#include <stdio.h>

#include <stdbool.h>

#include <stdlib.h>

#include <string.h>

#include "cJSON.c"

#include <winsock2.h>

#define MAX 80

#define PORT 12345

#define SA struct sockaddr

int reza(int x){

if(x == 1){

int a;

printf("1 : Register\n2 : Login\n3 : Exit\n");

scanf("%d", &a);

if(a == 1) reza(2);

if(a == 2) reza(3);

if(a == 3) exit(0);

}

if(x == 2){

char username[100];

char password[100];

char space[] = ", ";

char reg[] = "register ";

int client\_socket, server\_socket;

struct sockaddr\_in servaddr, cli;

WORD wVersionRequested;

WSADATA wsaData;

int err;

// Use the MAKEWORD(lowbyte, highbyte) macro declared in Windef.h

wVersionRequested = MAKEWORD(2, 2);

err = WSAStartup(wVersionRequested, &wsaData);

if (err != 0) {

// Tell the user that we could not find a usable Winsock DLL.

printf("WSAStartup failed with error: %d\n", err);

return 1;

}

// Create and verify socket

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

if (client\_socket == -1) {

printf("Socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

// Assign IP and port

memset(&servaddr, 0, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

servaddr.sin\_port = htons(PORT);

// Connect the client socket to server socket

if (connect(client\_socket, (SA\*)&servaddr, sizeof(servaddr)) != 0) {

printf("Connection to the server failed...\n");

exit(0);

}

else

printf("Successfully connected to the server..\n");

printf("Enter Your User Name\n");

scanf("%s", username);

printf("Enter Your Password\n");

scanf("%s", password);

strcat(reg , username);

strcat(reg , ", ");

strcat(reg , password);

char finalreg [strlen(reg)];

strcpy(finalreg , reg);

printf("%s",finalreg);

send(client\_socket, finalreg, sizeof(finalreg), 0);

char response[100];

shutdown(client\_socket,SD\_SEND);

recv(client\_socket, response, sizeof(response), 0);

// Print buffer which contains the server message

printf("Server : %s", response);

shutdown(client\_socket,SD\_BOTH);

reza(1);

}

if(x == 3){

char username[100];

char password[100];

char space[2] = ", ";

char login[] = "login ";

int client\_socket, server\_socket;

struct sockaddr\_in servaddr, cli;

WORD wVersionRequested;

WSADATA wsaData;

int err;

// Use the MAKEWORD(lowbyte, highbyte) macro declared in Windef.h

wVersionRequested = MAKEWORD(2, 2);

err = WSAStartup(wVersionRequested, &wsaData);

if (err != 0) {

// Tell the user that we could not find a usable Winsock DLL.

printf("WSAStartup failed with error: %d\n", err);

return 1;

}

// Create and verify socket

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

if (client\_socket == -1) {

printf("Socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

// Assign IP and port

memset(&servaddr, 0, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

servaddr.sin\_port = htons(PORT);

// Connect the client socket to server socket

if (connect(client\_socket, (SA\*)&servaddr, sizeof(servaddr)) != 0) {

printf("Connection to the server failed...\n");

exit(0);

}

else

printf("Successfully connected to the server..\n");

printf("Enter Your User Name\n");

scanf("%s", username);

printf("Enter Your Password\n");

scanf("%s", password);

strcat(login , username);

strcat(login , space);

strcat(login , password);

char finallog [strlen(login)];

strcpy(finallog , login);

send(client\_socket, finallog, sizeof(finallog), 0);

memset(login, 0, sizeof(login));

// Read the message from server and copy it to buffer

shutdown(client\_socket,SD\_SEND);

recv(client\_socket, login, sizeof(login), 0);

// Print buffer which contains the server message

printf("Server : %s", login);

shutdown(client\_socket,SD\_BOTH);

reza(5);

}

if(x == 5){

int b;

printf("1 : Create Channel\n2 : Join Channel\n3 : Logout\n");

scanf("%d", &b);

if(b == 1) reza(6);

if(b == 2) reza(7);

if(b == 3) reza(8);

