

**Universitatea Tehnică de Construcții din București**

**Facultatea de Hidrotehnică**

**Specializarea: Automatică și Informatică Aplicată**

**Proiect la LP2**

**Profesor coordonator: Student:**

**Fudulu Cosmin Vlaicu Maria Georgiana**

**Olteanu Gabriela**

**București, 2025**



**Universitatea Tehnică de Construcții din București**

**Facultatea de Hidrotehnică**

**Specializarea: Automatică și Informatică Aplicată**

**Trivia Quiz**

**Profesor coordonator: Student:**

**Maria-Georgiana VLAICU**

**București, 2025**

# 

# CUPRINS

Introducere 4

Motivarea alegerii temei 4

Obiectivele propuse in cadrul lucrarii 4

Structura 5

1. Tehnologii si instrumente folosite in implementarea proiectului 6

1.1 C++ 6

1.2 Visual Studio Code 6

2. Logica jocului 8

3. Detalii despre cod 9

3.1 Bibliotecile folosite 9

3.2 Funcția main() 9

3.3 Funcția intrebariQuiz() 12

3.4 Fișierele de întrebări 14

4. Imagini din joc 15

5. Posibile îmbunătățiri 16

6. Concluzii 18

Bibliografie 19

# 

# Introducere

## Motivarea alegerii temei

Tema aleasa pentru acest proiect este realizarea unui joc de tipul trivia quiz, folosind limbajul de programare C++. Am ales să fac un proiect de tip trivia quiz în C++ pentru că mi s-a părut o idee interesantă și ușor de pus în practică. Îmi plac jocurile de cultură generală și am vrut să încerc să fac unul simplu, în care utilizatorul să răspundă la întrebări și să primească un scor la final. Am ales această temă și pentru că îmi place ideea de a crea ceva interactiv, care poate fi testat ușor de prieteni sau colegi.

Alegerea acestei teme mi s-a părut un mod util și practic de a exersa cunoștințele dobândite în C++, precum și de a le îmbunătăți prin aplicarea lor într-un proiect concret. Totodată, un quiz este o temă ușor de extins, la care pot adăuga funcții noi, cum ar fi diferite niveluri de dificultate, un cronometru pentru răspunsuri sau întrebări împărțite pe categorii, în funcție de ce își dorește utilizatorul.

## Obiectivele propuse in cadrul lucrarii

Principalele obiective pe care mi le-am propus în cadrul acestui proiect sunt:

* Dezvoltarea unei aplicații simple în C++ care permite utilizatorului să răspundă la întrebări și să primească un scor final în funcție de răspunsurile corecte.
* Aplicarea și consolidarea cunoștințelor în limbajul C++, folosind concepte precum funcții, lucrul cu fișiere, manipularea string-urilor și condiționale.
* Utilizarea fișierelor externe pentru întrebări, astfel încât aplicația să fie ușor de întreținut și extins fără modificări în codul sursă.
* Crearea unei interfețe prietenoase în linia de comandă, prin folosirea culorilor și a pauzelor între întrebări, pentru o experiență mai plăcută pentru utilizator.
* Implementarea unui sistem de dificultate, care să permită utilizatorului să aleagă între mai multe niveluri de quiz, în funcție de cunoștințele sale.
* **Organizarea clară a codului**, pentru a-l face ușor de înțeles, modificat și extins în viitor.

## Structura

Lucrarea este structurată în șase capitole, fiecare abordând aspecte esențiale ale dezvoltării proiectului, precum și un capitol final care conține concluziile.

