

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

Scenario : 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_marks, Total_marks

Code :

1. Practical 6.1: Reading data from CSV

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

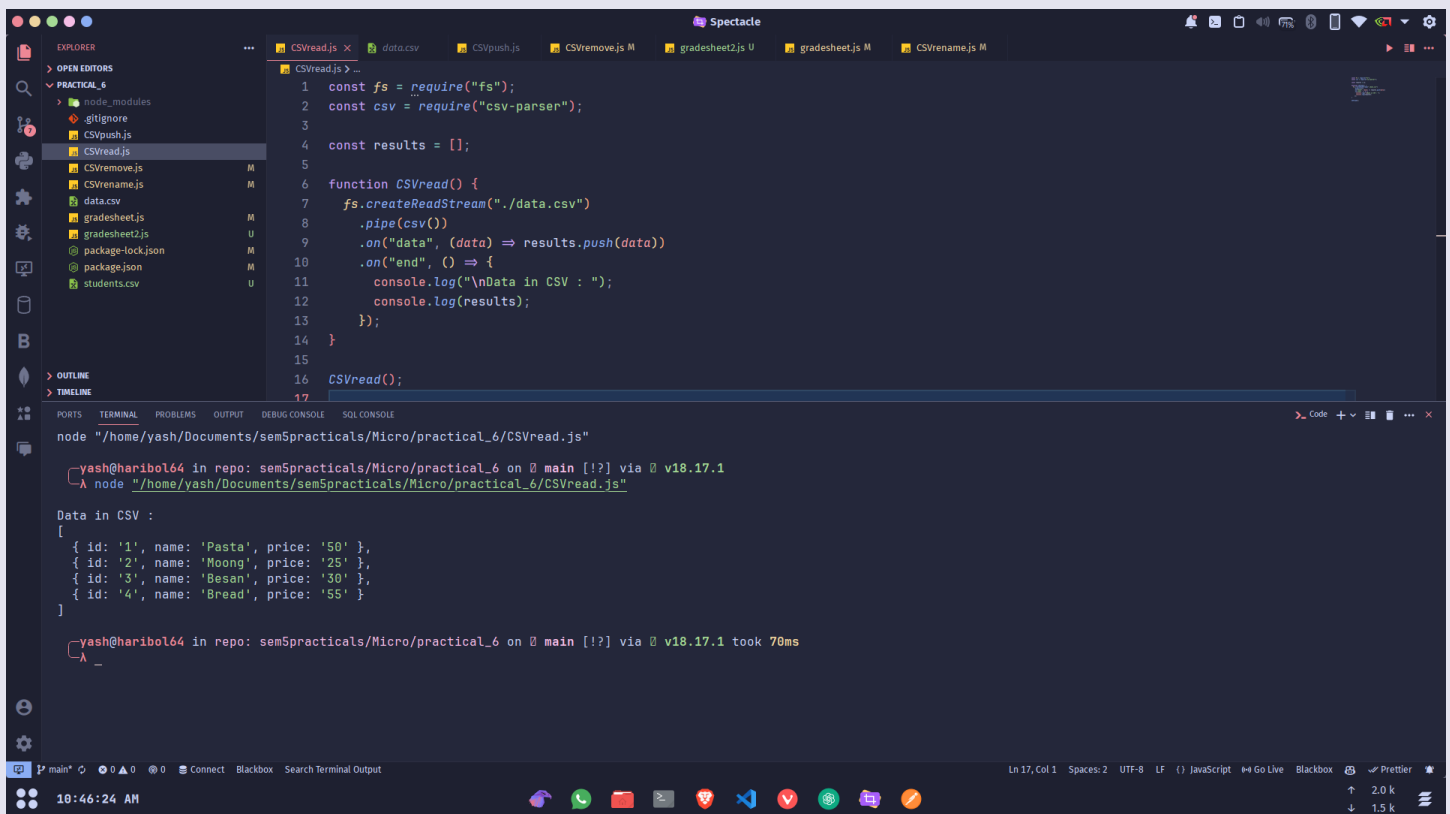
const results = [];

function CSVread() {
  fs.createReadStream("./data.csv")
    .pipe(csv())
    .on("data", (data) => results.push(data))
    .on("end", () => {
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

```
console.log("\nData in CSV : ");  
  
console.log(results);  
  
});  
  
}  
  
CSVread();
```

Screenshot :



The screenshot shows a VS Code editor window with the following content:

Explorer: A file tree on the left showing the project structure. The file `CSVread.js` is selected.

Editor: The main workspace displays the code in `CSVread.js`:

```
1 const fs = require("fs");  
2 const csv = require("csv-parser");  
3  
4 const results = [];  
5  
6 function CSVread() {  
7   fs.createReadStream("./data.csv")  
8     .pipe(csv())  
9     .on("data", (data) => results.push(data))  
10    .on("end", () => {  
11      console.log("\nData in CSV : ");  
12      console.log(results);  
13    });  
14 }  
15  
16 CSVread();  
17
```

Terminal: The terminal at the bottom shows the command `node "/home/yash/Documents/sem5practicals/Micro/practical_6/CSVread.js"` being executed. The output is:

```
yash@haribol64 in repo: sem5practicals/Micro/practical_6 on 0 main [!?] via v18.17.1  
node "/home/yash/Documents/sem5practicals/Micro/practical_6/CSVread.js"  
  
Data in CSV :  
[  
  { id: '1', name: 'Pasta', price: '50' },  
  { id: '2', name: 'Moong', price: '25' },  
  { id: '3', name: 'Besan', price: '30' },  
  { id: '4', name: 'Bread', price: '55' }  
]  
  
yash@haribol64 in repo: sem5practicals/Micro/practical_6 on 0 main [!?] via v18.17.1 took 70ms
```

The status bar at the bottom indicates the file is `Ln 17, Col 1`, using `UTF-8` encoding, and the editor is in `JavaScript` mode.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

2. Practical 6.2: Adding data to CSV

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

function CSVpush() {
  const DATA = [];
  fs.createReadStream("./data.csv")
    .pipe(csv())
    .on("data", (data) => DATA.push(data))
    .on("end", () => {
      const updates = { id: "5", name: "Cheese", price: "125" };
      DATA.push(updates);

      const writeStream = fs.createWriteStream("./data.csv");
      writeStream.write("id,name,price\n");
      DATA.forEach((row) => {
        writeStream.write(`${row.id},${row.name},${row.price}\n`);
      });
      writeStream.end(() => {
        console.log("Data added successfully.");
      });
    });
}
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

`CSVpush()`;

Screenshot :

The screenshot shows the Visual Studio Code editor with a project named 'practical_6'. The Explorer sidebar on the left shows the file structure, including 'node_modules', 'gitignore', 'CSVpush.js', 'CSVread.js', 'CSVremove.js', 'CSVrename.js', 'data.csv', 'gradesheet.js', 'gradesheet2.js', 'package-lock.json', 'package.json', and 'students.csv'. The main editor area displays the content of 'CSVpush.js'.

```
1 const fs = require("fs");
2 const csv = require("csv-parser");
3
4 function CSVpush() {
5   const DATA = [];
6   fs.createReadStream("./data.csv")
7     .pipe(csv())
8     .on("data", (data) => DATA.push(data))
9     .on("end", () => {
10       const updates = { id: "5", name: "Cheese", price: "125" };
11       DATA.push(updates);
12
13       const writeStream = fs.createWriteStream("./data.csv");
14       writeStream.write("id,name,price\n");
15       DATA.forEach((row) => {
16         writeStream.write(`${row.id},${row.name},${row.price}\n`);
17       });
18     });
19 }
```

The terminal window at the bottom shows the execution of the script:

```
node "/home/yash/Documents/sem5practicals/Micro/practical_6/CSVpush.js"
yash@haribol64 in repo: sem5practicals/Micro/practical_6 on main [!?] via v18.17.1
node "/home/yash/Documents/sem5practicals/Micro/practical_6/CSVpush.js"
Data added successfully.
yash@haribol64 in repo: sem5practicals/Micro/practical_6 on main [!?] via v18.17.1 took 60ms
cat data.csv
File: data.csv
id,name,price
1,Pasta,50
2,Moong,25
3,Besan,30
4,Bread,55
5,Cheese,125
yash@haribol64 in repo: sem5practicals/Micro/practical_6 on main [!?] via v18.17.1 took 23ms
_
```

The status bar at the bottom indicates the file is 'main.js', line 25, column 1, with 2 spaces, in UTF-8 encoding, using the JavaScript language, with the Go Live extension, in the Blackbox theme, and the Prettier formatter. The system clock shows 10:49:16 AM.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

3. Practical 6.3: Deleting data from CSV

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

function CSVremove() {
  const DATA = [];
  fs.createReadStream("./data.csv")
    .pipe(csv())
    .on("data", (data) => DATA.push(data))
    .on("end", () => {
      const id = "5";
      const updated = DATA.filter((row) => row.id !== id);

      const writeStream = fs.createWriteStream("./data.csv");
      writeStream.write("id,name,price\n");
      updated.forEach((row) => {
        writeStream.write(`${row.id},${row.name},${row.price}\n`);
      });
      writeStream.end(() => {
        console.log("Data removed successfully.");
      });
    });
}
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

CSVremove();

