PRACTICAL EXAM QUESTIONS JULY 2023 COURSE MCA SEMESTER 1 SUBJECT CODE V20PCA104 ADVANCED WEB DEVELOPMENT LAB

ENROLLMENT NUMBER – EC2332251010074  
NAME OF THE STUDENT – ANKUR CHAUDHARY  
DATE – 14-07-2023  
SUBJECT CODE- V20PCA104SUBJECT NAME - ADVANCED WEB DEVELOPMENTCOURSE NAME – MCA

**NAME OF THE PROGRAM:**

**QUESTION 1:** CREATE NODE.JS APPLICATION TO DISPLAY SOME MESSAGE BY USING CLIENT SERVER MODEL.

**AIM:**  TO CREATE NODE.JS APPLICATION TO DISPLAY SOME MESSAGE BY USING CLIENT SERVER MODEL.

**PROCEDURE:**

1.To create a Node.js application, need to install the Node.js runtime on our machine. We will also need a code editor, such as Visual Studio Code.

2.To display some message by using the client-server model, we will need to create two files: one for the server and one for the client. The server file will handle the requests from the client and send back the response. The client file will make the requests to the server and display the response on the web page.

**SERVER SIDE CODE IS GIVEN BELOW:**

// Import the http module

const http = require('http');

// Create a server object

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

// Write a response header

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

// Write a response body

res.write('Hello from Node.js server!');

// End the response

res.end();

});

// Listen on port 3000

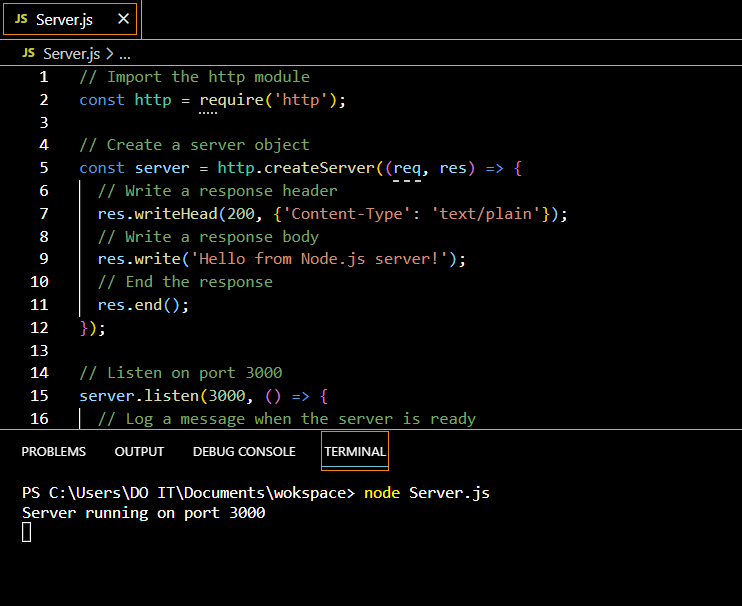
server.listen(3000, () => {

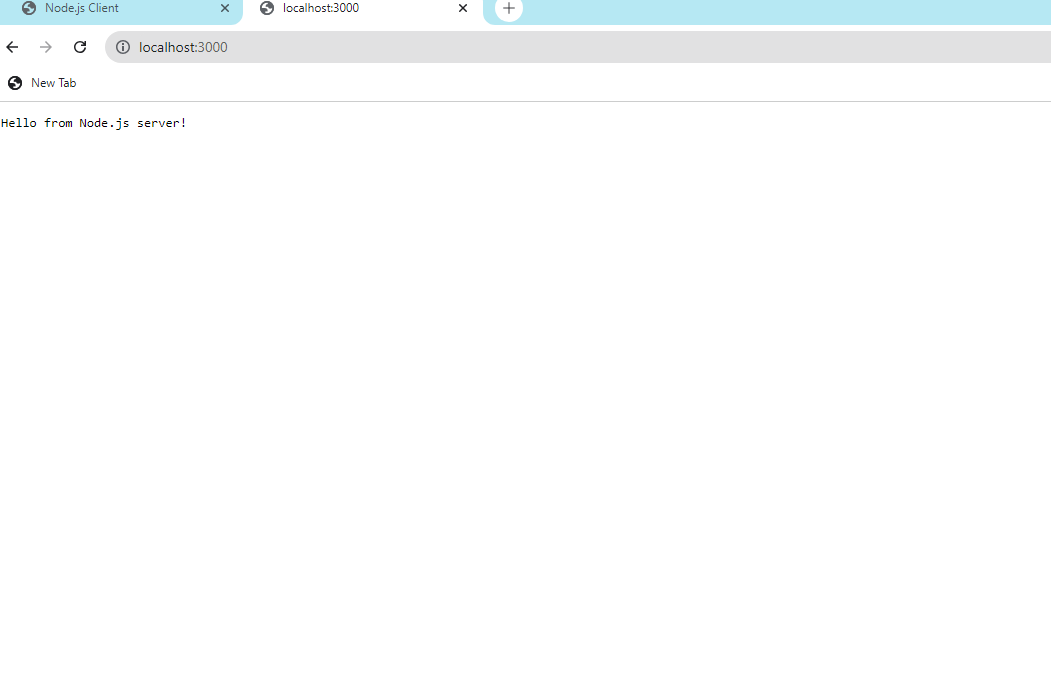
// Log a message when the server is ready

console.log('Server running on port 3000');