* În cadrul primului capitol, am prezentat tehnologiile și instrumentele utilizate pentru realizarea proiectului, în special limbajul de programare C++ și mediul de dezvoltare Visual Studio Code. Acestea au fost alese datorită eficienței, ușurinței în utilizare și compatibilității cu cerințele aplicației.
* În capitolul al doilea, am descris logica de funcționare a jocului quiz, explicând modul în care utilizatorul interacționează cu aplicația. Sunt evidențiate pașii principali ai jocului, de la afișarea întrebărilor până la calculul scorului final.
* Capitolul al treilea este dedicat detaliilor tehnice ale codului sursă. Am explicat rolul fiecărei funcții importante, utilizarea fișierelor externe și am prezentat modul de structurare clară a codului pentru o mai bună întreținere și extindere.
* În capitolul patru, am inclus imagini relevante din timpul rulării aplicației. Acestea oferă o viziune clară asupra interfeței text și modului în care utilizatorul interacționează cu aplicația în timp real.
* Capitolul cinci se axează pe posibile direcții de îmbunătățire ale proiectului. Sunt propuse idei precum adăugarea unui cronometru, salvarea rezultatelor sau diversificarea nivelurilor de dificultate, pentru a face jocul mai atractiv.
* Capitolul șase include concluziile personale desprinse din realizarea proiectului. Sunt evidențiate cunoștințele dobândite pe parcurs, dificultățile întâmpinate și felul în care acest proiect a contribuit la progresul meu în învățarea și aplicarea limbajului C++.

# TEHNOLOGII șI INSTRUMENTE FOLOSITE îN IMPLEMENTAREA PROIECTULUi

## C++

## C++ este un limbaj de programare compilat, care susține atât paradigma procedurală, cât și cea orientată pe obiect. Creat de Bjarne Stroustrup, C++ reprezintă o extensie a limbajului C, aducând îmbunătățiri semnificative precum clasele, funcțiile virtuale, supraîncărcarea operatorilor, template-urile și gestionarea excepțiilor.

## Acesta poate fi utilizat pentru a exprima idei complexe și pentru a dezvolta aplicații eficiente, fiind preferat în multe domenii, de la dezvoltarea software-ului de înaltă performanță la sisteme care necesită o utilizare intensivă a resurselor.

Avantaje:

* Performanță înaltă: C++ este recunoscut pentru performanțele sale excelente și este folosit în aplicații unde rapiditatea și eficiența sunt esențiale.
* Scalabilitate: C++ oferă o scalabilitate deosebită, putând gestiona atât seturi mici de date, cât și volume mari de informații.
* Programare multi-paradigmă: C++ sprijină atât programarea procedurală, cât și cea orientată pe obiecte, oferind dezvoltatorilor flexibilitate în alegerea abordării potrivite.
* Utilizare pe scară largă în industrie: Multe companii majore, precum Google și Facebook, utilizează C++ pentru dezvoltarea backend-urilor lor.
* Biblioteci puternice: C++ dispune de o gamă variată de biblioteci standard și externe, care pot extinde funcționalitățile codului.

## 1.2 Visual Studio Code

Visual Studio Code (VS Code) este un editor de cod gratuit, dezvoltat de Microsoft, care funcționează pe sistemele de operare Windows, macOS și Linux. Acesta este ușor de utilizat și rapid, oferind suport pentru limbaje de programare precum JavaScript, TypeScript și Node.js.

VS Code permite adăugarea de extensii pentru a suporta și alte limbaje de programare, cum ar fi C++, Python, Java sau PHP. Printre funcțiile importante ale acestui editor se numără completarea automată a codului, debugging-ul (corectarea erorilor) și integrarea cu Git pentru gestionarea versiunilor.

Avantaje:

* VS Code este ușor de folosit, având o interfață simplă și intuitivă, fiind accesibil atât pentru începători, cât și pentru dezvoltatori experimentați.
* VS Code oferă suport încorporat pentru multe limbaje de programare, iar prin extensii, poate fi utilizat pentru o gamă largă de limbaje (C++, Python, Java, etc.).
* VS Code include un sistem de debugging ușor de folosit, care ajută la identificarea și corectarea erorilor direct din editor.

# LOGICA JOCULUI

Jocul Trivia Quiz este un joc de tip quiz, în care jucătorul are ocazia să răspundă la un set de întrebări din diverse domenii, scopul fiind să obțină un scor cât mai bun. La lansarea aplicației, jucătorul trebuie să îți introducă numele, pentru a putea personaliza mesajele afișate pe parcursul jocului, oferind o experiență mai plăcută.

După introducerea numelui, jucătorul trebuie să aleagă nivelul de dificultate al quiz-ului. Sunt disponibile două niveluri: mediu și dificil. În funcție de opțiunea aleasă, acesta va primi un set de 12 întrebări adaptate nivelului selectat. Fiecare întrebare vine însoțită de patru variante de răspuns, marcate cu literele a, b, c și d, dintre care doar una este corectă. Dacă răspunsul introdus de jucător corespunde răspunsului corect predefinit, jocul afișează un mesaj pozitiv și adaugă un punct la scorul total. Dacă răspunsul este greșit, aplicația afișează varianta corectă, pentru a oferi feedback imediat.

La finalul celor 12 întrebări, jocul calculează scorul total obținut și afișează un mesaj de final. Dacă jucătorul a răspuns corect la mai mult de jumătate dintre întrebări, primește un mesaj de felicitări, iar dacă scorul este sub jumătate, acesta este îcurajat să încerce din nou. La final, aplicația afișează numărul total de întrebări și câte răspunsuri corecte a dat jucătorul.

# DETALII DESPRE COD

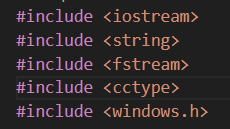
Aplicația **Trivia Quiz** este structurată în două funcții principale: main(), care controlează interacțiunea generală cu utilizatorul și logica de decizie a aplicației și intrebariQuiz(), care gestionează procesarea întrebărilor, afișarea acestora și verificarea răspunsurilor. Această structură face codul mai ușor de întreținut și mai ușor de înțeles.

## 3.1 Bibliotecile folosite

În cadrul acestui proiect în C++, am utilizat mai multe biblioteci din standardul limbajului C++ și o bibliotecă specifică sistemului de operare Windows, pentru a asigura o funcționalitate completă și o interacțiune cât mai prietenoasă cu utilizatorul.

Biblioteciile folosite sunt:

* **<iostream>** - pentru afișarea mesajelor și preluarea datelor de la tastatură, facilitând interacțiunea utilizatorului cu aplicația.
* **<string>** - pentru manipularea textului și a răspunsurilor. Tipul string este folosit pentru a reprezenta întrebările, variantele de răspuns și răspunsurile utilizatorului.
* **<fstream> -** permite citirea întrebărilor din fișiere externe (mediu.txt, dificil.txt) prin ifstream. Astfel, întrebările sunt ușor de actualizat fără a modifica sursa codului.
* **<cctype> -** folosită pentru funcția tolower() care transformă prima literă a răspunsului în literă mică, asigurând o verificare mai flexibilă a răspunsului (indiferent dacă utilizatorul introduce „A”, „a”, etc.).
* **<windows.h>** Bibliotecă specifică Windows, utilizată pentru funcția Sleep(ms) care introduce pauze între afișări, oferind un ritm mai natural și o experiență mai plăcută.

 Figura 3.1- Bibliotecile folosite

## 3.2 Funcția main()

Funcția main() reprezintă punctul de pornire al aplicației și locul în care se stabilesc primele interacțiuni cu utilizatorul. Rolul acesteia este de a gestiona logica principală a aplicației, de la afișarea mesajului introductiv și până la prezentarea scorului final.

La început, aplicația afișează un mesaj de bun venit, colorat special în terminal (folosind coduri ANSI pentru culori), cu scopul de a atrage atenția și de a crea o experiență vizuală mai plăcută. Apoi, utilizatorul trebuie să își introducă numele, ceea ce permite personalizarea mesajelor ulterioare. În continuare acesta trebuie să aleagă gradul de dificultate al întrebarilor (mediu, dificil). După ce a făcut acest lucru, aplicația afișează un mesaj de încurajare urmat de o pauză de două secunde realizată cu ajutorul funcției Sleep(2000). Această pauză are rolul de a oferi utilizatorului un mic moment de pregătire înainte de începerea propriu-zisă a întrebărilor.

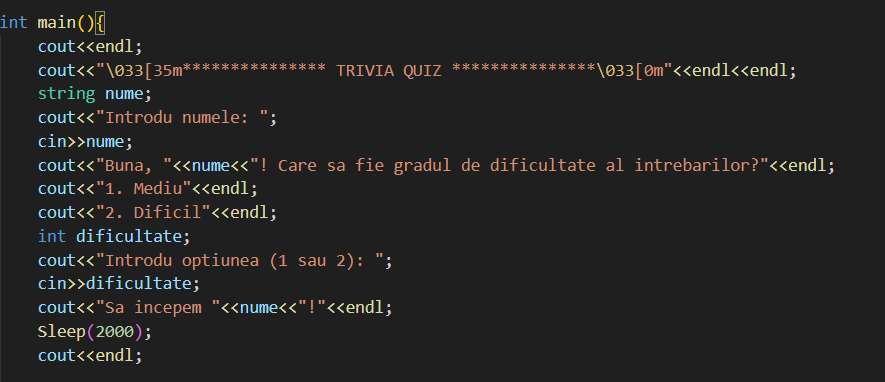


Figura 3.2.1 – Partea de introducere din main()

