Report On

Quiz App

Submitted in partial fulfillment of the requirements of the Course project in Semester IV of Second Year Computer Engineering

By
Pallavi Rajendra Dhandar (Roll no. 43)
Vaidehi Deepak Gadag (Roll No. 47)

Supervisor
Prof. Sneha Mhatre

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

(2023-24)

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

CERTIFICATE

This is to certify that the project entitled “Quiz app” is a bonafide work of "Pallavi Rajendra Dhandar (Roll no. 43),Vaidehi Deepak Gadag (Roll No. 47)” submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester IV of Second Year Computer Engineering.

Supervisor

Prof. Sneha Mhatre

Internal Examiner

Dr Megha Trivedi
Head of Department

Dr. H.V. Vankudre
Principal

ABSTRACT

The Quiz App is a Python program designed to help users enhance their math skills through interactive quizzes. Users can engage with a series of math questions presented by the app, testing their knowledge and problem-solving abilities. With a user-friendly interface and intuitive design, the Quiz App offers an engaging and educational experience for individuals looking to sharpen their math acumen. This program provides a versatile platform for users to practice and improve their mathematical proficiency.

The Quiz App is a software application developed using the Python programming language with the primary objective of assisting individuals in strengthening their mathematical skills by partaking in dynamic quizzes. The application enables users to interact with a succession of mathematical queries generated by the program, thereby assessing their understanding and aptitude for problem-solving. Through its simplistic interface and intelligible layout, the Quiz App ensures a stimulating and enlightening endeavor for those interested in honing their mathematical knowledge.

INDEX

Contents	Pg. No
1 Introduction	1
1.1 Introduction	
1.2 Problem Statement	
2 Proposed System	2
2.1 Block diagram , its description and working [ER diagram]	
2.2 Module Description	
2.3 Brief description of software & hardware used and its programming	
2.4 Code	
3 Results and conclusion	14
4 References	15

INTRODUCTION

1.1 Introduction

In the modern era, technology has brought about a transformative impact on the methods of learning and involvement in subjects such as mathematics. A noteworthy illustration of this progression is The Quiz App, which is a Python software application specifically crafted to support individuals in enhancing their mathematical abilities through interactive quizzes.

Users can put their knowledge to the test by answering a series of math questions ranging various basic arithmetic equations. This app not only serves as a fun way to challenge oneself but also provides a practical tool for improving mathematical proficiency. Through The Quiz App, users are presented with the chance to refine their capabilities conveniently and intriguingly, thereby rendering the process of math instruction more accessible and enjoyable within the digital landscape.

1.2 Problem Statement

Develop a Quiz App using Python that enables users to evaluate their math proficiency through a set of math-related queries. The current problem lies in providing an interactive and engaging platform for users to test their mathematical knowledge effectively. Additionally, ensuring the app's functionality, performance, and scalability to accommodate various difficulty levels and question types is paramount. The goal is to create an interactive learning tool that motivates users to enhance their math skills in a fun and engaging manner.

PROPOSED SYSTEM

2.1 Block diagram , its description and working [ER diagram]

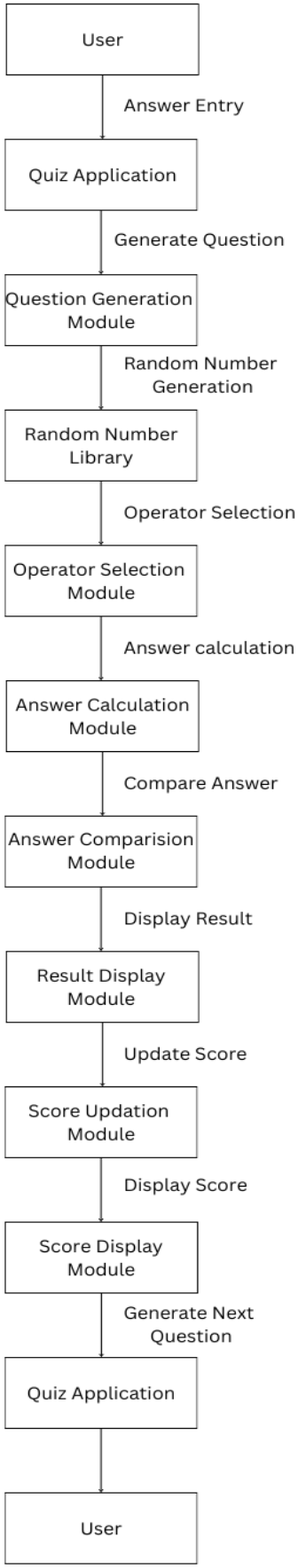
Description:

The Quiz App is a Python program designed for users to engage and challenge their math skills. With a user-friendly interface, individuals can test their mathematical knowledge by answering a variety of math questions. This interactive application offers a fun and educational way to practice arithmetic concepts. Whether you are a student looking to sharpen your skills or an adult wanting to keep your mind sharp, The Quiz App provides an engaging platform to enhance your mathematical calculation.

