Multi-Threaded Online Quiz System(M-OQS)

A COURSE PROJECT REPORT

By

STUDENT NAME	**(REGISTER NO)**
Chinnam Lakshmi Durga	**RA1911003010127**
Saksham Rajput	**RA1911003010131**
Vageesh Dixit	**RA191003010133**

Under the guidance of

Dr. G. Manju

In partial fulfilment for the

Course

of

18CSC302J - COMPUTER NETWORKS

in Computer

Science and

Engineering

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Chengalpattu District

NOVEMBER 2021

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report “Multithreaded Quiz System" is the bonafide work of **Chinnam Lakshmi Durga(RA1911003010127), Saksham Rajput(RA1911003010131), Vageesh Dixit(RA1911003010133)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. G Manju	**Dr.E. Sasikala,**
Course faculty	**Course Coordinator**
Department	**Associate Professor,**
SRM Institute of Science and Technology	**Data Science and Business Systems**
Potheri, SRM Nagar, Kattankulathur,	SRM Institute of Science and Technology
Tamil Nadu 603203	Potheri, SRM Nagar, Kattankulathur,
	Tamil Nadu 6032

ACKNOWLEDGEMENT

I express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

I would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement.

I express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

I would like to express my heartfelt thanks to the Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project.

We wish to express my sincere thanks to **Course Audit Professor Dr.M.LAKSHMI, Professor and Head, Data Science and Business Systems** and **Course Coordinator Dr.E. Sasikala, Associate Professor, Data Science and Business Systems** for their constant encouragement and support.

I am highly thankful to our Course project Internal guide **Dr. G Manju, Faculty Advisor, Computer Science and Engineering,** for **her** assistance, timely suggestion and guidance throughout the duration of this course project.

I extend our gratitude to the Student **HOD, Computer Networks** and my Departmental colleagues for their Support.

Finally, I thank my parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on us to complete my Course project

TABLE OF CONTENTS

ABSTRACT :

A server having more than one thread is known as Multithreaded Server. When a client sends the request, a thread is generated through which a user can communicate with the server. We need to generate multiple threads to accept multiple requests from multiple clients at the same time.

Using TCP Multithreading, which can be used to cater to unlimited clients at the same time, which contains three types of tests of subject:

- Science.
- Mathematics.
- English.

Every test type further contains different areas as following

- Science (Physics, Chemistry, Biology)
- Mathematics (Geometry, Algebra, IQ)
- English (Analogies, Antonyms, RC Questions)

The main objective of OQS is to efficiently evaluate the candidate through a fully automated system that not only saves a lot of time but also gives fast results.

The project is totally built at the administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for Marks, Courses, Papers.

INTRODUCTION

Online examination providers’ focus on creating effective assessment questions and exam's feedback delivery to students. In the report, we present techniques that are pertinent to the elements of the assessment process: answers submission and computerized grading after submission.
As modern organizations are automated and computers are working according to the instructions. It replaced the paperwork and overcame the outcomes of traditional ways of examinations using paper or pen. Being cost and time effective, it became essential for the coordination of human beings, commodities and computers in a modern organization.
The administrators, instructors, and students who are attending online examinations can communicate with the system through this project, thus facilitating effective implementation and monitoring of various activities of Online Examinations like conducting exams as per scheduled basis and delivering results to that particular user or student. The details of students who attempt the Online Examination are maintained by the administrator.
Candidates can give their course’s examination in a specific domain. The questions can appear in both mode MCQ (Multiple Choice Questions) and answer in paragraphs.Online Quiz System (OQS) is a web-based examination system where quizzes are taken online i.e. through the internet or intranet using a computer system.
The purpose of OQS is to take Semester Quizzes in an efficient manner and not waste time checking the paper. Teachers can administer quizzes using the OQS .The system will show results after the examination is finished. A teacher has control in the question bank and is supposed to make a schedule for the quiz. The system carries out the examination and auto-grading for multiple choice questions which is fed into the system. Administrative control of the whole system is provided.
The main objective of the project on MCQ Quiz Applications is to manage the details of Students, Examinations, Marks, Courses, Papers. It manages all the information about Students, Results, Papers,Students.

REQUIREMENT ANALYSIS:

REQUIREMENT SPECIFICATION

Hardware Requirements

Processor : Intel Core Duo or

higher Clock Speed RAM :

512 MB

Hard Disk : 40 MB (Minimum free space)

