**Aim: Write a program to underhand the working of the file system**

**Objective: To easily built fast and scalable application**

**Scope:** open-source runtime environment used for server-side development and Increased use in microservices architecture.

**Theory:-**

Node.js provides a built-in module called **fs** (File System) that allows you to interact with the file system on your computer. With the **fs** module, you can perform various file-related operations, such as reading, writing, updating, and deleting files.

**Basic File Operations:**

1. **Reading Files:**

* You can read the contents of a file using the **fs.readFile()** method..

var fs = require('fs');  
  
fs.readFile('demofile.txt', 'utf8', function(err, data) {  
  if (err) throw err;  
  console.log(data);  
});

1. **Writing Files:**

* You can write data to a file using the **fs.writeFile()** method.

var fs = require('fs');

fs.writeFile('newfile.txt', 'Hello, Node.js!', (err) => {

if (err) {

console.error(err);

return;

}

console.log('File written successfully.');

});

1. **Appending to Files:**

* You can append data to an existing file using the **fs.appendFile()** method.

var fs = require('fs');

fs.appendFile('myfile.txt', ' Appended content.', (err) => {

if (err) {

console.error(err);

return;

}

console.log('Data appended to file.');

});

1. **Renaming and Moving Files:**

* You can rename and move files using the **fs.rename()** method.

var fs = require('fs');

fs.rename('oldfile.txt', 'newname.txt', (err) => {

if (err) {

console.error(err);

return;

}

console.log('File renamed successfully.');

});

1. **Deleting Files:**

* You can delete files using the **fs.unlink()** method

var fs = require('fs');

fs.unlink('fileToDelete.txt', (err) => {

if (err) {

console.error(err);

return;

}

console.log('File deleted successfully.');

});.

**Conclusion:**

Node.js along with different features is an open source environment studied successfully.