}

if(x == 6){

char channelname[100];

char token[100];

char space[2] = ", ";

char cch[] = "create channel ";

int client\_socket, server\_socket;

struct sockaddr\_in servaddr, cli;

WORD wVersionRequested;

WSADATA wsaData;

int err;

// Use the MAKEWORD(lowbyte, highbyte) macro declared in Windef.h

wVersionRequested = MAKEWORD(2, 2);

err = WSAStartup(wVersionRequested, &wsaData);

if (err != 0) {

// Tell the user that we could not find a usable Winsock DLL.

printf("WSAStartup failed with error: %d\n", err);

return 1;

}

// Create and verify socket

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

if (client\_socket == -1) {

printf("Socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

// Assign IP and port

memset(&servaddr, 0, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

servaddr.sin\_port = htons(PORT);

// Connect the client socket to server socket

if (connect(client\_socket, (SA\*)&servaddr, sizeof(servaddr)) != 0) {

printf("Connection to the server failed...\n");

exit(0);

}

else

printf("Successfully connected to the server..\n");

printf("Enter Your Channel Name\n");

scanf("%s", channelname);

printf("Enter Your Token\n");

scanf("%s", token);

strcat(cch , channelname);

strcat(cch , space);

strcat(cch , token);

char finalcch [strlen(cch)];

strcpy(finalcch , cch);

send(client\_socket, finalcch, sizeof(finalcch), 0);

memset(cch, 0, sizeof(cch));

// Read the message from server and copy it to buffer

shutdown(client\_socket,SD\_SEND);

recv(client\_socket, cch, sizeof(cch), 0);

// Print buffer which contains the server message

printf("Server : %s", cch);

shutdown(client\_socket,SD\_BOTH);

reza(9);

}

if(x == 7){

char channelname[100];

char token[100];

char space[2] = ", ";

char jch[] = "join channel ";

int client\_socket, server\_socket;

struct sockaddr\_in servaddr, cli;

WORD wVersionRequested;

WSADATA wsaData;

int err;

// Use the MAKEWORD(lowbyte, highbyte) macro declared in Windef.h

wVersionRequested = MAKEWORD(2, 2);

err = WSAStartup(wVersionRequested, &wsaData);

if (err != 0) {

// Tell the user that we could not find a usable Winsock DLL.

printf("WSAStartup failed with error: %d\n", err);

return 1;

}

// Create and verify socket

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

if (client\_socket == -1) {

printf("Socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

// Assign IP and port

memset(&servaddr, 0, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

servaddr.sin\_port = htons(PORT);

// Connect the client socket to server socket

if (connect(client\_socket, (SA\*)&servaddr, sizeof(servaddr)) != 0) {

printf("Connection to the server failed...\n");

exit(0);

}

else

printf("Successfully connected to the server..\n");

printf("Enter Your Channel Name\n");

scanf("%s", channelname);

printf("Enter Your Token\n");

scanf("%s", token);

strcat(jch , channelname);

strcat(jch , space);

strcat(jch , token);

char finaljch [strlen(jch)];

strcpy(finaljch , jch);

send(client\_socket, finaljch, sizeof(finaljch), 0);

memset(jch, 0, sizeof(jch));

// Read the message from server and copy it to buffer

shutdown(client\_socket,SD\_SEND);

recv(client\_socket, jch, sizeof(jch), 0);

// Print buffer which contains the server message

printf("Server : %s", jch);

shutdown(client\_socket,SD\_BOTH);

reza(9);

}

if(x == 8){

int client\_socket, server\_socket;

struct sockaddr\_in servaddr, cli;

WORD wVersionRequested;

WSADATA wsaData;

int err;

// Use the MAKEWORD(lowbyte, highbyte) macro declared in Windef.h

wVersionRequested = MAKEWORD(2, 2);

err = WSAStartup(wVersionRequested, &wsaData);

if (err != 0) {

// Tell the user that we could not find a usable Winsock DLL.

