SECTION X OS LAB 2024

GROUP 1 – RAILWAY RESERVATION SYSTEM

Important Note: Follow the instructions below to successfully run the project

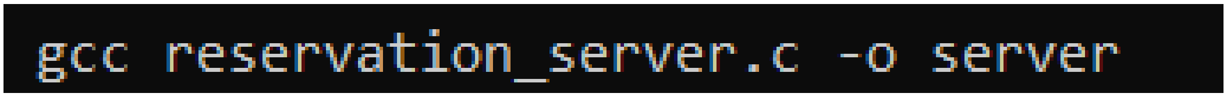
All the code files need to be present on both the server and client machines in order to run the login file functions

1. Clone the repository

https://docs.github.com/en/repositories/creating-and-managing-repositories/cloning-a-repository#cloning-a-repository

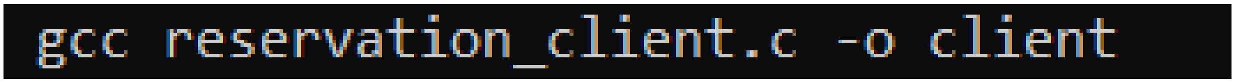
2. Compile each of the files in the terminal in the following order (It is important to compile the server and client files before running the login file):

- ✓ gcc reservation_server.c -o server



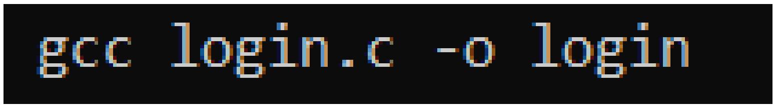
```
gcc reservation_server.c -o server
```

- ✓ gcc reservation_client.c -o client



```
gcc reservation_client.c -o client
```

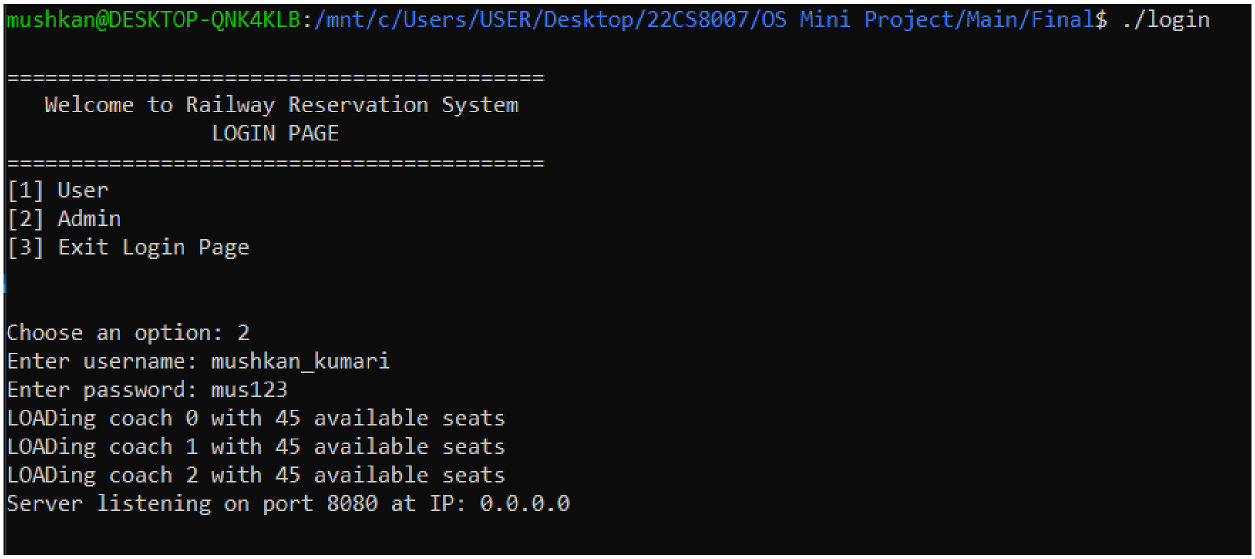
- ✓ gcc login.c -o login



```
gcc login.c -o login
```

