Socket Quiz System

A COURSE PROJECT REPORT

By

Shubham Gusain (RA2011027010048)
Mohammad Azhar Sofi (RA2011027010035)
Sarthak Sethi (RA2011027010051)
Konde Veerendranadh (RA2011027010063)

Under the guidance of

Dr. Anand M

In partial fulfilment for the Course

of

18CSC302J - COMPUTER NETWORKS

in DSBS

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Chenpalpattu District

NOVEMBER 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this mini project report "**Socket Based Quiz System**" is the bonafide work of **Sarthak Sethi (RA2011027010051), Mohammad Azhar Sofi (RA2011027010035), Veerendranadh (RA2011027010063), Shubham Gusain (RA2011027010048)** who carried out the project work under my supervision.

SIGNATURE

DR. M. ANAND
ASSISTANT PROFESSOR
DATA SCIENCE AND BUSINESS STUDIES
SRM Institute of Science and Technology

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr.C.MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement. We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V. Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing

Dr. Revathi Venkataraman, for imparting confidence to completemy course project

We wish to express my sincere thanks to

Course Audit Professor Dr. Annapurani Panaiyappan, Professor and Head, Department of Networking and Communications and **Course Coordinators** for their constantencouragement and support.

We are highly thankful to our my Course project Faculty **Dr. Anand M,** for his assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HOD Dr. Lakshmi, Data Science and Business Studies** and my Departmental colleagues for their Support.

Finally, we thank our 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 me to complete my Course project.

TABLE OF CONTENTS

1. ABSTRACT

Socket Quiz System is a system where the quiz is taken. The Quiz System aims to conduct Quizzes efficiently and conserve efforts put in for assessment. The main objective of the Online Quiz System is to efficiently evaluate the candidate through a fully automated system that saves much time and gives fast results. Quizzes can be administered using the Quiz System. A teacher has control of the question bank and is supposed to configure the question bank for the quiz. The system carries out the test and auto-grading for questions which is fed into the system. Administrative control of the whole system is provided. Quiz System aims to focus on creating a practical quiz experience. We employ socket programming and multithreading to generate a quiz environment that can easily host multiple clients with this project. The administrators and participants attempting the online quiz can communicate with the system through this project, thus facilitating effective implementation and monitoring of various activities of Quiz like conducting the quiz, generating questions and accepting the responses. The introduction of online quiz software can replace the conventional system of assessment. Various quiz running agencies can now perform and host many participants freely and cost-effectively through computer-based tests.

2. INTRODUCTION

2.1 Scenario Description

As modern organizations are automated, and computers are working as per the instructions, it becomes essential to coordinate human beings, commodities and computers in a current organization. The administrators and participants attempting the quiz can communicate with the system through this project, thus facilitating effective implementation and monitoring of various activities of Quiz like conducting the quiz, generating questions and accepting the responses. Technological advancements in this era of digitization and is a boon to the world have also been advantageous to the educational sector.

The introduction of online quiz software can replace the conventional system of assessment. Various quiz conducting agencies can now perform and host many participants freely and cost-effectively through computer-based tests. Quizzes can be administered using the Quiz System. A teacher has control of the question bank and is supposed to configure the question bank for the quiz. The system carries out the test and auto-grading for questions which is fed into the system. Administrative control of the whole system is provided.
Quiz System aims to focus on creating a practical quiz experience.
We employ socket programming and multithreading to generate a quiz environment that can easily host multiple clients with this project.

a. TCP/IP - TCP/IP stands for Transmission Control Protocol/Internet Protocol and is a suite of communication protocols used to interconnect network devices on the internet. TCP/IP is also used as a communications protocol in a private computer network (an intranet or extranet). TCP/IP specifies how data is exchanged over the internet by providing end-to-end communications that identify how it should be broken into packets, addressed, transmitted, routed and received at the destination. TCP/IP requires little central management and is designed to make networks reliable to recover automatically from the failure of any device on the web. TCP/IP uses the client-server model of communication. A user or machine (a client) is provided with a service (like sending a webpage) by another computer (a server) in the network. The advantages of using the TCP/IP model include the following:

i. helps establish a connection between different types of computers;
ii. works independently of the OS;
iii. supports many routing protocols;
iv. uses a client-server architecture that is highly scalable;
v. can be operated separately;
vi. supports several routing protocols; and
vii. is lightweight and doesn't place unnecessary strain on a network or computer.

b. Multithreading - Multithreading is a specialized form of multitasking. Multitasking is the feature that allows your computer to

run two or more programs concurrently. In general, there are two types of multitasking: processbased and thread-based. Process-based multitasking handles the concurrent execution of programs. Thread-based multitasking deals with the simultaneous performance of pieces of the same program. A multithreaded program contains two or more parts that can run concurrently. Each part of such a program is called a thread, and each thread defines a separate path of execution. C does not contain any built-in support for multithreaded applications.
Instead, it relies entirely upon the operating system to provide this feature. POSIX Threads or Pthreads provides API available on many Unix-like POSIX systems such as FreeBSD, NetBSD, GNU/Linux, Mac OS X and Solaris. The maximum number of threads that a process may create is implementation-dependent. Once created, threads are peers and may make other threads. There is no implied hierarchy or dependency between threads.

3. LITERATURE SURVEY

Article: International Journal of Computer Science and Information Technologies.

Author: Limi Kalita

Description: To run this program Download server.py and client.py,then run server.py file in terminal using the command "python server.py" and run client.py file in terminal using the command "python client.py" then client gets questions and asks for answer. When client enters answer which is an option the clicks enter,that option is encoded in to bytes and sent to server side on server sidethen server analysis the answer. If the answer is correct answer for that question then the server increases the marks scored by that client by 1, then next question is sent to client by server. After completing all thequestions server sends the marks scored by client to client and then the quiz is ended. In this code, we have server.py and client.py. In server.py file there are mainly four functions, they are: create_socket() - to create a socket on client side, bind_socket() - to bind the created socket to servers IP address and port number , accepting_connection() - to accept connections from clients, Communication_client() - to organize quiz competition for client. Also a threading concept is also used while communicating with client. In create_socket() global variables like host, ports are created. host is ip address of server. port consists of random port number of the server with which communication happens. In bind_socket() socket created is binded to the server IP and port and starts listening. In accepting_connection(). when a client requests for connection it is accepted by the server and a thread is created by server for conducting quiz for client. In Communication_client() quiz for client is conducted. Answers answered by client is received by server from client through socket. On client side a socket is created and an infinite loop is used to participatein quiz.

