PRACTICAL-3

Create a client-server application and show the interaction using 2-way communication and handshaking.

***server.js***

***const*** net **=** require('net');

***const*** server **=** net.createServer((socket) ***=>*** {

    console.log('Client connected!');

    socket.write('Hello, Client! Ready to communicate?');

    socket.on('data', (data) ***=>*** {

***const*** message **=** data.toString();

        console.log(`Client: ${message}`);

*if* (message.toLowerCase() **===** 'exit') {

            console.log('Client disconnected.');

            socket.end();

        } *else* {

            process.stdout.write('Server: ');

            process.stdin.once('data', (input) ***=>*** {

                socket.write(input.toString());

            });

        }

    });

    socket.on('end', () ***=>*** {

        console.log('Connection closed by the client.');

    });

});

server.listen(5000, '127.0.0.1', () ***=>*** {

    console.log('Server is listening on port 5000...');

});

***client.js***

***const*** net **=** require('net');

***const*** client **=** net.createConnection({ port: 5000, host: '127.0.0.1' }, () ***=>*** {

    console.log('Connected to the server.');

});

client.on('data', (data) ***=>*** {

***const*** message **=** data.toString();

    console.log(`Server: ${message}`);

*if* (message.toLowerCase() **===** 'exit') {

        console.log('Disconnected from server.');

        client.end();

    } *else* {

        process.stdout.write('Client: ');

        process.stdin.once('data', (input) ***=>*** {

            client.write(input.toString());

        });

    }

});

client.on('end', () ***=>*** {

    console.log('Disconnected from the server.');

});

**Output Screenshot**

