|  |  |
| --- | --- |
| **Ex. No: 6** | **Web Server Creation using NodeJS** |
| **21.09.2023** |

**Aim:**To Create a Web Server offering basic web service(s) to the front-end.

**Algorithm:**

1. Ensure you have Node.js installed on your system.
2. Develop a JavaScript file (e.g., server.js) for your web server.
3. In server.js, require Node.js's built-in http module using require('http').
4. Use the http.createServer() method to create an HTTP server, specifying a request handling function.
5. Inside the request handling function, use the request and response objects to define how your server should respond to different routes and HTTP methods.
6. Test your web server using tools like cURL or Postman. Debug and refine your route handling as needed.
7. Optionally, configure the web server to serve static HTML, CSS, and JavaScript files if your front-end includes them, using the fs (file system) module.

**Program:**

const http = require('http');

const url = require('url');

const fs = require('fs');

// Create an HTTP server

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

// Parse the request URL

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

const pathname = parsedUrl.pathname;

// Set the response header with a status code and content type

res.setHeader('Content-Type', 'text/html');

if (pathname === '/') {

// Serve the homepage

fs.readFile('index.html', (err, data) => {

if (err) {

res.writeHead(500);

res.end('Error reading the file');

} else {

res.writeHead(200);

res.end(data);

}

});

} else if (pathname === '/about') {

// Serve an about page

res.writeHead(200);

res.end('<h1>About Us</h1>');

} else if (pathname === '/contact') {

// Serve a contact form

if (req.method === 'GET') {

res.writeHead(200);

res.end(`

<h1>Contact Us</h1>

<form method="post" action="/contact">

<input type="text" name="name" placeholder="Your Name"><br>

<input type="email" name="email" placeholder="Your Email"><br>

<textarea name="message" placeholder="Your Message"></textarea><br>

<input type="submit" value="Submit">

</form>

`);

} else if (req.method === 'POST') {

// Handle form submission

let body = '';

req.on('data', (chunk) => {

body += chunk.toString();

});

req.on('end', () => {

const formData = new URLSearchParams(body);

const name = formData.get('name');

const email = formData.get('email');

const message = formData.get('message');

// The form data can be stored in a

console.log("Here is the form information from the user: \n", name);

console.log('Name:', name);

console.log('Email:', email);

console.log('Message:', message);

res.writeHead(200);

res.end('<h1>Thank you for your message!</h1>');

});

}

} else {

// Handle 404 Not Found

res.writeHead(404);

res.end('<h1>404 Not Found</h1>');

}

});

// Listen on port 3000

const port = 3000;

server.listen(port, () => {

console.log(`Server is listening on port ${port}`);

});

**Output:**

Github Link: <https://github.com/kavin-t28/CS3809-Web-Technologies-Lab>

A screenshot of a computer

Description automatically generated

Server Side output:

A computer screen with white text

Description automatically generated

**Result:**

Therefore, we've successfully implemented a web server backend using NodeJS .