Working in Simple Words:

1. This is a simple math quiz app created using the tkinter library in Python.
2. The app displays a math question with an operator (+, -, *, /) and random numbers.
3. The user is expected to enter the answer in the provided text entry box and submit it to check if the answer is correct.
4. If the answer is correct, the user's score increases by 1, and a new question is displayed.
5. If the answer is incorrect, the user is notified, and a new question is displayed.
6. The quiz continues until 10 questions have been answered, and the user's final score is displayed.
7. The user can restart the quiz at any time.

Block Diagram:



2.2 Module Description

Here are the module descriptions for the quiz app:

1. **User Module:** This module represents the user of the quiz app. It includes the user's answer entry and display modules.

2. **Quiz Application Module:** This module is the main module of the quiz app. It includes the following sub-modules:
    - **Question Generation Module:** This module generates a new question by selecting random numbers and an operator.
    - **Answer Calculation Module:** This module calculates the answer to the generated question.
    - **Answer Comparison Module:** This module compares the user's answer to the correct answer and determines if the answer is correct or not.
    - **Result Display Module:** This module displays the result of the user's answer (correct or incorrect) on the screen.
    - **Score Updation Module:** This module updates the user's score based on the result of the user's answer.
    - **Score Display Module:** This module displays the user's score on the screen.

3. **Random Number Generation Module:** This module generates random numbers for the question generation module.

4. **Operator Selection Module:** This module selects an operator (+, -, *) for the question generation module.

5. **Tkinter Module:** This is a standard Python module for creating graphical user interfaces (GUIs). It is used to create the GUI for the quiz app.

These modules work together to create a functional quiz app that generates random math questions, compares the user's answers to the correct answers, updates the user's score, and displays the result and score on the screen.

2.3 Brief Description of Software & Hardware Used and Its Programming

Software:

The quiz app is programmed using Python, which is a high-level, interpreted language that is easy to learn and use. The quiz app's code is organized into modules, which are logical units of code that perform specific functions. The modules are written in Python and use standard Python syntax and constructs. The quiz app's code is designed to be modular, reusable, and easy to maintain. The code is also well-documented, with comments and documentation strings that explain the purpose and functionality of each module and function.

Python: The quiz app is a software application developed using the Python programming language. Python is a high-level, interpreted programming language that is widely used for web development, scientific computing, data analysis, artificial intelligence, and more. The quiz app uses several standard Python modules, including tkinter for creating the GUI, random for generating random numbers, and math for mathematical operations.

Tkinter: The quiz app's GUI is created using the tkinter module, which is a standard Python module for creating GUIs. The tkinter module provides a set of widgets, such as buttons, labels, text entry boxes, and scales, that can be used to create a functional and visually appealing GUI. The quiz app's GUI is designed to be intuitive and user-friendly, with clear instructions and feedback for the user.

Hardware:

The quiz app runs on a computer or mobile device with a compatible operating system, such as Windows, macOS, Linux, or Android. The hardware requirements for the quiz app are minimal, as it is a lightweight application that does not require significant computing resources. A computer or mobile device with a modern processor, a few gigabytes of RAM, and a few hundred megabytes of free storage space should be sufficient to run the quiz app smoothly.

