ASSIGNMENT NO : 3B

```javascript
const http = require('http');

const fs = require('fs');

const path = require('path');

const url = require('url');

const PORT = 3000;

// Mapping file extensions to Content-Type headers

const mimeTypes = {

  '.html': 'text/html',

  '.js': 'text/javascript',

  '.css': 'text/css',

  '.json': 'application/json',

  '.png': 'image/png',

  '.jpg': 'image/jpg',

  '.gif': 'image/gif',

};
const server = http.createServer((req, res) => {

  // 1. Parse the URL to get the file path

  const parsedUrl = url.parse(req.url);

  let filePath = '.' + parsedUrl.pathname;

  // 2. Default to index.html if the path is just '/'

  if (filePath === './') {

    filePath = './public/index.html';

  } else {

    filePath = './public' + parsedUrl.pathname;

  } const extname = String(path.extname(filePath)).toLowerCase();

  const contentType = mimeTypes[extname] || 'application/octet-stream';

  // 3. Read the file from the File System (fs)

  fs.readFile(filePath, (error, content) => {

    if (error) {

      if (error.code === 'ENOENT') {

        // File not found

        res.writeHead(404, { 'Content-Type': 'text/html' });

        res.end('<h1>404 Not Found</h1>', 'utf-8');

      } else {

        // Some server error (e.g., permissions)

        res.writeHead(500);

        res.end(`Server Error: ${error.code}`);

      }

    } else {

      // 4. Success! Serve the file

      res.writeHead(200, { 'Content-Type': contentType });

      res.end(content, 'utf-8');

    }

  });});server.listen(PORT, () => {

  console.log(`Server running at http://localhost:${PORT}/`);

});
```
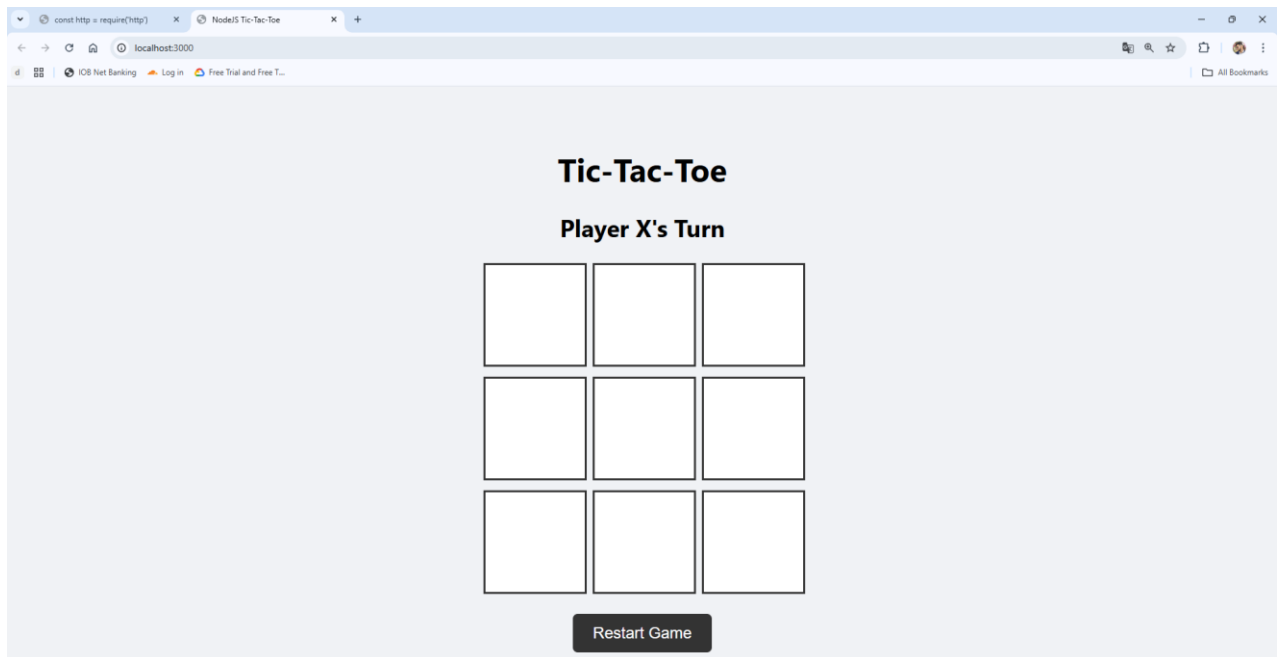
```
34    server.listen(PORT, () => {
55        console.log(`Server running at http://localhost:${PORT}/`);
56    });
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   **TERMINAL**   PORTS

```
PS C:\DRIVE d\ACADEMICS\3a> node server.js
Server running at http://localhost:3000/
(node:10648) [DEP0169] DeprecationWarning: `url.parse()` behavior is not standardized and prone to errors that have
WHATWG URL API instead. CVEs are not issued for `url.parse()` vulnerabilities.
(Use `node --trace-deprecation ...` to show where the warning was created)
```

Index.html

Conclusion: