Batch: D-2 **Roll No.:** 16010122151

Experiment 2

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

Title: Node js Implementation

AIM: Node JS Implementation

Problem Definition:

-Demonstrate the Concept of Nodejs With the help of Example.

*(Students have to perform the task assigned within group and demonstrate the same).
Resources used:

Expected OUTCOME of Experiment:

CO 1:.Build full stack applications in JavaScript using the MERN technologies.

Books/ Journals/ Websites referred:

1. Shelly Powers Learning Node O' Reilly 2 nd Edition, 2016.

Pre Lab/ Prior Concepts:

1) File operation

- CRUD operations

- Check Permissions of a File or Directory.

- Checking if a file or a directory exists.

- Determining the line count of a text file.

- Reading a file line by line.

- See the file content through browser.

2) Building your custom modules

-To demonstrate this use some mathematics function to create custom module.

3) Basic Routing:

1. Build First server application using http module

2. Basic routing: Demonstrate it using simple HTML/Json file

3. Demonstrate the callback in node.js

4) Blocking and Non Blocking

Implementation Details:

1) **File operation**

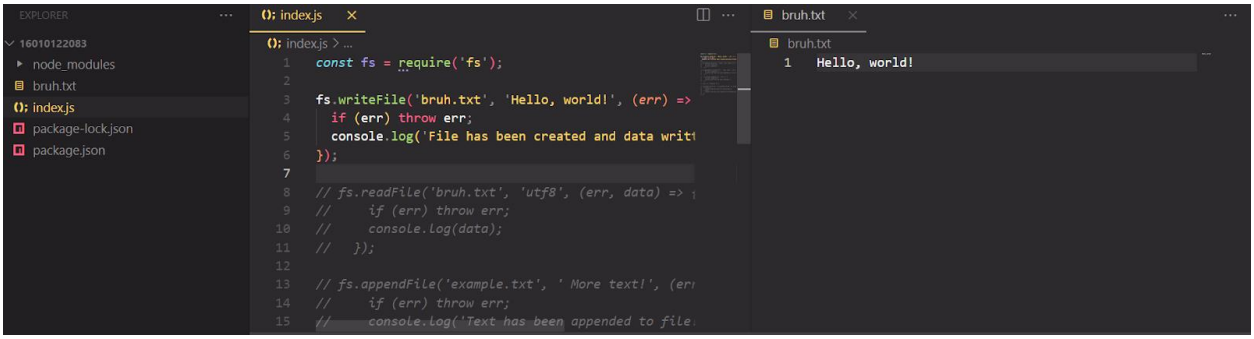
- **CRUD operations**
- **Check Permissions of a File or Directory.**
- **Checking if a file or a directory exists.**
- **Determining the line count of a text file.**

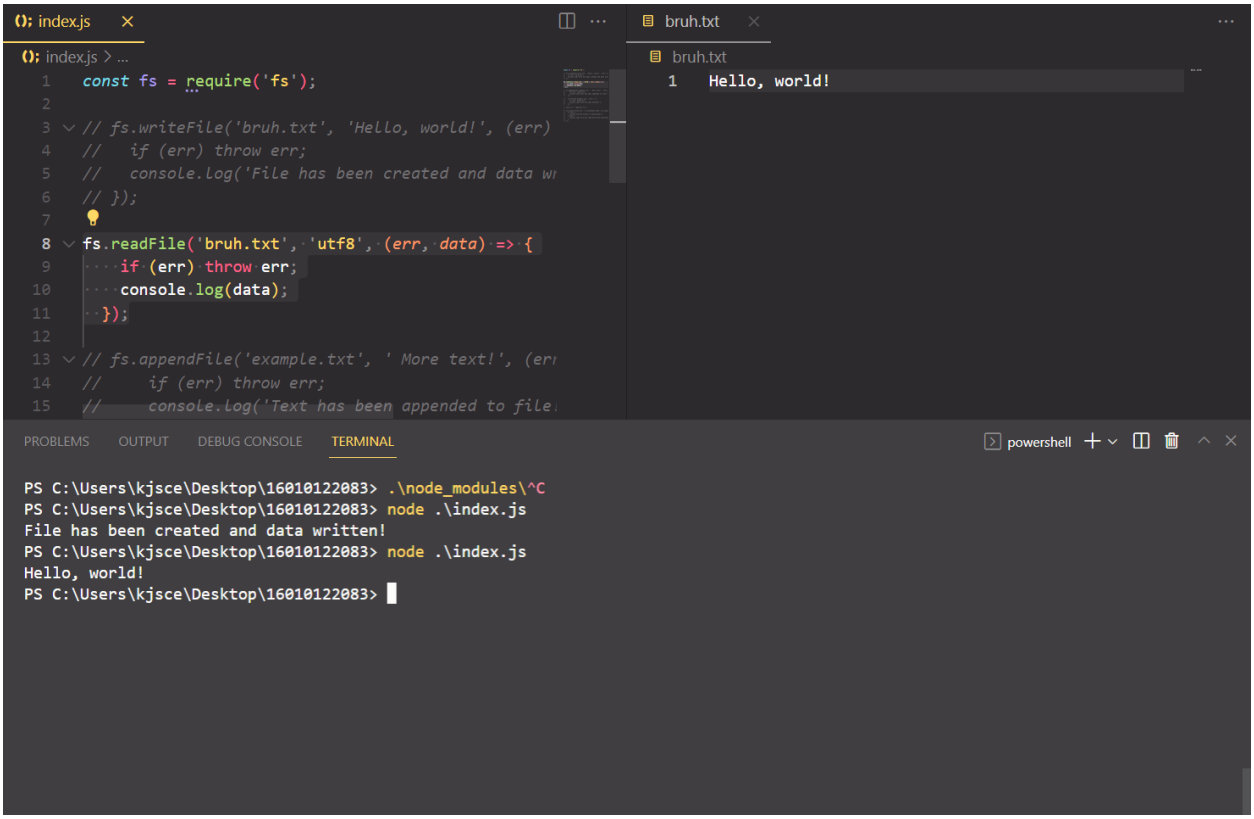
Code 1:

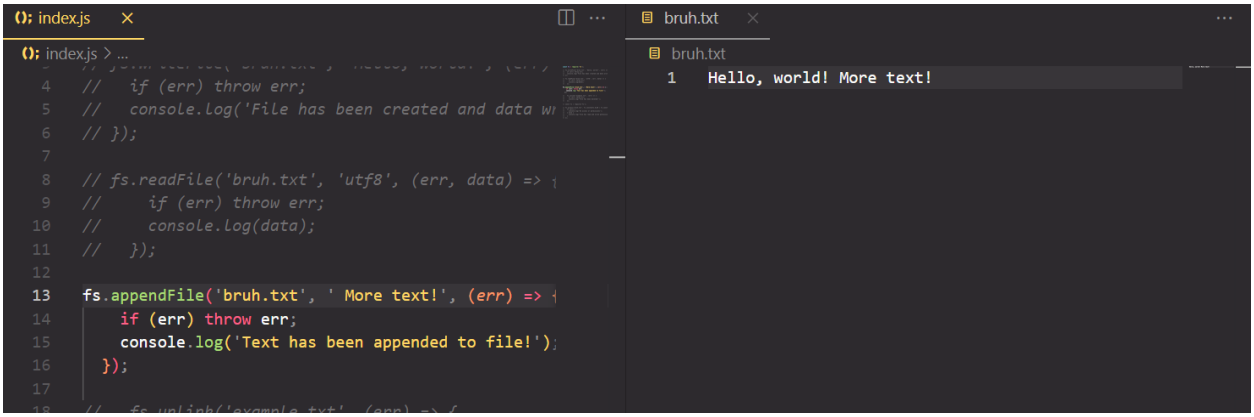
```
const fs = require('fs');

fs.writeFile('bruh.txt', 'Hello, world!', (err) => {
  if (err) throw err;
  console.log('File has been created and data written!');
});

fs.readFile('bruh.txt', 'utf8', (err, data) => {
    if (err) throw err;
    console.log(data);
  });

fs.appendFile('bruh.txt', ' More text!', (err) => {
    if (err) throw err;
    console.log('Text has been appended to file!');
  });

  fs.unlink('bruh.txt', (err) => {
    if (err) throw err;
    console.log('File has been deleted!');
  });

const fs = require('fs');

fs.access('mern.txt', fs.constants.R_OK | fs.constants.W_OK, (err) => {
  if (err) {
    console.log('No access or permissions');
  } else {
    console.log('File has read and write permissions');
  }
});
```

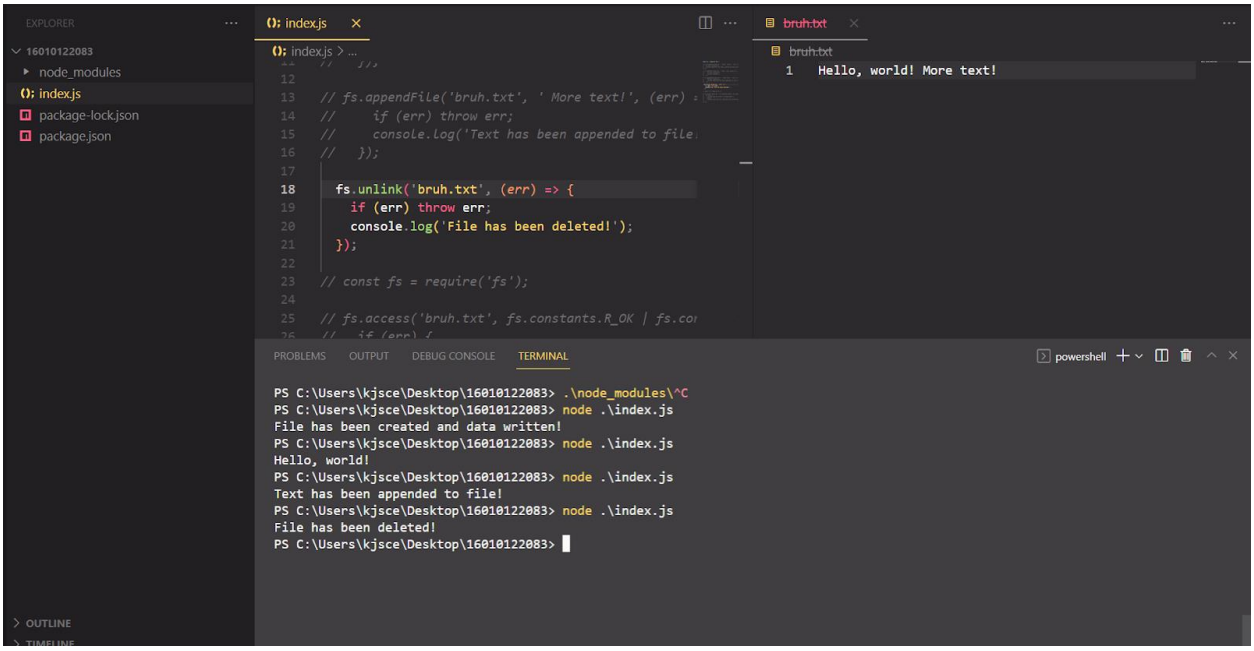
```
fs.access('example.txt', fs.constants.F_OK, (err) => {
  if (err) {
    console.log('File does not exist');
  } else {
    console.log('File exists');
  }
});
```