2.4 Code

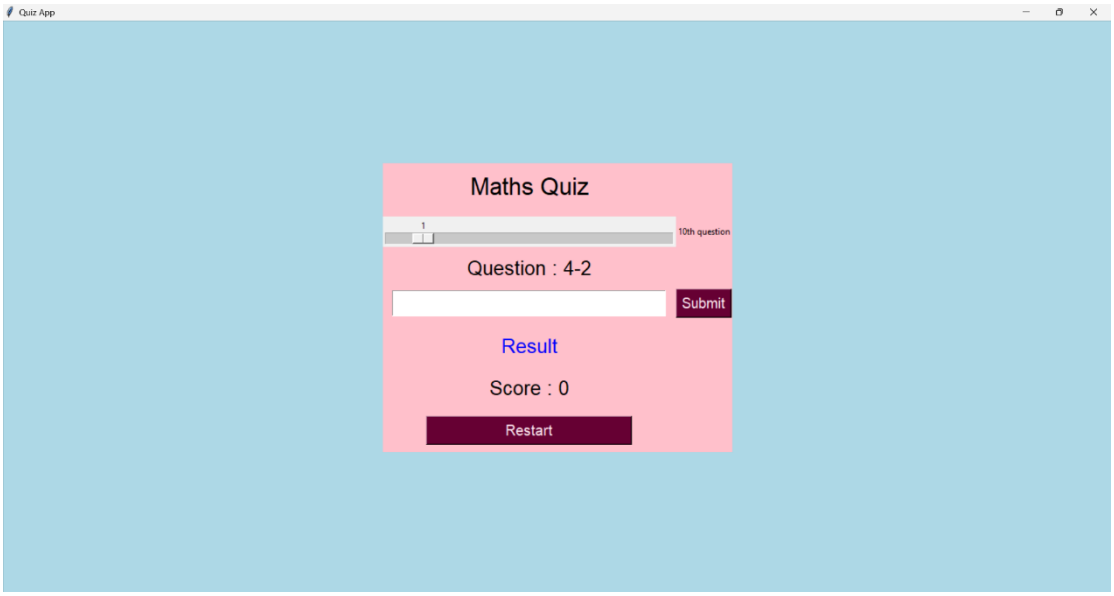
```
from tkinter import *
from random import randint , choice

root = Tk()
root.geometry("600x500")
root.title("Quiz App")

question = StringVar()
answer = StringVar()
givenAnswer = StringVar()
score = IntVar()
questionNumber = IntVar()

def generateQuestion():

    global questionLabel

    questionNumber.set(questionNumber.get() + 1)

    number1 = randint(1 ,10)
    number2 = randint(1 ,10)

    operator = choice(['+' , '-' , '*' ])

    question.set(str(number1) + operator + str(number2))
    answer.set(eval(question.get()))

    if questionLabel:
        questionLabel.destroy()

    questionLabel = Label(root , text=f"Question : {question.get()}" , font=('arial' , 20))
    questionLabel.grid(row=2 , column=0)

def checkAnswer():
```

```
    global scoreLabel

    if questionNumber.get() > 10:
        return

    global resultLabel

    if resultLabel:
        resultLabel.destroy()

    if str(answer.get()) == givenAnswer.get():
        print("Correct")
        score.set(score.get() + 1)
        resultLabel = Label(root, text="Right" , font=('arial' , 20) , fg="green")
        resultLabel.grid(row=4 , column=0)
        scoreLabel = Label(root, text=f"Score : {score.get()}" , font=('arial' , 20) , fg="black")
        scoreLabel.grid(row=5 , column=0)
    else:
        print("Incorrect")
        resultLabel = Label(root, text="Wrong" , font=('arial' , 20) , fg="red")
        resultLabel.grid(row=4 , column=0)

    if questionNumber.get() == 10:
        scoreLabel.destroy()
        scoreLabel = Label(root, text=f"Final Score : {score.get()}" , font=('arial' , 20) , fg="black")
        scoreLabel.grid(row=5 , column=0)
    else:
        generateQuestion()

def restart():
    global scoreLabel
    scoreLabel.destroy()
```

```
    score.set(0)
    questionNumber.set(0)
    generateQuestion()
    scoreLabel = Label(root, text=f"Score : {score.get()}" , font=('arial' , 20) , fg="black")
    scoreLabel.grid(row=5 , column=0)

# User Interface

headingLabel = Label(root , text="Maths Quiz" , font=('arial' , 25))
headingLabel.grid(row=0 , column=0)

questionScale = Scale(root , from_=0 , to=10 , orient=HORIZONTAL , length=400 , variable=questionNumber)
questionScale.grid(row=1 , column=0)

completeQuestionLabel = Label(root , text="10th question")
completeQuestionLabel.grid(row=1 , column=1)

questionLabel = Label(root , text=question.get() , font=('arial' , 20))
questionLabel.grid(row=2 , column=0)

answerEntry = Entry(root , textvariable=givenAnswer , font=('arial' , 20) , width=25)
answerEntry.grid(row=3 , column=0)

submitButton = Button(root , text="Submit" , fg="yellow" , bg = "grey" , font=('arial' , 15) , command=checkAnswer)
submitButton.grid(row=3 , column=1)

resultLabel = Label(root, text="Result" , font=('arial' , 20) , fg="blue")
resultLabel.grid(row=4 , column=0)

scoreLabel = Label(root, text=f"Score : {score.get()}" , font=('arial' , 20) , fg="black")
scoreLabel.grid(row=5 , column=0)
```

```
submitButton = Button(root , text="Restart" , fg="red" , font=('arial' , 15) , width= 25 , command=restart)
submitButton.grid(row=6 , column=0)

generateQuestion()

root.mainloop()
```

RESULT AND CONCLUSION



The Quiz App is a simple Python program that allows users to test their math skills. It generates random math questions, checks the user's answers, and keeps track of the score. The app provides a user-friendly interface using Tkinter and demonstrates the use of variables, functions, and random number generation in Python programming.

REFERENCES

For Web References

https://docs.python.org/3/library/tk.html

https://docs.python.org/3/library/random.html

www.geeksofgeeks.com