**MINI PROJECT**

**Name- Priya Verma**

**Batch-CS-38**

**Roll-191428**

**Topic-Quiz Application**

**Description:**

**In this project I made a Quiz Application in which I have used following modules:-**

* **random module**
* **tkinter module**
* **sys module**
* **time module**

**Steps followed in this project:-**

**Import all modules**

**In this project random module used to reshuffle all questions.**

**In this , we import tkinter, from tkinter.ttk we import separator and from tkinter.messagebox we import showinfo().**

**In this project time module is used .We import time and strftime from time module.**

|  |
| --- |
| **import random,sys from tkinter import \* from tkinter.ttk import Separator from tkinter.messagebox import showinfo from time import time,strftime** |

**Creating main window**

**Create root object of tkinter class , change the name of window and set title as Quiz Application.To set the dimensions of window and to set the position of the window. I am using geometry() function. As the width is 700 pixels and the height is 500 pixels. We set root background colour as white .Then we initialize value 0 to a variable i.Then we set total time for the quiz and rading instructions part time as 2 min and 30 seconds.**

**Then we write an intro part and we print it.**

|  |
| --- |
| **root=Tk() root.geometry('700x500') root.resizable(0,0) root.title('QUIZ APP')  root.config(bg='White') i=0 timeLeft={'min':2,'sec':5} intro= '''\t\t :: Instructions ::  <><><><><><><><><><><><><><><><><><><><>  1. Total Quiz Time is : 02:00 Min  2. Total questions : 02  3. Total Score : 04 x 100 =400  as you have only one chance to select.  \t\t Good Luck!!  <><><><><><><><><><><><><><><><><><><><> ''' print(intro)** |

**Creating toplevel part**

* **Defining timeShow():**

**In this we define a function timeShow() in which we declare globally I, timeLeft variable in which we take 4 conditions:**

1. **timeLeft[‘min’]==2 and timeLeft[‘sec’]>0(Reading the instructions time)**
2. **timeLeft[‘sec’]>0(start the test after reading the instructions)**
3. **timeLeft[‘min’]!=0 and timeLeft[‘sec’]==0(giving the test)**
4. **timeLeft[‘min’]==0 and timeLeft[‘sec’]==0(result window will open)**

**And also we use strftime which show the time in %H:%M:%S format.**

* **Defining getDetails():**

**Then we define getDetails() function in which we declare gloablly**

**Name, roll, mainWindow, Name, Roll variables**

**Then we get the Name,Roll using get() function and then we use deiconify() function to return the window from icon.**

**We call timeShow() function then and finally we destroy the mainWindow function.**

* **Defining attendance():**

**Then we define attendance() function in which we globally declare name, roll, mainWindow .**

**In his we create toplevel of mainWindow part having Geometry 700x500 and resize it to (0,0). We give this mainWindow a title ‘Quiz App’. Then we define appName same as root.Using label we fix the bg, fg, font, justification of mainWindow.**

**And finally we pack the appName in Top level and fill both side.**

**Then we define the label info ,and we create entry of name and roll and we finally insert Name and Roll in name and roll respectively.**

**Finally we create a submit button and we complete the loop of mainWindow using mainWindow.mainloop().**

**In this way we finish the Toplevel.**

|  |
| --- |
| **def timeShow():  global i,timeLeft  if timeLeft['min']==2 and timeLeft['sec']>0:  note.config(text='You can start Quiz after {} Seconds.'.format(timeLeft['sec']))  timeLeft['sec']-=1  elif timeLeft['sec']>0:  submit.config(state=NORMAL)  instruction.config(text='')  timeLeft['sec']-=1  note.config(text='Time left: {}min:{}sec'.format(timeLeft['min'],timeLeft['sec']))  elif timeLeft['min']!=0 and timeLeft['sec']==0:  timeLeft['min']-=1  timeLeft['sec']=59  note.config(text='Time left: {}min:{}sec'.format(timeLeft['min'],timeLeft['sec']))  elif timeLeft['min']==0 and timeLeft['sec']==0:  print('Time up!')  result()  showtime.config(text=strftime('%H:%M:%S'))  showtime.after(1000,timeShow)  #Attendance  def getDetails():  global name,roll,mainWindow,Name,Roll  Name=name.get()  Roll=roll.get()  root.deiconify()  timeShow()  mainWindow.destroy()  def attendance():  global name,roll,mainWindow  mainWindow=Toplevel(root)  mainWindow.geometry('700x500')  mainWindow.resizable(0,0)  mainWindow.title('QUIZ APP')  #mainWindow.tk.call(img)  mainWindow.config(bg='white')   #app name same as root   appName=Label(mainWindow,text=title,font=('impact',20,'italic'),  justify=CENTER,bg='blue',fg='black')  appName.pack(side=TOP,fill=BOTH)   #label to show info of attendance   info=Label(mainWindow,text="Enter Your Name and Rollno.",bg='white',fg='sky blue',font=('arial',15))  info.place(x=210,y=200)  name=Entry(mainWindow,width=30)  name.place(x=250,y=235)  roll=Entry(mainWindow,width=30)  roll.place(x=250,y=260)  name.insert(END,'Name')  roll.insert(END,'Roll')  submit=Button(mainWindow,text='Confirm & Start',width=20,bg='sky blue',fg='green',command=getDetails)  submit.place(x=265,y=300)  mainWindow.mainloop()  #Toplevel finish** |

