

```

1  /* UDP Server */
2  #define NTDDI_VERSION NTDDI_VISTA
3  #define WINVER _WIN32_WINNT_VISTA
4  #define _WIN32_WINNT _WIN32_WINNT_VISTA
5
6  #include <winsock2.h>
7  #include <iostream>
8  #include <mysql.h>
9  #include <sstream>
10 #include <ws2tcpip.h>
11 #include <windows.h>
12
13 std::string mainServer = "--INSERT SERVER--";
14 std::string mainDbUser = "--INSERT USER--";
15 std::string mainDbPass = "--INSERT PASS--";
16
17 char * client;
18 std::string time;
19 std::string points;
20 std::string apples;
21 char ClientIP[100];
22 char buffer[256];
23
24 int x = 0;
25
26 MYSQL_RES* Perform_Query(MYSQL* connection, const char* query) {
27     if (mysql_query(connection, query)) {
28         std::cout << "MySQL query error : %s\n" << mysql_error(connection) << std::
endl;
29         exit(1);
30     }
31     return mysql_use_result(connection);
32 }
33
34 int main(){
35     WSADATA Form;
36     WORD Version = MAKEWORD(2,2);
37     SOCKET S;
38     sockaddr_in SocketData;
39     sockaddr_in ClientInformation;
40     int ClientLength = sizeof(ClientInformation);
41     int Instance = WSStartup(Version, &Form);
42     if(Instance!=0)
43         return 1;
44     S = socket(AF_INET,SOCK_DGRAM,0);
45     SocketData.sin_addr.S_un.S_addr = ADDR_ANY;
46     SocketData.sin_family = AF_INET;
47     SocketData.sin_port = htons(54000);
48     if ((bind(S,(sockaddr*)&SocketData,sizeof(SocketData)))==SOCKET_ERROR)
49         return 1;
50     ZeroMemory(&ClientInformation, sizeof(ClientInformation));
51     while(true){
52     ReturnToServerLoop:
53         ZeroMemory(buffer,256);
54         int bytes = recvfrom(S,buffer,256,0,(sockaddr*)&ClientInformation,&
ClientLength);
55         if(bytes==SOCKET_ERROR){
56             std::cout << "Error received from client: " << WSAGetLastError() << std
::endl;
57             continue;
58         }
59         ZeroMemory(ClientIP, 100);
60         InetNtop(AF_INET,&ClientInformation.sin_addr,ClientIP,100);
61         std::cout << "Information received from [" << ClientIP << "] >> " << buffer
<< std::endl;
62         if(bytes>0)

```

```

63         goto GetDataForProcessing;
64     }
65 GetDataForProcessing:
66     switch(x){
67         case 0:{
68             client = ClientIP;
69             time = std::string(buffer);
70             std::cout << "Client and Time variables set." << std::endl;
71             std::cout << "Time variable set to: " << time << std::endl;
72             x++;
73             break;
74         }
75         case 1:{
76             points = std::string(buffer);
77             std::cout << "Points variable set to: " << points << std::endl;
78             x++;
79             break;
80         }
81         case 2:{
82             apples = std::string(buffer);
83             std::cout << "Points variable set to: " << apples << std::endl;
84             x++;
85             break;
86         }
87     }
88     if(x<3)
89         goto ReturnToServerLoop;
90     else{
91         std::cout << "Sending all information for MySQL insertion." << std::endl;
92         goto InsertMySQL;
93     }
94 InsertMySQL:
95     if(x==3){
96         MYSQL *connect;
97         connect = mysql_init(NULL);
98         mysql_real_connect(connect, mainServer.c_str(), mainDbUser.c_str(),
mainDbPass.c_str(), "hegel_phenom" ,0,NULL,0);
99         std::stringstream ss;
100        ss << "INSERT INTO stats (client, time, points, apples) VALUES (" << client
<< ", " << time << ", " << points << ", " << apples << ")";
101        std::string Query = ss.str();
102        Perform_Query(connect,Query.c_str());
103        mysql_close (connect);
104        x = 0;
105        std::cout << "Information posted to MySQL database.\n" << std::endl;
106        goto ReturnToServerLoop;
107    }
108    closesocket(S);
109    WSACleanup();
110    return 0;
111 }

```