Article: The Socket Programming and Software Design for Communication Based on Client/Server

Author : Ming Xue.

Description: This paper introduces the application of the client/server(C/S) mode, the concept and the programming principle of the socket based on C/S. The method of software design for the communication between the client, server-process using the socket mechanism is mainly analyzed, and gives examples of connection-oriented service program. The transmission layer can provide connection oriented to use TCP protocol, connectionless-oriented to use UDP protocol. There are two different kinds of services with different kinds of sockets. Only by understanding the characters of TCP protocol and UDP protocol, the service provided by the two protocols for application, can we know what deal with in these protocols, what need to deal with in application, can we easier compile vigorously and healthily, highly efficient client service program.A network is composed of computers which is either a client or a server. A server is a program that is offering some service whereas a client is a program that is requesting some service. Servers are powerful computers or processes dedicated to managing disk drives (file servers), printers (print servers), or network traffic (network services) whereas clients are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices and even processing power. When these programs are executed, as a result, a client and a server process are created simultaneously and these two processes communicate with each other by reading from and writing to sockets. These sockets are the programming interfaces provided by the TCP and UDP protocols for stream and datagram communication respectively of the transport layer which is a part of the TCP/IP stack. When creating a network application, the developer's main task is to write the code for both the client and server programs. The client/server application that is covered here is a proprietary client/server application. A single developer (or development team) creates both the client and server programs, and the developer has complete control over what goes in the code. How to get started with socket programming for beginners José Manuel Ortega

Article: Socket Programming- Internet Networking Protocol

Author: Ajay Yadav

Description: This article will give you a broad understanding of key networking concepts, such as ISO stack and TCP/IP, and how applications can logically and physically distribute in a network environment. This article also covers .NET base classes for using various network protocols, particularly HTTP, TCP and UDP, to access networks and the Internet as a client. Apart from that, we shall learn to manipulate IP address and DNS lookup. Finally, we'll examine the anatomy of sockets in depth and see how to develop a client and server application in order to set up socket connection. A network protocol is a standard set of rules that determine how systems will communicate across networks. Two different systems that use the same protocols can communicate and understand each other despite their differences. The OSI reference model provides important guidelines used by vendors, engineers and others. Packet switched networks mean that data travelling along network pathways is divided into small, routable packages referred to as a packet. If a communication link between two points on a network goes down, the packet can be routed through the remaining network connection to their intended destination. The internet protocols work together as a layered protocol stack which consists of the application, presentation, network, transport, data link layer and physical layers. Each layer in the protocol stack provides a set of services to the layer above it. The server's job is to set up an endpoint for clients to connect to and passively wait for connections. The typical TCP server first constructs a TcpListener instance, specifying the local address and port, and then calls the Start() method. This socket listens for incoming connections on the specified port then calls the AcceptTcpClient() method of TcpListener to get the next incoming client connection. Upon establishment of a new client connection, an instance of TcpClient for the new connection is created and returned by the AcceptTcp- Client() call. Thereafter, it communicates with the client using the Read() and Write() methods of TcpClient's NetworkStream and finally closes the new client socket connection and stream using the Close() methods of NetworkStream and TcpClient.

Article: Socket Programming for scalable systems.

Author: Steven Heines.

Description: The java.net.Socket class is Java's interface to a network socket and allows you to perform all four fundamental socket operations.In computer networking, a port is an application-specific or process-specific software construct serving as a communications endpoint in a computer's host operating system. A port is associated with an IP address of the host, as well as the type of protocol used for communication. The purpose of ports is to uniquely identify different applications or processes running on a single computer. The protocols that primarily use ports are the Transport Layer protocols, such as the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) of the Internet software stack, often called TCP/IP (Transport Control Protocol/Internet Protocol)stack, as shown in figure :1.4. They use ports to map incoming data to a particular process running on a computer. So a port will identify a socket on a host. The Internet has been very popular in the past few years. With its popularity still growing, increased demand for Internet network software has grown as well. One of the greatest advantages to developing Internet software with Java is in its robust networking support built into the core language. The java.net package provides us with classes representing URLs, URL connections and sockets.

Article: Multiple Multi Instance Secured Distribution File server.

Author: Ajay.K.Sharma.

Description: Java Sockets, RMI (Remote Method Invocation) of Sun Microsystems or CORBA (Common Object Requests Broker Architecture) of OMG (Object Management Group), are well known used for distributed system development. RMI, one of the core Java APIs since version 1.1., is a way that Java programmers can write object-oriented programs in which objects on different computers can interact with each other. A client can call a remote object in a server, and the server can also be a client of other remote objects. CORBA is well suited for Distributed system working on different platforms with varying specifications and environment [2, 3]. But the implementation of RMI and CORBA are very complex and specialized task. Hence, Java Socket is better choice for easy implementation. A P2P network is a network of computers or nodes where there is a concept of equality that is anyone could act like a client if required a data and the one who serves act like a server, it is different from traditional client/server architecture where only dedicated system could act like a server and requesting computers.

Article: Socket Programming in Python.

Author: Nathen Jennings

Descripton : Sockets and the socket API are used to send messages across a network. They provide a form of IPC. The network can be a logical, local network to the computer, or one that's physically connected to an external network, with its own connections to other networks. The obvious example is the Internet, which you connect to via your ISP. By the end of this tutorial, you'll understand how to use the main functions and methods in Python's socket module to write your own client-server applications. You'll know how to use a custom class to send messages and data between endpoints, which you can build upon and utilize for your own applications. The client calls.connect to establish a connection to the server and initiate the three-way handshake. The handshake step is important because it ensures that each side of the connection is reachable in the network, in other words that the client can reach the server and vice-versa. It may be that only one host, client, or server can reach the other.

Article: Learn socket programming with Python.

