# **Documentație: Trivia Quiz folosind Tkinter**

## **1. Descrierea aplicației**

Trivia Quiz este o aplicație interactivă scrisă în Python folosind biblioteca `tkinter`.   
Aplicația afișează o serie de întrăbări cu variante multiple de răspuns, oferă un cronometru de 15 secunde pentru fiecare întrebare și permite utilizatorului să-și verifice cunoștințele.  
  
Contribuții la dezvoltare:  
- \*\*Un membru al echipei\*\* a lucrat la implementarea interfeței utilizator și a funcției de cronometru (`update\_timer`).  
- \*\*Celălalt membru al echipei\*\* a dezvoltat funcțiile pentru gestionarea întrebărilor și răspunsurilor: `next\_question`, `show\_score`, `load\_question` și `check\_answer`.

## **2. Cerințe de rulare**

1. Python 3.6+ instalat pe sistem.  
2. Biblioteca `tkinter` (inclusă implicit în Python standard).  
  
### Pași pentru rulare:  
1. Salvați codul într-un fișier Python, de exemplu, `trivia\_quiz.py`.  
2. Deschideți un terminal sau o fereastră de comandă.  
3. Executați fișierul cu comanda:  
 ```bash  
 python trivia\_quiz.py  
 ```

## **3. Descrierea funcțiilor și fluxul aplicației**

1. `load\_question()` - Încărcă întrebarea curentă și actualizează textul butoanelor de opțiuni.  
2. `check\_answer(selected\_option)` - Validează răspunsul utilizatorului comparându-l cu răspunsul corect.  
3. `update\_timer()` - Actualizează cronometrul afișat și gestionează expirarea timpului.  
4. `show\_score()` - Afișează scorul final și dezactivează toate butoanele din interfață.

## **4. Referințe utilizate**

- Documentația oficială `tkinter`: https://docs.python.org/3/library/tkinter.html  
- Tutorial Tkinter: https://realpython.com/python-gui-tkinter/  
- Video YouTube despre quiz-uri: https://www.youtube.com/watch?v=gfV1a3ri1tk

## **5. Note adiționale**

- Codul poate fi extins pentru a include mai multe întrăbări, categorii, sau niveluri de dificultate.  
- Persistența scorurilor utilizatorilor într-un fișier sau bază de date poate fi adăugată pentru o funcționalitate suplimentară.  
- Interfața poate fi personalizată folosind stiluri avansate sau alte biblioteci (de exemplu, `ttk`).

## **6. Integrare cu GitHub și testare automată**

### GitHub:  
Codul poate fi găzduit pe o platformă precum GitHub pentru versionare și colaborare. Exemplu de pași pentru publicare:  
  
1. Inițializați un repository:  
 git init  
  
2. Adăugați fișierele proiectului și faceți commit:  
 git add .  
 git commit -m "Initial commit"

3. Conectați-vă la un repository GitHub și împingeți codul:  
 git remote add origin <URL-ul-repository-ului>  
 git push -u origin main  
  
### Testare automată cu `pytest`:  
  
Aplicația poate fi testată automat utilizând `pytest`. Exemple de teste care pot fi create:  
  
1. \*\*Testarea validării răspunsurilor:\*\*  
 - Verificați dacă funcția `check\_answer()` actualizează corect scorul utilizatorului pentru răspunsuri corecte și greșite.  
  
2. \*\*Testarea cronometrelor:\*\*  
 - Simulați expirarea timpului și verificați dacă funcția `update\_timer()` apelează `next\_question()` corect.  
  
3. \*\*Testarea întrebărilor și opțiunilor:\*\*  
 - Verificați dacă funcția `load\_question()` afișează corect întrebările și opțiunile.  
  
Pentru a utiliza `pytest`:  
 pip install pytest  
 pytest test\_trivia\_quiz.py