**THE QUIZ GAME**

*A*

*Mini Project Report*

*Submitted in partial fulfillment of the*

*Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

**KONDURU PRANEETH 1602-20-737-092**

**PENDYALA KRISHNA SAI 1602-20-737-077**

**SHAIK SOHAIL AKBAR 1602-20-737-106**



**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2022**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Information Technology**



**DECLARATION BY THE CANDIDATE**

We, **KONDURU PRANEETH, PENDYALA KRISHNA SAI and SHAIK SOHAIL AKBAR** bearing hall ticket numbers, **1602-20-737-092, 1602-20-737-077 and 1602-20-737-106** hereby declare that the project report entitled "**THE QUIZ GAME“** is submitted in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**KONDURU PRANEETH**

**1602-20-737-092**

**PENDYALA KRISHNA SAI 1602-20-737-077**

**SHAIK SOHAIL AKBAR 1602-20-737-106**

(Faculty In-Charge) (Head, Dept of IT)

**ACKNOWLEDGEMENT**

The enduring pages of the work are the cumulative sequence of extensive guidance and arduous work. We wish to acknowledge and express our personal gratitude to all those without whom this work could not have been reality.

We feel very delighted to get this rare opportunity to show our profound senses of reverences and indebtedness to our esteemed teacher **DRL PRASANNA** ma’am and **RADHA GADIGE** ma’am, for their keen and sustained interest, valuable advice, throughout the course of which led our mini project, to a successful completion. For this kind act of consideration, we are beholder to them in special manner and no one can fully convey our feelings of respect and regard for them.

**ABSTRACT**

The Quiz Application(Game) In Python is written in python programming language, This Simple Quiz On Python Programming is a simple project for testing one’s knowledge power in the given topic examination. The project Quiz In Python contains only the user side. The user must log in or sign up first to start the Quiz On Python. Also, there is a time duration set for solving the questions. The user should solve the questions before time is up.

A Quiz Project In Python just contains the user section. The user can log-in to give the exam. After giving the exam, the user can see their score and evaluate themselves. The design of this project Quiz Program In Python Using Tkinter is pretty simple so that the user won’t find any difficulties while working on it.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| S.NO | TOPIC | Page No. |
| 1. | Abstract & Introduction | 6 |
| 2. | Technology | 8 |
| 3. | Design | 9 |
| 4. | Implementation | 14 |
| 5. | Github link and Testing | 45 |
| 6. | Results | 47 |
| 7. | Additional Knowledge Acquired | 60 |
| 8. | Conclusion and Future work | 61 |
| 9. | References | 62 |

**INTRODUCTION & ABSTRACT**

**ABSTRACT**

The Quiz Application(Game) In Python is written in python programming language, This Simple Quiz On Python Programming is a simple project for testing one’s knowledge power in the given topic examination. The project Quiz In Python contains only the user side. The user must log in or sign up first to start the Quiz On Python. Also, there is a time duration set for solving the questions. The user should solve the questions before time is up.

A Quiz Project In Python just contains the user section. The user can log in to give the exam. After giving the exam, the user can see their score and evaluate themselves. The design of this project Quiz Program In Python Using Tkinter is pretty simple so that the user won’t find any difficulties while working on it.

**INTRODUCTION**

Python provides a standard GUI framework Tkinter which is used to develop fast and easy GUI applications. Here we will be developing a simple multiple-choice quiz in python with GUI. We will be creating a multiple choice quiz in Python with Tkinter.

The application that we have build aims for the implementation of a Quiz game using tkinter a python based GUI.It is a basic quiz game where we have to signup for the first time the signup will use a small database to store the information of the user and log in to game if user gave invalid login then a error massage is appeared. And the game contain 3 levels (easy,medium,hard) the user have to select one level and then the questions will be popped up and user have to answer the questions in the given time the question only take a single option these are build using radio buttons in tkinter and at last the score will be appeared on the screen.

A Quiz Project In Python just contains the user section. The user can log-in to give the exam. After giving the exam, the user can see their score and evaluate themselves. The design of this project Quiz Program In Python Using Tkinter is pretty simple so that the user won’t find any difficulties while working on it.

In this quiz game we have 3 types of levels based on the hardness easy,medium and hard. The end-user has to select one of these 3 levels to start playing the game. After selecting the level a random 5 question appear on screen one-by-one. There's a timer of 10 seconds for each question. If no option is chosen at the end of 10 seconds, the next question is displayed and to the previous question 0 marks are awarded.

**Features of project :**

* **Login**
* **Register**
* **Different quiz level**
* **Timer**
* **Randomize questions.**
* **Scorecard**
* **Replay and exit**

