DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS

**PANJAB UNIVERSITY CHANDIGARH**



**PROJECT**

**REPORT**

**ON**

**QUIZ GAME**

(Data and file Structure using C)

2021-2022

**SUBMITTED TO: SUBMITTED BY:**

**MS. PROF. M SYAMALA DEVI RINKI**

**MCA –1(Morning) Roll No - 33**

**CERTIFICATE**

This is to certify that **Rinki (Roll no. 33), a** student of MCA-1 (Morning) hay completed his project **“QUIZ GAME”** under my guidance positively. The project progress has been continuously reported and has been in my knowledge consistently. The project is the original work done by the mentioned student only.

**Date: Project Guide:**

MS. PROF. M Syamala Devi

**ACKNOWLEDGEMENT**

We thank the almighty for giving us the courage & perseverance in completing the project. This project itself is an acknowledgment of all those who have given us their felt Cooperation in making it a grand success.

We are highly thankful to our (HOD) for providing the necessary instruction and for providing a great opportunity to develop this minor project. Apart from the effort, the success of any project depends largely on the encouragement & guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would like to show our heartiest appreciation to our project supervisor (Ms. Prof.M Syamala Devi) for extending their sincere & heartfelt guidance throughout this project work. We feel motivated & encouraged every time we attended your meeting. Without their encouragement & guidance, this project would not have been materialized. The guidance & support received from all the members who contributed and who are contributing to this

project were vital for the success of this effective project. We are grateful for their constant support and help.

**ABSTRACT**

The sole intention behind the consideration of this project is to check student IQ levels and tell him in which part he works and increase his general knowledge. The quiz game is designed to increase the learning of players by playing a multiple choice quiz contest on various topics which is interactive, user-friendly, and fun to play.

This project is to generate and manage a simple database for the question in which each level has 30 questions in its database and it will randomly take any 10 questions from it. This game has three-level. The first level is made of 10 questions which are randomly picked from the file and their level is very if the player answers any 7 questions out of 10 it will reach to next level other it. Furthermore, the second level is also of 10 questions which are randomly picked from the file and their level is very if the player answers any 7 questions out of 10 it will reach to next level other it. Now in the third level, 10 questions are randomly picked from the file and their level is very if the player answers any 7 questions out of 10. If it completed level three it will print the beautiful message on the screen and the game is over.

We have furthermore options in the main menu like view the highest score, reset the score, view the all student information who has played the game and the last option is to quit the game.

Table of Contents

[PROJECT TITLE 5](#_bookmark0)

[INTRODUCTION AND PROJECT DETAILS (OBJECTIVES) 7](#_bookmark1)

[METHODOLOGY FOLLOWED 9](#_bookmark2)

[Feasibility study 9](#_bookmark3)

[Design phase and DFD 10](#_bookmark4)

[Hardware and Software specification: 13](#_bookmark5)

[IMPLEMENTATION 14](#_bookmark6)

TESTING WITH VALIDATIONS 60

SIGNIFICANCE OF THE PROJECT 61

To society 61

To IT community 61

To Education and Research 62

CONCLUSION AND FUTURE SCOPE 63

BIBLIOGRAPHY/ REFERENCES 64

# PROJECT TITLE

# (QUIZ GAME)

To begin with, let us first understand the importance of the project title. It is dictum technically that the more the users of the application, the more successful the application is. **“Project Title”** - the most eye-catching aspect of a project, etc. The project title should be attractive enough will bring user’s/customer’s attention. As a result, they will go through the description of the Project at least once. So, while giving the project a title, there are a few things that should be kept in mind which are as follows:-

* The title of the project should be small and concise.
* The title name should depict the project’s purpose.
* The name should be attractive enough to steal the limelight over other similar projects.

****So keeping these things in mind, I have named my project **“QUIZ GAME”**. Well, the aim of the project will be discussed in the next section, after which this title will make even more sense. The output screen mentioning the title of the project is:

# AIM AND PROJECT DETAILS (OBJECTIVES)

## INTRODUCTION:

This 'QUIZ GAME' Project is designed for a question in which users can generate and manage a simple database for questions. The question number is automatically generated by the software and is stored in a text file by the name 'QUIZ. This database is used as a MASTER file to be used as a look-up table for information like Quiz Option, SubMenu, like play game, view the highest score, reset file, quit the game. There is a sub-menu application that contains various topics like science, mathematics, computers, etc, user can play a quiz on any of these topics and he gets a 4 for every correct answer and zeroes for every wrong answer and in the end, his total score is a is calculated.

## PROJECT DETAILS AND OBJECTIVES:

Now let us start with the detailed description (detailing) of the project. My project “Quiz game” has a lot of key features which eventually contribute to enhancing the popularity of the project. Some of the key features are discussed as follows:

* + The C graphics have been used appropriately to make the project user-friendly and attractive.
  + All the validations have been implemented to ensure the correctness of the data supplied.
  + As far as technicalities are concerned, the code has been properly commented on so that the code is readable.
  + The project has been developed by taking care of the time complexity of the project so that the results are faster.
  + File manipulations have been used to store the data permanently and manipulate that data later on.