Author: Ashish Kumar

Description: Sockets use protocols to determine the connection type used to do port-to-port communication between client and server machines. There are protocols for IP addressing, Dynamic Name Servers (DNS), email, FTP, and so on. Each of these server types uses different protocol definitions to transfer data. There are two types of socket connections: stateful and stateless (or connectionless) connections. With stateful connections, the protocol requires an acknowledgement from the target machine that the data did indeed get there and that all the data is intact. The Terminal Control Protocol is a stateful protocol because it needs packet routing confirmations between source and destination ports. Services that require that the data be sent and acknowledged often use the TCP protocol as the basis for their specific protocol. For example, the Simple Mail Transfer Protocol (SMTP) for sending email is a TCP-based protocol. It is important to know that an email got to where it was meant to go. Stateless connections do not require data transfer acknowledgement like TCP does. A commonly used stateless protocol is the Universal Datagram Protocol (UDP). A DNS service uses this protocol as the basis for packet routing. UDP is often used for larger data packet routing and extends the UDP protocol to include more information about packet routing.

Article: Socket Programming in python and how to master it.

Author: Suresh

Descriptor: Sockets are the backbone of networking. They make the transfer of information possible between two different programs or devices. For example, when you open up your browser, you as a client are creating a connection to the server for the transfer of information. A server is either a program, a computer, or a device that is devoted to managing network resources. Servers can either be on the same device or computer or locally connected to other devices and computers or even remote. There are various types of servers such as database servers, network servers, print servers. The bind() method accepts two parameters as a tuple (host, port). However, it's better to use 4-digit port numbers as the lower ones are usually occupied. The listen() method allows the server to accept connections. Here, 5 is the queue for multiple connections that come up simultaneously. The minimum value that can be specified here is 0 (If you give a lesser value, it's changed to 0). In case no parameter is specified, it takes a default suitable one. A client is either a computer or software that receives information or services from the server. In a client-server module, clients requests for services from servers. The best example is a web browser such as Google Chrome, Firefox, etc. These web browsers request web servers for the required web pages and services as directed by the user. Other examples include online games, online chats.

4. REQUIREMENTS

a. Hardware Requirements -

Processor: Intel Core Duo or Higher

RAM: 1 GB

Hard Disk: 500 MB (Minimum free space)

b. Software Requirements –

i. Client Side

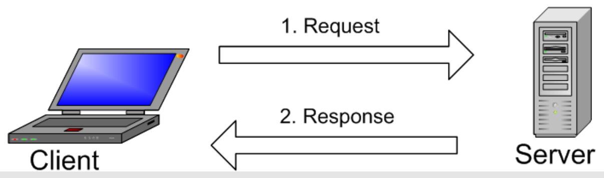
Operating System: Windows 7 or higher

ii. Server Side

Operating Server: Windows 7 or higher

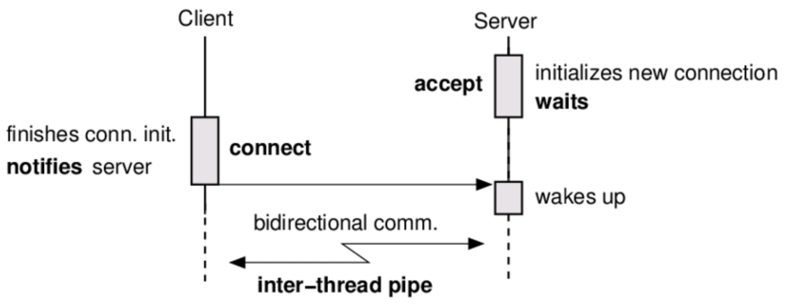
5. ARCHITECTURE AND DESIGN

I. Architecture Diagram



The client and server distribution model is structured on the roles of master and slave; the client acts as the master and requests service from the server (the slave). The server responds to the request from the client. This model implies a one-to-many relationship; the server typically serves multiple clients, while each client deals with a single server.

