

A screenshot of the Visual Studio Code (VS Code) interface. The top half shows a code editor with a dark theme, displaying a file named `HelloWorld.js`. The code is a simple Node.js application that creates a web server on port 8080, responding with 'Hello World!' in HTML. The bottom half shows a terminal window with the same content, indicating the application is running.

JS HelloWorld.js U X

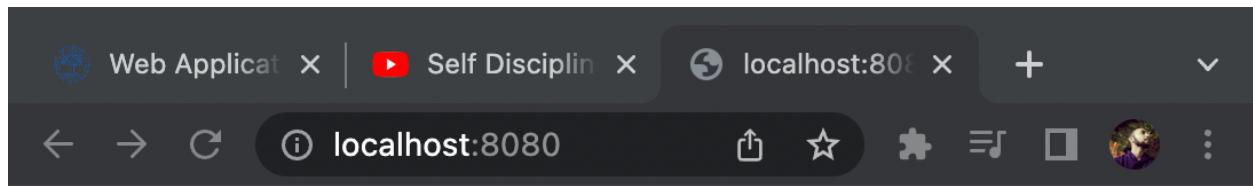
Labs12 > NodeFolder > `JS` HelloWorld.js > ...

```
1  var http = require('http');
2
3  ^ http.createServer(function(req, res){
4      res.writeHead(200, {'Content-Type': 'text/html'})
5      res.end('Hello World!');
6  }).listen(8080);
```

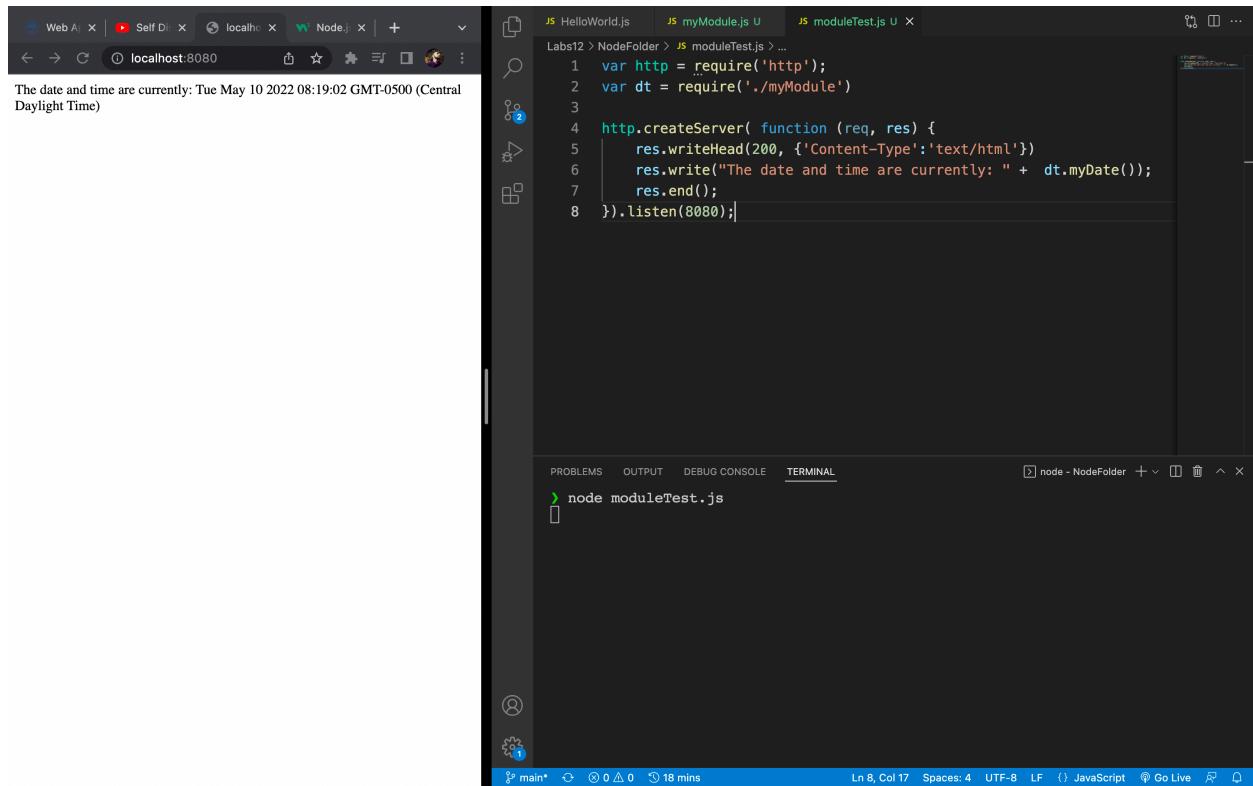
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

node - NodeFolder + X

```
› node HelloWorld.js
```



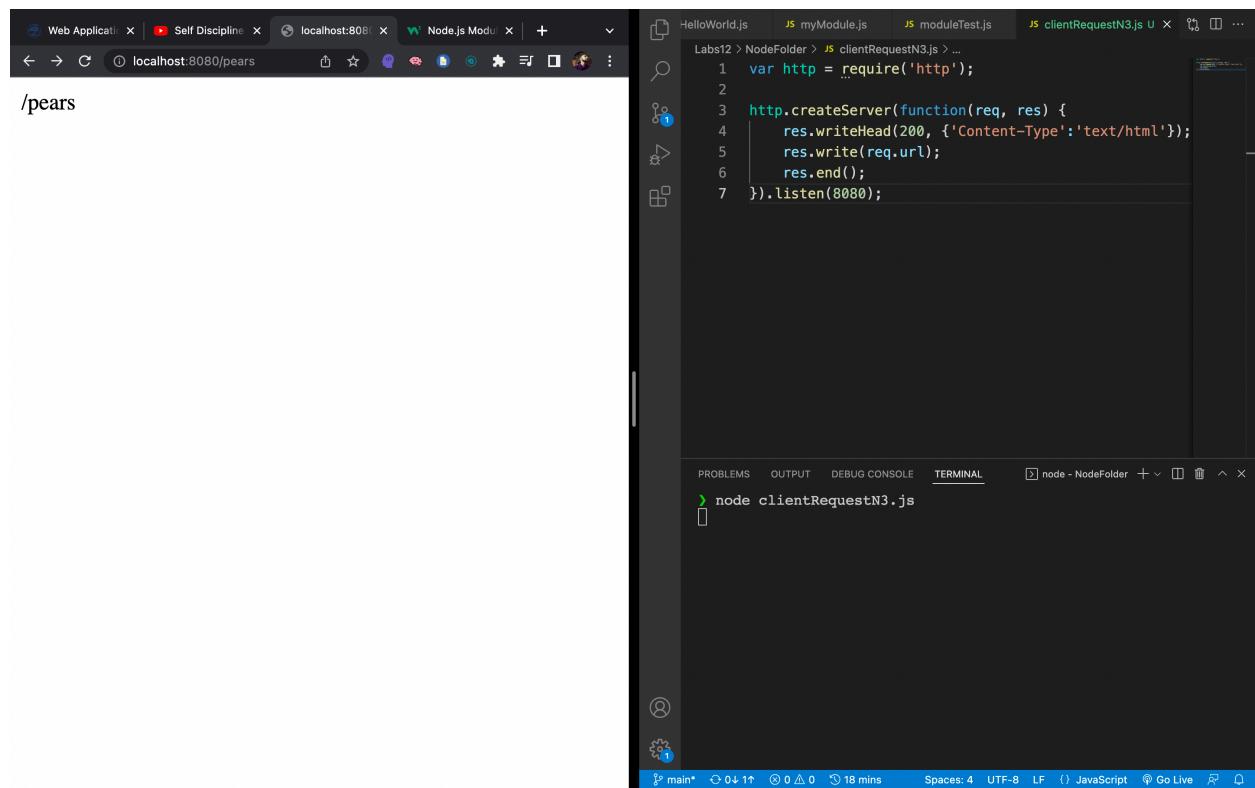
Hello World!