**Creating the Functions used in this project**

* **Defining quit\_function():**

**If answer==ok**

**We destroy the root.**

* **Defining dis\_button():**

**In this we set the state of all 4 options of multiple choice question DISABLED.**

* **Defining en\_buttton():**

**In this we set the state of all 4 options of multiple choice question NORMAL.**

* **Defining result():**

**In this we declare score, Name, Roll globally.**

**Then we remove the root without destroying it using withdraw().**

**Then we create a result window having title ‘QUIZ RESULT’ and geometry(200x100) and resize it (0,0).Then we use a protocol ‘WM\_DELETE\_WINDOW’ by using function quit\_function.**

**In this function we created a text file in append mode having all registrations and marks. Then we create a label having text ‘QUIZ OVER’.**

|  |
| --- |
| **#Quit quiz  def quit\_function():  answer = showinfo(message="Try next time")  if answer=='ok':  sys.exit(root.destroy())  #Disable all Button  def dis\_button():  option1.config(state=DISABLED)  option2.config(state=DISABLED)  option3.config(state=DISABLED)  option4.config(state=DISABLED)  #Enable all Button  def en\_button():  option1.config(state=NORMAL)  option2.config(state=NORMAL)  option3.config(state=NORMAL)  option4.config(state=NORMAL)  #Show final result  def result():  global score,Name,Roll  root.withdraw()  top=Toplevel(root)  #top.tk.call('wm','iconphoto',top.\_w,PhotoImage(file=''))  top.geometry('200x100')  top.resizable(0,0)  top.title('QUIZ RESULT')  top.config(bg='sky blue')  top.protocol('WM\_DELETE\_WINDOW',quit\_function)  filename=Name+'\_'+Roll+'.txt'  data='\nStudent:'+Name+'\nRoll:'+Roll+'\nScore: '+str(score)+'\nCompleted quiz at : '+strftime('%d/%m/%y --%H:%M:%S')  with open(filename,'a') as file:  file.write(data)  label=Label(top,text='QUIZ OVER...\n Score: '+str(score),font=30,fg='black',bg='sky blue').place(x=50,y=25)  exitBtn=Button(top,text='Exit',width=10,bg='white',fg='green',command=quit\_function).place(x=50,y=70)  top.mainloop()** |

**Writing Questions and its Answers and options**

**Here we created a dictionary having all questions and its answers. Then we create two empty file que and ans and we divide the questions and answers in between these two lists using for loop.**

**Then we create options of all questions in 2-D list.**

**And we define some variables current, queNo, score, qn, var**

**Here we define var as stringVar().**

**Creating \_next() function and answer() function**

* **When que>0:(when questions are unanswered)**

**Firstly in this function we randomize the questions using random module.**

**Then we create a next button to go to the next question and enable all**

**the options button. And finally we remove that question which is**

**already answered and remove that option that is chosen by the user.**

* **When que==0:(when questions are answered)**

**Then we call the result() function and show the result window.**

* **We define a answer() function in which we increase the value of score by 100 if the option chosen is same as the answer of the question.**

