Name : Kunal Rajesh Kumbhare
Prn : 120A3024
Branch : IT
3rd Year (5th semester)

Experiment No: 10

Aim: - Write a program to implement file system in Node.js
Theory :-

Node.js is a JavaScript runtime built on Chrome’s V8 JavaScript engine. Node.js helps developers to write JavaScript code to run on the server-side, to generate dynamic content and deliver to the web clients. The two features that make Node.js stand-out are:
Event-driven

Non-blocking, I/O model

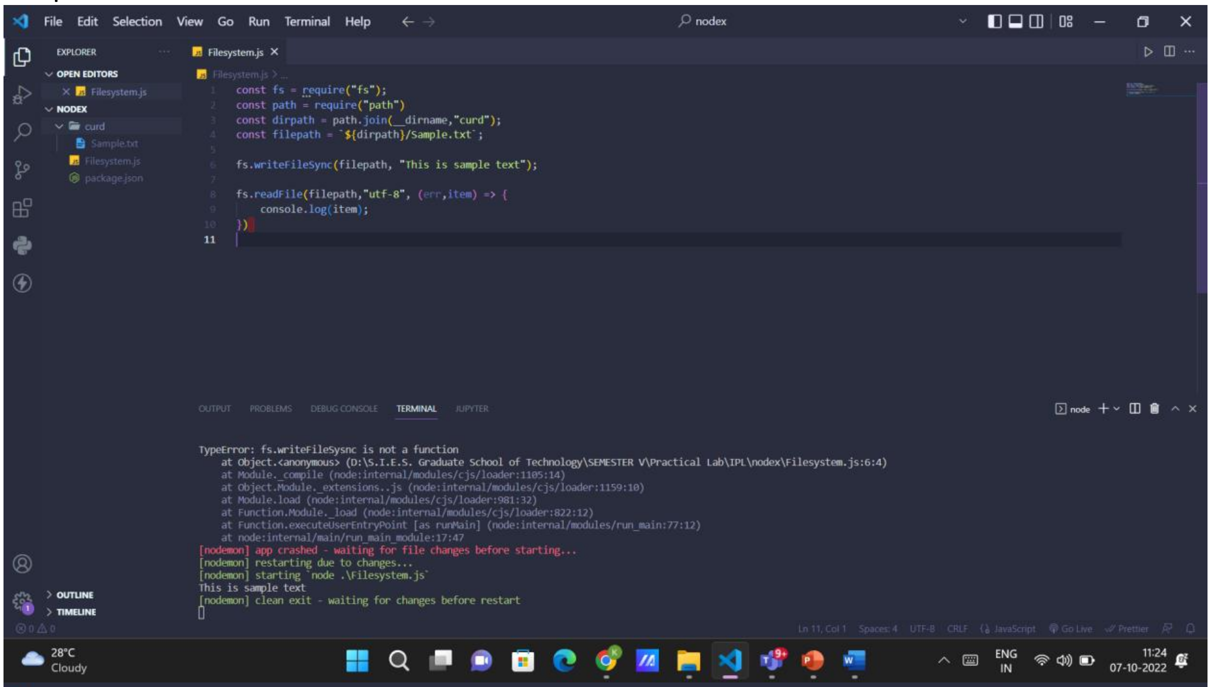
About Node.js file system: To handle file operations like creating, reading, deleting, etc., Node.js provides an inbuilt module called FS (File System). Node.js gives the functionality of file I/O by providing wrappers around the standard POSIX functions. All file system operations can have synchronous and asynchronous forms depending upon user requirements. To use this File System module, use the require() method:
var fs = require('fs');
Common use for File System module:

- Read Files
- Write Files
- Append Files
- Close Files
- Delete Files

Code:

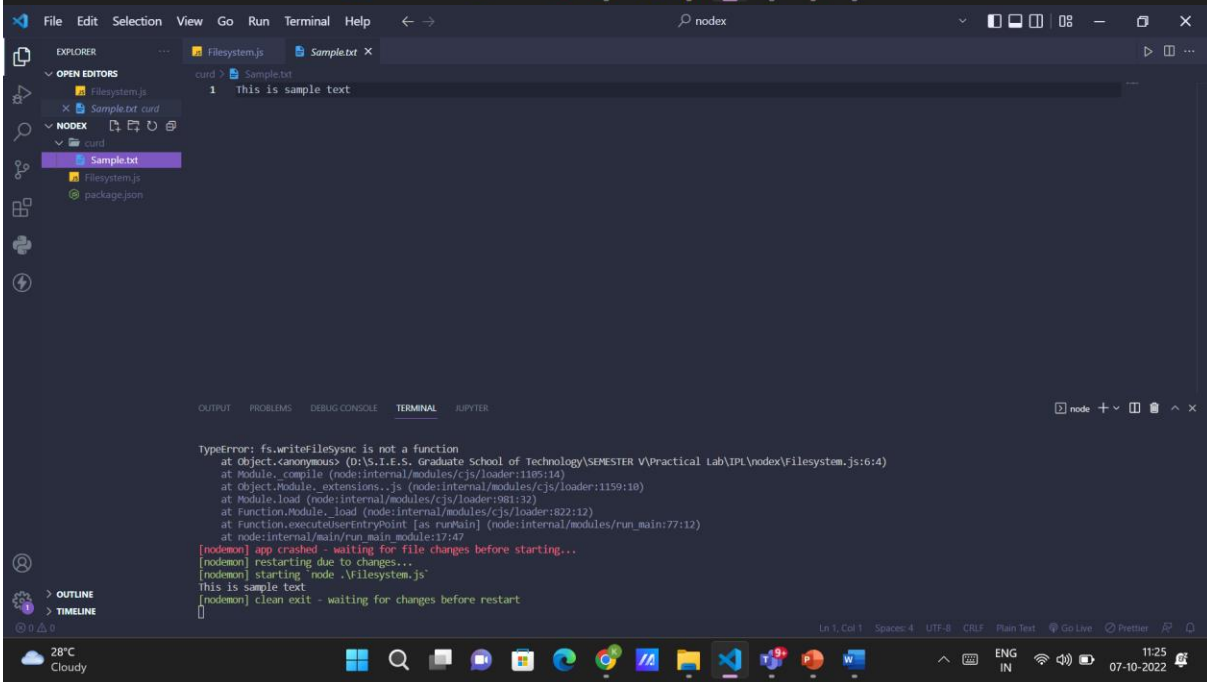
```
const fs = require("fs");
const path = require("path")
const dirpath = path.join(__dirname,"curd");
const filepath = `${dirpath}/Sample.txt`;

fs.writeFileSync(filepath, "This is sample text");

fs.readFile(filepath,"utf-8", (err,item) => {
    console.log(item);
})

fs.appendFile(filepath, " To read and append something on these file", (err)
=> {
    if(!err) console.log("sample.txt file is created");
});

fs.rename(filepath, `${dirpath}/example.txt`, (err) => {
   if(!err) console.log ("Filename is updated");
});

fs.unlinkSync(`${dirpath}/example.txt`, (del) => {
    if(del) console.log("file is unlinked")
});
```

Output :