The image shows a dark-themed interface, likely from a code editor like VS Code. At the top, there are three tabs: `JS HelloWorld.js`, `JS myModule.js U X`, and `JS moduleTest.js U`. The `myModule.js` tab is active. Below the tabs, a file structure is displayed: `Labs12 > NodeFolder > JS myModule.js > ...`. The content of `myModule.js` is as follows:

```
1  function myDate () {
2    return Date();
3  };
4
5  exports.myDate = myDate;
```

On the right side of the editor, there is a small sidebar with icons for file operations. Below the editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is active, indicated by an underline. The terminal window shows the command `node moduleTest.js` being run, with a green arrow icon preceding the command. The terminal window has a close button (X) and a refresh button (refresh icon).

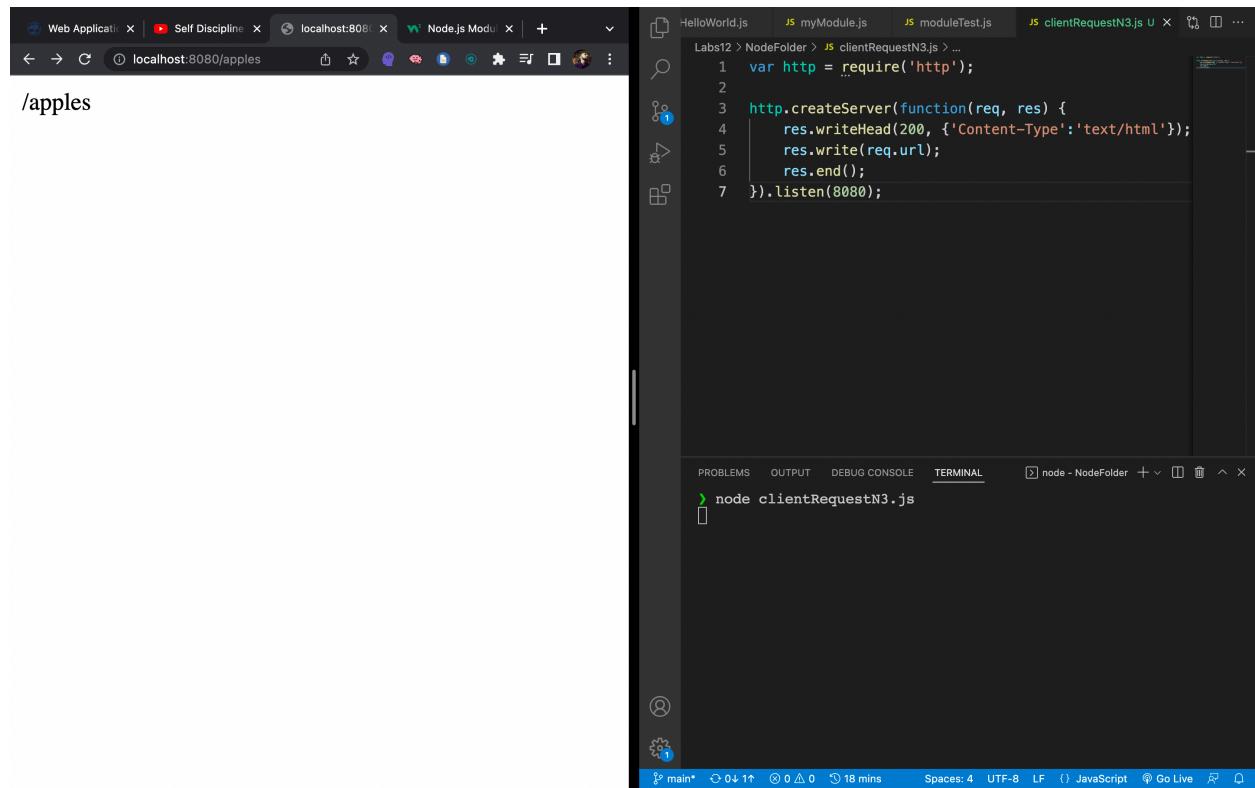


localhost:8080/pears

```
1 var http = require('http');
2
3 http.createServer(function(req, res) {
4     res.writeHead(200, {'Content-Type':'text/html'});
5     res.write(req.url);
6     res.end();
7 }).listen(8080);
```