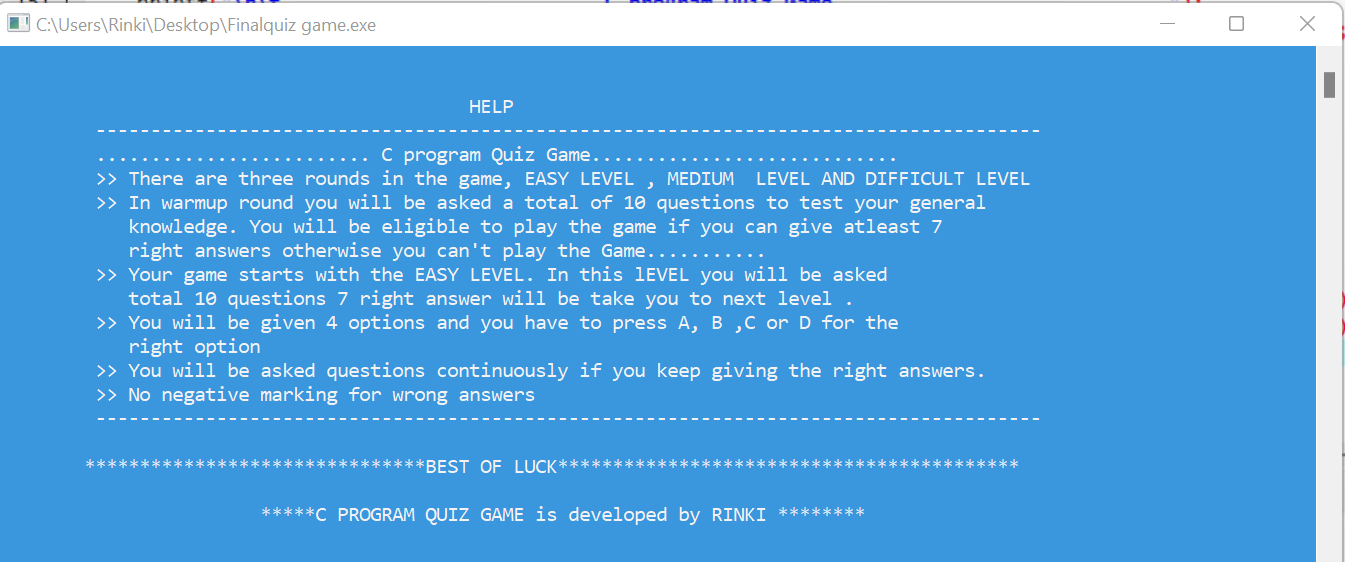
Along with these features, there are some **objectives** of the projective. These objectives are a must to achieve the proper and efficient functioning of the project.

All these **objectives** have been achieved in this project. Some of these **objectives**

are as follows:

1. The main objective of the Quiz application is to overcome all the drawbacks of the old Quiz Game.

2. The application has been designed to keep user interaction and friendliness the top priority. According to the application, a user can play an interactive quiz without the need of having a book and searching for various topics. He simply needs to open the application and test and enhance his abilities. The output screen (prepared using graphics), mentioning the functionalities achieved in the project by keeping the objectives in mind, is:



# METHODOLOGY FOLLOWED

Whenever a project needs to be started, a proper roadmap is to be followed. It is the best way to follow the step-by-step procedure to achieve any goal. The procedure which is followed to develop any software is called **SDLC (System Development Life Cycle)**. This SDLC includes steps that are to be followed sequentially. In this project, the SDLC steps have been followed. The important steps and the way they have been followed in this project is as follows:

## Feasibility study:

After the identification of requirements, an initial investigation of a proprophet determines whether an alternative system is feasible or not. A proposal summarizing the thinking of the analyst is presented to the user for review. When approved, the proposal initiates a feasibility study that describes and evaluates candidate systems and provides for the selection of the best system that meets system performance require feasibility study the involves the following considerations:

1. **Operational Feasibility** is the measure of how well a proposed system solves the problem and takes advantage of the opportunities identified during scope definition.
2. **Economic Feasibility** determines whether the application to be developed will be within the budget and would also provide Return On Investment (ROI) or not.
3. **Technical Feasibility** is an evaluation of hardware and software and how it meets the need of the proposed system. It ensures that the currently available resources would meet the basic requirement of the application.
4. **Time Feasibility** is determining whether the project would be developed within the given deadline or not. This would ensure the main aspect of the project i.e. customer satisfaction.

All these feasibility considerations were taken care of while developing this project. Before the actual implementation of the project, these four feasibility considerations were considered in “Quiz Game” using the following procedures.

* The operational feasibility was checked by using Research and Development also called **R&D**. All the important inbuilt functions and required functionalities were examined by the desired output and the offered benefits.
* The economic feasibility was not much of a concern as far as this project is concerned. There was no such big requirement of money in terms of investment and more projects would yield propitious outputs.
* The project was considered technically feasible, as the hardware and software required for this project are very basic. The hardware of a normal PC or laptop would be enough to support this project. The software used to run the project is also open-source makes it technically feasible.
* Well, the big challenge of this project was the time limit. So time was a big concern. But when it was analyzed, the proper planning and working hours were prepared, which eventually resulted in the time feasibility of the project.

## Design phase and DFD:

Based on user needs and detailed analysis of the system software must be designed. It is the most crucial phase in the development of a system. In this phase, the process continues to move from what question to **how** question. The logical design is turned into a physical design. The major tools and techniques used for describing the system design of the system are Flowcharts, Data Flow Diagram (DFD), Data Dictionary, Structured English, Decision Table, and Decision Tree.

Wireframes are the main part of the design phase. Because before the actual implementation of the project, the blueprints are to be prepared so that customer satisfaction can be achieved in accord with their requirement. So the number of screens and the task of each screen were wire-framed to ease the task of the actual implementation.

Now come to the tools used for the design phase. As discussed earlier, many tools are used to design a project. I have used the most common and probably the most beneficial tool called **DFD.** Data flow diagrams are used to graphically represent the flow of data in an information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and report generation.

To represent different entities, operations, and objects there are various **symbols** used. These symbols are standard and thus standardize the Data Flow Diagram. These **DFD symbols** make it easy to distinguish between different parameters of the object. So, understanding the importance of symbols, let us start with the various symbols used to design Data Flow Diagram (DFD).

|  |  |
| --- | --- |
| or | PROCESS |
|  | DATA STORE |
|  | EXTERNAL ENTITIES |
|  | DATA FLOW |

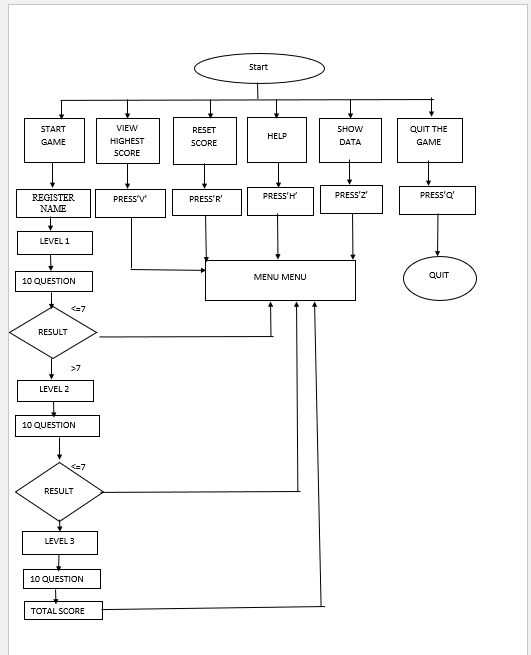
Data flow diagrams are also categorized by **levels**. Starting with the most basic, level 0, DFDs get increasingly complex as the level increases. As you build your own data flow diagram, you will need to decide which level of details your diagram will be following.

**Level 0 DFDs**, also known as context diagrams, are the most basic data flow diagrams. They provide a broad view that is easily digestible but offers little detail. Level 0 data flow diagrams show a single process node and its connections to external entities.

**Level 1 DFDs** are still a general overview, but they go into more detail than a context diagram. In a level 1 data flow diagram, the single process node from the context diagram is broken down into sub processes. As these processes are added, the diagram will need additional data flows and data stores to link them together.

**Level 2+ DFDs** simply break processes down into more detailed sub processes. In theory, DFDs could go beyond level 3, but they rarely do. Level 3 data flow diagrams are detailed enough that it doesn’t usually make sense to break them down further.

The DFD prepared during design phase of this project is as follows:



## Hardware and Software specification:

Computer hardware is the collection of physical elements that constitutes a computer system. Computer hardware refers to a physical parts and components of the computer such as monitor, mouse, keyboard, computer data storage, Hard Drive Disk (HDD), system unit (graphic cards, sound cards, memory, motherboard, and chips), etc. that can be touched.

The minimum **Hardware** requirements for our program are as follows:

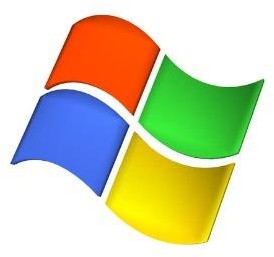
* PROCESSOR: Pentium IV processor or greater
* RAM: 128 MegaByte (MB) or greater.
* HARD DISK: 1.2 Giga Byte (GB) or greater.
* Keyboard and mouse.
* MONITOR: color (For best results)



Computer software, or just software is any set of machine readable instructions that directs a computer’s processor to perform specific operations. Computer hardware and software require each other and neither can be realistically used without the other.

The minimum **Software** requirements for our program are as follows:

* + OPERATING SYSTEM: Windows 2000 and later.
  + IDE USED: CodeBlocks or Dev C++.



# IMPLEMENTATION (INCLUDING CODE)

Now comes the actual implementation of the project. For the implementation the first step is to choose the IDE for building the project. Dev C++has have been used. Now the designing part is done. For the coding part, all the C Language concepts and IDE syntax should be clear. There are several **in built functions** in C that are invoked to use the functions that have already been developed with the best possible algorithms. The major advantages of these in-built functions are that their time complexity is less, they can be called anywhere in the program, and they need not be compiled again and again. Now, we know that in-built functions are there in the language. But we still need to know what the **libraries** actually are. The built-in functions are stored in the libraries. So whenever you want to use any function then the header file, where the function definition is stored, needs to be included. A single library contains a group of functions and those functions are grouped together on the basis of their similar functionalities. Moreover, the name given to the library also demonstrates the type of functions saved in the library. For example, library stdio stands for standard input-output and the functions defined in this libraries are printf (standard output) and scanf (standard input). Similarly, math.h include all mathematical functions for example sin, cos, pow, etc. In this project, several libraries have been used. Some of these libraries along with their need to be included are discussed as follows:

1. **graphics.h:** This header file is used to use graphics that serve various purposes. The functions for using various colors have been defined in this header file. Apart from that, the functions for changing the text style, drawing shapes such as rectangle, circles, lines, etc., and time delay has also been defined in this library file.
2. **dos.h:** This header file is one of the most interesting files. It has many functions which enhance the capability of the program. In my project, I have used this header file to use delay() function, which makes the screen sleep for the specific time supplied as the argument to this function.
3. **string.h:** This header file is used to manipulate strings in different ways. There are various functions that have been used in this project. Some of the crucial functions used are strlen() to get the length of a string, strcmpi() to compare two strings, and strcpy() to copy one string to another.
4. **stdio.h:**  This header file refers Standard input-output headerUsed to perform input and output operations in C like scanf() and printf().
5. **conio.h:** This header file refers Console input-output header Perform console input and console output operations like clrscr() to clear the screen and getch() to get the character from the keyboard.
6. **stdlib.h:** This header file refers to the Standard library header Perform standard utility functions like dynamic memory allocation, using functions such as malloc() and calloc().
7. **ctype.h:** This header file refers to the Character type header Perform character type functions like isaplha() and isdigit(). To find whether the given character is an alphabet or a digit respectively.
8. **time.h**: (Time header) Perform functions related to date and time like setdate() and getdate(). To modify the system date and get the CPU time respectively.

**DATA STRUCTURE USED**

**ARRAYS:**

An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together. This makes it easier to calculate the position of each element by simply adding an offset to a base value, i.e., the memory location of the first element of the array (generally denoted by the name of the array).



# Project Design

# First Screen: When we open our output screen it will display the first screen of the project. In which the name of the project is written.

# When you press the enter key you will enter to the next screen.

# When we enter the second it will display the main menu bar which has a certain options like play the game, view the highest score, reset the score, help, and the last option is quit the game.

# 

# 

# The operation to be performed is selected by entering the operation number to be performed. If a valid operation number is entered then that particular operation number is performed. If the operation number is invalid then it shows an error.

# 

# Code for option 1: It contain several functions like second\_screen(), random(), asked(), random3(), asked3(), random2(), asked2().

# void second\_screen()

# {

# //char playername[20];

# 

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

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

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

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

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

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

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

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

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

# printf("\n\t\t\t\tEnter your choice:: ");

# char choice;

# choice=toupper(getch());

# printf("%c \n",choice);

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

# system("pause");

# 

# if (choice=='V')

# {

# show\_record();

# }

# else if (choice=='H')

# {

# help();getch();

# 

# }

# else if (choice=='R')

# {reset\_score();

# getch();

# }

# else if (choice=='Q')

# exit(1);

# else if(choice=='S')

# {

# system("cls");

# trial:

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

# printf("\n\t\t\tRegsister your name:");

# gets(playername);

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

# if(strlen(playername)==0)

# {

# system("cls");

# printf("\t\t\t Please enter your name to play game");

# goto trial;

# }

# printf("\n");

# system("pause");

# system("cls");

# RANDOM3();

# }

# else

# { system("cls");

# printf("\t\t\t Please enter the correct choice to start the game");

# second\_screen();

# }

# }

# RANDOM3(){

# char text[size],option1[size], option2[size], option3[size], option4[size], option[size], correct[size], category[size];