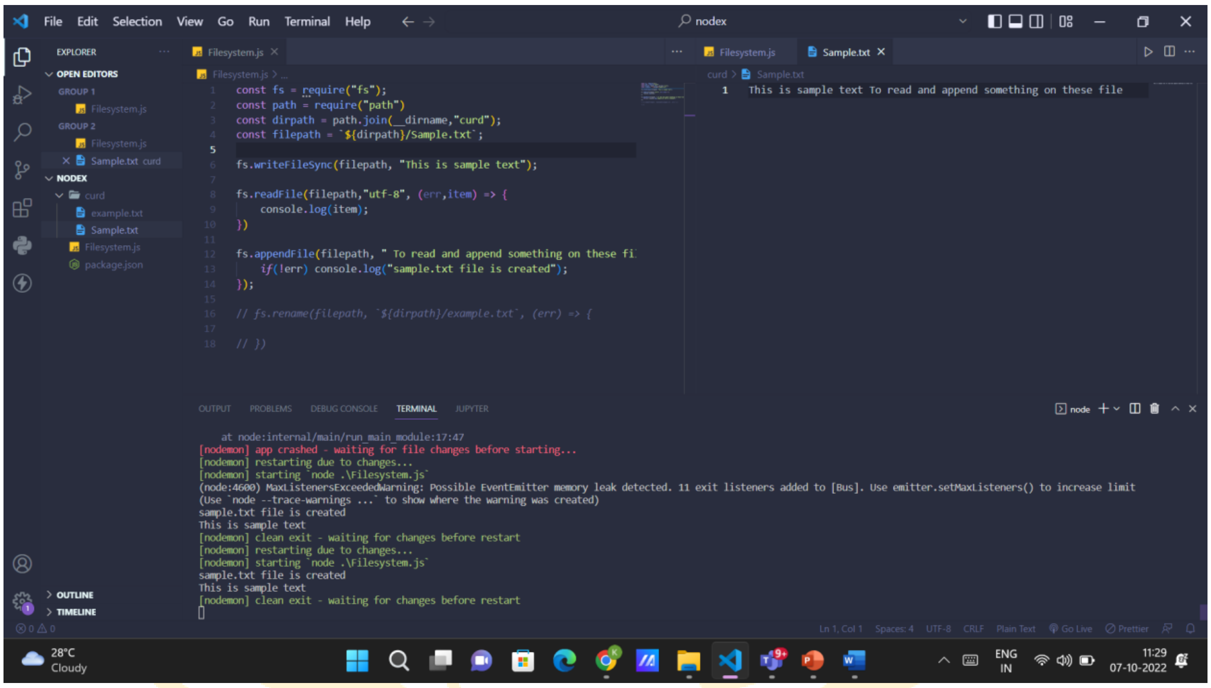
```javascript
const fs = require("fs");
const path = require("path")
const dirpath = path.join(__dirname,"curd");
const filepath = `${dirpath}/Sample.txt`;

fs.writeFileSync(filepath, "This is sample text");

fs.readFile(filepath,"utf-8", (err,item) => {
    console.log(item);
})
```

```
TypeError: fs.writeFileSysnc is not a function
    at Object.<anonymous> (D:\S.I.E.S. Graduate School of Technology\SEMESTER V\Practical Lab\IPL\nodex\Filesystem.js:6:4)
    at Module._compile (node:internal/modules/cjs/loader:1105:14)
    at Object.Module._extensions..js (node:internal/modules/cjs/loader:1159:10)
    at Module.load (node:internal/modules/cjs/loader:981:32)
    at Function.Module._load (node:internal/modules/cjs/loader:822:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:77:12)
    at node:internal/main/run_main_module:17:47
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting `node .\Filesystem.js`
This is sample text
[nodemon] clean exit - waiting for changes before restart
```



```
This is sample text
```

```
TypeError: fs.writeFileSysnc is not a function
    at Object.<anonymous> (D:\S.I.E.S. Graduate School of Technology\SEMESTER V\Practical Lab\IPL\nodex\Filesystem.js:6:4)
    at Module._compile (node:internal/modules/cjs/loader:1105:14)
    at Object.Module._extensions..js (node:internal/modules/cjs/loader:1159:10)
    at Module.load (node:internal/modules/cjs/loader:981:32)
    at Function.Module._load (node:internal/modules/cjs/loader:822:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:77:12)
    at node:internal/main/run_main_module:17:47
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting `node .\Filesystem.js`
This is sample text
[nodemon] clean exit - waiting for changes before restart
```