|  |
| --- |
| **def \_next():  global currentQ,currentA,queNo,score,i,qn  i=0   #till last question is left   if len(que)>0:  currentQ=random.choice(que)  print(currentQ)  q=Label(root,text='Que. '+str(qn),font=('arial',10)).place(x=20,y=80)  qn+=1  #\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  queNo=que.index(currentQ)  print(options[queNo])  currentA=questions[currentQ]  #firstly change caption of button   submit.config(text='Next')   #print current question on quelabel   queLabel.config(text=currentQ,fg='green',height=6)   #print options for Questions on labels--- option 1,option2,.....   en\_button()  option1.config(text=options[queNo][0],bg='sky blue',value=options[queNo][0],bd=1,command=answer)  option2.config(text=options[queNo][1], bg='sky blue', value=options[queNo][1], bd=1, command=answer)  option3.config(text=options[queNo][2], bg='sky blue', value=options[queNo][2], bd=1, command=answer)  option4.config(text=options[queNo][3], bg='sky blue', value=options[queNo][3], bd=1, command=answer)   #remove question from list which are asked   que.remove(currentQ)  ans.remove(currentA)  options.remove(options[queNo])  elif len(que)==0:  result() def answer():  global currentQ,currentA,score   #print selected radiobutton   a= var.get()  if currentA==str(a):  score+=100  dis\_button()  else:  dis\_button()** |

**Setting all the attributes of buttons**

**Then we created all the buttons as radiobuttons ,submit button and setting the attributes of all the buttons .**

**Now we check the condition if \_\_name\_\_==\_\_main\_\_**

**Then we call the root.withdraw to remove the root ithout destroying it and then we call the attendance() to register the user and finally we call the root.mainloop() to enter the loop to give the quiz and take the result.**

|  |
| --- |
| **title='''C programming Questions''' appName=Label(root,text=title,font=('impact',20,'italic'),  justify=CENTER,bg='blue',fg='black') appName.pack(side=TOP,fill=BOTH) s=Separator(root).place(x=0,y=195,relwidth=1) #label to show current question  queLabel=Label(root,text='',justify=LEFT,font=25) queLabel.pack(side=TOP,fill=BOTH) s=Separator(root).place(x=0,y=195,relwidth=1)  #options label  option1=Radiobutton(root,text='',bg='white',font=20,width=20,relief=FLAT,indicator=0,value=1,variable=var,bd=0) option1.place(x=150,y=250) option2=Radiobutton(root,text='',bg='white',font=20,width=20,relief=FLAT,indicator=0,value=2,variable=var,bd=0) option2.place(x=400,y=250) option3=Radiobutton(root,text='',bg='white',font=20,width=20,relief=FLAT,indicator=0,value=3,variable=var,bd=0) option3.place(x=150,y=300) option4=Radiobutton(root,text='',bg='white',font=20,width=20,relief=FLAT,indicator=0,value=4,variable=var,bd=0) option4.place(x=400,y=300)  #instructions of Quiz  instruction=Label(root,text=intro,bg='sky blue',fg='white',font=('calibri',15),justify=LEFT) instruction.place(x=150,y=200)  #note to quiz  note=Label(root,text='',font=('impact',10),bg='black',fg='green') note.pack(side=BOTTOM)  #submit button  submit=Button(root,text='Start Quiz',font=('impact',15),fg='white',bg='blue',state=DISABLED,command=\_next) submit.pack(side=BOTTOM)  #show current time  showtime=Label(root,text='',font=20,fg='black',bg='sky blue') showtime.place(x=620,y=50)  #progress bar for time left for each question  if \_\_name\_\_=="\_\_main\_\_":  root.withdraw()  attendance()  root.mainloop()** |

**Code:**

#imports  
  
import random,sys  
from tkinter import \*  
from tkinter.ttk import Separator  
from tkinter.messagebox import showinfo  
from time import time,strftime  
#Window  
  
root=Tk()  
root.geometry('700x500')  
root.resizable(0,0)  
root.title('QUIZ APP')  
  
root.config(bg='White')  
i=0  
timeLeft={'min':2,'sec':5}  
intro= '''\t\t :: Instructions ::  
 <><><><><><><><><><><><><><><><><><><><>  
 1. Total Quiz Time is : 02:00 Min  
 2. Total questions : 02  
 3. Total Score : 04 x 100 =400  
 as you have only one chance to select.  
 \t\t Good Luck!!  
 <><><><><><><><><><><><><><><><><><><><>  
'''  
print(intro)  
  
