

Name: Patel Shivam S.
Enroll no: 21162101019
Sem: 5 Batch: 51 Branch: CBA
Sub: Microservices

Practical 6

AIM: 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/Shivam3783/microservice_practicals/tree/main/prac6

Practical 6.1: Reading data from CSV

OUTPUT:

The screenshot shows a Mac desktop environment with several windows open:

- Terminal Window:** The title bar says "prac6". It contains the following code:

```
const fs = require('fs');
const csv = require('csv-parser');

const results = [];

fs.createReadStream('shop_items.csv')
  .pipe(csv())
  .on('data', (data) => {
    results.push(data);
  })
  .on('end', () => {
    console.log(results);
  });

node read.js
```
- Numbers Application:** A spreadsheet titled "shop_items" is open. The "Sheet 1" tab is selected. The data is as follows:

name	quantity	price
abc	1	5000
Apples	50	4000
new	3	10

- Code Editor:** An "EXPLORER" view is shown, listing files like "read.js", "add.js", "delete_column.js", etc. The "read.js" file is open in the editor, displaying the same code as the terminal.

Practical 6.2: Adding data to CSV

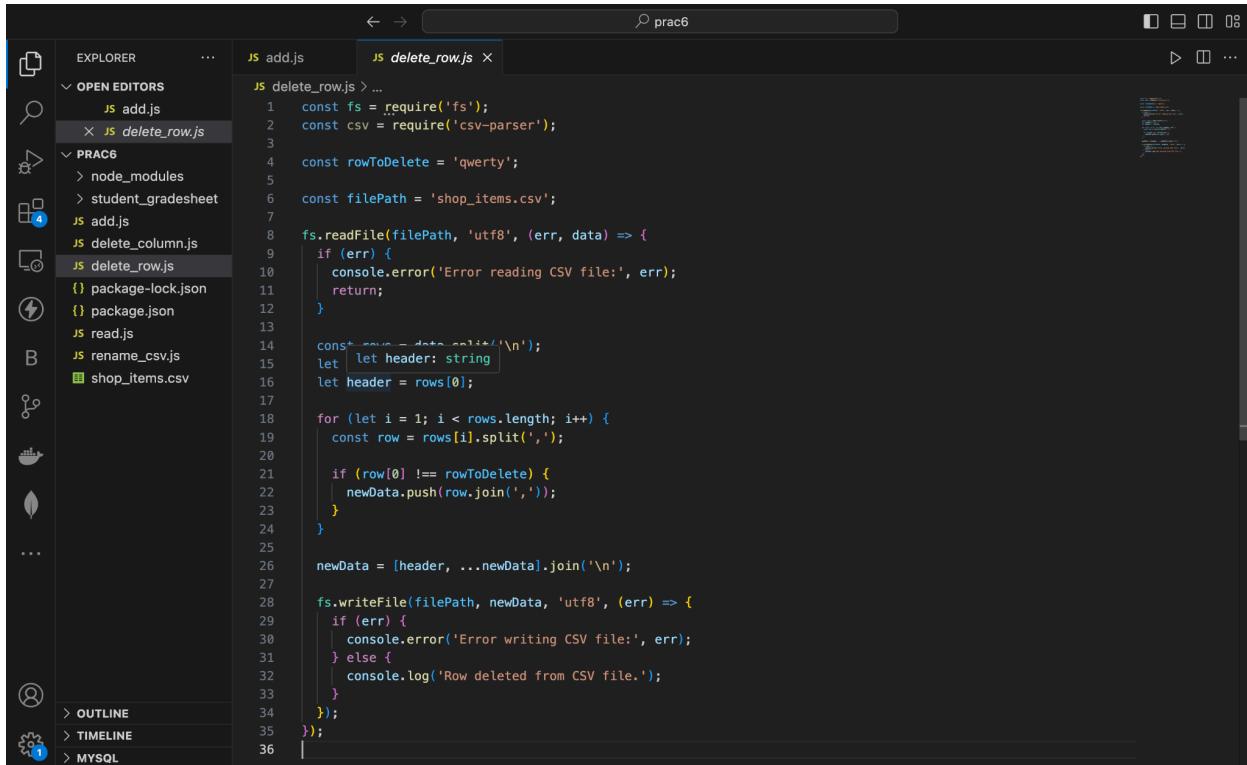
OUTPUT:

The screenshot shows a Mac desktop environment with three main windows open:

- Terminal Window:** The title bar says "prac6". It contains the command `node add` and the output "Item added to CSV file.".
- CSV Editor Window:** The title bar says "shop_items". It shows a table named "shop_items" with columns "name", "quantity", and "price". The data rows are: abc (1, 5000), Apples (50, 4000), new (3, 10), and latest (555, 5555).
- File Explorer Window:** The title bar says "Numbers". It shows a file structure under "PRAC6": "add.js", "delete_column.js", "delete_row.js", "package-lock.json", "package.json", "read.js", "rename_csv.js", and "shop_items.csv".

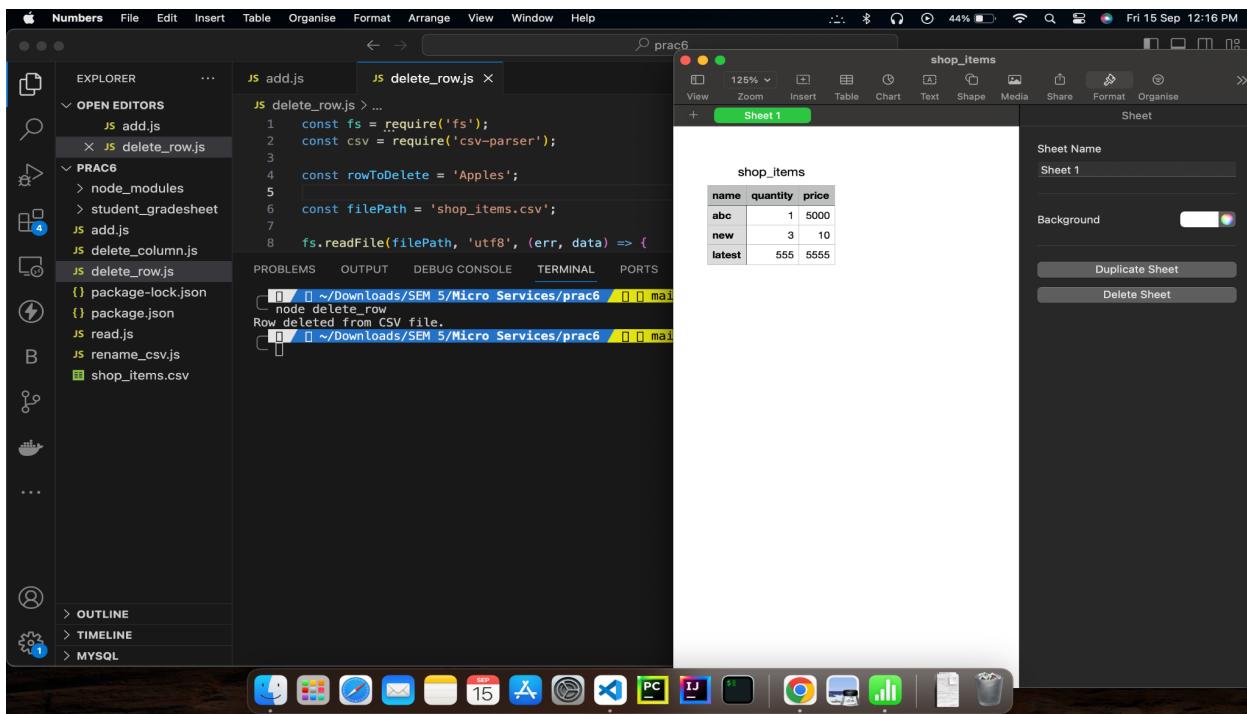
Practical 6.3: Deleting data from CSV

CODE: Delete data by row



```
JS delete_row.js > ...
1 const fs = require('fs');
2 const csv = require('csv-parser');
3
4 const rowToDelete = 'qwerty';
5
6 const filePath = 'shop_items.csv';
7
8 fs.readFile(filePath, 'utf8', (err, data) => {
9   if (err) {
10     console.error('Error reading CSV file:', err);
11     return;
12   }
13
14   const header = `-----\n`;
15   let header: string;
16   let header = rows[0];
17
18   for (let i = 1; i < rows.length; i++) {
19     const row = rows[i].split(',');
20
21     if (row[0] !== rowToDelete) {
22       newData.push(row.join(','));
23     }
24   }
25
26   newData = [header, ...newData].join('\n');
27
28   fs.writeFile(filePath, newData, 'utf8', (err) => {
29     if (err) {
30       console.error('Error writing CSV file:', err);
31     } else {
32       console.log('Row deleted from CSV file.');
33     }
34   });
35 });
36
```

OUTPUT:



The screenshot shows the 'shop_items' sheet in Numbers. The table contains the following data:

	name	quantity	price
abc	1	5000	
new	3	10	
latest	555	5555	

In the terminal tab of VS Code, the output shows the command run and the confirmation of the row being deleted:

```
[~] ~/Downloads/SEM 5/Micro Services/prac6 [1] main> node delete_row
Row deleted from CSV file.
```

Practical 6.3: Deleting data from CSV

CODE: Delete data by column

```
prac6

EXPLORER      ...
