<table>
<tr><td colspan="3">Name of Student: Samyak Lamsoge</td></tr>
<tr><td colspan="2">Roll Number: 29</td><td>LAB Assignment Number: 5</td></tr>
<tr><td colspan="3">Title of LAB Assignment: Create an application to demonstrate Node.js Functions-timer function(displays every 10 seconds)</td></tr>
<tr><td colspan="2">DOP: 09-10-2023</td><td>DOS: 10-10-2023</td></tr>
<tr><td>CO Mapped: CO1</td><td>PO Mapped: PO3, PO5, PSO1, PSO2</td><td>Signature:</td></tr>
</table>

PRACTICAL 5

Aim: Using File Handling demonstrate all basic file operations (Create, write, read, delete)

Theory:

fs module:

The Node.js file system module allows you to work with the file system on your computer.

fs methods

Method	Description
appendFile()	Appends data to a file
appendFileSync()	Same as appendFile(), but synchronous instead of asynchronous
close()	Closes a file
closeSync()	Same as close(), but synchronous instead of asynchronous
exists()	Deprecated. Checks if a file or folder exists
existsSync()	Same as exists(), but synchronous instead of asynchronous. This method is NOT deprecated
link()	Makes an addition name for a file. Both the old and the new name may be used

linksync()	Same as link(), but synchronous instead of asynchronous
open()	Opens a file
openSync()	Same as open(), but synchronous instead of asynchronous
read()	Reads the content of a file
readdir()	Reads the content of a directory
readdirSync()	Same as readdir(), but synchronous instead of asynchronous
readFile()	Reads the content of a file
readFileSync()	Same as readFile(), but synchronous instead of asynchronous
readlink()	Reads the value of a link
readlinkSync()	Same as readlink(), but synchronous instead of asynchronous
realpath()	Returns the absolute pathname
realpathSync()	Same as realpath(), but synchronous instead of asynchronous
rename()	Renames a file

renameSync()	Same as rename(), but synchronous instead of asynchronous
rmdir()	Removes a directory
rmdirSync()	Same as rmdir(), but synchronous instead of asynchronous
stat()	Returns the status of a file
statSync()	Same as stat(), but synchronous instead of asynchronous
symlink()	Makes a symbolic name for a file
symlinkSync()	Same as symlink(), but synchronous instead of asynchronous
truncate()	Truncates a file
truncateSync()	Same as truncate(), but synchronous instead of asynchronous
unlink()	Removes a link
unlinkSync()	Same as unlink(), but synchronous instead of asynchronous
write()	Writes buffer to a file
write()	Writes data to a file

writeFile()	Writes data to a file
writeFileSync()	Same as writeFile(), but synchronous instead of asynchronous
writeSync()	Same as write(); writes buffer to a file synchronous instead of asynchronous
writeSync()	Same as write(); writes data to a file synchronous instead of asynchronous

1. Aim: Using File Handling demonstrate all basic file operations (Create, write, read, delete)

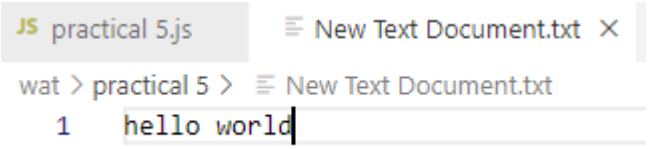
a) Aim: Read a file

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


// read a file

let buffer = fs.readFileSync('New Text Document.txt')


let data = buffer.toString()


console.log('reading file: New Text Document.txt \n', data)
```

Output

```
reading file: New Text Document.txt
 hello world
```

New Text Document.txt



b) Aim: Write to an existing file

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


// write to a file

console.log('\nappending "practical 5 file system" to New
Text Document.txt');


fs.appendFileSync('New Text Document.txt', "\npractical 5
file system") // appends the new data after the file's old
data




// writeFileSync() will replace the file's old data with the
new data


// read a file after writing to it

buffer = fs.readFileSync('New Text Document.txt')


data = buffer.toString()
```

```
console.log('\nreading file: New Text Document.txt after
writing to it \n', data)
```

Output

```
appending "practical 5 file system" to New Text Document.txt

reading file: New Text Document.txt after writing to it
 hello world
practical 5 file system
```

New Text Document.txt

```
JS practical 5.js    New Text Document.txt ×
wat > practical 5 > New Text Document.txt
1   hello world
2   practical 5 file system
```

c) Aim: Create a file

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


// create and new file and write to a it

console.log('\ncreating new file: practical 5.txt with contents "practical 5 file
system"');


fs.writeFileSync('practical 5.txt', "practical 5 file system")


// read a file after writing to it

buffer = fs.readFileSync('practical 5.txt')
```

```
data = buffer.toString()


console.log('\nreading file: practical 5.txt \n', data)
```

Output

```
creating new file: practical 5.txt with contents "practical 5 file sy
stem"

reading file: practical 5.txt
 practical 5 file system
```

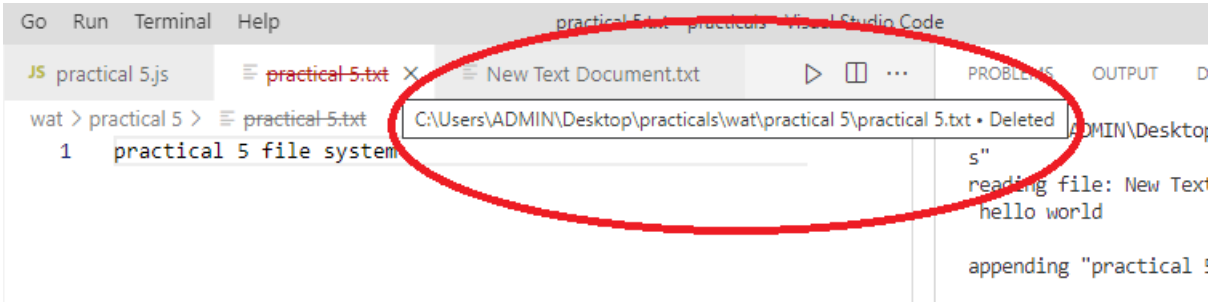
practical 5.txt

```
JS practical 5.js    practical 5.txt ×
wat > practical 5 > practical 5.txt
1   practical 5 file system
```

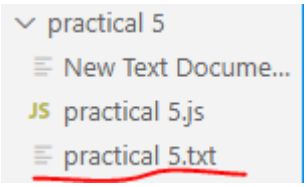
d) Aim: Delete a file

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


// delete the created file

fs.unlinkSync('practical 5.txt')
```

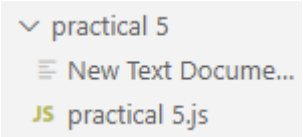
Output (file is seen as deleted)



Folder contents before deleting the file



Folder contents after deleting the file



Conclusion: We learnt about the fs module in Nodejs.