def timeShow():  
 global i,timeLeft  
 if timeLeft['min']==2 and timeLeft['sec']>0:  
 note.config(text='You can start Quiz after {} Seconds.'.format(timeLeft['sec']))  
 timeLeft['sec']-=1  
 elif timeLeft['sec']>0:  
 submit.config(state=NORMAL)  
 instruction.config(text='')  
 timeLeft['sec']-=1  
 note.config(text='Time left: {}min:{}sec'.format(timeLeft['min'],timeLeft['sec']))  
 elif timeLeft['min']!=0 and timeLeft['sec']==0:  
 timeLeft['min']-=1  
 timeLeft['sec']=59  
 note.config(text='Time left: {}min:{}sec'.format(timeLeft['min'],timeLeft['sec']))  
 elif timeLeft['min']==0 and timeLeft['sec']==0:  
 print('Time up!')  
 result()  
 showtime.config(text=strftime('%H:%M:%S'))  
 showtime.after(1000,timeShow)  
  
#Attendance  
  
def getDetails():  
 global name,roll,mainWindow,Name,Roll  
 Name=name.get()  
 Roll=roll.get()  
 root.deiconify()  
 timeShow()  
 mainWindow.destroy()  
  
def attendance():  
 global name,roll,mainWindow  
 mainWindow=Toplevel(root)  
 mainWindow.geometry('700x500')  
 mainWindow.resizable(0,0)  
 mainWindow.title('QUIZ APP')  
 #mainWindow.tk.call(img)  
 mainWindow.config(bg='white')  
  
 #app name same as root  
  
 appName=Label(mainWindow,text=title,font=('impact',20,'italic'),  
 justify=CENTER,bg='blue',fg='black')  
 appName.pack(side=TOP,fill=BOTH)  
  
 #label to show info of attendance  
  
 info=Label(mainWindow,text="Enter Your Name and Rollno.",bg='white',fg='sky blue',font=('arial',15))  
 info.place(x=210,y=200)  
 name=Entry(mainWindow,width=30)  
 name.place(x=250,y=235)  
 roll=Entry(mainWindow,width=30)  
 roll.place(x=250,y=260)  
 name.insert(END,'Name')  
 roll.insert(END,'Roll')  
 submit=Button(mainWindow,text='Confirm & Start',width=20,bg='sky blue',fg='green',command=getDetails)  
 submit.place(x=265,y=300)  
 mainWindow.mainloop()  
  
#Toplevel finish  
  
#Quit quiz  
  
def quit\_function():  
 answer = showinfo(message="Try next time")  
 if answer=='ok':  
 sys.exit(root.destroy())  
  
#Disable all Button  
  
def dis\_button():  
 option1.config(state=DISABLED)  
 option2.config(state=DISABLED)  
 option3.config(state=DISABLED)  
 option4.config(state=DISABLED)  
  
#Enable all Button  
  
def en\_button():  
 option1.config(state=NORMAL)  
 option2.config(state=NORMAL)  
 option3.config(state=NORMAL)  
 option4.config(state=NORMAL)  
  
#Show final result  
  
def result():  
 global score,Name,Roll  
 root.withdraw()  
 top=Toplevel(root)  
 #top.tk.call('wm','iconphoto',top.\_w,PhotoImage(file=''))  
 top.geometry('200x100')  
 top.resizable(0,0)  
 top.title('QUIZ RESULT')  
 top.config(bg='sky blue')  
 top.protocol('WM\_DELETE\_WINDOW',quit\_function)  
 filename=Name+'\_'+Roll+'.txt'  
 data='\nStudent:'+Name+'\nRoll:'+Roll+'\nScore: '+str(score)+'\nCompleted quiz at : '+strftime('%d/%m/%y --%H:%M:%S')  
 with open(filename,'a') as file:  
 file.write(data)  
 label=Label(top,text='QUIZ OVER...\n Score: '+str(score),font=30,fg='black',bg='sky blue').place(x=50,y=25)  
 exitBtn=Button(top,text='Exit',width=10,bg='white',fg='green',command=quit\_function).place(x=50,y=70)  
 top.mainloop()  
  
#questions and corresponding answers  
  
questions={" What is the 16-bit compiler allowable range for integer constants?":'-32768 to 32767',  
 'What is required in each C program?':'at least one function.',  
 '''main(){   
 int i = 2;{   
 int i = 4, j = 5;   
 printf("%d %d", i, j);}   
 printf("%d %d", i, j);}''':'4525',  
 'What is a lint?':'Analyzing tool'}  
  
