRIPTION**

* **MENU 1**: *“REGISTER”*

This menu gets and stores the details about the user.

* **MENU 2: “***LOGIN”*

This menu is used for getting the information they need.

* **MENU 3:** *“ABOUT US”*

This menu will give the user an idea about who we are and what this program is all about.

* **MENU 4:** *“HELP”*

This menu will let the user know about how to use this program in form of text.

* **MENU 5:** *“FEEDBACK”*

This menu can only be used by registered users. They can give feedback about our program by entering their user ID which would have been provided.

* **MENU 6:** *“EXIT”*

This menu is used to end the execution of the program.

#2  
def display():  
 #3  
 def quit():  
 frame.destroy()  
 frame1.destroy()  
  
  
 global a  
 a = age.get()  
 if 18 > int(a) >= 0 or 99> int(a) >40:  
 frame1 = Tk()  
 frame1.geometry("300x100")  
 destry = Label(frame1,text = "We are sorry, you cant use this program!!").grid(row = 0,column = 2)  
 b = Button(frame1,text='Quit Application',command=quit).grid(row = 2,column = 2)  
  
 #4  
 elif int(a) >= 18 and int(a) <= 50:  
 #7  
 def regorlog():  
  
  
 # 10  
 def login():  
  
 # 11  
 def check\_presence():  
  
 # 12  
 def state\_find():  
  
 #13  
 def city():  
  
 #14  
 def sector():  
  
 #15  
 def feedback():  
  
 def criteria():  
  
 fedbac = str([fb.get()])  
 with open('FB.txt','a') as K:  
 K.write(fedbac+'\n')  
 messagebox.showinfo("","Thank You")  
 with open("sectlink.txt", 'r') as file:  
 for line in file:  
 if sec in line:  
 x = messagebox.askquestion("Redirecting",  
 "Redirecting to " + line + " site")  
 if x == "yes":  
 l = (line.split("="))[1]  
 import webbrowser  
 webbrowser.open(l)  
 break  
  
  
 frame11 = Tk()  
 frame11.geometry('500x500')  
 fbLabel = Label(frame11,text = "Feedback about us").grid(row =0,column = 0)  
 fb = StringVar()  
 fbEntry = Entry(frame11,textvariable = fb).grid(row=1,column=0)  
 fedbac = fb.get()  
 but = Button(frame11,text = "Ok",command = criteria).grid(row = 2,column = 0)  
  
  
  
  
  
 global sec  
 sec = sectorr.get()  
 with open('secdetails.txt', 'r') as p:  
 for lline in p:  
 if sec in lline:  
 frame10 = Tk()  
 frame10.geometry('2000x500')  
 ll = lline.split(";")  
 k = 0  
 for i in ll:  
 secdetLabel = Label(frame10, text=i).grid(row=k, column=0)  
 k+=1  
 buton = Button(frame10, text="Done Reading", command=feedback).grid(row=7,  
 column=0)  
 break  
 else:  
 messagebox.showwarning("Warning", "Not found. Try Again!!")  
  
  
  
 f = open('States', 'r')  
 s = f.readlines()  
 global st  
 st = Statee.get()  
 for line in s:  
 if st in line:  
 with open("Sectors.txt", 'r') as g:  
 for linee in g:  
 if st in linee:  
 frame8.destroy()  
 frame9 = Tk()  
 frame9.geometry('1000x100')  
 statelabel = Label(frame9,text = linee).grid(row = 0,column = 0)  
 sectorLabel = Label(frame9,text = "Enter the sector chosen:").grid(row=1,column=0)  
 sectorr = StringVar()  
 sectorEntry = Entry(frame9,textvariable = sectorr).grid(row = 1,column =1)  
 button = Button(frame9,text = "Ok",command = sector).grid(row = 2,column = 1)  
 break  
 else:  
 messagebox.showwarning("Warning","State Not found. Try Again")  
  
  
  
  
  
 frame7.destroy()  
 frame8 = Tk()  
 frame8.geometry('400x100')  
 StateLabel = Label(frame8,text = "Enter your state:").grid(row = 0,column = 0)  
 Statee = StringVar()  
 StateEntry = Entry(frame8,textvariable = Statee).grid(row = 0,column = 1)  
 ok = Button(frame8,text = "Ok",command = city).grid(row = 1,column = 1)  
  
  
 #check presence  
 eid = emaill.get()  
 LD = open("logindetaile.txt", "r")  
 id = LD.read()  
 if eid in id:  
 frame6.destroy()  
 frame7 = Tk()  
 frame7.geometry('200x90')  
 Login\_success = Label(frame7,text = "Successfully logged in").grid(row = 0,column = 0)  
 okk = Button(frame7,text = "Ok",command = state\_find).grid(row = 1,column = 1)  
 else:  
 messagebox.showwarning("Warning", "Not found. Please Try Again!!")  
  
