# KAUN BANEGA CROREPATI BY SANDEEP JAAT :-

import tkinter as tk

from tkinter import messagebox

# Login Credentials

USERNAME = "kbc24"

PASSWORD = "Jaat@123"

# Quiz Questions (Structured by Rounds)

rounds = {

    "Round 1": [

        {"question": "Q1. Who developed Python programming language?",

         "options": ["Wick van Rossum", "Rasmus Lerdorf", "Guido van Rossum", "Niene Stom"],

         "answer": "Guido van Rossum"},

        {"question": "Q2. Which keyword is used to define a function in Python?",

         "options": ["fun", "def", "function", "define"],

         "answer": "def"},

        {"question": "Q3. Which character is used to give single-line comments in Python?",

         "options": ["# (Pound)", "// (Slash)", "-- (Dash)", "/\* (Asterisk)"],

         "answer": "# (Pound)"},

        {"question": "Q4. Which function is used to take input directly from user in python?",

         "options": ["print()", "INPUT()", "format()", "input()"],

         "answer": "input()"},

        {"question": "Q5. Is python code?",

         "options": ["Compiled", "Interpreted", "Both a and b", "None of these"],

         "answer": "Both a and b"}

    ],

    "Round 2": [

        {"question": "Q1. Giddha is the folk dance of which state in India?",

         "options": ["Assam", "West Bengal", "Odisha", "Punjab"],

         "answer": "Punjab"},

        {"question": "Q2. Highest dam of India is?",

         "options": ["Sardar Sarovar Dam", "Tehri Dam", "Bhakra Nangal Dam", "None of these"],

         "answer": "Tehri Dam"},

        {"question": "Q3. Who heads the RBI?",

         "options": ["Home Minister", "Governor", "President ", "Finance Minister"],

         "answer": "Governor"},

        {"question": "Q4. On which date and year did the Titanic sink?",

         "options": ["18-April-1912", "14-April-1907", "14-April-1912", "18-April-1907"],

         "answer": "14-April-1912"},

        {"question": "Q5. Who is known as The God of Cricket?",

         "options": ["Virat Kohli", "MS Dhoni", "Sachin Tendulkar", "Rohit Sharma"],

         "answer": "Sachin Tendulkar"}

    ],

    "Round 3": [

        {"question": "Q1. The first AI programming language was called:",

         "options": ["BASIC", "FORTRAN", "IPL", "LISP"],

         "answer": "LISP"},

        {"question": "Q2. What was the first iPhone model to feature a fingerprint sensor?",

         "options": ["iPhone 5", "iPhone 6", "iPhone 4s", "iPhone 5s"],

         "answer": "iPhone 5s"},

        {"question": "Q3. Which Of The Following Does Not Belong To Java?",

         "options": ["Switch", "Double", "Instance Of ", "Then "],

         "answer": "Then "},

        {"question": "Q4. Artificial Intelligence Introduce In?",

         "options": ["1970", "1978", "1956", "1965"],

         "answer": "1956"},

        {"question": "Q5. Which data structure is used to implement recursion?",

         "options": ["Queue", "Stack", "Linked list", "Tree"],

         "answer": "Stack"}

        ]

}

# Global variables

current\_round = list(rounds.keys())[0]

current\_question\_index = 0

score = 0 # cumulative score

current\_round\_score = 0  # score in current round

round\_scores={"Round 1": 0,"Round 2": 0,"Round 3": 0}

# Function to check login

def check\_login():

    username = username\_entry.get()

    password = password\_entry.get()

    if username == USERNAME and password == PASSWORD:

        messagebox.showinfo("Login","Login Successful!\nWelcome to KBC \nMade by SANDEEP JAAT ")

        login\_frame.pack\_forget()

        quiz\_frame.pack()

        load\_question()

    else:

        messagebox.showerror("Login Failed", "Invalid Username or Password \nPlease try again.")

