Experimentno:-6 Date:-

Node.js

Course Name: Node.js
Module Name: How to use Node.js

Verify how toexecute different functions successfully in the Node.js platform.

Aim: Learning about use of Node.js and verifying how to execute different functions successfully in the Node.js platform.

Sample.js

```
console.log(“Node.js program to proceed”);
```

```
C:\Users\giridhar>cd nodejs
C:\Users\giridhar\nodejs>node sample.js
Node.js program to proceed
```

Program:-

```
function tester()
{
var  m=10; var
message;if
(m%2==0)
{
message = "m is an even number";
}
else
{
message = "m is not an even number";
}
console.log(message);
}
tester();
```

Output:-

```
PS C:\python programs\mean stack> node week6a.jsm is
an even number
```

Exp No : 6.b Date :

Course Name: Node.js

Module Name: Create a web server in Node.js

Write a program to show the workflow of JavaScript code executable by creating web server in Node.js.

Aim: Creating a web server in Node.js and showing the workflow of JavaScript code executable by creating web server in Node.js.

Program:-

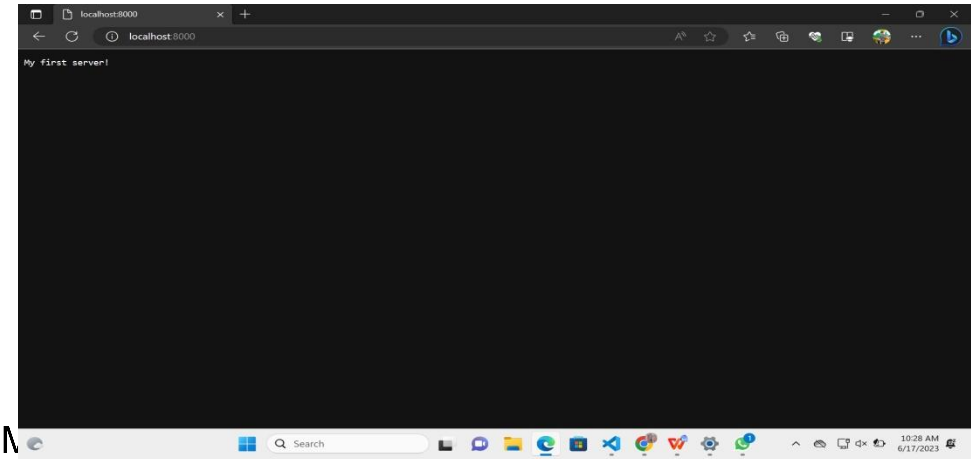
```
const http = require("http");
const host = 'localhost'; const
port = 8000;
const requestListner = function(req, res) {



    res.end("My first server!");
};
const server = http.createServer(requestListner);
server.listen(port, host, () => {
    console.log(`Server is running on http://${host}:${port}`);
})
```

Output:-

```
PS C:\python programs\mean stack> node week6b.js
Server is running on http://localhost:8000
```



```
My first server!
```

Exp No : 6.c Date :

Course Name: Node.js

Module Name: Modular programming in Node.js

Write aNode.js module to show the workflow of Modularization of Node application.

Aim: Write a Node.js module to show the workflow of Modularization of Node application

Program:-

Module.js

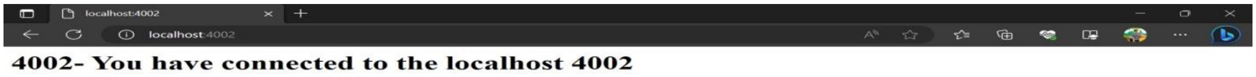
```
exports.authenticateUser = (a, b) =&gt;
{return a+b;
};
```

Auth.js

```
const http = require("http");
var dbmodule = require("./giri");
var server = http.createServer((request, response) => { result
= dbmodule.authenticateUser(4000,2);
response.writeHead(200, { "Content-Type": "text/html" });
response.end("<html><body><h1>" + result + "- You have connected to the localhost4002
</h1></body></html>");
console.log("Request received"); });
server.listen(4002);
console.log("Server is running at port 4002");
```

Output:-

```
PS C:\python programs\mean stack> node Auth.jsServer is
running at port 4002
Request received
Request received
```



4002- You have connected to the localhost 4002

Exp No : 6.d Date :

Course Name: Node.js

Module Name: Restarting Node Application

Writea program to show the workflow of restarting a Node application.

