

Assignemnt 3A

Index.html

```
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 === '/') {
    var fileList = fs.readdirSync("./public/");
    var filesLink = "<ul>";
    fileList.forEach(element => {
      if (fs.statSync("./public/" + element).isFile())
        filesLink += `<li><a
href='./${element}'>${element}</a></li>`;
    }
  });
  filesLink += "</ul>";
  var html = `
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>Directory Listing</title>
  <style>
    body {
      font-family: Arial, Helvetica, sans-serif;
      background-color: #f4f4ff;
      width:50%;
      margin:0 auto;
    }
  </style>
</head>
<body>
  <h1>Welcome To My Web Server
<br></h1>
  <h2 style="text-align: center"> We have
following Public files to share</h2>
  <hr>
  ${filesLink}
</body>
</html>
`;
  res.setHeader('Content-type', 'text/html');
  res.end(html);
} else {
```

```
h1 {
  text-align: center;
  color: #555;
  margin-top:5rem;
}

ul {
  list-style: none;
  padding: 0;
  margin: 0;
}

li {
  margin-bottom: 10px;
  background-color: #fff;
  padding: 1rem ;
  border-radius: 5px;
  width: 55%;
  margin: 1rem auto;
}

a {
  text-decoration: none;
  font-weight: bold;
  color:#5555ee;
  font-size:1.2rem;
}

a:hover {
  text-decoration: underline;
}
</style>
</head>

<body>
  <h1>Welcome To My Web Server
<br></h1>
  <h2 style="text-align: center"> We have
following Public files to share</h2>
  <hr>
  ${filesLink}
</body>
</html>
`;
  res.setHeader('Content-type', 'text/html');
  res.end(html);
} else {
```

```

const sanitizePath =
path.normalize(parsedUrl.pathname).replace(/^(\/\.|
.[\V\])+/ , "");
let pathname = path.join(__dirname +
"/public", sanitizePath);
if (!fs.existsSync(pathname)) {
res.statusCode = 404;
res.end('File ${pathname} not found!');
} else {
fs.readFile(pathname, function (err, data) {
if (err) {
res.statusCode = 500;

```

```

res.end('Error in getting the file.');
```

```

} else {
const ext = path.parse(pathname).ext;
res.setHeader('Content-type',
mimeType[ext] || 'text/plain');
res.end(data);
}
});
}
}
}).listen(PORT);

```

output:

