

# PROJECT REPORT

## COMPUTER NETWORKS

### TOPIC:-

SIMPLE VERSION OF A GAME SHOW

### DESCRIPTION:-

The project is a game show. There is a host who conducts the show and participants/players who provide answers. Three participants are required to play the game. The host has a long list of questions and correct answers with him. He randomly chooses one of the questions (making sure it is not a repeat of previous questions) and sends to all three players. The players receive the question, think about the answer for a while and press the buzzer. There is a timer for 10 seconds for buzzer to be pressed. Otherwise, the host moves on to the next question. The first one to press the buzzer is given a chance to provide the answer within 10 seconds. If the answer is correct, he is given 1 point, otherwise -0.5. Nobody gets chance to answer this question again. The host then proceeds with the next question. The game stops when any player gets 5 points and that player is declared the winner. If the number of questions are completed the game is declared to be tied and the scores are displayed.

### PROBLEMS FACED:-

These are the basic problems I faced while coding the project:

- Connecting clients to server
- Sending and receiving information from client on the server side
- Sending and receiving information from server on the client side
- Creating questions
- Implementing the buzzer timing and checking who pressed the buzzer first.
- Receiving the answer in 10 seconds.
- Receiving answers, checking them and updating the scores.
- Ending the program and closing the connections without any bugs.

### CODING - SOLVING THE PROBLEMS :-

I have programmed the project using python. These are the ways I solved the problems:

- Learning the basic of sockets we can easily connect the server and client. We use the socket module in python to do the starting connections.
- Sending and receiving strings on both the side can be done using the functions recv and sendall in socket module.
- I have used a for loop and generated questions from 1+0==?? to 1+49==??

- Implementation of buzzer is done using the select module in python.  
First question that arise is what does select do??
  - It checks if there is any change in any of the sockets that are connected and reports it. We can use timeout feature in this to wait.
  - But what if more than one client press the buzzer? I have used the select function twice to solve this. I have called a select function that checks who pressed the buzzer first. Then to stop the server program for a while I have uses time module and sleep function. This waits for 10sec. Then I used the select function again which this time gives a list of all responded clients.

```
read_sockets, _, _ = select.select(conn_list, [], [], 10)
time.sleep(10)
read_sockets_all, _, _ = select.select(conn_list, [], [], 3)
```

- Checking the answer and updating the score can be done by using basic if conditions.
- I used the select function again to maintain the timing. It waits 10 seconds for any response from input. If nothing received it proceeds and send the reply to the server.

```
i, o, e = select.select([sys.stdin], [], [], 10)
answer=" "
if(i):
    answer=sys.stdin.readline().strip()
    print("Your ans:",answer)
else:
    print("You said nothing!")
```

- The program has two ways to end. If all the questions finished or if any player reaches score 5. These are done using the condition given in the loop.
- Finally the sockets are closed using conn.shutdown() in socket module.

## DEBUGGING:-

There where some bugs after completing the code. I fixed many of the bugs.

- Timing between all the clients was not correct. I used time.sleep() wherever necessary to fix it.
- Syntax errors were corrected.
- Buzzers also took much time to debug. Checked many cases to understand select function.
- If a client is answering and some other client presses any thing it is taken as buzzer to the next question. This bug is fixed by clearing the input stream using termos module.

```
termios.tcflush(sys.stdin, termios.TCIFLUSH)
```

## **INSTRUCTIONS TO RUN:-**

1. Open terminal and go to the directory in which the project files are present.
2. Run `python3 server.py`
3. Then open 3 other terminals in the same directory.
4. Run `python3 client .py` in all the three terminals and play the quiz.

## **EXPERIENCE:-**

I have learnt socket programming in python and got a clear idea about sockets. Apart from that I also learnt debugging the code and time management for submission of the project in time.

## **REFERENCES:-**

- <https://youtu.be/McoDjOCb2Zo>
- <https://github.com/eeshan9815/Realtime-Quiz-Server>
- Google
- Stack overflow