Output:





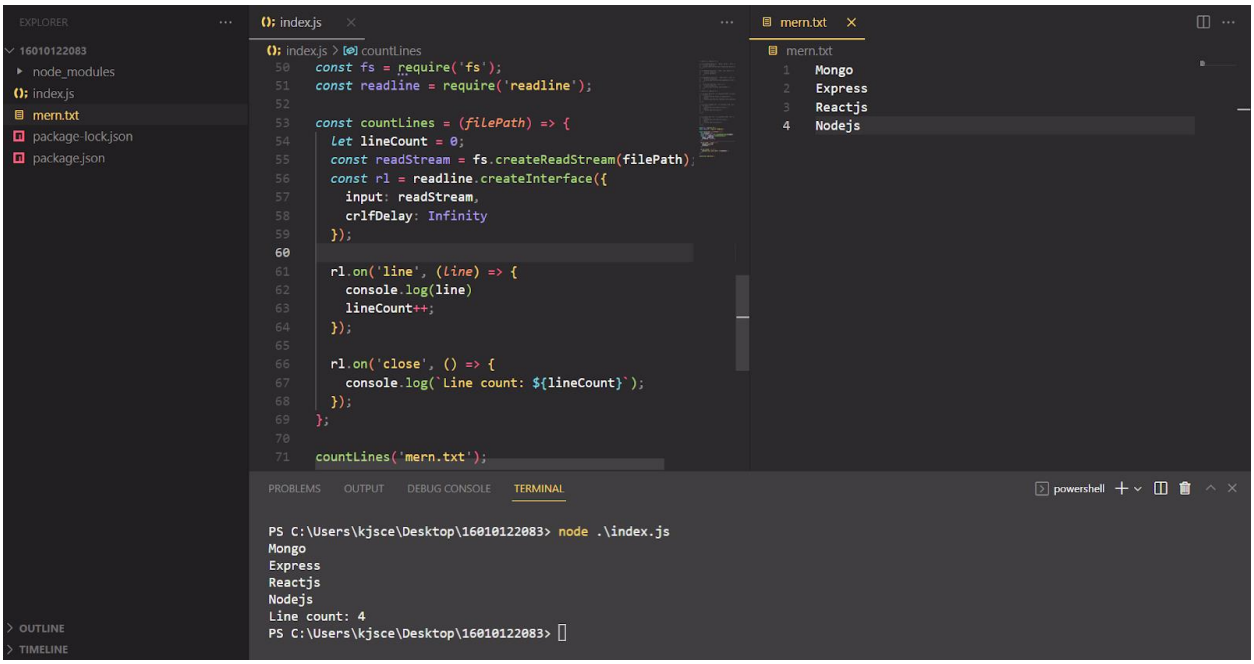

index.js
bruh.txt
// if (err) throw err;
// console.log('File has been created and data wr
// });
// fs.readFile('bruh.txt', 'utf8', (err, data) =>
// if (err) throw err;
// console.log(data);
// });
fs.appendFile('bruh.txt', ' More text!', (err) =>
if (err) throw err;
console.log('Text has been appended to file!');
});
Hello, world! More text!


EXPLORER
16010122083
node_modules
index.js
package-lock.json
package.json
OUTLINE
index.js
bruh.txt
// fs.appendFile('bruh.txt', ' More text!', (err)
// if (err) throw err;
// console.log('Text has been appended to file
// });
fs.unlink('bruh.txt', (err) => {
if (err) throw err;
console.log('File has been deleted!');
});
// const fs = require('fs');
// fs.access('bruh.txt', fs.constants.R_OK | fs.co
Hello, world! More text!
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
powershell
PS C:\Users\kjsce\Desktop\16010122083> .\node_modules\^C
PS C:\Users\kjsce\Desktop\16010122083> node .\index.js
File has been created and data written!
PS C:\Users\kjsce\Desktop\16010122083> node .\index.js
Hello, world!
PS C:\Users\kjsce\Desktop\16010122083> node .\index.js
Text has been appended to file!
PS C:\Users\kjsce\Desktop\16010122083> node .\index.js
File has been deleted!
PS C:\Users\kjsce\Desktop\16010122083>

