A Report on

QUIZ APPLICATION USING PYTHON TKINTER

Submitted by

Anjaneya Mustoor (21GANSD001)

Gajendra M (21GANSD002)

5th Sem, ISE

Under the guidance of

Dr. Pushpa C N

Associate Professor

Dept. of CSE

UVCE

Department of Information Science and Engineering

UNIVERSITY VISVESVARAYA COLLEGE OF ENGINEERING

K.R. Circle, Bengaluru – 560 001

January-2023

Bangalore University

UNIVERSITY VISVESVARAYA COLLEGE OF ENGINEERING

K R Circle, Bengaluru-560 001

Department of Information Science and Engineering

CERTIFICATE

This is to certify that **Anjaneya Mustoor (21GANSD001), Gajendra M (21GANSD002)** has successfully completed the Mini-Project work entitled **Quiz Application Using Python Tkinter**, in partial fulfillment for the requirement of the **Python Programming** of 5th Semester Information Science and Engineering during the academic year 2022 - 2023.

Dr. Pushpa C N
Associate Professor
Department of CSE
UVCE

INTRODUCTION

A Quiz application using Python Tkinter would be a program that uses the Tkinter library to create a graphical user interface for a quiz. Tkinter is a built-in Python library for creating desktop applications, and it provides a set of tools for creating forms, buttons, labels, and other widgets.

To create a Quiz application using Tkinter, you could first design a layout for the quiz using Tkinter's widgets. This could include creating a window with a title, labels for the question and answer choices, and buttons for submitting answers. Then, you would need to write code to handle the logic of the quiz, such as displaying questions, checking answers, and keeping track of the user's score.

Python's Tkinter library provides a way to create graphical user interfaces (GUIs) for Python programs. To create a quiz application using Tkinter, you would first import the Tkinter module and create a window for the application using the Tk() function. You would then use various Tkinter widgets, such as labels, buttons, and entry fields, to create the layout for the quiz. To implement the quiz functionality, you would write event handlers for the buttons that check the user's answers and update the application accordingly.

Objectives:

1. To create a user-friendly interface for users to take a quiz.
2. To design and implement a logic for checking and scoring the answers provided by the user.
3. To display the results of the quiz to the user, including their score and potentially feedback on their answers.
4. To provide an easy way for the developer to add, update, or delete questions and answers from the quiz.
5. To potentially allow for multiple quizzes to be created and stored within the application.
6. To improve the user experience by adding features such as leaderboard, timer, and different types of question styles.

IMPLEMENTATION

This is an Implementation code that creates a GUI for an application with a title "Python-Assignment" and dimensions of 600x400 pixels. The application contains four buttons:

1. "Student Info" which on click shows basic info of students using a tkinter messagebox.
2. "Open Program" which on click opens a program quizapplication.py using os library
3. "Run the Program " which on click runs the function calledbytkinter() from quizapplication.py file and exits the application.
4. start Function: defining a function "start()" which creates a Tkinter root window and assigns it the title "Welcome To Quiz App". It then creates a canvas widget with a yellow background, places it in the first row, and loads an image named "output-onlinepngtools.png" and displays it on the canvas.
5. loginPage Function: provides a login page for a quiz app. The page allows the user to enter their username and password, which are then checked against a database of users.
6. signUpPage Function: The page allows the user to enter their full name, username, password, and country. Then it checks the field whether it's empty or not. If any of the fields is empty, It will show the error message. And If all the fields are filled,
7. Menu Function: When the user successfully logs in, the menu function is called and it allows the user to select the difficulty level of the quiz they want to take, either easy, medium, or hard. The user can select the level by clicking on the corresponding radiobutton.
8. Display Function: It creates a canvas and a frame on the window, and then defines several variables and lists that will be used for the questions and answers.

Code: (Main.py)

```
from tkinter import *
from tkinter import messagebox
import tkinter as tk
# import os
import subprocess

def open_my_info():
    messagebox.showinfo("Team Details","Name : Anjaneya Mustoor\n Reg no:
21GANSD001\n""Name :  Gajendra M\n Reg no: 21GANSD002\n" )

def open_program():
    # os.system("qui.py")
    text = tk.Text(root)
    text.pack()
    with open("main.py", "r") as f:
        program_text = f.read()
        text.insert(tk.END, program_text)

def run_program():
    subprocess.run(["python","qui.py"])

root=Tk()
root.title('Python-Assignment')
root.geometry("1000x1000+200+200")
Label(root,text="Quiz Application Using Python Tkinter",font=("Roboto",30)).pack()

Button(root,text="Student Details",height = "2", width = "30",font =
("Roboto",14),command=open_my_info).place(x=150,y=100)
Button(root,text="Open the Program",height = "2", width = "30",font =
("Roboto",14),command=open_program).place(x=150,y=200)
Button(root,text="Run the Program",height = "2", width = "30",font =
("Roboto",14),command=run_program).place(x=150,y=300)

root.focus_force()
root.mainloop()
```