OPEN EDITORS  1 unsaved
JS add.js
JS delete_column.js ●
JS delete_column.js > ⚡ fs.readFile('utf8') callback
1 const fs = require('fs');
2 Const csv = require('csv-parser');
3
4 const columnToDelete = 'quantity';
5 const filePath = 'shop_items.csv';
6
7 fs.readFile(filePath, 'utf8', (err, data) => [
8   if (err) {
9     console.error('Error reading CSV file:', err);
10    return;
11  }
12
13  const rows = data.split('\n');
14  let newData = [];
15  let header = rows[0];
16
17  const columns = header.split(',');
18  const columnIndexToDelete = columns.indexOf(columnToDelete);
19
20  if (columnIndexToDelete === -1) {
21    console.error('Column not found in CSV file.');
22    return;
23  }
24
25  for (let i = 1; i < rows.length; i++) {
26    const row = rows[i].split(',');
27    row.splice(columnIndexToDelete, 1);
28    newData.push(row.join(','));
29  }
30
31  newData = [header.split(',').filter((_, idx) => idx !== columnIndexToDelete).join(',') , ...newData].join(',')
32
33  fs.writeFile(filePath, newData, 'utf8', (err) => {
34    if (err) {
35      console.error('Error writing CSV file:', err);
36    } else {
37      console.log(`Column ${columnToDelete} deleted from CSV file.`);
38    }
39  });
40}

OUTLINE
TIMELINE
MYSQL
```

OUTPUT:

The screenshot shows a Mac desktop environment with several open applications:

- VS Code (Left):** The Explorer sidebar shows files like `add.js`, `delete_column.js`, `node_modules`, `student_gradesheet`, and `shop_items_renamed.csv`. The `delete_column.js` file is open in the editor, containing code to delete a column from a CSV file.
- Terminal (Bottom Left):** Shows the command `node delete_column` running, with the output indicating the column 'quantity' was deleted from the CSV file.
- Numbers (Right):** A spreadsheet titled "shop_items_renamed" is open, showing a table with columns "name" and "price". The data includes rows for "abc" (price 5000), "new" (price 10), and "latest" (price 5555).
- System Dock (Bottom):** Shows icons for Finder, Mail, Safari, Calendar, Stocks, App Store, System Preferences, Terminal, and others.