# char empty[256][256];

# int i,j,k;

# int arr[QNO],x,loc;

# time\_t t;

# char line[256];

# FILE \* fp;

# FILE \* output;

# int trialscore1=0;

# fp = fopen("Quiz2.txt","r");

# output = fopen("NewText2.txt","w");

# if(fp == NULL)

# {

# printf("\n\n\n \t\t\t Quiz File Can't Be OPENED \t\t\t \n\n\n");

# }

# if(output == NULL)

# {

# printf("\n\n\n \t\t\t NewText File Can't Be CREATED \t\t\t \n\n\n");

# }

# 

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

# printf("\t\t\t\*\t WELCOME TO LEVEL ONE \*\n");

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

# getch();

# system("cls");

# k = 0;

# 

# for (i = 0; i < 30; i++)

# {

# arr[i] = 10 \* i + 1;

# }

# for (j = 0; j < totques; j++)

# {

# srand((unsigned)time(&t));

# x = rand() % 3;

# loc = (3 \* j) + x;

# for (i = arr[3 \* j]; i < arr[loc]; i++)

# {

# fgets(empty[k],sizeof(empty[k]),fp);

# k++;

# }

# for (i = 1; i <= 10; i++)

# {

# fgets(line,sizeof(line),fp);

# if(i==1)

# {

# printf("\n\n\n\n\n\nQ%d) %s",j+1,line);

# fprintf(output,"Q%d) %s",j+1,line);

# 

# }

# 

# if(i==2)

# {

# fputs(line,output);

# }

# 

# if(i == 3)

# strncpy(option1,line,30);

# 

# if(i == 4)

# strncpy(option2,line,30);

# 

# if(i == 5)

# strncpy(option3,line,30);

# 

# if(i == 6)

# strncpy(option4,line,30);

# 

# if (i >1&&i<=7)

# {

# printf("%s",line);

# 

# }

# if (i == 8)

# strncpy(correct,line,30);

# 

# if (i == 9)

# strncpy(category,line,30);

# }

# asked(option, option1, option2, option3, option4, correct, category,output,&score1);

# 

# if(j!=9)

# {

# 

# if (arr[loc] + 10 < arr[3 \* (j + 1)])

# {

# for (i = arr[loc] + 10; i < arr[3 \* (j + 1)]; i++)

# {

# fgets(empty[k],sizeof(empty[k]),fp);

# k++;

# }

# }

# }

# 

# 

# system("cls");

# }

# 

# fputs(playername,output);

# fprintf( output," got total %d out of 40 ",score1);

# fclose(fp);

# 

# fclose(output);

# 

# output = fopen("Newtext2.txt","r");

# 

# while(fgets(line,sizeof(line),output))

# {

# printf(" %s",line);

# }

# 

# fclose(output);

# system("pause");

# system("cls");

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

# printf( "\t\t\t %s got total %d out of 40 ",playername,score1);

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

# getch();

# system("cls");

# 

# if(score1>28)

# {

# system("cls");

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

# printf("\t\t\t\* CONGRATS!!! YOU REACHED TO LEVEL 2 \*\n");

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

# system("pause");

# system("cls");

# RANDOM();

# }

# else

# {

# system("cls");

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

# printf("\t\t\t\* STUDY HARD!!! BETTER LUCK NEXT TIME \*\n");

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

# ff();

# add\_file();

# 

# getch();

# system("cls");

# second\_screen();

# }

# }

# void asked3(char option[size], const char option1[size], const char option2[size], const char option3[size], const char option4[size],

# const char correct[size], const char category[size],FILE \* fp,int \*score1)

# {

# printf("Your Option : ");

# scanf("%str",&option[0]);

# option[1] = '\0';

# strupr(option);

# 

# char correctanswer[size];

# char wronganswer[size];

# 

# fprintf(fp,"%s \n",category);

# 

# if (option[0] == correct[0])

# {

# if (option[0] == option1[0])

# fprintf(fp," \nYou entered correct answer : %s \n ", option1);

# if (option[0] == option2[0])

# fprintf(fp," \nYou entered correct answer : %s \n ", option2);

# if (option[0] == option3[0])

# fprintf(fp," \nYou entered correct answer : %s \n ", option3);

# if (option[0] == option4[0])

# fprintf(fp," \nYou entered correct answer : %s \n ", option4);

# 

# \*score1 += 4;

# 

# fprintf(fp," \n You are awarded 4 marks for this correct answer \n ");

# }

# else

# {

# 

# if (correct[0] == option1[0])

# strncpy(correctanswer, option1,24);

# if (correct[0] == option2[0])

# strncpy(correctanswer, option2,24);

# if (correct[0] == option3[0])

# strncpy(correctanswer, option3,24);

# if (correct[0] == option4[0])

# strncpy(correctanswer, option4,24);

# 

# correctanswer[100] = '\0';

# 

# if (option[0] == option1[0])

# strncpy(wronganswer, option1,24);

# if (option[0] == option2[0])

# strncpy(wronganswer, option2,24);

# if (option[0] == option3[0])

# strncpy(wronganswer, option3,24);

# if (option[0] == option4[0])

# strncpy(wronganswer, option4,24);

# wronganswer[100] = '\0';

# 

# fprintf(fp," You entered : %s \n", wronganswer);

# fprintf(fp," Correct answer : %s \n", correctanswer);

# fprintf(fp," \n You are awarded 0 marks for this wrong answer \n ");

# }

# 

# fprintf(fp," -----------------------------------------------------------------------------\n");

# fprintf(fp," -----------------------------------------------------------------------------\n");

# }

# RANDOM(){

# char text[size],option1[size], option2[size], option3[size], option4[size], option[size], correct[size], category[size];