Code(quizapplication.py):

```
import tkinter as tk
from tkinter import *
import random
import sqlite3
import time

def loginPage(logdata):
    sup.destroy()
    global login
    login = Tk()
    login.title('Quiz App Login')

    user_name = StringVar()
    password = StringVar()

    login_canvas = Canvas(login,width=720,height=440,bg="#B64D4D")
    login_canvas.pack()

    login_frame = Frame(login_canvas,bg="orange")
    login_frame.place(relwidth=0.8,relheight=0.8,relx=0.1,rely=0.1)

    heading = Label(login_frame,text="Quiz App Login",fg="white",bg="orange")
    heading.config(font=('calibri 40'))
    heading.place(relx=0.2,rely=0.1)

    #USER NAME
    ulabel = Label(login_frame,text="Username",fg='white',bg='black')
    ulabel.place(relx=0.21,rely=0.4)
    uname = Entry(login_frame,bg='white',fg='black',textvariable = user_name)
    uname.config(width=42)
    uname.place(relx=0.31,rely=0.4)

    #PASSWORD
    plabel = Label(login_frame,text="Password",fg='white',bg='black')
    plabel.place(relx=0.215,rely=0.5)
    pas = Entry(login_frame,bg='white',fg='black',textvariable = password,show="*")
    pas.config(width=42)
    pas.place(relx=0.31,rely=0.5)

    def check():
        for a,b,c,d in logdata:
            if b == uname.get() and c == pas.get():
                print(logdata)

                menu(a)
                break
        else:
            error = Label(login_frame,text="Wrong Username or Password!",fg='black',bg='white')
        error.place(relx=0.37,rely=0.7)

    #LOGIN BUTTON
    log =
Button(login_frame,text='Login',padx=5,pady=5,width=5,command=check,fg="white",bg="blac
k")
    log.configure(width = 15,height=1, activebackground = "#33B5E5", relief = FLAT)
    log.place(relx=0.4,rely=0.6)


    login.mainloop()

def signUpPage():
    root.destroy()
    global sup
    sup = Tk()
    sup.title('Quiz App')

    fname = StringVar()
    uname = StringVar()
    passW = StringVar()
    country = StringVar()


    sup_canvas = Canvas(sup,width=720,height=440,bg="#FFBC25")
    sup_canvas.pack()

    sup_frame = Frame(sup_canvas,bg="#BADA55")
    sup_frame.place(relwidth=0.8,relheight=0.8,relx=0.1,rely=0.1)

    heading = Label(sup_frame,text="Quiz App SignUp",fg="#FFA500",bg="#BADA55")
    heading.config(font=('calibri 40'))
    heading.place(relx=0.2,rely=0.1)

    #full name
    flabel = Label(sup_frame,text="Full Name",fg='white',bg='black')
    flabel.place(relx=0.21,rely=0.4)
    fname = Entry(sup_frame,bg='white',fg='black',textvariable = fname)
    fname.config(width=42)
    fname.place(relx=0.31,rely=0.4)

    #username
    ulabel = Label(sup_frame,text="Username",fg='white',bg='black')
    ulabel.place(relx=0.21,rely=0.5)
    user = Entry(sup_frame,bg='white',fg='black',textvariable = uname)
    user.config(width=42)
    user.place(relx=0.31,rely=0.5)
```

```
#password
plabel = Label(sup_frame,text="Password",fg='white',bg='black')
plabel.place(relx=0.215,rely=0.6)
pas = Entry(sup_frame,bg='white',fg='black',textvariable = passW,show="*")
pas.config(width=42)
pas.place(relx=0.31,rely=0.6)



#country
clabel = Label(sup_frame,text="Country",fg='white',bg='black')
clabel.place(relx=0.217,rely=0.7)
c = Entry(sup_frame,bg='white',fg='black',textvariable = country)
c.config(width=42)
c.place(relx=0.31,rely=0.7)
def addUserToDataBase():

    fullname = fname.get()
    username = user.get()
    password = pas.get()
    country = c.get()

    if len(fname.get())==0 and len(user.get())==0 and len(pas.get())==0 and len(c.get())==0:
        error = Label(text="You haven't enter any field...Please Enter all the
fields",fg='black',bg='white')
        error.place(relx=0.37,rely=0.7)

    elif len(fname.get())==0 or len(user.get())==0 or len(pas.get())==0 or len(c.get())==0:
        error = Label(text="Please Enter all the fields",fg='black',bg='white')
        error.place(relx=0.37,rely=0.7)

    elif len(user.get()) == 0 and len(pas.get()) == 0:
        error = Label(text="Username and password can't be empty",fg='black',bg='white')
        error.place(relx=0.37,rely=0.7)

    elif len(user.get()) == 0 and len(pas.get()) != 0 :
        error = Label(text="Username can't be empty",fg='black',bg='white')
        error.place(relx=0.37,rely=0.7)

    elif len(user.get()) != 0 and len(pas.get()) == 0:
        error = Label(text="Password can't be empty",fg='black',bg='white')
        error.place(relx=0.37,rely=0.7)

    else:

        conn = sqlite3.connect('quiz.db')
        create = conn.cursor()
```

```
            create.execute('CREATE TABLE IF NOT EXISTS userSignUp(FULLNAME text,
USERNAME text,PASSWORD text,COUNTRY text)')
            create.execute("INSERT INTO userSignUp VALUES
(?,?,?,?)",(fullname,username,password,country))
            conn.commit()
            create.execute('SELECT * FROM userSignUp')
            z=create.fetchall()
            print(z)
            #L2.config(text="Username is "+z[0][0]+"\nPassword is "+z[-1][1])
            conn.close()