Software Requirements

Operating System : Windows 7 and

above

Platform : Linux

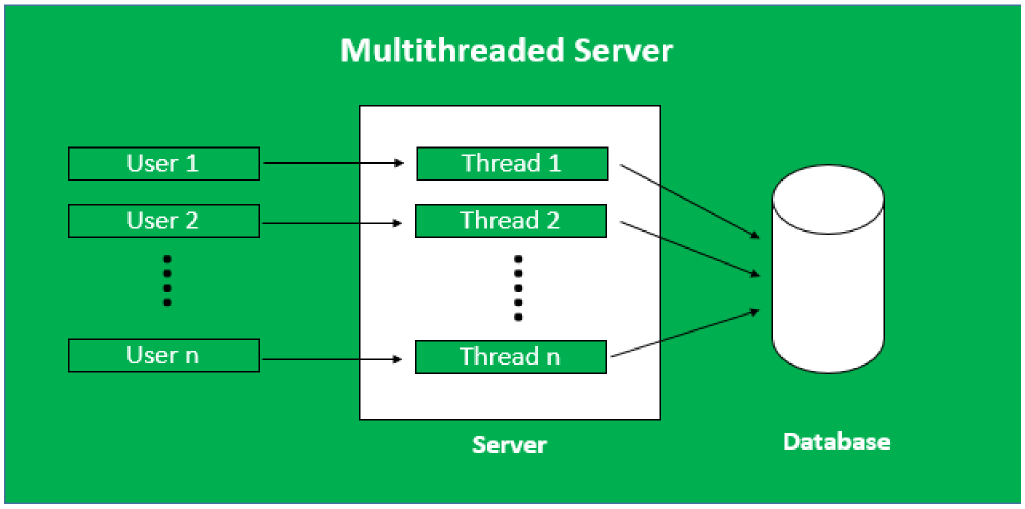
Back End : MySql

Special Tools : Opencv,

Xuggle Server : Apache

Tomcat

ARCHITECTURE & DESIGN



1. Design a Main Server.
2. Design a Client.
3. Design 3 TCP Sub-servers.

The string a sub-server send to the main server will have the following format: Test_name, Sub-server IP, Sub-server Port

1. Each sub-server will connect to the main server. Main server will make a database about the sub-server information it gets from the sub-servers.

2. When a client connects with the main server and makes a request to give a specific test (e.gMath), the main server will search the record in the database it has for the corresponding sub- server and will send sub-server information to the client. The string a main server will send to the client will have the following format:

   Test_name, Sub-server IP, Sub-server Port

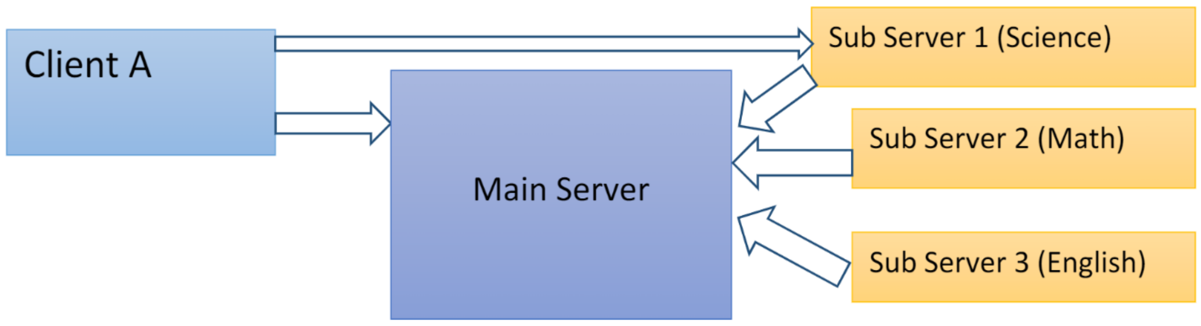
3. Using that information the client will reconnect to the sub server using TCP connection (Note: more than one client can be connected to any particular sub server at a single time). The sub-server will then show different test-types to the client. Client will select one and the sub-server will assign random marks to the client on that test type in the range of 1 to 100.

4. The sub server will send the final result to the client and also to the main server for the record of that client. The format of the information will be as follows:

Client IP, Client Port, Test Name, Test Type, Marks

5. The client will show the result on the terminal and then quit. The main server will store the information in a file with the below mentioned format and keep listening for next clients.

Client IP, Client Port, Test Name, Test Type, Marks



IMPLEMENTATION

The application consists mainly of 5 servers:

- A main server.
- 3 sub-servers (for three different tests).
- A client server.

The main server does not want to overburden itself by communicating with all the incoming clients and handle the tests, so it only stores the information about which sub-server contains what sort of test (Three subservers each of which contains one type of test i.e., Science, Math or English).

Following are the steps on how the application works:

When a sub-server connects to the main server it sends the main server a string containing the Test Name which it can carry out along with its IP and Port number (e.g. Mathematics, Sub-server IP, Sub-server Port). The format of the string is as follows:

Test_name, Sub-server IP, Sub-server Port

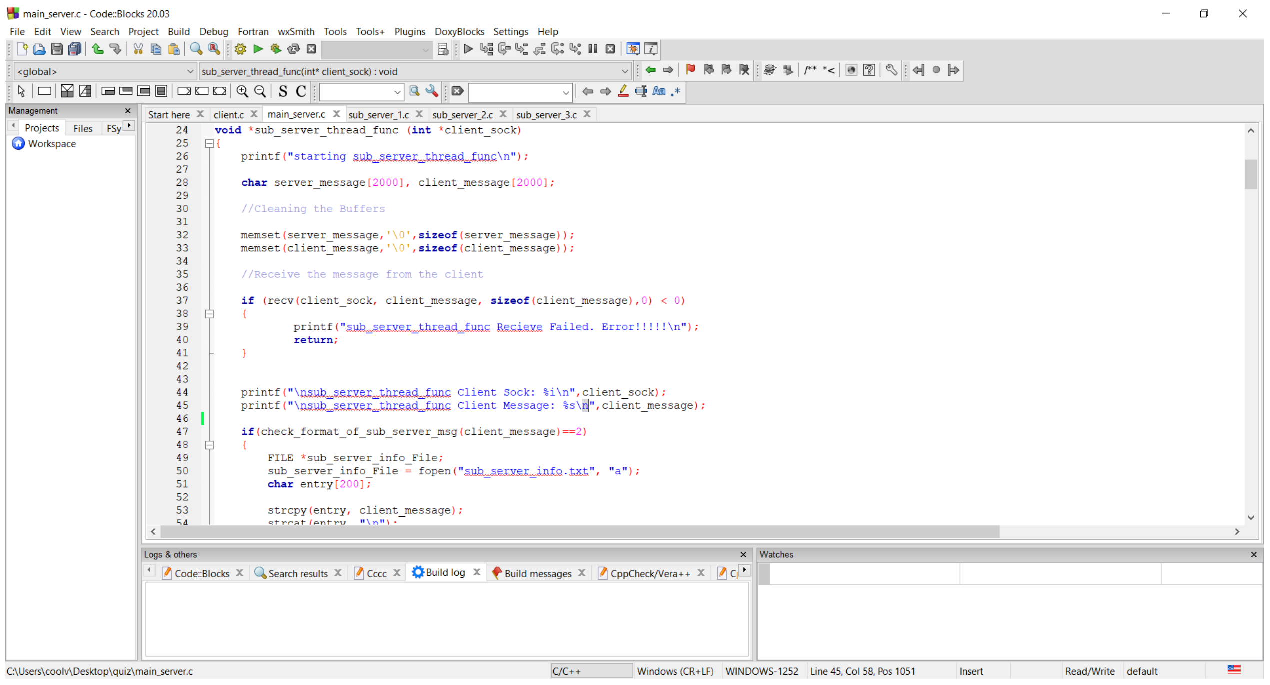
1. Whenever a client connects to the main server, it will display three kinds of tests (Science, Mathematics and English) to the client. The client will then choose the type of test (e.g., Math). The main server will then send the information of the corresponding sub-server to the client. The string a main server will send to the client will have the following format:

   Test_name, Sub-server IP, Sub-server Port

2. The client will then connect to the corresponding sub-server. The sub-server will then show different types of the corresponding test (e.g, Geometry, Algebra and IQ for Math Sub-server) to the client. The client will select the test type and complete the test (You don't need any test. Just make the sub-server to assign random marks to the client from 1 to 100).
3. Then the sub-server will send the information related to the client (Client IP, Client Port, Test Name, Test Type, Marks) back to the main server and also to the client.
4. The client will then terminate. The main server will store the information within a file for all the clients:

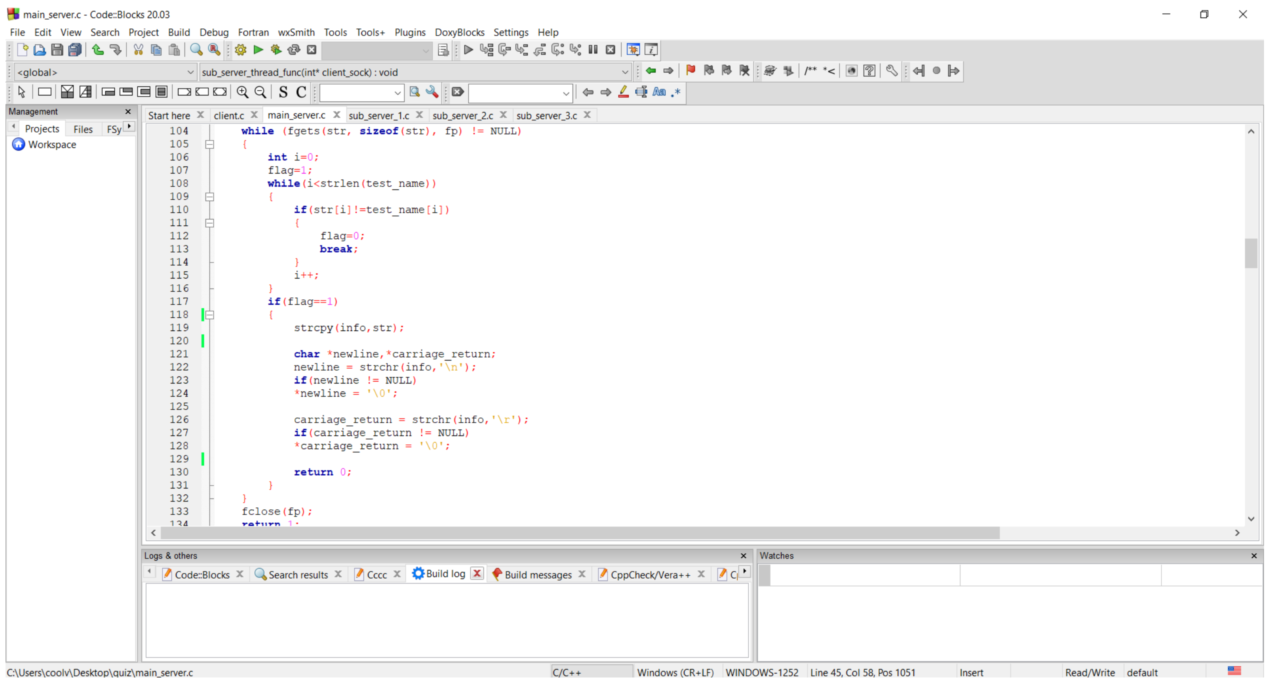
   Client IP, Client Port, Test Name, Test Type, Marks

Main Server Implementation:



```c
void *sub_server_thread_func (int *client_sock)
{
    printf("starting sub_server_thread_func\n");

    char server_message[2000], client_message[2000];

    //Cleaning the Buffers

    memset(server_message,'\0',sizeof(server_message));
    memset(client_message,'\0',sizeof(client_message));

    //Receive the message from the client

    if (recv(client_sock, client_message, sizeof(client_message),0) < 0)
    {
            printf("sub_server_thread_func Recieve Failed. Error!!!!!\n");
            return;
    }


    printf("\nsub_server_thread_func Client Sock: %i\n",client_sock);
    printf("\nsub_server_thread_func Client Message: %s\n",client_message);

    if(check_format_of_sub_server_msg(client_message)==2)
    {
        FILE *sub_server_info_File;
        sub_server_info_File = fopen("sub_server_info.txt", "a");
        char entry[200];

        strcpy(entry, client_message);
        strcat(entry, "\n");
```

C:\Users\coolv\Desktop\quiz\main_server.c



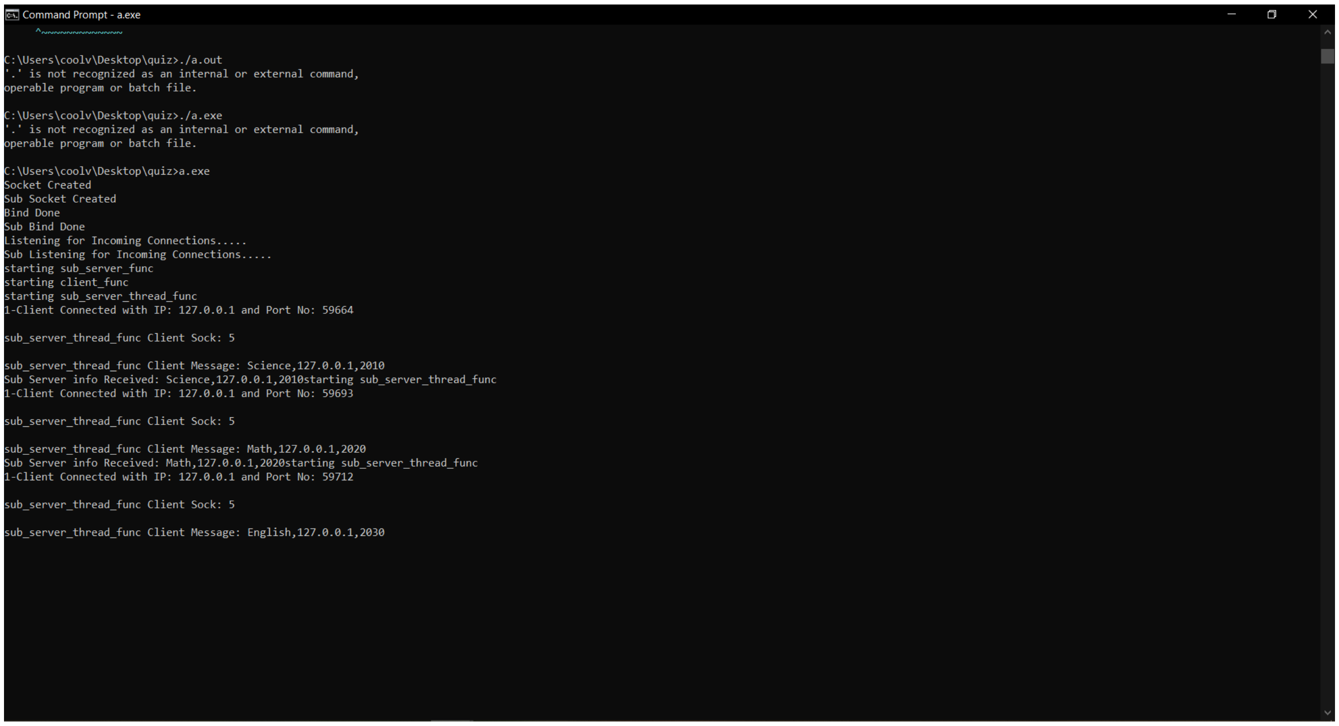
```c
    while (fgets(str, sizeof(str), fp) != NULL)
    {
        int i=0;
        flag=1;
        while(i<strlen(test_name))
        {
            if(str[i]!=test_name[i])
            {
                flag=0;
                break;
            }
            i++;
        }
        if(flag==1)
        {
            strcpy(info,str);

            char *newline,*carriage_return;
            newline = strchr(info,'\n');
            if(newline != NULL)
            *newline = '\0';

            carriage_return = strchr(info,'\r');
            if(carriage_return != NULL)
            *carriage_return = '\0';

            return 0;
        }
    }
    fclose(fp);
    return 1;
```