După ce utilizatorul și-a ales opțiunea privind dificultatea întrebărilor, selecția sa este preluată și tratată prin intermediul instrucțiunii switch. Această structură de control este utilizată pentru a gestiona în mod eficient mai multe căi de execuție, în funcție de valoarea introdusă de utilizator. În acest caz, fiecare ramură din instrucțiunea switch corespunde unui nivel de dificultate: 1 pentru întrebări de nivel mediu și 2 pentru întrebări dificile.

Prin acest mecanism, aplicația face legătura directă între opțiunea selectată și fișierul corespunzător care conține întrebările (mediu.txt sau dificil.txt), apelând funcția intrebariQuiz() cu numele fișierului adecvat. Astfel, întrebările sunt încărcate automat, fără a repeta secvențe de cod pentru fiecare caz în parte. Acest lucru face codul mai clar, mai ușor de întreținut și mai scalabil – putând fi adăugate ușor și alte niveluri în viitor.

În plus, instrucțiunea default din switch se ocupă de situațiile în care utilizatorul introduce o opțiune invalidă. Astfel, aplicația afișează un mesaj de avertizare și se oprește, prevenind eventuale erori sau comportamente neașteptate. Această abordare asigură un control clar al fluxului aplicației și o experiență de utilizare robustă.

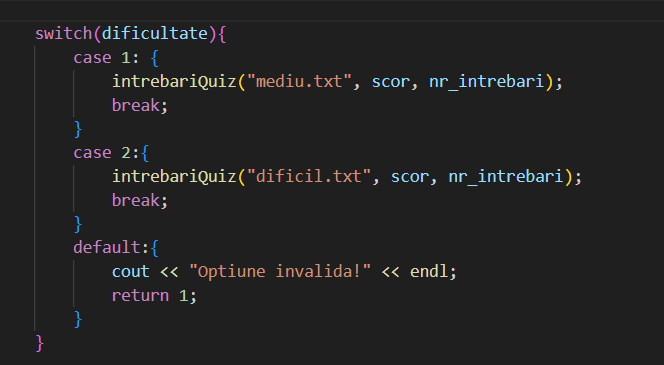


Figura 3.2.2 – Instrucțiunea switch

La finalul quiz-ului, programul afișează rezultatul într-un mod simplu și sugestiv. Dacă scorul obținut de utilizator este mai mare decât jumătate din numărul total de întrebări, acesta primește un mesaj de felicitări. În caz contrar, este afișat un mesaj de încurajare ( „Mai mult noroc data viitoare!” ). După afișarea mesajului motivațional, aplicația prezintă utilizatorului și numărul exact de răspunsuri corecte obținute din totalul întrebărilor parcurse, oferind astfel o fimagine clară și completă asupra performanței sale.

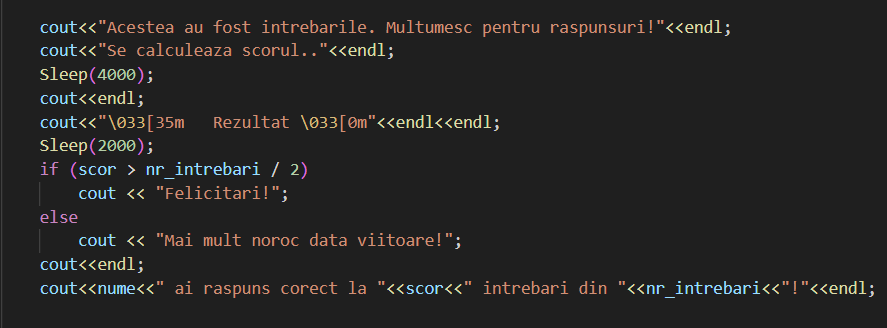


Figura 3.2.3 – Afișare mesaje și scor la finalul jocului

## 3.3 Funcția intrebariQuiz()

Această funcție are rolul de a citi întrebările din fișier, de a le afișa și de a verifica răspunsurile introduse de utilizator. Se ocupă cu citirea fișierului linie cu linie, verificarea corectitudinii răspunsurilor și actualizarea scorului.

