CLIENT – SERVER CHAT APPLICATION

ABSTRACT

Client Server Architecture is a computing model in which the server hosts, delivers and manages most of the resources and services to be consumed by the client. This type of architecture has one or more client computers connected to a central server over a network. Client-server architecture is an architecture of a computer network in which many clients (remote processors) request and receive service from a centralized server (host computer) (CIO Wiki, 2020).

Chat refers to process of connecting or communicating between two or multiple devices over the network. Chat may be delivered through Text, Images, Audio or Video communication via the Network. This chat application uses he client server architecture model.

This chat application has basic two components, namely server and client. This chat application which is based on Transmission Control Protocol (TCP). TCP is connection oriented and reliable protocol.

PURPOSE

Purpose of this project was to practice and learn fundamental networking concepts like Client-Server Architecture, Protocols especially Transmission Control Protocol, Socket Programming, Multithreading and Python programming language for Networking.

DESIGN, BUILD AND TEST

Design - The system has one server and multiple clients. No more than three connections are allowed at a time. Connections are limited to three clients at a given time. Tkinter is used to create a graphical user interface for user interaction. There are various limitations with this application from User design limitations to security design.

BUILD – Python programming language is used to build this application (course requirement). Wide features, operations and functions provided by inbuilt python libraries were enough to make simple chat application. Client.py handles the client side of connection whereas server.py handles server-side function like listening. Server.py have three main functions – incoming\_connections, handle\_client and broadcast. Whereas client.py have two main function receive\_message and send\_message. Besides, GUI is implemented in Client side ofcourse.

TEST – Basic testing is done to see whether the messages are transmitted or not. To run multiple connections, client.py should be run in new terminals. Testing has been successful.