

```

# UDP Chat Server

# Prompt for server IP and port
$serverIP = Read-Host "Enter the server IP address (or press Enter for any available IP)"
if ([string]::IsNullOrEmpty($serverIP)) {
    $serverIP = "0.0.0.0"
}
$port = Read-Host "Enter the UDP port number"

# Create UDP server
$udpServer = New-Object System.Net.Sockets.UdpClient $port

Write-Host "Server listening on $serverIP`:$port"
Write-Host "Waiting for client messages..."

$remoteEndPoint = New-Object System.Net.IPEndPoint([System.Net.IPAddress]::Any, 0)
$lineNumber = 1

while ($true) {
    # Receive message
    $receivedBytes = $udpServer.Receive([ref]$remoteEndPoint)
    $receivedMessage = [Text.Encoding]::ASCII.GetString($receivedBytes)

    # Add timestamp and line number to received message
    $timestamp = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
    Write-Host "[$timestamp] # $lineNumber Client ($($remoteEndPoint.Address): $($remoteEndPoint.Port)): $receivedMessage"

    # Send response
    $response = Read-Host "Server"

    # Add timestamp and line number to response
    $responseTimestamp = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
    $formattedResponse = "[$responseTimestamp] # $lineNumber $response"
    $responseBytes = [Text.Encoding]::ASCII.GetBytes($formattedResponse)
    $udpServer.Send($responseBytes, $responseBytes.Length, $remoteEndPoint)

    # Display sent message with timestamp and line number
    Write-Host "Sent: $formattedResponse"

    $lineNumber++
}

# Note: This server will run indefinitely. Press Ctrl+C to stop it.

```