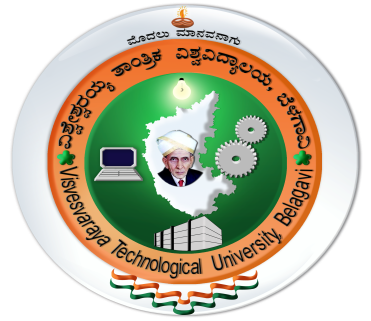
**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**Jnana Sangama, Belagavi - 590018**

****

**Assignment 1**

**Design & Analysis Of Algorithms (21CS42)**

**“Quiz Game Project”**

**Aadil S        : 4MT21CS002**

**Albin Reji    : 4MT21CS019**

**Goutam V : 4MT21CS055**

**Anirudh P K : 4MT21CS023**

**Name of the faculty**

**Ms. Sunitha .N.V**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

***(Accredited by NBA)***

**MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING**

*Accredited by NAAC with A+ Grade, An ISO 9001: 2015 Certified Institution*

(*A Unit of Rajalaxmi Education Trust®, Mangalore - 575001*)

                                              Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi

**BadagaMijar, Moodabidri-574225, Karnataka**

# 

Table of Contents

[Quiz Game Project 3](#_Toc144618903)

[Abstract 3](#_Toc144618904)

[Key Features: 3](#_Toc144618905)

[Introduction : 4](#_Toc144618906)

[Background: 4](#_Toc144618907)

[Objectives: 4](#_Toc144618908)

Code:………………..……………………………………………………………………………………………………………………………5

[Technologies Used: 15](#_Toc144618909)

[System Architecture: 16](#_Toc144618910)

[Front-End Design: 16](#_Toc144618911)

[Back-End 17](#_Toc144618912)

[Features and Functionality: 17](#_Toc144618913)

[Database: 18](#_Toc144618914)

[Testing :- 18](#_Toc144618915)

[Challenges Faced : 19](#_Toc144618916)

[Future Enhancements: 19](#_Toc144618917)

[Conclusion: 20](#_Toc144618918)

[References: 20](#_Toc144618919)

[Screenshots: 21](#_Toc144618920)

# 

# Quiz Game Project

# Abstract

The Quiz Game Application is a digital platform designed to provide an engaging and interactive experience for users to test their knowledge on various topics through quizzes. The primary purpose of this project is to offer an entertaining and educational tool that can be accessed via web or mobile devices. The scope of the project includes creating a user-friendly interface, developing a diverse set of quiz categories, and implementing features for user registration, quiz creation, and leaderboards.

## Key Features:

* **User Registration and Profiles**: Users can create accounts, customize their profiles, and track their quiz performance over time.
* **Quiz Categories**: The application will offer a wide range of quiz categories, including general knowledge, science, sports, entertainment, history, and more. Users can select their preferred categories to take quizzes.
* **Quiz Creation**: Registered users can create their quizzes, providing questions and answers in a user-friendly interface. They can also choose the category and difficulty level for their quizzes.
* **Quiz Playing**: Users can browse and select quizzes to play. The application will provide a timer, score tracking, and instant feedback on the correctness of each answer.
* **Gamification Elements**: To enhance user engagement, the application will incorporate gamification elements such as badges, achievements, and rewards for completing quizzes and reaching certain milestones.
* **Feedback and Support**: Users can provide feedback and report issues within the application, ensuring continuous improvement and a positive user experience.

The Quiz Game Application aims to cater to a broad audience interested in expanding their knowledge while enjoying a fun and competitive experience. It provides an opportunity for users to create and share quizzes, fostering a community of knowledge enthusiasts. The project's ultimate goal is to promote learning through an entertaining and interactive medium.

# Introduction :

In a world where knowledge is highly valued and where the thrill of competition never gets old, our Quiz Game Project brings the perfect blend of fun, learning, and rewards to your fingertips. This digital platform is designed to challenge your intellect, provide a chance to showcase your expertise, and, most importantly, offer the opportunity to win cash prizes with each correct answer. Welcome to the Quiz Game that not only tests your knowledge but also rewards your wisdom.

## Background:

The Quiz Game Project stems from the growing fascination with trivia, quizzes, and competitive learning experiences. In today's fast-paced world, people are constantly seeking ways to expand their knowledge in an engaging and entertaining manner. Traditional quiz games have provided an outlet for this, but our project takes it a step further. By offering cash prizes for each correct answer, we aim to not only educate but also incentivize learning.

In an age where information is readily available at our fingertips, it's essential to ensure that knowledge is not only accessible but also enticing. This project addresses that need by transforming the conventional quiz game into a captivating and rewarding experience. It caters to a diverse audience, from students looking to reinforce their learning to trivia enthusiasts seeking a thrilling challenge.

## Objectives:

The Quiz Game Project has several key objectives:

**Knowledge Promotion**: Our primary goal is to promote knowledge acquisition and retention. By offering a wide array of quiz categories, we aim to inspire users to explore new topics and deepen their understanding of subjects they are passionate about.

**Engagement:** We want to captivate users with an immersive and interactive quiz game experience. The project incorporates gamification elements, social sharing, and leaderboards to keep users engaged and motivated.

**Competition:** Fostering healthy competition is one of our objectives. Users can test their knowledge against others and strive to top the leaderboards, creating a sense of achievement and recognition.