});

**OUTPUT:** Server running on port 3000





**CLIENT SIDE CODE IS GIVEN BELOW:**

<!DOCTYPE html>

<html>

<head>

<title>Node.js Client</title>

</head>

<body>

<h1>Node.js Client</h1>

<p id="message"></p>

<script>

// Create a new XMLHttpRequest object

const xhr = new XMLHttpRequest();

// Open a GET request to the server

xhr.open('GET', 'http://localhost:3000');

// Set the response type to text

xhr.responseType = 'text';

// Handle the onload event

xhr.onload = () => {

// Check if the status is OK

if (xhr.status === 200) {

// Get the response text

const response = xhr.responseText;

// Display the response text in the paragraph element

document.getElementById('message').textContent = response;

}

};

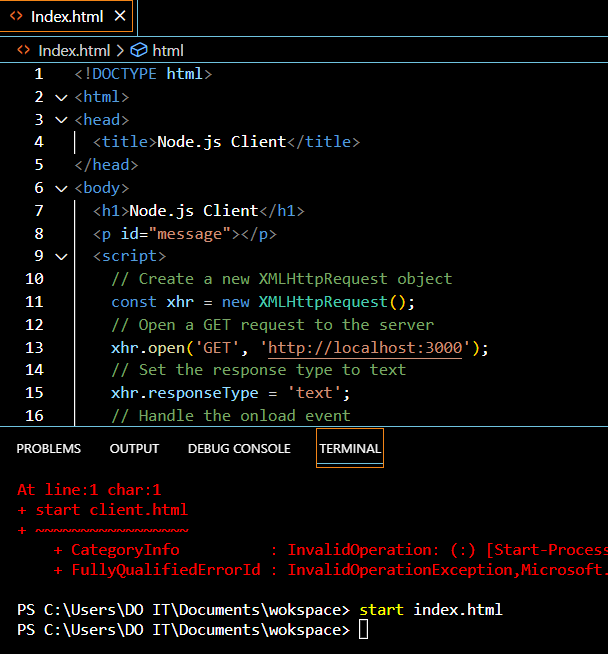
// Send the request

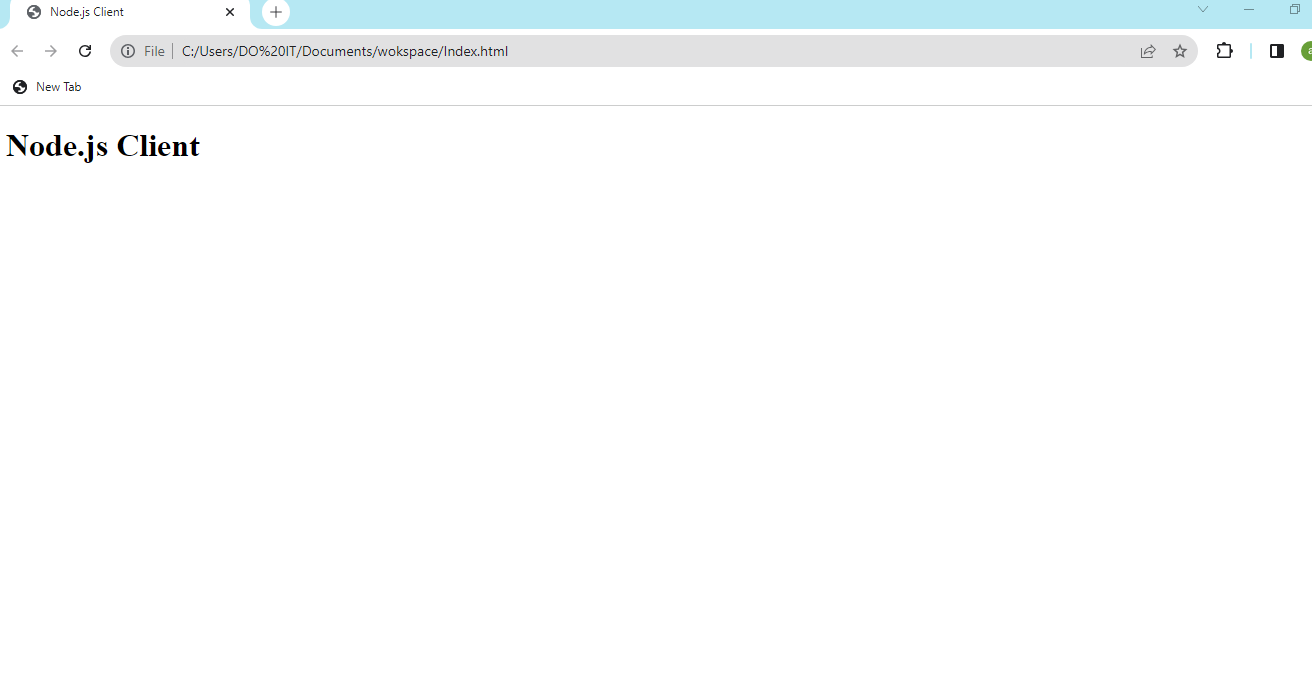
xhr.send();

</script>

</body>

</html>





**RESULT:**  Thus the create a simple Node.JS and running it as client server model and the simple node is created with server and the output is verified.