Implement all CRUD operations on a File System using Node JS

Step1:

Open your USN folder in VS Code → Create a new folder called (5th_prog) → Open terminal in VS Code and write the following commands

```
npm init -y

npm install fs
```

Step2: Create a new file called **script.js** in 5th_prog folder

```
const fs = require('fs'); //import fs package

const path = './data'; // directory/folder where files will be stored while creating the file



// checking whether the directory/folder exist if yes ignore else creating the directory/folder

if (!fs.existsSync(path)) {

  fs.mkdirSync(path);

}



// Create function

const createFile = (filename, content) => {

    // If the file does exist, its content will be replaced with the new content. If file exist content will be
replaced with new content.

//err--> This is a callback function that gets called when the write operation completes. It takes one
parameter, err, which will contain an error object if an error occurred, or null if the operation was successful

  fs.writeFile(`${path}/${filename}`, content, (err) => {

        // If there was an error during the file write operation, this block will execute, and it will log an error
message to the console along with the error object.

    if (err) console.error('Error creating file:', err);

    else console.log('File created:', filename);

  });

};



// Read function
```

```
const readFile = (filename) => {
    // utf-8 This specifies the encoding to use when reading the file. takes 2 parameter: 1--> err(error object) and 2--> data(content)
  fs.readFile(`${path}/${filename}`, 'utf8', (err, data) => {
    if (err) console.error('Error reading file:', err);
    else console.log('File content:', data);
  });
};


// Update function
const updateFile = (filename, content) => {
  fs.writeFile(`${path}/${filename}`, content, (err) => {
    if (err) console.error('Error updating file:', err);
    else console.log('File updated:', filename);
  });
};


// Delete function
const deleteFile = (filename) => {
    // fs.unlink is used to delete a file.
  fs.unlink(`${path}/${filename}`, (err) => {
    if (err) console.error('Error deleting file:', err);
    else console.log('File deleted:', filename);
  });
};


// filename given in variable which will be called in function call
const filename = 'sample.txt';


// file content written
// const content = 'This is a sample content';    // call this variable in function call


// Creating the file

createFile(filename, "Sample file created using node js");


// Reading the  file

readFile(filename);


// Updating the file and updating its content {you can create a new varibale and call that variable in fucntion call just like const content present above}

updateFile(filename, 'Sample file is updated using node js');


// Deleting the file

deleteFile(filename)
```

Step3: To run the script open terminal and write the below command

```
node script.js
```