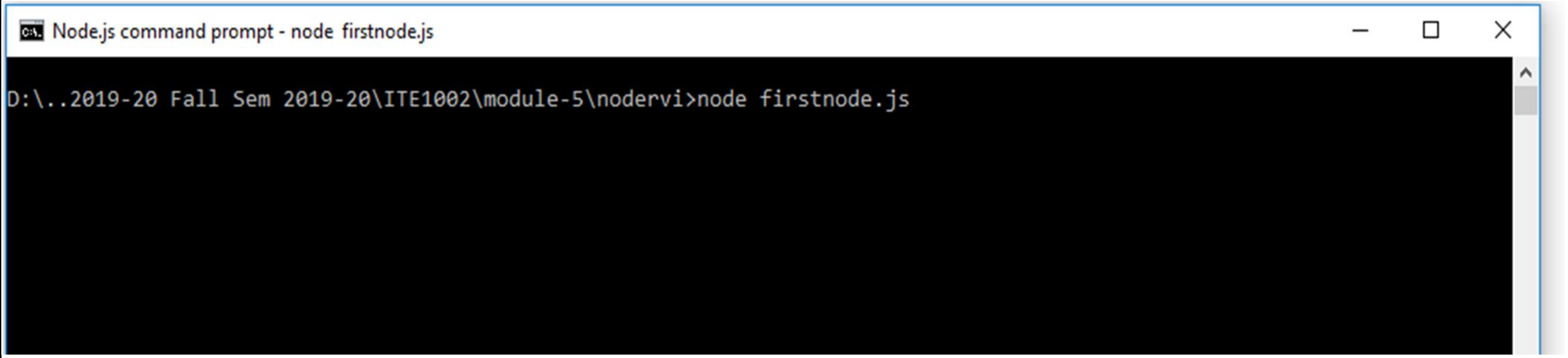
C:\Users\admin\firstnode.js

```
var http = require('http');
http.createServer(function (req, res)
{
    res.writeHead(200,{'Content-Type':'text/html'});
    res.end('Hello vijayan Your Node JS Server is ready!');
}).listen(8080);
```



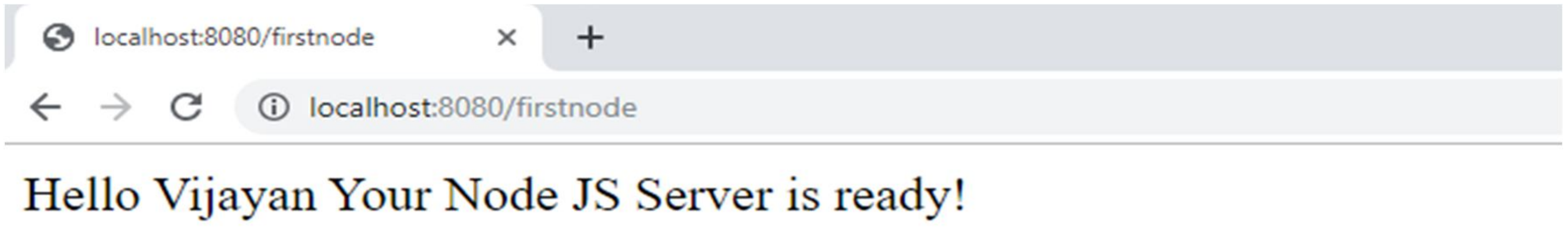
```
D:\..2019-20 Fall Sem 2019-20\ITE1002\module-5\nodervi>node firstnode.js
```

firstnode.js - explanation

- The basic functionality of the "require" function is that it reads a JavaScript file, executes the file, and then proceeds to return an object. Using this object, one can then use the various functionalities available in the module called by the require function. So in our case, since we want to use the functionality of http and we are using the require(http) command.
- In this 2nd line of code, we are creating a server application which is based on a simple function. This function is called, whenever a request is made to our server application.
- When a request is received, we are asking our function to return a 'Hello vijayan Your Node JS Server is ready!' response to the client. The writeHead function is used to send header data to the client and while the end function will close the connection to the client.
- We are then using the server.listen function to make our server application listen to client requests on port no 8080. You can specify any available port over here.