Code :Reading a file line by line.

```
fs.access('mern.txt', fs.constants.F_OK, (err) => {
  if (err) {
    console.log('File does not exist');
  } else {
    console.log('File exists');
  }
});

const countLines = (filePath) => {
  let lineCount = 0;
  const readStream = fs.createReadStream(filePath);
  const rl = readline.createInterface({
    input: readStream,
    crlfDelay: Infinity
  });

  rl.on('line', (line) => {
    console.log(line)
    lineCount++;
  });

  rl.on('close', () => {
    console.log(`Line count: ${lineCount}`);
  });
};

countLines('mern.txt');
```

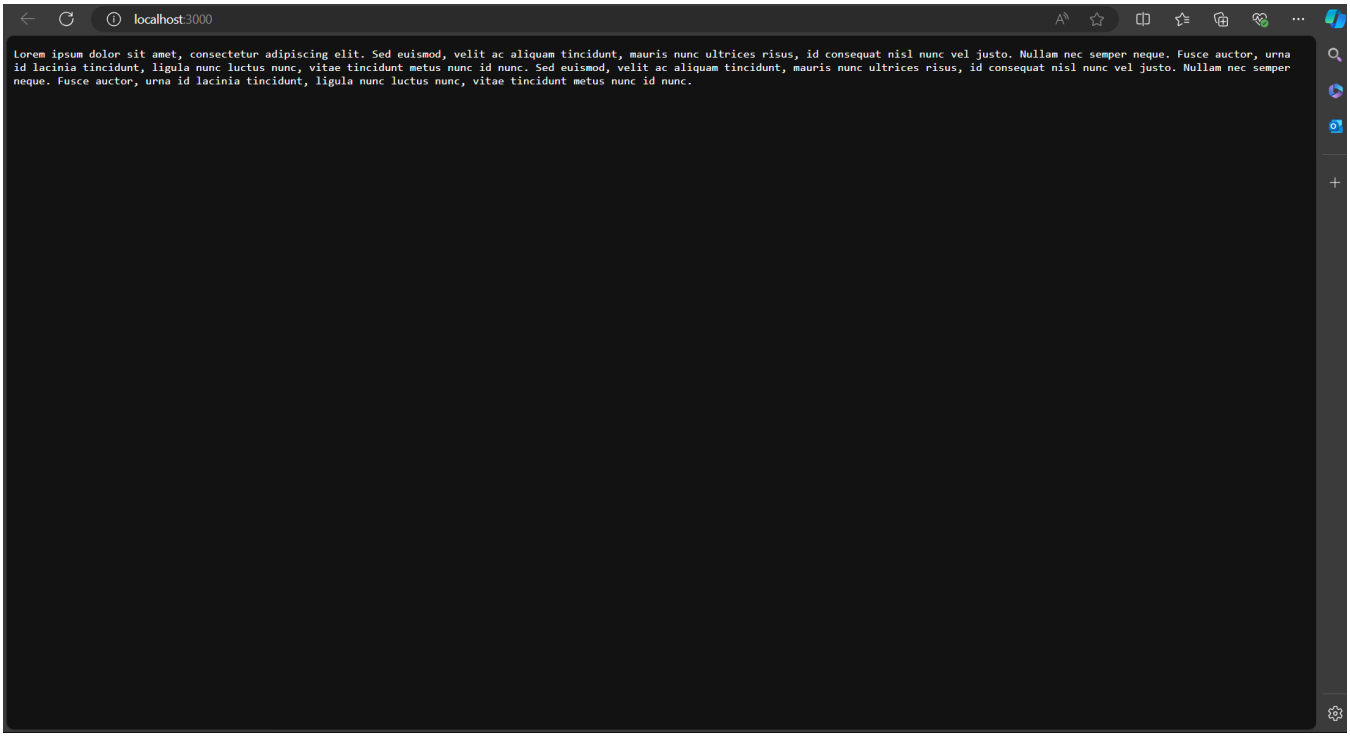
Output 2:

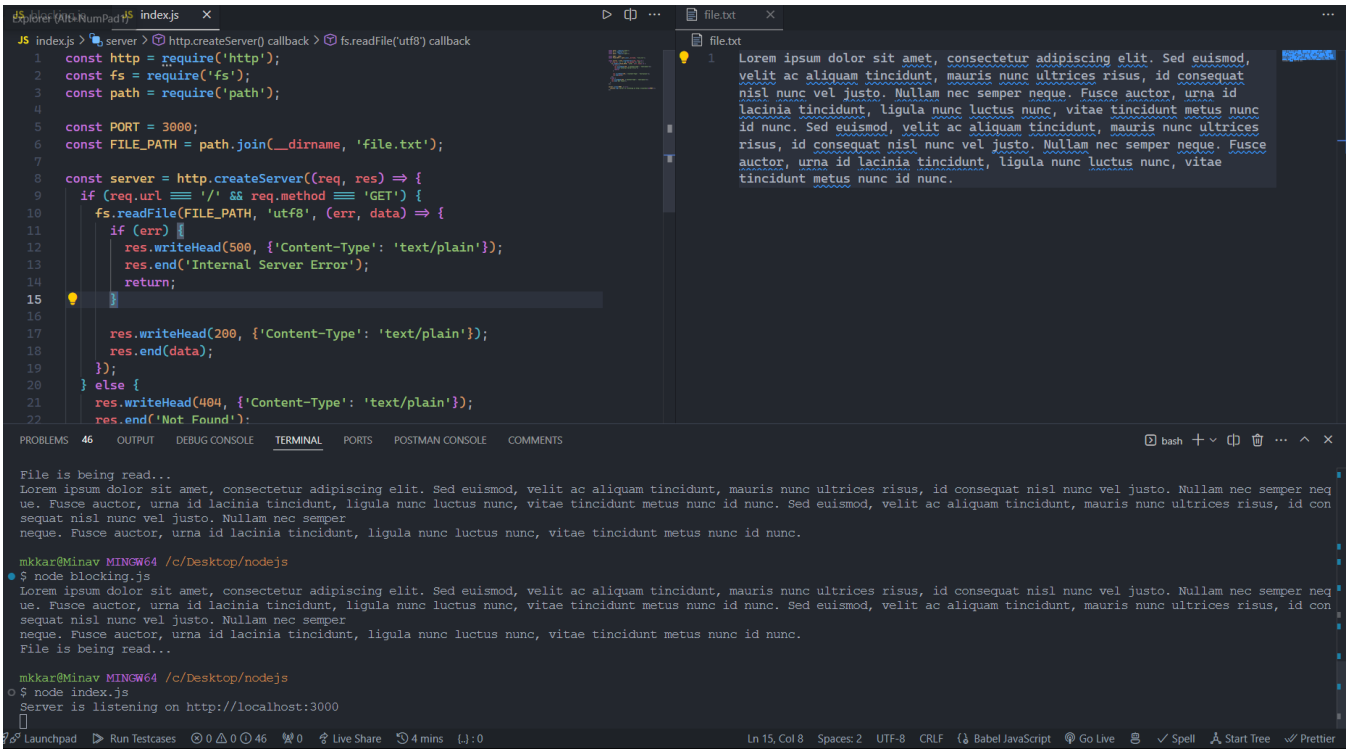

EXPLORER
16010122083
node_modules
index.js
mern.txt
package-lock.json
package.json
index.js
const fs = require('fs');
const readline = require('readline');
const countLines = (filePath) => {
let lineCount = 0;
const readStream = fs.createReadStream(filePath);
const rl = readline.createInterface({
input: readStream,
crlfDelay: Infinity
});
rl.on('line', (line) => {
console.log(line)
lineCount++;
});
rl.on('close', () => {
console.log(`Line count: ${lineCount}`);
});
};
countLines('mern.txt');
mern.txt
Mongo
Express
Reactjs
Nodejs
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
powershell
PS C:\Users\kjsce\Desktop\16010122083> node .\index.js
Mongo
Express
Reactjs
Nodejs
Line count: 4
PS C:\Users\kjsce\Desktop\16010122083>
OUTLINE
TIMELINE

Code :See the file content through browser.

```
const http = require('http');
const fs = require('fs');
const path = require('path');

const PORT = 3000;
const FILE_PATH = path.join(__dirname, 'file.txt');
const server = http.createServer((req, res) => {
  if (req.url === '/' && req.method === 'GET') {
    fs.readFile(FILE_PATH, 'utf8', (err, data) => {
      if (err) {
        res.writeHead(500, {'Content-Type': 'text/plain'});
        res.end('Internal Server Error');
        return;
      }
      res.writeHead(200, {'Content-Type': 'text/plain'});
      res.end(data);
    });
  } else {
    res.writeHead(404, {'Content-Type': 'text/plain'});
    res.end('Not Found');
  }
});
server.listen(PORT, () => {
  console.log(`Server is listening on http://localhost:${PORT}`);
});
```

Output:


localhost:3000
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed euismod, velit ac aliquam tincidunt, mauris nunc ultrices risus, id consequat nisl nunc vel justo. Nullam nec semper neque. Fusce auctor, urna id lacinia tincidunt, ligula nunc luctus nunc, vitae tincidunt metus nunc id nunc. Sed euismod, velit ac aliquam tincidunt, mauris nunc ultrices risus, id consequat nisl nunc vel justo. Nullam nec semper neque. Fusce auctor, urna id lacinia tincidunt, ligula nunc luctus nunc, vitae tincidunt metus nunc id nunc.


