**“Project Report”**

**Chat Application using C++**

From, **17IT** **84,85,86,87,88**

1. **Introduction:**

Chatting or teleconferencing, is an effort to bring people and ideas together despite all the geographical barriers using some technical knowledge of Programming and networking.

The technology was available years before but accepted by civilians quite lately.

Our Project is an example of chat server. It consists of two applications, first is the client application and second is server application, both run on user pc’s local IP. To start chatting client should run it’s client application which will get himself connected to server application.

Our application can undergo private type chatting between server and client.

1. **Concepts used:**

In this client/server architecture, we used low level sockets in C++ on Windows.

Starting with sockets without the knowledge of which such client/server programs can’t be constructed,

A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.

So, a socket might represent a client or server.

The basic functionality used in application is from winsock2.h headers and there respective libraries.

In this application, client would connect to network (server) at some port number say 8888 and local IP address of 127.0.0.1 of same PC.

1. **Function definitions:**

First, we declared a WSADATA object, which will be actually used only once, at WSAStartup. This function just tells the computer we are going to use sockets. The first parameter, MAKEWORD(2,0) is for specifying the version of winsock we want to use, it depends of the include, here it is winsock2.h. The second parameter is just the address of the WSAData object we created.

We initialized the socket at this line:

server = socket(AF\_INET, SOCK\_STREAM, 0);

Here, first argument AF\_INET denotes that we’re going to use IPV4 address, for IPV6 we use AF\_INET6,

SOCK\_STREAM which opens a connection between two distant computers and allows them to communicate: this protocol is called TCP (Transmission Control Protocol),

Then we, assigned the port number and ip at which port is located.

Then, client connects with server using connect function

connect(server, (SOCKADDR \*)&addr, sizeof(addr));

Here, first argument is server socket, second one for socket address info.

Like connect function in client side, there is similar function at server side:

bind(server, (SOCKADDR \*)&serverAddr, sizeof(serverAddr));

In other to exchange messages between client and server, this function’s used

send(server, buffer, sizeof(buffer), 0)

recv(client, buffer, sizeof(buffer), 0)

First argument decides whom to send or receive, if client specified then message will be sent/receive to/from client.

Here, buffer is character array where message is stored.

listen(server, 0);

The starts listening to allow clients to connect.

accept(server, (SOCKADDR \*)&clientAddr, &clientAddrSize))

The accept function permits an incoming connection attempt on a socket.

We can disconnect the socket from the server with closesocket() and call WSACleanup() to tell the computer we won't use sockets anymore. Do not use WSACleanup() until you completely finished your work with sockets, otherwise you will have to call WSAStartup() again.

1. **Keeping and maintaining records.**

ALL the data(conversations) held between client and server will be restored when client connects with server in the network.

File handling concept of C++ is used to write data to file, conversation history to clients console.