 frame3.destroy()  
  
 frame6 = Tk()  
 frame6.geometry('500x200')  
  
 # welcome back msg display  
 backLabel = Label(frame6, text="WELCOME BACK!").grid(row=0, column=0)  
  
 # typing in login details  
 emailLabel = Label(frame6, text="Enter your email id to retrieve your details and past records:").grid(  
 row=1, column=0)  
 emaill = StringVar()  
 emailEntry = Entry(frame6, textvariable=emaill).grid(row=1, column=1)  
 ko = Button(frame6, text="Ok", command=check\_presence).grid(row=2, column=1)  
  
  
  
 #8  
 def register():  
 frame3.destroy()  
  
 #9  
 def details\_of\_user():  
  
  
 #13\_1  
 def state\_find():  
  
 #14\_1  
 def city():  
  
  
 #15\_1  
 def sector():  
  
 #16\_1  
 def feedback():  
  
 def criteria():  
  
 fedbac = str([fb.get()])  
 with open('FB.txt', 'a') as K:  
 K.write(fedbac + '\n')  
 messagebox.showinfo("", "Thank You")  
 with open("sectlink.txt", 'r') as file:  
 for line in file:  
 if sec in line:  
 x = messagebox.askquestion("Redirecting",  
 "Redirecting to " + line + " site")  
 if x == "yes":  
 l = (line.split("="))[1]  
 import webbrowser  
 webbrowser.open(l)  
 break  
  
  
  
 frame11 = Tk()  
 frame11.geometry('500x500')  
 fbLabel = Label(frame11, text="Feedback about us").grid(row=0, column=0)  
 fb = StringVar()  
 fbEntry = Entry(frame11, textvariable=fb).grid(row=1, column=0)  
 fedbac = fb.get()  
 but = Button(frame11, text="Ok", command=criteria).grid(row=2, column=0)  
  
  
  
 global sec  
 sec = sectorr.get()  
 with open('secdetails.txt', 'r') as p:  
 for lline in p:  
 if sec in lline:  
 frame10 =Tk()  
 frame10.geometry('2000x500')  
 l = lline.split(";")  
 k = 0  
 for i in l:  
 secdetLabel = Label(frame10, text=i).grid(row=k, column=0)  
 k += 1  
 but = Button(frame10, text="Done Reading", command=feedback).grid(row=7, column=0)  
 break  
 else:  
 messagebox.showwarning("Warning","Not found. Try Again!!")  
  
  
  
  
  
 f = open('States', 'r')  
 s = f.readlines()  
 global st  
 st = Statee.get()  
 for line in s:  
 if st in line:  
 with open("Sectors.txt", 'r') as g:  
 for linee in g:  
 if st in linee:  
 frame8.destroy()  
 frame9 = Tk()  
 frame9.geometry('1000x100')  
 statelabel = Label(frame9,text = linee).grid(row = 0,column = 0)  
 sectorLabel = Label(frame9, text="Enter the sector chosen:").grid(row=1, column=0)  
 sectorr = StringVar()  
 sectorEntry = Entry(frame9, textvariable=sectorr).grid(row=1, column=1)  
 button = Button(frame9,text = "Ok",command = sector).grid(row = 2,column = 0)  
 break  
  
 else:  
 messagebox.showwarning("Warning","State Not found. Try Again")  
  
  
  
  
  
 frame8\_1.destroy()  
 frame8 = Tk()  
 frame8.geometry('400x100')  
 StateLabel = Label(frame8,text = "Enter your state:").grid(row = 0,column = 0)  
 Statee = StringVar()  
 StateEntry = Entry(frame8,textvariable = Statee).grid(row = 0,column = 1)  
 ok = Button(frame8,text = "Ok",command = city).grid(row = 1,column = 1)  
  
  
  
  
 def detail(a, info2): # file to save age,info1  
 logindetail = str([a, info2])  
 LD = open("logindetaile.txt", "a+")  
 LD.write(logindetail + "\n")  
 LD.close()  
  
  
 frame5.destroy()  
 # displaying user details  
 global name  
 global eid  
 name = namee.get()  
 eid = eeid.get()  
 frame8\_1 = Tk()  
 frame8\_1.geometry('500x500')  
 check\_detailLabel = Label(frame8\_1,text="Check your details"+"\n"+"Your name: " + name + '\n' + "Your age: " + a + '\n' + "Your email: " + eid + '\n' + "Your random-id: " + str(ID)).grid(row=0 ,column = 0)  
 info2 = [name, eid, ID]  
 detail(a, info2)  
 button = Button(frame8\_1,text = "Ok",command = state\_find).grid(row = 1,column = 1)  
  