index.js
const http = require('http');
const fs = require('fs');
const path = require('path');
const PORT = 3000;
const FILE_PATH = path.join(__dirname, 'file.txt');
const server = http.createServer((req, res) => {
if (req.url === '/' && req.method === 'GET') {
fs.readFile(FILE_PATH, 'utf8', (err, data) => {
if (err) {
res.writeHead(500, {'Content-Type': 'text/plain'});
res.end('Internal Server Error');
return;
}
res.writeHead(200, {'Content-Type': 'text/plain'});
res.end(data);
});
} else {
res.writeHead(404, {'Content-Type': 'text/plain'});
file.txt
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed euismod, velit ac aliquam tincidunt, mauris nunc ultrices risus, id consequat nisl nunc vel justo. Nullam nec semper neque. Fusce auctor, urna id lacinia tincidunt, ligula nunc luctus nunc, vitae tincidunt metus nunc id nunc. Sed euismod, velit ac aliquam tincidunt, mauris nunc ultrices risus, id consequat nisl nunc vel justo. Nullam nec semper neque. Fusce auctor, urna id lacinia tincidunt, ligula nunc luctus nunc, vitae tincidunt metus nunc id nunc.
PROBLEMS 46 OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE COMMENTS
File is being read...
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed euismod, velit ac aliquam tincidunt, mauris nunc ultrices risus, id consequat nisl nunc vel justo. Nullam nec semper neque. Fusce auctor, urna id lacinia tincidunt, ligula nunc luctus nunc, vitae tincidunt metus nunc id nunc. Sed euismod, velit ac aliquam tincidunt, mauris nunc ultrices risus, id consequat nisl nunc vel justo. Nullam nec semper
neque. Fusce auctor, urna id lacinia tincidunt, ligula nunc luctus nunc, vitae tincidunt metus nunc id nunc.
mkkar@Minav MINGW64 /c/Desktop/nodejs
$ node blocking.js
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed euismod, velit ac aliquam tincidunt, mauris nunc ultrices risus, id consequat nisl nunc vel justo. Nullam nec semper neque. Fusce auctor, urna id lacinia tincidunt, ligula nunc luctus nunc, vitae tincidunt metus nunc id nunc. Sed euismod, velit ac aliquam tincidunt, mauris nunc ultrices risus, id consequat nisl nunc vel justo. Nullam nec semper
neque. Fusce auctor, urna id lacinia tincidunt, ligula nunc luctus nunc, vitae tincidunt metus nunc id nunc.
File is being read...
mkkar@Minav MINGW64 /c/Desktop/nodejs
$ node index.js
Server is listening on http://localhost:3000

2) Building your custom modules

-To demonstrate this use some mathematics function to create custom module.

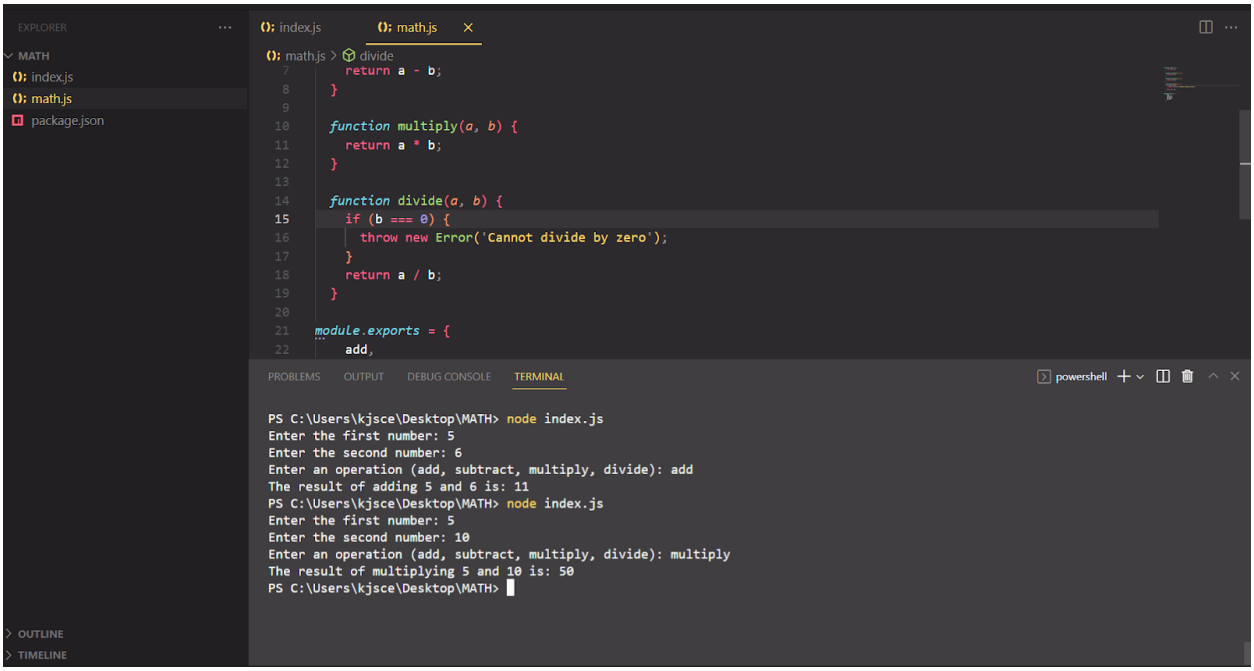
Code :

```
function add(a, b) {
    return a + b;
  }

  function subtract(a, b) {
    return a - b;
  }

  function multiply(a, b) {
    return a * b;
  }

  function divide(a, b) {
    if (b === 0) {
      throw new Error('Cannot divide by zero');
    }
    return a / b;
  }

module.exports = {
    add,
    subtract,
    multiply,
    divide
  };
```

```
const readline = require('readline');
const math = require('./math');

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

const promptUser = () => {
  rl.question('Enter the first number: ', (num1) => {
```

```
    rl.question('Enter the second number: ', (num2) => {
      rl.question('Enter an operation (add, subtract, multiply, divide): ',
(operation) => {
        const number1 = parseFloat(num1);
        const number2 = parseFloat(num2);

        try {
          let result;
          switch (operation.toLowerCase()) {
            case 'add':
              result = math.add(number1, number2);
              break;
            case 'subtract':
              result = math.subtract(number1, number2);
              break;
            case 'multiply':
              result = math.multiply(number1, number2);
              break;
            case 'divide':
              result = math.divide(number1, number2);
              break;
            default:
              console.log('Invalid operation. Please choose from add, subtract,
multiply, or divide.');
              rl.close();
              return;
          }
          console.log(`The result of ${operation}ing ${number1} and ${number2}
is: ${result}`);
        } catch (error) {
          console.error('Error:', error.message);
        }

        rl.close();
      });
    });
  });
};

promptUser();
```

