

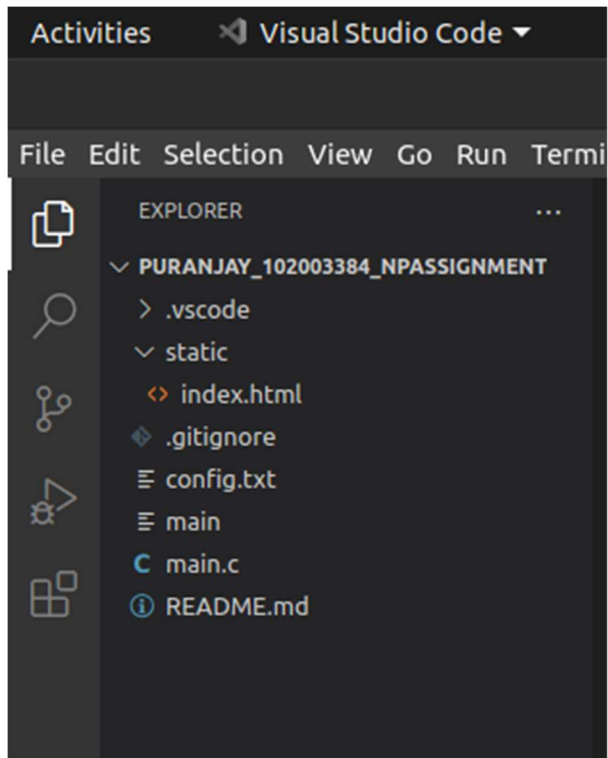
1. Roll number: 102003384

2. Name: Puranjay Singh

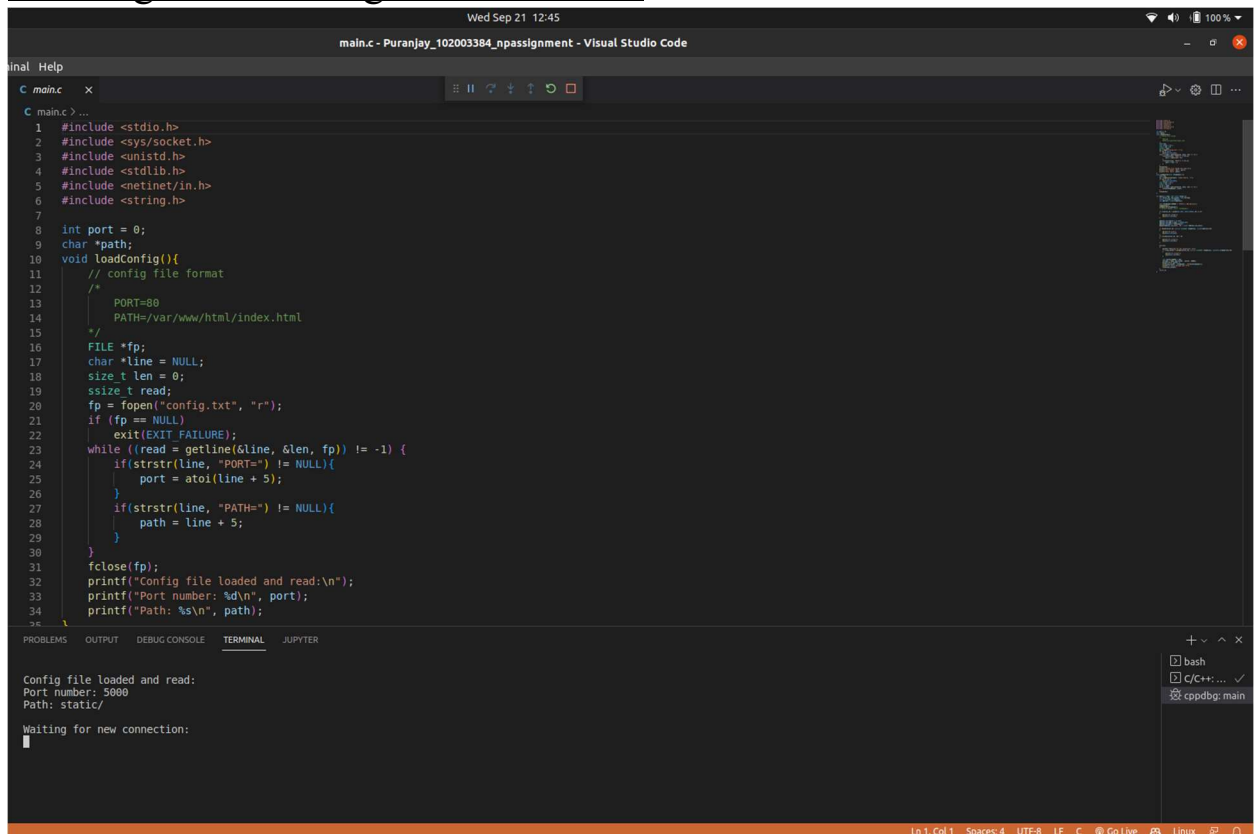
3. Group No: 3CO9

4. Output Screenshots:

a. Directory structure of the server



b. Running code waiting for connection



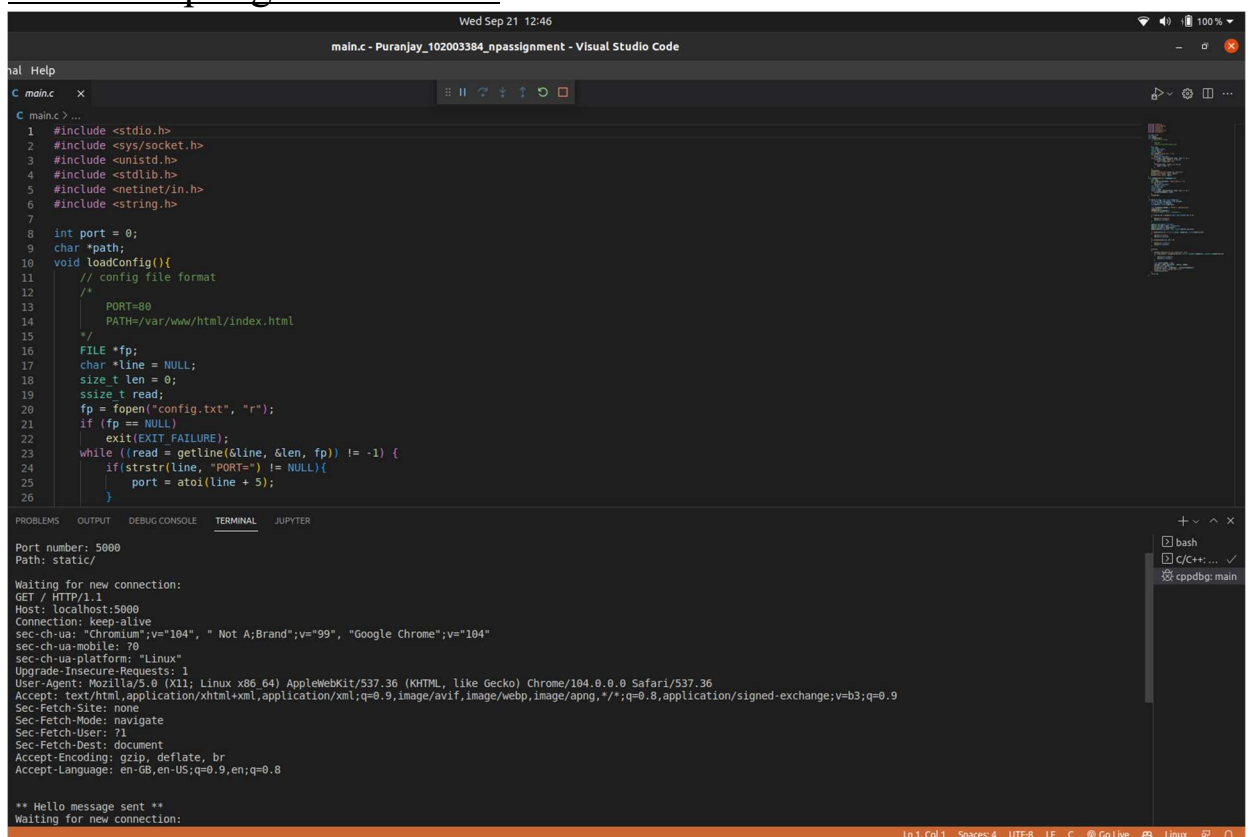
The screenshot shows the Visual Studio Code editor with a C program named `main.c`. The program includes headers for `stdio.h`, `sys/socket.h`, `unistd.h`, `stdlib.h`, `netinet/in.h`, and `string.h`. It defines a port (0) and a path. A `loadConfig()` function is defined to read a config file. The main function calls `loadConfig()` and