Aim: Program to show the workflow of restarting a node application.

Nodemon

To install it in the application, run the below command.npm
install nodemon –g

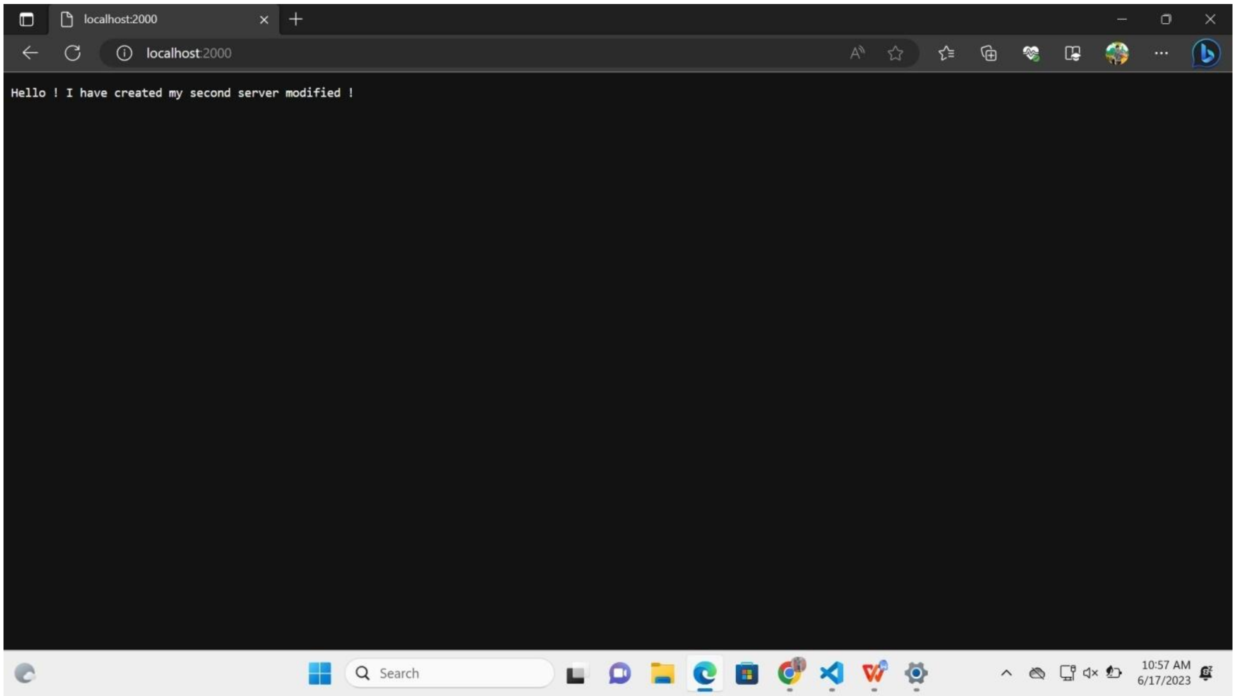
Program:-

```
const http = require("http");
var server = http.createServer((req, res) => {
    res.write("Hello ! I have created my second server modified giri!");
     res.end();
});
server.listen(3000);
console.log("Server started... Running on localhost:3000");
```

Output:-

```
PS C:\python programs\mean stack> nodemon week6c.js[nodemon]
2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node week6c.js` Server
started... Running on localhost:5000
```

Output in the browser:



```
Hello ! I have created my second server modified !
```

Exp No : 6.e Date :