            loginPage(z)

    def gotoLogin():
        conn = sqlite3.connect('quiz.db')
        create = conn.cursor()
        conn.commit()
        create.execute('SELECT * FROM userSignUp')
        z=create.fetchall()
        loginPage(z)

    #signup BUTTON
    sp = Button(sup_frame,text='SignUp',padx=5,pady=5,width=5,command =
addUserToDataBase, bg="black",fg="white")
    sp.configure(width = 15,height=1, activebackground = "#33B5E5", relief = FLAT)
    sp.place(relx=0.4,rely=0.8)

    log = Button(sup_frame,text='Already have a Account?',padx=5,pady=5,width=5,command
= gotoLogin,bg="#BADA55", fg="black")
    log.configure(width = 16,height=1, activebackground = "#33B5E5", relief = FLAT)
    log.place(relx=0.393,rely=0.9)

    sup.mainloop()

def menu(abcdefgh):
    login.destroy()
    global menu
    menu = Tk()
    menu.title('Quiz App Menu')


    menu_canvas = Canvas(menu,width=720,height=440,bg="orange")
    menu_canvas.pack()

    menu_frame = Frame(menu_canvas,bg="#7FFFD4")
    menu_frame.place(relwidth=0.8,relheight=0.8,relx=0.1,rely=0.1)
    wel = Label(menu_canvas,text=' W E L C O M E  T O  Q U I Z  ',fg="white",bg="orange")
    wel.config(font=('Broadway 22'))
    wel.place(relx=0.1,rely=0.02)

    abcdefgh='Welcome '+ abcdefgh
    level34 = Label(menu_frame,text=abcdefgh,bg="black",font="calibri 18",fg="white")
    level34.place(relx=0.17,rely=0.15)

    level = Label(menu_frame,text='Select your Difficulty Level !!',bg="orange",font="calibri 18")
    level.place(relx=0.25,rely=0.3)


    var = IntVar()
    easyR = Radiobutton(menu_frame,text='Easy',bg="#7FFFD4",font="calibri
16",value=1,variable = var)
    easyR.place(relx=0.25,rely=0.4)

    mediumR = Radiobutton(menu_frame,text='Medium',bg="#7FFFD4",font="calibri
16",value=2,variable = var)
    mediumR.place(relx=0.25,rely=0.5)

    hardR = Radiobutton(menu_frame,text='Hard',bg="#7FFFD4",font="calibri
16",value=3,variable = var)
    hardR.place(relx=0.25,rely=0.6)


    def navigate():

        x = var.get()
        print(x)
        if x == 1:
            menu.destroy()
            easy()
        elif x == 2:
            menu.destroy()
            medium()

        elif x == 3:
            menu.destroy()
            difficult()
        else:
            pass
    letsgo = Button(menu_frame,text="Let's Go",bg="black",fg="white",font="calibri
12",command=navigate)
    letsgo.place(relx=0.25,rely=0.8)
    menu.mainloop()
def easy():
    global e
    e = Tk()
    e.title('Quiz App - Easy Level')

    easy_canvas = Canvas(e,width=720,height=440,bg="orange")
    easy_canvas.pack()

    easy_frame = Frame(easy_canvas,bg="#BADA55")
    easy_frame.place(relwidth=0.8,relheight=0.8,relx=0.1,rely=0.1)


    def countDown():
        check = 0
        for k in range(10, 0, -1):

            if k == 1:
                check=-1
            timer.configure(text=k)
            easy_frame.update()
            time.sleep(1)

        timer.configure(text="Times up!")
        if check==-1:
            return (-1)
        else:
            return 0
    global score
    score = 0

    easyQ = [
                [
                    "What will be the output of the following Python code? \nl=[1, 0, 2, 0, 'hello', '', []]
\nlist(filter(bool, nl))",
                    "[1, 0, 2, 'hello', '', []]",
                    "Error",
                    "[1, 2, 'hello']",
                    "[1, 0, 2, 0, 'hello', '', []]"
                ] ,
                [
                    "What will be the output of the following Python expression if the value of x is 34?
\nprint("%f"%x)" ,
                   "34.00",
                   "34.000000",
                   "34.0000",
                   "34.00000000"

                ],
               [
                   "What will be the value of X in the following Python expression? \nX =
2+9*((3*12)-8)/10" ,
                   "30.8",
                   "27.2",
                   "28.4",
                   "30.0"
               ],
               [
                   "Which of these in not a core data type?" ,
                   "Tuples",
                   "Dictionary",
                   "Lists",
                   "Class"
               ],
               [
                   "Which of the following represents the bitwise XOR operator?" ,
                   "&",
                   "!",
                   "^",
                   "|"
               ]
            ]
    answer = [
               "[1, 2, ‘hello’]",
               "34.000000",
               "27.2",
               "Class",
               "^"
            ]
    li = ['',0,1,2,3,4]
    x = random.choice(li[1:])

    ques = Label(easy_frame,text =easyQ[x][0],font="calibri 12",bg="orange")
    ques.place(relx=0.5,rely=0.2,anchor=CENTER)

    var = StringVar()

    a = Radiobutton(easy_frame,text=easyQ[x][1],font="calibri 10",value=easyQ[x][1],variable =
var,bg="#BADA55")
    a.place(relx=0.5,rely=0.42,anchor=CENTER)