Executing the code

- Save the file on your computer: C:\Users\admin\firstnode.js
- In the command prompt, navigate to the folder where the file is stored. Enter the command node node.js
- Now, your computer works as a server! If anyone tries to access your computer on port 8080, they will get a "Hello vijayan Your Node JS Server is ready!" message in return!
- Start your internet browser, and type in the address: http://localhost:8080



```
Hello Vijayan Your Node JS Server is ready!
```

What are modules in Node.js?

- Consider modules to be the same as JavaScript libraries. A set of functions you want to include in your application.
- Modules in Node js are a way of encapsulating code in a separate logical unit. There are many readymade modules available in the market which can be used within Node js.
- Below are some of the popular modules which are used in a Node js application
- **Express framework** – Express is a minimal and flexible Node js web application framework that provides a robust set of features for the web and Mobile applications.
- **Socket.io** - Socket.IO enables real-time bidirectional event-based communication. This module is good for creation of chatting based applications.
- **Jade** - Jade is a high-performance template engine and implemented with JavaScript for node and browsers.
- **MongoDB** - The MongoDB Node.js driver is the officially supported node.js driver for MongoDB.
- **Restify** - restify is a lightweight framework, similar to express for building REST APIs
- **Bluebird** - Bluebird is a fully featured promise library with focus on innovative features and performance

Procedure in Lab

- Save program in /home/mock3/node.js
- Run in the command prompt like node node.js
- Open the web browser and type localhost:8080/ you will get the output.

Node.js HTTP Module

The Built-in HTTP Module

- Node.js has a built-in module called HTTP, which allows Node.js to transfer data over the Hyper Text Transfer Protocol (HTTP).
- To include the HTTP module, use the require() method:
- var http = require('http');

Node.js as a Web Server

- The HTTP module can create an HTTP server that listens to server ports and gives a response back to the client.
- Use the createServer() method to create an HTTP server.
- The function passed into the http.createServer() method, will be executed when someone tries to access the computer on port 8080.

Add an HTTP Header

- If the response from the HTTP server is supposed to be displayed as HTML, you should include an HTTP header with the correct content type.
- The first argument of the res.writeHead() method is the status code, 200 means that all is OK, the second argument is an object containing the response headers.

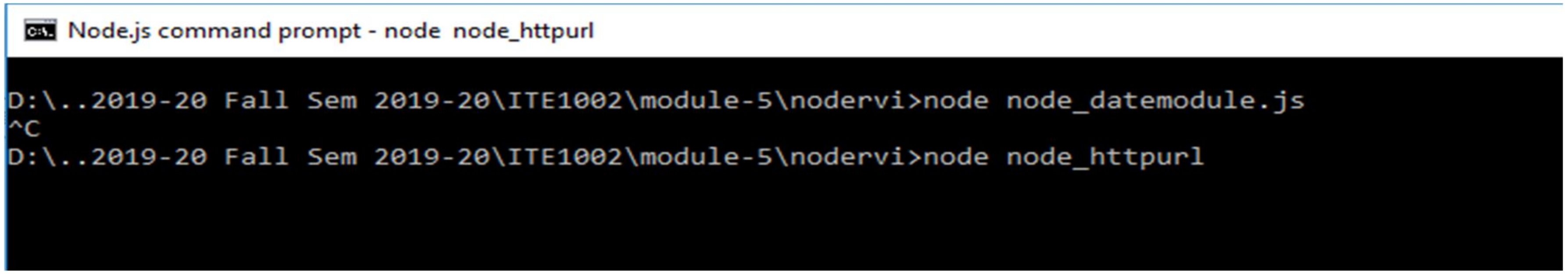
Read the Query String

- The function passed into the http.createServer() has a req argument that represents the request from the client, as an object (http.IncomingMessage object).
- This object has a property called "url" which holds the part of the url that comes after the domain name:

C:\Users\admin\node_httpurl.js

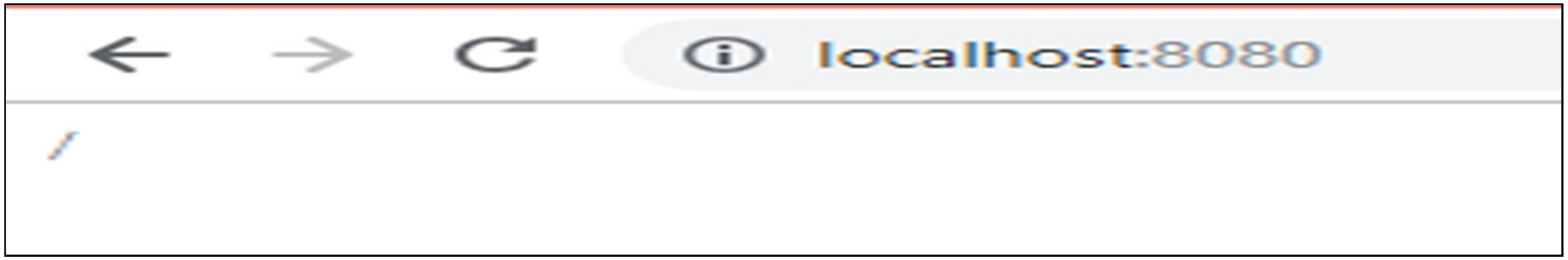
```
var http = require('http');
http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(req.url);
    res.end();
}).listen(8080);
```

Initiate the node js server


Node.js command prompt - node node_httpurl

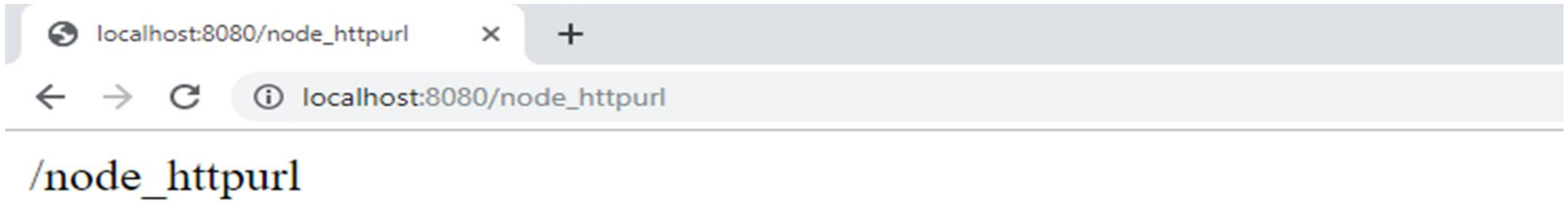
```
D:\..2019-20 Fall Sem 2019-20\ITE1002\module-5\nodervi>node node_datemodule.js
^C
D:\..2019-20 Fall Sem 2019-20\ITE1002\module-5\nodervi>node node_httpurl
```

Run the browser


localhost:8080

```
/
```

Run the browser with adding contents in url


localhost:8080/node_httpurl

```
/node_httpurl
```

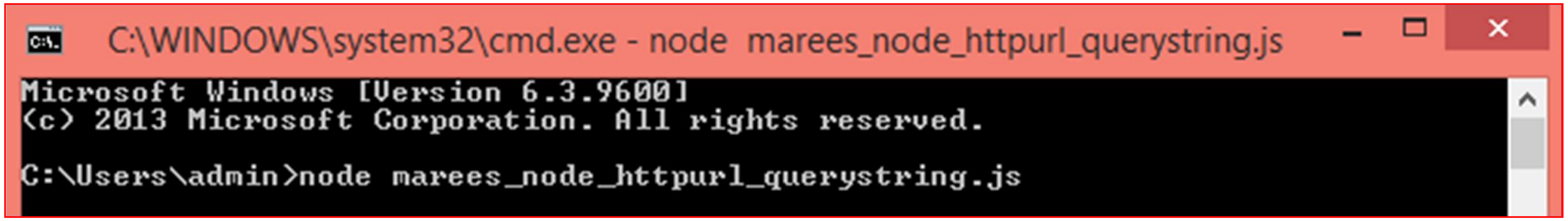
C:\Users\admin\node_httpurl_querystring.js