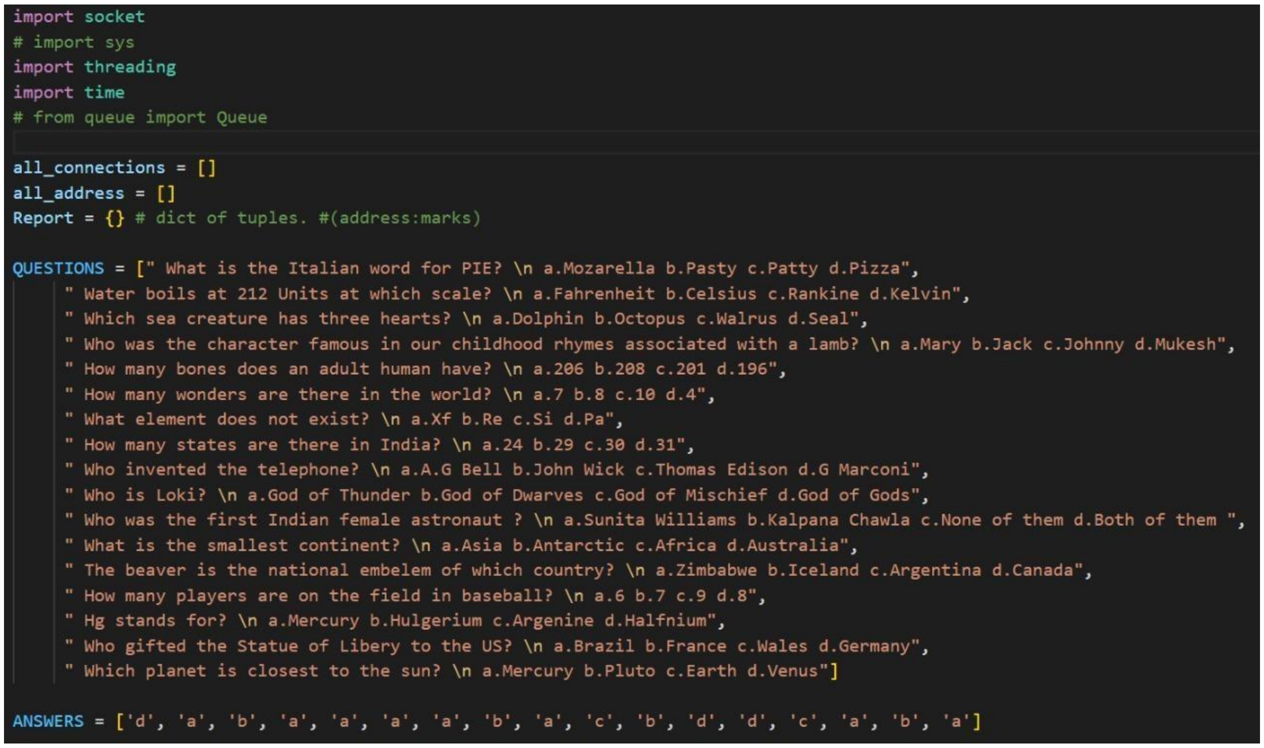
II. Communication Diagram



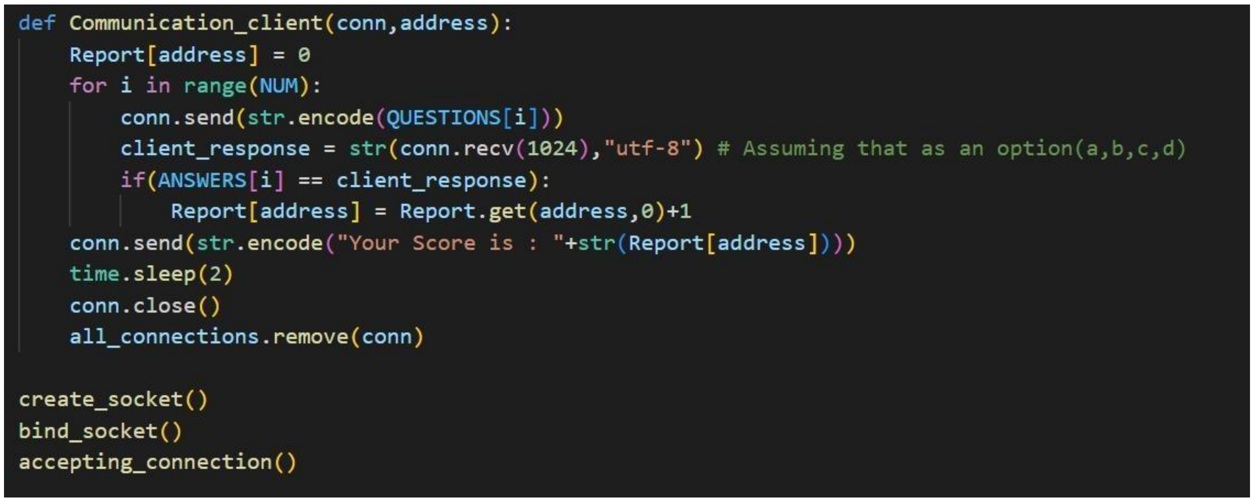
Datagram socket processes, unlike stream socket processes, are not clearly distinguished by server and client roles. The distinction lies in connected and unconnected sockets. An unconnected socket can be used to communicate with any host, but a connected socket can send data to and receive data from one host only. Both connected and unconnected sockets transmit data without verification. After a packet has been accepted by the datagram interface, neither its integrity nor its delivery can be assured.

6. IMPLEMENTATION

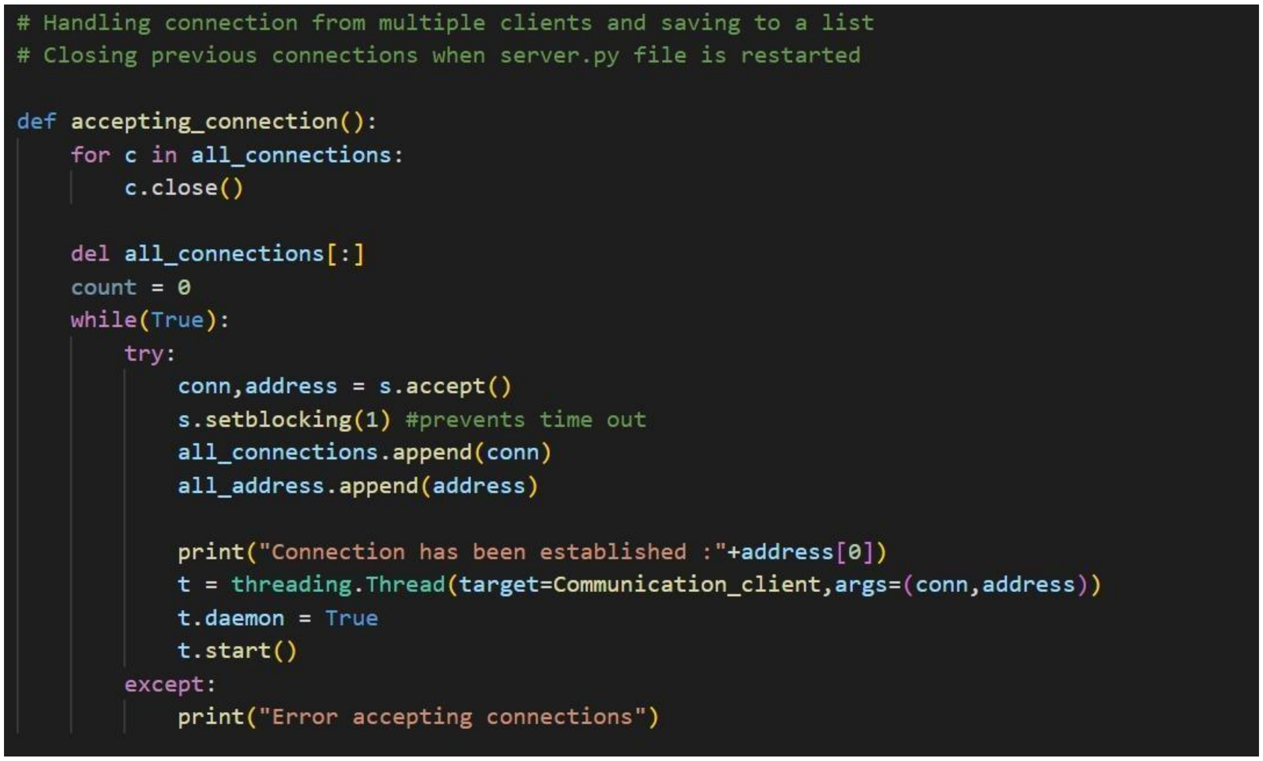
Server Side:



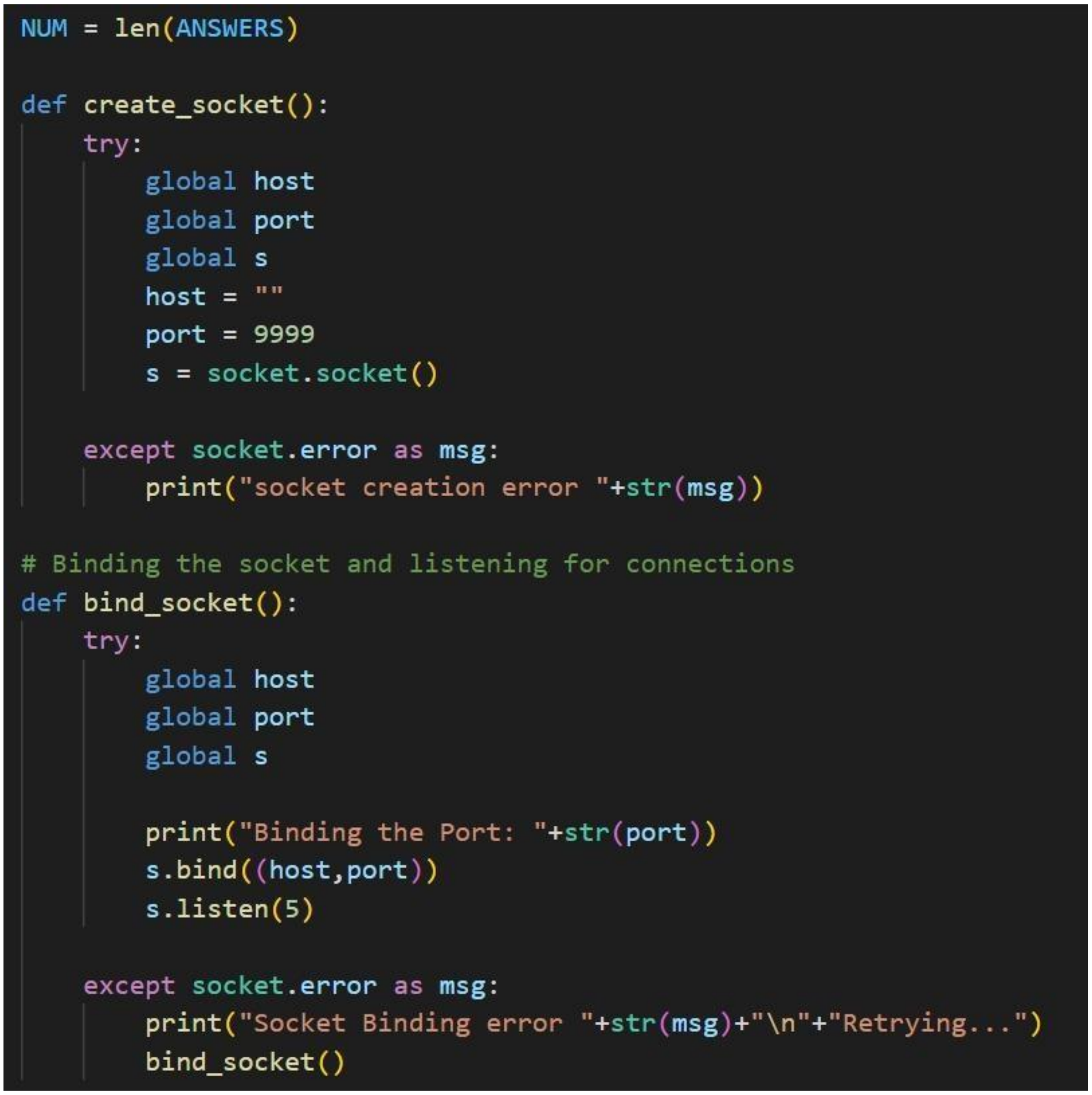
```python
import socket
# import sys
import threading
import time
# from queue import Queue

all_connections = []
all_address = []
Report = {} # dict of tuples. #(address:marks)

QUESTIONS = [" What is the Italian word for PIE? \n a.Mozarella b.Pasty c.Patty d.Pizza",
        " Water boils at 212 Units at which scale? \n a.Fahrenheit b.Celsius c.Rankine d.Kelvin",
        " Which sea creature has three hearts? \n a.Dolphin b.Octopus c.Walrus d.Seal",
        " Who was the character famous in our childhood rhymes associated with a lamb? \n a.Mary b.Jack c.Johnny d.Mukesh",
        " How many bones does an adult human have? \n a.206 b.208 c.201 d.196",
        " How many wonders are there in the world? \n a.7 b.8 c.10 d.4",
        " What element does not exist? \n a.Xf b.Re c.Si d.Pa",
        " How many states are there in India? \n a.24 b.29 c.30 d.31",
        " Who invented the telephone? \n a.A.G Bell b.John Wick c.Thomas Edison d.G Marconi",
        " Who is Loki? \n a.God of Thunder b.God of Dwarves c.God of Mischief d.God of Gods",
        " Who was the first Indian female astronaut ? \n a.Sunita Williams b.Kalpana Chawla c.None of them d.Both of them ",
        " What is the smallest continent? \n a.Asia b.Antarctic c.Africa d.Australia",
        " The beaver is the national embelem of which country? \n a.Zimbabwe b.Iceland c.Argentina d.Canada",
        " How many players are on the field in baseball? \n a.6 b.7 c.9 d.8",
        " Hg stands for? \n a.Mercury b.Hulgerium c.Argenine d.Halfnium",
        " Who gifted the Statue of Libery to the US? \n a.Brazil b.France c.Wales d.Germany",
        " Which planet is closest to the sun? \n a.Mercury b.Pluto c.Earth d.Venus"]

ANSWERS = ['d', 'a', 'b', 'a', 'a', 'a', 'a', 'b', 'a', 'c', 'b', 'd', 'd', 'c', 'a', 'b', 'a']
```



```python
def Communication_client(conn,address):
    Report[address] = 0
    for i in range(NUM):
        conn.send(str.encode(QUESTIONS[i]))
        client_response = str(conn.recv(1024),"utf-8") # Assuming that as an option(a,b,c,d)
        if(ANSWERS[i] == client_response):
            Report[address] = Report.get(address,0)+1
    conn.send(str.encode("Your Score is : "+str(Report[address])))
    time.sleep(2)
    conn.close()
    all_connections.remove(conn)

create_socket()
bind_socket()
accepting_connection()
```

