Experiment -6

Aim: To implement Module in NodeJs.

Description:

A module in Node.js is a collection of independent and reusable code that can be imported into any Node.js application. As the name suggests, modules enable a modular and structured approach for developing a Node.js application. Instead of putting all the functions, classes and methods of an application in a single .js file, these resources are arranged in separate files (called modules) based on their relevance. This gives a better control over the maintenance and troubleshooting of the Node.js application.

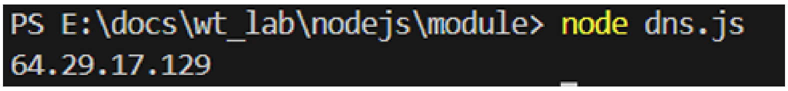
In Node.js, modules are categorized into two types:

1. Default (Built-in) Modules – Pre-installed modules provided by Node.js.

2. User-defined Modules – Custom modules created by developers.

Dns Module:

```
const dns=require('dns');
 dns.lookup('natasha-portfolio-rho.vercel.app',(err,value)=>{
     if(err){
         console.log(err);
         return;
    }
  console.log(value);
})
```

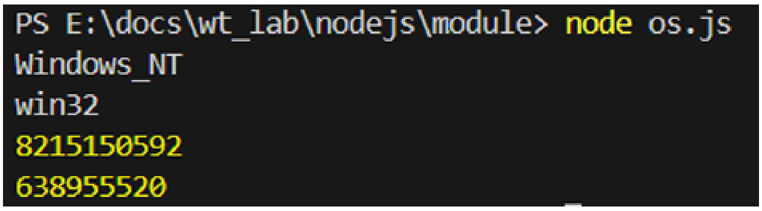


```
PS E:\docs\wt_lab\nodejs\module> node dns.js
64.29.17.129
```

Os module:

```
var os=require("os");
console.log(os.type());
console.log(os.platform());
console.log(os.totalmem());
```

console.log(os.freemem());

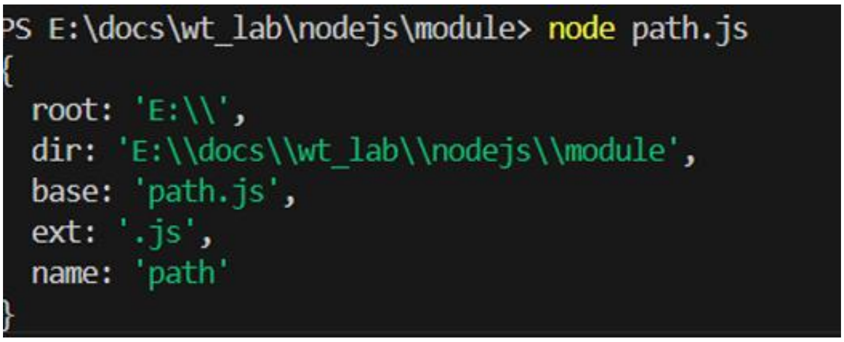


```
PS E:\docs\wt_lab\nodejs\module> node os.js
Windows_NT
win32
8215150592
638955520
```

Path module:

var path=require('path')

console.log(path.parse(__filename));



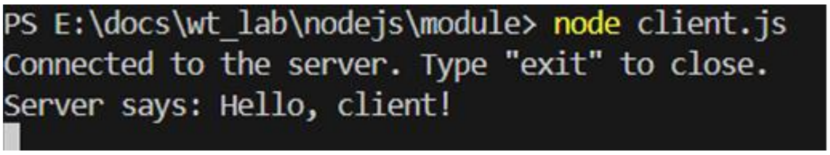
```
PS E:\docs\wt_lab\nodejs\module> node path.js
{
  root: 'E:\\',
  dir: 'E:\\docs\\wt_lab\\nodejs\\module',
  base: 'path.js',
  ext: '.js',
  name: 'path'
}
```

Client.js

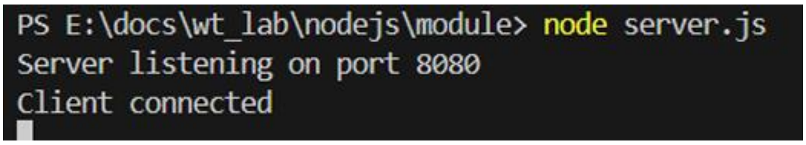
```
const net = require('net');
const readline = require('readline');

const client = net.createConnection({ port: 8080 }, () => {
  console.log('Connected to the server. Type "exit" to close.');
});

const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

client.on('data', (data) => {
  console.log('Server says:', data.toString().trim());
});

rl.on('line', (line) => {
  if (line.trim().toLowerCase() === 'exit') {
    client.end();
    rl.close();
  } else {
    client.write(line + '\r\n');
  }
});

client.on('end', () => {
  console.log('Disconnected from server');
});
```

Server.js

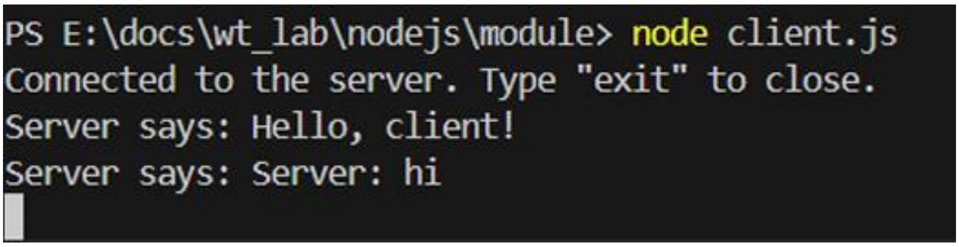
```
const net = require('net');
const readline = require('readline');

const client = net.createConnection({ port: 8080 }, () => {
  console.log('Connected to the server. Type "exit" to close.');
});

const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

client.on('data', (data) => {
  console.log('Server says:', data.toString().trim());
});

rl.on('line', (line) => {
  if (line.trim().toLowerCase() === 'exit') {
    client.end();
    rl.close();
  } else {
    client.write(line + '\r\n');
  }
});

client.on('end', () => {
  console.log('Disconnected from server');
});
```



```
PS E:\docs\wt_lab\nodejs\module> node client.js
Connected to the server. Type "exit" to close.
Server says: Hello, client!
```



```
PS E:\docs\wt_lab\nodejs\module> node server.js
Server listening on port 8080
Client connected
```



```
PS E:\docs\wt_lab\nodejs\module> node client.js
Connected to the server. Type "exit" to close.
Server says: Hello, client!
Server says: Server: hi
```