C:\Users\coolv\Desktop\quiz\main_server.c

EXPERIMENT RESULTS & ANALYSIS

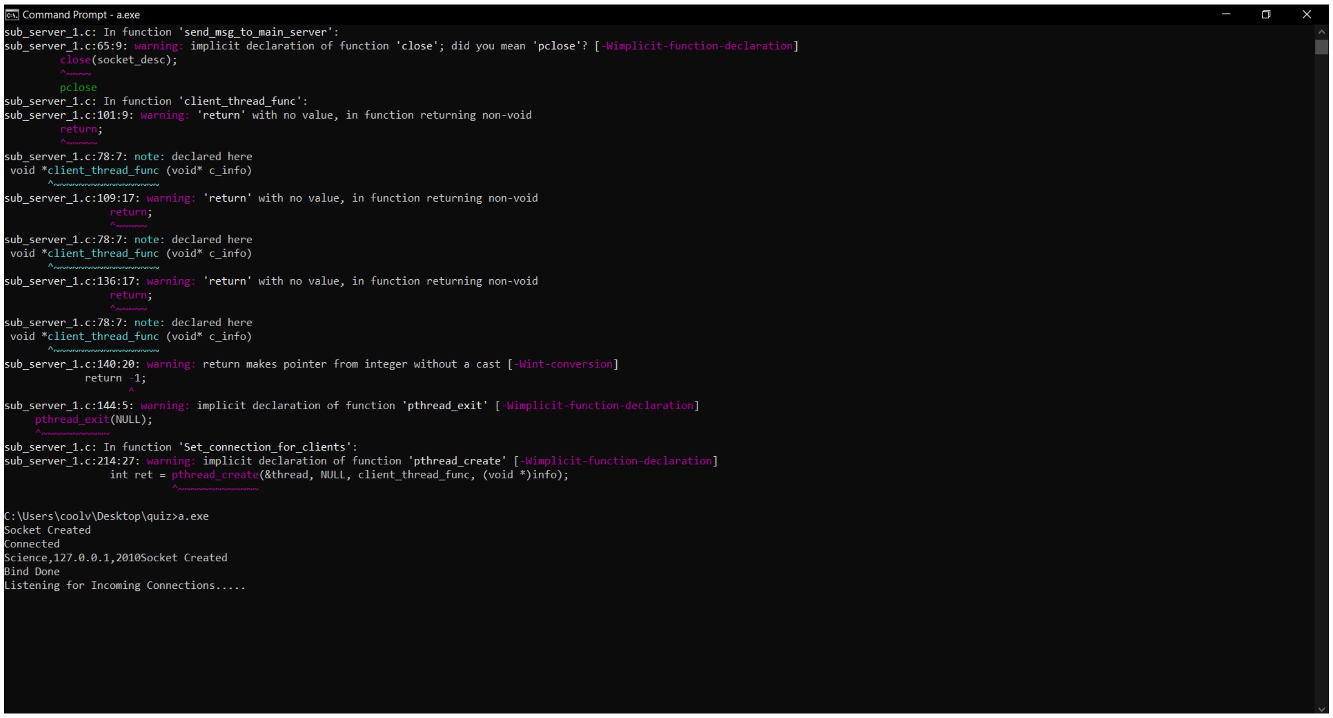
RESULTS:

#Server: Configure the main server first to connect to the sub servers and client.