# char empty[256][256];

# int i,j,k;

# int arr[QNO],x,loc;

# time\_t t;

# char line[256];

# FILE \* fp;

# FILE \* output;

# fp = fopen("Quiz.txt","r");

# output = fopen("NewText.txt","w");

# if(fp == NULL)

# {

# printf("\n\n\n \t\t\t Quiz File Can't Be OPENED \t\t\t \n\n\n");

# }

# if(output == NULL)

# {

# printf("\n\n\n \t\t\t NewText File Can't Be CREATED \t\t\t \n\n\n");

# }

# k = 0;

# 

# for (i = 0; i < 30; i++)

# {

# arr[i] = 10 \* i + 1;

# }

# for (j = 0; j < totques; j++)

# {

# srand((unsigned)time(&t));

# x = rand() % 3;

# loc = (3 \* j) + x;

# for (i = arr[3 \* j]; i < arr[loc]; i++)

# {

# fgets(empty[k],sizeof(empty[k]),fp);

# k++;

# }

# for (i = 1; i <= 10; i++)

# {

# fgets(line,sizeof(line),fp);

# if(i==1)

# {

# printf("\n\n\n\n\n\nQ%d) %s",j+1,line);

# fprintf(output,"Q%d) %s",j+1,line);

# 

# }

# if(i==2)

# {

# fputs(line,output);

# }

# 

# if(i == 3)

# strncpy(option1,line,30);

# 

# if(i == 4)

# strncpy(option2,line,30);

# 

# if(i == 5)

# strncpy(option3,line,30);

# 

# if(i == 6)

# strncpy(option4,line,30);

# 

# if (i >1&&i<=7)

# {

# printf("%s",line);

# 

# }

# if (i == 8)

# strncpy(correct,line,30);

# 

# if (i == 9)

# strncpy(category,line,30);

# }

# 

# asked(option, option1, option2, option3, option4, correct, category,output,&score2);

# 

# if(j!=9)

# {

# 

# if (arr[loc] + 10 < arr[3 \* (j + 1)])

# {

# for (i = arr[loc] + 10; i < arr[3 \* (j + 1)]; i++)

# {

# fgets(empty[k],sizeof(empty[k]),fp);

# k++;

# }

# }

# }

# 

# 

# system("cls");

# }

# //fprintf(output,"nice to meeet u\n");

# fputs(playername,output);

# fprintf( output," got total %d out of 40 ",score2);

# fclose(fp);

# 

# fclose(output);

# 

# 

# output = fopen("Newtext.txt","r");

# 

# while(fgets(line,sizeof(line),output))

# {

# printf(" %s",line);

# }

# 

# fclose(output);

# system("pause");

# system("cls");

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

# printf( "\t\t\t %s got total %d out of 40 ",playername,score2);

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

# getch();

# system("cls");

# if(score2>28)

# {

# system("cls");

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

# printf("\t\t\t\* CONGRATS!!! YOU REACHED TO LEVEL 3 \*\n");

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

# system("pause");

# system("cls");

# Random1();

# }

# else

# {

# system("cls");

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

# printf("\t\t\t\* STUDY HARD!!! BETTER LUCK NEXT TIME \t \*\n");

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

# ff();

# add\_file();

# getch();

# second\_screen();

# }

# 

# }

# void asked(char option[size], const char option1[size], const char option2[size], const char option3[size], const char option4[size],

# const char correct[size], const char category[size],FILE \* fp,int \*score2)

# {

# printf("Your Option : ");

# scanf("%str",&option[0]);

# option[1] = '\0';

# strupr(option);

# 

# char correctanswer[size];

# char wronganswer[size];

# 

# fprintf(fp,"%s \n",category);

# 

# if (option[0] == correct[0])

# {

# if (option[0] == option1[0])

# fprintf(fp," \nYou entered correct answer : %s \n ", option1);

# if (option[0] == option2[0])

# fprintf(fp," \nYou entered correct answer : %s \n ", option2);

# if (option[0] == option3[0])

# fprintf(fp," \nYou entered correct answer : %s \n ", option3);

# if (option[0] == option4[0])

# fprintf(fp," \nYou entered correct answer : %s \n ", option4);

# 

# \*score2 += 4;

# 

# fprintf(fp," \n You are awarded 4 marks for this correct answer \n ");

# }

# else

# {

# 

# if (correct[0] == option1[0])

# strncpy(correctanswer, option1,24);

# if (correct[0] == option2[0])

# strncpy(correctanswer, option2,24);

# if (correct[0] == option3[0])

# strncpy(correctanswer, option3,24);

# if (correct[0] == option4[0])

# strncpy(correctanswer, option4,24);

# 

# correctanswer[24] = '\0';

# 

# if (option[0] == option1[0])

# strncpy(wronganswer, option1,24);

# if (option[0] == option2[0])

# strncpy(wronganswer, option2,24);

# if (option[0] == option3[0])

# strncpy(wronganswer, option3,24);

# if (option[0] == option4[0])

# strncpy(wronganswer, option4,24);

# wronganswer[24] = '\0';

# 

# fprintf(fp," You entered : %s \n", wronganswer);

# fprintf(fp," Correct answer : %s \n", correctanswer);

# fprintf(fp," \n You are awarded 0 marks for this wrong answer \n ");

# }

# 

# fprintf(fp," -----------------------------------------------------------------------------\n");

# fprintf(fp," -----------------------------------------------------------------------------\n");

# }

# Random1(){

# char text[size],option1[size], option2[size], option3[size], option4[size], option[size], correct[size], category[size];

