COMPUTER NETWORKS PROJECT

By – Agrim Jain(imt2019005)

I. INTRODUCTION

This project simulates a quiz with **server client setup**, using **socket programming** in

python. It uses the concept of **multithreading**.

A thread is sub process that runs a set of commands individually of other threads.

Every time a user connects to the server, a separate thread is created for that client and

communication from server to client takes place along individual threads.

Each client is identified by a unique socket object.

II. Language Used : Python 2.7

III. Modules Used : sys, select, time ,socket , threading

random , termios

IV . DESCRIPTION

- The server program waits until all 3 clients are connected and then only starts the game.
- Every client is assigned a number (For ex. 1st Client will has userid = 0) as soon as client-server connection is set up.
- Total questions = 50. All questions are unique.

- Marking Scheme :-
  - **+1** => Correct Answer.
  - **0** => No Answer.
  - **-0.5** => Wrong Answer.
- First one to press the buzzer (**ENTER KEY**) can answer. Others cannot press the buzzer after that and hence cannot answer.
- Clients have **10 seconds to press the buzzer**. The server proceeds to next question if no one pressed the buzzer within 10 seconds.
- After pressing the buzzer, the client has **10 seconds to answer**. The server proceeds to next question if the client does not answer within 10 seconds.
- Each player can see-
  - Player Who buzzed(or no one buzzed).
  - Given answer (ONLY for clients who do not pressed the buzzer first AND if answered) otherwise “No Answer Received” (FOR All clients).
  - Whether the given answer is correct or incorrect.
  - Correct answer.
  - Current Score.
- On the server’s terminal these are displayed:-
  - Player Who buzzed (if buzzed) otherwise no one buzzed.
  - Given answer (if answered) otherwise no answer received.
  - Whether the given answer is correct or incorrect.
  - Correct answer.

- The game stops when any player gets 5 points or all 50 questions are done, final score of all clients and whether that client wins/loses/all lose is declared for each client.

V. RUNNING THE PROJECT

- On Terminal 1 – “python server.py”.
- On Terminal 2 – “python client.py”.
- On Terminal 3 – “python client.py”.
- On Terminal 4 – “python client.py”.

VI. REFERENCES

- https://www.geeksforgeeks.org/socket-programming-multi-threading-python/
- https://www.youtube.com/watch?v=6jteAOmdsYg&list=PLhTjy8cBISErYuLZUvVOYsR1giva2payF
- https://www.youtube.com/watch?v=T0rYSFPAR0A