    b = Radiobutton(easy_frame,text=easyQ[x][2],font="calibri 10",value=easyQ[x][2],variable =
var,bg="#BADA55")
    b.place(relx=0.5,rely=0.52,anchor=CENTER)

    c = Radiobutton(easy_frame,text=easyQ[x][3],font="calibri 10",value=easyQ[x][3],variable =
var,bg="#BADA55")
    c.place(relx=0.5,rely=0.62,anchor=CENTER)
```

```
  d = Radiobutton(easy_frame,text=easyQ[x][4],font="calibri 10",value=easyQ[x][4],variable =
var,bg="#BADA55")
  d.place(relx=0.5,rely=0.72,anchor=CENTER)

  li.remove(x)

  timer = Label(e)
  timer.place(relx=0.8,rely=0.82,anchor=CENTER)



  def display():

    if len(li) == 1:
        e.destroy()
        showMark(score)
    if len(li) == 2:
      nextQuestion.configure(text='End',command=calc)

    if li:
      x = random.choice(li[1:])
      ques.configure(text =easyQ[x][0])

      a.configure(text=easyQ[x][1],value=easyQ[x][1])

      b.configure(text=easyQ[x][2],value=easyQ[x][2])

      c.configure(text=easyQ[x][3],value=easyQ[x][3])

      d.configure(text=easyQ[x][4],value=easyQ[x][4])

      li.remove(x)
      y = countDown()
      if y == -1:
        display()


  def calc():
    global score
    if (var.get() in answer):
      score+=1
    display()

  submit = Button(easy_frame,command=calc,text="Submit", fg="white", bg="black")
  submit.place(relx=0.5,rely=0.82,anchor=CENTER)

  nextQuestion = Button(easy_frame,command=display,text="Next", fg="white", bg="black")
```

```
    nextQuestion.place(relx=0.87,rely=0.82,anchor=CENTER)

    # pre=Button(easy_frame,command=display, text="Previous", fg="white", bg="black")
    # pre.place(relx=0.75, rely=0.82, anchor=CENTER)

    y = countDown()
    if y == -1:
        display()
    e.mainloop()


def medium():

    global m
    m = Tk()
    m.title('Quiz App - Medium Level')

    med_canvas = Canvas(m,width=720,height=440,bg="#101357")
    med_canvas.pack()

    med_frame = Frame(med_canvas,bg="#A1A100")
    med_frame.place(relwidth=0.8,relheight=0.8,relx=0.1,rely=0.1)


    def countDown():
        check = 0
        for k in range(10, 0, -1):

            if k == 1:
                check=-1
            timer.configure(text=k)
            med_frame.update()
            time.sleep(1)

        timer.configure(text="Times up!")
        if check==-1:
            return (-1)
        else:
            return 0

    global score
    score = 0

    mediumQ = [
              [
                "Which of the following is not an exception handling keyword in Python?",
                 "accept",
                 "finally",
                "except",
                "try"
            ],
            [
                "Suppose list1 is [3, 5, 25, 1, 3], what is min(list1)?",
                "3",
                "5",
                "25",
                "1"
            ],
            [
                "Suppose list1 is [2, 33, 222, 14, 25], What is list1[-1]?",
                "Error",
                "None",
                "25",
                "2"
            ],
            [
                "print(0xA + 0xB + 0xC):",
                "0xA0xB0xC",
                "Error",
                "0x22",
                "33"
            ],
            [
                "Which of the following is invalid?",
                "_a = 1",
                "__a = 1",
                "__str__ = 1",
                "none of the mentioned"
            ],
        ]
answer = [
        "accept",
        "1",
        "25",
        "33",
        "none of the mentioned"
        ]

li = ['',0,1,2,3,4]
x = random.choice(li[1:])

ques = Label(med_frame,text =mediumQ[x][0],font="calibri 12",bg="#B26500")
ques.place(relx=0.5,rely=0.2,anchor=CENTER)

var = StringVar()
```

```
   a = Radiobutton(med_frame,text=mediumQ[x][1],font="calibri
10",value=mediumQ[x][1],variable = var,bg="#A1A100")
   a.place(relx=0.5,rely=0.42,anchor=CENTER)

   b = Radiobutton(med_frame,text=mediumQ[x][2],font="calibri
10",value=mediumQ[x][2],variable = var,bg="#A1A100")
   b.place(relx=0.5,rely=0.52,anchor=CENTER)

   c = Radiobutton(med_frame,text=mediumQ[x][3],font="calibri
10",value=mediumQ[x][3],variable = var,bg="#A1A100")
   c.place(relx=0.5,rely=0.62,anchor=CENTER)

   d = Radiobutton(med_frame,text=mediumQ[x][4],font="calibri
10",value=mediumQ[x][4],variable = var,bg="#A1A100")
   d.place(relx=0.5,rely=0.72,anchor=CENTER)

   li.remove(x)

   timer = Label(m)
   timer.place(relx=0.8,rely=0.82,anchor=CENTER)



   def display():

      if len(li) == 1:
            m.destroy()
            showMark(score)
      if len(li) == 2:
         nextQuestion.configure(text='End',command=calc)

      if li:
         x = random.choice(li[1:])
         ques.configure(text =mediumQ[x][0])

         a.configure(text=mediumQ[x][1],value=mediumQ[x][1])

         b.configure(text=mediumQ[x][2],value=mediumQ[x][2])

         c.configure(text=mediumQ[x][3],value=mediumQ[x][3])

         d.configure(text=mediumQ[x][4],value=mediumQ[x][4])

         li.remove(x)
         y = countDown()
         if y == -1:
            display()
```