# Function to load questions

def load\_question():

    global current\_question\_index, current\_round

    if current\_round in rounds and current\_question\_index < len(rounds[current\_round]):

        q = rounds[current\_round][current\_question\_index]

        question\_label.config(text=f"{current\_round}: {q['question']}")

        answer\_var.set("")  # Reset selection

        selected\_option\_label.config(text="Selected Option: None")  # Reset displayed selection

        for i in range(len(q["options"])):

            options[i].config(text=q["options"][i], value=q["options"][i])

    else:

        end\_of\_round()

# Function to update selected option label

def show\_selected():

    selected\_option\_label.config(text=f"Selected Option: {answer\_var.get()}")

# Function to check the answer

def check\_answer():

    global current\_question\_index, score , current\_round\_score

    selected = answer\_var.get()

    if not selected:

        messagebox.showwarning("No Selection", "Please select an option before submitting!")

        return

    correct\_answer = rounds[current\_round][current\_question\_index]["answer"]

    if selected == correct\_answer:

        score += 10

        current\_round\_score += 10

        messagebox.showinfo("Correct!", f"Your answer is correct!\nYour score: {score}")

    else:

        messagebox.showerror("Wrong!", f"Wrong answer!\nCorrect answer: {correct\_answer}")

    current\_question\_index += 1

    load\_question()

# Function called when round ends

def end\_of\_round():

    global round\_scores, current\_round\_score

    round\_scores[current\_round] = current\_round\_score

    messagebox.showinfo("Round Over", f"{current\_round} is over!\nYour score in this round: {round\_scores[current\_round]}\nTotal score: {score}")

    ask\_next\_round()

# Function to ask user whether to go to next round

def ask\_next\_round():

    global current\_round, current\_question\_index, score,current\_round\_score

    round\_keys = list(rounds.keys())

    current\_index = round\_keys.index(current\_round)

    if current\_index + 1 < len(round\_keys):

        # Store the score of current round before moving to next

        previous\_scores = {k: round\_scores[k] for k in round\_scores if round\_scores[k] != 0}

        result = messagebox.askyesno("Next Round", "Do you want to go for the next round?")

        if result:

            current\_round = round\_keys[current\_index + 1]

            current\_question\_index = 0

            current\_round\_score=0

            # Build message according to which round is coming next

            message = ""

            if current\_round == "Round 2":

                message = f"Round 1 total score is {round\_scores['Round 1']}\n"

            elif current\_round == "Round 3":

                total\_so\_far = round\_scores["Round 1"] + round\_scores["Round 2"]

                message = (f"Round 1 score is {round\_scores['Round 1']}\n"

                           f"Round 2 score is {round\_scores['Round 2']}\n"

                           f"Total score till now is {total\_so\_far}\n")

            messagebox.showinfo("Starting Next Round",message )

            load\_question()

        else:

            show\_final\_score()

    else:

        show\_final\_score()

# Function to show final score at the end

def show\_final\_score():

    total = round\_scores["Round 1"] + round\_scores["Round 2"] + round\_scores["Round 3"]

    messagebox.showinfo("Game Over",

                        f"Final Scores:\n"

                        f"Round 1: {round\_scores['Round 1']}\n"

                        f"Round 2: {round\_scores['Round 2']}\n"

                        f"Round 3: {round\_scores['Round 3']}\n\n"

                        f"Total Score: {total}"

                        f"\nThank you for playing KBC!")

    root.quit()

# GUI Setup

root = tk.Tk()

root.title("Kaun Banega Crorepati")

root.geometry("600x450")

root.configure(bg="black")

# Login Frame

login\_frame = tk.Frame(root, bg="black")

tk.Label(login\_frame, text="KBC LOGIN", fg="yellow", bg="black", font=("Arial", 16)).pack(pady=10)

tk.Label(login\_frame, text="Username:", fg="white", bg="black").pack()

username\_entry = tk.Entry(login\_frame)

username\_entry.pack()

tk.Label(login\_frame, text="Password:", fg="white", bg="black").pack()

password\_entry = tk.Entry(login\_frame, show="\*")

password\_entry.pack()

tk.Button(login\_frame, text="Login", command=check\_login, fg="black", bg="yellow").pack(pady=10)

login\_frame.pack()

# Quiz Frame

quiz\_frame = tk.Frame(root, bg="black")

question\_label = tk.Label(quiz\_frame, text="", fg="yellow", bg="black", font=("Arial", 14), wraplength=500)

question\_label.pack(pady=10)

answer\_var = tk.StringVar()

options = []

for i in range(4):

    rb = tk.Radiobutton(quiz\_frame, text="", variable=answer\_var, value="", fg="white", bg="black", font=("Arial", 12), command=show\_selected)

    rb.pack()

    options.append(rb)

selected\_option\_label = tk.Label(quiz\_frame, text="Selected Option: None", fg="white", bg="black", font=("Arial", 12))

selected\_option\_label.pack(pady=5)

tk.Button(quiz\_frame, text="Submit", command=check\_answer, fg="black", bg="yellow").pack(pady=10)

root.mainloop()