Split the Query String

```
var http = require('http');
var url = require('url');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  var q = url.parse(req.url, true).query;
  var txt = q.year + " " + q.month;
var txt=q.host+ " " +q.pathname+ " " +q.search;
  res.end(txt);
}).listen(8080);
```

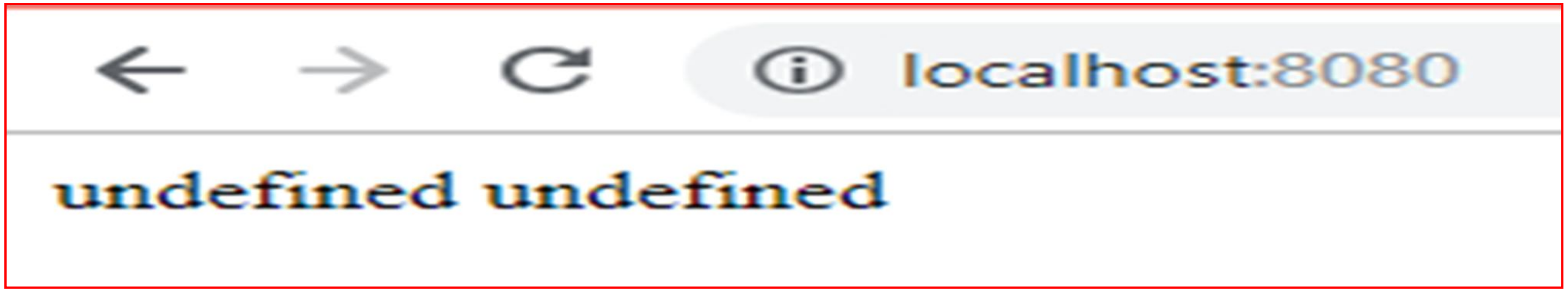
There are built-in modules to easily split the query string into readable parts, such as the URL module.

Initiate the node js server


C:\WINDOWS\system32\cmd.exe - node marees_node_httpurl_querystring.js

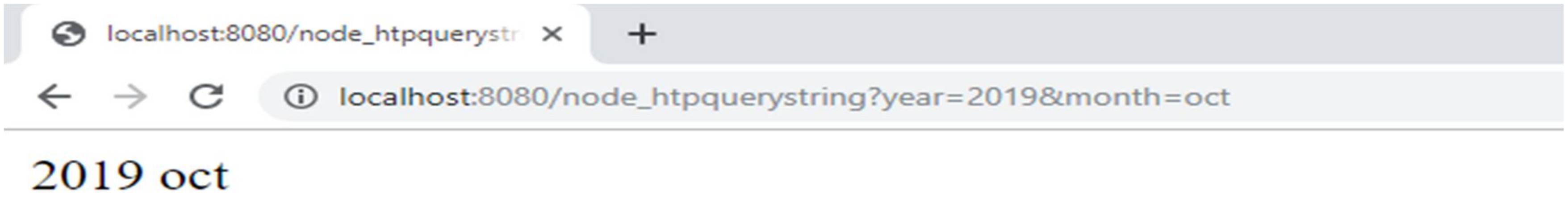
```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\admin>node marees_node_httpurl_querystring.js
```

Run the browser



```
undefined undefined
```

Run the browser with adding query string by ? in url


localhost:8080/node_htpquerystr
localhost:8080/node_htpquerystring?year=2019&month=oct

```
2019 oct
```

