

The screenshot displays a web browser window with a REST client interface. The top navigation bar includes 'File', 'Edit', 'Selection', 'View', and a search icon. The main content area is divided into three panes: Explorer, Output, and Console.

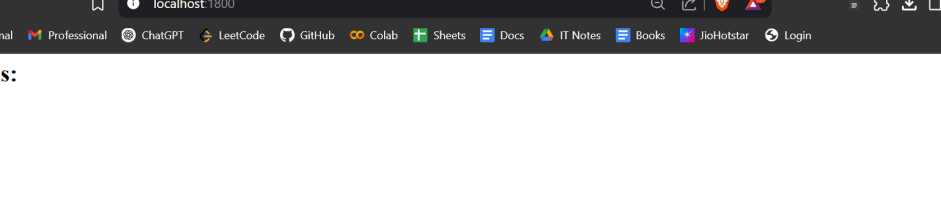
**Explorer Pane:** Shows a project structure with folders for 'WADL', 'Assignment\_1', 'Assignment\_2', 'Assignment\_3', 'next-assignment', 'static-file-server', 'data.json', 'file.txt', 'file2.txt', 'read.txt', 'write.js', 'R server.js', 'R.js', 'R.js.css', 'Assignment\_4', and 'output.html'.

**Output Pane:** Displays the WADL (Web Application Description Language) for the 'next-assignment' endpoint. The WADL is a JSON object containing metadata and a list of resources. The 'resources' array lists several endpoints, including 'static-file-server', 'data.json', 'file.txt', 'file2.txt', 'read.txt', 'write.js', 'R server.js', 'R.js', 'R.js.css', 'Assignment\_4', and 'output.html'.

**Console Pane:** Shows the output of the REST client. It displays the 'next-assignment' endpoint, which is a GET request to 'http://localhost:3000/next-assignment'. The response is a 200 OK status with a Content-Type of 'application/json'. The response body is a JSON object containing the WADL for the 'next-assignment' endpoint.

The WADL JSON object is as follows:

```
{
  "title": "next-assignment",
  "description": "next-assignment",
  "resources": [
    {
      "name": "static-file-server",
      "description": "static-file-server",
      "methods": [
        {
          "name": "GET",
          "description": "GET",
          "parameters": [
            {
              "name": "url",
              "description": "url",
              "required": true,
              "type": "string"
            }
          ],
          "responses": [
            {
              "code": 200,
              "description": "200",
              "headers": [
                {
                  "name": "Content-Type",
                  "value": "application/json"
                }
              ],
              "body": "static-file-server"
            }
          ]
        }
      ]
    },
    {
      "name": "data.json",
      "description": "data.json",
      "methods": [
        {
          "name": "GET",
          "description": "GET",
          "parameters": [
            {
              "name": "url",
              "description": "url",
              "required": true,
              "type": "string"
            }
          ],
          "responses": [
            {
              "code": 200,
              "description": "200",
              "headers": [
                {
                  "name": "Content-Type",
                  "value": "application/json"
                }
              ],
              "body": "data.json"
            }
          ]
        }
      ]
    },
    {
      "name": "file.txt",
      "description": "file.txt",
      "methods": [
        {
          "name": "GET",
          "description": "GET",
          "parameters": [
            {
              "name": "url",
              "description": "url",
              "required": true,
              "type": "string"
            }
          ],
          "responses": [
            {
              "code": 200,
              "description": "200",
              "headers": [
                {
                  "name": "Content-Type",
                  "value": "application/json"
                }
              ],
              "body": "file.txt"
            }
          ]
        }
      ]
    },
    {
      "name": "file2.txt",
      "description": "file2.txt",
      "methods": [
        {
          "name": "GET",
          "description": "GET",
          "parameters": [
            {
              "name": "url",
              "description": "url",
              "required": true,
              "type": "string"
            }
          ],
          "responses": [
            {
              "code": 200,
              "description": "200",
              "headers": [
                {
                  "name": "Content-Type",
                  "value": "application/json"
                }
              ],
              "body": "file2.txt"
            }
          ]
        }
      ]
    },
    {
      "name": "read.txt",
      "description": "read.txt",
      "methods": [
        {
          "name": "GET",
          "description": "GET",
          "parameters": [
            {
              "name": "url",
              "description": "url",
              "required": true,
              "type": "string"
            }
          ],
          "responses": [
            {
              "code": 200,
              "description": "200",
              "headers": [
                {
                  "name": "Content-Type",
                  "value": "application/json"
                }
              ],
              "body": "read.txt"
            }
          ]
        }
      ]
    },
    {
      "name": "write.js",
      "description": "write.js",
      "methods": [
        {
          "name": "GET",
          "description": "GET",
          "parameters": [
            {
              "name": "url",
              "description": "url",
              "required": true,
              "type": "string"
            }
          ],
          "responses": [
            {
              "code": 200,
              "description": "200",
              "headers": [
                {
                  "name": "Content-Type",
                  "value": "application/json"
                }
              ],
              "body": "write.js"
            }
          ]
        }
      ]
    },
    {
      "name": "R server.js",
      "description": "R server.js",
      "methods": [
        {
          "name": "GET",
          "description": "GET",
          "parameters": [
            {
              "name": "url",
              "description": "url",
              "required": true,
              "type": "string"
            }
          ],
          "responses": [
            {
              "code": 200,
              "description": "200",
              "headers": [
                {
                  "name": "Content-Type",
                  "value": "application/json"
                }
              ],
              "body": "R server.js"
            }
          ]
        }
      ]
    },
    {
      "name": "R.js",
      "description": "R.js",
      "methods": [
        {
          "name": "GET",
          "description": "GET",
          "parameters": [
            {
              "name": "url",
              "description": "url",
              "required": true,
              "type": "string"
            }
          ],
          "responses": [
            {
              "code": 200,
              "description": "200",
              "headers": [
                {
                  "name": "Content-Type",
                  "value": "application/json"
                }
              ],
              "body": "R.js"
            }
          ]
        }
      ]
    },
    {
      "name": "R.js.css",
      "description": "R.js.css",
      "methods": [
        {
          "name": "GET",
          "description": "GET",
          "parameters": [
            {
              "name": "url",
              "description": "url",
              "required": true,
              "type": "string"
            }
          ],
          "responses": [
            {
              "code": 200,
              "description": "200",
              "headers": [
                {
                  "name": "Content-Type",
                  "value": "application/json"
                }
              ],
              "body": "R.js.css"
            }
          ]
        }
      ]
    },
    {
      "name": "Assignment_4",
      "description": "Assignment_4",
      "methods": [
        {
          "name": "GET",
          "description": "GET",
          "parameters": [
            {
              "name": "url",
              "description": "url",
              "required": true,
              "type": "string"
            }
          ],
          "responses": [
            {
              "code": 200,
              "description": "200",
              "headers": [
                {
                  "name": "Content-Type",
                  "value": "application/json"
                }
              ],
              "body": "Assignment_4"
            }
          ]
        }
      ]
    },
    {
      "name": "output.html",
      "description": "output.html",
      "methods": [
        {
          "name": "GET",
          "description": "GET",
          "parameters": [
            {
              "name": "url",
              "description": "url",
              "required": true,
              "type": "string"
            }
          ],
          "responses": [
            {
              "code": 200,
              "description": "200",
              "headers": [
                {
                  "name": "Content-Type",
                  "value": "application/json"
                }
              ],
              "body": "output.html"
            }
          ]
        }
      ]
    }
  ]
}
```