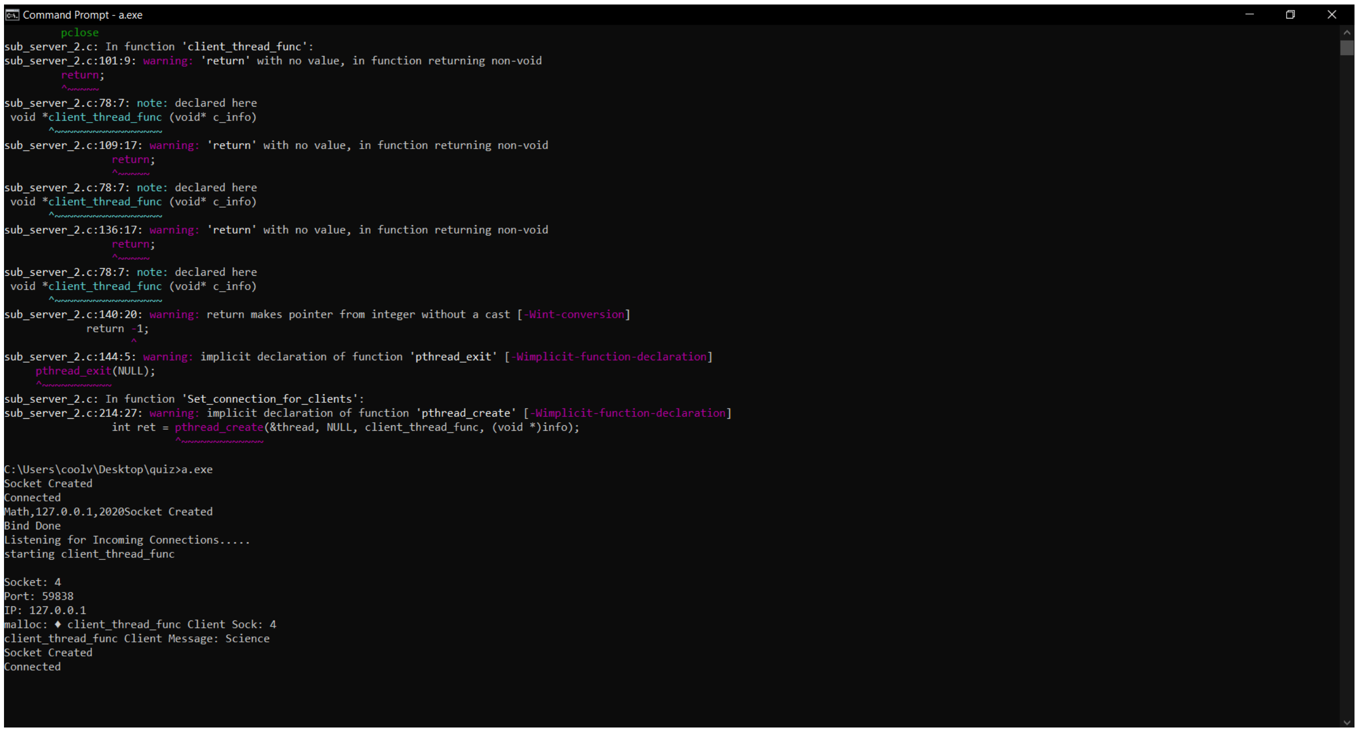
```
Command Prompt - a.exe
     ^~~~~~~~~~~~~~

C:\Users\coolv\Desktop\quiz>./a.out
'.' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\coolv\Desktop\quiz>./a.exe
'.' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\coolv\Desktop\quiz>a.exe
Socket Created
Sub Socket Created
Bind Done
Sub Bind Done
Listening for Incoming Connections.....
Sub Listening for Incoming Connections.....
starting sub_server_func
starting client_func
starting sub_server_thread_func
1-Client Connected with IP: 127.0.0.1 and Port No: 59664

sub_server_thread_func Client Sock: 5

sub_server_thread_func Client Message: Science,127.0.0.1,2010
Sub Server info Received: Science,127.0.0.1,2010starting sub_server_thread_func
1-Client Connected with IP: 127.0.0.1 and Port No: 59693

sub_server_thread_func Client Sock: 5

sub_server_thread_func Client Message: Math,127.0.0.1,2020
Sub Server info Received: Math,127.0.0.1,2020starting sub_server_thread_func
1-Client Connected with IP: 127.0.0.1 and Port No: 59712

sub_server_thread_func Client Sock: 5

sub_server_thread_func Client Message: English,127.0.0.1,2030
```

#Sub_server_1.c



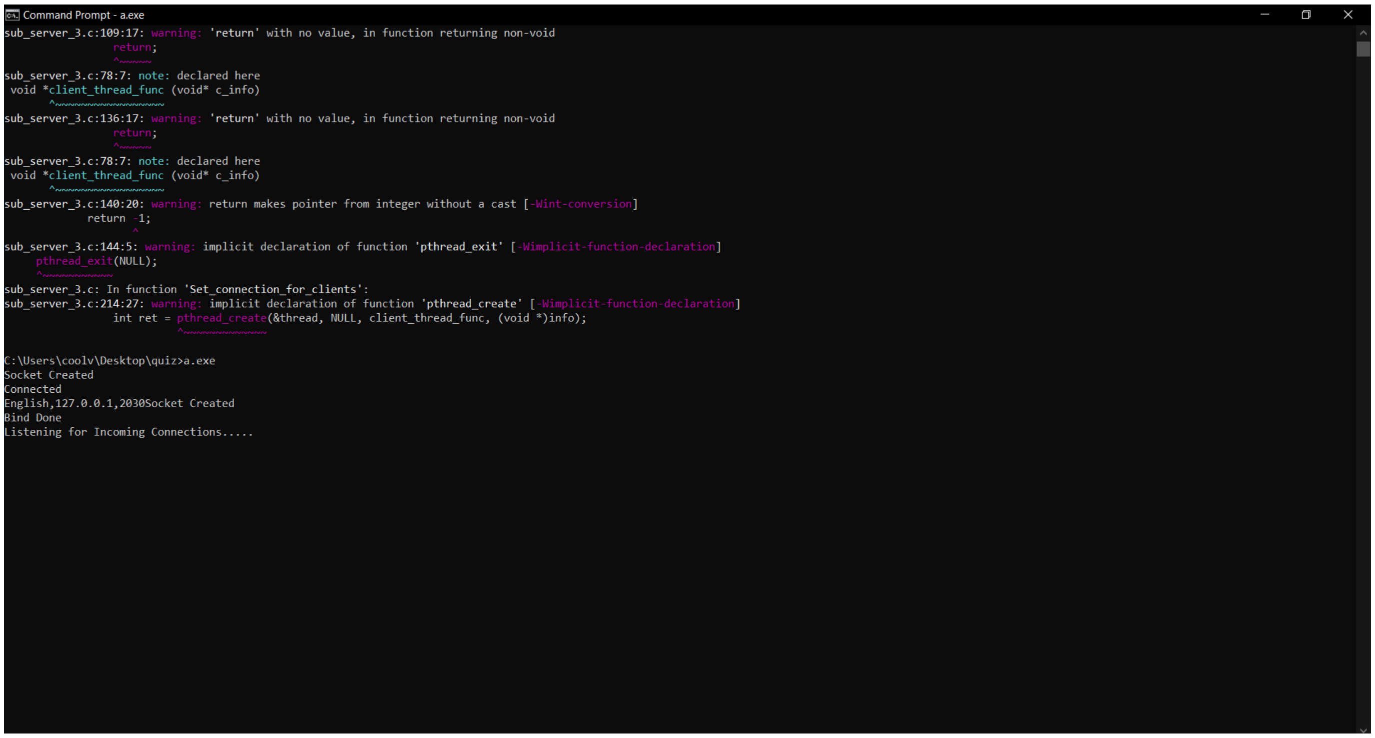
```
Command Prompt - a.exe
sub_server_1.c: In function 'send_msg_to_main_server':
sub_server_1.c:65:9: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
         close(socket_desc);
         ^~~~~
         pclose
sub_server_1.c: In function 'client_thread_func':
sub_server_1.c:101:9: warning: 'return' with no value, in function returning non-void
         return;
         ^~~~~~
sub_server_1.c:78:7: note: declared here
 void *client_thread_func (void* c_info)
       ^~~~~~~~~~~~~~~~~~
sub_server_1.c:109:17: warning: 'return' with no value, in function returning non-void
                 return;
                 ^~~~~~
sub_server_1.c:78:7: note: declared here
 void *client_thread_func (void* c_info)
       ^~~~~~~~~~~~~~~~~~
sub_server_1.c:136:17: warning: 'return' with no value, in function returning non-void
                 return;
                 ^~~~~~
sub_server_1.c:78:7: note: declared here
 void *client_thread_func (void* c_info)
       ^~~~~~~~~~~~~~~~~~
sub_server_1.c:140:20: warning: return makes pointer from integer without a cast [-Wint-conversion]
             return -1;
                    ^
sub_server_1.c:144:5: warning: implicit declaration of function 'pthread_exit' [-Wimplicit-function-declaration]
     pthread_exit(NULL);
     ^~~~~~~~~~~~
sub_server_1.c: In function 'Set_connection_for_clients':
sub_server_1.c:214:27: warning: implicit declaration of function 'pthread_create' [-Wimplicit-function-declaration]
                 int ret = pthread_create(&thread, NULL, client_thread_func, (void *)info);
                           ^~~~~~~~~~~~~~

C:\Users\coolv\Desktop\quiz>a.exe
Socket Created
Connected
Science,127.0.0.1,2010Socket Created
Bind Done
Listening for Incoming Connections.....
```

