

Using Node.js to read HTML file and send HTML response



Project Name: MongoDB with Nodejs June Minor Project.

Project Description: Create a Node.js file that reads the HTML file, and return the content in local server, Use NodeJs File System Module to read HTML file.

HTML file contains.

- Headings
- Paragraphs
- List Items etc

used fs.readFile method.

Procedure:

Step1: create folder with any name for nodejs project

Step2: create file-name called "index.html" and produce code.

Step3: create file-name called "Node.js" in same folder and produce code.

Source code

Index.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta charset="utf-8" />
<title>The Page Returned by Making Http Call to
Node.js</title>
```

```
<style type="text/css">
table, td {
border:double;
}
</style>
</head>
<body>
<h1>Item List</h1>
<p>In this project data is collected from client to
Server as response</p>
<table>
<tr>
<td>Item 01:</td>
<td>
<input type="text" />
</td>
</tr>
<tr>
<td>Item 02:</td>
<td>
<input type="text" />
</td>
</tr>
<tr>
<td>Item 03:</td>
<td>
<input type="text" />
</td>
</tr>
<tr>
<td>Item 04:</td>
<td>
<input type="text" />
</td>
</tr>
```

```
    </tr>
<tr>
<td></td>
<td>
<input type="button" value="Save"/>
</td>
</tr>
</table>
</body>
</html>
```

Node.js