**TECHNOLOGY**

**Hardware Requirements**

* 512 MB RAM
* 2GB HDD
* CORE i5

**Software Requirements**

* Windows XP/ Windows 2000
* PYTHON INTERPRETER

**Packages used**

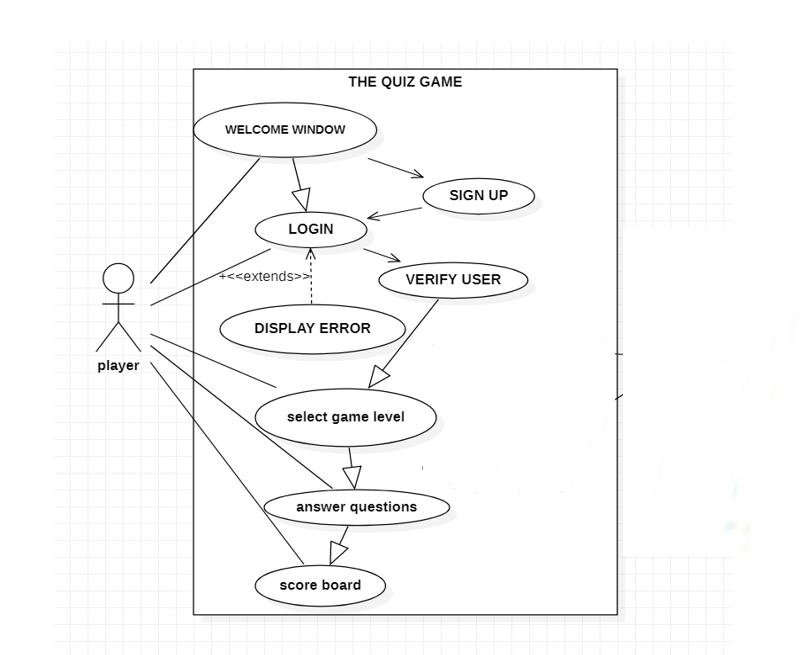
* Tkinter

**Modules imported from package :**

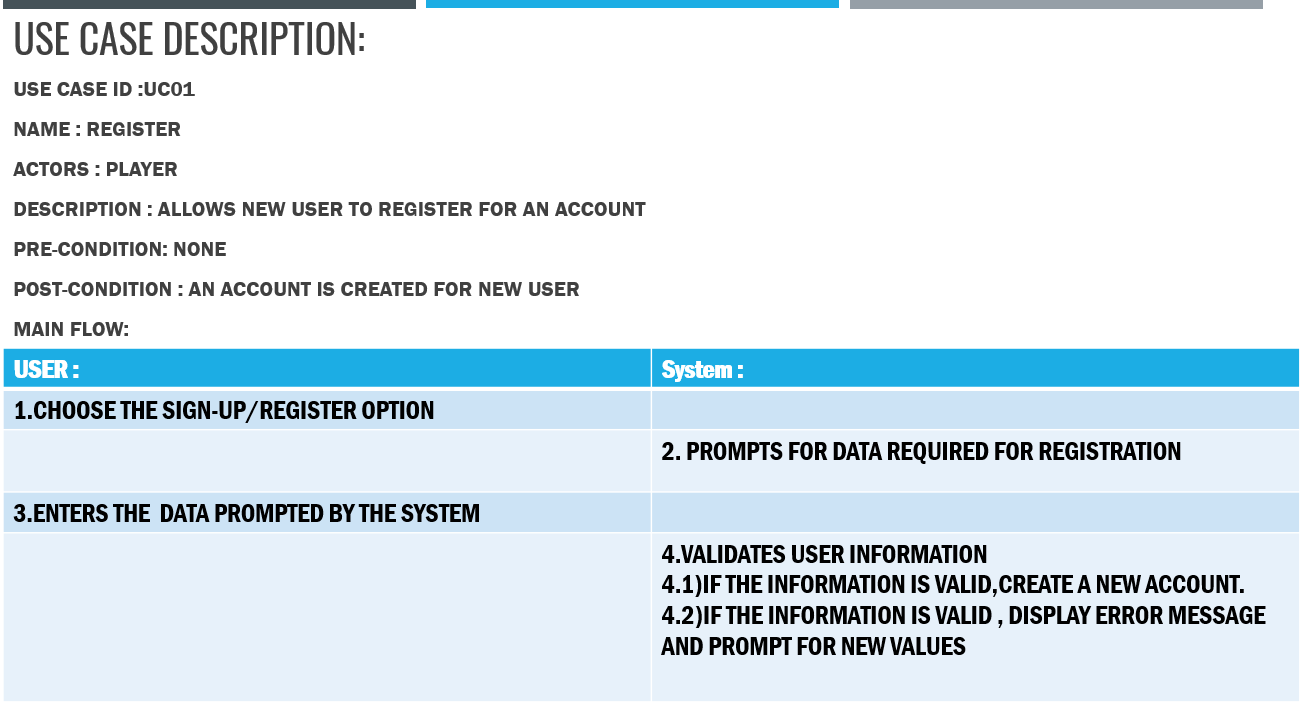
* random
* sqlite3
* time

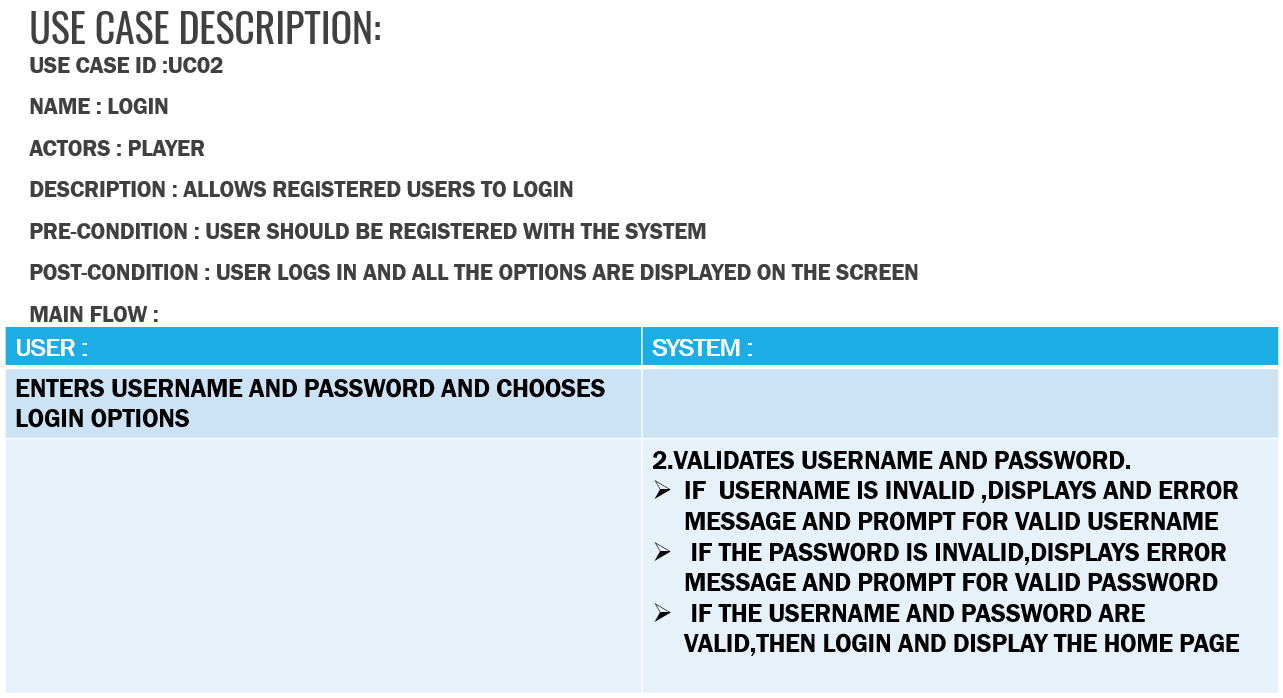
**PROPOSED WORK**

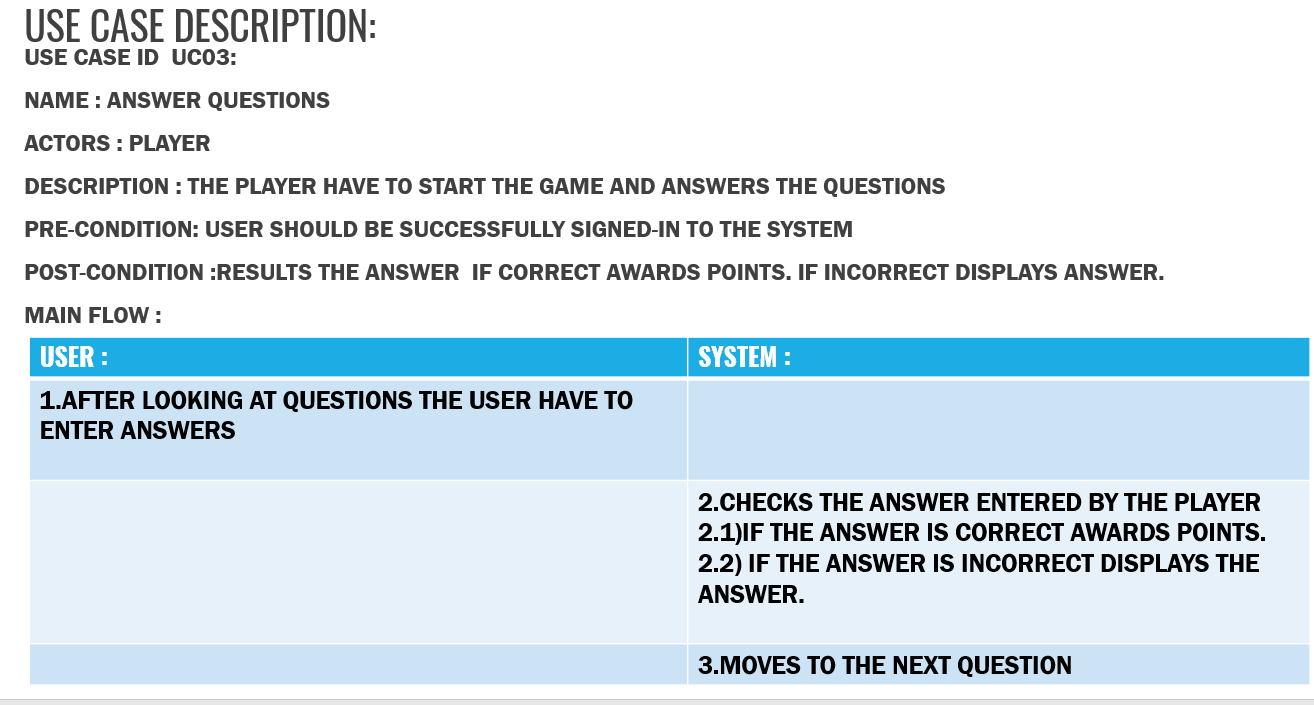
USER CASE DIAGRAM:

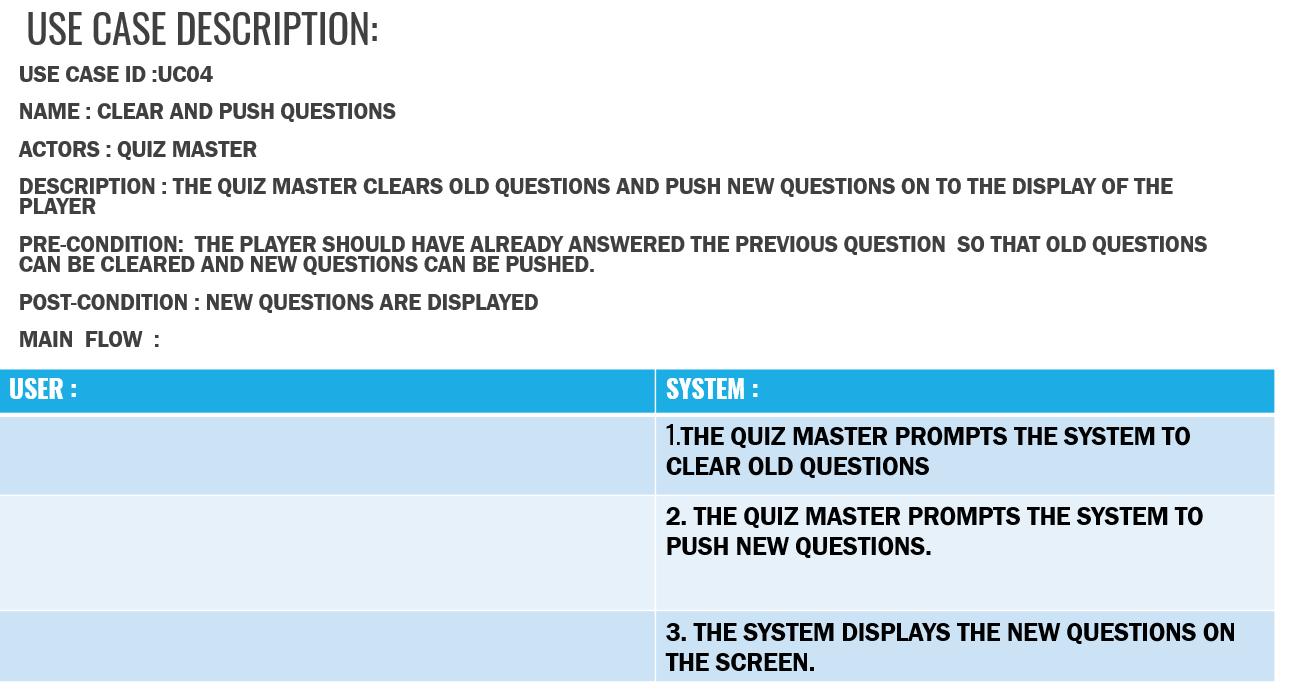


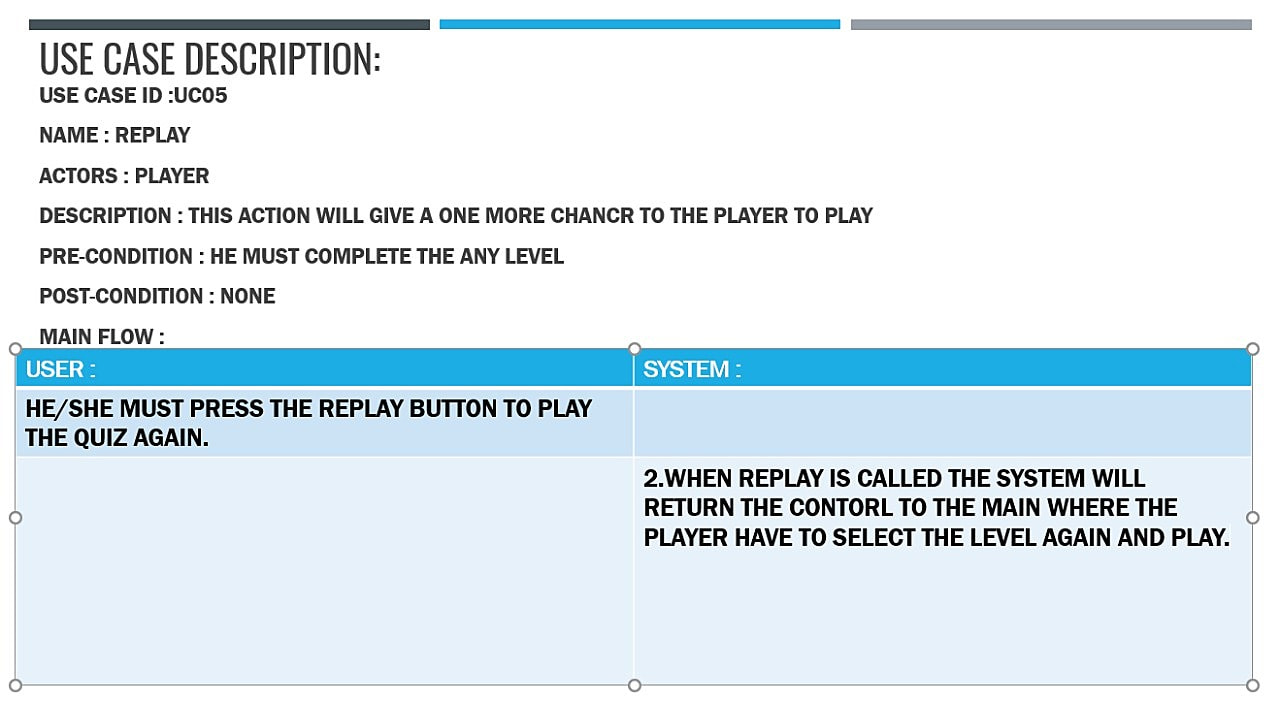
**USER CASE DESCRIPTION:**

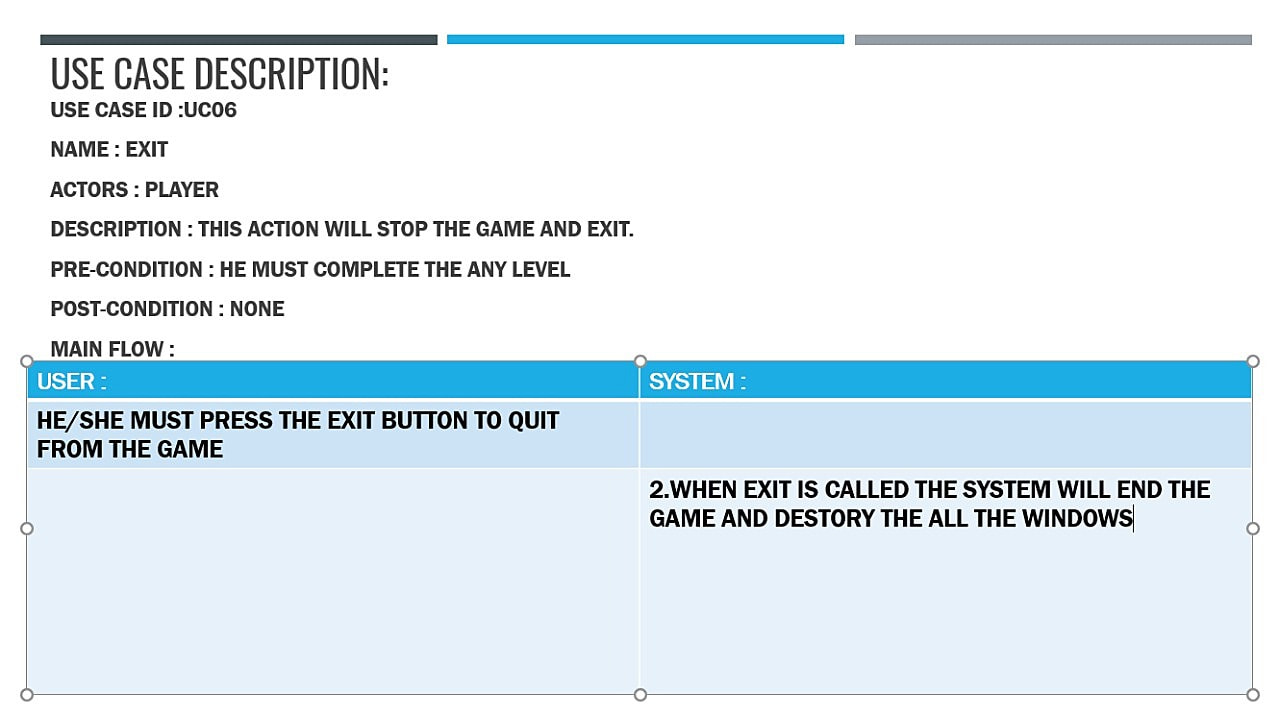




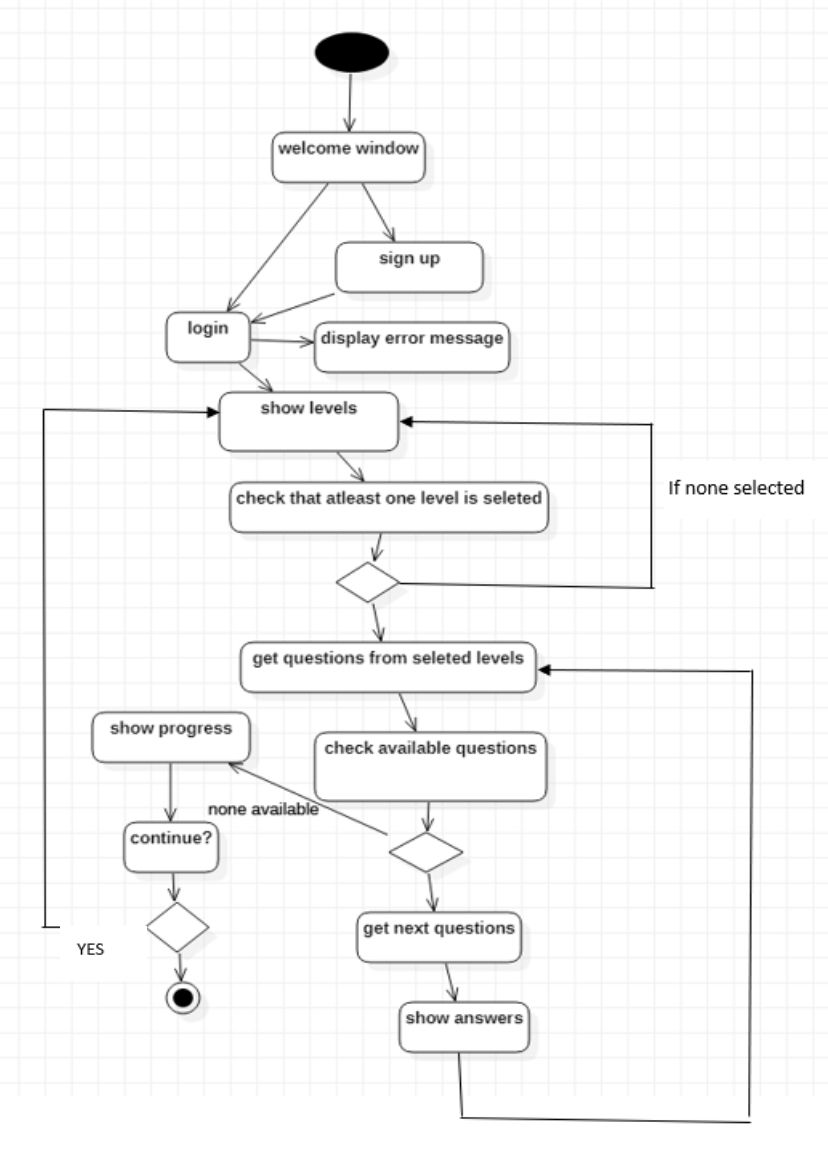








**ACTIVITY DIAGRAM:**



**IMPLEMENTATION**

**MODULE FOR LOG-IN PAGE :**

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,d 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()

**MODULE FOR SIGN-IN PAGE**

def signUpPage():

root.destroy()

global sup

sup = Tk()

fname = StringVar()

uname = StringVar()

passW = StringVar()

country = 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)

