Department of Computer Engineering

PYTHON MINI PROJECT REPORT

on

" PYTHON QUIZ APPLICATION WITH TKINTER AND SQLITE DATABASE"

By

121A1097 Serin Kaushik

121A1098 Aayush Shah

121A1101 Dhruv Shetty

121A1113 Devdatta Temgire

DEPARTMENT OF COMPUTER ENGINEERING

SIES GRADUATE SCHOOL OF TECHNOLOGY

NERUL, NAVI MUMBAI – 400706

ACADEMIC YEAR

2022 – 2023

CONTENTS

1. Abstract:

In this project, we have developed a quiz application using the Python programming language and the Tkinter GUI library. The application allows users to log in with a password and then attempt a quiz. Each question has a time limit of 10 seconds, and the user's score is displayed at the end of the quiz. The application also uses an SQLite database to store user signup data. The main objective of this project is to create an interactive quiz application that is easy to use and provides a fun learning experience. By using the Tkinter library, we were able to create a graphical user interface that is simple, yet elegant. The use of the SQLite database allowed us to store user information securely and efficiently.

2. Introduction:

The quiz application is an innovative and user-friendly project that offers a unique and interactive experience to users. The purpose of this project is to provide a platform where users can test their knowledge and skills in various subject areas. The application is designed to cater to a broad audience, including students, educators, and general knowledge enthusiasts.

The primary goal of the quiz application is to create an engaging and informative experience for users. To achieve this, the application is designed to be easy to use, visually appealing, and feature rich. The user interface is intuitive and user-friendly, providing users with a seamless experience. The application offers a wide range of quizzes covering various subjects, making it an excellent learning tool.

The project is built using the Python programming language and the Tkinter GUI library. Python is a powerful and versatile language that is widely used in the development of desktop applications. It is known for its simplicity, readability, and ease of use. Tkinter, on the other hand, is a standard GUI library for Python that provides a robust set of tools for building interactive graphical user interfaces.

The quiz application features an intuitive and user-friendly interface that is designed to be easy to navigate. The application allows users to sign up using a password, after which they can access the quiz section. Each quiz has a ten-second time limit per question, adding an exciting and challenging element to the application. The application also features a score system that calculates and displays the user's performance, providing instant feedback.

3. Detailed System Design:

The detailed system design section should provide a detailed description of the system design, including the software architecture, user interface design, and database schema. In this section, we will discuss the key components of the quiz application and how they are connected. The quiz application consists of three main components: the user interface, the quiz engine, and the database. The user interface is designed using the Tkinter library and provides an intuitive interface for users to navigate through the quiz. The quiz engine is responsible for generating questions and tracking the user's progress throughout the quiz. The database is used to store user information and quiz data.

The user interface is designed to be simple and easy to use. The main screen of the application displays a login form that users can use to sign into the application. Once logged in, users are taken to the quiz screen, where they can select a quiz category and start the quiz. The quiz screen displays one question at a time and provides users with a 10-second time limit to answer each question. Once the user has answered all the questions, their score is displayed on the screen.

The quiz engine is responsible for generating questions and tracking the user's progress. The quiz engine uses a set of pre-defined questions and answers to generate quizzes for users. The engine tracks the user's progress throughout the quiz and provides instant feedback on their performance.

The database is used to store user information and quiz data. The database schema consists of two tables: the users table and the quiz table. The users table stores user information such as their name, email address, and password. The quiz table stores quiz data such as the quiz category, the user's score, and the date and time the quiz was taken.