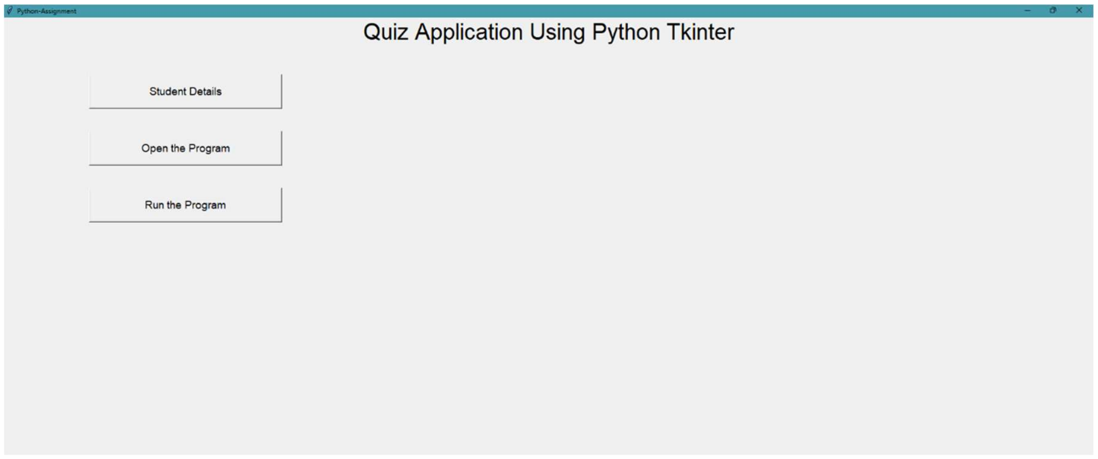
```
    def calc():
        global score
        if (var.get() in answer):
            score+=1
        display()

    submit = Button(med_frame,command=calc,text="Submit", fg="white", bg="black")
    submit.place(relx=0.5,rely=0.82,anchor=CENTER)

    nextQuestion = Button(med_frame,command=display,text="Next", fg="white", bg="black")
    nextQuestion.place(relx=0.87,rely=0.82,anchor=CENTER)

   # pre=Button(med_frame,command=display, text="Previous", fg="white", bg="black")
   # pre.place(relx=0.75, rely=0.82, anchor=CENTER)

    y = countDown()
    if y == -1:
        display()
    m.mainloop()
def difficult():


    global h
    #count=0
    h = Tk()
    h.title('Quiz App - Hard Level')

    hard_canvas = Canvas(h,width=720,height=440,bg="#101357")
    hard_canvas.pack()

    hard_frame = Frame(hard_canvas,bg="#008080")
    hard_frame.place(relwidth=0.8,relheight=0.8,relx=0.1,rely=0.1)


    def countDown():
        check = 0
        for k in range(10, 0, -1):

            if k == 1:
                check=-1
            timer.configure(text=k)
            hard_frame.update()
            time.sleep(1)

        timer.configure(text="Times up!")
        if check==-1:
            return (-1)
```

```
      else:
         return 0

   global score
   score = 0

   hardQ = [
            [
     "All keywords in Python are in _________",
      "lower case",
      "UPPER CASE",
      "Capitalized",
      "None of the mentioned"
   ],
   [
      "Which of the following cannot be a variable?",
      "__init__",
      "in",
      "it",
      "on"
   ],
   [
    "Which of the following is a Python tuple?",
      "[1, 2, 3]",
      "(1, 2, 3)",
      "{1, 2, 3}",
      "{}"
   ],
   [
      "What is returned by math.ceil(3.4)?",
      "3",
      "4",
      "4.0",
      "3.0"
   ],
   [
      "What will be the output of print(math.factorial(4.5))?",
      "24",
      "120",
      "error",
      "24.0"
   ]

]
   answer = [
         "None of the mentioned",
         "in",
         "(1,2,3)",
         "4",
         "error"
         ]

   li = ['',0,1,2,3,4]
   x = random.choice(li[1:])

   ques = Label(hard_frame,text =hardQ[x][0],font="calibri 12",bg="#A0DB8E")
   ques.place(relx=0.5,rely=0.2,anchor=CENTER)

   var = StringVar()

   a = Radiobutton(hard_frame,text=hardQ[x][1],font="calibri 10",value=hardQ[x][1],variable =
var,bg="#008080",fg="white")
   a.place(relx=0.5,rely=0.42,anchor=CENTER)

   b = Radiobutton(hard_frame,text=hardQ[x][2],font="calibri 10",value=hardQ[x][2],variable =
var,bg="#008080",fg="white")
   b.place(relx=0.5,rely=0.52,anchor=CENTER)

   c = Radiobutton(hard_frame,text=hardQ[x][3],font="calibri 10",value=hardQ[x][3],variable =
var,bg="#008080",fg="white")
   c.place(relx=0.5,rely=0.62,anchor=CENTER)

   d = Radiobutton(hard_frame,text=hardQ[x][4],font="calibri 10",value=hardQ[x][4],variable =
var,bg="#008080",fg="white")
   d.place(relx=0.5,rely=0.72,anchor=CENTER)

   li.remove(x)

   timer = Label(h)
   timer.place(relx=0.8,rely=0.82,anchor=CENTER)

   def display():

      if len(li) == 1:
            h.destroy()
            showMark(score)
      if len(li) == 2:
         nextQuestion.configure(text='End',command=calc)

      if li:
         x = random.choice(li[1:])
         ques.configure(text =hardQ[x][0])

         a.configure(text=hardQ[x][1],value=hardQ[x][1])

         b.configure(text=hardQ[x][2],value=hardQ[x][2])
```