Practical 6.4: Renaming csv

CODE:

The screenshot shows a Mac desktop environment with several open applications:

- VS Code (Frontmost):** The terminal tab is active, displaying the command `node rename_csv` and the output "File renamed successfully". The code editor shows a JavaScript file named `rename_csv.js` with the following content:

```
1 const fs = require('fs');
2
3 const oldFilePath = 'shop_items.csv';
4 const newFilePath = 'shop_items_renamed.csv';
5
6 fs.rename(oldFilePath, newFilePath, (error) => {
7   if (error) {
8     console.error('Error renaming file:', error);
9   } else {
10     console.log('File renamed successfully');
11   }
12 });
13 );
```

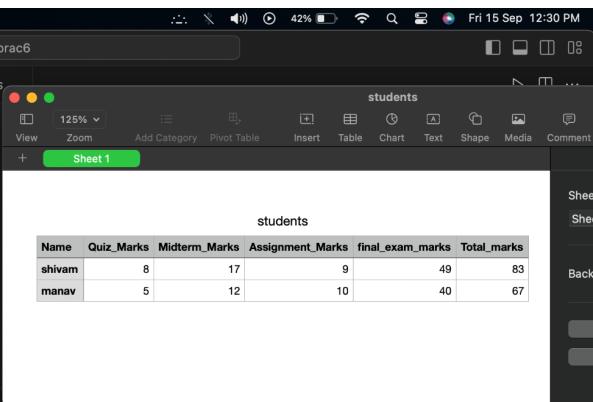
- Numbers:** A sheet titled "shop_items_renamed" is open, showing a table with three rows:

	name	quantity	price
abc	1	5000	
new	3	10	
latest	555	5555	

- Terminal:** Shows the command `node rename_csv` and the output "File renamed successfully".
- File Explorer:** Shows a folder named "PRAC6" containing various JavaScript files like `add.js`, `delete_row.js`, etc.
- System Dock:** Shows the Dock with various application icons.

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

→ Reading data from CSV



Name	Quiz_Marks	Midterm_Marks	Assignment_Marks	final_exam_marks	Total_marks
shivam	8	17	9	49	83
manav	5	12	10	40	67

The screenshot shows a Mac OS X desktop environment with the Numbers application open. The application window has tabs for 'EXPLORER', 'JS add.js', 'JS read_data.js', and 'JS delete_column.js'. The 'JS read_data.js' tab is active, displaying the following code:

```
const fs = require('fs');
const csv = require('csv-parser');

const results = [];

fs.createReadStream('students.csv')
  .pipe(csv())
  .on('data', (data) => {
    results.push(data);
  })
  .on('end', () => {
    console.log(results);
  });

The terminal below shows the output of running the script:
```

```
node read_data
[{"Name": "shivam", "Quiz_Marks": "8", "Midterm_Marks": "17", "Assignment_Marks": "9", "final_exam_marks": "49", "Total_marks": "83"}, {"Name": "manav", "Quiz_Marks": "5", "Midterm_Marks": "12", "Assignment_Marks": "10", "final_exam_marks": "40", "Total_marks": "67"}]
```

→ Adding data to CSV

The screenshot shows the Numbers application on a Mac. On the left, the sidebar lists files in the 'PRAC6' folder, including 'JS add_row.js', 'JS delete_column.js', 'JS delete_row.js', 'JS read_data.js', 'JS rename_csv.js', 'students.csv', 'JS add.js', 'JS delete_column.js', 'JS delete_row.js', 'package-lock.json', 'package.json', 'JS read.js', 'JS rename_csv.js', and 'shop_items_renamed.csv'. The main area displays the content of 'JS add_row.js':

```
1 const fs = require('fs');
2
3 const newItem = {
4   Name: 'latest',
5   Quiz_Marks: 9,
6   Midterm_Marks: 9,
7   Assignment_Marks: 9,
8   final_exam_marks: 9,
9   Total_marks: 36
10 };
11
12 const newRow = `${newItem.Name},${newItem.Quiz_Marks},${newItem.Midterm_Marks},${newItem.Assignment_Marks},$`+
13   `fs.appendFile('students.csv', '\n' + newRow, (err) => {
14     if (err) throw err;
15     console.log('Item added to CSV file.');
16   });
17 };
```

To the right, a new sheet titled 'Sheet 1' is shown with the heading 'students'. The data is:

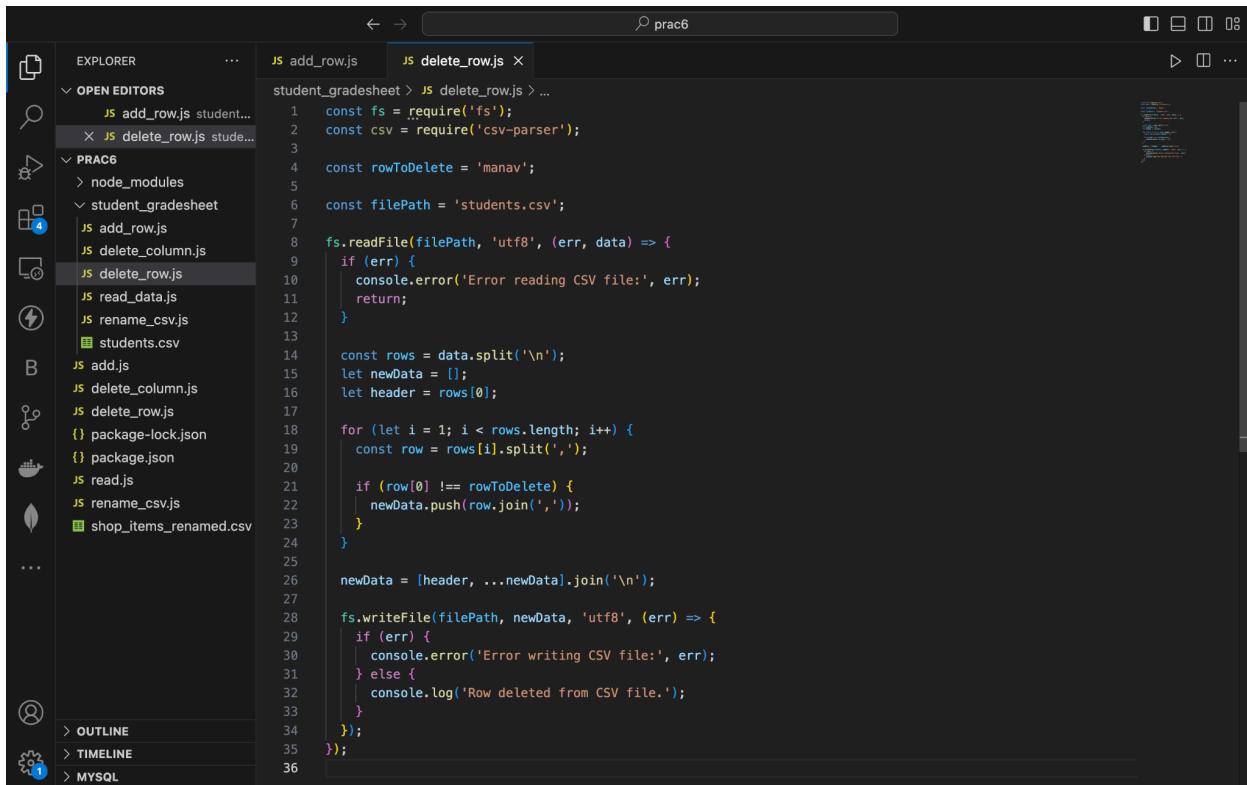
Name	Quiz_Marks	Midterm_Marks	Assignment_Marks	final_exam_marks	Total_marks
shivam	8	17	9	49	83
manav	5	12	10	40	67
latest	9	9	9	9	36

The terminal at the bottom shows two log entries:

```
[1] 12:31:54 PM ~/Downloads/SEM 5/Micro Services/prac6/student_gradesheet node add_row
Item added to CSV file.
[1] 12:32:03 PM ~/Downloads/SEM 5/Micro Services/prac6/student_gradesheet
```

