**Student Name: Riya**  **UID: 24MCI10226**

**Branch: MCA(AIML) Section/Group: MAM4(A)**

**Semester: Ist Sem Date of Performance:**

**Subject Name: Python Programming Subject Code: 24CAH-606**

**PROJECT WORKSHEET**

**Title of Project:**

**Quiz Application in Python**

**Aim/Overview of the practical:**

The aim of this project is to create a simple quiz application using Python, where the user will be presented with multiple-choice questions. The quiz will evaluate the user's performance by keeping track of the correct answers and displaying the final score at the end.

**Task to be done:**

* Design a quiz application using Python.
* Define a set of multiple-choice questions.
* Present each question to the user one by one.
* Take the user's input as the answer.
* Keep track of the correct and incorrect answers.
* Display the user's final score at the end of the quiz.

**Code for experiment/practical**:

import tkinter as tk

from tkinter import messagebox

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# Sample quiz data

data = {

"question": [

"Who is the father of Computers?",

"What is smallest unit of the information?",

"Which of the following language does the computer understand?",

" Which of the following is the brain of the computer?",

"Which of the following is the smallest unit of data in a computer?"

],

"options": [

["James Gosling", "Charles Babbage", "Dennis Ritchie", "Bjarne Stroustrup"],

["A bit", "A byte", "A block", "A nibble"],

["Computer understands only C Language", "Computer understands only Assembly Language", "Computer understands only Binary Language", "Computer understands only BASIC"],

["Central Processing Unit", "Memory", "Arithmetic and Logic unit", "Control unit"],

["Bit", "150,000 km/s", "KB", "Nibble", "Byte"]

],

"answers": ["Charles Babbage", "A bit", "Computer understands only Binary Language", "Central Processing Unit", "Bit"]

}

class QuizApp:

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

self.master = master

self.master.title("AI Quiz")

self.question\_index = 0

self.score = 0

self.question\_label = tk.Label(master, text="", wraplength=400)

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

self.var = tk.StringVar()

self.options = [tk.Radiobutton(master, text="", variable=self.var, value=i) for i in range(4)]

for option in self.options:

option.pack(anchor='w')

self.submit\_button = tk.Button(master, text="Submit", command=self.submit\_answer)

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

self.show\_question()

def show\_question(self):

if self.question\_index < len(data["question"]):

self.question\_label.config(text=data["question"][self.question\_index])

for i, option in enumerate(self.options):

option.config(text=data["options"][self.question\_index][i])

option.select() # Reset selection

else:

self.show\_result()

def submit\_answer(self):

selected\_answer = self.var.get()

correct\_answer = data["answers"][self.question\_index]

if selected\_answer is not None and data["options"][self.question\_index][int(selected\_answer)] == correct\_answer:

self.score += 1

self.question\_index += 1

self.show\_question()

def show\_result(self):

messagebox.showinfo("Quiz Finished", f"Your score: {self.score}/{len(data['question'])}")

self.plot\_results()

def plot\_results(self):

labels = ['Correct', 'Incorrect']

sizes = [self.score, len(data['question']) - self.score]

plt.figure(figsize=(6, 6))

plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)

plt.axis('equal') # Equal aspect ratio ensures pie chart is circular.

plt.title("Quiz Results")

plt.show()

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

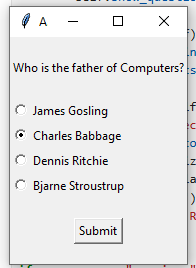
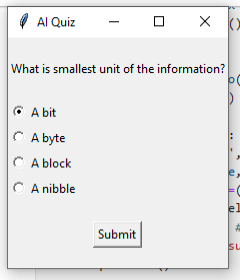
root = tk.Tk()

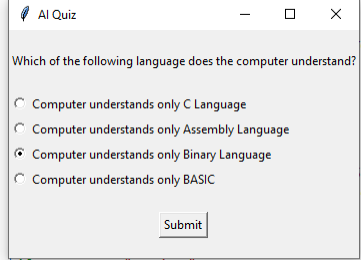
quiz\_app = QuizApp(root)

root.mainloop()

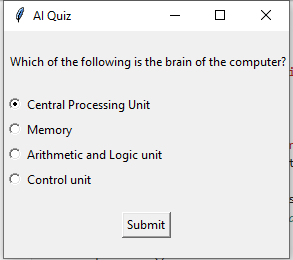
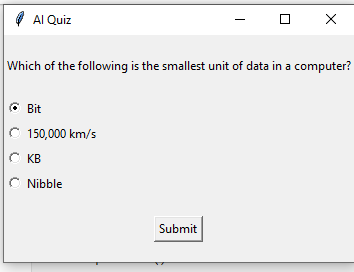
 

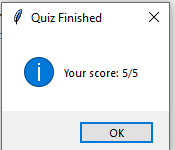
**Output:**

** **

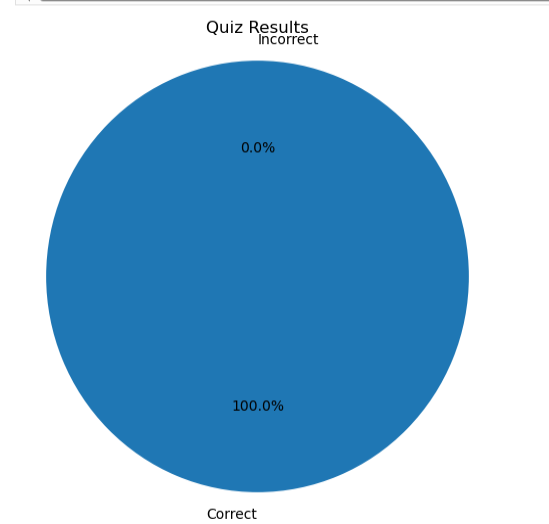
****

** **

****

****

**Learning Outcomes:**

How to structure and manage multiple-choice questions in Python using lists and dictionaries.

How to collect user input and validate it.

The use of conditional statements to compare answers.

Basic input/output operations in Python.

How to keep track of a score and provide feedback to the user.