Output :


EXPLORER
MATH
index.js
math.js
package.json
index.js
math.js
math.js > divide
return a - b;
}
function multiply(a, b) {
return a * b;
}
function divide(a, b) {
if (b === 0) {
throw new Error('Cannot divide by zero');
}
return a / b;
}
module.exports = {
add,
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
powershell
PS C:\Users\kjsce\Desktop\MATH> node index.js
Enter the first number: 5
Enter the second number: 6
Enter an operation (add, subtract, multiply, divide): add
The result of adding 5 and 6 is: 11
PS C:\Users\kjsce\Desktop\MATH> node index.js
Enter the first number: 5
Enter the second number: 10
Enter an operation (add, subtract, multiply, divide): multiply
The result of multiplying 5 and 10 is: 50
PS C:\Users\kjsce\Desktop\MATH>
OUTLINE
TIMELINE

3) Basic Routing:

1. Build First server application using http module

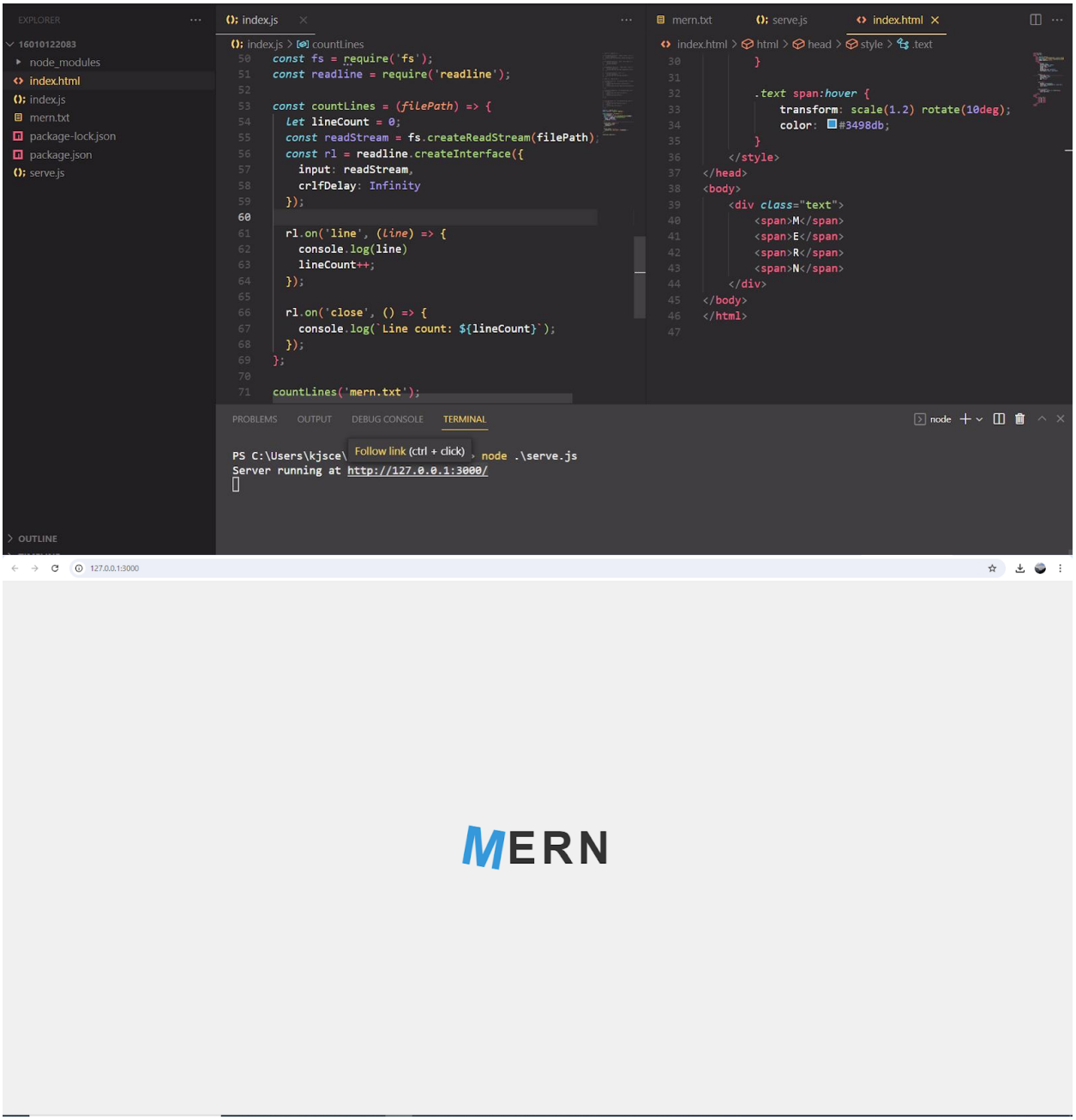
2. Basic routing: Demonstrate it using simple HTML/Json file

3. Demonstrate the callback in node.js

Code :