TERMINAL

```
node clientRequestN3.js
```



localhost:8080/apples

```
1 var http = require('http');
2
3 http.createServer(function(req, res) {
4     res.writeHead(200, {'Content-Type':'text/html'});
5     res.write(req.url);
6     res.end();
7 }).listen(8080);
```

TERMINAL

```
node clientRequestN3.js
```

The screenshot shows a Node.js development environment in VS Code. The terminal at the bottom displays the command `node clientRequestN3.js`. The code editor on the right shows a file named `clientRequestN3.js` with the following content:

```
1 var http = require('http');
2
3 http.createServer(function(req, res) {
4   res.writeHead(200, {'Content-Type':'text/html'});
5   res.write(req.url);
6   res.end();
7 }).listen(8080);
```

The browser tab at the top shows the URL `localhost:8080/strawberries`, which corresponds to the output of the running server.

The screenshot shows a development environment with a browser window and a code editor.

Browser Tab: The browser tab shows the URL `localhost:8080/add.j...`. The page content is "The sum is: 20".

Code Editor: The code editor displays two files:

```
JS addmod.js U JS AdderWebServer.js U SimpleAdder.html U
Labs12 > NodeFolder > JS AdderWebServer.js > http.createServer() callback
1 var http = require('http');
2 var url = require('url');
3 var fs = require('fs');
4 var addmod = require('./addmod.js');
5
6 http.createServer(function (req, res) {
7   var q = url.parse(req.url, true);
8   var filename = "." + q.pathname;
9   if (q.pathname == "/add.js")
10     addmod.add(req, res, q.query)
11   else
12     fs.readFile(filename, function (err, data) {
13       if (err) {
14         res.writeHead(404, { 'Content-Type': 'text/html' })
15         return res.end("404 Not Found");
16       }
17       res.writeHead(200); // Content-Type not included
18       res.write(data);
19       return res.end();
20     });
21 }).listen(8080);
```

Terminal: The terminal shows the command `node AdderWebServer.js` being run.

Bottom Status Bar: The status bar shows the file `main*`, line `Ln 20, Col 12`, spaces `Spaces: 4`, encoding `UTF-8 LF`, and port `Port: 5500`.

Labs12 > NodeFolder > **JS** addmod.js > ...

```
1  exports.add = function (req, res, vals) {
2    var sum = parseInt(vals.first) + parseInt(vals.second);
3    res.writeHead(200, { 'Content-Type': 'text/html' });
4    res.write(`<!DOCTYPE html>
5    <html>
6      <head><meta charset=\"utf-8\"/>
7      <title>Calculator Web Site</title>
8    </head>
9    <body>
10   <p>The sum is: ${String(sum)}</p>
11   </body>
12 </html> ` );
13   return res.end();
14 };
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

node - NodeFolder + ×