then enters a loop waiting for a new connection. The terminal output shows the config file loaded and read, with port number 9000 and path static/. The program is currently waiting for a new connection.

```
1 #include <stdio.h>
2 #include <sys/socket.h>
3 #include <unistd.h>
4 #include <stdlib.h>
5 #include <netinet/in.h>
6 #include <string.h>
7
8 int port = 0;
9 char *path;
10 void loadConfig(){
11     // config file format
12     /*
13      PORT=80
14      PATH=/var/www/html/index.html
15     */
16     FILE *fp;
17     char *line = NULL;
18     size_t len = 0;
19     ssize_t read;
20     fp = fopen("config.txt", "r");
21     if (fp == NULL)
22         exit(EXIT_FAILURE);
23     while ((read = getline(&line, &len, fp)) != -1) {
24         if (strstr(line, "PORT=") != NULL) {
25             port = atoi(line + 5);
26         }
27         if (strstr(line, "PATH=") != NULL) {
28             path = line + 5;
29         }
30     }
31     fclose(fp);
32     printf("Config file loaded and read:\n");
33     printf("Port number: %d\n", port);
34     printf("Path: %s\n", path);
35 }
36
37 int main() {
38     loadConfig();
39     printf("Waiting for new connection:\n");
40     while (1) {
41         // ... (code for accepting connection) ...
42     }
43 }
```

Terminal Output:

```
Config file loaded and read:
Port number: 9000
Path: static/
Waiting for new connection:
```

c. After accepting a connection

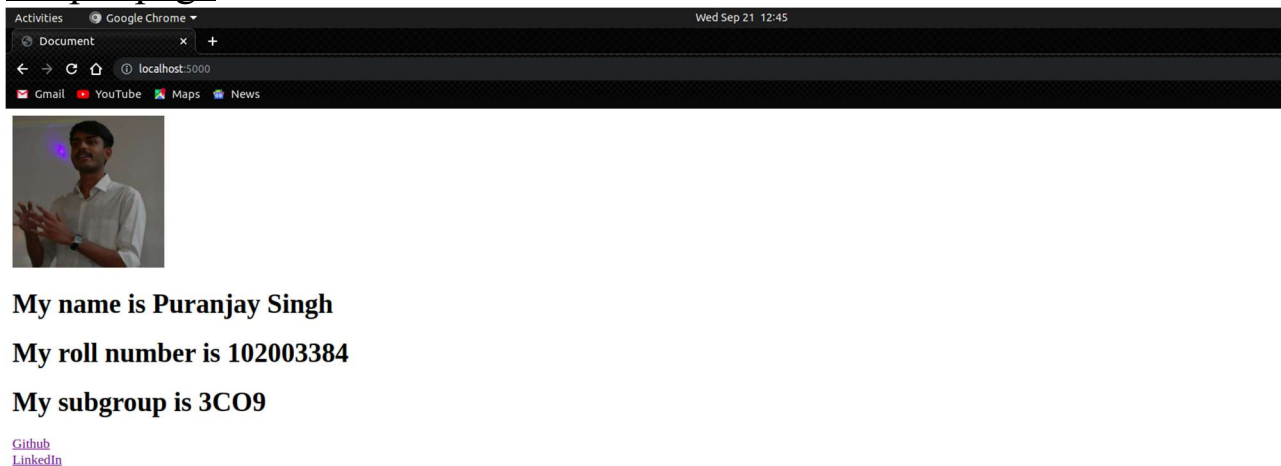


The screenshot shows the Visual Studio Code editor with the same C program. The terminal output now shows the program has accepted a connection from `Host: localhost:5000`. It displays the HTTP request details, including the `GET / HTTP/1.1` request, connection type, user agent, and headers. The program has sent a "Hello message" and is now waiting for a new connection.

```
Port number: 5000
Path: static/
Waiting for new connection:
GET / HTTP/1.1
Host: localhost:5000
Connection: keep-alive
sec-ch-ua: "Chromium";v="104", " Not A;Brand";v="99", "Google Chrome";v="104"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8

** Hello message sent **
Waiting for new connection:
```

d. Output page



5. Running code:

a. main.c(server code)

```
#include <stdio.h>
#include <sys/socket.h>
#include <unistd.h>
#include <stdlib.h>
#include <netinet/in.h>
#include <string.h>
```

```
int port = 0;
char *path;
void loadConfig(){
    // config file format
    /*
        PORT=80
        PATH=/var/www/html/index.html
    */
    FILE *fp;
    char *line = NULL;
    size_t len = 0;
```

```

    ssize_t read;
    fp = fopen("config.txt", "r");
    if (fp == NULL)
        exit(EXIT_FAILURE);
    while ((read = getline(&line, &len, fp)) != -1) {
        if(strstr(line, "PORT=") != NULL){
            port = atoi(line + 5);
        }
        if(strstr(line, "PATH=") != NULL){
            path = line + 5;
        }
    }
    fclose(fp);
    printf("Config file loaded and read:\n");
    printf("Port number: %d\n", port);
    printf("Path: %s\n", path);
}

void setUpServer(char httpHeader[]){
    FILE *fp;
    fp = fopen(strcat(path, "index.html"), "r");
    if (fp == NULL)
        exit(EXIT_FAILURE);
    char *line = NULL;
    size_t len = 0;
    ssize_t read;
    while ((read = getline(&line, &len, fp)) != -1) {
        strcat(httpHeader, line);
    }
    fclose(fp);
}

int main(int argc, char const *argv[]){
    int server_fd, new_socket; long valread;
    struct sockaddr_in address;
    int addrlen = sizeof(address);

```

```

char httpHeader[50000] = "HTTP/1.1 200 OK\r\n\n";
loadConfig();
setUpServer(httpHeader);
// printf("Header: %s\n", httpHeader);

if ((server_fd = socket(AF_INET, SOCK_STREAM, 0)) ==
0)
{
    perror("In socket");
    exit(EXIT_FAILURE);
}

address.sin_family = AF_INET;
address.sin_addr.s_addr = INADDR_ANY;
address.sin_port = htons(port);
memset(address.sin_zero, '\0', sizeof address.sin_zero);

if (bind(server_fd, (struct sockaddr *)&address,
sizeof(address))<0)
{
    perror("In bind");
    exit(EXIT_FAILURE);
}
if (listen(server_fd, 10) < 0)
{
    perror("In listen");
    exit(EXIT_FAILURE);
}

while(1)
{
    printf("\nWaiting for new connection: \n");
    if ((new_socket = accept(server_fd, (struct sockaddr
*)&address, (socklen_t*)&addrlen))<0)

```

```

    {
        perror("In accept");
        exit(EXIT_FAILURE);
    }

    char buffer[30000] = {0};
    valread = read( new_socket , buffer, 30000);
    printf("%s\n",buffer );
    write(new_socket , httpHeader , strlen(httpHeader));
    printf("** Hello message sent **");
    close(new_socket);
}
return 0;
}

```

b. config file

```

PORT=5000
PATH=static/

```

c. index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <a href="https://ibb.co/LrpZkqZ"></a>

```

```
<br>
<h1>My name is Puranjay Singh</h1>
<h1>My roll number is 102003384</h1>
<h1>My subgroup is 3CO9</h1>
<a href="https://github.com/purjaysin">Github</a>
<br>
<a
href="https://www.linkedin.com/in/singhpuranjay/">LinkedIn</
a>
</body>
</html>
```