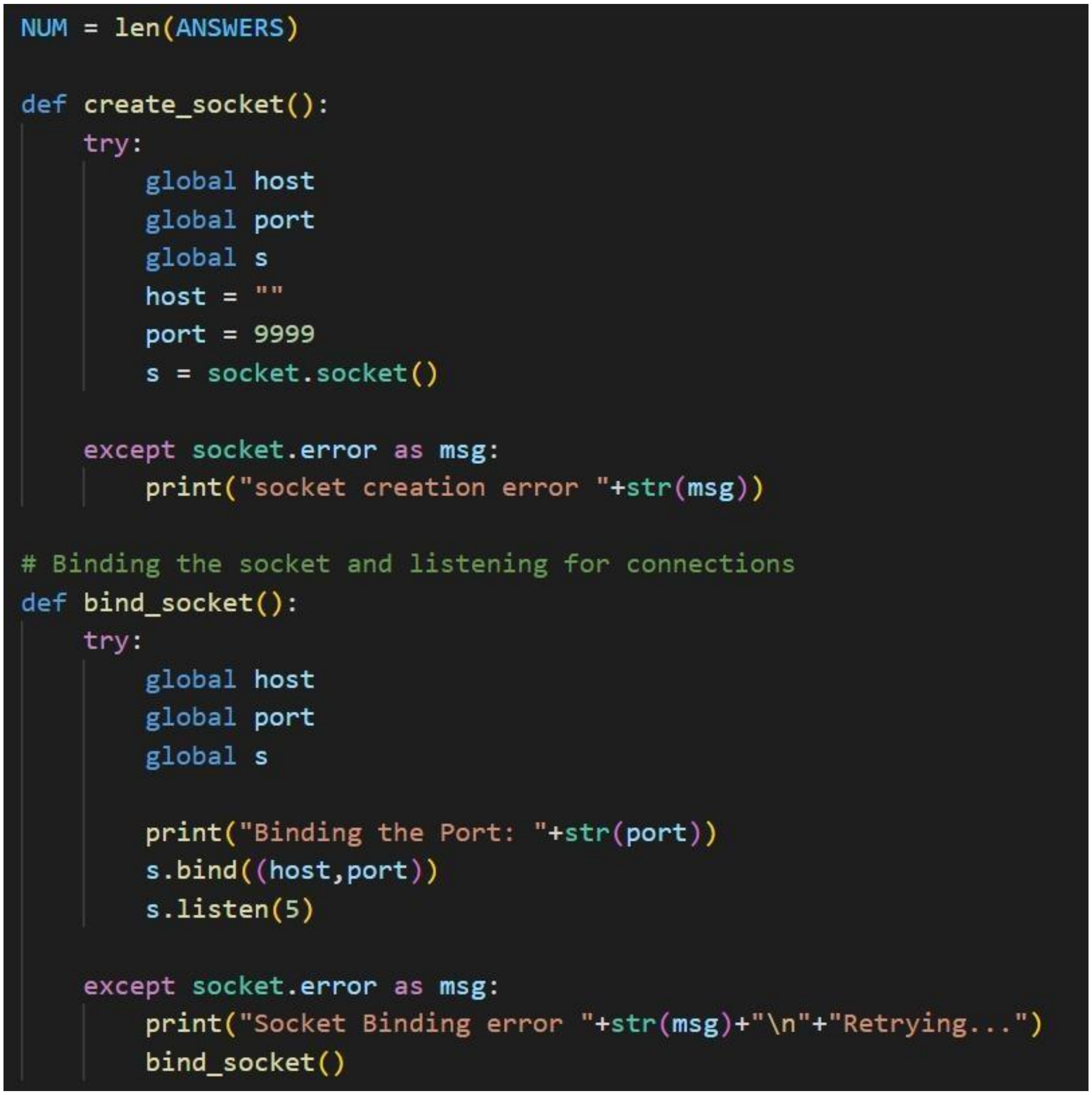


```python
# Handling connection from multiple clients and saving to a list
# Closing previous connections when server.py file is restarted

def accepting_connection():
    for c in all_connections:
        c.close()

    del all_connections[:]
    count = 0
    while(True):
        try:
            conn,address = s.accept()
            s.setblocking(1) #prevents time out
            all_connections.append(conn)
            all_address.append(address)

            print("Connection has been established :"+address[0])
            t = threading.Thread(target=Communication_client,args=(conn,address))
            t.daemon = True
            t.start()
        except:
            print("Error accepting connections")
```



```python
NUM = len(ANSWERS)

def create_socket():
    try:
        global host
        global port
        global s
        host = ""
        port = 9999
        s = socket.socket()

    except socket.error as msg:
        print("socket creation error "+str(msg))

# Binding the socket and listening for connections
def bind_socket():
    try:
        global host
        global port
        global s

        print("Binding the Port: "+str(port))
        s.bind((host,port))
        s.listen(5)

    except socket.error as msg:
        print("Socket Binding error "+str(msg)+"\n"+"Retrying...")
        bind_socket()
```

Client Side:

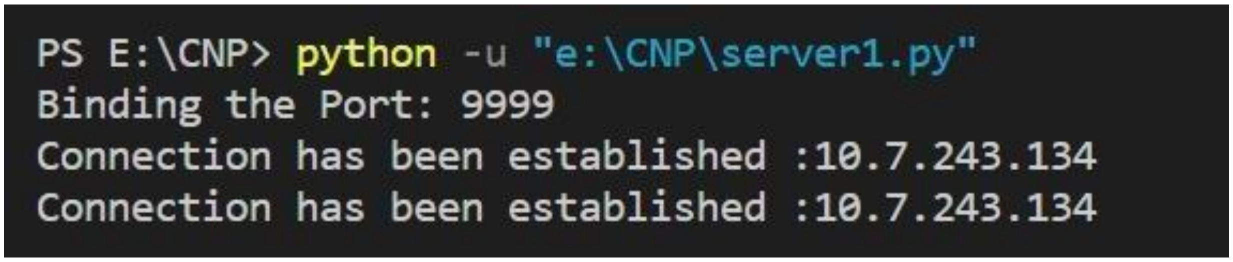


```
NUM = len(ANSWERS)

def create_socket():
    try:
        global host
        global port
        global s
        host = ""
        port = 9999
        s = socket.socket()

    except socket.error as msg:
        print("socket creation error "+str(msg))

# Binding the socket and listening for connections
def bind_socket():
    try:
        global host
        global port
        global s

        print("Binding the Port: "+str(port))
        s.bind((host,port))
        s.listen(5)

    except socket.error as msg:
        print("Socket Binding error "+str(msg)+"\n"+"Retrying...")
        bind_socket()
```

7. RESULTS AND DISCUSSION

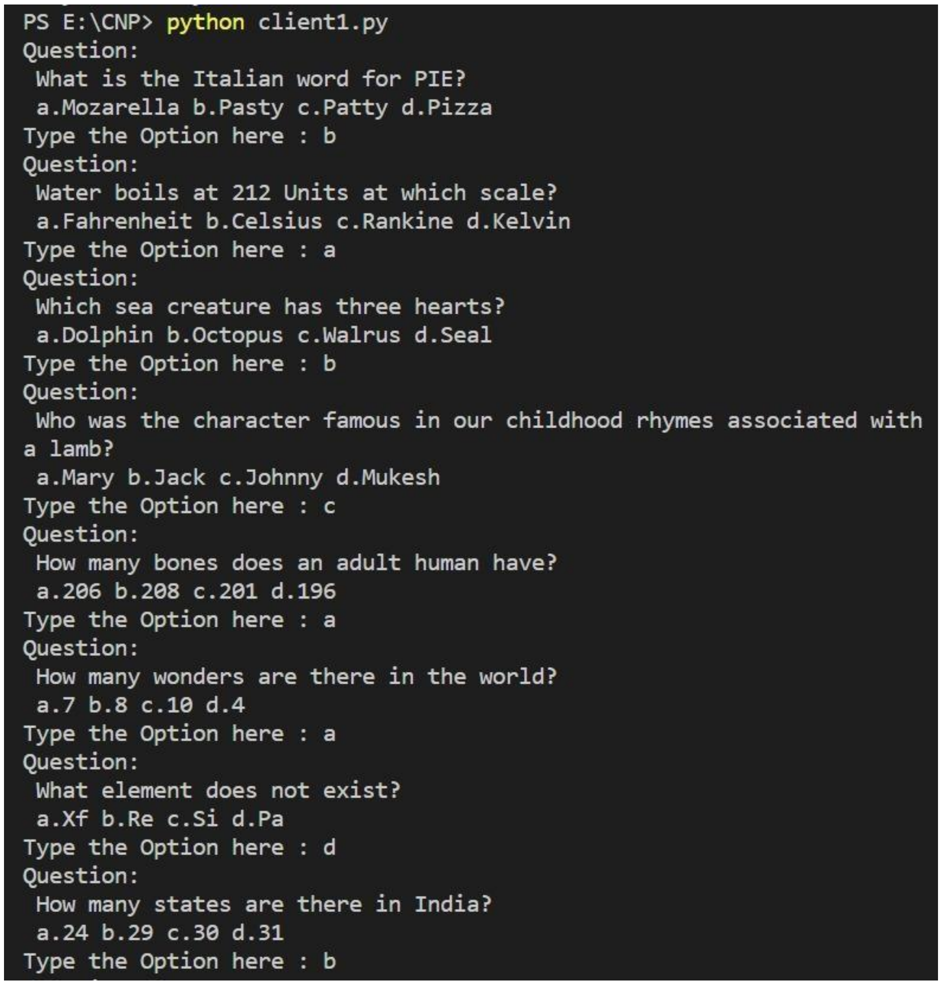
7.1 RESULTS:

Server Side Output:

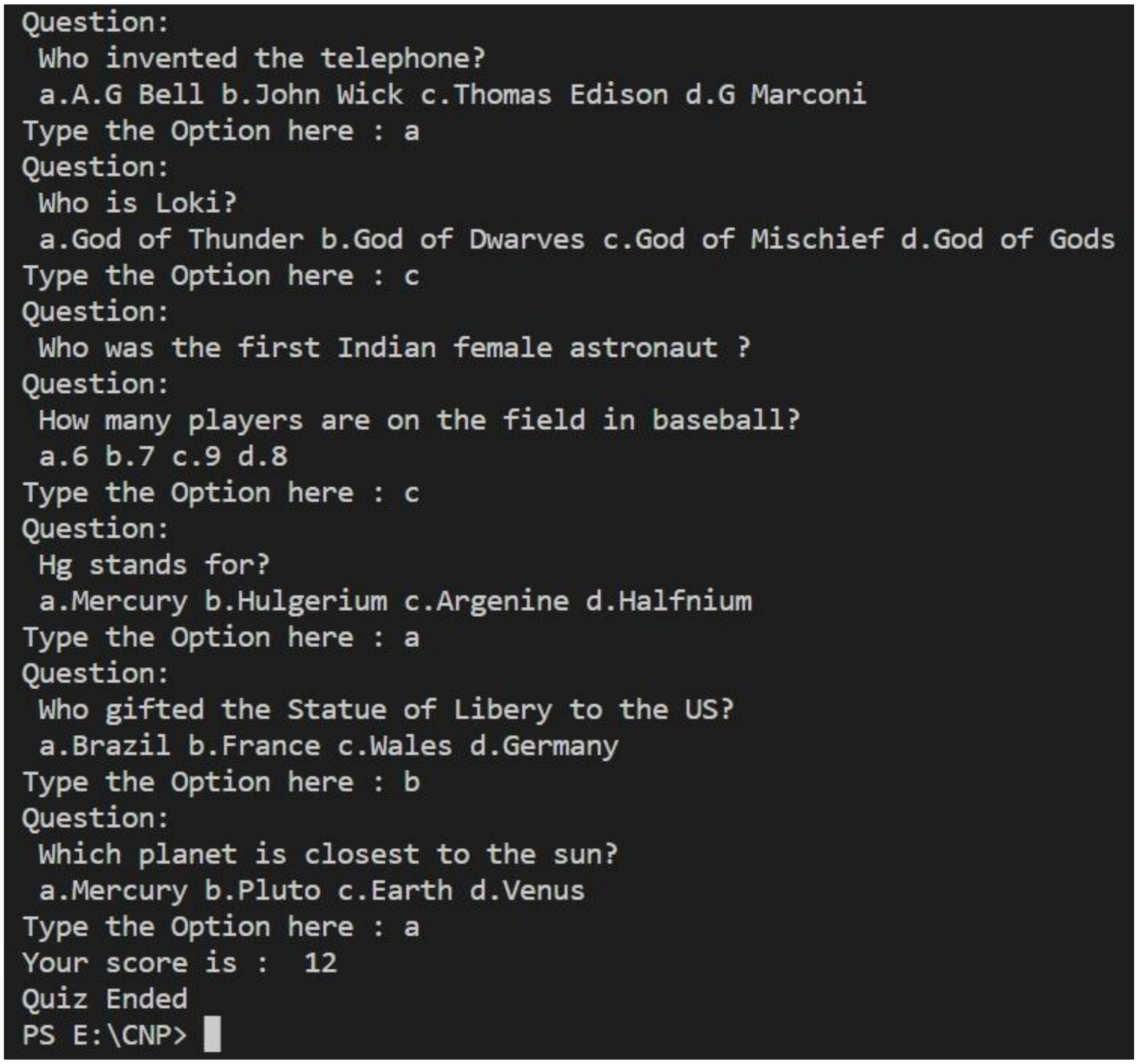


```
PS E:\CNP> python -u "e:\CNP\server1.py"
Binding the Port: 9999
Connection has been established :10.7.243.134
Connection has been established :10.7.243.134
```

