

Name: Tushar Panchal

En.No: 21162101014

Sub: MICROSEVICES

Branch: CBA

Batch:51

----PRACTICAL 06-----

Question (TASK):

To Implement the file system and its operation with NodeJS: "A-1" grocery shop owner wants to manage shop items using the asynchronous coding technique of node and want to perform the following task:

Practical 6.1: Reading data from CSV

Practical 6.2: Adding data to CSV

Practical 6.3: Deleting data from CSV

Practical 6.4: Renaming csv

Practical 6.5: Create an application to manage the students' grade sheet using a

CSV file. Columns include Student name, Quiz_Marks, Mid-

term_Marks,Assignment_Marks, final_exam_marksTotal_marks

Github Link:

https://github.com/Tushar007079/MICROSERVICES_PRACTICALS/tree/0e351382b192ebaff410bf4d4f193f4030603a64/6

Steps to perform:

- Install Dependencies.
- Open a terminal or command prompt, navigate to the project directory, and install the required dependencies. In this case, we need ,'fs(file-system)','csv-parser', to run the server.
- >> Run this following command to initialize the package.json file:
- >>> Run this following command to install the fs Module :
- >>> Run this following command to install the csv-parser Module : npm install csv-parser

6.1 <u>Reading data from CSV</u>:

✓ Code:

```
const fs = require('fs');
const csv = require('csv-parser');
function readCSV(filePath) {
  return new Promise((resolve, reject) => {
    const results = [];
    fs.createReadStream(filePath)
      .pipe(csv())
      .on('data', (data) => results.push(data))
      .on('end', () => resolve(results))
      .on('error', (error) => reject(error));
  });
readCSV('organizations-100.csv')
  .then((data) => {
    console.log(data);
  })
  .catch((error) => {
    console.error(error);
  });
```

✓ Output :

```
Index: '100',
    'Organization Id': 'e9eB5A60Cef8354',
    Name: 'Watkins-Kaiser',
    Website: 'http://www.herring.com/',
    Country: 'Togo',
    Description: 'Synergistic background access',
    Founded: '2009',
    Industry: 'Financial Services',
    'Number of employees': '2785'
}

>>pwsh

>>pwsh

>>Description: 'Synergistic background access',
    Founded: '2009',
    Industry: 'Financial Services',
    'Number of employees': '2785'
```

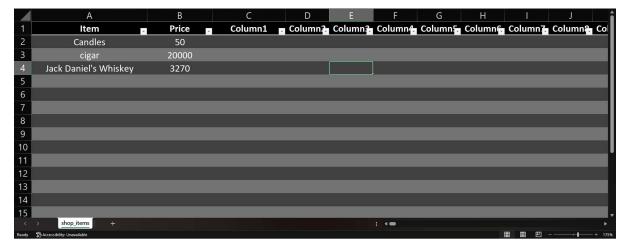
6.2 Adding data to CSV:

✓ Code:

```
const fs = require('fs');
const csv = require('csv-parser');
const readline = require('readline');
function addToCSV(filePath, newData) {
 return new Promise((resolve, reject) => {
    const results = [];
    fs.createReadStream(filePath)
      .pipe(csv())
      .on('data', (data) => results.push(data))
      .on('end', () => {
        results.push(newData);
        const ws = fs.createWriteStream(filePath);
        ws.write('Item, Price\n'); // Assuming CSV format: "Item, Price"
        results.forEach((row) => {
          ws.write(`${row.Item},${row.Price}\n`);
        });
        ws.end();
        resolve('Data added to CSV successfully.');
      })
      .on('error', (error) => reject(error));
  });
const rl = readline.createInterface({
  input: process.stdin,
 output: process.stdout
});
rl.question('Enter the Item: ', (item) => {
  rl.question('Enter the Price: ', (price) => {
    const newData = { Item: item, Price: parseFloat(price) };
    addToCSV('shop_items.csv', newData)
      .then((message) => {
        console.log(message);
        rl.close();
      })
      .catch((error) => {
        console.error(error);
        rl.close();
      });
  });
```

✓ Output:

>> shop_items.csv:



•• 6.3 <u>Deleting data from CSV</u>:

✓ Code:

```
const fs = require('fs');
const csv = require('csv-parser');
const readline = require('readline');
// Function to delete data from a CSV file
function deleteFromCSV(filePath, itemNameToDelete) {
 return new Promise((resolve, reject) => {
    const results = [];
    fs.createReadStream(filePath)
      .pipe(csv())
      .on('data', (data) => {
        if (data.Item !== itemNameToDelete) {
          results.push(data);
        }
      })
      .on('end', () => {
        const ws = fs.createWriteStream(filePath);
        ws.write('Item,Price\n'); // Assuming CSV format: "Item,Price"
        results.forEach((row) => {
          ws.write(`${row.Item},${row.Price}\n`);
        });
        ws.end();
        resolve('Data deleted from CSV successfully.');
      .on('error', (error) => reject(error));
  });
const rl = readline.createInterface({
 input: process.stdin,
 output: process.stdout
});
rl.question('Enter the CSV file name: ', (fileName) => {
  rl.question('Enter the Item to delete: ', (itemName) => {
    deleteFromCSV(fileName, itemName)
      .then((message) => {
        console.log(message);
        rl.close();
      })
      .catch((error) => {
        console.error(error);
        rl.close();
```

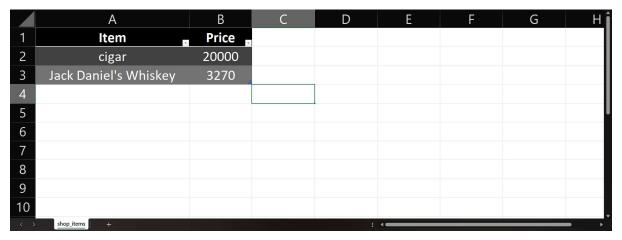
```
});
});
```

✓ Output:

pwsh P26 P13s 558ms

node "c:\Users\tusha\OneDrive\Docume
Enter the CSV file name: shop_items.csv
Enter the Item to delete: Candles
Data deleted from CSV successfully.