Flowchart for the system:

CODE :

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

def loginPage(logdata):
    sup.destroy()
    global login
    login = Tk()

    user_name = StringVar()
    password = StringVar()

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

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

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

    #USER NAME
    ulabel = Label(login_frame,text="Username",fg='black',bg='white')
    ulabel.place(relx=0.21,rely=0.4)
    uname = Entry(login_frame,bg='#d3d3d3',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='black',bg='white')
    plabel.place(relx=0.215,rely=0.5)
    pas = Entry(login_frame,bg='#d3d3d3',fg='black',show="*",textvariable = password)
    pas.config(width=42)
    pas.place(relx=0.31,rely=0.5)

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

```
    fname = StringVar()
    uname = StringVar()
    passW = StringVar()



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

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

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

    #full name
    flabel = Label(sup_frame,text="Full Name",fg='black',bg='white')
    flabel.place(relx=0.21,rely=0.4)
    fname = Entry(sup_frame,bg='#d3d3d3',fg='black',textvariable = fname)
    fname.config(width=42)
    fname.place(relx=0.31,rely=0.4)

    #username
    ulabel = Label(sup_frame,text="Username",fg='black',bg='white')
    ulabel.place(relx=0.21,rely=0.5)
    user = Entry(sup_frame,bg='#d3d3d3',fg='black',textvariable = uname)
    user.config(width=42)
    user.place(relx=0.31,rely=0.5)


    #password
    plabel = Label(sup_frame,text="Password",fg='black',bg='white')
    plabel.place(relx=0.215,rely=0.6)
    pas = Entry(sup_frame,bg='#d3d3d3',fg='black',show="*",textvariable = passW)
    pas.config(width=42)
    pas.place(relx=0.31,rely=0.6)




    def addUserToDataBase():

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


        conn = sqlite3.connect('quiz1.db')
        create = conn.cursor()
        create.execute('CREATE TABLE IF NOT EXISTS userSignUp(FULLNAME text, USERNAME
text,PASSWORD text)')
        create.execute("INSERT INTO userSignUp VALUES (?,?,?)",(fullname,username,password))
        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('quiz1.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='green')
    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="white",fg='blue')
    log.configure(width = 16,height=1, activebackground = "#33B5E5", relief = FLAT)
