**EXPERIMENT - 3**

**Aim:** Develop Node JS Application and Implement HTTP Services in Node JS (Request and Response)

**Tools Used:** NPM, JS, NodeJS

**Description:**

**NPM (Node Package Manager):**

NPM (Node Package Manager) is a package manager for the JavaScript programming language. It is used to install, manage, and share libraries and dependencies that are used in JavaScript projects. NPM is typically used with Node.js, a JavaScript runtime environment. It allows developers to install packages from the NPM registry, which is a large collection of open-source libraries. NPM helps in automating the process of managing dependencies, ensuring that the necessary packages are available for a project to run. NPM can be used through the command line to install packages locally or globally.

**JavaScript (JS):**

JavaScript (JS) is a dynamic, high-level programming language that is primarily used for creating interactive effects within web browsers. It enables developers to build dynamic websites and applications by adding behavior to web pages. JavaScript is a client-side language, meaning it runs in the user's browser, but it can also be used on the server-side through environments like Node.js. JavaScript allows for tasks like form validation, animations, fetching data from servers (AJAX), and much more, making it an essential tool for modern web development.

**Node.js:**

Node.js is an open-source, cross-platform JavaScript runtime environment that enables developers to run JavaScript code on the server-side. It is built on the V8 JavaScript engine, which powers Google Chrome. Node.js is designed to be efficient and scalable, particularly for I/O-heavy applications like web servers and real-time applications. It uses an event-driven, non-blocking I/O model, allowing it to handle a large number of simultaneous connections with high throughput. Developers use Node.js to create server-side applications, REST APIs, and build full-stack applications using JavaScript.

**1. Write a JS program to implement File operations using File stream module in Node.JS**

**Program:**

let http = require('http');

let fs = require('fs');

let url = require('url');

http.createServer(function (req, res) {

  // Parse the URL

  var q = url.parse(req.url, true);

  console.log(req.url);

  var filename = "." + q.pathname;

  fs.readFile(filename, function (err, data) {

    if (err) {

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

      return res.end("404 Not Found");

    }

    res.writeHead(200, { 'Content-Type': 'text/html' });

    res.write('<h1>Read operation:</h1>');

    res.write(data);

    fs.writeFile(filename, "<h3>I'm writing in this file..!</h3>", function (err) {

      if (err) {

        res.write('<h1>Write operation failed!</h1>');

        return res.end();

      }

      res.write('<h1>Write operation:</h1>');

      res.write("<h3>Data successfully written to the file!</h3>");

      return res.end();

    });

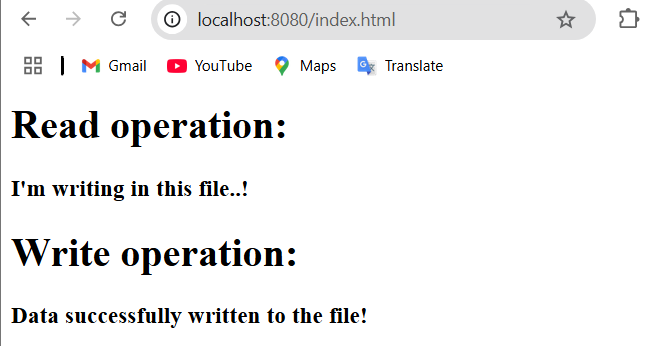
  });

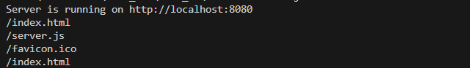
}).listen(8080, () => {

  console.log("Server is running on http://localhost:8080");

});

**Output:**





**2.** **Write a JS program to implement basic calculator operations on Node.js by using user defined Calc module.**

**Program:**

**Server.js:**

const http = require('http');

const url = require('url');

const calc = require('./cal.js');

http.createServer(function (req, res) {

  const q = url.parse(req.url, true);

  const pathname = q.pathname;

  const query = q.query;

  if (pathname === '/calculate') {

    const num1 = parseFloat(query.num1);

    const num2 = parseFloat(query.num2);

    const operation = query.operation;

    if (isNaN(num1) || isNaN(num2)) {

      res.writeHead(400, { 'Content-Type': 'text/html' });

      res.write("<h1>Error:</h1>");

      res.write("<p>Invalid numbers provided. Please provide valid numbers for num1 and num2.</p>");

      return res.end();