Antetul funcției este: int intrebariQuiz(const string& numeFisier, int& scor, int& nr\_intrebari), unde: primul parametru, const string& numeFisier, reprezintă numele fișierului din care sunt citite întrebările. Parametrul este de tip string pentru a putea stoca texte, este transmis prin referință (&), și este declarat const deoarece funcția nu modifică acest text, doar îl citește.

Următorii doi parametri, int& scor și int& nr\_intrebari, sunt variabile de tip întreg transmise tot prin referință. Acest lucru înseamnă că modificările făcute asupra acestor variabile în interiorul funcției vor afecta direct valorile din main.

Înainte să înceapă executarea propriu-zisă a funcției intrebariQuiz, este necesară verificarea existenței fișierului transmis ca parametru în antetul funcției. Acest lucru se face cu ajutorul unei instrucțiuni if, care verifică dacă fișierul a fost deschis corect. În cazul în care fișierul nu există sau nu poate fi accesat programul va afișa un mesaj de eroare și va opri executarea funcției.

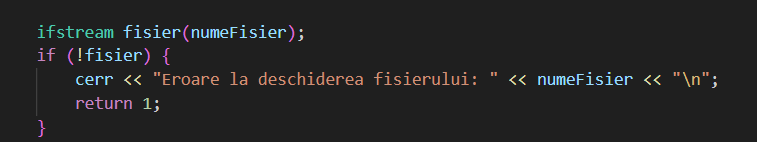


Figura 3.3.1 Verificare deschidere fișier

După ce fișierula fost deschis, pentru a citi conținutul acestuia, am folosit o buclă while cu funcția getline() care extrage câte o linie pe rând. Fiecare întrebare este formată din 6 linii: enunțul, patru variante și răspunsul corect. Această structură oferă claritate codului și îl face mai ușor de înțeles.

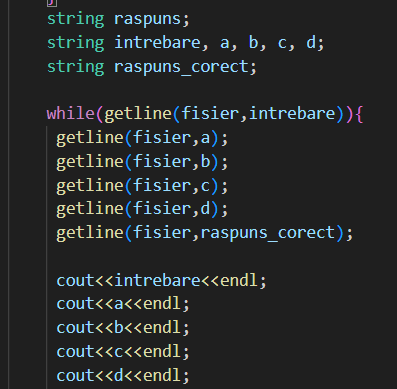


Figura 3.3.2 - Citirea și afișarea întrebărilor

După afișarea întrebării și preluarea răspunsului, programul îi cere jucătorului să își introducă răspunsul. Înainte de a verifica dacă răspunsul este corect, se face conversia primei litere la minusculă cu tolower() pentru a permite compararea chiar dacă utilizatorul a introdus litere mari. După convertirea răspunsului, se trece la verificarea acestuia prin intermediul instrucțiunii if. Dacă jucătorul a introdus varianta corectă se afișează mesajul „Corect!”, iar scorul crește, în caz contrar, se va afișa afișa mesajul „Greșit”, urmat de răspunsul corect. Mesajele sunt colorate Corect! (verde) și Greșit! (roșu) pentru a-i oferi jucătorului un semnal vizual rapid despre răspunsurile sale, făcând jocul mai clar și mai plăcut. La finalul instrucțiunii numărul total de întrebări crește.

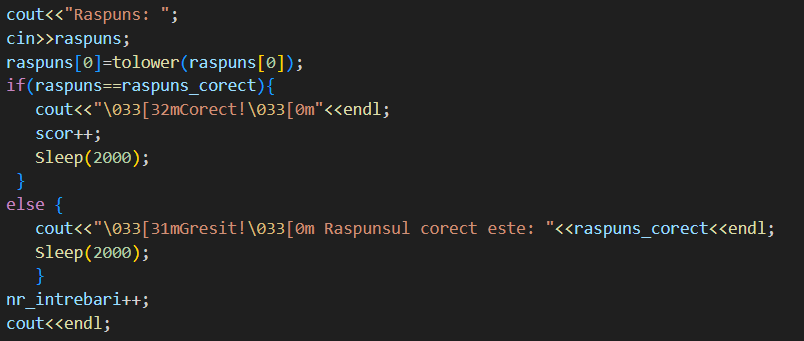
 De asemenea, a fost folosită funcția Sleep(2000) pentru a face o pauză de 2 secunde între întrebări, astfel încât jucătorul să aibă timp să vadă dacă a răspuns corect sau greșit, înainte de a trece la următoarea întrebare.