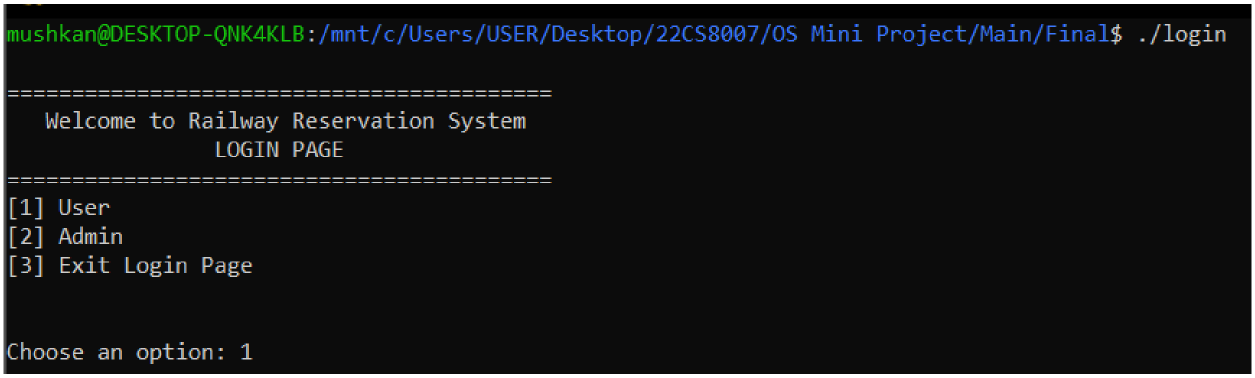
2. Execute the login file by the following command: **./login**

In server terminal:



```
mushkan@DESKTOP-QNK4KLB:/mnt/c/Users/USER/Desktop/22CS8007/OS Mini Project/Main/Final$ ./login

==========================================
   Welcome to Railway Reservation System
                LOGIN PAGE
==========================================
[1] User
[2] Admin
[3] Exit Login Page


Choose an option: 2
Enter username: mushkan_kumari
Enter password: mus123
LOADing coach 0 with 45 available seats
LOADing coach 1 with 45 available seats
LOADing coach 2 with 45 available seats
Server listening on port 8080 at IP: 0.0.0.0
```

In client terminal:



```
mushkan@DESKTOP-QNK4KLB:/mnt/c/Users/USER/Desktop/22CS8007/OS Mini Project/Main/Final$ ./login

==========================================
   Welcome to Railway Reservation System
                LOGIN PAGE
==========================================
[1] User
[2] Admin
[3] Exit Login Page


Choose an option: 1
```

❖ First time starting the server:

Admin username: mushkan_kumari

Admin_password: mus123

(Can be changed later on by editing the users.txt file)

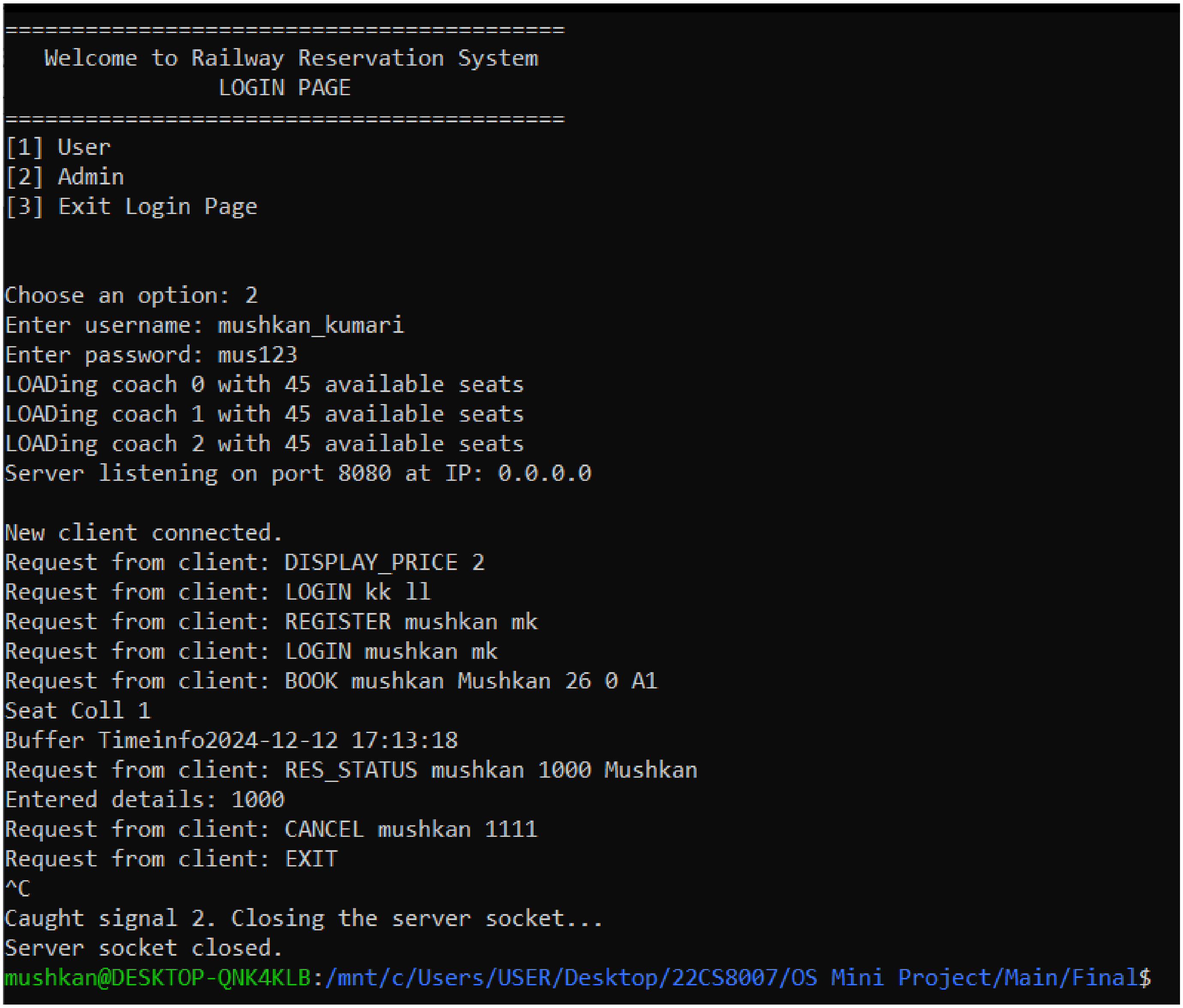
Make sure to uncomment the init_file() line in the main function of the server code . This will initialize the seat.txt file for a fresh run.

In case of any errors, use the sample seat.txt file provided in the repo

❖ Subsequent runs of the server:

Comment the init_file() function in the main function of the server code.

Sample Server View:



```
==========================================
   Welcome to Railway Reservation System
                LOGIN PAGE
==========================================
[1] User
[2] Admin
[3] Exit Login Page


Choose an option: 2
Enter username: mushkan_kumari
Enter password: mus123
LOADing coach 0 with 45 available seats
LOADing coach 1 with 45 available seats
LOADing coach 2 with 45 available seats
Server listening on port 8080 at IP: 0.0.0.0

New client connected.
Request from client: DISPLAY_PRICE 2
Request from client: LOGIN kk ll
Request from client: REGISTER mushkan mk
Request from client: LOGIN mushkan mk
Request from client: BOOK mushkan Mushkan 26 0 A1
Seat Coll 1
Buffer Timeinfo2024-12-12 17:13:18
Request from client: RES_STATUS mushkan 1000 Mushkan
Entered details: 1000
Request from client: CANCEL mushkan 1111
Request from client: EXIT
^C
Caught signal 2. Closing the server socket...
Server socket closed.
mushkan@DESKTOP-QNK4KLB:/mnt/c/Users/USER/Desktop/22CS8007/OS Mini Project/Main/Final$
```

Provided code will work for server and client(s) running on the same machine.

In order to run it from different machines on the same network, follow the steps below:

To run of different machines on the same network:

(These are the steps to be followed if the server is running on WSL):

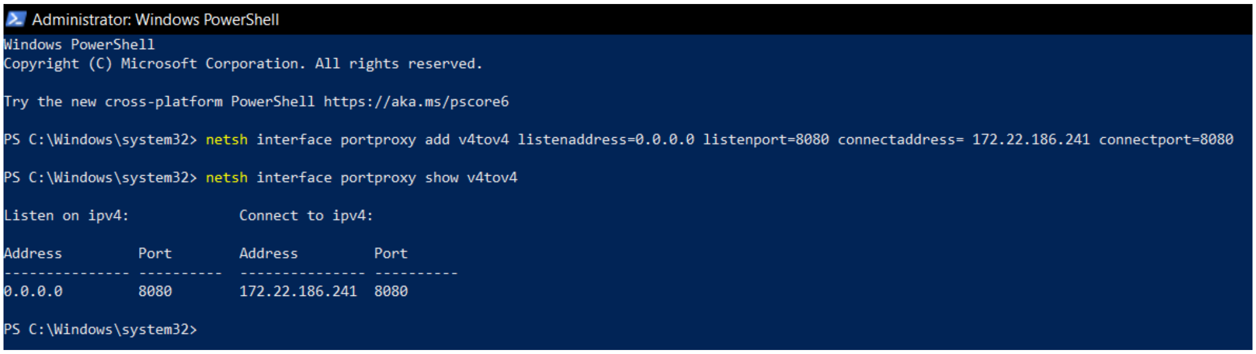
➔ Enable port forwarding and disable firewall