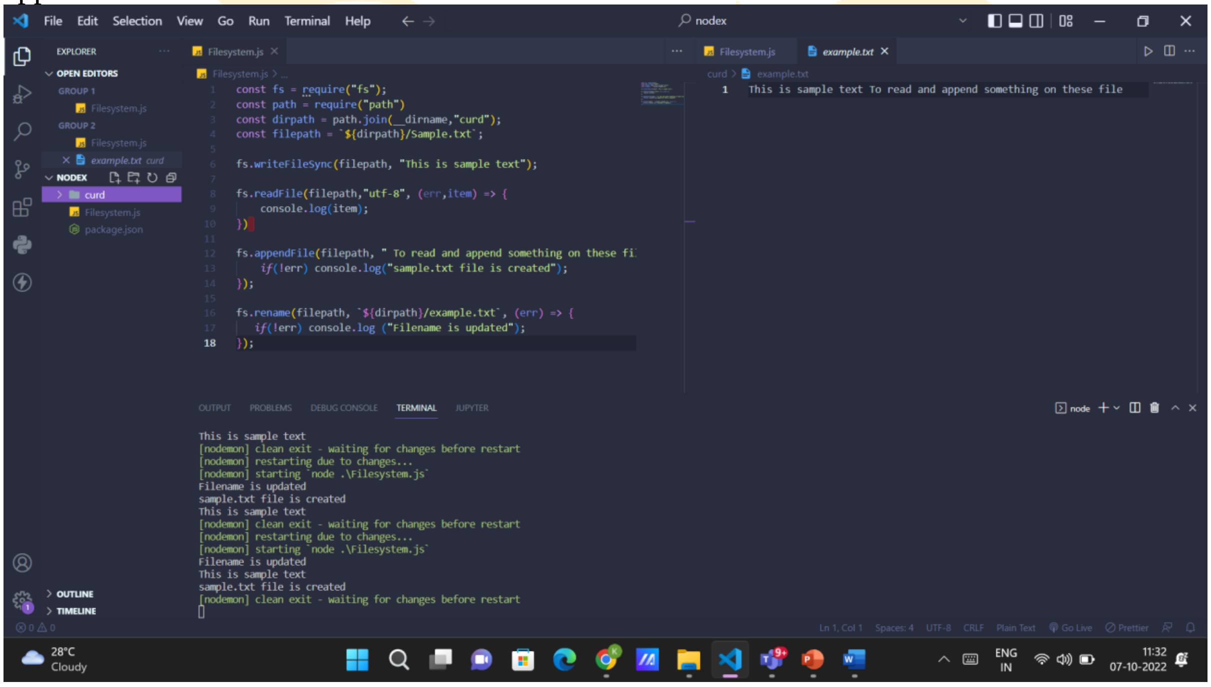


```
const fs = require("fs");
const path = require("path")
const dirpath = path.join(__dirname,"curd");
const filepath = `${dirpath}/Sample.txt`;

fs.writeFileSync(filepath, "This is sample text");

fs.readFile(filepath,"utf-8", (err,item) => {
    console.log(item);
})

fs.appendFile(filepath, " To read and append something on these fi
    if(!err) console.log("sample.txt file is created");
});

// fs.rename(filepath, `${dirpath}/example.txt`, (err) => {

// })
```

```
This is sample text To read and append something on these file
```

```
    at node:internal/main/run_main_module:17:47
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting `node .\Filesystem.js`
(node:4600) MaxListenersExceededWarning: Possible EventEmitter memory leak detected. 11 exit listeners added to [Bus]. Use emitter.setMaxListeners() to increase limit
(Use `node --trace-warnings ...` to show where the warning was created)
sample.txt file is created
This is sample text
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting `node .\Filesystem.js`
sample.txt file is created
This is sample text
[nodemon] clean exit - waiting for changes before restart
```

