

Documentație QuizzGame

Pavel Petronel - Costel A3

Facultatea de Informatică, Universitatea “Alexandru Ioan Cuza”, Strada General
Berthelot, nr. 16, Iași, România secretariat@info.uaic.ro
<https://www.info.uaic.ro>

Abstract. QuizzGame este o aplicație cu întrebări de tip grilă, fiecare jucător va trebui să răspundă la 5 întrebări, având la dispoziție 20 de secunde pentru a răspunde la fiecare dintre acestea. Pentru implementarea acestei aplicații s-a folosit protocolul TCP, fire de execuție și baza de date SQLite.

Keywords: QuizzGame · TCP/IP · SQLite.

1 Introducere

QuizzGame este o aplicație de tip client-server care oferă posibilitatea clienților de a își testa cunoștințele unul împotriva celorlalți.

Fiecare utilizator își va seta un username, după care va trebui să răspundă la un set de 5 întrebări. Timpul alocat pentru a răspunde la fiecare dintre întrebări este de 20 de secunde. Dacă acest timp este depășit, utilizatorul este descalificat și nu va mai putea răspunde la următoarele întrebări.

La final, fiecare utilizator își va putea vedea punctajul, respectiv câștigătorul/câștigătorii sesiunii respective.

2 Tehnologiile utilizate

În cadrul acestei aplicații, comunicarea dintre client și server se realizează prin intermediul protocolului TCP (“Transmission Control Protocol”). Acesta este un protocol de comunicații orientat-conexiune care facilitează schimbul de date dintr-o rețea și este cel mai comun protocol care utilizează IP.

Aplicația utilizează acest protocol din următoarele motive:

- Stabilește o conexiune sigură între client și server
- Asigură transmiterea datelor în întregime între client și server

Pentru ca serverul să servească concomitent mai mulți clienți ne vom folosi de thread-uri.

Întrebările, împreună cu variantele de răspuns ale acestora sunt stocate într-o bază de date SQLite.

3 Arhitectura aplicației

Funcția prin care creăm baza de date

```

1 void create_database()
2 {
3     sqlite3* database;
4     char* sql = "DROP TABLE IF EXISTS intrebari;"
5                 "CREATE TABLE intrebari("
6                 "ID INT PRIMARY KEY NOT NULL, "
7                 "INTREBARE TEXT NOT NULL, "
8                 "R1 TEXT NOT NULL, "
9                 "R2 TEXT NOT NULL, "
10                "R3 TEXT NOT NULL, "
11                "R4 TEXT NOT NULL, "
12                "R_CORRECT CHAR(1) NOT NULL);"
13    "INSERT INTO intrebari VALUES(0, '1+1= ?', '1',
14    '2', '3', '4', '2');"
15    "INSERT INTO intrebari VALUES(1, 'What is a
16    correct syntax to output \"Hello World\" in C++?', 'print
17    (\"Hello World\");', 'cout << \"Hello World\";', 'Console.
18    WriteLine(\"Hello World\");', 'System.out.println(\"Hello
19    World\");', '2');"
20    "INSERT INTO intrebari VALUES(2, 'How do you
21    insert COMMENTS in C++ code?', '/* This is a comment', '#
22    This is a comment', '<!-- This is a comment', '// This is
23    a comment', '4');"
24    "INSERT INTO intrebari VALUES(3, 'How do you
25    create a variable with the floating number 5.2?', 'int x =
26    5.2;', 'x = 5.2;', 'double x = 5.2;', 'byte x = 5.2',
27    '3');"
28    "INSERT INTO intrebari VALUES(4, 'Which method
29    can be used to find the length of a string?', 'length()',
30    'getSize()', 'len()', 'getLength()', '1');"
31
32    int db = sqlite3_open("intrebari.db", &database);
33    if(db)
34        perror("[create_database()] Eroare la deschiderea bazei de
35        date.");
36    else printf("[create_database()] Baza de date a fost
37        deschisa cu succes.\n");
38    char* error;
39    db = sqlite3_exec(database, sql, NULL, 0, &error);
40    if(db != SQLITE_OK)
41    {
42        perror("[create_database()] Eroare la baza de date.\n");
43        sqlite3_free(error);
44    }
45    else printf("[create_database()] Baza de date a fost creata
46        cu succes.\n");
47    sqlite3_close(database);
48 }

```

Funcția prin care preluăm întrebarea de la runda K

```

1 int get_question(int k, char question[], char correct_answer
  [])
2 {
3     sqlite3* database;
4     sqlite3_stmt* statement;
5
6     int db = sqlite3_open("intrebari.db", &database);
7     if(db)
8         perror("[get_question()] Eroare la deschiderea bazei de
9         date.\n");
10    else printf("[get_question()] Baza de date a fost deschisa
11    cu succes.\n");
12
13    char *sql = "select * from intrebari where id = ?";
14    db = sqlite3_prepare_v2(database, sql, -1, &statement, NULL)
15    ;
16    if(db != SQLITE_OK)
17    {
18        perror("[get_question()] Eroare la sqlite3_prepare_v2.\n")
19        ;
20        return 0;
21    }
22
23    sqlite3_bind_int(statement, 1, k);
24    int step = sqlite3_step(statement);
25    if(step == SQLITE_ROW)
26    {
27        sprintf(question, "%s\n[1] %s\n[2] %s\n[3] %s\n[4] %s\n
28        \nRaspunsul tau: ",
29        sqlite3_column_text(statement, 1), sqlite3_column_text(
30        statement, 2), sqlite3_column_text(statement, 3),
31        sqlite3_column_text(statement, 4), sqlite3_column_text(
32        statement, 5));
33        sprintf(correct_answer, "%s", sqlite3_column_text(
34        statement, 6));
35    }
36    sqlite3_finalize(statement);
37    sqlite3_close(database);
38
39    return 1;
40 }

```

5 Concluzii

Idei de dezvoltare

Ca modificări ulterioare ale aplicației putem realiza următoarele:

- Crearea unui sistem de login care permite și logarea unui administrator care să modifice baza de date astfel încât să nu hard codăm întrebările și răspunsurile.
- Adăugarea unui clasament pentru cele mai multe runde câștigate/jucate/punctaj total.
- Implementarea unei interfețe grafice.

References

1. Computer Networks website, <https://profs.info.uaic.ro/~computernetworks/>.
2. Introduction To The SQLite C/C++ Interface, <https://www.sqlite.org/cintro.html>.
3. SQL using C/C++ and SQLite, <https://www.geeksforgeeks.org/sql-using-c-c-and-sqlite/>.
4. select(2)-Linux manual page, <https://man7.org/linux/man-pages/man2/select.2.html>.
5. Stack Overflow, <https://stackoverflow.com/questions/18563462/exit-a-for-loop-if-not-input-is-entered-after-some-time>.