```
var http = require('http');
var fs = require('fs');
//2.
var server = http.createServer(function (req, resp) {
//3.
if (req.url === "/create") {
fs.readFile("./index.html", function (error, pgResp) {
if (error) {
resp.writeHead(404);
resp.write('Contents you are looking are Not Found');
} else {
resp.writeHead(200, { 'Content-Type': 'text/html' });
resp.write(pgResp);
}
resp.end();
});
} else {
//4.
resp.writeHead(200, { 'Content-Type': 'text/html' });
resp.write('<h1>This is a initial page goto URL address
below </h1><p>This page as recivied from Server to Client
```

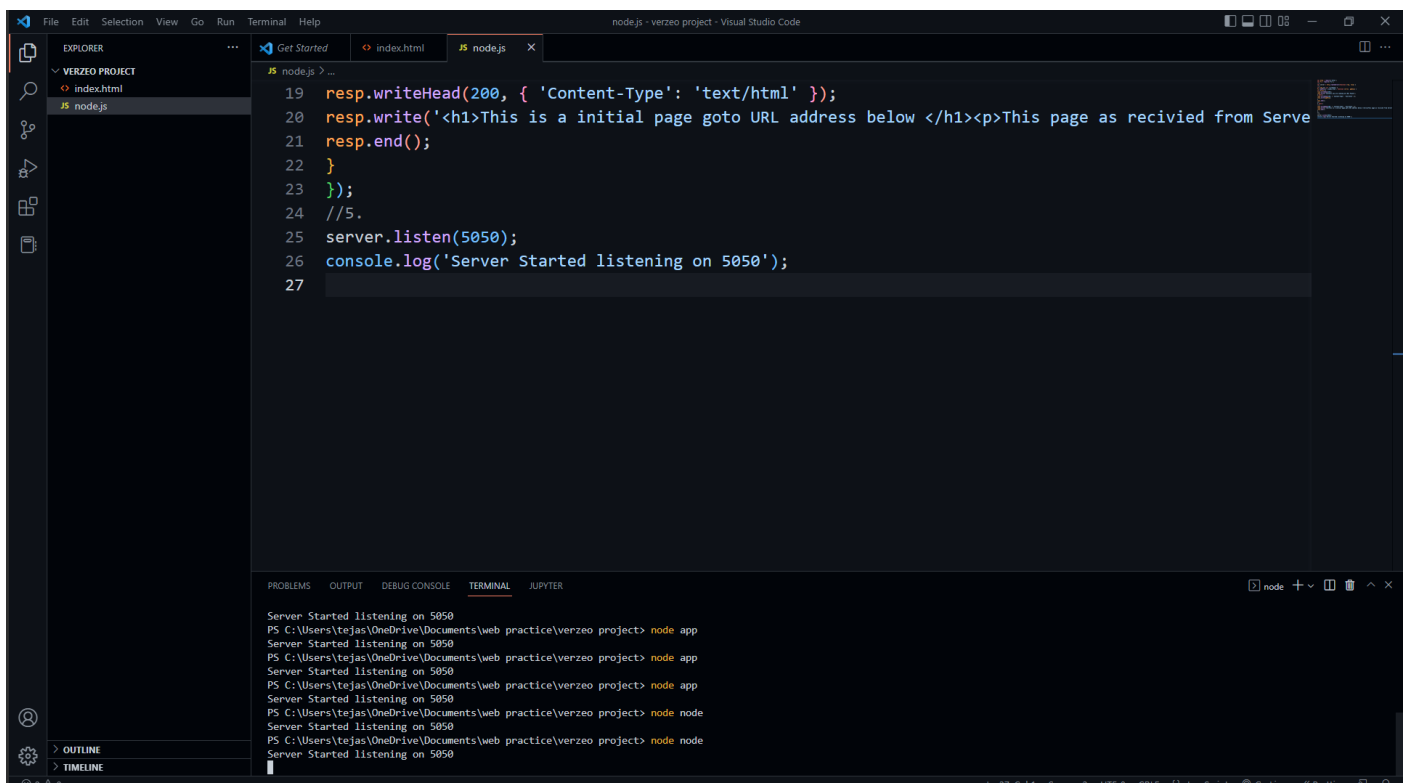
```
via "localhost:5050" port number. Enter below Address in
the url bar</p><br /><br />URL Address =>
localhost:5050/create: ' + req.url);
resp.end();
}
});
//5.
server.listen(5050);
console.log('Server Started listening on 5050');
```

Output:

1) On the terminal enter as->node file_name

i.e

node node



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows a project named 'VERZEO PROJECT' with files 'index.html' and 'node.js'. The main editor displays the contents of 'node.js', which is a JavaScript file for a simple web server. The code includes a response header, HTML content, and a listener on port 5050. The bottom panel shows the 'TERMINAL' tab with the output of running 'node app' multiple times, each time displaying 'Server Started listening on 5050'.

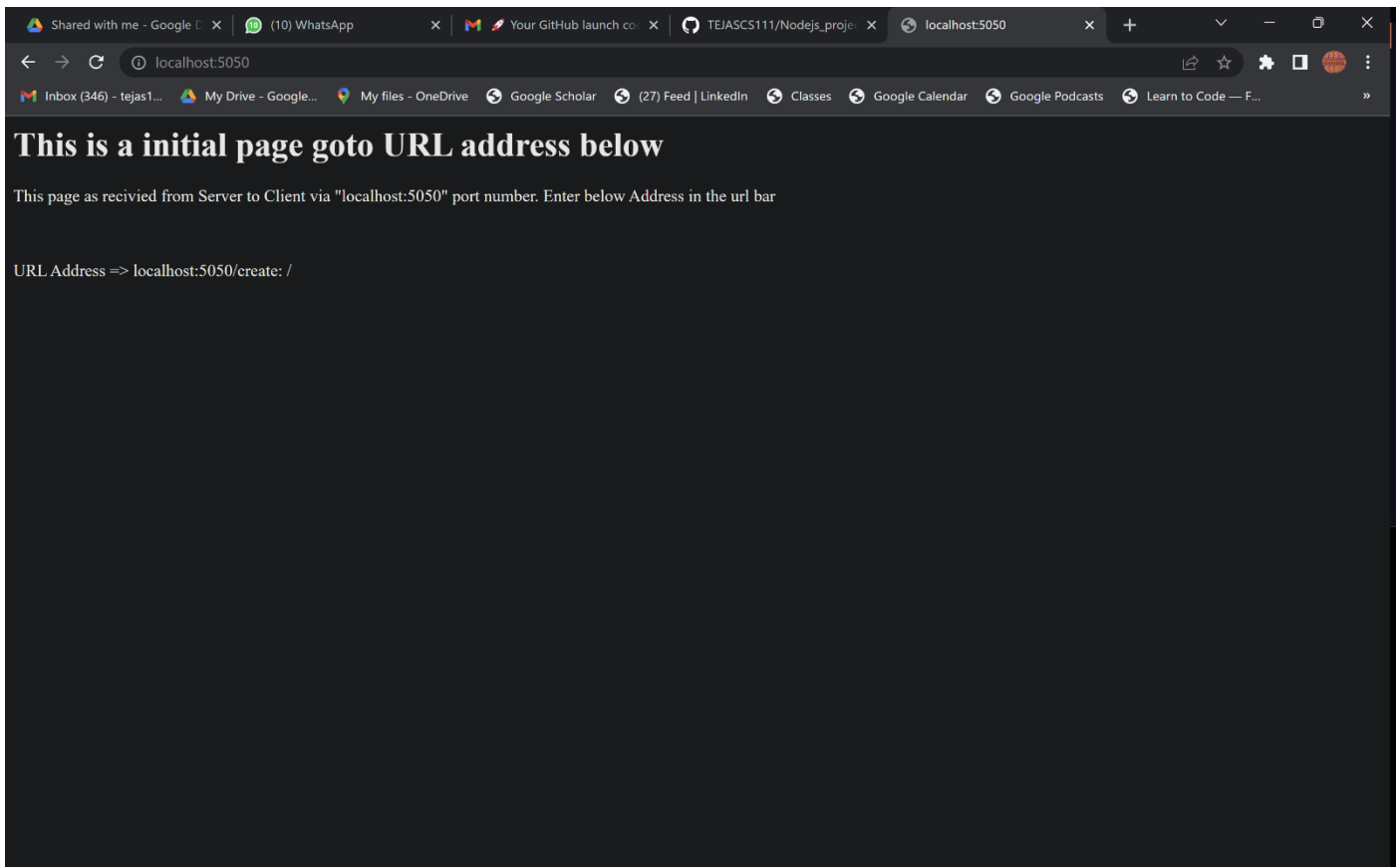
```
19 resp.writeHead(200, { 'Content-Type': 'text/html' });
20 resp.write('<h1>This is a initial page goto URL address below </h1><p>This page as recieved from Serve
21 resp.end();
22 }
23 });
24 //5.
25 server.listen(5050);
26 console.log('Server Started listening on 5050');
27
```

Server Started listening on 5050
PS C:\Users\tejas\OneDrive\Documents\web practice\verzeo project> node app
Server Started listening on 5050
PS C:\Users\tejas\OneDrive\Documents\web practice\verzeo project> node app
Server Started listening on 5050
PS C:\Users\tejas\OneDrive\Documents\web practice\verzeo project> node app
Server Started listening on 5050
PS C:\Users\tejas\OneDrive\Documents\web practice\verzeo project> node app
Server Started listening on 5050
PS C:\Users\tejas\OneDrive\Documents\web practice\verzeo project> node node
Server Started listening on 5050
PS C:\Users\tejas\OneDrive\Documents\web practice\verzeo project> node node
Server Started listening on 5050

confirmation line:

Server started listening on 5050

2)On client software(Browser) Page:"localhost:5050/"



3)localhost:5050/create

Item List

In this project data is collected from client to Server as response

Item 01:	
Item 02:	
Item 03:	
Item 04:	
	Save