#country

clabel = Label(sup\_frame,text="Country",fg='black',bg='white')

clabel.place(relx=0.215,rely=0.7)

c = Entry(sup\_frame,bg='#d3d3d3',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()

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&#91;0]&#91;0]+"\nPassword is "+z&#91;-1]&#91;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='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()

**MODULE FOR MAIN MENU**

def menu1(i):

if(i==0):

login.destroy()

else:

sh.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 G A M E ',fg="white",bg="#101357")

wel.config(font=('Broadway 22'))

wel.place(relx=0.15,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:

#var.deselect()

menu.destroy()

easy()

elif x == 2:

menu.destroy()

medium()

#var.deselect()

elif x == 3:

menu.destroy()

difficult()

#var.deselect()

'''else:

if j==2 :

menu.destroy()

medium()

elif j==3:

menu.destroy()

difficult()

else :

menu.destroy()

easy()'''

letsgo = Button(menu\_frame,text="Let's Go",bg="white",font="calibri 12",command=navigate)

letsgo.configure(bg = "green")

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()

def exit():

menu1(1)

pass

def exit2():

sh.destroy()

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="green")

mlabel.config(font=('Helvetica 25'))

mlabel.place(relx=0.5,rely=0.5,anchor=CENTER)

exit1 = Button(show\_frame,command=exit,text='REPLAY')

exit1.configure(width = 8,height=2, activebackground = "#33B5E5", bg ='skyblue', relief = RAISED)

exit1.place(relx=0.8,rely=0.82,anchor=CENTER)

exit = Button(show\_frame,command=exit2,text='EXIT')

exit.configure(width = 5,height=2, activebackground = "#33B5E5", bg ='red', relief = RAISED)

exit.place(relx=0.05,rely=0.82,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="quiz.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()

**COMPLETE 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 Game 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,d in logdata:

if b == uname.get() and c == pas.get():

menu1(0)

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", bg ="green" ,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()

country = 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 Game 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)

#country

clabel = Label(sup\_frame,text="Country",fg='black',bg='white')

clabel.place(relx=0.215,rely=0.7)

c = Entry(sup\_frame,bg='#d3d3d3',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()

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='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 menu1(i):

if(i==0):

login.destroy()

else:

sh.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 G A M E ',fg="white",bg="#101357")

wel.config(font=('Broadway 22'))

wel.place(relx=0.15,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:

#var.deselect()

menu.destroy()

easy()

elif x == 2:

menu.destroy()

medium()

#var.deselect()

elif x == 3:

menu.destroy()

difficult()

#var.deselect()

'''else:

if j==2 :

menu.destroy()

medium()

elif j==3:

menu.destroy()

difficult()

else :

menu.destroy()

easy()'''

letsgo = Button(menu\_frame,text="Let's Go",bg="white",font="calibri 12",command=navigate)

letsgo.configure(bg = "green")

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()

def exit():

menu1(1)

pass

def exit2():

sh.destroy()

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="green")

mlabel.config(font=('Helvetica 25'))

mlabel.place(relx=0.5,rely=0.5,anchor=CENTER)

exit1 = Button(show\_frame,command=exit,text='REPLAY')

exit1.configure(width = 8,height=2, activebackground = "#33B5E5", bg ='skyblue', relief = RAISED)

exit1.place(relx=0.8,rely=0.82,anchor=CENTER)

exit = Button(show\_frame,command=exit2,text='EXIT')

exit.configure(width = 5,height=2, activebackground = "#33B5E5", bg ='red', relief = RAISED)

exit.place(relx=0.05,rely=0.82,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="quiz.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()

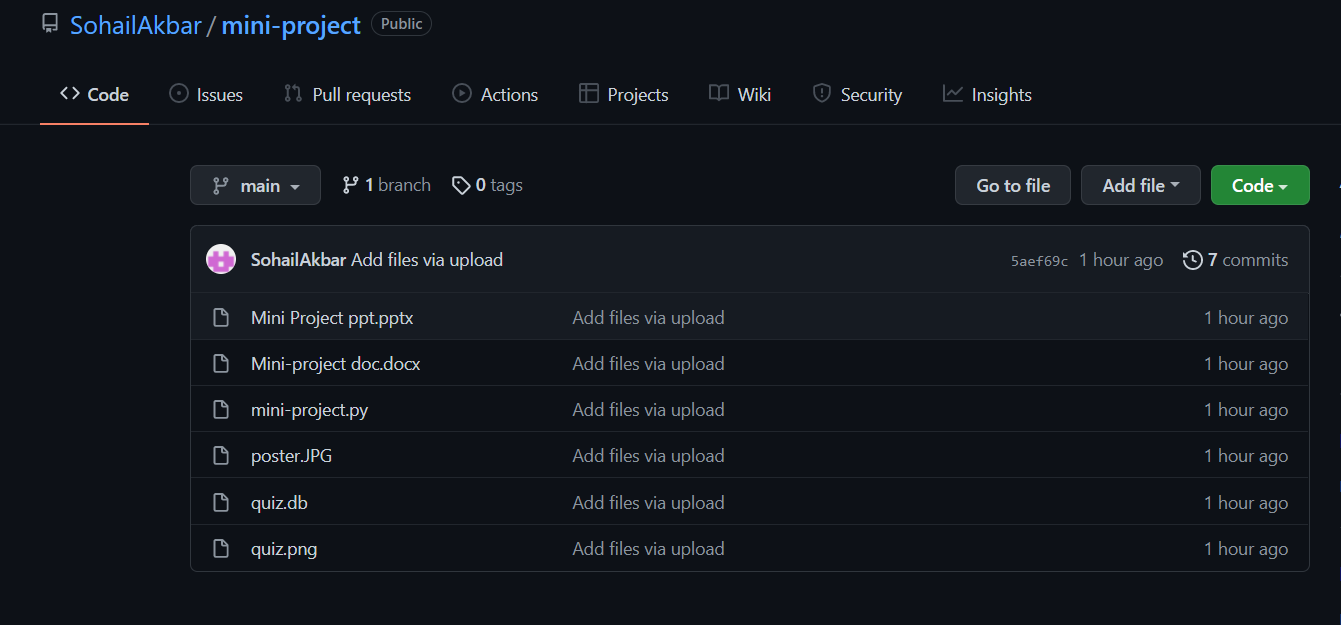
**ADDITIONAL FILES**

****

**Git Hub Link**

<https://github.com/SohailAkbar/mini-project.git>

**Folder Structure**

****

**Git Lab Link**

https://202.65.154.100/team12-b-web/the-quiz-game

**Test cases**

Demo sign-up details used (if we use our database in github):

1. Username :Krishna\_04

password : vasavi

2. Username : Sohail\_024

Password : vasavi

In Log-in pages

* If wrong log-in user-id or password are given it displays a message to give the correct user details.

In Menu page

* Three levels are displayed easy, medium and hard. User have to select one of three to play the game.

**For easy level Questions**

**Answers**

"[1, 2, ‘hello’]",

"34.000000",

"27.2",

"Class",

"^"

**For Medium level questions**

**Answers**

"accept",

"1",

"25",

"33",

"none of the mentioned"

**For Hard level questions**

**Answers**

"None of the mentioned",

"in",

"(1,2,3)",

"4",

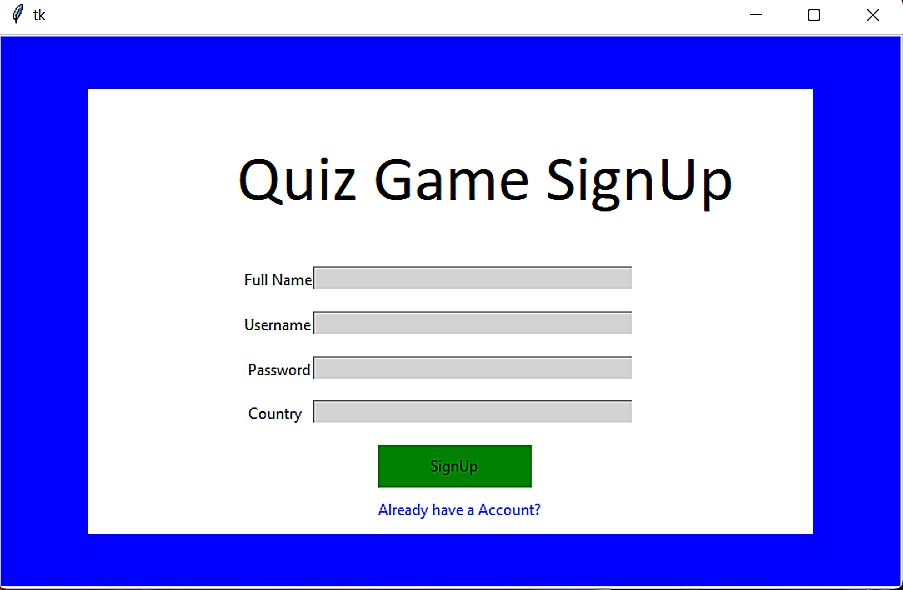
"error"