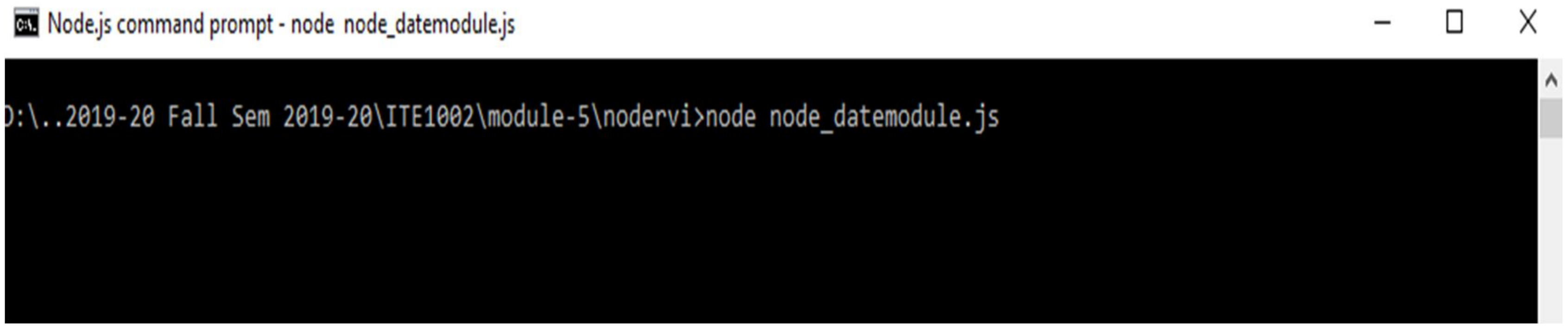
C:\Users\admin\datemodule.js

```
exports.myDateTime = function () {
    return Date();
};
```

Use the exports keyword to make properties and methods available outside the module file.

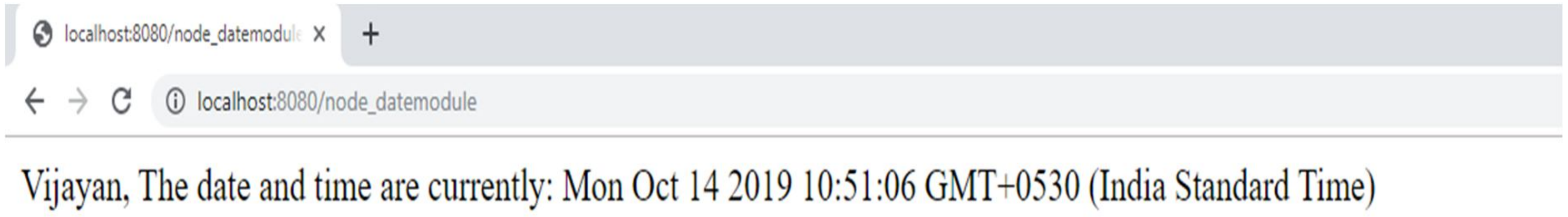
C:\Users\admin\node_datemodule.js

```
var http = require('http');
var dt = require('./datemodule');
http.createServer(function (req, res) {
    res.writeHead(200,{'Content-Type':'text/html'});
    res.write("Vijayan, The date and time are currently: " +
dt.myDateTime());
    res.end();}).listen(8080);
```



Node.js command prompt - node node_datemodule.js

```
D:\..2019-20 Fall Sem 2019-20\ITE1002\module-5\nodervi>node node_datemodule.js
```



localhost:8080/node_datemodule

```
Vijayan, The date and time are currently: Mon Oct 14 2019 10:51:06 GMT+0530 (India Standard Time)
```

Node.js File System Module

- The Node.js file system module allows you to work with the file system on your computer.
- To include the File System module, use the require() method:

```
var fs = require('fs');
```

- Common use for the File System module: Read files,Create files, Update files, Delete files and Rename files.
- The fs.readFile() method is used to read files on your computer.