printf("WSAStartup failed with error: %d\n", err);

return 1;

}

// Create and verify socket

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

if (client\_socket == -1) {

printf("Socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

// Assign IP and port

memset(&servaddr, 0, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

servaddr.sin\_port = htons(PORT);

// Connect the client socket to server socket

if (connect(client\_socket, (SA\*)&servaddr, sizeof(servaddr)) != 0) {

printf("Connection to the server failed...\n");

exit(0);

}

else

printf("Successfully connected to the server..\n");

char lo = "logout ";

char token[100];

printf("Enter Your Token\n");

scanf("%s", token);

strcat(lo , token);

char finallo [strlen(lo)];

strcpy(finallo , lo);

send(client\_socket, finallo, sizeof(finallo), 0);

memset(lo, 0, sizeof(lo));

// Read the message from server and copy it to buffer

shutdown(client\_socket,SD\_SEND);

recv(client\_socket, lo, sizeof(lo), 0);

// Print buffer which contains the server message

printf("Server : %s", lo);

shutdown(client\_socket,SD\_BOTH);

reza(1);

}

if(x == 9){

int c;

printf("1 : Send Message\n2 : Refresh\n3 : Channel Members\n4 : Leave Channel\n");

scanf("%d", &c);

if(c == 1) reza(10);

if(c == 2) reza(11);

if(c == 3) reza(12);

if(c == 4) reza(13);

}

if(x == 10){

int client\_socket, server\_socket;

struct sockaddr\_in servaddr, cli;

WORD wVersionRequested;

WSADATA wsaData;

int err;

// Use the MAKEWORD(lowbyte, highbyte) macro declared in Windef.h

wVersionRequested = MAKEWORD(2, 2);

err = WSAStartup(wVersionRequested, &wsaData);

if (err != 0) {

// Tell the user that we could not find a usable Winsock DLL.

printf("WSAStartup failed with error: %d\n", err);

return 1;

}

// Create and verify socket

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

if (client\_socket == -1) {

printf("Socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

// Assign IP and port

memset(&servaddr, 0, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

servaddr.sin\_port = htons(PORT);

// Connect the client socket to server socket

if (connect(client\_socket, (SA\*)&servaddr, sizeof(servaddr)) != 0) {

printf("Connection to the server failed...\n");

exit(0);

}

else

printf("Successfully connected to the server..\n");

char snd = "send ";

char msg[400];

char token[100];

char space = ", ";

printf("Enter Your Message\n");

scanf("%s", msg);

printf("Enter Your Token\n");

scanf("%s", token);

strcpy(snd , msg);

strcpy(snd , space);

strcpy(snd , token);

char finalsnd [strlen(snd)];

strcpy(finalsnd , snd);

send(client\_socket, finalsnd, sizeof(snd), 0);

memset(send, 0, sizeof(send));

// Read the message from server and copy it to buffer

shutdown(client\_socket,SD\_SEND);

recv(client\_socket, snd, sizeof(snd), 0);

// Print buffer which contains the server message

printf("Server : %s", snd);

shutdown(client\_socket,SD\_BOTH);

reza(9);

}

if(x == 11){

int client\_socket, server\_socket;

struct sockaddr\_in servaddr, cli;

WORD wVersionRequested;

WSADATA wsaData;

int err;

// Use the MAKEWORD(lowbyte, highbyte) macro declared in Windef.h

wVersionRequested = MAKEWORD(2, 2);

err = WSAStartup(wVersionRequested, &wsaData);

if (err != 0) {

// Tell the user that we could not find a usable Winsock DLL.

printf("WSAStartup failed with error: %d\n", err);

return 1;

}

// Create and verify socket

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

if (client\_socket == -1) {

printf("Socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

// Assign IP and port

memset(&servaddr, 0, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

servaddr.sin\_port = htons(PORT);