The screenshot shows a web browser window with the address bar set to 'localhost:1800'. The browser's top bar includes various navigation and utility icons. Below the address bar, a horizontal menu contains links to 'Personal', 'Professional', 'ChatGPT', 'LeetCode', 'GitHub', 'Colab', 'Sheets', 'Docs', 'IT Notes', 'Books', 'JioHotstar', and 'Login'. The main content area of the browser displays the heading 'List of files:' followed by a bulleted list of files: 'data.json', 'file1.txt', 'file2.html', 'rootfile.txt', 'script.js', 'server.js', and 'style.css'. Each file name is preceded by a small purple dot.

localhost:1800

localhost:1800

Personal Professional ChatGPT LeetCode GitHub Colab Sheets Docs IT Notes Books JioHotstar Login

## List of files:

- data.json
- file1.txt
- file2.html
- rootfile.txt
- script.js
- server.js
- style.css

## Assignment 3a

### Web Application Development Laboratory

### 3. Contents of server.js

```

const http = require('http');
const url = require('url');
const fs = require('fs');
const path = require('path');

const PORT = 1800;

const mimeType = {
  '.ico': 'image/x-icon',
  '.html': 'text/html',
  '.js': 'text/javascript',
  '.json': 'application/json',
  '.css': 'text/css',
  '.png': 'image/png',
  '.jpg': 'image/jpeg',
  '.wav': 'audio/wav',
  '.mp3': 'audio/mpeg',
  '.svg': 'image/svg+xml',
  '.pdf': 'application/pdf',
  '.doc': 'application/msword',
  '.eot': 'application/vnd.ms-fontobject',
  '.ttf': 'application/font-sfnt'
};

http.createServer((req, res) => {
  const parsedUrl = url.parse(req.url);
  if (parsedUrl.pathname === '/') {
    // Show list of files
    res.setHeader('Content-type', 'text/html');
    let files = fs.readdirSync("./");
    let html = "<h1>List of files:</h1><ul>";
    files.forEach(file => {
      if (fs.statSync(file).isFile()) {
        html += `<li><a href="${file}">${file}</a></li>`;
      }
    });
    html += "</ul>";
    res.end(html);
  } else {
    // Serve the file
    const sanitizePath = path.normalize(parsedUrl.pathname).replace(/^(\\.\\.\\.\\|\\/\\|\\|)+/, '');
    const pathname = path.join(__dirname, sanitizePath);
    if (!fs.existsSync(pathname)) {
      res.statusCode = 404;
      res.end(`File ${pathname} not found!`);
    } else {
      fs.readFile(pathname, (err, data) => {
        if (err) {
          res.statusCode = 500;
          res.end(`Error getting the file.`);
        } else {
          const ext = path.parse(pathname).ext;
          res.setHeader('Content-type', mimeType[ext] || 'text/plain');
          res.end(data);
        }
      });
    }
  }
}).listen(PORT);

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

```