# char empty[256][256];

# int i,j,k;

# int arr[QNO],x,loc;

# time\_t t;

# char line[256];

# 

# FILE \* fp;

# FILE \* output;

# fp = fopen("Quiz1.txt","r");

# output = fopen("NewText1.txt","w");

# if(fp == NULL)

# {

# printf("\n\n\n \t\t\t Quiz File Can't Be OPENED \t\t\t \n\n\n");

# }

# if(output == NULL)

# {

# printf("\n\n\n \t\t\t NewText File Can't Be CREATED \t\t\t \n\n\n");

# }

# k = 0;

# 

# for (i = 0; i < 30; i++)

# {

# arr[i] = 10 \* i + 1;

# }

# for (j = 0; j < totques; j++)

# {

# srand((unsigned)time(&t));

# x = rand() % 3;

# loc = (3 \* j) + x;

# for (i = arr[3 \* j]; i < arr[loc]; i++)

# {

# fgets(empty[k],sizeof(empty[k]),fp);

# k++;

# }

# for (i = 1; i <= 10; i++)

# {

# fgets(line,sizeof(line),fp);

# if(i==1)

# {

# printf("\n\n\n\n\n\nQ%d) %s",j+1,line);

# fprintf(output,"Q%d) %s",j+1,line);

# 

# }

# 

# if(i==2)

# {

# fputs(line,output);

# }

# 

# if(i == 3)

# strncpy(option1,line,30);

# 

# if(i == 4)

# strncpy(option2,line,30);

# 

# if(i == 5)

# strncpy(option3,line,30);

# 

# if(i == 6)

# strncpy(option4,line,30);

# 

# if (i >1&&i<=7)

# {

# printf("%s",line);

# 

# }

# if (i == 8)

# strncpy(correct,line,30);

# 

# if (i == 9)

# strncpy(category,line,30);

# }

# 

# asked(option, option1, option2, option3, option4, correct, category,output,&score3);

# 

# if(j!=9)

# {

# 

# if (arr[loc] + 10 < arr[3 \* (j + 1)])

# {

# for (i = arr[loc] + 10; i < arr[3 \* (j + 1)]; i++)

# {

# fgets(empty[k],sizeof(empty[k]),fp);

# k++;

# }

# }

# }

# 

# 

# system("cls");

# }

# 

# fputs(playername,output);

# fprintf( output," got total %d out of 40 ",score3);

# fclose(fp);

# 

# fclose(output);

# 

# output = fopen("Newtext1.txt","r");

# 

# while(fgets(line,sizeof(line),output))

# {

# printf(" %s",line);

# }

# 

# fclose(output);

# 

# system("pause");

# system("cls");

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

# printf( "\t\t\t %s got total %d out of 40 ",playername,score3);

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

# getch();

# system("cls");

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

# printf("\t\t\t\* YOU HAVE A VERY GOOD IQ KEEP IT UP \*\n");

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

# ff();

# add\_file();

# getch();

# system("cls");

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

# printf("\t\t\t %s you got %d for 120 \n",playername,finalscore);

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

# getch();

# system("cls");

# second\_screen();

# 

# }

# void asked1(char option[size], const char option1[size], const char option2[size], const char option3[size], const char option4[size],

# const char correct[size], const char category[size],FILE \* fp,int \*score3)

# {

# printf("Your Option : ");

# scanf("%str",&option[0]);

# option[1] = '\0';

# strupr(option);

# 

# char correctanswer[size];

# char wronganswer[size];

# 

# fprintf(fp,"%s \n",category);

# 

# if (option[0] == correct[0])

# {

# if (option[0] == option1[0])

# fprintf(fp," \nYou entered correct answer : %s \n ", option1);

# if (option[0] == option2[0])

# fprintf(fp," \nYou entered correct answer : %s \n ", option2);

# if (option[0] == option3[0])

# fprintf(fp," \nYou entered correct answer : %s \n ", option3);

# if (option[0] == option4[0])

# fprintf(fp," \nYou entered correct answer : %s \n ", option4);

# 

# \*score3 += 4;

# 

# fprintf(fp," \n You are awarded 4 marks for this correct answer \n ");

# }

# else

# {

# 

# if (correct[0] == option1[0])

# strncpy(correctanswer, option1,24);

# if (correct[0] == option2[0])

# strncpy(correctanswer, option2,24);

# if (correct[0] == option3[0])

# strncpy(correctanswer, option3,24);

# if (correct[0] == option4[0])

# strncpy(correctanswer, option4,24);

# 

# correctanswer[24] = '\0';

# 

# if (option[0] == option1[0])

# strncpy(wronganswer, option1,24);

# if (option[0] == option2[0])

# strncpy(wronganswer, option2,24);

# if (option[0] == option3[0])

# strncpy(wronganswer, option3,24);

# if (option[0] == option4[0])

# strncpy(wronganswer, option4,24);

# wronganswer[24] = '\0';

# 

# fprintf(fp," You entered : %s \n", wronganswer);

# fprintf(fp," Correct answer : %s \n", correctanswer);

# fprintf(fp," \n You are awarded 0 marks for this wrong answer \n ");

# }

# 

# fprintf(fp," -----------------------------------------------------------------------------\n");

# fprintf(fp," -----------------------------------------------------------------------------\n");

# }Output will be shown below:

# 

# 

# 

# 

# 

# 

# 

# 

# Next screen which will be shown is given to tell that your answer vs correct answer

# 

# 

# 

# 

# 

# 

# 

# 

# If the student score less than or equal 28 marks he will come out of quiz and come back to main menu else it will go to next level that is 2 level.

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Now as you see you have scored the 36 out of 40 and the passing mark is 28 which means you reached to next level. And that is level 2.

# 

# 

# 

# 

# 

# 

# 

# 

# 

# So we see that score is <= 28 then it will come out of quiz come back to main menu.

# 

# 

# 

# 

# 

# 

# 

# 

# Now as you see you have scored the 32 out of 40 and the passing mark is 28 which means you reached to next level. And that is level 3.

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Code for option 2

# void show\_record()

# {

# sorting();

# system("cls");

# FILE \*f;

# f=fopen("score.txt","r");

# fscanf(f,"%s%d",&playername,&finalscore);

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

# printf("\n\n\t\t %s has secured the Highest Score %d",playername,finalscore);

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

# fclose(f);

# getch();

# second\_screen();

# }

# Output will be:

# 

# Code for option 3:

# void reset\_score()

# {

# system("cls");

# float sc;

# char nm[20];

# FILE \*f;

# f=fclose(fopen("score.txt", "w"));

# f=fclose(fopen("marks.txt", "w"));

# fclose(f);

# system("cls");

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

# printf("\t\t\t\* Ohh!! NO DATA TO SHOW \*\n");

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

# getch();

# system("cls");

# second\_screen();

# }

# Output will be:

# 

# Code for option 4:

# void help()

# {

# system("cls");

# printf("\n\n\t HELP");

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

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

# printf("\n\t >> There are three rounds in the game, EASY LEVEL , MEDIUM LEVEL AND DIFFICULT LEVEL");

# printf("\n\t >> In warmup round you will be asked a total of 10 questions to test your general");

# printf("\n\t knowledge. You will be eligible to play the game if you can give atleast 7");

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

# printf("\n\t >> Your game starts with the EASY LEVEL. In this lEVEL you will be asked");

# printf("\n\t total 10 questions 7 right answer will be take you to next level .");

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

# printf("\n\t right option");

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

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

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

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

# printf("\n\n\t\t\t\*\*\*\*\*C PROGRAM QUIZ GAME is developed by RINKI \*\*\*\*\*\*\*\*");

# getch();

# second\_screen();

# 

# }

# Output will be:

# 

# Code for option 5:

# display(){

# sorting();

# system("cls");

# char ch;

# FILE \*fp;

# printf("Name Score\n");

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

# fp=fopen("score.txt","r");

# while ((ch=getc(fp))!=EOF){

# if(ch == '\t')

# printf("\t");

# else

# printf("%c",ch);

# }

# fclose(fp);

# 

# getch();

# system("cls");

# second\_screen();

# }

# 

# Code for option 6:

# exit(1);

# 

# TESTING WITH VALIDATIONS

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is [Defect](https://www.guru99.com/defect-management-process.html) free. It involves the execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps, or missing requirements contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a [White Box](https://www.guru99.com/white-box-testing.html) and [Black Box Testing](https://www.guru99.com/black-box-testing.html). In simple terms, Software Testing means Verification of Application Under Test (AUT). For checking the project is running properly under different circumstances following **datasets have** been supplied to the program:

# Validation of User choice:

# 

# Validation of user name:

# 

# SIGNIFICANCE OF THE PROJECT

# Now that the project has been developed, it should have some significance. Any project that has been developed would be having some purpose. Significance will tell, how the project would be beneficial for people of different application areas. As far as application areas are concerned, there is a wide range. Out of them, let us select the most popular application areas to understand the significance of the project. The significance of the project will be demonstrated in the following application areas:

# Significance to society

# Significance to the IT community

# Significance to Education and Research

# To society: The quiz game project is designed in Dev C++ to test the general knowledge of the user. The topics for the quiz can be science, sports, geography, movies, or any other. You cannot run this project in a turbo c compiler. The quiz will have questions connecting all the areas in the contest. The user can gain knowledge from my project by playing different levels of the quiz and improving his/her score. The quiz even provides a healthy competition to the user as it compares the marks with other users and displays the result. This quiz can help the user prepare for any general knowledge exams with different fields of questions. So, in this way my project “Quiz Game” has significance to society.

# To IT community: The IT community is making the world right now. Information Technology (IT) is a hot burning topic in this era. Everything, these days, is being done by using technology by hook or by crook, even if it concerns the medical industry. Every country is focusing on strengthening its IT sector. Although my project is very basic, still it can be of some significance to the IT community and help the medical industry. Based on my idea for this project, the IT community can think of preparing a Quiz Game that will be more knowledgeable, accurate, and faster. Many beginners in the IT field can use a platform like GitHub to seek knowledge and many IT professionals can help me improve the project marginally. Whatever you develop there is always a scope. From beginners to experts everyone can gain and give knowledge at any point in time. Hopefully, my project would bring the interests of many IT experts

# and beginners, as a result, the IT community, as well as the conspiring students, might be benefitted from my project and also, and I can gain some knowledge to improve my skills as well. All in all, the significance to the IT community is not assured, but the world is full of possibilities and hopes and I hope I might contribute something to strengthen the IT field and provide knowledge to the aspiring students.

# To Education and Research: My project “Quiz Game” has great significance as far as education and research are concerned. Not only IT students but students preparing for exams can also seek a lot from this project. Being a student, I know about the problems that arise while researching and being educated. Students and research scholars find it difficult to follow the step to enhance their skills. My project could possibly first step towards their goal. The simplicity of this project would make them more interested towards inject and they will try to learn and would also try to implement their knowledge to the project. This way both the project and student knowledge will be improved. My project might motivate them to think something innovative. So, this way my project can serve the area of Education and Research.

# 

**CONCLUSION AND RECOMMENDATIONS**

The Quiz Game designed is very interactive, user-friendly, and easy to play. The game allows the user to choose the topic of his interest and test his knowledge. Recommendations are more and more games like these should be designed as it improves the knowledge and at the same time test your IQ level.

Quiz games should be designed in a more interactive and friendly way so that more people can play, enjoy, and at the same time improve, enhance and test their knowledge. In the quiz game, random questions will appear when you will play the game so that you cannot bored while the same game again and again. It saves time as it allows the number of students to give the exam at a time and displays the results as the test gets over, so no need to wait for the result.

**FUTURE ENHANCEMENTS**

* The main aim of our project is to create a good interaction between the student and teacher.
* We are trying to do the project at the best level to satisfy all the end-users.
* In the future we are decided to provide more security to our website which may not be hacked.
* And we give the choice to students to add their name under the faculty who they wish and get advice for their betterment.
* It will be more empowering.
* Next we are aiming to provide some online classes on our website for general knowledge so that students can become more confident to appear in government exams.

# BIBLIOGRAPHY/ REFERENCES

* **C Programming Absolute Beginner’s Guide** by ‘Greg Perry and Dean Miller,
* **The C Programming Language** by Brian W Kernighan and Dennis M Ritchie
* **C: The Complete Reference** by Herbert Schildt
* [https://www.tutorialspoint.com/c/index.htm](https://www.tutorialspoint.com/cplusplus/index.htm)
* <https://www.w3schools.com/c/>
* <https://www.geeksforgeeks.org/c-tutorial/>
* <https://beginnersbook.com/2017/08/c-tutorial-for-beginners/>