**Rewarding Learning:** By awarding cash prizes for correct answers, we aim to motivate users to participate actively and invest in their learning journey. This financial incentive adds an exciting dimension to the quiz game.

**Accessibility:** The project aims to be accessible to a wide audience, ensuring that anyone with an internet connection can join the quiz game and potentially win prizes.

# Code:

#include<stdio.h>

#include<conio.h>

#include<ctype.h>

#include<stdlib.h>

#include<string.h>

#define MAX\_QUESTIONS 20

#define MAX\_USERS 100

#define MAX\_NAME\_LENGTH 20

#define MAX\_PASSWORD\_LENGTH 30

struct User {

    char username[MAX\_NAME\_LENGTH];

    char password[MAX\_PASSWORD\_LENGTH];

};

struct Question {

    char question\_text[500];

    char options[4][100];

    char correct\_option;

};

void read\_questions(FILE \*file, struct Question \*questions, int num\_questions) {

    int i,j;

    for ( i = 0; i < num\_questions; i++) {

        fscanf(file, "%499[^\n]\n", questions[i].question\_text);

        for ( j = 0; j < 4; j++) {

            fscanf(file, "%99[^\n]\n", questions[i].options[j]);

        }

        fscanf(file, "%c\n", &questions[i].correct\_option);

    }

}

void show\_record() {

    FILE \*f = fopen("score.txt", "r");

    if (f == NULL) {

        printf("No high score available.\n");

        return;

    }

    char name[MAX\_NAME\_LENGTH];

    float score;

    fscanf(f, "%s%f", name, &score);

    printf("\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

    printf("\n\n\t\t %s has secured the Highest Score %.2f", name, score);

    printf("\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

    fclose(f);

    getchar();

}

void reset\_score() {

    FILE \*f = fopen("score.txt", "w");

    if (f == NULL) {

        printf("Error opening score file.\n");

        return;

    }

    fprintf(f, "%s %.2f", "NoName", 0.0);

    fclose(f);

    printf("\n\n\t\tScore is reset to zero.\n");

}

void help()

    {system("cls");

    printf("\n\n                              HELP");

    printf("\n -------------------------------------------------------------------------");

    printf("\n ......................... C program Quiz Game...........");

    printf("\n >> There are two rounds in the game, WARMUP ROUND & CHALLANGE ROUND");

    printf("\n >> In warmup round you will be asked a total of 3 questions to test your general");

    printf("\n    knowledge. You will be eligible to play the game if you can give atleast 2");

    printf("\n    right answers otherwise you can't play the Game...........");

    printf("\n >> Your game starts with the CHALLANGE ROUND. In this round you will be asked");

    printf("\n    total 10 questions each right answer will be awarded $100,000.");

    printf("\n    By this way you can win upto ONE MILLION cash prize in USD...............");

    printf("\n >> You will be given 4 options and you have to press A, B ,C or D for the");

    printf("\n    right option");

    printf("\n >> You will be asked questions continuously if you keep giving the right answers.");

    printf("\n >> No negative marking for wrong answers");

    printf("\n\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*BEST OF LUCK\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

    printf("\n\n\t\*\*\*\*\*C PROGRAM QUIZ GAME is developed by Technical Team 5\*\*\*\*\*\*\*\*");}

void edit\_score(float score, char plnm[20]) {

    system("cls");

    float sc;

    char nm[20];

    FILE \*f;

    f = fopen("score.txt", "r");

    fscanf(f, "%s%f", nm, &sc);  // Corrected line

    if (score >= sc) {

        sc = score;

        fclose(f);

        f = fopen("score.txt", "w");

        fprintf(f, "%s\n%.2f", plnm, sc);

        fclose(f);

    }

}

void shuffle\_questions(struct Question arr[], int n) {

    int i,j;

    srand(time(NULL));

    for (i = n - 1; i > 0; i--) {

        j = rand() % (i + 1);

        struct Question temp = arr[i];

        arr[i] = arr[j];

        arr[j] = temp;

    }

}

void register\_user() {

    struct User users[MAX\_USERS];

    int num\_users = 0;

    int i;

    // Read existing user data from file

    FILE \*users\_file = fopen("users.txt", "r");

    if (users\_file) {

        while (fscanf(users\_file, "%s %s", users[num\_users].username, users[num\_users].password) != EOF) {

            num\_users++;

        }

        fclose(users\_file);

    }

    // Register a new user

    char new\_username[MAX\_NAME\_LENGTH];

    char new\_password[MAX\_PASSWORD\_LENGTH];

    printf("\nEnter a new username: ");

    scanf("%s", new\_username);

    // Check if the username already exists

    int username\_exists = 0;

    for (i = 0; i < num\_users; i++) {

        if (strcmp(new\_username, users[i].username) == 0) {

            username\_exists = 1;

            break;

        }

    }

    if (username\_exists) {

        printf("Username already exists. Registration failed.\n");

    }

    else {

        printf("Enter a password: ");

        scanf("%s", new\_password);

        // Add the new user to the array

        strcpy(users[num\_users].username, new\_username);

        strcpy(users[num\_users].password, new\_password);

        num\_users++;

        // Write the updated user data back to the file

        users\_file = fopen("users.txt", "w");

        for (i = 0; i < num\_users; i++) {

            fprintf(users\_file, "%s %s\n", users[i].username, users[i].password);

        }

        fclose(users\_file);

        printf("User registered successfully!\n");

        getch();

    }

}

char player\_name[MAX\_NAME\_LENGTH];

int login() {

    struct User users[MAX\_USERS];

    int num\_users = 0;

    int i;

    // Read user data from file

    FILE \*users\_file = fopen("users.txt", "r");

    if (users\_file) {

        while (fscanf(users\_file, "%s %s", users[num\_users].username, users[num\_users].password) != EOF) {

            num\_users++;

        }

        fclose(users\_file);

    }

    // Prompt for login credentials

    char entered\_username[MAX\_NAME\_LENGTH];

    char entered\_password[MAX\_PASSWORD\_LENGTH];

    printf("LOGIN");

    printf("\nEnter your username: ");

    scanf("%s", entered\_username);

    printf("Enter your password: ");

    scanf("%s", entered\_password);

    // Check if the entered credentials match any user

    for (i = 0; i < num\_users; i++) {

        if (strcmp(entered\_username, users[i].username) == 0 && strcmp(entered\_password, users[i].password) == 0) {

            printf("\n\n  LOGIN SUCCESSFUL");

            strcpy(player\_name, entered\_username);

            return 1; // Login successful

        }

    }

    return 0; // Login failed

}

int main()

     {

    int count = 0;

    int i, n,ch;

    float score;

    char choice;

    struct Question questions[15];

    struct Question football[MAX\_QUESTIONS];

    struct Question gk[MAX\_QUESTIONS];

    struct Question sports[MAX\_QUESTIONS];

    struct Question TV[MAX\_QUESTIONS];

    struct Question music[MAX\_QUESTIONS];

    struct Question geography[MAX\_QUESTIONS];

    struct Question cricket[MAX\_QUESTIONS];

    FILE \*questions\_file= fopen("questions.txt","r");

    FILE \*football\_file = fopen("football.txt", "r");

    FILE \*gk\_file = fopen("gk.txt", "r");

    FILE \*sports\_file = fopen("sports.txt", "r");

    FILE \*TV\_file=fopen("TV.txt","r");

    FILE \*music\_file=fopen("music.txt","r");

    FILE \*geography\_file=fopen("geography.txt","r");

    FILE \*cricket\_file=fopen("cricket.txt","r");

    if (football\_file && gk\_file && sports\_file && TV\_file && music\_file && geography\_file && cricket\_file) {

        // Read questions from files into arrays

        read\_questions(questions\_file, questions, 15);

        read\_questions(football\_file, football, MAX\_QUESTIONS);

        read\_questions(gk\_file, gk, MAX\_QUESTIONS);

        read\_questions(sports\_file, sports, MAX\_QUESTIONS);

        read\_questions(TV\_file, TV, MAX\_QUESTIONS);

        read\_questions(music\_file, music, MAX\_QUESTIONS);

        read\_questions(geography\_file, geography, MAX\_QUESTIONS);

        read\_questions(cricket\_file, cricket, MAX\_QUESTIONS);

        // Close the question files

        fclose(questions\_file);

        fclose(football\_file);

        fclose(gk\_file);

        fclose(sports\_file);

        fclose(TV\_file);

        fclose(music\_file);

        fclose(geography\_file);

        fclose(cricket\_file);

    } else {

        printf("Error opening question files.\n");

        return 1;

    }

    mainhome:

    printf("\n\n\t\t\tC PROGRAM QUIZ GAME");

    printf("\n\n\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

    printf("\n\t\t\t   WELCOME ");

    printf("\n\t\t\t      to ");

    printf("\n\t\t\t   THE GAME ");

    printf("\n\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

    printf("\n\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

    printf("\n\n\t\t > Press S to start the game");

    printf("\n\t\t > Press V to view the highest score");

    printf("\n\t\t > Press R to reset score");

    printf("\n\t\t > Press H for help");

    printf("\n\t\t > Press N to register");

    printf("\n\t\t > Press Q to quit");

    printf("\n\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n\t\t ");

    choice = toupper(getch());

    if (choice == 'V') {

        show\_record();

        goto mainhome;

    }

    else if (choice == 'H') {

        help();

        getch();

        goto mainhome;

    }

    else if (choice == 'R') {

        reset\_score();

        printf("\n\n\t\tScore is reset to zero.");

        getch();

        goto mainhome;

    }

    else if (choice == 'Q')

        return 0;

    else if (choice == 'N') {

        printf("REGISTRATION");

        register\_user();

        goto mainhome;

    }

    else if (choice == 'S') {

        if (login()) {

            goto start;

        } else {

            printf("\nLogin failed. Press any key to return to the main menu.");

            getch();

            goto mainhome;