✓ For port forwarding:

Run on Windows Powershell as admin:

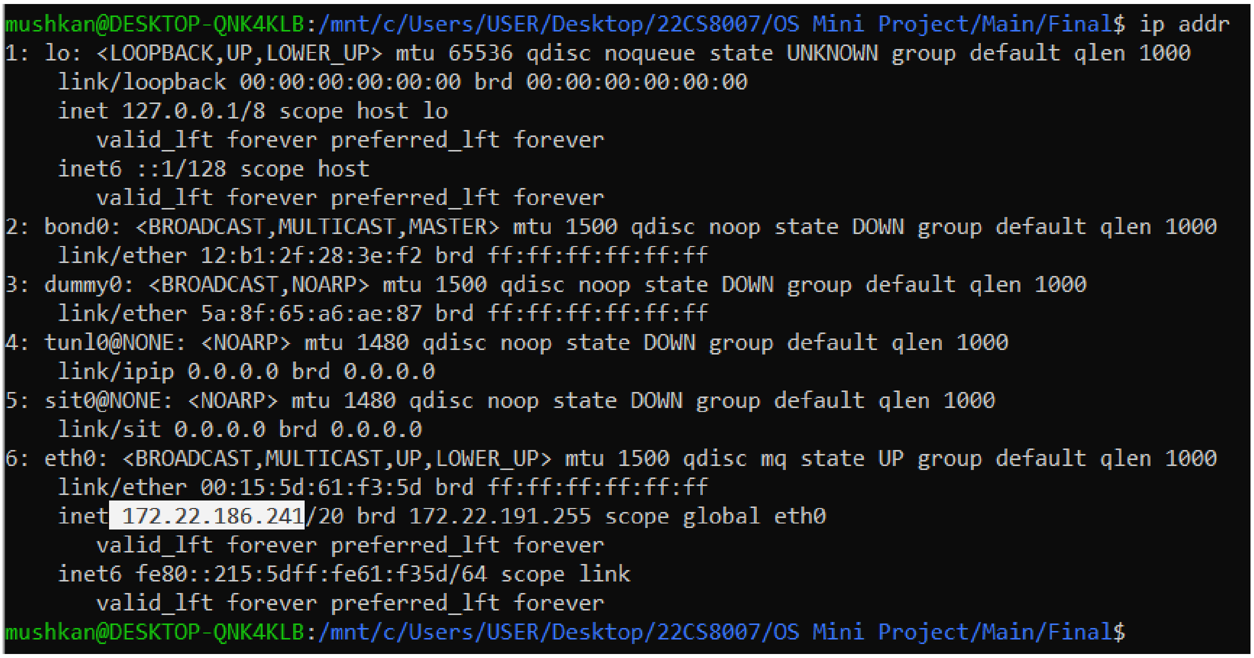
netsh interface portproxy add v4tov4 listenaddress=0.0.0.0 listenport=8080 connectaddress=<server ip> connectport=8080

(Replace with actual IP address of your server instance, which can be found out by typing "ip addr" in the WSL instance where the server will run)



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> netsh interface portproxy add v4tov4 listenaddress=0.0.0.0 listenport=8080 connectaddress= 172.22.186.241 connectport=8080

PS C:\Windows\system32> netsh interface portproxy show v4tov4

Listen on ipv4:             Connect to ipv4:

Address         Port        Address         Port
--------------- ----------  --------------- ----------
0.0.0.0         8080        172.22.186.241  8080