**SCREENSHOTS OF APPLICATION**

**WELCOME PAGE**



**SIGN-UP PAGE**



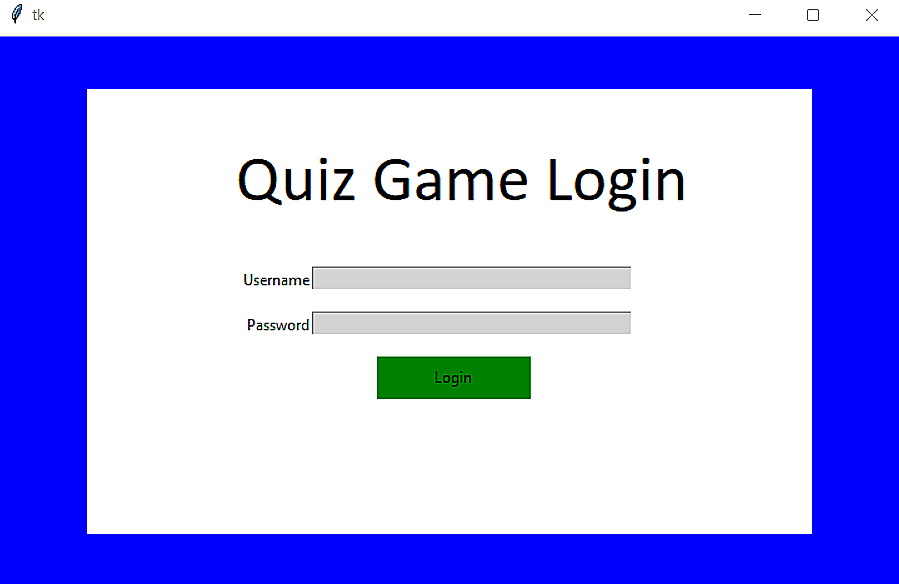
**DISCRIPTION**

In the Sign-up page we have to create an account with a unique username and password for end-users.



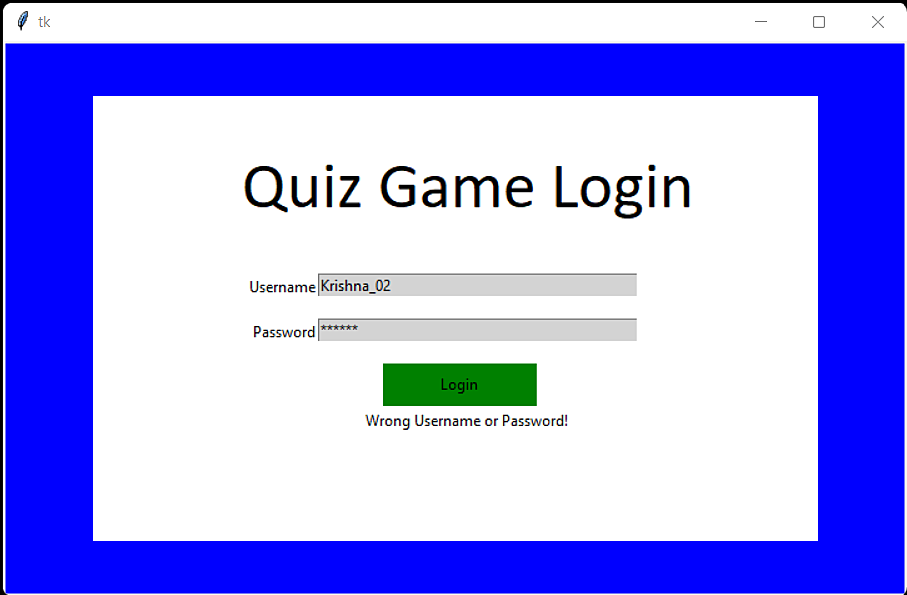
>>If we already have and account we can click the given already have and account button and enter into log-in page.

**LOG-IN PAGE**



**DISCRIPTION**

>>If wrong log-in details or password are entered.



>>If correct log-in and passwords are given main menu is opened.

**MAIN MENU**

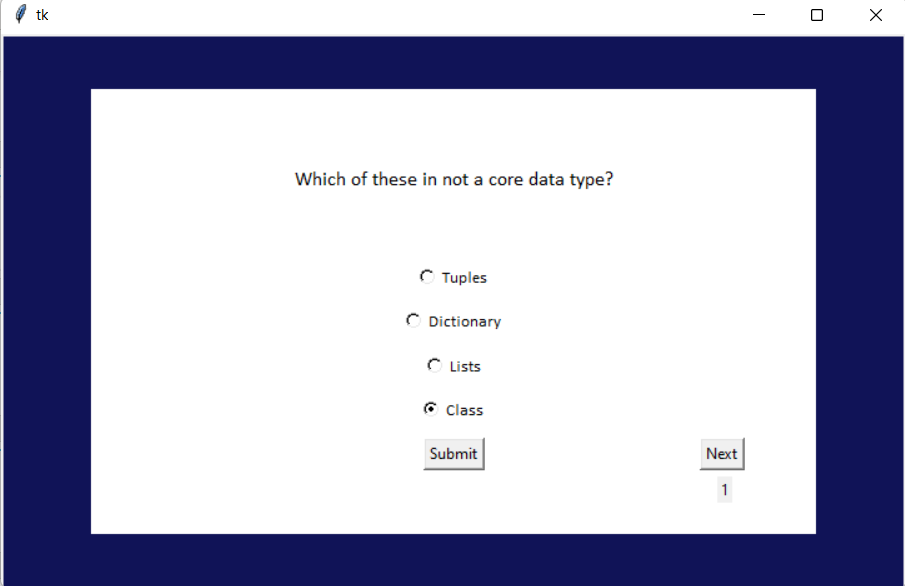
>>In main menu 3 levels are displayed easy,medium and hard.

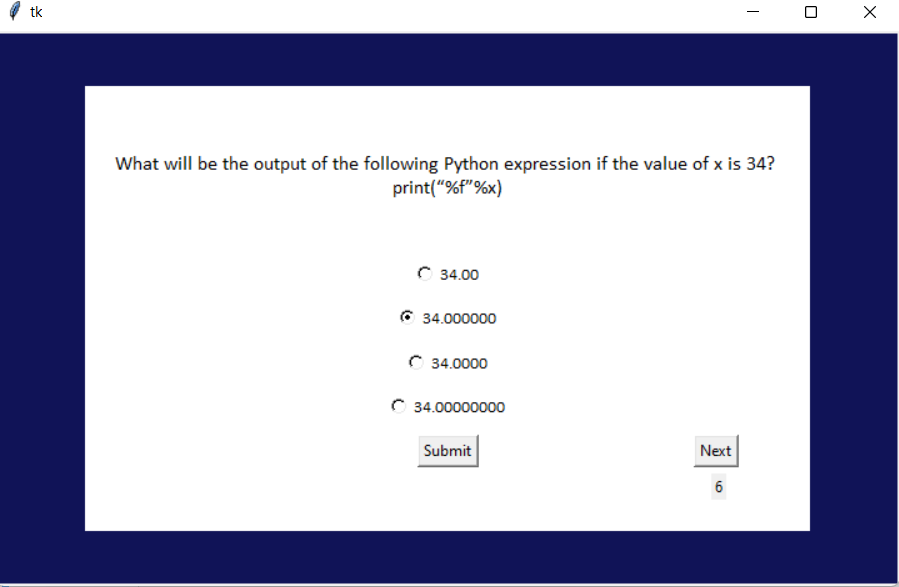
The end-user has to choose one of the three to start playing the quiz game.