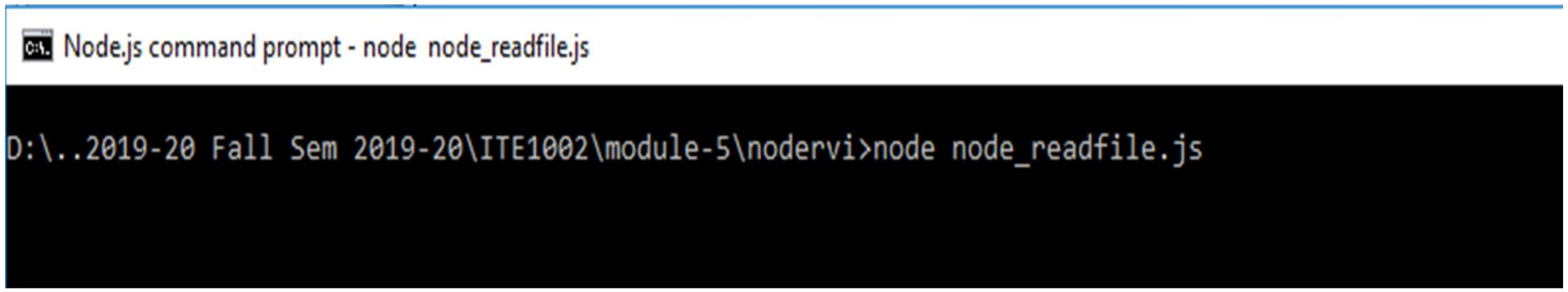
C:\Users\admin\readfile1.html

```
<html>
<body>
<h1> My First Page</h1>
<p> Hi, everybody.....<br>I am Vijayan </p>
</body>
</html>
```

C:\Users\admin\node_readfile.js

```
var http = require('http');
var fs = require('fs');
http.createServer(function (req, res)
{
 fs.readFile('readfile1.html', function(err, data) {
   res.writeHead(200, {'Content-Type': 'text/html'});
   res.write(data);
   res.end();
 });}).listen(8080);
```

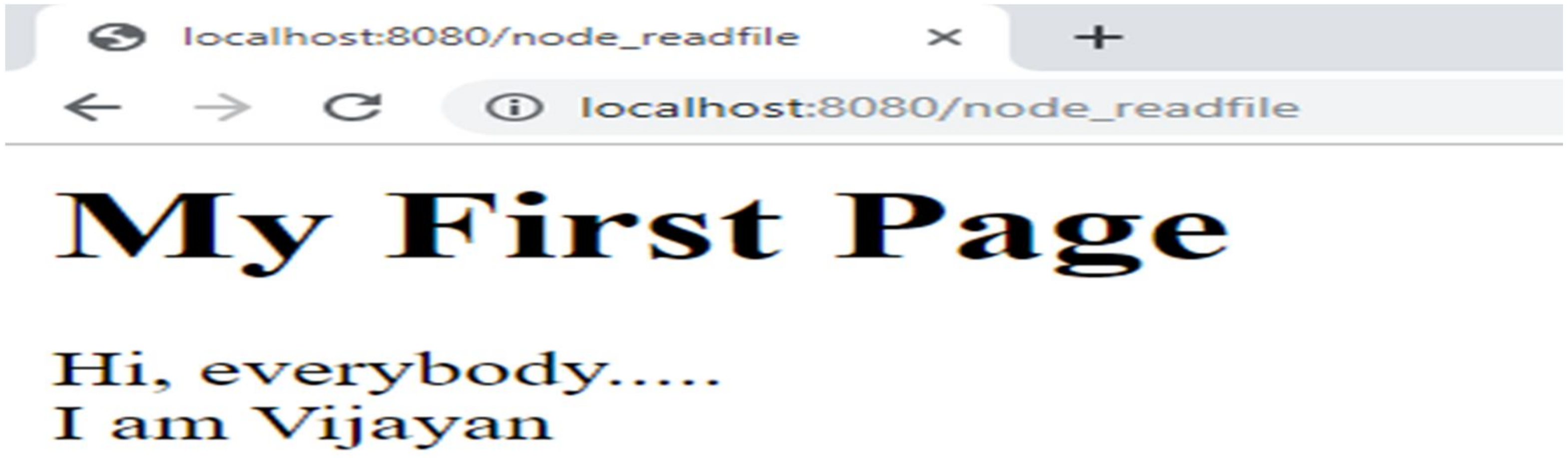
Initiate node_readfile.js



Node.js command prompt - node node_readfile.js

```
D:\..2019-20 Fall Sem 2019-20\ITE1002\module-5\nodervi>node node_readfile.js
```

Run in browser



My First Page

Hi, everybody.....
I am Vijayan