>> shop_items.csv:



6.4 Renaming the CSV:

✓ Code:

```
const fs = require('fs');
const readline = require('readline');
function renameCSV(oldFilePath, newFilePath) {
  return new Promise((resolve, reject) => {
    fs.rename(oldFilePath, newFilePath, (error) => {
      if (error) {
        reject(error);
      } else {
        resolve('CSV file renamed successfully.');
    });
  });
const rl = readline.createInterface({
  input: process.stdin,
 output: process.stdout
});
rl.question('Enter the old CSV file name: ', (oldFileName) => {
  rl.question('Enter the new CSV file name: ', (newFileName) => {
    renameCSV(oldFileName, newFileName)
      .then((message) => {
        console.log(message);
        rl.close();
      })
      .catch((error) => {
        console.error(error);
        rl.close();
      });
 });
```

✓ Output :

```
Pwsh P26 P22ms
node "c:\Users\tusha\OneDrive\Documents\SEM 5\V
Enter the old CSV file name: old_shop_items.csv
Enter the new CSV file name: new_shop_items.csv
CSV file renamed successfully.
```

6.5 <u>Create an application to manage the students' grade sheet using a CSV file.</u>

<u>Columns include Student name, Quiz_Marks, Mid-term_Marks, Assignment_Marks, final_exam_marksTotal_marks</u>:

✓ <u>Code :</u>

```
const fs = require('fs');
const readline = require('readline');
const rl = readline.createInterface({
    input: process.stdin,
    output: process.stdout
});
function mainMenu() {
    console.log('Main Menu:');
    console.log('1. Add Student');
    console.log('2. View Students');
    console.log('3. Delete Student');
    console.log('4. Exit');
    rl.question('Enter your choice: ', choice => {
        switch (choice) {
            case '1':
                addStudent();
               break;
            case '2':
                viewStudents();
            case '3':
                deleteStudent();
                break;
            case '4':
                rl.close();
                break;
            default:
                console.log('Invalid choice. Try again.');
                mainMenu();
                break;
   });
```

```
function addStudent() {
    // Get student details from the user
    rl.question('Enter student name: ', name => {
        rl.question('Enter Quiz Marks: ', quiz => {
            rl.question('Enter Mid-term Marks: ', midterm => {
                rl.question('Enter Assignment Marks: ', assignment => {
                    rl.question('Enter Final Exam Marks: ', finalExam => {
                        const totalMarks =
                            parseFloat(quiz) +
                            parseFloat(midterm) +
                            parseFloat(assignment) +
                            parseFloat(finalExam);
                        const studentData =
`${name},${quiz},${midterm},${assignment},${finaLExam},${totalMarks}\n`;
                        fs.appendFile('grade_sheet.csv', studentData, 'utf8',
err => {
                            if (err) {
                                console.error(err);
                            } else {
                                console.log('Student added successfully!');
                                mainMenu();
                   });
               });
      });
   });
function viewStudents() {
    fs.readFile('grade_sheet.csv', 'utf8', (err, data) => {
        if (err) {
            console.error(err);
        } else {
            console.log('Student Grade Sheet:');
            console.log(data);
        }
       mainMenu();
    });
function deleteStudent() {
    rl.question('Enter the name of the student to delete: ', nameToDelete => {
        fs.readFile('grade_sheet.csv', 'utf8', (err, data) => {
            if (err) {
                console.error(err);
               mainMenu();
```

```
} else {
                // Split the CSV data into an array of lines
                const lines = data.split('\n');
                // Find the index of the student to delete
                const indexToDelete = lines.findIndex(line => {
                    const studentData = line.split(',');
                    return studentData[0] === nameToDelete;
                });
                if (indexToDelete !== -1) {
                    // Remove the student's data from the array
                    lines.splice(indexToDelete, 1);
                    // Join the remaining lines back into CSV format
                    const updatedData = lines.join('\n');
                    // Write the updated data back to the CSV file
                    fs.writeFile('grade_sheet.csv', updatedData, 'utf8', err
=> {
                        if (err) {
                            console.error(err);
                        } else {
                            console.log(`Student "${nameToDeLete}" deleted
successfully.`);
                        mainMenu();
                    });
                } else {
                    console.log(`Student "${nameToDeLete}" not found.`);
                    mainMenu();
                }
      });
   });
mainMenu();
```

✓ Output:

Adding Student:

```
node "c:\Users\tusha\OneDri\
Main Menu:
1. Add Student
2. View Students
3. Delete Student
4. Exit
Enter your choice: 1
Enter student name: James Bond
Enter Quiz Marks: 20
Enter Mid-term Marks: 30
Enter Assignment Marks: 40
Enter Final Exam Marks: 50
Student added successfully!
```

View Student Records :

```
Main Menu:
1. Add Student
2. View Students
3. Delete Student
4. Exit
Enter your choice: 2
Student Grade Sheet:
James Bond, 20, 30, 40, 50, 140
```

Deleting Student Records:

```
Main Menu:
1. Add Student
2. View Students
3. Delete Student
4. Exit
Enter your choice: 3
Enter the name of the student to delete: James Bond
Student "James Bond" deleted successfully.
```

To close the Program choose 4. on Main menu's Choice.