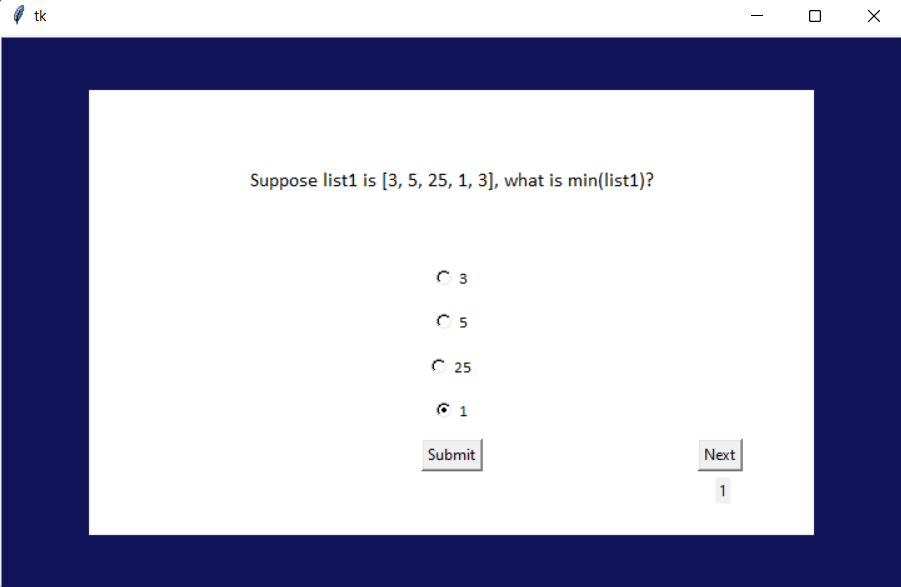


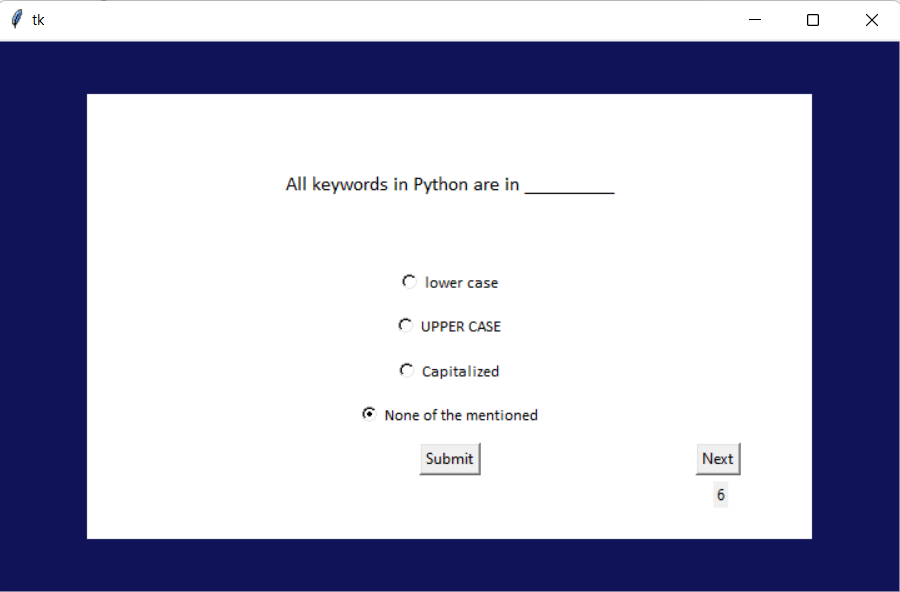
**EASY LEVEL**

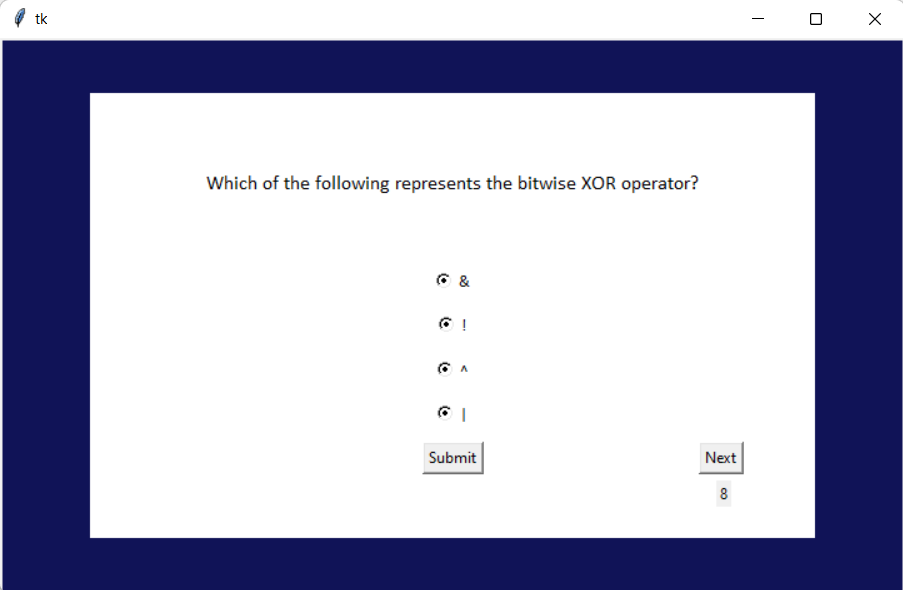
>>Questions appear randomly from the question bank.



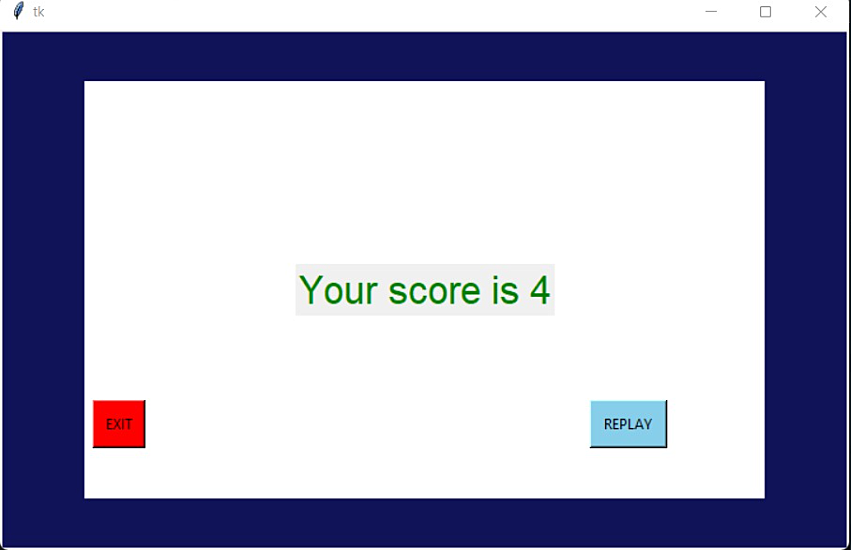






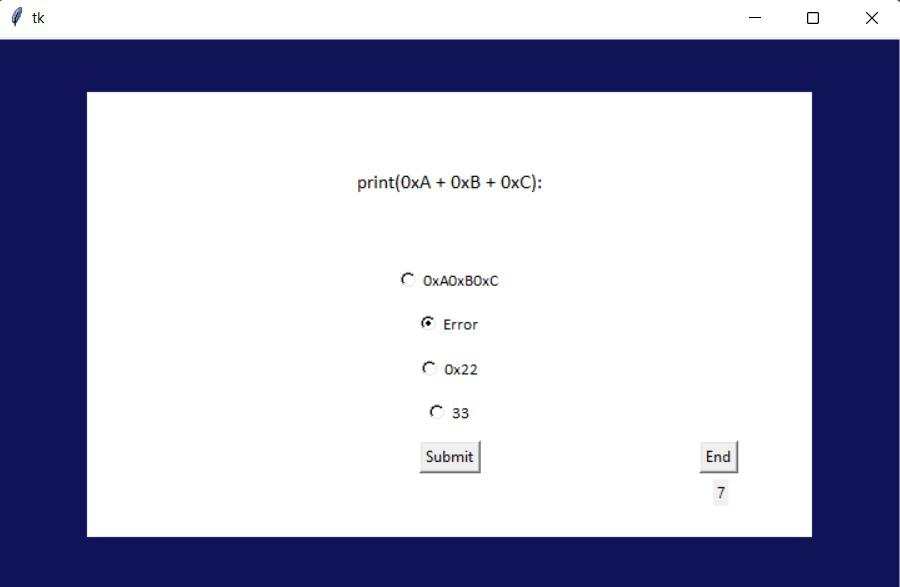
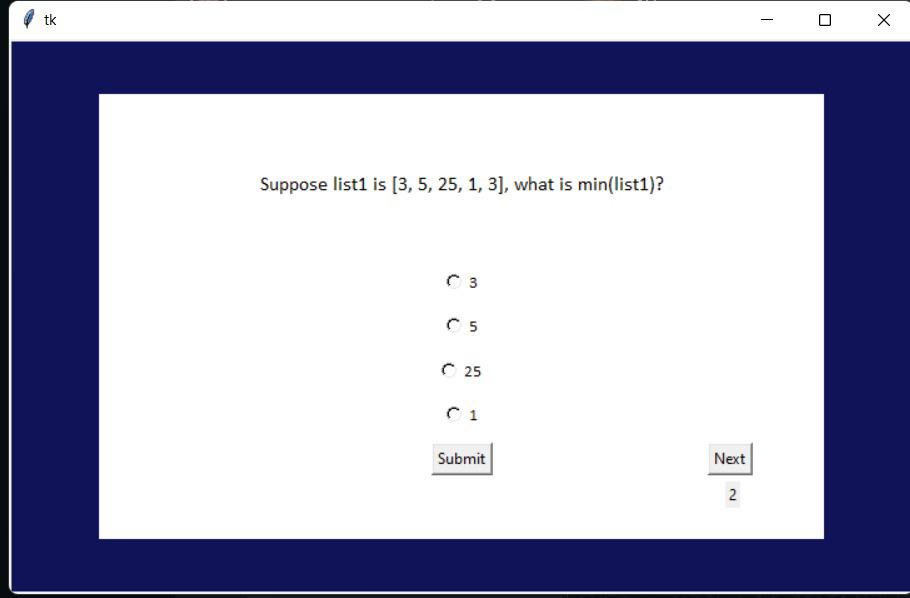
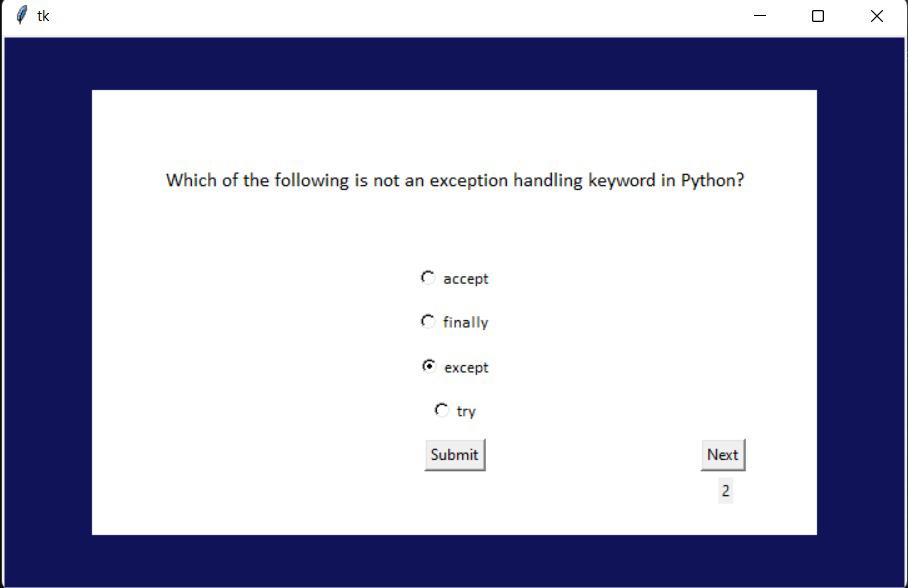
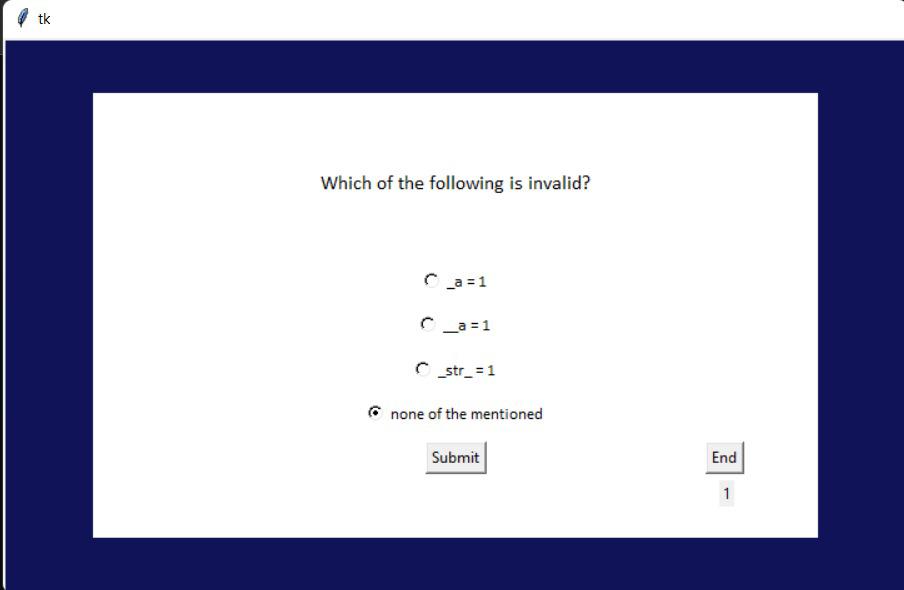
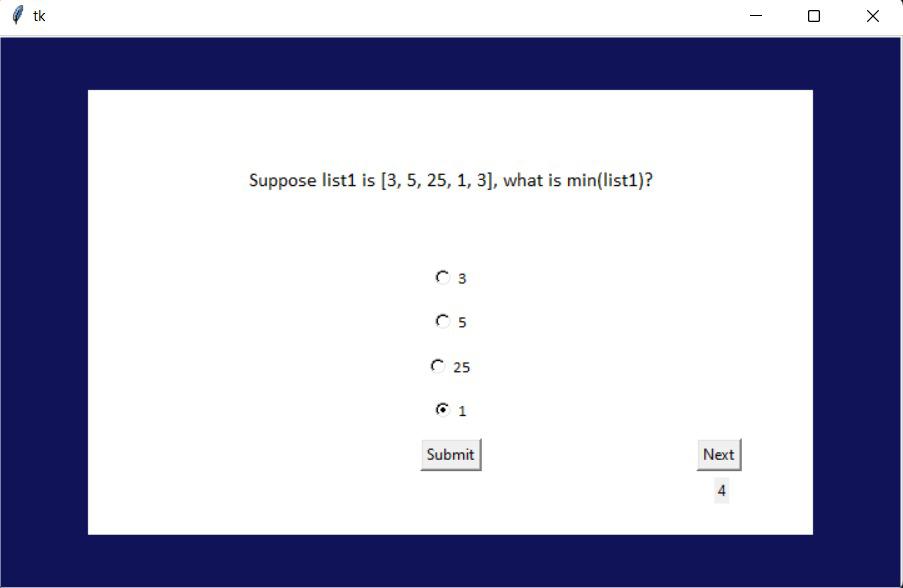