Figura 3.3.3 – Verificare răspuns

## 3.4 Fișierele de întrebări

Întrebările sunt stocate în fișierele text mediu.txt și dificil.txt. Acestea au o structură fixă: fiecare întrebare ocupă exact șase rânduri. Primele cinci rânduri reprezintă întrebarea și variantele de răspuns, iar al șaselea este litera variantei corecte.

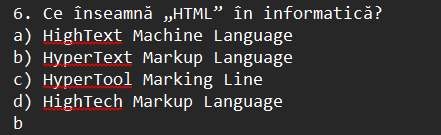
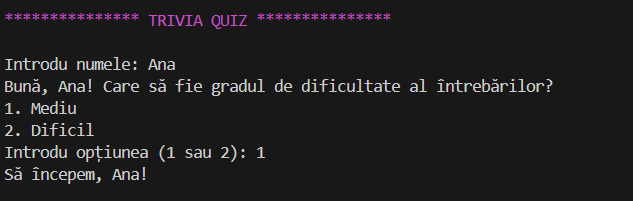
 Am ales să folosesc fișiere externe deoarece acest lucru face codul mai ușor de întreținut și permite adăugarea sau modificarea întrebărilor fără a fi nevoie de modificarea codului. Astfel, quiz-ul poate fi extins cu noi seturi de întrebări sau corectat foarte ușor, doar editând fișierele text.

Figura 3.4 – Exemplu de întrebare din fișier

# Imagini din joc



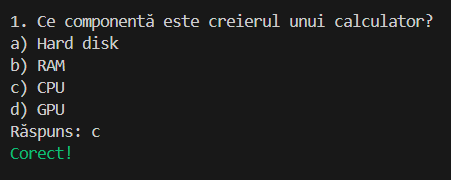
Figura 4.1 – Introducere nume și alegerea gradului de dificulate

Figura 4.2 – Întrebare corectă

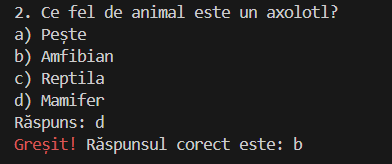


Figura 4.3 – Întrebare greșită

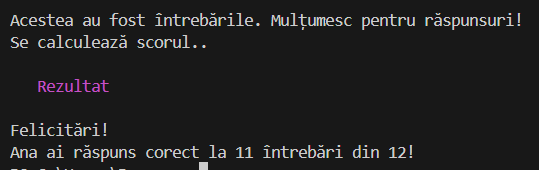


Figura 4.4 – Afișare rezultat

# Posibile îmbunătățiri

* Salvarea rezultatelor într-un fișier

O funcționalitate utilă pentru aplicație ar fi salvarea rezultatelor obținute de fiecare jucător într-un fișier extern. În acest fișier pot fi înregistrate informații precum numele utilizatorului și scorul obținut. Acest lucru ar fi util pentru păstrarea unui istoric al jucătorilor și pentru generarea unor statistici sau clasamente care să reflecte performanțele acestora de-a lungul timpului.

* Adăugarea mai multor niveluri de dificultate

Pe lângă cele două niveluri de dificultate deja existente („mediu” și „dificil” ) s-ar putea adăuga încă două: „ușor”, pentru începători, și „expert”, pentru utilizatorii avansați. Astfel, aplicația ar oferi o experiență de joc mai variată și adaptată unui public mai larg. Fiecare utilizator ar putea alege nivelul care i se potrivește cel mai bine, iar quiz-ul ar deveni astfel mai atractiv și provocator pentru toți, indiferent de experiența sau cunoștințele pe care le au în acel moment.

* Cronometrarea timpului

Adăugarea unui cronometru ar îmbunătăți semnificativ dinamica jocului, introducând un element de presiune. Se poate măsura fie timpul total alocat pentru tot quiz-ul, fie un timp limitat pentru fiecare întrebare în parte. Dacă jucătorul nu răspunde în intervalul stabilit, întrebarea ar fi considerată greșită automat. Această funcționalitate ar face jocul mai antrenant și ar testa nu doar cunoștințele jucătorului, ci și capacitatea de reacție sub presiune. Astfel, quiz-ul ar deveni mai competitiv și mai captivant.

* Întrebări din mai multe categorii

În loc să fie împărțite pe niveluri de dificultate, întrebările ar putea fi grupate pe categorii precum „istorie”, „știință”, „filme”, „geografie” și altele. Fiecare categorie ar avea propriul set de întrebări, organizat separat. Astfel, utilizatorul are posibilitatea să aleagă domeniul care îl interesează cel mai mult, ceea ce face experiența jocului mai plăcută și mai personalizată.

* Interfață grafică (GUI)

În prezent, aplicația funcționează în consolă. Totuși, experiența ar putea fi mult mai plăcută dacă s-ar crea o versiune cu interfață grafică. Astfel, quiz-ul ar putea avea butoane, ferestre, culori și imagini, ceea ce îl face mai atractiv și mai ușor de folosit. O interfață grafică ar ajuta jucătorul să înțeleagă mai repede cum funcționează aplicația și transformă totul într-o experiență mai modernă și prietenoasă.

# Concluzii

Realizarea acestui proiect a reprezentat mai mult decât un simplu exercițiu practic, a fost o ocazie reală de a învăța prin aplicare și de a înțelege mai bine cum pot transforma teoria în ceva concret. Am reușit să pun în aplicare noțiunile învățate în C++, într-un mod care mi-a testat atât cunoștințele tehnice, cât și gândirea logică și organizarea.

Prin intermediul acestui quiz, am învățat cum să folosesc fișiere externe pentru a separa datele de cod, ceea ce mi-a oferit o perspectivă mai clară asupra modularității și a importanței organizării unui proiect. Am exersat folosirea funcțiilor pentru a evita repetarea codului și pentru a menține o structură clară și ușor de urmărit. În plus, m-am concentrat pe detalii legate de experiența utilizatorului: mesaje clare, pauze între întrebări, culori care fac interfața mai prietenoasă, toate acestea contribuind la o interacțiune mai plăcută și mai naturală.

A fost o provocare să gândesc pas cu pas tot procesul, de la citirea întrebărilor din fișiere până la verificarea răspunsurilor și afișarea rezultatelor. Totuși, satisfacția de a vedea cum aplicația funcționează și poate fi folosită de oricine m-a motivat să continui și să îmbunătățesc tot ce am construit.

Pentru viitor, am în plan să dezvolt proiectul prin adăugarea unor funcționalități suplimentare, cum ar fi salvarea scorurilor într-un fișier, posibilitatea de a relua jocul, introducerea unui cronometru pentru fiecare întrebare sau chiar realizarea unei interfețe grafice simple. Cred că toate aceste îmbunătățiri ar duce aplicația la un alt nivel și ar face-o și mai interesantă.

În concluzie, acest proiect m-a ajutat să-mi consolidez în mod practic cunoștințele de programare. A fost o experiență valoroasă, care mi-a arătat cât de important este să experimentezi, să greșești, să înveți și, cel mai important, să te bucuri de ceea ce creezi.

Partea superioară a formularului

Partea inferioară a formularului

# Bibliografie

<https://www.newtech.ro/blog-ce-este-c/>

<https://infogenius.ro/introducere-cpp/>

<https://github.com/Mubeen-Channa/Quiz-Game-Project-in-Cpp/blob/main/Quiz_Game.cpp>

<https://psycatgames.com/>

<https://www.kudika.ro/articol/beautiful-minds/80508/intrebari-de-cultura-generala.html>

<https://www.kudika.ro/quiz/test_g4000-15-intrebari-de-cultura-generala-care-iti-vor-da-batai-de-cap/index.html>

<https://www.shtiu.ro/200-de-intrebari-de-cultura-generala-cu-raspunsuri-163462.html>