// Connect the client socket to server socket

if (connect(client\_socket, (SA\*)&servaddr, sizeof(servaddr)) != 0) {

printf("Connection to the server failed...\n");

exit(0);

}

else

printf("Successfully connected to the server..\n");

char ref = "refresh ";

char token[100];

printf("Enter Your Token\n");

scanf("%s", token);

strcat(ref , token);

char finalref [strlen(ref)];

strcpy(finalref , ref);

send(client\_socket, finalref, sizeof(finalref), 0);

memset(ref, 0, sizeof(ref));

// Read the message from server and copy it to buffer

shutdown(client\_socket,SD\_SEND);

recv(client\_socket, ref, sizeof(ref), 0);

// Print buffer which contains the server message

printf("Server : %s", ref);

shutdown(client\_socket,SD\_BOTH);

reza(9);

}

if(x == 12){

int client\_socket, server\_socket;

struct sockaddr\_in servaddr, cli;

WORD wVersionRequested;

WSADATA wsaData;

int err;

// Use the MAKEWORD(lowbyte, highbyte) macro declared in Windef.h

wVersionRequested = MAKEWORD(2, 2);

err = WSAStartup(wVersionRequested, &wsaData);

if (err != 0) {

// Tell the user that we could not find a usable Winsock DLL.

printf("WSAStartup failed with error: %d\n", err);

return 1;

}

// Create and verify socket

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

if (client\_socket == -1) {

printf("Socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

// Assign IP and port

memset(&servaddr, 0, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

servaddr.sin\_port = htons(PORT);

// Connect the client socket to server socket

if (connect(client\_socket, (SA\*)&servaddr, sizeof(servaddr)) != 0) {

printf("Connection to the server failed...\n");

exit(0);

}

else

printf("Successfully connected to the server..\n");

char chm = "channel members ";

char token[100];

printf("Enter Your Token\n");

scanf("%s", token);

strcat(chm , token);

char finalchm [strlen(chm)];

strcpy(finalchm , chm);

send(client\_socket, finalchm, sizeof(finalchm), 0);

memset(chm, 0, sizeof(chm));

// Read the message from server and copy it to buffer

shutdown(client\_socket,SD\_SEND);

recv(client\_socket, chm, sizeof(chm), 0);

// Print buffer which contains the server message

printf("Server : %s", chm);

shutdown(client\_socket,SD\_BOTH);

reza(9);

}

if(x == 13){

int client\_socket, server\_socket;

struct sockaddr\_in servaddr, cli;

WORD wVersionRequested;

WSADATA wsaData;

int err;

// Use the MAKEWORD(lowbyte, highbyte) macro declared in Windef.h

wVersionRequested = MAKEWORD(2, 2);

err = WSAStartup(wVersionRequested, &wsaData);

if (err != 0) {

// Tell the user that we could not find a usable Winsock DLL.

printf("WSAStartup failed with error: %d\n", err);

return 1;

}

// Create and verify socket

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

if (client\_socket == -1) {

printf("Socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

// Assign IP and port

memset(&servaddr, 0, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

servaddr.sin\_port = htons(PORT);

// Connect the client socket to server socket

if (connect(client\_socket, (SA\*)&servaddr, sizeof(servaddr)) != 0) {

printf("Connection to the server failed...\n");

exit(0);

}

else

printf("Successfully connected to the server..\n");

char le = "leave ";

char token[100];

printf("Enter Your Token\n");

scanf("%s", token);

strcat(le , token);

char finalle [strlen(le)];

strcpy(finalle , le);

send(client\_socket, finalle, sizeof(finalle), 0);

memset(le, 0, sizeof(le));

// Read the message from server and copy it to buffer

shutdown(client\_socket,SD\_SEND);

recv(client\_socket, le, sizeof(le), 0);

// Print buffer which contains the server message

printf("Server : %s", le);

shutdown(client\_socket,SD\_BOTH);

reza(5);

}

}

int main(){

reza(1);

}