        }

        start:

            system("cls");

            printf("\n ------------------  Welcome %s to C Program Quiz Game --------------------------", player\_name);

            printf("\n\n Here are some tips you might wanna know before playing:");

            printf("\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

            printf("\n >> There are two rounds in this Quiz Game, WARMUP ROUND & CHALLANGE ROUND");

            printf("\n >> In warmup round, you will be asked a total of 3 questions to test your general knowledge.");

            printf("\n    You need to answer at least 2 questions correctly to qualify for the next round.");

            printf("\n >> Your game starts with the CHALLANGE ROUND. In this round, you will be asked a total of 10 questions.");

            printf("\n    Each correct answer will be awarded $100,000.");

            printf("\n    By this way you can win up to ONE MILLION cash prize in USD.");

            printf("\n >> You will be given 4 options and you have to press A, B, C or D for the right option.");

            printf("\n >> You will be asked questions continuously as long as you keep giving the right answers.");

            printf("\n >> There is no negative marking for wrong answers.");

            printf("\n\n\t\t!!!!!!!!!!!!! BEST OF LUCK !!!!!!!!!!!!!");

            printf("\n\n\n Press Y if you are ready to start the game!");

            printf("\n Press any other key to return to the main menu!");

            if (toupper(getch()) == 'Y') {

                goto home;

            } else {

                goto mainhome;

            }

        home:

            system("cls");

            score = 0;

            printf("\n ------------------  WARMUP ROUND --------------------------");

            shuffle\_questions(questions, sizeof(questions) / sizeof(questions[0]));

            for (i = 0; i < 3; i++) {

                printf("\n\n%s\n", questions[i].question\_text);

                for (n = 0; n < 4; n++) {

                    printf("\n\t%s", questions[i].options[n]);

                }

                printf("\n\n\n");

                choice = toupper(getch());

                if (choice == questions[i].correct\_option) {

                    printf("\n\nCorrect!");

                    count += 1;

                    getch();

                } else {

                    printf("\n\nWrong!");

                    printf("\n\nThe correct option is %c", questions[i].correct\_option);

                    getch();

                }

            }

            if(count>=2)

            {

                goto test;}

            else

            {

                system("cls");

                printf("\n\nSORRY YOU ARE NOT ELIGIBLE TO PLAY THIS GAME, BETTER LUCK NEXT TIME");

                getch();

                goto mainhome;

            }

        test:

            system("cls");

        printf("\n\n\t\*\*\* CONGRATULATION %s you are eligible to play the Game \*\*\*",player\_name);

        printf("\n\n\n\n\t!Press any key to Start the Game!");

        if(toupper(getch())=='p')

        {goto game;}

        game:

            system("cls");

            printf("\n ------------------  CHALLENGE ROUND --------------------------");

            printf("\n\t\tEnter your choice of Interest\n\n");

            printf("\n\t\t1. Football\n\n");

            printf("\n\t\t2. General Knowledge\n\n");

            printf("\n\t\t3. Sports\n\n");

            printf("\n\t\t4. TV Shows and Movies\n\n");

            printf("\n\t\t5. Music or Song\n\n");

            printf("\n\t\t6. Geography\n\n");

            printf("\n\t\t7. Cricket\n\n");

            scanf("%d",&ch);

            if(ch==1){

                    system("cls");

                     score = 0;

                    shuffle\_questions(football, sizeof(football) / sizeof(football[0]));

                    for (i = 0; i < 10; i++) {

                         printf("\n\n%s\n", football[i].question\_text);

                         for (n = 0; n < 4; n++) {

                             printf("\n\t%s", football[i].options[n]);

                        }

                        printf("\n\n\n");

                        choice = toupper(getch());

                        if (choice == football[i].correct\_option) {

                             printf("\n\nCorrect!");

                             score += 100000;

                             getch();

                        }

                        else {

                            printf("\n\nWrong!");

                            printf("\n\nThe correct option is %c", football[i].correct\_option);

                            getch();

                         }

                    }

                }

            else if(ch==2){

                    system("cls");

                    score = 0;

                    shuffle\_questions(gk, sizeof(gk) / sizeof(gk[0]));

                    for (i = 0; i < 10; i++) {

                         printf("\n\n%s\n", gk[i].question\_text);

                         for (n = 0; n < 4; n++) {

                             printf("\n\t%s", gk[i].options[n]);

                        }

                        printf("\n\n\n");

                        choice = toupper(getch());

                        if (choice == gk[i].correct\_option) {

                             printf("\n\nCorrect!");

                             score += 100000;

                             getch();

                        }

                        else {

                            printf("\n\nWrong!");

                            printf("\n\nThe correct option is %c", gk[i].correct\_option);

                            getch();

                        }

                }

            }

            else if(ch==3){

                    system("cls");

                     score = 0;

                    shuffle\_questions(sports, sizeof(sports) / sizeof(sports[0]));

                    for (i = 0; i < 10; i++) {

                         printf("\n\n%s\n", sports[i].question\_text);

                         for (n = 0; n < 4; n++) {

                             printf("\n\t%s", sports[i].options[n]);

                        }

                        printf("\n\n\n");

                        choice = toupper(getch());

                        if (choice == sports[i].correct\_option) {

                             printf("\n\nCorrect!");

                             score += 100000;

                             getch();

                        }

                        else {

                            printf("\n\nWrong!");

                            printf("\n\nThe correct option is %c", sports[i].correct\_option);

                            getch();

                        }

                    }

                }

            else if(ch==4){

                    system("cls");

                     score = 0;

                    shuffle\_questions(TV, sizeof(TV) / sizeof(TV[0]));

                    for (i = 0; i < 10; i++) {

                         printf("\n\n%s\n", TV[i].question\_text);