    }

    if (!operation) {

      res.writeHead(400, { 'Content-Type': 'text/html' });

      res.write("<h1>Error:</h1>");

      res.write("<p>No operation specified. Please provide an operation (add, subtract, multiply, divide).</p>");

      return res.end();

    }

    let result;

    try {

      if (operation === 'add') {

        result = calc.add(num1, num2);

      } else if (operation === 'subtract') {

        result = calc.subtract(num1, num2);

      } else if (operation === 'multiply') {

        result = calc.multiply(num1, num2);

      } else if (operation === 'divide') {

        result = calc.divide(num1, num2);

      } else {

        throw new Error("Invalid operation");

      }

      // Send a success response

      res.writeHead(200, { 'Content-Type': 'text/html' });

      res.write("<h1>Calculator Result:</h1>");

      res.write(`<p>${num1} ${operation} ${num2} = ${result}</p>`);

    } catch (error) {

      // Handle calculation errors

      res.writeHead(400, { 'Content-Type': 'text/html' });

      res.write("<h1>Error:</h1>");

      res.write(`<p>${error.message}</p>`);

    }

    return res.end();

  }

  // Handle invalid paths

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

  res.write("<h1>Page Not Found</h1>");

  return res.end();

}).listen(8000, () => {

  console.log("Server is running on http://localhost:8000");

});

**Cal.js:**

module.exports.add = function (a, b) {

    return a + b;

  };

  module.exports.subtract = function (a, b) {

    return a - b;

  };

  module.exports.multiply = function (a, b) {

    return a \* b;

  };

  module.exports.divide = function (a, b) {

    if (b === 0) {

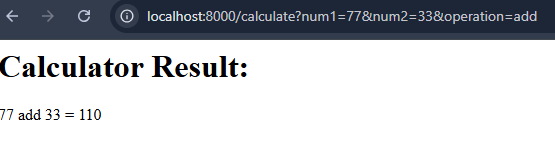
      throw new Error("Cannot divide by zero");

    }

    return a / b;

  };

**Output:**



**3. Register and Publish user package in NPM global registry**

Index.js:

function hello () {

    return "hello world !"

  }

  module.exports = hello;

**Output:**



**4. Write a JS program to establish client server communication using HTTP module in**

**NODE.JS**

**Client.js:**

const http = require('http');

const options = {

hostname: 'localhost', // The server's hostname

port: 3000,

path: '/', // The path to request

method: 'GET',

};

const req = http.request(options, (res) => {

let data = '';

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

data += chunk;

});

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

console.log(`Server response: ${data}`);

});

});

req.on('error', (e) => {

console.error(`Problem with the request: ${e.message}`);

});

req.end();

**Server.js:**

const http = require('http');

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

res.writeHead(200, { 'Content-Type': 'text/plain' });

res.end('Hello from the server!');

console.log(`Request Status Code: ${req.statusCode}`);

console.log(`Request Method: ${req.method}`);

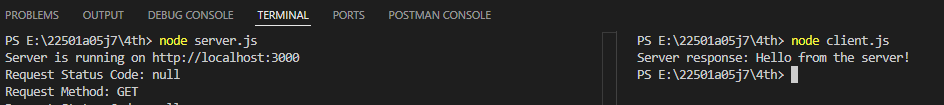
});

server.listen(3000, () => {

console.log('Server is running on http://localhost:3000');

});

**Output:**

****

**Result:** Successfully implemented various Node.js programs, including file operations with the fs module, a custom calculator module for basic arithmetic, registration and publishing of an NPM package, and client-server communication using the http module.