PS C:\Windows\system32>
```

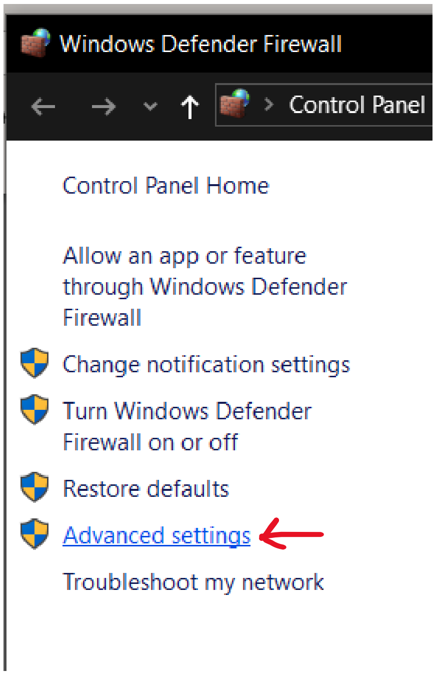
Example: The highlighted portion below is the IP address of the WSL instance where the server will run



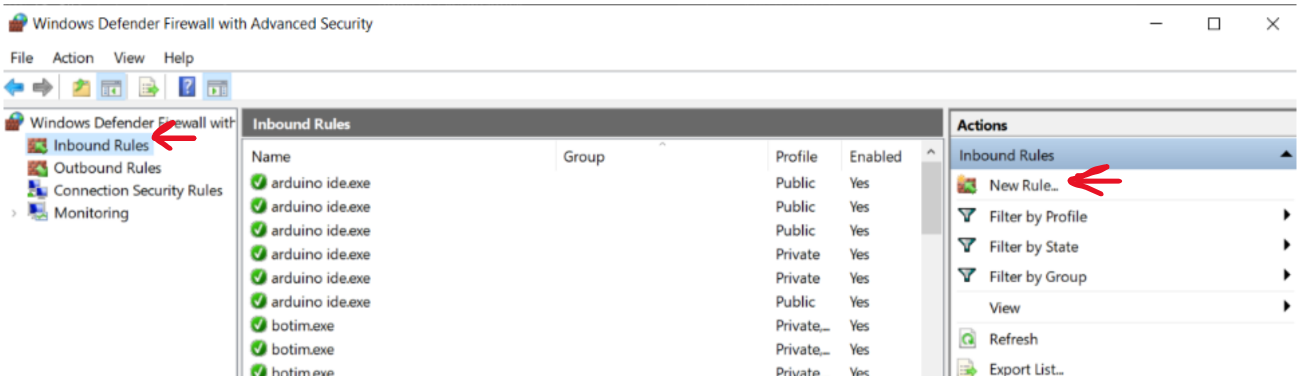
```
mushkan@DESKTOP-QNK4KLB:/mnt/c/Users/USER/Desktop/22CS8007/OS Mini Project/Main/Final$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 12:b1:2f:28:3e:f2 brd ff:ff:ff:ff:ff:ff
3: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 5a:8f:65:a6:ae:87 brd ff:ff:ff:ff:ff:ff
4: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
5: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
6: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:61:f3:5d brd ff:ff:ff:ff:ff:ff
    inet 172.22.186.241/20 brd 172.22.191.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe61:f35d/64 scope link
       valid_lft forever preferred_lft forever
mushkan@DESKTOP-QNK4KLB:/mnt/c/Users/USER/Desktop/22CS8007/OS Mini Project/Main/Final$
```

➔ To disable Firewall:

- Open Windows Defender Firewall. Open Advanced Settings.


Windows Defender Firewall
Control Panel
Control Panel Home