#Sub_server_2.c



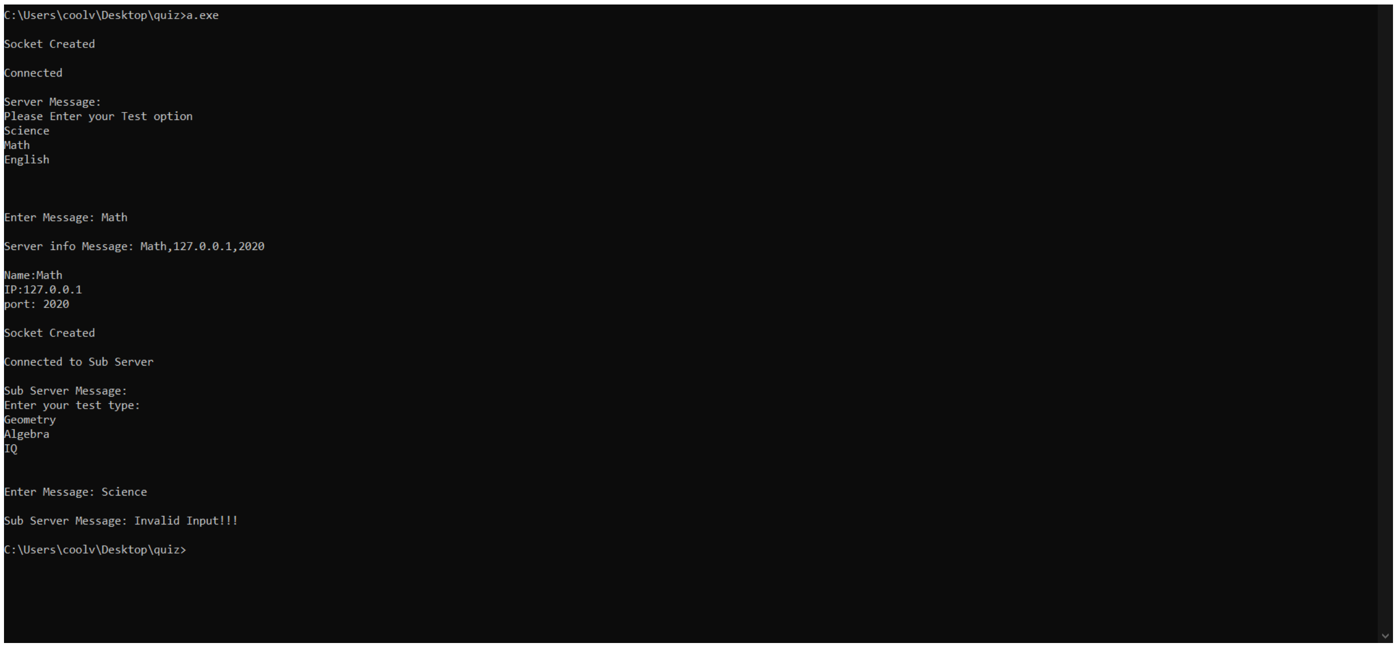
```
Command Prompt - a.exe
        pclose
sub_server_2.c: In function 'client_thread_func':
sub_server_2.c:101:9: warning: 'return' with no value, in function returning non-void
        return;
        ^~~~~~
sub_server_2.c:78:7: note: declared here
 void *client_thread_func (void* c_info)
       ^~~~~~~~~~~~~~~~~~
sub_server_2.c:109:17: warning: 'return' with no value, in function returning non-void
                return;
                ^~~~~~
sub_server_2.c:78:7: note: declared here
 void *client_thread_func (void* c_info)
       ^~~~~~~~~~~~~~~~~~
sub_server_2.c:136:17: warning: 'return' with no value, in function returning non-void
                return;
                ^~~~~~
sub_server_2.c:78:7: note: declared here
 void *client_thread_func (void* c_info)
       ^~~~~~~~~~~~~~~~~~
sub_server_2.c:140:20: warning: return makes pointer from integer without a cast [-Wint-conversion]
             return -1;
                    ^
sub_server_2.c:144:5: warning: implicit declaration of function 'pthread_exit' [-Wimplicit-function-declaration]
     pthread_exit(NULL);
     ^~~~~~~~~~~~
sub_server_2.c: In function 'Set_connection_for_clients':
sub_server_2.c:214:27: warning: implicit declaration of function 'pthread_create' [-Wimplicit-function-declaration]
                int ret = pthread_create(&thread, NULL, client_thread_func, (void *)info);
                          ^~~~~~~~~~~~~~

C:\Users\coolv\Desktop\quiz>a.exe
Socket Created
Connected
Math,127.0.0.1,2020Socket Created
Bind Done
Listening for Incoming Connections.....
starting client_thread_func

Socket: 4
Port: 59838
IP: 127.0.0.1
malloc: ♦ client_thread_func Client Sock: 4
client_thread_func Client Message: Science
Socket Created
Connected
```