    log.place(relx=0.4,rely=0.9)

    sup.mainloop()

def menu():
    login.destroy()
    global menu
    menu = Tk()


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

    menu_frame = Frame(menu_canvas,bg="white")
    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  S T A T I O N ',fg="white",bg="#101357")
    wel.config(font=('Broadway 22'))
    wel.place(relx=0.1,rely=0.02)


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


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

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

    hardR = Radiobutton(menu_frame,text='Hard',bg="white",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="white",font="calibri 12",command=navigate)
    letsgo.place(relx=0.25,rely=0.8)
    menu.mainloop()
def easy():

    global e
    e = Tk()

    easy_canvas = Canvas(e,width=720,height=440,bg="#101357")
    easy_canvas.pack()

    easy_frame = Frame(easy_canvas,bg="white")
    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="white")
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="white")
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="white")
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="white")
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="white")
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)
            print(li)
            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")
    submit.place(relx=0.5,rely=0.82,anchor=CENTER)

    nextQuestion = Button(easy_frame,command=display,text="Next")
    nextQuestion.place(relx=0.87,rely=0.82,anchor=CENTER)

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


def medium():

    global m
    m = Tk()

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

    med_frame = Frame(med_canvas,bg="white")
    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="white")
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="white")
   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="white")
   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="white")
   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="white")
   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)
         print(li)
         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")
   submit.place(relx=0.5,rely=0.82,anchor=CENTER)

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

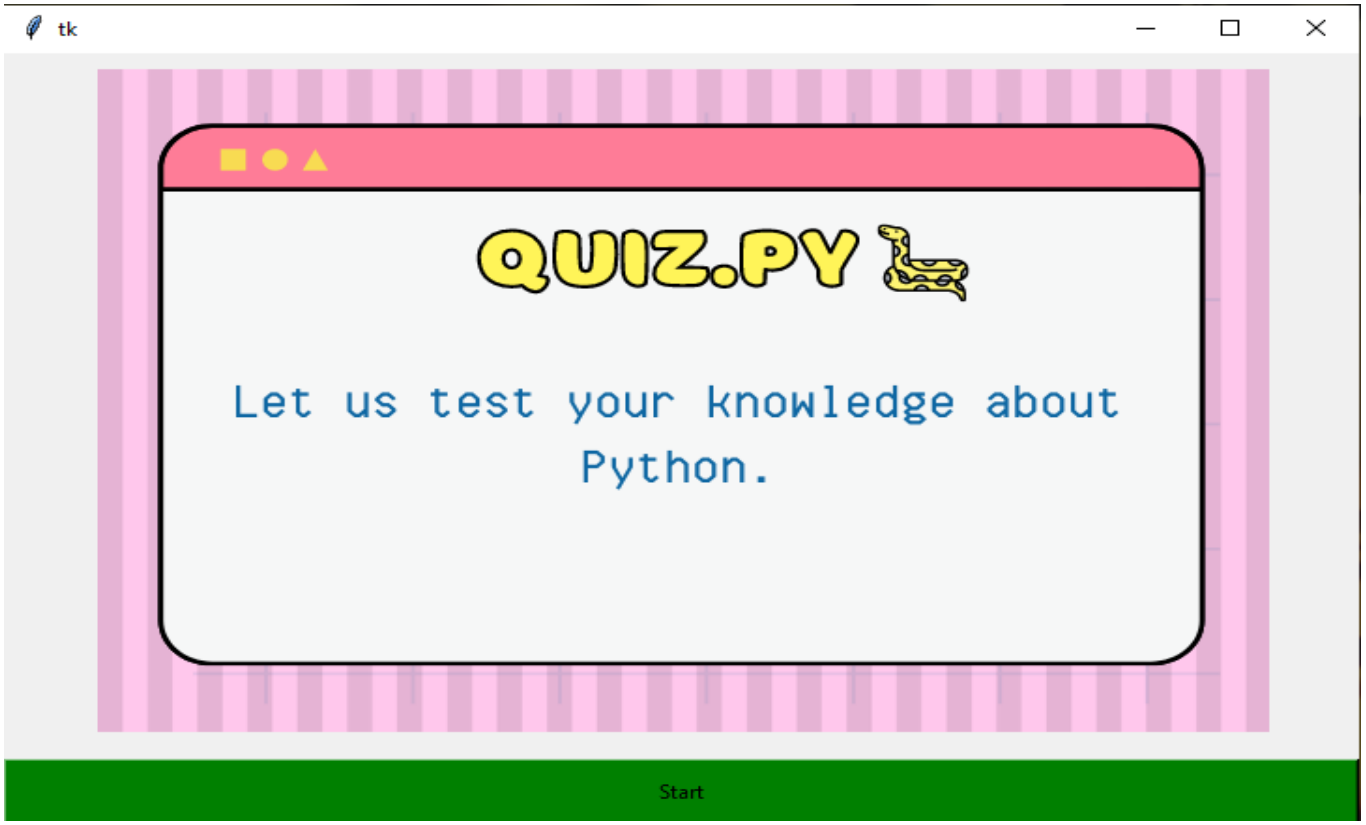
```
    y = countDown()
    if y == -1:
        display()
    m.mainloop()
def difficult():