Allow an app or feature through Windows Defender Firewall
Change notification settings
Turn Windows Defender Firewall on or off
Restore defaults
Advanced settings
Troubleshoot my network

- Create a New Inbound Rule.



Name	Group	Profile	Enabled
arduino ide.exe		Public	Yes
arduino ide.exe		Public	Yes
arduino ide.exe		Public	Yes
arduino ide.exe		Private	Yes
arduino ide.exe		Private	Yes
arduino ide.exe		Public	Yes
botim.exe		Private,...	Yes
botim.exe		Private,...	Yes
botim.exe		Private	Yes

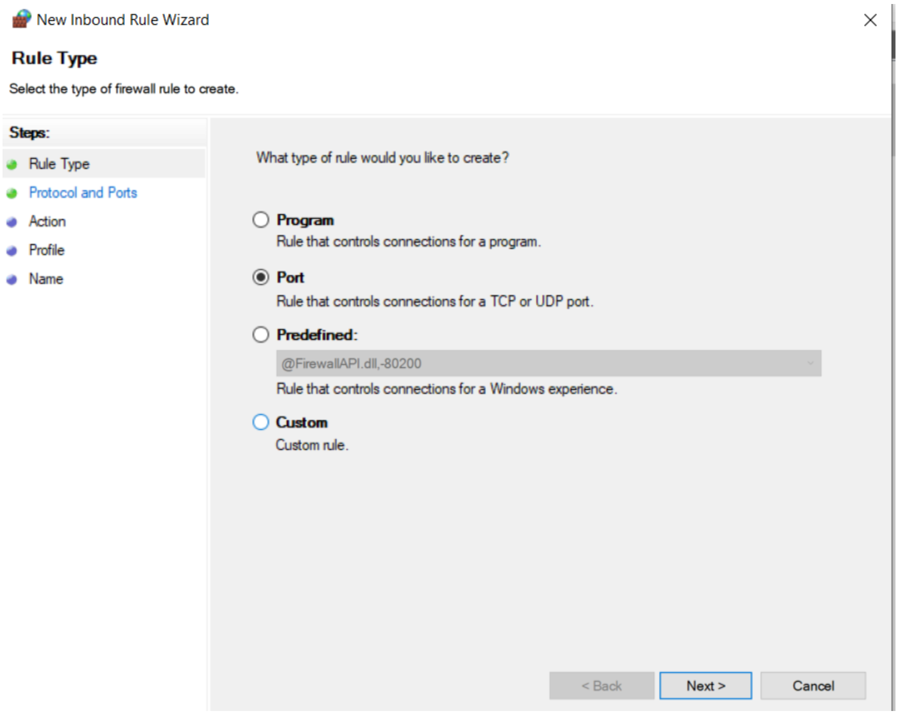
-> Select Rule Type: Specify Port: Choose TCP (or TCP/UDP if necessary).

-> Select Specific local ports and enter the port number you want to open (e.g., 8080).

-> Choose Allow the connection and click Next.

-> Profile Selection: Select when this rule applies:

(You can select all profiles if unsure, but make sure to at least select Private for most local network connections)



Rule Type

Select the type of firewall rule to create.

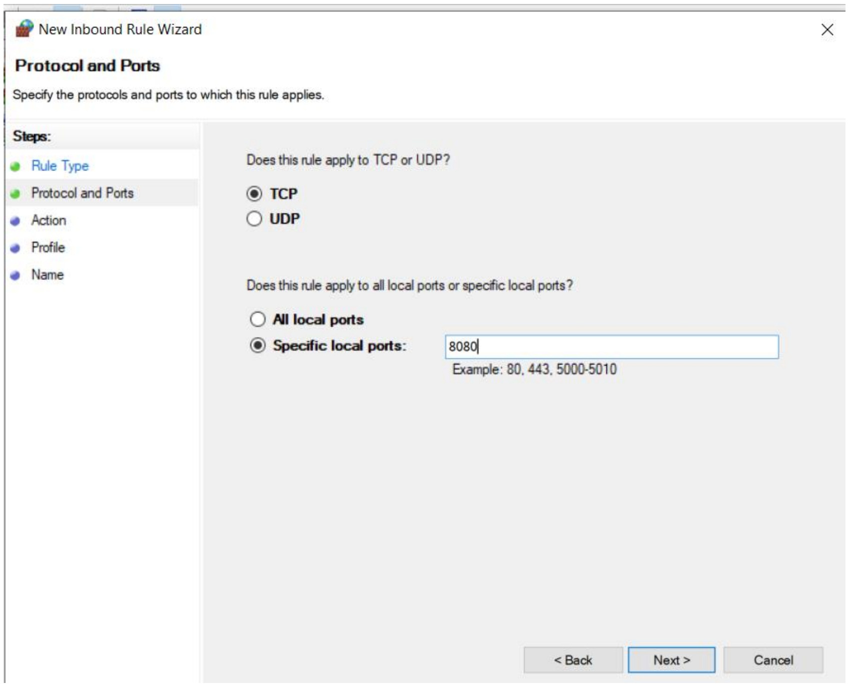
What type of rule would you like to create?

○ **Program**
