**COMPUTER NETWORKS LABORATORY**

MINI PROJECT

Submitted to:

Mrs. P.B. Shanthi | Department of CSE

November 10, 2016

ENQ – ENQ Not QuizUp

Online quizzing platform using concepts of socket programming

Aneesh Joshi | Kritik Mathur | Nimish Agrawal

# INTRODUCTION

What does ENQ stand for?

ENQ (pronounced as *Enque*) is a recursive acronym which stands for “**E**NQ **N**ot **Q**uizUp”.

What does ENQ do?

ENQ lets you quiz against other online users. It also gives you an option to spectate an ongoing quiz. It runs for multiple clients – meaning, the server can support multiple pairs of users playing at a time.

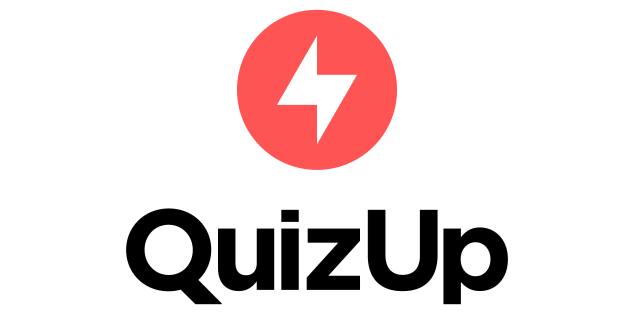
# PROBLEM DEFINITION

To provide a fast, entertaining and a robust service of a quizzing platform which can support multiple users at a time.

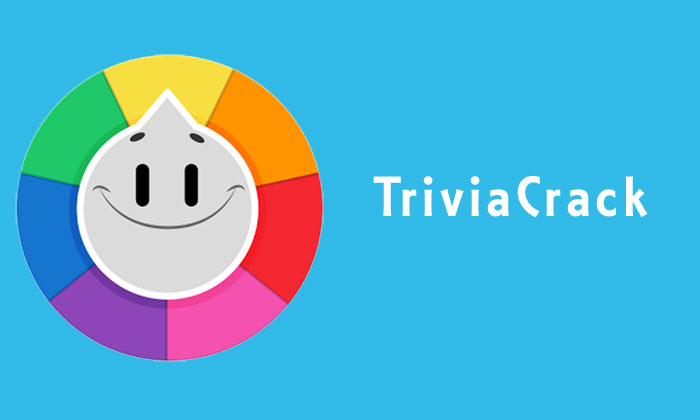
# OBJECTIVES

* To build a quizzing platform where the server can host multiple quizzes at a time.
* To calculate the scores after each question and display it to the players.
* To provide the users to enter into ‘Spectate Mode’, which allows the user to spectate a quiz without her directly participating in the quiz.
* Implementing the project by using concepts of socket programming in C++ and Linux Commands.
* Understanding Socket Programming concepts, basic concepts in Computer Networks and Linux Commands.

# OTHER EXISTING QUIZZING PLATFORMS







# BACKGROUND AND IMPLEMENTAION

The programming language used in our project is: **C++**

The concepts used in our project are:

* Sockets
* Multithreading
* Linux File Handling

**Sockets:** Sockets are communication points on the same or different computers to exchange data. Sockets allow communication between two different processes on the same or different machines. It's a way to talk to other computers using standard Unix file descriptors. A Unix Socket is used in a client-server application framework.

**Multithreading:** In simple words, multithreading is the ability of a central processing unit (CPU) or a single core in a multi-core processor to execute multiple processes or threads concurrently, appropriately supported by the operating system. This way it lets us make efficient use of multi-core processor systems.

**Linux File Handling:**  A Linux system, makes no difference between a file and a directory, since a directory is just a file containing names of other files. A file represents a sequence of bytes on the disk where a group of related data is stored. File is created for permanent storage of data. It is a ready-made structure. Thus, it is easy to implement file handing in the LINUX environment.

# METHODOLOGY

The following are the steps the user has to take to use the platform:

* **STEP 1:** The player chooses between *playing* or *spectating*.
* **STEP 2:** If she chooses *playing*, then the server finds another waiting player and sets them up for the game. If no other waiting player was found, then the player has to wait for a new player to join in.
* **STEP 3:** Once set up, the players try to get as many questions correct as possible in the least amount of time. The questions are stored on the server and are selected at random.
* **STEP 4**: At the end of the quiz, the user can either stay for another or choose to leave.
* **STEP 5**: If *spectating* option was chosen, then the user gets to choose a game from a list of ongoing games to spectate.

# SCORING

* The scoring system is based on *dynamic scoring*.
* This means, the earlier the correct answer is submitted for a question, more the number of points are awarded to the player.
* If both the players submit their answers at the same time, both are awarded equal number of points.
* Number of marks awarded for a correct answer = Time left (in seconds) to answer the question. The player gets 10 seconds to answer a question.
* An incorrect answer yields no points.

# LIMITATIONS

The limitations of the project:

* A maximum of five games can be hosted by the server.
* A second player is always needed to play the game. There is no provision for single player quizzing.
* This platform can be used by seasoned players only, as it lacks a robust UI.

# CONCLUSION

With this project, we see the implementation of a quizzing platform using the fundamentals of socket programming.

Working on this project was a great learning experience for all of us and it has definitely broadened our horizon in the field of networking.

# REFERENCES

1. [www.stackoverflow.com](http://www.stackoverflow.com/)
2. www.askubuntu.com
3. www.wikipedia.org
4. Computer Networks: A Top-Down Approach, Forouzan