#separate Questions and answers from questions variable  
  
que=[]  
ans=[]  
for key,value in questions.items():  
 que.append(key)  
 ans.append(value)  
  
#corresponding answers with answers including at random  
  
options=[  
 ['-3.4e38 to 3.4e38','-32767 to 32768','-32668 to 32667',ans[0]],  
 [ans[1],'any function.','Input data','Output data'],  
 [ans[2],'2525','4545','NOT'],  
 ['C compiler','Interactive debugger',ans[3],'C interpreter']  
 ]  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
currentQ=''  
queNo=None  
currentA=''  
score=0  
qn=1  
var=StringVar()  
def \_next():  
 global currentQ,currentA,queNo,score,i,qn  
 i=0  
  
 #till last question is left  
  
 if len(que)>0:  
 currentQ=random.choice(que)  
 print(currentQ)  
 q=Label(root,text='Que. '+str(qn),font=('arial',10)).place(x=20,y=80)  
 qn+=1  
 #\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 queNo=que.index(currentQ)  
 print(options[queNo])  
 currentA=questions[currentQ]  
 #firstly change caption of button  
  
 submit.config(text='Next')  
  
 #print current question on quelabel  
  
 queLabel.config(text=currentQ,fg='green',height=6)  
  
 #print options for Questions on labels--- option 1,option2,.....  
  
 en\_button()  
 option1.config(text=options[queNo][0],bg='sky blue',value=options[queNo][0],bd=1,command=answer)  
 option2.config(text=options[queNo][1], bg='sky blue', value=options[queNo][1], bd=1, command=answer)  
 option3.config(text=options[queNo][2], bg='sky blue', value=options[queNo][2], bd=1, command=answer)  
 option4.config(text=options[queNo][3], bg='sky blue', value=options[queNo][3], bd=1, command=answer)  
  
 #remove question from list which are asked  
  
 que.remove(currentQ)  
 ans.remove(currentA)  
 options.remove(options[queNo])  
 elif len(que)==0:  
 result()  
def answer():  
 global currentQ,currentA,score  
  
 #print selected radiobutton  
  
 a= var.get()  
 if currentA==str(a):  
 score+=100  
 dis\_button()  
 else:  
 dis\_button()  
title='''C programming Questions'''  
appName=Label(root,text=title,font=('impact',20,'italic'),  
 justify=CENTER,bg='blue',fg='black')  
appName.pack(side=TOP,fill=BOTH)  
s=Separator(root).place(x=0,y=195,relwidth=1)  
#label to show current question  
  
queLabel=Label(root,text='',justify=LEFT,font=25)  
queLabel.pack(side=TOP,fill=BOTH)  
s=Separator(root).place(x=0,y=195,relwidth=1)  
  
#options label  
  
option1=Radiobutton(root,text='',bg='white',font=20,width=20,relief=FLAT,indicator=0,value=1,variable=var,bd=0)  
option1.place(x=150,y=250)  
option2=Radiobutton(root,text='',bg='white',font=20,width=20,relief=FLAT,indicator=0,value=2,variable=var,bd=0)  
option2.place(x=400,y=250)  
option3=Radiobutton(root,text='',bg='white',font=20,width=20,relief=FLAT,indicator=0,value=3,variable=var,bd=0)  
option3.place(x=150,y=300)  
option4=Radiobutton(root,text='',bg='white',font=20,width=20,relief=FLAT,indicator=0,value=4,variable=var,bd=0)  
option4.place(x=400,y=300)  
  
#instructions of Quiz  
  
instruction=Label(root,text=intro,bg='sky blue',fg='white',font=('calibri',15),justify=LEFT)  
instruction.place(x=150,y=200)  
  
#note to quiz  
  
note=Label(root,text='',font=('impact',10),bg='black',fg='green')  
note.pack(side=BOTTOM)  
  
#submit button  
  
submit=Button(root,text='Start Quiz',font=('impact',15),fg='white',bg='blue',state=DISABLED,command=\_next)  
submit.pack(side=BOTTOM)  
  
#show current time  
  
showtime=Label(root,text='',font=20,fg='black',bg='sky blue')  
showtime.place(x=620,y=50)  
  
#progress bar for time left for each question  
  
if \_\_name\_\_=="\_\_main\_\_":  
 root.withdraw()  
 attendance()  
 root.mainloop()

**Output:**

****

