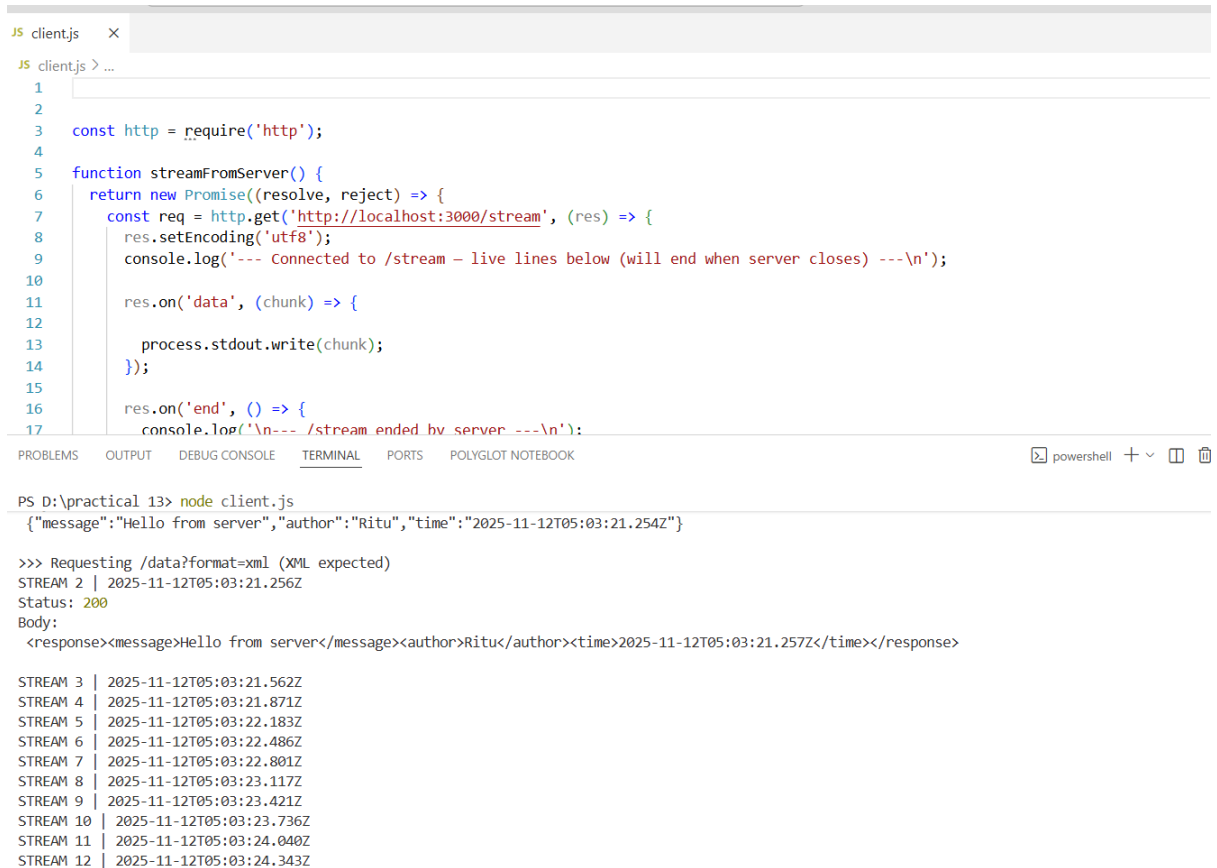


Practical 13-



The image shows a Visual Studio Code editor with a file named `client.js` open. The code is a Node.js script that makes an HTTP GET request to `http://localhost:3000/stream`. It sets the encoding to 'utf8' and logs a message when connected. It then listens for 'data' chunks and writes them to `process.stdout`, and for 'end' to log a message.

```
1
2
3 const http = require('http');
4
5 function streamFromServer() {
6   return new Promise((resolve, reject) => {
7     const req = http.get('http://localhost:3000/stream', (res) => {
8       res.setEncoding('utf8');
9       console.log('--- Connected to /stream - live lines below (will end when server closes) ---\n');
10
11       res.on('data', (chunk) => {
12         process.stdout.write(chunk);
13       });
14     });
15     res.on('end', () => {
16       console.log('\n--- /stream ended by server ---\n');
17     });
18   });
19 }
```

The terminal output shows the command `node client.js` being executed. It displays the JSON response from the server, the status 200, and the body of the response in XML format. It also shows a series of stream logs with timestamps.

```
PS D:\practical 13> node client.js
{"message":"Hello from server","author":"Ritu","time":"2025-11-12T05:03:21.254Z"}

>>> Requesting /data?format=xml (XML expected)
STREAM 2 | 2025-11-12T05:03:21.256Z
Status: 200
Body:
<response><message>Hello from server</message><author>Ritu</author><time>2025-11-12T05:03:21.257Z</time></response>

STREAM 3 | 2025-11-12T05:03:21.562Z
STREAM 4 | 2025-11-12T05:03:21.871Z
STREAM 5 | 2025-11-12T05:03:22.183Z
STREAM 6 | 2025-11-12T05:03:22.486Z
STREAM 7 | 2025-11-12T05:03:22.801Z
STREAM 8 | 2025-11-12T05:03:23.117Z
STREAM 9 | 2025-11-12T05:03:23.421Z
STREAM 10 | 2025-11-12T05:03:23.736Z
STREAM 11 | 2025-11-12T05:03:24.040Z
STREAM 12 | 2025-11-12T05:03:24.343Z
```

Repo- <https://github.com/RituSingh0204/Practical-13-NODE-JS.git>