                         for (n = 0; n < 4; n++) {

                             printf("\n\t%s", TV[i].options[n]);

                        }

                        printf("\n\n\n");

                        choice = toupper(getch());

                        if (choice == TV[i].correct\_option) {

                             printf("\n\nCorrect!");

                             score += 100000;

                             getch();

                        }

                        else {

                            printf("\n\nWrong!");

                            printf("\n\nThe correct option is %c", TV[i].correct\_option);

                            getch();

                        }

                    }

                }

            else if(ch==5){

                system("cls");

                 score = 0;

                shuffle\_questions(music, sizeof(music) / sizeof(music[0]));

                for (i = 0; i < 10; i++) {

                    printf("\n\n%s\n", music[i].question\_text);

                    for (n = 0; n < 4; n++) {

                         printf("\n\t%s", music[i].options[n]);

                    }

                    printf("\n\n\n");

                    choice = toupper(getch());

                    if (choice == music[i].correct\_option) {

                        printf("\n\nCorrect!");

                        score += 100000;

                        getch();

                    }

                    else {

                        printf("\n\nWrong!");

                        printf("\n\nThe correct option is %c", music[i].correct\_option);

                        getch();

                    }

                }

            }

            else if(ch==6){

                    system("cls");

                     score = 0;

                    shuffle\_questions(geography, sizeof(geography) / sizeof(geography[0]));

                    for (i = 0; i < 10; i++) {

                         printf("\n\n%s\n", geography[i].question\_text);

                         for (n = 0; n < 4; n++) {

                             printf("\n\t%s", geography[i].options[n]);

                        }

                        printf("\n\n\n");

                        choice = toupper(getch());

                        if (choice == geography[i].correct\_option) {

                             printf("\n\nCorrect!");

                             score += 100000;

                             getch();

                        }

                        else {

                            printf("\n\nWrong!");

                            printf("\n\nThe correct option is %c", geography[i].correct\_option);

                            getch();

                        }

                    }

                }

            else if(ch==7){

                    system("cls");

                     score = 0;

                    shuffle\_questions(cricket, sizeof(cricket) / sizeof(cricket[0]));

                    for (i = 0; i < 10; i++) {

                         printf("\n\n%s\n", cricket[i].question\_text);

                         for (n = 0; n < 4; n++) {

                             printf("\n\t%s", cricket[i].options[n]);

                        }

                        printf("\n\n\n");

                        choice = toupper(getch());

                        if (choice == cricket[i].correct\_option) {

                             printf("\n\nCorrect!");

                             score += 100000;

                             getch();

                        }

                        else {

                            printf("\n\nWrong!");

                            printf("\n\nThe correct option is %c", cricket[i].correct\_option);

                            getch();

                        }

                    }

                }

            else{

                printf("\n\t\tINVALID OPTION\n\t\tEXITING!!!");

                goto mainhome;

            }

            goto score;

        score:

        system("cls");

        if(score>10.00 && score<1000000.00)

        {

        printf("\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONGRATULATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

        printf("\n\t You won $%.2f",score);goto go;}

        else if(score==1000000.00)

        {

            printf("\n\n\n \t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONGRATULATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

            printf("\n\t\t\t\t YOU ARE A MILLIONAIRE!!!!!!!!!");

            printf("\n\t\t You won $%.2f",score);

            printf("\t\t Thank You!!");

        }

        else

        {

            printf("\n\n\t\*\*\*\*\*\*\*\* SORRY YOU DIDN'T WIN ANY CASH \*\*\*\*\*\*\*\*");

            printf("\n\t\t Thanks for your participation");

            printf("\n\t\t TRY AGAIN");goto go;}

        go:

        puts("\n\n Press Y if you want to play next game");

        puts(" Press any key if you want to go main menu");

        if (toupper(getch())=='Y')

            goto home;

        else

        {

            edit\_score(score,player\_name);

            goto mainhome;

        }

    }

    else{

        printf("Invalid choice");

        goto mainhome;

    }

}

# Technologies Used:

**C Standard Library**: This is the foundation for C programming and includes functions for input/output, string manipulation, and basic data structures.

**Curses Library**: If you want to create a text-based graphical interface for your quiz game within the terminal, the Curses library can be used to create interactive menus, windows, and text formatting.

**SQLite**: For storing and managing quiz questions, answers, and user data, you can use SQLite as a lightweight, embedded relational database management system (RDBMS).

**JSON Parsing Library**: To handle JSON data for storing and retrieving quiz questions and answers, you may use a lightweight JSON parsing library like cJSON.

**Random Number Generation**: To randomize the selection of quiz questions or to shuffle answer choices, you can use the C standard library's rand() function or a more sophisticated random number generator library like PCG.

**User Input Handling**: Utilize functions from the standard library to read and validate user input for answering questions and participating in the quiz.

**File I/O**: To read and write quiz data to files, such as questions, answers, and user progress, you can utilize C's file I/O functions like fopen(), fread(), and fwrite().

**Multi-threading (Optional)**: If you want to implement time limits for answering questions or parallel processing for improved performance, you can explore C's multi-threading capabilities.

**Encryption (Optional)**: If you want to enhance security, consider using C libraries for encryption, such as OpenSSL, for storing sensitive user data.

**Dynamic Memory Allocation**: C provides memory allocation functions like malloc() and free() for managing memory efficiently, which can be useful for dynamically allocating memory for user data structures.