Course Name: Node.js

Module Name: File Operations

Create a text filesrc.txt and add the following data to it. Mongo, Express, Angular, Node. Aim:To create a text file src.txt and add the following data to it. Progrm:-

```
// Node.js program to demonstrate the
// fs.copyFile() method

// Import the filesystem module
const fs = require('fs');

// Get the current filenames
// before the function
getCurrentFilenames();
console.log("\nFile Contents of source.txt:",
  fs.readFileSync("source.txt", "utf8"));

// Copying the file to a the same name
fs.copyFile("source.txt", "target.txt", (err) => {if
(err) {
    console.log("Error Found:", err);
  }
  else {

    // Get the current filenames
    // after the function
    getCurrentFilenames();
    console.log("\nFile Contents of copied_file:",
      fs.readFileSync("target.txt", "utf8"));
  }
});

// Function to get current filenames
// in directory
function getCurrentFilenames()
  { console.log("\nCurrent filenames:");
  fs.readdirSync(___dirname).forEach(file =>
    {console.log(file);
  });
}
```

Output:-

```
PS C:\python programs\mean stack> node week6e.js

Current filenames:
colour.css
colour.html
CSE.html
download.jpg durga-
devi-png.png
exam1.html
exam2.html
exam3.html
exam4.html
first.html
FLS-Blog-Black-Logos_Hero.jpg
function.js
giri.jpg
giri.js
ice.jpg
images.jpg
lukcy.mp4
ma_platinum_200_iso_white.jpg
mo.jpeg
mystyle.css
node_modules
package-lock.json
package.json Rolex
Bgm.mp3
source.txt src.html
style.html
table.html
target.txt
title.html
Vaarasudu.mp4
week.js week.ts
week1.html
```

```
week10a.js
week10a.ts
week10b.js
week10b.ts
week10c.js
week10c.ts
week10d.js
week10d.ts
week1b.html
week1c.2.html
week1c.html
week1d.html
week1e.html
week1f.html
week2a.html
week2b.html
week2c.html
week2d.html
week3a.html
week3b.html
week3c.html
week3d.html
week3e.html
week4a.html
week4b.html
week4c.html
week4d.html
week5a.html
week5b.html
week5c.html
week6.js
week6a.js
week6b.js
week6c.js
week6d.js
week6e.js
week9a.js
week9a.ts
week9b.js
week9b.ts
week9c.js
week9c.ts
week9d.js
```

```
week9d.ts
week9e.js
week9e.ts
weekq.js

File Contents of source.txt: giridhar is good person

Current filenames:
colour.css
colour.html
CSE.html
download.jpg durga-
devi-png.png
exam1.html
exam2.html
exam3.html
exam4.html
first.html
FLS-Blog-Black-Logos_Hero.jpg
function.js
giri.jpg
giri.js
ice.jpg
images.jpg
lukcy.mp4
ma_platinum_200_iso_white.jpg
mo.jpeg
mystyle.css
node_modules
package-lock.json
package.json Rolex
Bgm.mp3
source.txt src.html
style.html
table.html
target.txt
title.html
Vaarasudu.mp4
week.js week.ts
week1.html
week10a.js
```

```
week10a.ts
week10b.js
week10b.ts
week10c.js
week10c.ts
week10d.js
week10d.ts
week1b.html
week1c.2.html
week1c.html
week1d.html
week1e.html
week1f.html
week2a.html
week2b.html
week2c.html
week2d.html
week3a.html
week3b.html
week3c.html
week3d.html
week3e.html
week4a.html
week4b.html
week4c.html
week4d.html
week5a.html
week5b.html
week5c.html
week6.js
week6a.js
week6b.js
week6c.js
week6d.js
week6e.js
week9a.js
week9a.ts
week9b.js
week9b.ts
week9c.js
week9c.ts
week9d.js
week9d.ts
```

```
week9e.js
week9e.ts
weekq.js

File Contents of copied_file: giridhar is good personPS
C:\python programs\mean stack>
```