appendFile

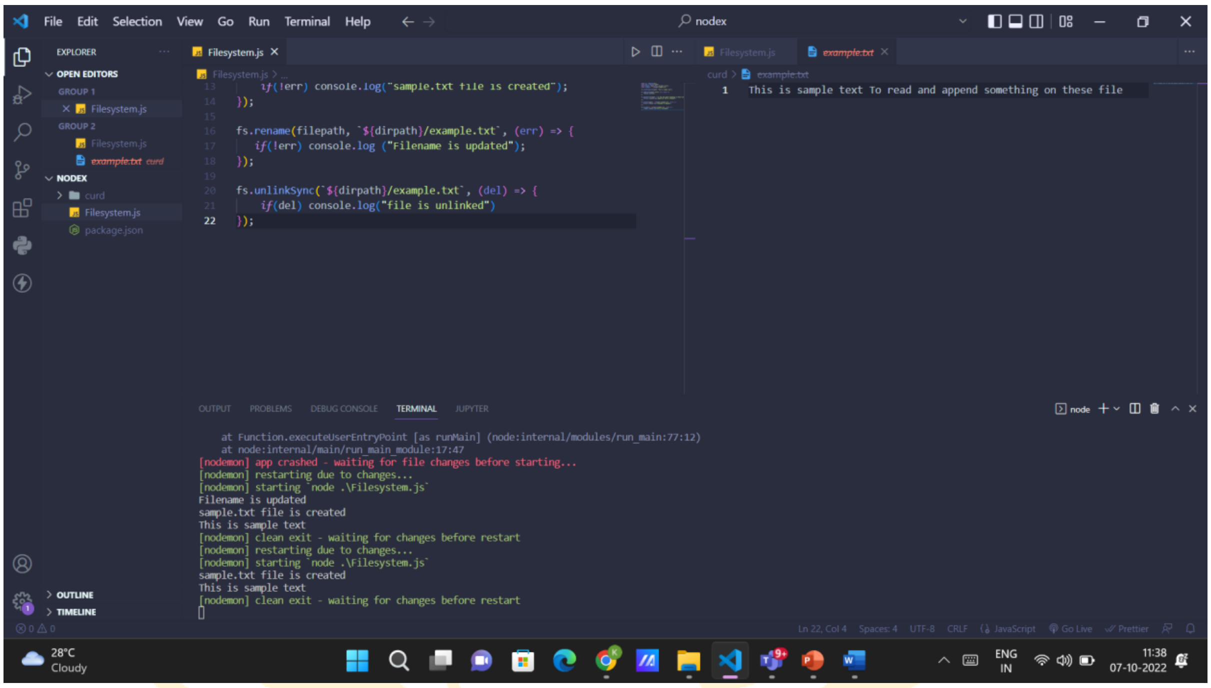


```
const fs = require("fs");
const path = require("path")
const dirpath = path.join(__dirname,"curd");
const filepath = `${dirpath}/Sample.txt`;

fs.writeFileSync(filepath, "This is sample text");

fs.readFile(filepath,"utf-8", (err,item) => {
    console.log(item);
})

fs.appendFile(filepath, " To read and append something on these fi
    if(!err) console.log("sample.txt file is created");
});

fs.rename(filepath, `${dirpath}/example.txt`, (err) => {
   if(!err) console.log ("Filename is updated");
});
```

```
This is sample text To read and append something on these file
```

```
This is sample text
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting `node .\Filesystem.js`
Filename is updated
sample.txt file is created
This is sample text
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting `node .\Filesystem.js`
Filename is updated
This is sample text
sample.txt file is created
[nodemon] clean exit - waiting for changes before restart
```

UnlinkSync



```
    if(!err) console.log("sample.txt file is created");
});

fs.rename(filepath, `${dirpath}/example.txt`, (err) => {
  if(!err) console.log ("Filename is updated");
});

fs.unlinkSync(`${dirpath}/example.txt`, (del) => {
    if(del) console.log("file is unlinked")
});
```

```
This is sample text To read and append something on these file
```

```
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:77:12)
    at node:internal/main/run_main_module:17:47
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting `node .\Filesystem.js`
Filename is updated
sample.txt file is created
This is sample text
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting `node .\Filesystem.js`
sample.txt file is created
This is sample text
[nodemon] clean exit - waiting for changes before restart
```

Conclusion:

Thus, we have successfully able to implement file system in Node.js