**Error Handling**: Implement robust error handling mechanisms using C's errno and perror() functions to gracefully handle unexpected situations.

**Text Formatting and Output**: Utilize C's printf() and string manipulation functions for formatting and presenting quiz questions, answer choices, and feedback to the user.

**Data Structures**: You may need to implement custom data structures, such as linked lists or arrays, to manage user scores, leaderboards, and other game-related data.

While C may not be the most common choice for developing interactive quiz games due to its low-level nature, it can still be used effectively for console-based games. Remember that building a game in C might require more effort and low-level programming compared to higher-level languages, but it can be a rewarding learning experience for programmers.

## System Architecture:

The design and implementation of a Quiz Game Project in C language involves structuring the front-end, defining features, and implementing functionality to create an interactive and engaging quiz game. Here's a high-level overview of the design and implementation, focusing on Front-End Design, Features, and Functionality:

## Front-End Design:

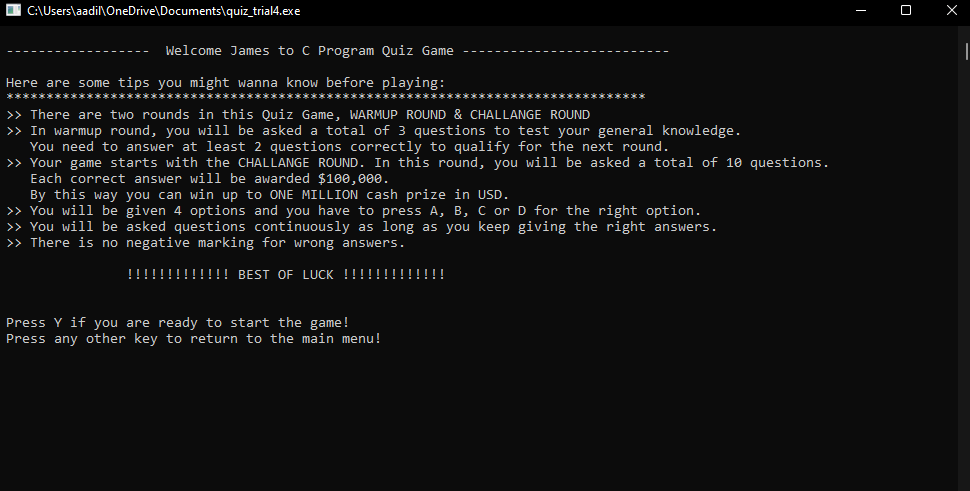
Console-Based User Interface (UI):

Design a text-based interface that provides clear instructions, displays questions, and accepts user responses.

Use ASCII art or text formatting to make the interface visually appealing.

Create a menu system for navigation, including options for starting a quiz, viewing user profiles, and exiting the game.

Implement a timer display if you want to add time constraints to questions.

****

Back-End**:**

**Game Logic:** This component controls the flow of the game, including presenting questions, calculating scores, and determining when the game ends.

**Randomization**: Use a random number generator to shuffle question order and randomize answer choices for each question.

**User Management**: If you implement user profiles, the back-end handles user registration, authentication, and user progress tracking.

# Features and Functionality:

Quiz Questions :- Create a question database or file format to store quiz questions, answer choices, and correct answers.

Randomization :- Randomly select questions from the database to ensure a different quiz experience each time.

User Interaction :- Accept user input for answering questions, selecting menu options, and navigating through the game.

Validate user input to prevent errors and ensure the game operates smoothly

Quiz Gameplay :- Display questions and answer choices in an organized manner.

Include a timer for each question to add excitement and challenge.

Allow users to select answers and move to the next question.

Feedback and Progress :- Provide immediate feedback on the correctness of answers.

Show the user's current score and progress throughout the game.

Offer hints or lifelines if desired.

Leaderboard : - Create a leaderboard that displays the top scorers.

## Database:

Data Storage : Set up a database to store quiz questions, user profiles, scores, leaderboard data, and payment records.

Ensure data integrity and implement backups.

# Testing :-

**1. User Registration and Authentication:**

How it works: Users can create accounts with unique usernames and passwords. They must log in before participating in quizzes.

Importance: Registration and authentication ensure user identity, allowing for progress tracking and secure participation.

**2. Quiz Categories and Selection:**

How it works: Users can choose from various quiz categories (e.g., general knowledge, science, sports) before starting a quiz.

Importance: This feature provides user choice and customization, making the game more appealing by catering to diverse interests.

**3. Quiz Gameplay:**

How it works: Questions are presented one by one, along with multiple-choice answer options. Users select their answers, and the system provides immediate feedback.

Importance: The core of the game, this feature engages users by challenging their knowledge and providing an interactive experience.

# Challenges Faced :

Developing a quiz game project, especially one that awards cash prizes for correct answers, can be a rewarding but challenging endeavor. Here are some common challenges that you might encounter during the development process and potential ways to overcome them:

* **Question Database Creation:**

Challenge: Building a substantial and diverse question database can be time-consuming and require extensive research.

Solution: Start with a small set of questions and gradually expand the database over time. You can also crowdsource question creation or use existing online resources with proper attribution.

* **Answer Validation:**

Challenge: Ensuring accurate validation of user answers can be complex, as answers might have multiple correct forms or spellings.