→ Deleting data from CSV

CODE: Delete data by row



```
const fs = require('fs');
const csv = require('csv-parser');

const rowToDelete = 'manav';
const filePath = 'students.csv';

fs.readFile(filePath, 'utf8', (err, data) => {
  if (err) {
    console.error('Error reading CSV file:', err);
    return;
  }

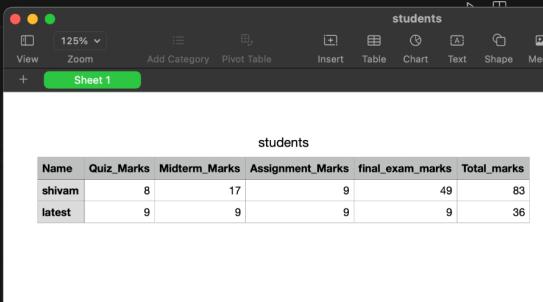
  const rows = data.split('\n');
  let newData = [];
  let header = rows[0];

  for (let i = 1; i < rows.length; i++) {
    const row = rows[i].split(',');
    if (row[0] !== rowToDelete) {
      newData.push(row.join(','));
    }
  }

  newData = [header, ...newData].join('\n');

  fs.writeFile(filePath, newData, 'utf8', (err) => {
    if (err) {
      console.error('Error writing CSV file:', err);
    } else {
      console.log('Row deleted from CSV file.');
    }
  });
});
```

OUTPUT:

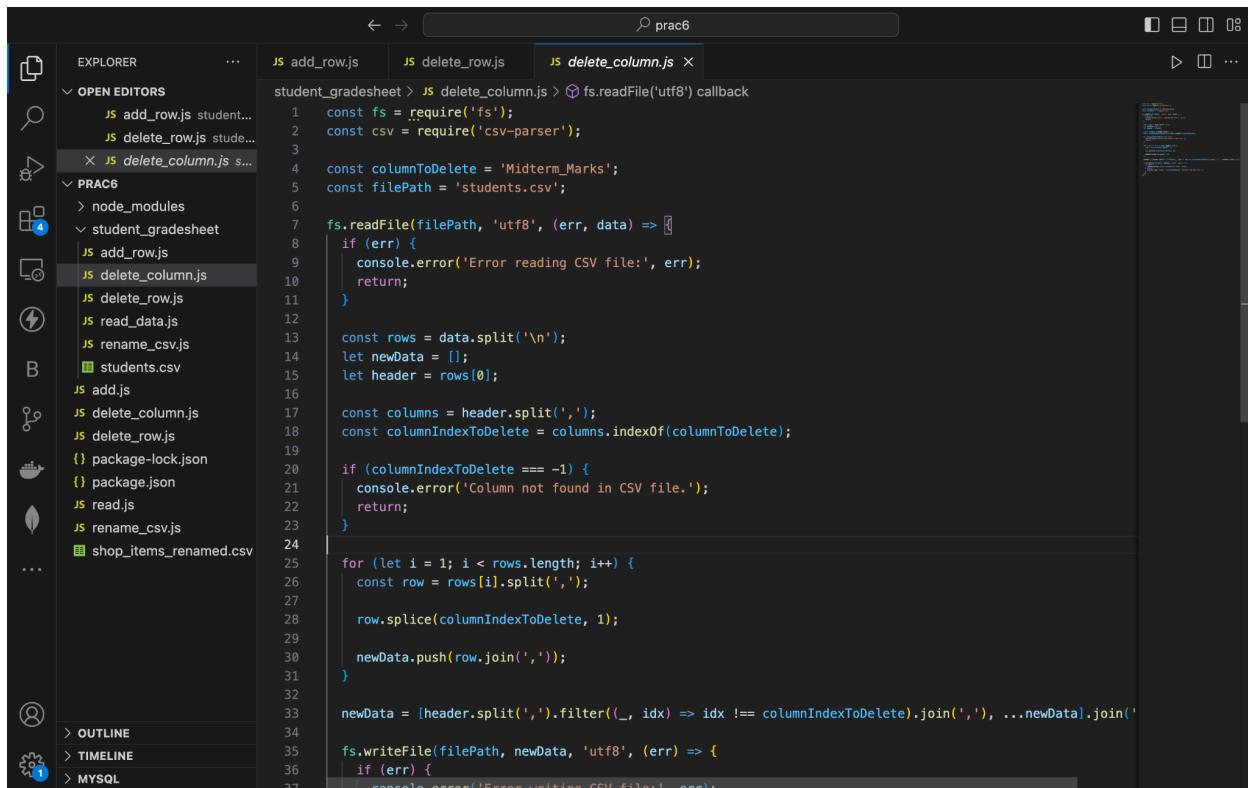


Name	Quiz_Marks	Midterm_Marks	Assignment_Marks	final_exam_marks	Total_marks
shivam	8	17	9	49	83
latest	9	9	9	9	36

The terminal below shows the command run and the confirmation message.