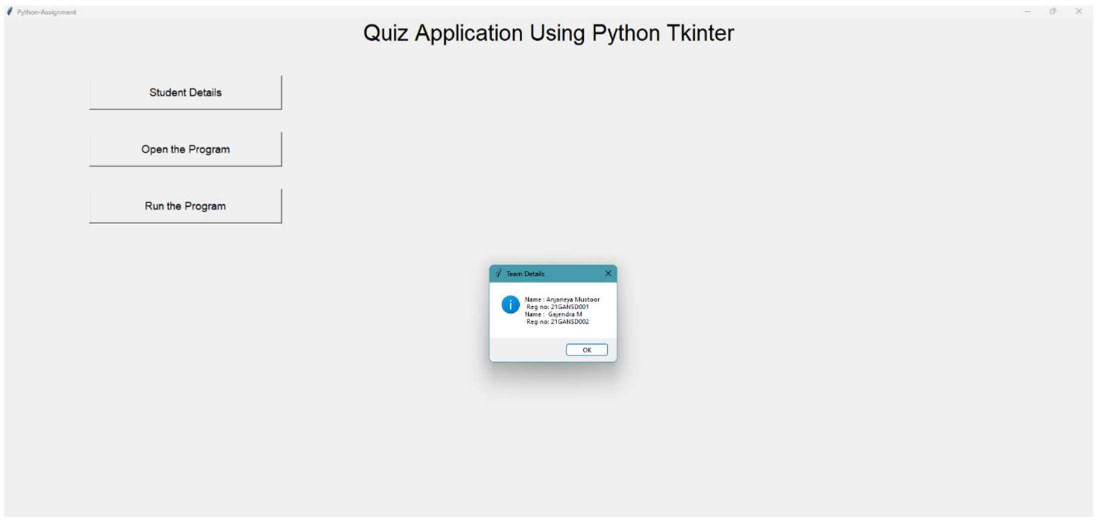
```
        c.configure(text=hardQ[x][3],value=hardQ[x][3])

        d.configure(text=hardQ[x][4],value=hardQ[x][4])

        li.remove(x)
        y = countDown()
        if y == -1:
            display()


    def calc():
        global score
        #count=count+1
        if (var.get() in answer):
            score+=1
        display()


    submit = Button(hard_frame,command=calc,text="Submit", fg="white", bg="black")
    submit.place(relx=0.5,rely=0.82,anchor=CENTER)

    nextQuestion = Button(hard_frame,command=display,text="Next", fg="white", bg="black")
    nextQuestion.place(relx=0.87,rely=0.82,anchor=CENTER)

    #pre=Button(hard_frame,command=display, text="Previous", fg="white", bg="black")
    #pre.place(relx=0.75, rely=0.82, anchor=CENTER)

   # end=Button(hard_frame,command=showMark(m), text="End", fg="white", bg="black")
    # end.place(relx=0.8, rely=0.82, anchor=CENTER)

    y = countDown()
    if y == -1:
        display()
    h.mainloop()

def showMark(mark):
    sh = Tk()
    sh.title('Your Marks')

    st = "Your score is "+str(mark)+"/5"
    mlabel = Label(sh,text=st,fg="black", bg="white")
    mlabel.pack()

    def callsignUpPage():
        sh.destroy()
        start()
    def myeasy():
        sh.destroy()
        easy()

    b24=Button(text="Re-attempt", command=myeasy, bg="black", fg="white")
    b24.pack()

    from matplotlib.backends.backend_tkagg import (FigureCanvasTkAgg,
NavigationToolbar2Tk)
    from matplotlib.backend_bases import key_press_handler
    from matplotlib.figure import Figure

    import numpy as np

    fig = Figure(figsize=(5, 4), dpi=100)
    labels = 'Marks Obtained','Total Marks'
    sizes = [int(mark),5-int(mark)]
    explode = (0.1,0)
    fig.add_subplot(111).pie(sizes, explode=explode, labels=labels,
autopct='%1.1f%%',shadow=True, startangle=0)


    canvas = FigureCanvasTkAgg(fig, master=sh)  # A tk.DrawingArea.
    canvas.draw()
    canvas.get_tk_widget().pack(side=TOP, fill=BOTH, expand=1)

    b23=Button(text="Sign Out",command=callsignUpPage,fg="white", bg="black")
    b23.pack()

    sh.mainloop()
def start():
    global root
    root = Tk()
    root.title('Welcome To Quiz App')
    canvas = Canvas(root,width = 720,height = 640, bg = 'yellow')
    canvas.grid(column = 0 , row = 1)
    img = PhotoImage(file="output-onlinepngtools.png")
    canvas.create_image(50,10,image=img,anchor=NW)

    button = Button(root, text='Start',command = signUpPage,bg="red",fg="yellow")
    button.configure(width = 102,height=2, activebackground = "#33B5E5", relief = RAISED)
    button.grid(column = 0 , row = 2)

    root.mainloop()


if __name__=='__main__':
    start()
```