Rule that controls connections for a program.

● **Port**
Rule that controls connections for a TCP or UDP port.

○ **Predefined:**
@FirewallAPI.dll,-80200
Rule that controls connections for a Windows experience.

○ **Custom**
Custom rule.



New Inbound Rule Wizard

Protocol and Ports

Specify the protocols and ports to which this rule applies.

Steps:

- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

Does this rule apply to TCP or UDP?

● **TCP**

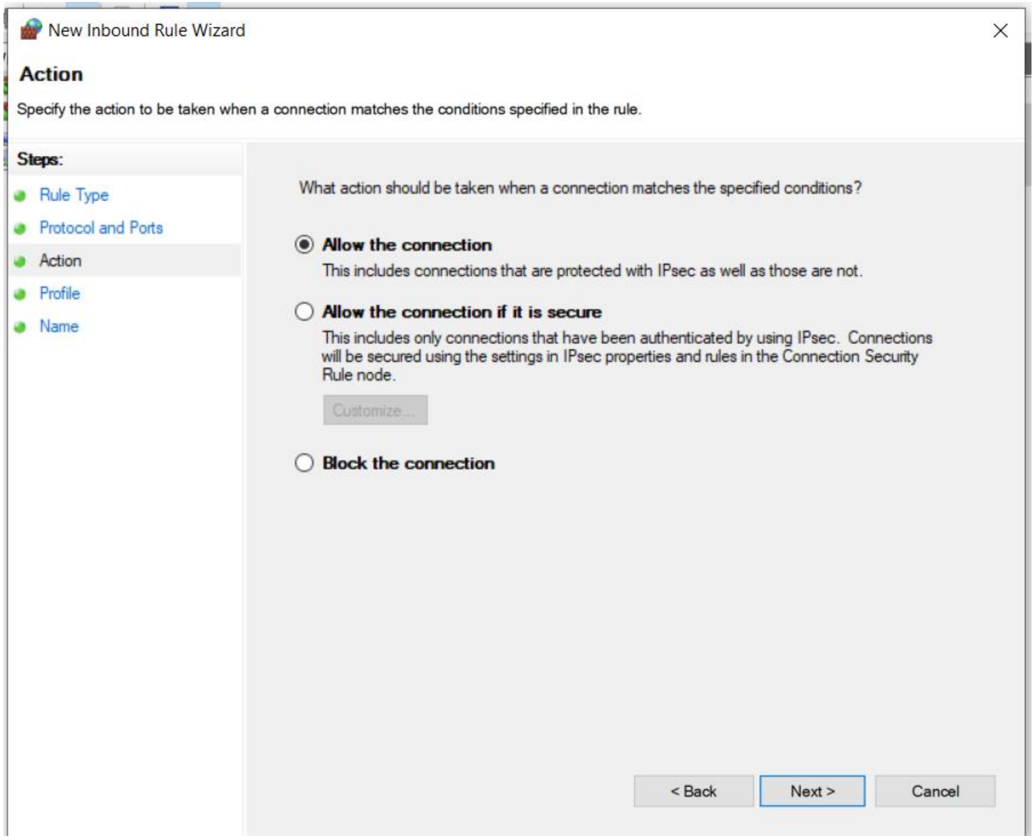
○ **UDP**

Does this rule apply to all local ports or specific local ports?

○ **All local ports**

● **Specific local ports:** 8080

Example: 80, 443, 5000-5010



Action

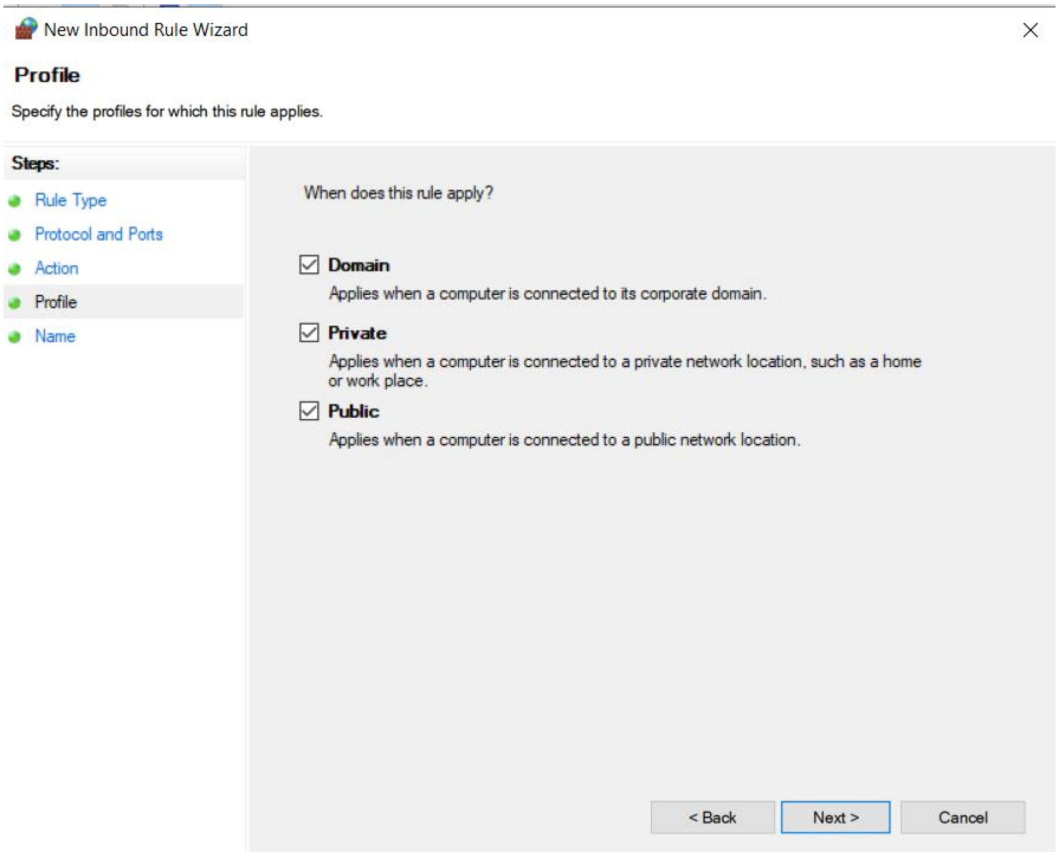
Specify the action to be taken when a connection matches the conditions specified in the rule.

What action should be taken when a connection matches the specified conditions?

● **Allow the connection**
This includes connections that are protected with IPsec as well as those are not.

○ **Allow the connection if it is secure**
This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.