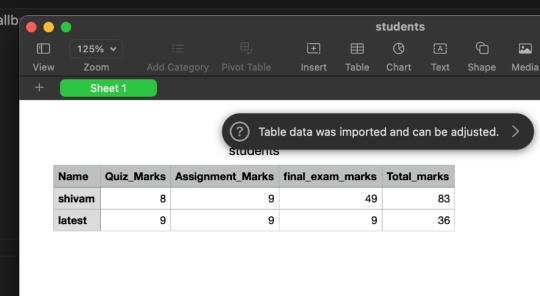
```
node delete_row
Row deleted from CSV file.
```

CODE: Delete data by column



```
student_gradesheet > JS delete_column.js > fs.readFile('utf8') callback
1 const fs = require('fs');
2 const csv = require('csv-parser');
3
4 const columnToDelete = 'Midterm_Marks';
5 const filePath = 'students.csv';
6
7 fs.readFile(filePath, 'utf8', (err, data) => {
8   if (err) {
9     console.error('Error reading CSV file:', err);
10    return;
11  }
12
13  const rows = data.split('\n');
14  let newData = [];
15  let header = rows[0];
16
17  const columns = header.split(',');
18  const columnIndexToDelete = columns.indexOf(columnToDelete);
19
20  if (columnIndexToDelete === -1) {
21    console.error('Column not found in CSV file.');
22    return;
23  }
24
25  for (let i = 1; i < rows.length; i++) {
26    const row = rows[i].split(',');
27
28    row.splice(columnIndexToDelete, 1);
29
30    newData.push(row.join(','));
31  }
32
33  newData = [header.split(',').filter((_, idx) => idx !== columnIndexToDelete)].join(',') + newData.join(',');
34
35  fs.writeFile(filePath, newData, 'utf8', (err) => {
36    if (err) {
37      console.error('Error writing CSV file:', err);
38    }
39  });
40});
```

OUTPUT:



Name	Quiz_Marks	Assignment_Marks	final_exam_marks	Total_marks
shivam	8	9	49	83
latest	9	9	9	36

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
zsh - student_gradesheet
node delete_column
Column 'Midterm_Marks' deleted from CSV file.
```

→ Renaming csv

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface on a Mac OS X desktop. The title bar indicates it's running on Fri 16 Sep 5:51 PM with battery at 34%.

The Explorer sidebar on the left lists several files and folders under the 'PRAC6' project, including 'node_modules', 'student_gradesheet', and various JavaScript files like 'add_row.js', 'delete_column.js', etc. A file named 'JS rename_csv.js' is currently selected.

The main editor area displays the code for 'JS rename_csv.js'. The code uses the Node.js 'fs' module to rename a file from 'students.csv' to 'students_result.csv'.

```
1 const fs = require('fs');
2
3 const oldFilePath = 'students.csv';
4 const newFilePath = 'students_result.csv';
5
6 // const newPath = 'students.csv';
7 // const oldPath = 'students_result.csv';
8
9
10 fs.rename(oldFilePath, newFilePath, (error) => {
11   if (error) {
12     console.error('Error renaming file:', error);
13   } else {
14     console.log('File renamed successfully');
15   }
16 });
17 }
```

The bottom right corner of the screen shows the macOS Dock with various application icons.

The Terminal tab is active, showing two entries from the zsh shell:

- [1] ~ /Downloads/SEM 5/Micro Services/prac6/student_gradesheet main !1 ?1 05:50:48 PM node rename_csv
- [2] ~ /Downloads/SEM 5/Micro Services/prac6/student_gradesheet main !1 ?1 05:50:55 PM File renamed successfully