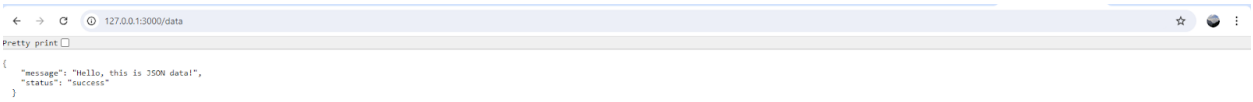
```
const http = require('http');
const fs = require('fs');
const path = require('path');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  if (req.url === '/') {
    serveFile('index.html', 'text/html', res);
  } else if (req.url === '/data') {
    serveFile('data.json', 'application/json', res);
  } else {
    res.statusCode = 404;
    res.setHeader('Content-Type', 'text/plain');
    res.end('404 Not Found');
  }
});

function serveFile(filePath, contentType, res) {
  const fullPath = path.join(__dirname, filePath);
  fs.readFile(fullPath, (err, data) => {
    if (err) {
      res.statusCode = 500;
      res.setHeader('Content-Type', 'text/plain');
      res.end('500 Internal Server Error');
      return;
    }
    res.statusCode = 200;
    res.setHeader('Content-Type', contentType);
    res.end(data);
  });
}

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Output :


index.js
const fs = require('fs');
const readline = require('readline');
const countLines = (filePath) => {
let lineCount = 0;
const readStream = fs.createReadStream(filePath);
const rl = readline.createInterface({
input: readStream,
crlfDelay: Infinity
});
rl.on('line', (line) => {
console.log(line)
lineCount++;
});
rl.on('close', () => {
console.log(`Line count: ${lineCount}`);
});
};
countLines('mern.txt');
index.html
.text span:hover {
transform: scale(1.2) rotate(10deg);
color: #3498db;
}
</style>
</head>
<body>

M
E
R
N

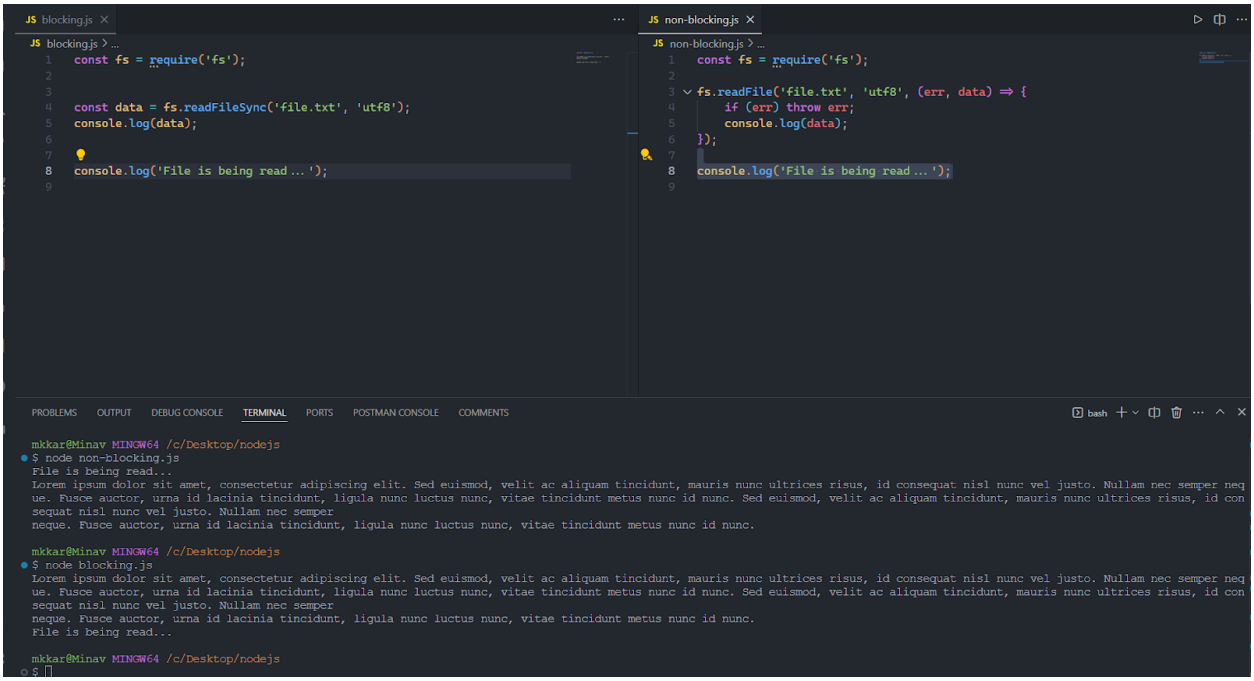
</body>
</html>
PS C:\Users\kjsce\ node .\serve.js
Server running at http://127.0.0.1:3000/
MERN


127.0.0.1:3000/data
Pretty print
{
"message": "Hello, this is JSON data!",
"status": "success"
}

4) Blocking and Non Blocking

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


const data = fs.readFileSync('file.txt', 'utf8');
console.log(data);


console.log('File is being read...');
```

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

fs.readFile('file.txt', 'utf8', (err, data) => {
    if (err) throw err;
    console.log(data);
});

console.log('File is being read...');
```



Conclusion:

We learned file handling and brower routing with nodejs and its implementation