  
  
  
 frame5 = Tk()  
 frame5.geometry('500x400')  
  
 #name entry  
 nameLabel = Label(frame5,text = "Your name:").grid(row = 0,column = 0)  
 namee = StringVar()  
 nameEntry = Entry(frame5,textvariable = namee).grid(row = 0,column = 1)  
  
  
  
 #email entry  
 eidLabel = Label(frame5,text = "Your email-id:").grid(row = 1,column = 0)  
 eeid = StringVar()  
 eidEntry = Entry(frame5,textvariable = eeid).grid(row = 1,column = 1)  
  
  
  
 #random id generation  
 import random  
 idf = random.random()  
 ids = random.randint(0, 100)  
 global ID  
 ID = idf + ids \* 1000  
 idLabel = Label(frame5,text = "Your random-id: "+str(ID)).grid(row = 2,column = 0)  
 b = Button(frame5,text = "Done",command = details\_of\_user).grid(row = 3,column = 1)  
  
  
  
  
  
 frame3 = Tk()  
 frame3.geometry("500x200")  
 r\_or\_l = Label(frame3,text = "Register or Login").grid(row = 0,column = 0)  
 register = Button(frame3, text = "Register",command = register ).grid(row = 2,column = 1)  
 login = Button(frame3, text = "Login",command = login).grid(row = 2,column = 2)  
  
  
  
  
  
  
 #6  
 def quit1():  
  
  
 def quit2():  
 frame4.destroy()  
  
 #inner quit  
 frame2.destroy()  
 frame4 = Tk()  
 frame4.geometry("500x200")  
 exit = Label(frame4,text = "Not registering yourself would mean you will not be able to use this program.\n"  
 "Are you sure you do not want to register?").grid(row = 0,column = 0)  
 yes = Button(frame4, text="Yes", command=quit2).grid(row=1, column=1)  
 no = Button(frame4, text='No', command=regorlog).grid(row=1, column=2)  
  
 #5  
 #inner main  
 messagebox.showinfo("About us","We are trying to provide jobs for youngsters who are awaiting to get their salaries.\nWe have provided jobs for large number of people who further poped out with success.\nWe have been awarded as the number one job assistant in our country.\nWe have incorporated several options for the user to get a good job for which they have been awaiting for.\nWhile selecting the job it is important that one is well aware of the perks that come along with the job.\nWe are aimed to provide user with the idea about the jobs available in a selected city.\nWith just few clicks the user will get all basic information about the job they are looking forward!\nWe hope that this would help you to get a job for which you have been looking for!!")  
  
 frame2 = Tk()  
 frame2.geometry("500x200")  
  
 # register or login  
 frame.destroy()  
 r= Label(frame2, text="This takes you to register or login window.\nDo you want to go ahead?").grid(row=0, column=0)  
 yes = Button(frame2, text='Yes', command= regorlog).grid(row=1, column=1)  
 no = Button(frame2, text = "No",command = quit1).grid(row = 1, column = 2)  
 '''   
 b = Button(frame2,  
 text='Quit Application',  
 command=call)  
  
 canvas.create\_window(100, 100,  
 window=b)'''  
  
 else :  
 messagebox.showwarning("Warning",'Check Your input and Try Again')  
  
  
#1  
from tkinter import \*  
from tkinter import messagebox  
  
frame = Tk()  
frame.geometry("500x200")  
  
#age  
ageLabel = Label(frame, text="Please enter your age").grid(row=0, column=0)  
age = StringVar()  
ageEntry = Entry(frame, textvariable=age).grid(row=0, column=1)  
  
  
#done button  
agebutton = Button(frame,text = 'Done', command = display).grid(row=1, column = 1)  
frame.mainloop()

**FUTURESCOPE OF THE PROJECT**

* This project can be further developed by expanding the category of sectors and by adding more city from the states.
* More options will also make this project more informative. With new scenarios coming up, new jobs sectors will also raise which can also be added to the project.
* The option of linking to other job-related organizing program will also help the project be more useful for the users.
* In addition to this, the option to add more information or information related to job conditions by the user will help program be more accurate and relatable as user can add their experience in that particular workspace.
* Creating resume using this program is another way of extending this project in future.

**BIBLIOGRAPHY**

* Sumita Arora- Computer science with python
* Together with- Computer science with python (Rachna Sagar publications)
* Computer Science with Python (CBSE material)