Solution: Use string comparison techniques that account for variations in capitalization, spacing, and common synonyms. Implement a robust algorithm to evaluate answers against predefined correct answers.

* **User Authentication and Security:**

Challenge: Managing user accounts, especially when dealing with cash prizes, requires robust authentication and security measures to prevent fraud.

Solution: Implement strong user authentication and authorization processes. Store sensitive user data securely, and regularly update security protocols to protect against potential threats

* **Technical Issues and Bugs:**

Challenge: Like any software project, quiz games may encounter technical issues and bugs that affect the user experience.

Solution: Implement thorough testing, including unit testing, integration testing, and user testing, to identify and address bugs promptly. Maintain a process for user-reported bug tracking and resolution.

# Future Enhancements:

* **Multiple Difficulty Levels:**

Implement different difficulty levels (easy, medium, hard) for questions. Assign higher prize amounts for harder questions**.**

* **Category Selection:**

Allow users to choose quiz categories (e.g., geography, history, science) before starting the game.

* **Leaderboard:**

Create a leaderboard to display top scores and encourage competition among users.

* **Multiplayer Mode:**

Add a multiplayer mode where users can compete against each other in real-time...

# Conclusion:

For developers and creators, working on a quiz game project can be personally satisfying, as it combines elements of creativity, programming, and game design.

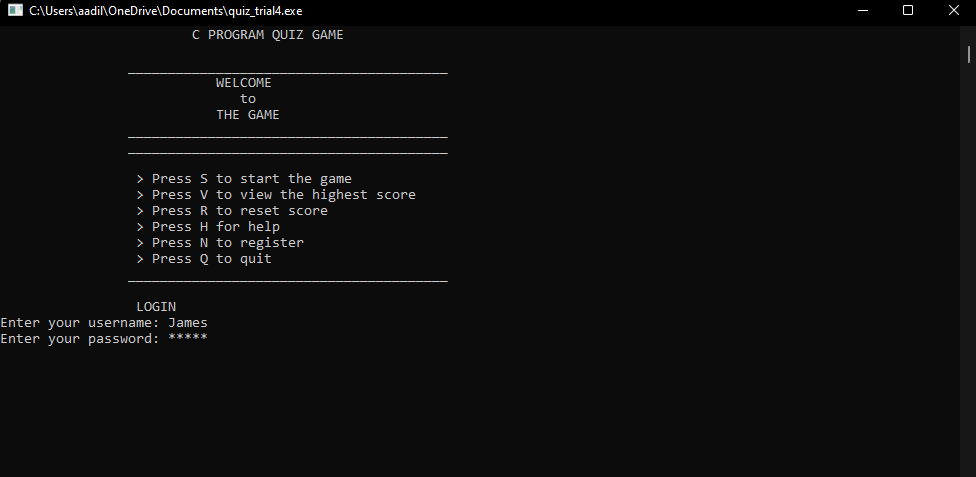
n summary, a quiz game project has both educational and entertainment value, fostering learning while providing users with an enjoyable experience. Its potential to engage users, improve cognitive skills, and entertain makes it a significant project that can cater to a diverse audience. Furthermore, it offers opportunities for personal development and potential revenue generation for developers.

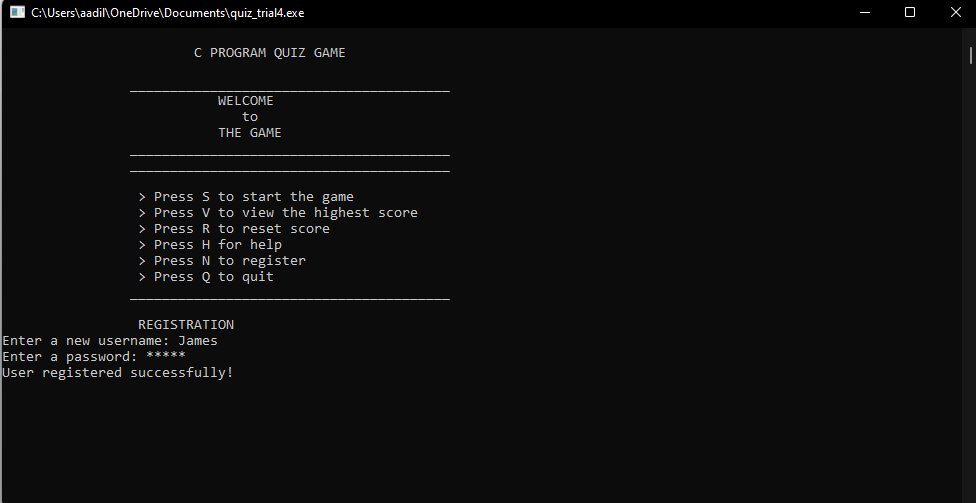
# References:

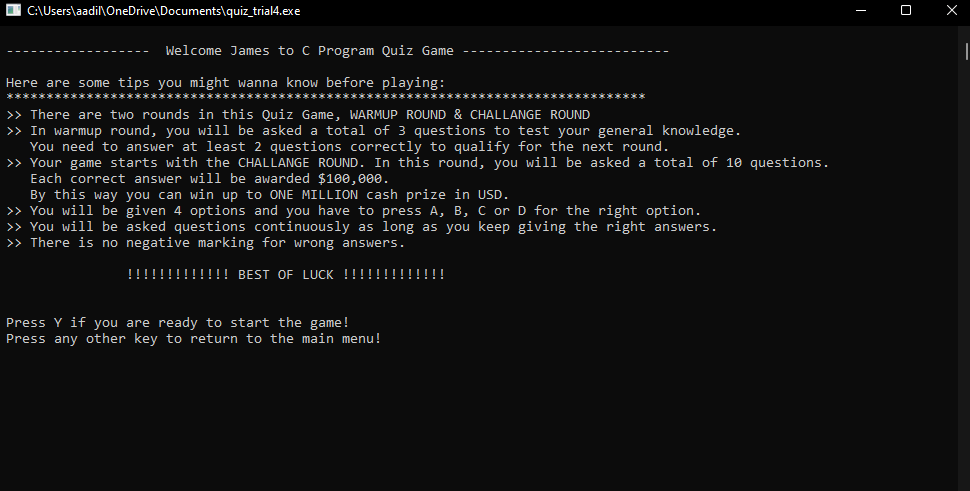
<https://itsourcecode.com/free-projects/c-projects/quiz-game-in-c-with-source-code/>

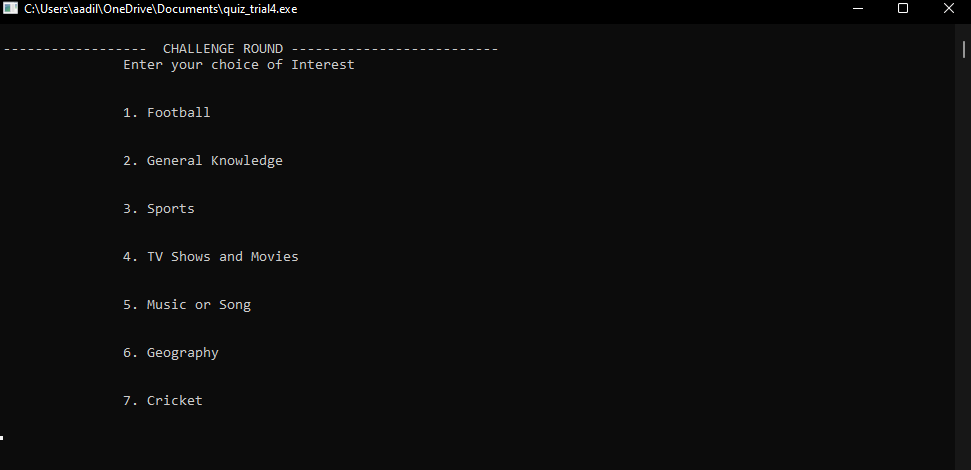
<https://github.com/Deepak06/Quiz-Game-Mini-Project-in-C/blob/master/quiz.c>

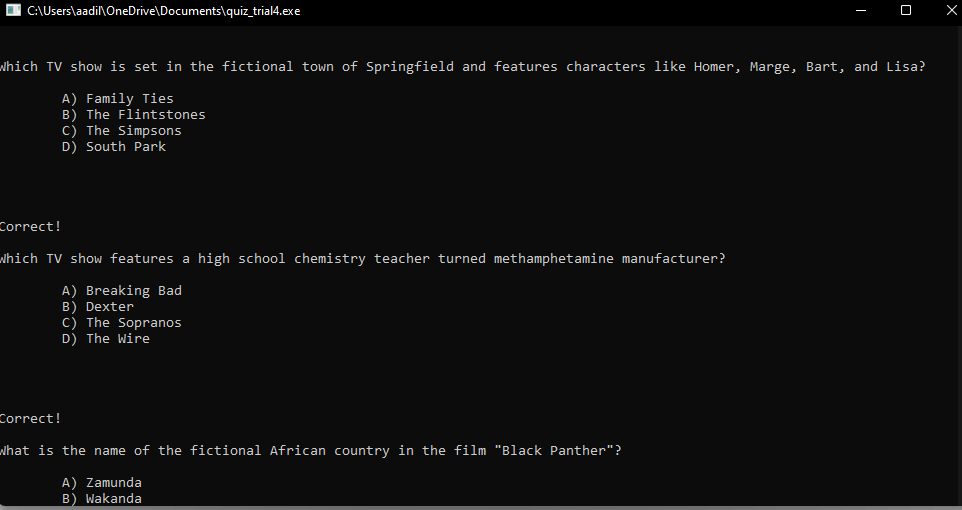
# Screenshots:

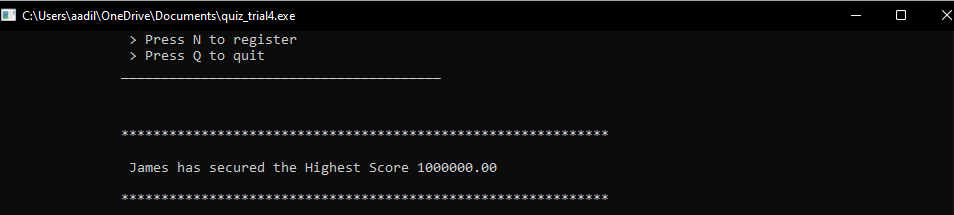
****

****

****

****



****