AIMEN NADEEM

TASK 4:

PROGRAM:

#include <iostream>

#include <unistd.h>

#include <netinet/in.h>

#include <sstream>

#include <vector>

#include <cstring>

#include <fstream>

#include <thread>

using namespace std;

int PORT = 8080;

void handleclient(int client\_socket)

{

char buffer[1024] = { 0 };

read(client\_socket, buffer, 1024);

istringstream request(buffer);

string method, path, version;

request >> method >> path >> version;

if (path[0] == '/')

{

path = path.substr(1);

}

if (path.empty())

{

path = "index.html";

}

ifstream file(path);

if (file)

{

stringstream response;

response << "HTTP/1.1 200 OK\nContent-Type: text/html\n\n" << file.rdbuf();

string response\_str = response.str();

send(client\_socket, response\_str.c\_str(), response\_str.length(), 0);

}

else

{

const char\* not\_found = "HTTP/1.1 404 Not Found\nContent-Type: text/html\nContent-length:9\n\nNot Found";

send(client\_socket, not\_found, strlen(not\_found), 0);

}

close(client\_socket);

}

int main()

{

int server\_fd, client\_socket;

struct sockaddr\_in address;

int opt = 1;

int addrlen = sizeof(address);

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

{

perror("socket failed");

exit(EXIT\_FAILURE);

}

if (setsockopt(server\_fd, SOL\_SOCKET, SO\_REUSEADDR | SO\_REUSEPORT, &opt, sizeof(opt)))

{

perror("setsockopt");

exit(EXIT\_FAILURE);

}

address.sin\_family = AF\_INET;

address.sin\_addr.s\_addr = INADDR\_ANY;

address.sin\_port = htons(PORT);

if (bind(server\_fd, (struct sockaddr\*)&address, sizeof(address)) < 0)

{

perror("bind failed");

exit(EXIT\_FAILURE);

}

if (listen(server\_fd, 3) < 0)

{

perror("listen");

exit(EXIT\_FAILURE);

}

cout << "Server listening on port" << PORT << endl;

vector<thread>threads;

while (true)

{

if ((client\_socket = accept(server\_fd, (struct sockaddr\*)&address, (socklen\_t\*)&addrlen)) < 0)

{

perror("accept");

exit(EXIT\_FAILURE);

}

threads.push\_back(thread(handleclient, client\_socket));

}

for (auto& t : threads)

{

if (t.joinable())

{

t.join();

}

}

close(server\_fd);

return 0;

}