    global h
    h = Tk()

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

    hard_frame = Frame(hard_canvas,bg="white")
    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",
```

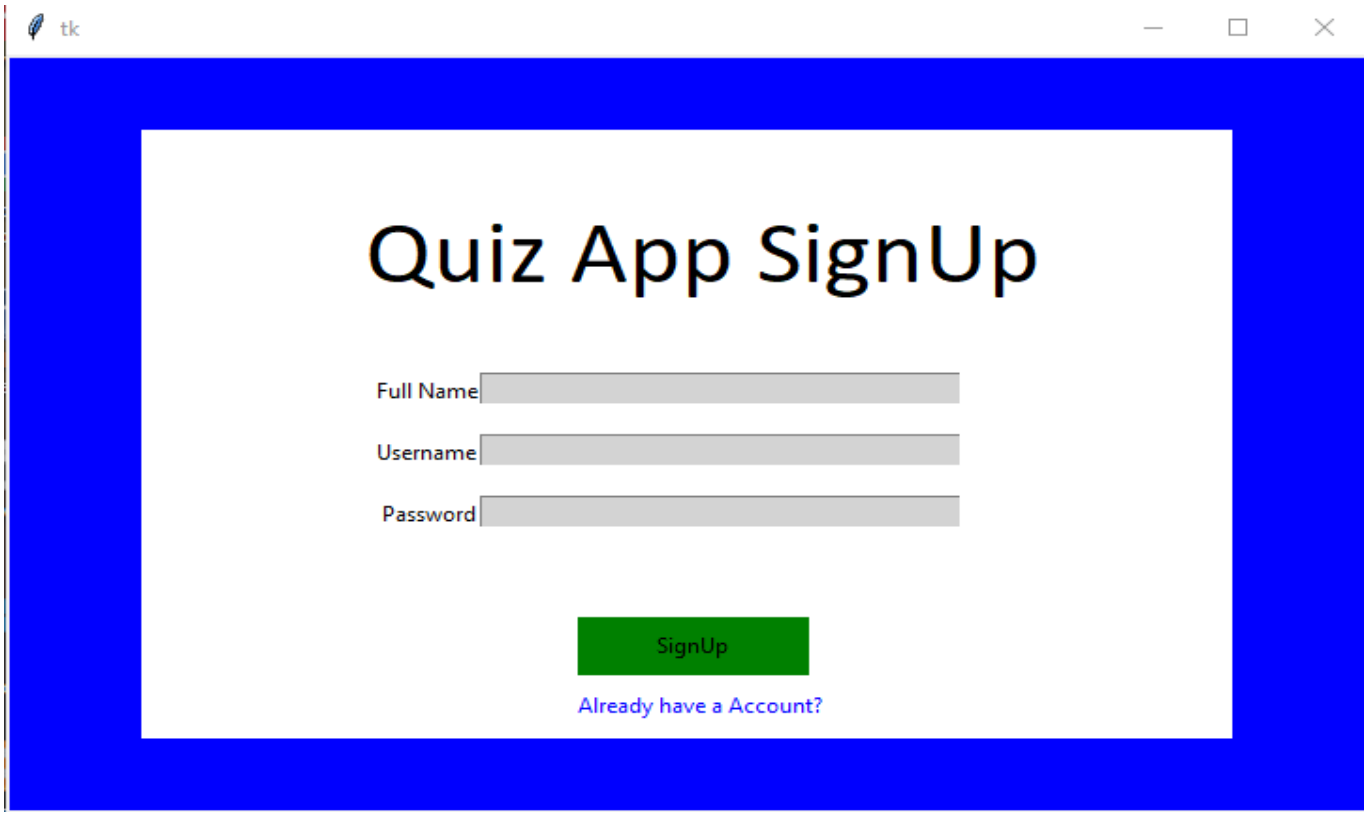
```
        "4",
        "4.0",
        "3.0"
    ],
    [
        "What will be the output of print(math.factorial(4.5))?",
        "24",
        "120",
        "error",
        "24.0"
    ]