○ **Block the connection**



Profile

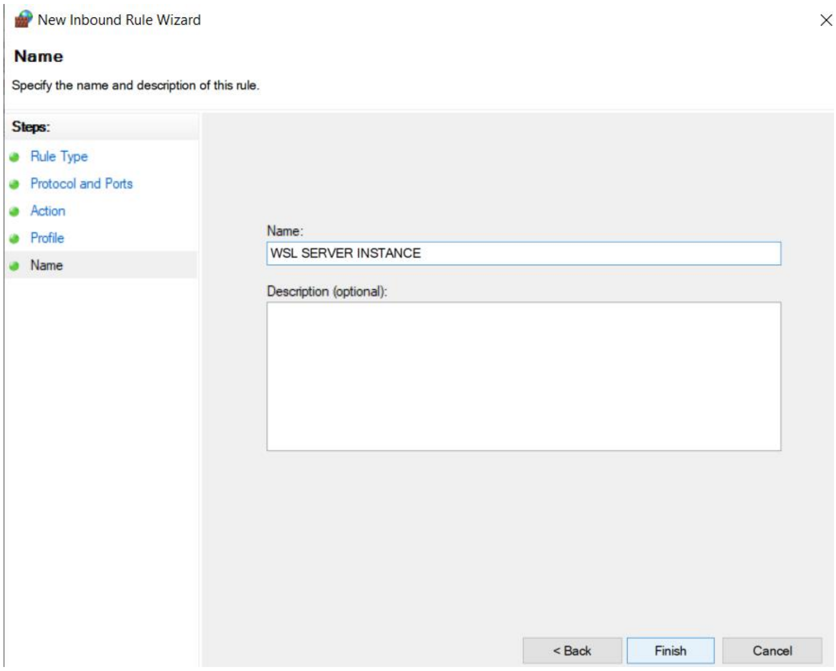
Specify the profiles for which this rule applies.

When does this rule apply?

☑ **Domain**
Applies when a computer is connected to its corporate domain.

☑ **Private**
Applies when a computer is connected to a private network location, such as a home or work place.

☑ **Public**
Applies when a computer is connected to a public network location.



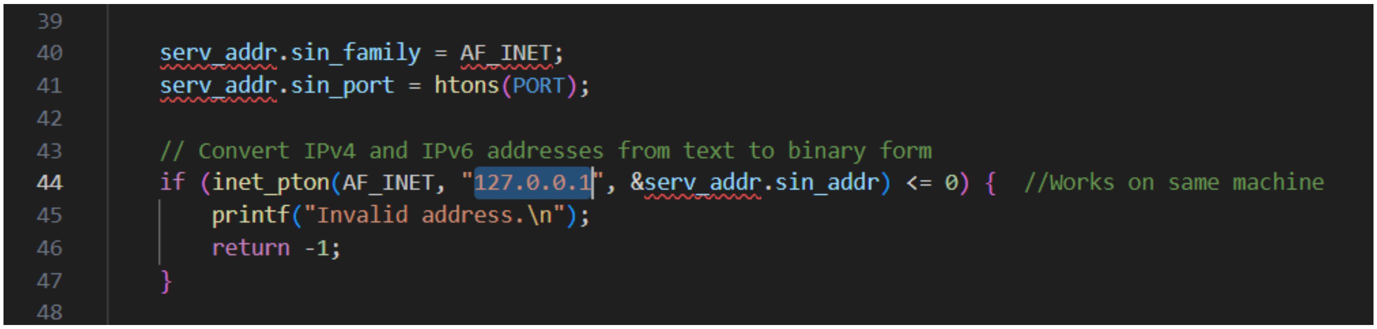
Name

Specify the name and description of this rule.

Name: WSL SERVER INSTANCE

Description (optional):

- Replace the IP address in the client code with the physical IP address of your server machine, (Not the WSL instance)
  Provide the Private IP Address of the server machine(Not the public IP Address)



```

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(PORT);

    // Convert IPv4 and IPv6 addresses from text to binary form
    if (inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr) <= 0) {  //Works on same machine
        printf("Invalid address.\n");
        return -1;
    }

```