

Echoserver

Echo server and client using python socket

AIM:

To develop a simple webserver to serve html programming pages.

DESIGN STEPS:

Step 1:

Design of echo server and client using python socket

Step 2:

Implementation using Python code

Step 3:

Testing the server and client

PROGRAM:

DEVELOPED BY: HARSHAVARDHAN

REGISTER NO: 212222240114

SERVER SIDE

```
import socket
HOST = "127.0.0.1" # The server's hostname or IP address
PORT = 65432 # The port used by the server
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    s.sendall(b"Hello, world")
    data = s.recv(1024)
    print(f"Received {data!r}")
```



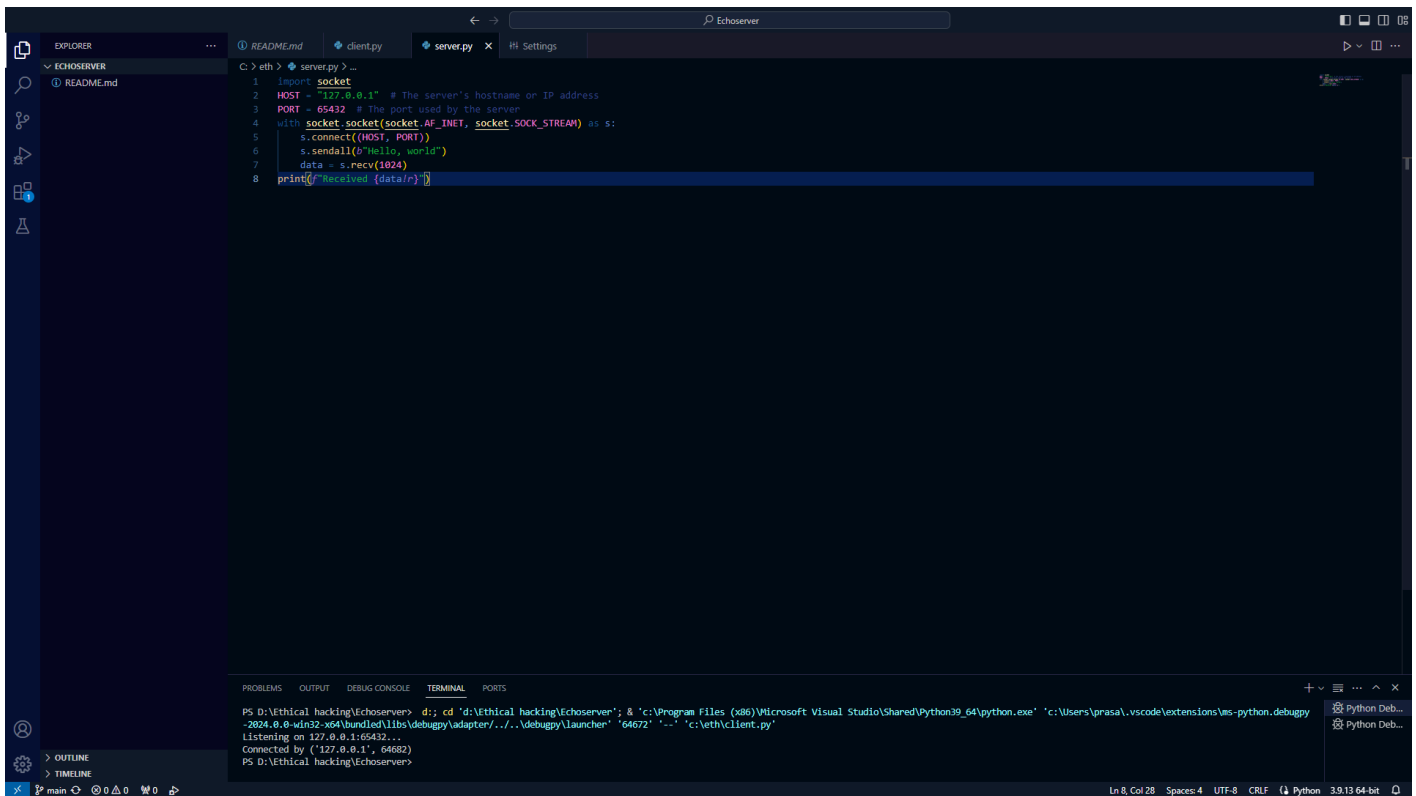
CLIENT SIDE



```
import socket
HOST = "127.0.0.1" # Standard loopback interface address (localhost)
PORT = 65432 # Port to listen on (non-privileged ports are > 1023)
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    try:
        s.bind((HOST, PORT))
    except Exception as e:
        print(f"Error binding to {HOST}:{PORT}: {e}")
        exit()
    s.listen()
    print(f"Listening on {HOST}:{PORT}...")
    try:
        conn, addr = s.accept()
    except Exception as e:
        print(f"Error accepting connection: {e}")
        exit()
    with conn:
        print(f"Connected by {addr}")
        while True:
            try:
                data = conn.recv(1024)
                if not data:
                    break
                conn.sendall(data)
            except Exception as e:
                print(f"Error receiving/sending data: {e}")
                exit()
```

OUTPUT:

SERVER SIDE



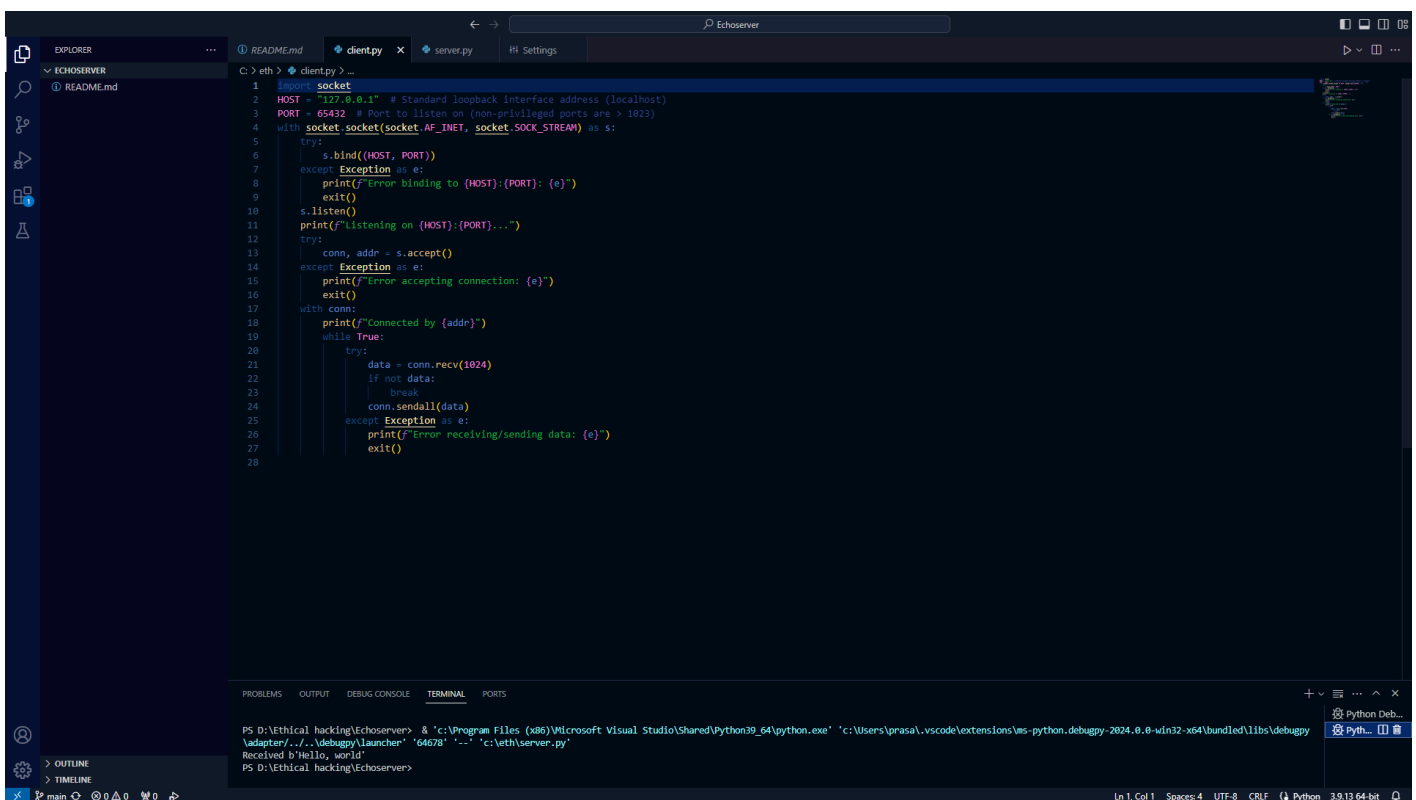
The screenshot shows the Visual Studio Code editor with the `server.py` file open. The code is a simple Python socket server that listens on `127.0.0.1` at port `65432`. It receives a connection from `127.0.0.1` and prints the received data.

```
1 import socket
2 HOST = "127.0.0.1" # The server's hostname or IP address
3 PORT = 65432 # The port used by the server
4 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
5     s.connect((HOST, PORT))
6     s.sendall(b"Hello, world")
7     data = s.recv(1024)
8     print(f"Received {data}")
```

The terminal output shows the command prompt running the server and the output of the client.

```
PS D:\ethical hacking\Echoserver> d; cd 'd:\ethical hacking\Echoserver'; & 'c:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe' 'c:\Users\prasa\.vscode\extensions\ms-python.debugpy-2024.0.0-win32-x64\buggy\launcher' '64672' '...' 'c:\eth\client.py'
Listening on 127.0.0.1:65432...
Connected by ('127.0.0.1', 64682)
PS D:\ethical hacking\Echoserver>
```

CLIENT SIDE



The screenshot shows the Visual Studio Code editor with the `client.py` file open. The code is a simple Python socket client that connects to `127.0.0.1` at port `65432`. It sends the data `b"Hello, world"` and receives the response.

```
1 import socket
2 HOST = "127.0.0.1" # Standard loopback interface address (localhost)
3 PORT = 65432 # Port to listen on (non-privileged ports are > 1023)
4 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
5     try:
6         s.bind((HOST, PORT))
7     except Exception as e:
8         print(f"Error binding to {HOST}:{PORT}: {e}")
9         exit()
10    s.listen()
11    print(f"Listening on {HOST}:{PORT}...")
12    try:
13        conn, addr = s.accept()
14    except Exception as e:
15        print(f"Error accepting connection: {e}")
16        exit()
17    with conn:
18        print(f"Connected by {addr}")
19        while True:
20            try:
21                data = conn.recv(1024)
22                if not data:
23                    break
24                conn.sendall(data)
25            except Exception as e:
26                print(f"Error receiving/sending data: {e}")
27                exit()
28
```

The terminal output shows the command prompt running the client and the output of the server.

```
PS D:\ethical hacking\Echoserver> & 'c:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe' 'c:\Users\prasa\.vscode\extensions\ms-python.debugpy-2024.0.0-win32-x64\buggy\launcher' '64678' '...' 'c:\eth\server.py'
Received b'Hello, world'
PS D:\ethical hacking\Echoserver>
```

RESULT:

The program is executed successfully