**SCORECARD for easy level**

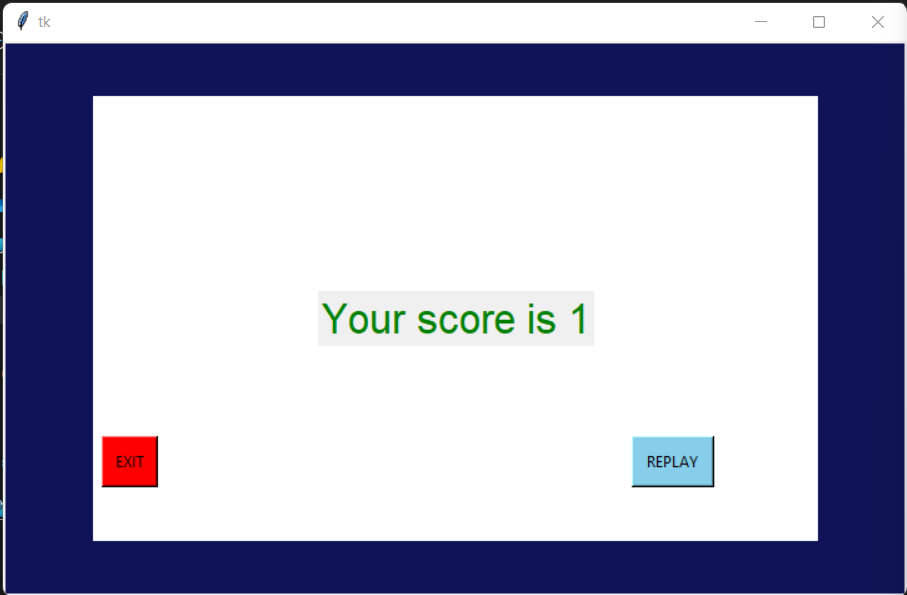


**MEDIUM LEVEL**

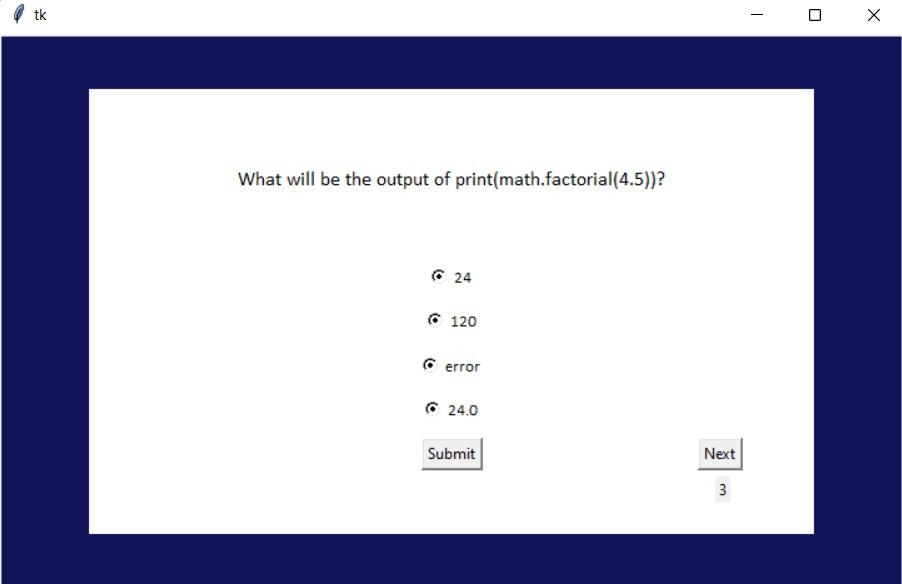
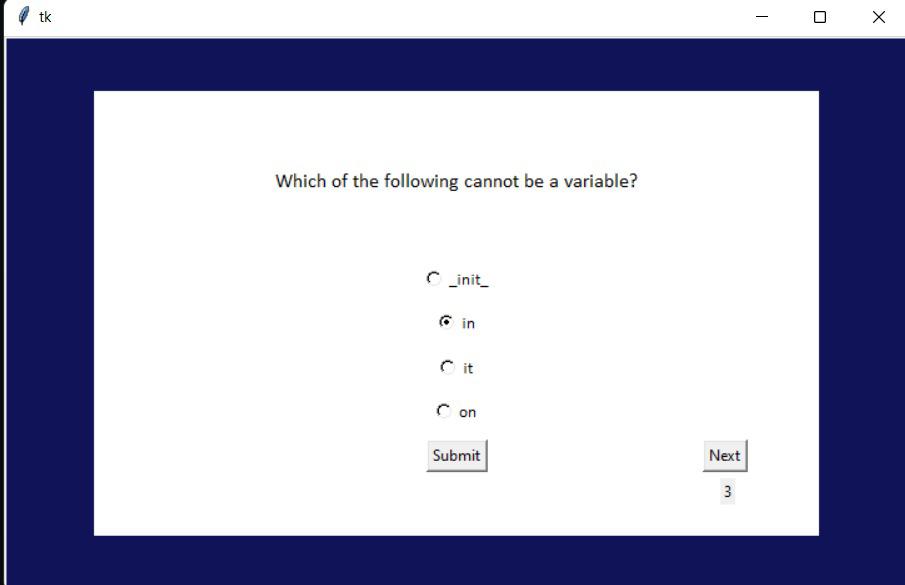
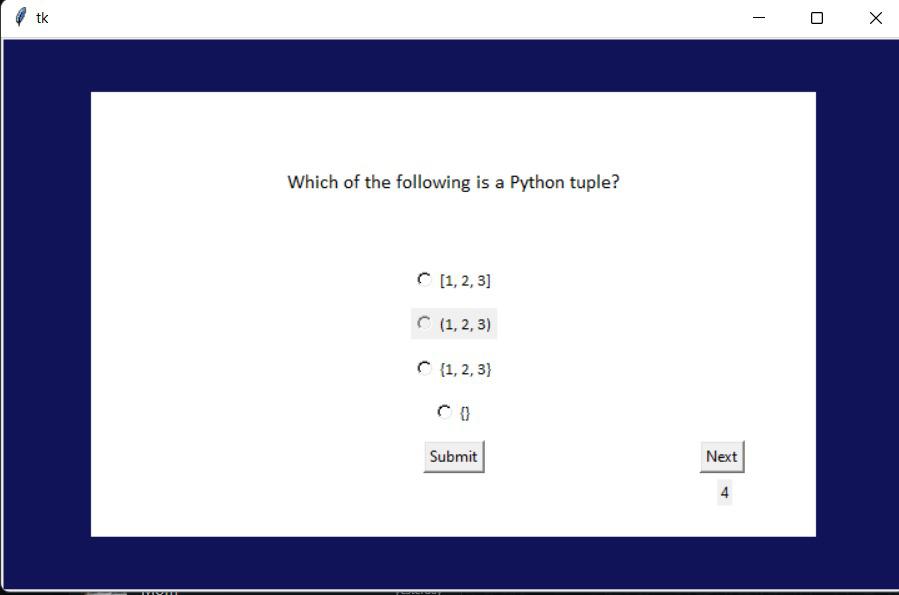
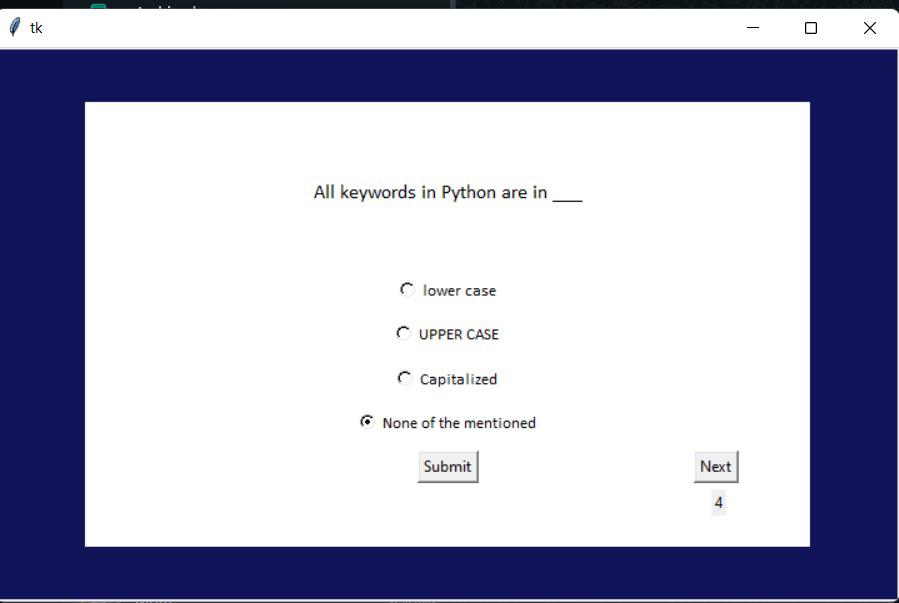
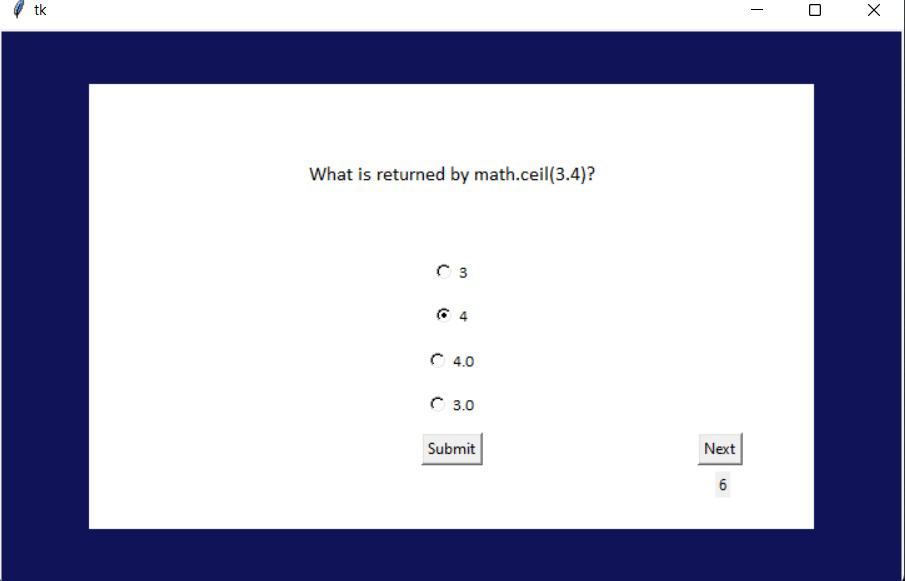
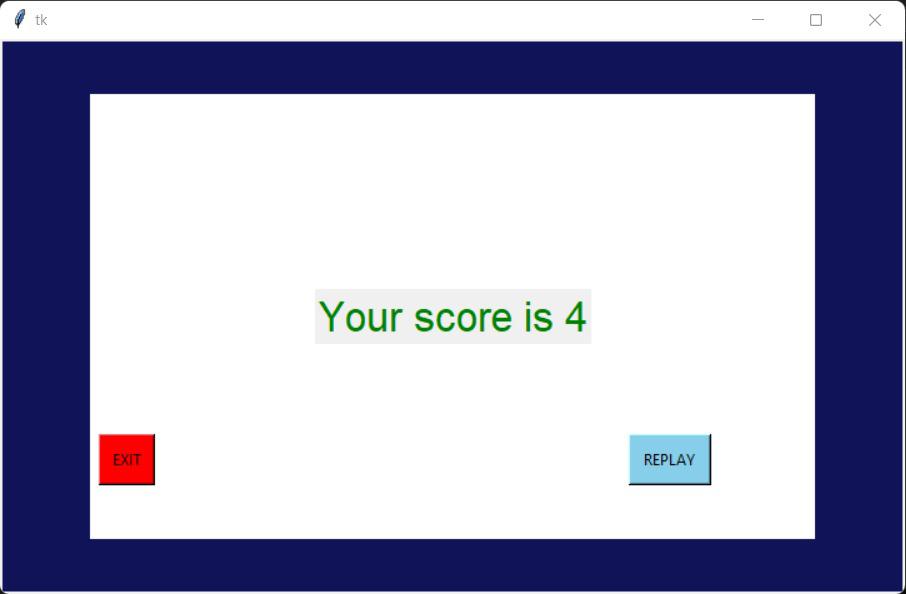
>>Questions appear randomly from question bank.



**SCORECARD for medium level**



**HARD LEVEL**

>>Questions appear randomly from question bank. **SCORECARD for hard** 

**What is the additional knowledge gained as a result of implementing this mini-project apart from the syllabus covered in the course python programming?**

We were exposed to Graphical user interface(GUI) tkinter that helped us make our project beautiful and add visual effects to it.We had to explore the whole tkinter application and had to build our application form the scratch and we exposed to database management system.

**CONCLUSION**

* The scope of this project is enormous in terms of gaining knowledge and sharing knowledge among the world.
* In future we can conduct a mega quiz with multiple participants and also we can have entry quiz and we can present gifts to winners. And we can improve more questions and toughness of the game.
* Also, we can provide e books for different categories and which are also useful for different competitive exams
* In future all exams are going to be online so this game will create a virtual experience for the students.

**FUTURE WORK**

* The scope of this project is wide in terms of gaining knowledge and sharing knowledge among the world.
* In future we can conduct a mega quiz with multiple participants and also we can have entry quiz and we can present gifts to winners. And we can improve more questions and toughness of the game.
* Also, we can provide e books for different categories and which are also useful for different competitive exams
* In future all exams are going to be online so this game will create a virtual experience for the students.

**REFERENCES**

**Basic Python**

1. Course covered during 2nd semester by Dr. Ramesh Vassapanavara Sir.
2. Ppts and handouts provided by the sir.
3. Python Programming - Using Problem Solving Approach First Edition by Reema Thareja.

**Tkinter tutorials**

1.<https://www.youtube.com/watch?v=YXPyB4XeYLA>

2. <https://www.youtube.com/watch?v=VMP1oQOxfM0&t=271s>

3. <https://www.youtube.com/watch?v=NkAwxoQkdOA>

**Use cases and activity diagram**

1. Hand-outs provided by DRL PRASANNA ma’am.
2. <https://www.youtube.com/watch?v=zid-MVo7M-E>
3. <https://www.youtube.com/watch?v=knM8BGY9yVI&t=161s>