Client Side Output



```
PS E:\CNP> python client1.py
Question:
 What is the Italian word for PIE?
 a.Mozarella b.Pasty c.Patty d.Pizza
Type the Option here : b
Question:
 Water boils at 212 Units at which scale?
 a.Fahrenheit b.Celsius c.Rankine d.Kelvin
Type the Option here : a
Question:
 Which sea creature has three hearts?
 a.Dolphin b.Octopus c.Walrus d.Seal
Type the Option here : b
Question:
 Who was the character famous in our childhood rhymes associated with
a lamb?
 a.Mary b.Jack c.Johnny d.Mukesh
Type the Option here : c
Question:
 How many bones does an adult human have?
 a.206 b.208 c.201 d.196
Type the Option here : a
Question:
 How many wonders are there in the world?
 a.7 b.8 c.10 d.4
Type the Option here : a
Question:
 What element does not exist?
 a.Xf b.Re c.Si d.Pa
Type the Option here : d
Question:
 How many states are there in India?
 a.24 b.29 c.30 d.31
Type the Option here : b
```



```
Question:
 Who invented the telephone?
 a.A.G Bell b.John Wick c.Thomas Edison d.G Marconi
Type the Option here : a
Question:
 Who is Loki?
 a.God of Thunder b.God of Dwarves c.God of Mischief d.God of Gods
Type the Option here : c
Question:
 Who was the first Indian female astronaut ?
Question:
 How many players are on the field in baseball?
 a.6 b.7 c.9 d.8
Type the Option here : c
Question:
 Hg stands for?
 a.Mercury b.Hulgerium c.Argenine d.Halfnium
Type the Option here : a
Question:
 Who gifted the Statue of Libery to the US?
 a.Brazil b.France c.Wales d.Germany
Type the Option here : b
Question:
 Which planet is closest to the sun?
 a.Mercury b.Pluto c.Earth d.Venus
Type the Option here : a
Your score is :  12
Quiz Ended
PS E:\CNP>
```

Server program runs on server side. Client connects to server. Then server sends question to client, and four options and prompts to answer the question. Then client answers the question and clicks enter then the option is sent towards server side. Then on server side server analyses the answer sent by client and awards either 0 or 1 mark. At the end of the test total marks scored by client is sent to client side and quiz is ended.

7.2 RESULT ANALYSIS:

As depicted through the above screenshots our model consists of a server and a client and can be effectively used to conduct quizzes as well as exams and can be further expanded for as many clients as possible by little modification. We have used TCP server connections to perform the task. The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

The following conclusions can be deduced from the development of the project:

i. Automation of the entire system improves the efficiency

ii. It provides a friendly graphical user interface which proves to be better when compared to the existing system.

iii. It gives appropriate access to the authorized users depending on their permissions.

iv. It effectively overcomes the delay in communications.

v. Updating of information becomes so easier.

vi. System security, data security and reliability are the striking features. From these points we can conclude that this could be an efficient system for online examination modes and as a result of its simplicity, can easily be built further on in multiple ways.

8. CONCLUSION AND FUTURE ENHANCEMENT

The following conclusions can be deduced from the development of the project:

i. Automation of the entire system improves the efficiency

ii. It provides a friendly graphical user interface which proves to be better when compared to the existing system.

iii. It gives appropriate access to the authorized users depending on their permissions.

iv. It effectively overcomes the delay in communications.

v. Updating of information becomes so easier.

vi. System security, data security and reliability are the striking features. From these points we can conclude that this could be an efficient system for online examination modes and as a result of its simplicity, can easily be built further on in multiple ways.

REFERENCES

Article: International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 4802-4807
Author: Limi Kalita

Article: The Socket Programming and Software Design for Communication Based on Client/Server
Author : Ming Xue.

Article: Socket Programming- Internet Networking Protocol
Author: Ajay Yadav

Article: Socket Programming for scalable systems.
Author: Steven Heines.

Article: Multiple Multi Instance Secured Distribution File server.
Author: Ajay.K.Sharma.

Article: Socket Programming in Python.
Author: Nathen Jennings

Article: Learn socket programming with Python.
Author: Ashish Kumar

Article: Socket Programming in python and how to master it.
Author: Suresh