SCREENSHOTS

1. Main Screen

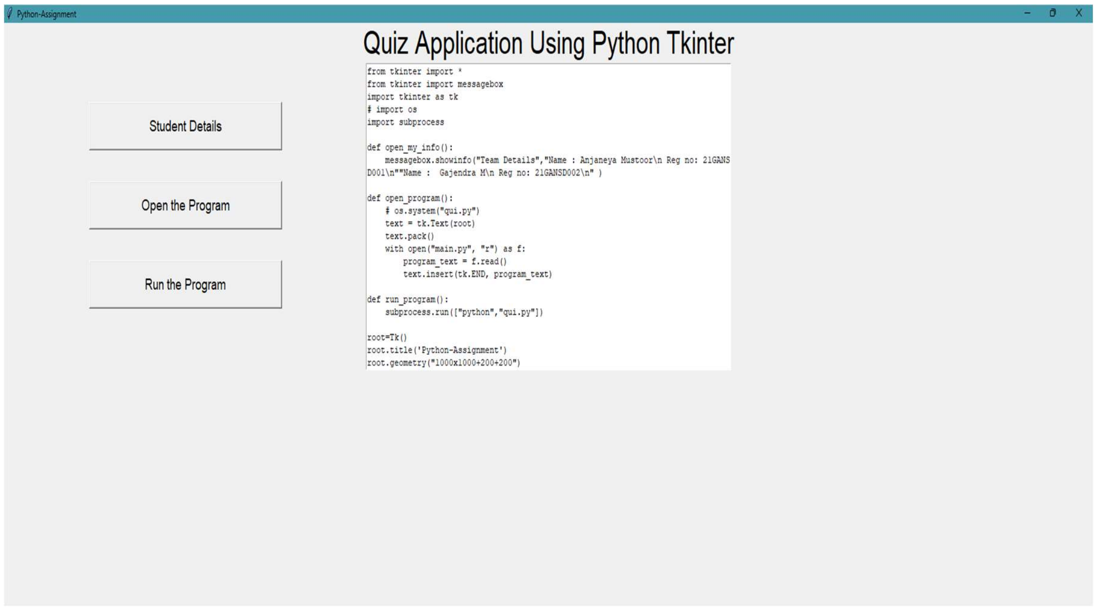


2. Student Details



Name : Anjaneya Mustoor
Reg no: 21GANSD001
Name : Gajendra M
Reg no: 21GANSD002

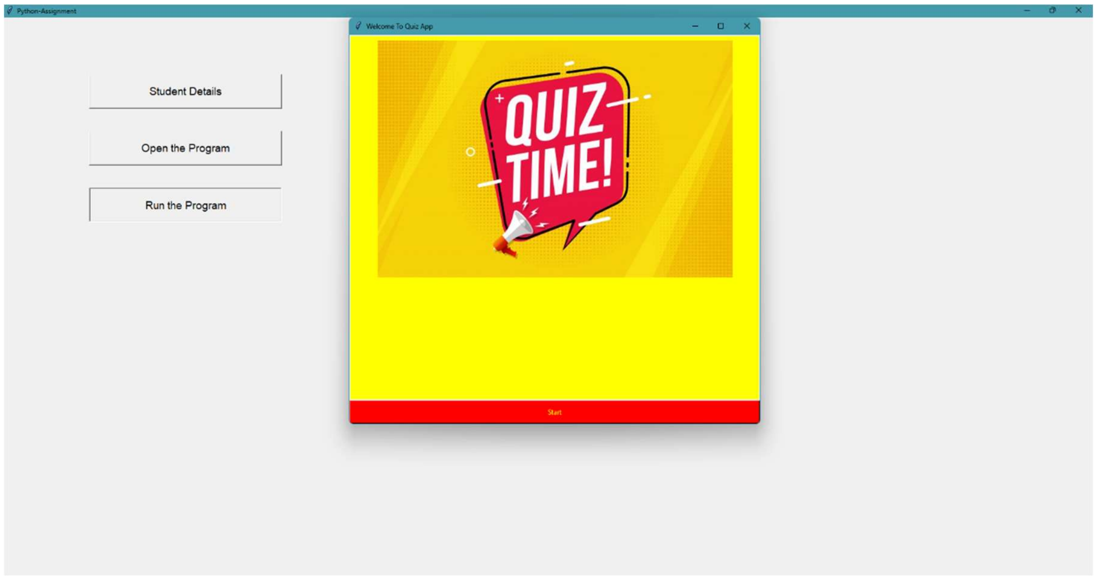
3. Open Program



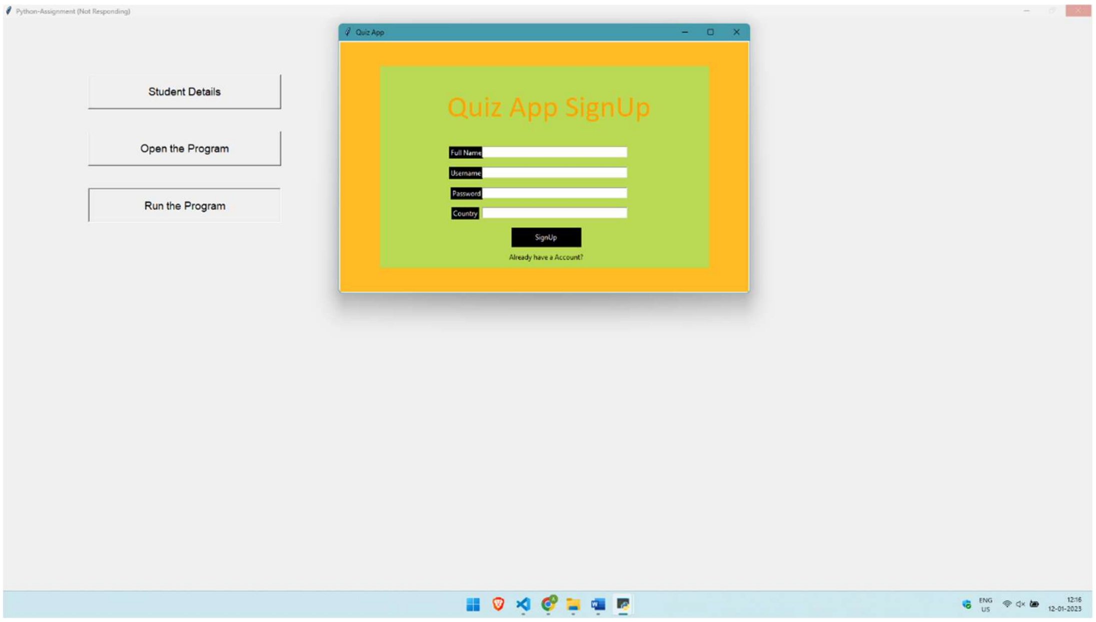
Quiz Application Using Python Tkinter

```
from tkinter import *
from tkinter import messagebox
import tkinter as tk
# import os
import subprocess

def open_my_info():
    messagebox.showinfo("Team Details","Name : Anjaneya Mustoor\n Reg no: 21GANS
D001\n""Name :  Gajendra M\n Reg no: 21GANSD002\n" )

def open_program():
    # os.system("qui.py")
    text = tk.Text(root)
    text.pack()
    with open("main.py", "r") as f:
        program_text = f.read()
        text.insert(tk.END, program_text)

def run_program():
    subprocess.run(["python","qui.py"])

root=Tk()
root.title('Python-Assignment')
root.geometry("1000x1000+200+200")
```

