**05 Aug 2024\_Monday**

**Q. Udemy, an online website to learn coding recently gave a free online course on Basics of Python which included the fundamentals of python, Data Structures, creating basic GUI. At last there was a quiz conducted by Udemy to all and everyone secured good marks and also got the certification. Matthew, who was a part of this course as well, thought if he could create one GUI same as Udemy using Python itself for a quiz but Matthew is not aware about few concepts like how to show a message box and stuff in GUI.**

**So to help matthew write a code in python that will show the questions, should be able to select any one answer using radio button, and a button to go to the next question and a button to quit from the quiz and a message pop up to show the result once the quiz is done**

**#import required modules**

**#read front he json file and show each question and options on the GUI using labels and ask the user to choose one option using radio button options check if its right and update the score variable parallely**

**#finally show the score of the user when all the questions are done answering**

import tkinter as tk

from tkinter import messagebox

import json

# Load quiz questions from JSON file

with open('quiz\_questions.json', 'r') as file:

quiz\_data = json.load(file)

class QuizApp:

def \_\_init\_\_(self, root):

self.root = root

self.root.title("Quiz Application")

self.score = 0

self.current\_question = 0

self.question\_label = tk.Label(root, text="", wraplength=400, justify="left")

self.question\_label.pack(pady=20)

self.options\_var = tk.StringVar()

self.option\_buttons = []

for \_ in range(4):

rb = tk.Radiobutton(root, text="", variable=self.options\_var, value="", wraplength=400, justify="left")

rb.pack(anchor='w', padx=20, pady=5)

self.option\_buttons.append(rb)

self.next\_button = tk.Button(root, text="Next", command=self.next\_question)

self.next\_button.pack(pady=20)

self.quit\_button = tk.Button(root, text="Quit", command=self.quit\_quiz)

self.quit\_button.pack(pady=10)

self.show\_question()

def show\_question(self):

question\_data = quiz\_data[self.current\_question]

self.question\_label.config(text=question\_data['question'])

self.options\_var.set(None)

for i, option in enumerate(question\_data['options']):

self.option\_buttons[i].config(text=option, value=option)

def next\_question(self):

if self.options\_var.get() == quiz\_data[self.current\_question]['answer']:

self.score += 1

self.current\_question += 1

if self.current\_question < len(quiz\_data):

self.show\_question()

else:

self.show\_result()

def show\_result(self):

messagebox.showinfo("Quiz Completed", f"Your score is: {self.score}/{len(quiz\_data)}")

self.root.quit()

def quit\_quiz(self):

if messagebox.askokcancel("Quit", "Are you sure you want to quit?"):

self.root.after(0, self.show\_thank\_you\_message)

def show\_thank\_you\_message(self):

messagebox.showinfo("Thank You", "Thank you for participating in the quiz!")

self.root.quit()

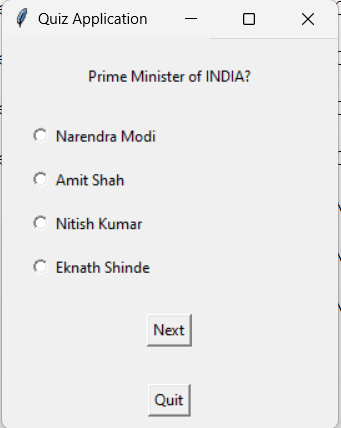
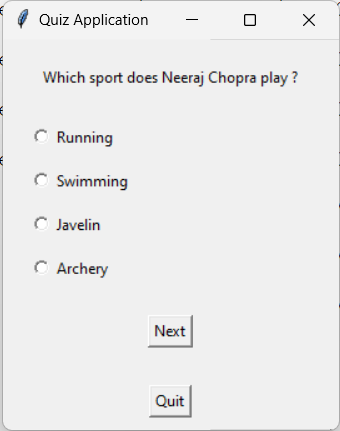
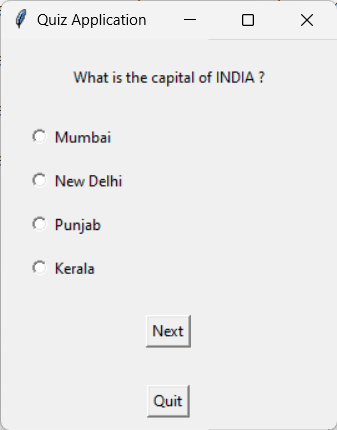
if \_\_name\_\_ == "\_\_main\_\_":

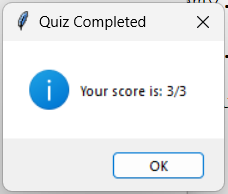
root = tk.Tk()

app = QuizApp(root)

root.mainloop()

**Output**

****

****