]
    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="white")
    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="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="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="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="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])
```

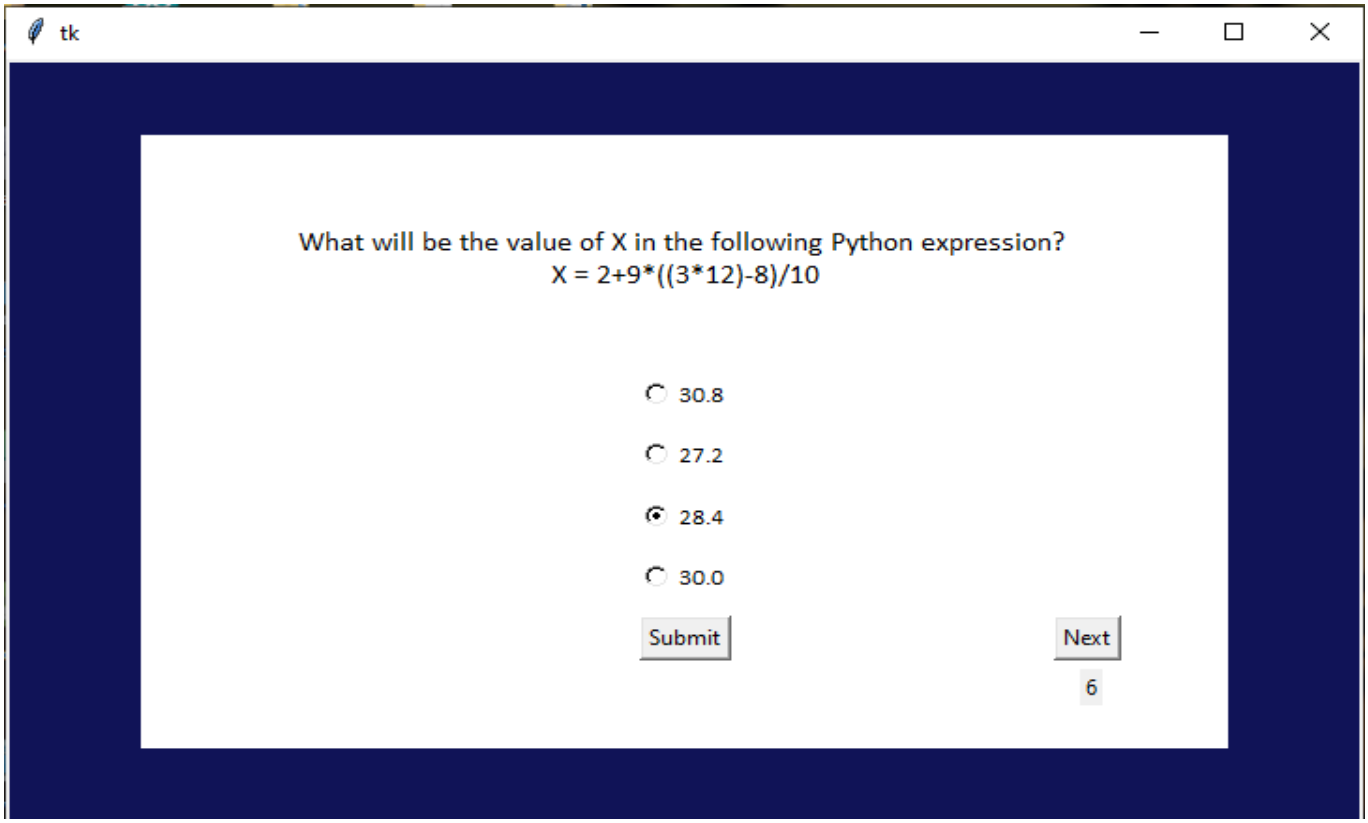
```
            b.configure(text=hardQ[x][2],value=hardQ[x][2])

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

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

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


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

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

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

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

    def showMark(mark):
        global sh
        sh = Tk()

        show_canvas = Canvas(sh,width=720,height=440,bg="#101357")
        show_canvas.pack()

        show_frame = Frame(show_canvas,bg="white")
        show_frame.place(relwidth=0.8,relheight=0.8,relx=0.1,rely=0.1)

        st = "Your score is "+str(mark)
        mlabel = Label(show_canvas,text=st,fg="black", font=('calibri', 40))
        mlabel.place(relx=0.5,rely=0.2,anchor=CENTER)

        sh.mainloop()
    def start():
        global root
        root = Tk()
        canvas = Canvas(root,width = 720,height = 440)
        canvas.grid(column = 0 , row = 1)
        img = PhotoImage(file="back.png")
        canvas.create_image(50,10,image=img,anchor=NW)

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

        root.mainloop()
```

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

5. Snapshots:


tk
QUIZ.PY
Let us test your knowledge about Python.
Start


tk
Quiz App SignUp
Full Name
Username
Password
SignUp
Already have a Account?



4. Conclusion and Future Scope:

The conclusion section should summarize the project and its achievements. It should also discuss any challenges encountered during the development process and how they were addressed. Additionally, this section can include future scope for the project, such as potential enhancements or new features that could be added in future versions.

In conclusion, we have successfully developed a quiz application using the Python programming language and the Tkinter GUI library. The application allows users to log in with a password and attempt quizzes in a variety of subject areas. Each question has a time limit of 10 seconds, and the user's score is displayed at the end of the quiz.