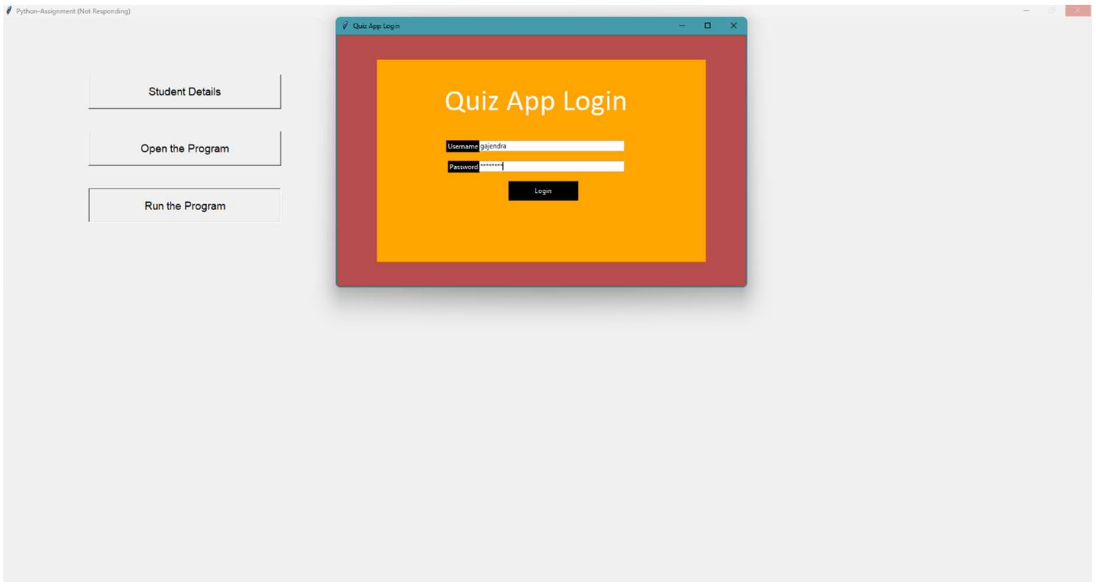
4. Run Program



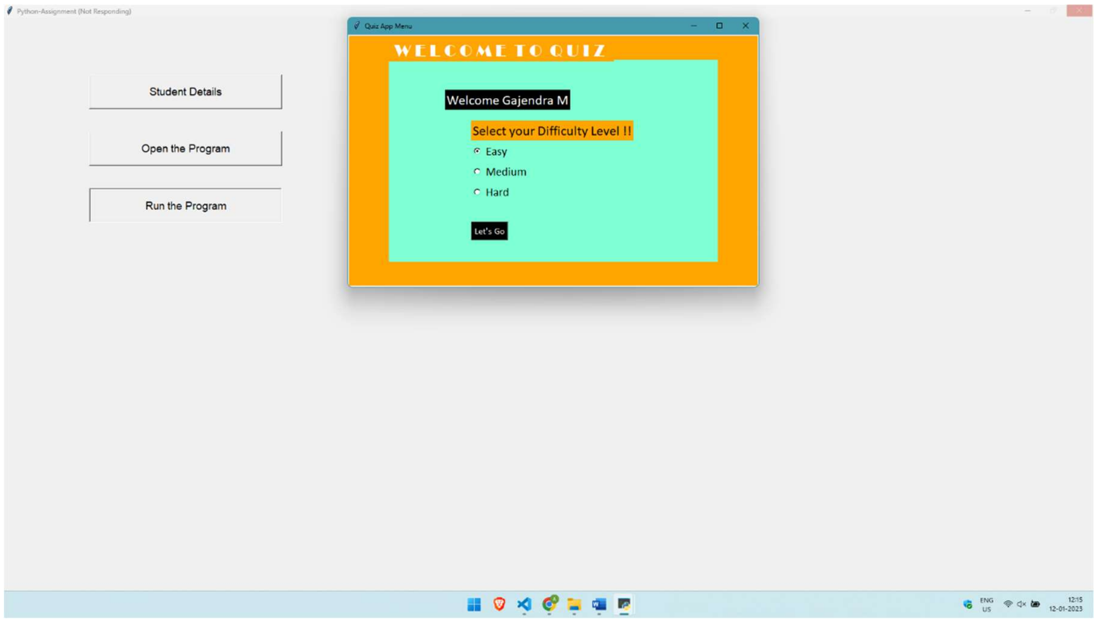
5. Signup Screen



6. Login Screen



7. Options for the User

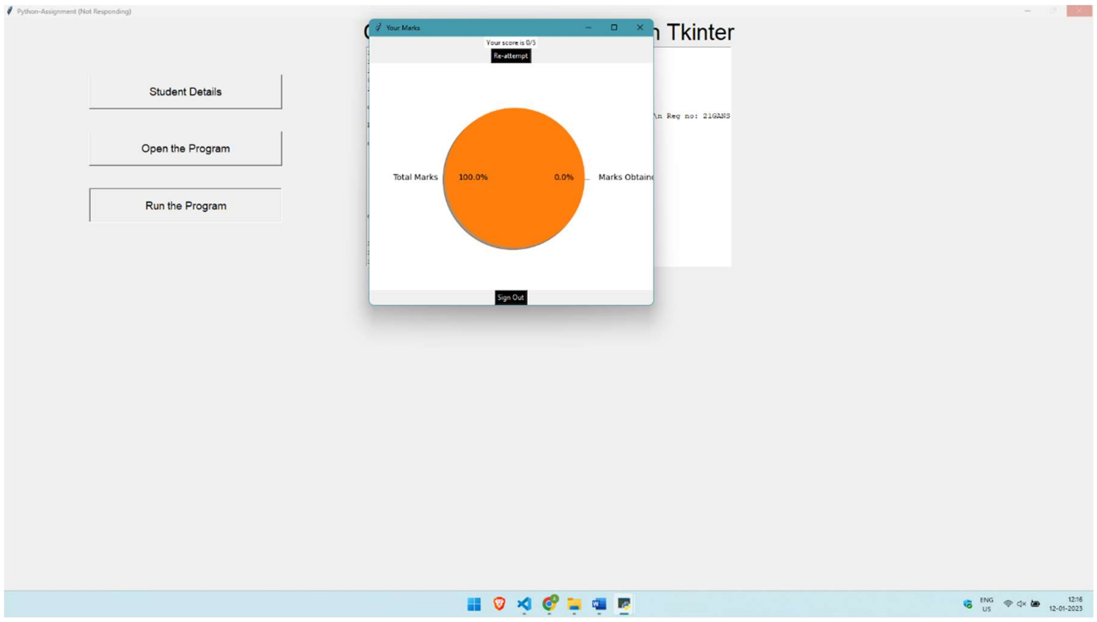


Welcome Gajendra M

Select your Difficulty Level !!

- [✓ Easy]
- Medium
- Hard

8. Results of Quiz



Your score is 0/5

```
\n Reg no: 21GANS
```

References:

1. https://docs.python.org/3/library/tkinter.html
2. https://docs.python.org/
3. https://www.geeksforgeeks.org/
4. https://www.google.com/
5. https://www.tutorialspoint.com/python/python_gui_programming.html