```
❯ node AdderWebServer.js
```

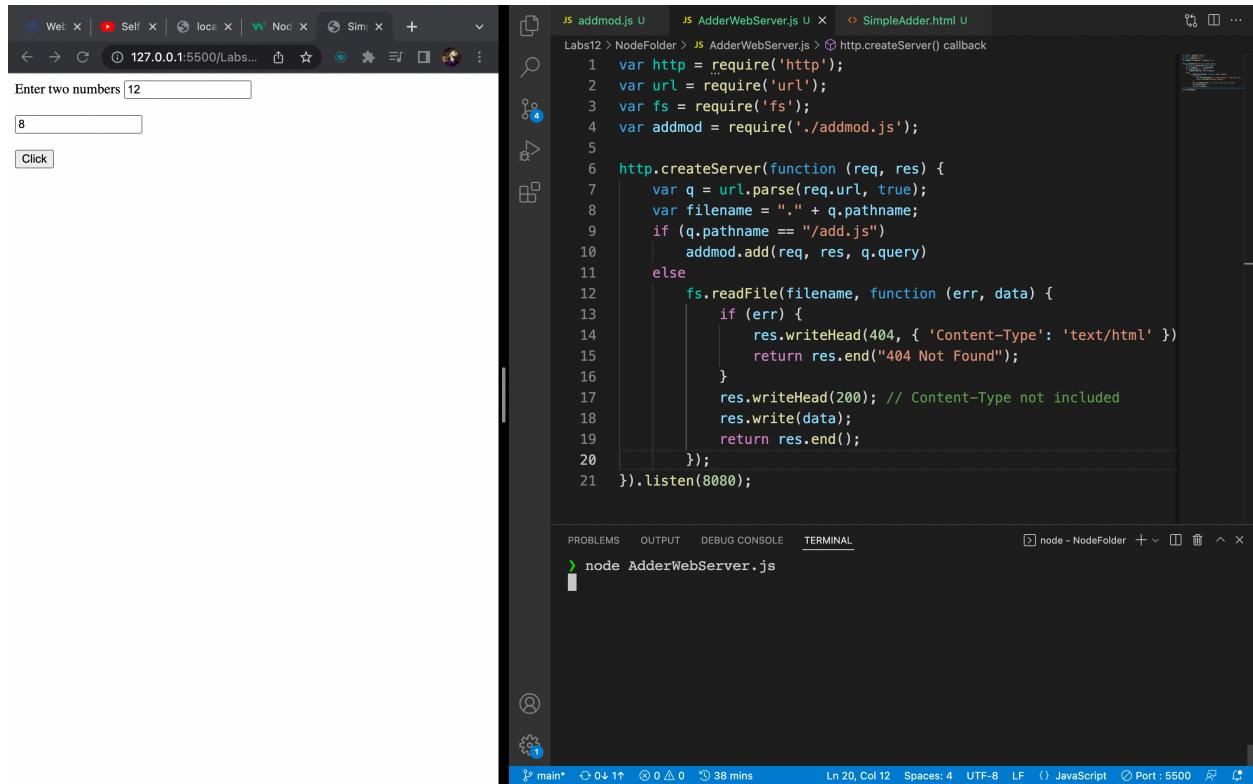
VS Code interface showing a Node.js application structure:

- Files: addmod.js, AdderWebServer.js, SimpleAdder.html
- SimpleAdder.html content (highlighted):

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Simple Adder</title>
9  </head>
10 <body>
11     <form action="http://localhost:8080/add.js">
12         <div>Enter two numbers
13             <input type="text" name="first" /><br /><br />
14             <input type="text" name="second" /><br /><br />
15             <input type="submit" value="Click" />
16         </div>
17     </form>
18 </body>
19
20 </html>
```

Terminal output:

```
node AdderWebServer.js
```



The screenshot shows a development environment with a browser window and a code editor.

Browser View: The browser window displays a simple form with an input field containing "8" and a button labeled "Click". The URL is 127.0.0.1:5500/Labs... .

Code Editor (VS Code): The code editor shows two files:

```
JS addmod.js U JS AdderWebServer.js U SimpleAdder.html U
Labs12 > NodeFolder > JS AdderWebServer.js > http.createServer() callback
1 var http = require('http');
2 var url = require('url');
3 var fs = require('fs');
4 var addmod = require('./addmod.js');
5
6 http.createServer(function (req, res) {
7   var q = url.parse(req.url, true);
8   var filename = "." + q.pathname;
9   if (q.pathname == "/add.js")
10     addmod.add(req, res, q.query)
11   else
12     fs.readFile(filename, function (err, data) {
13       if (err) {
14         res.writeHead(404, { 'Content-Type': 'text/html' })
15         return res.end("404 Not Found");
16       }
17       res.writeHead(200); // Content-Type not included
18       res.write(data);
19       return res.end();
20     });
21 }).listen(8080);
```

Terminal: The terminal shows the command: node AdderWebServer.js

Bottom Status Bar: The status bar indicates the file is main*, line 20, column 12, spaces: 4, encoding: UTF-8, LF, JavaScript, and port: 5500.