#Sub_server_3.c



```
Command Prompt - a.exe
sub_server_3.c:109:17: warning: 'return' with no value, in function returning non-void
                return;
                ^~~~~~
sub_server_3.c:78:7: note: declared here
 void *client_thread_func (void* c_info)
       ^~~~~~~~~~~~~~~~~~
sub_server_3.c:136:17: warning: 'return' with no value, in function returning non-void
                return;
                ^~~~~~
sub_server_3.c:78:7: note: declared here
 void *client_thread_func (void* c_info)
       ^~~~~~~~~~~~~~~~~~
sub_server_3.c:140:20: warning: return makes pointer from integer without a cast [-Wint-conversion]
             return -1;
                    ^
sub_server_3.c:144:5: warning: implicit declaration of function 'pthread_exit' [-Wimplicit-function-declaration]
     pthread_exit(NULL);
     ^~~~~~~~~~~~
sub_server_3.c: In function 'Set_connection_for_clients':
sub_server_3.c:214:27: warning: implicit declaration of function 'pthread_create' [-Wimplicit-function-declaration]
                int ret = pthread_create(&thread, NULL, client_thread_func, (void *)info);
                          ^~~~~~~~~~~~~~

C:\Users\coolv\Desktop\quiz>a.exe
Socket Created
Connected
English,127.0.0.1,2030Socket Created
Bind Done
Listening for Incoming Connections.....
```

#Client.c- Configure the client at the end and start the Quiz system based on the choices.



```
C:\Users\coolv\Desktop\quiz>a.exe

Socket Created

Connected

Server Message:
Please Enter your Test option
Science
Math
English



Enter Message: Math

Server info Message: Math,127.0.0.1,2020

Name:Math
IP:127.0.0.1
port: 2020

Socket Created

Connected to Sub Server

Sub Server Message:
Enter your test type:
Geometry
Algebra
IQ


Enter Message: Science

Sub Server Message: Invalid Input!!!

C:\Users\coolv\Desktop\quiz>
```

RESULT ANALYSIS:

Successfully, created a socket based project involving 3- sub-servers, a main server and a client to check a student’s marks in the sub-server subjects and to further create questions and answers quiz system between client and the sub server based on the choice of the user.

CONCLUSION & FUTURE WORK:

Therefore, an Online Quiz System was successfully designed and developed using socket programming concepts and using multiple servers.

Hence we created a deeper understanding of computer networking concepts.

Further scope of this project depends upon various parameters and requirements of the user such as scalability, usability and security, which have been achieved and executed to an extent in this project.

REFERENCES:

1. Object Oriented Analysis and Design with Applications by Grady Booch.
2. Object Oriented Software Engineering - Ivar Jacobson Pearson Education
3. www.wikipedia.com
4. www.c-sharpcorner.com
5. www.slideshare.com
6. www.w3Cschool.com
7. https://www.geeksforgeeks.org/multithreaded-servers-in-java/
8. https://computernetworking747640215.wordpress.com/2018/07/05/dns-server-configuration-in-packet-tracer/