Screenshot :

```
1 const fs = require("fs");
2 const csv = require("csv-parser");
3
4 function CSVremove() {
5   const DATA = [];
6   fs.createReadStream("./data.csv")
7     .pipe(csv())
8     .on("data", (data) => DATA.push(data))
9     .on("end", () => {
10       const id = "5";
11       const updated = DATA.filter((row) => row.id !== id);
12
13       const writeStream = fs.createWriteStream("./data.csv");
14       writeStream.write("id,name,price\n");
15       updated.forEach((row) => {
16         writeStream.write(`${row.id},${row.name},${row.price}\n`);
17       });
18     });
19 }
```

```
node "/home/yash/Documents/sem5practicals/Micro/practical_6/CSVremove.js"
yash@haribol64 in repo: sem5practicals/Micro/practical_6 on  main [!?] via  v18.17.1
node "/home/yash/Documents/sem5practicals/Micro/practical_6/CSVremove.js"
Data removed successfully.
yash@haribol64 in repo: sem5practicals/Micro/practical_6 on  main [!?] via  v18.17.1 took 87ms
cat data.csv
File: data.csv
id,name,price
1,Pasta,50
2,Moong,25
3,Besan,30
4,Bread,55
yash@haribol64 in repo: sem5practicals/Micro/practical_6 on  main [!?] via  v18.17.1 took 15ms
_
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

4. Practical 6.4: Renaming CSV

```
const fs = require("fs");  
  
const newName = "grocery.csv";  
  
function CSVrename() {  
  fs.rename("./data.csv", newName, (err) => {  
    if (err) {  
      console.error("Error renaming file : ", err);  
    } else {  
      console.log("File renamed successfully!");  
    }  
  });  
}  
  
CSVrename();
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

Screenshot :

The screenshot shows the Visual Studio Code interface with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'PRACTICAL_6' with several files including 'node_modules', '.gitignore', 'CSVpush.js', 'CSVread.js', 'CSVremove.js', 'CSVrename.js', 'gradesheet.js', 'gradesheet2.js', 'grocery.csv', 'package-lock.json', 'package.json', and 'students.csv'. The code editor shows the 'CSVrename.js' file with the following code:

```
1 const fs = require("fs");
2 const newName = "grocery.csv";
3
4 function CSVrename() {
5   fs.rename("./data.csv", newName, (err) => {
6     if (err) {
7       console.error("Error renaming file : ", err);
8     } else {
9       console.log("File renamed successfully!");
10    }
11  });
12 }
13
14 CSVrename();
15
```

The terminal at the bottom shows the command 'node "/home/yash/Documents/sem5practicals/Micro/practical_6/CSVrename.js"' being executed, resulting in the output 'File renamed successfully!'. Below this, the terminal shows the command 'ls' being executed, resulting in a list of files and their sizes.

5. 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_marks, Total_marks

```
const express = require("express");
const fs = require("fs");
const csv = require("csv-parser");
const app = express();
const port = 4200;

app.use(express.json());
```


Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

```
const dataFilePath = "./students.csv";

function readCSV(callback) {
  const results = [];
  fs.createReadStream(dataFilePath)
    .pipe(csv())
    .on("data", (data) => results.push(data))
    .on("end", () => {
      callback(results);
    });
}

function writeCSV(data, callback) {
  const writeStream = fs.createWriteStream(dataFilePath);
  writeStream.write(
    "StudentName,Quiz_Marks,Mid-term_Marks,Assignment_Marks,Final_Exam_Marks,T  
otal_Marks\n"
  );
  data.forEach((row) => {
    writeStream.write(
      `${row.StudentName},${row.Quiz_Marks},${row["Mid-term_Marks"]},${row["Assi  
gnment_Marks"]},${row["Final_Exam_Marks"]},${row.Total_Marks}\n`
    );
  });
}
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

```
});  
  
writeStream.end(() => {  
  callback();  
});  
}  
  
app.get("/students", (req, res) => {  
  readCSV((data) => {  
    res.json(data);  
  });  
});  
  
app.get("/students/:name", (req, res) => {  
  const studentName = req.params.name;  
  
  readCSV((data) => {  
    const student = data.find((student) => student.StudentName ===  
      studentName);  
  
    if (student) {  
      res.json(student);  
    } else {  
      res.status(404).json({ message: "Student not found" });  
    }  
  })  
});
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

```
});  
});  
  
app.post("/students", (req, res) => {  
  readCSV((data) => {  
    const newStudent = req.body;  
    data.push(newStudent);  
  
    writeCSV(data, () => {  
      res.status(201).json({ message: "Student added successfully" });  
    });  
  });  
});  
  
app.put("/students/:name", (req, res) => {  
  const studentName = req.params.name;  
  const updatedStudent = req.body;  
  
  readCSV((data) => {  
    const studentIndex = data.findIndex(  
      (student) => student.StudentName === studentName  
    );  
  
    if (studentIndex === -1) {
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

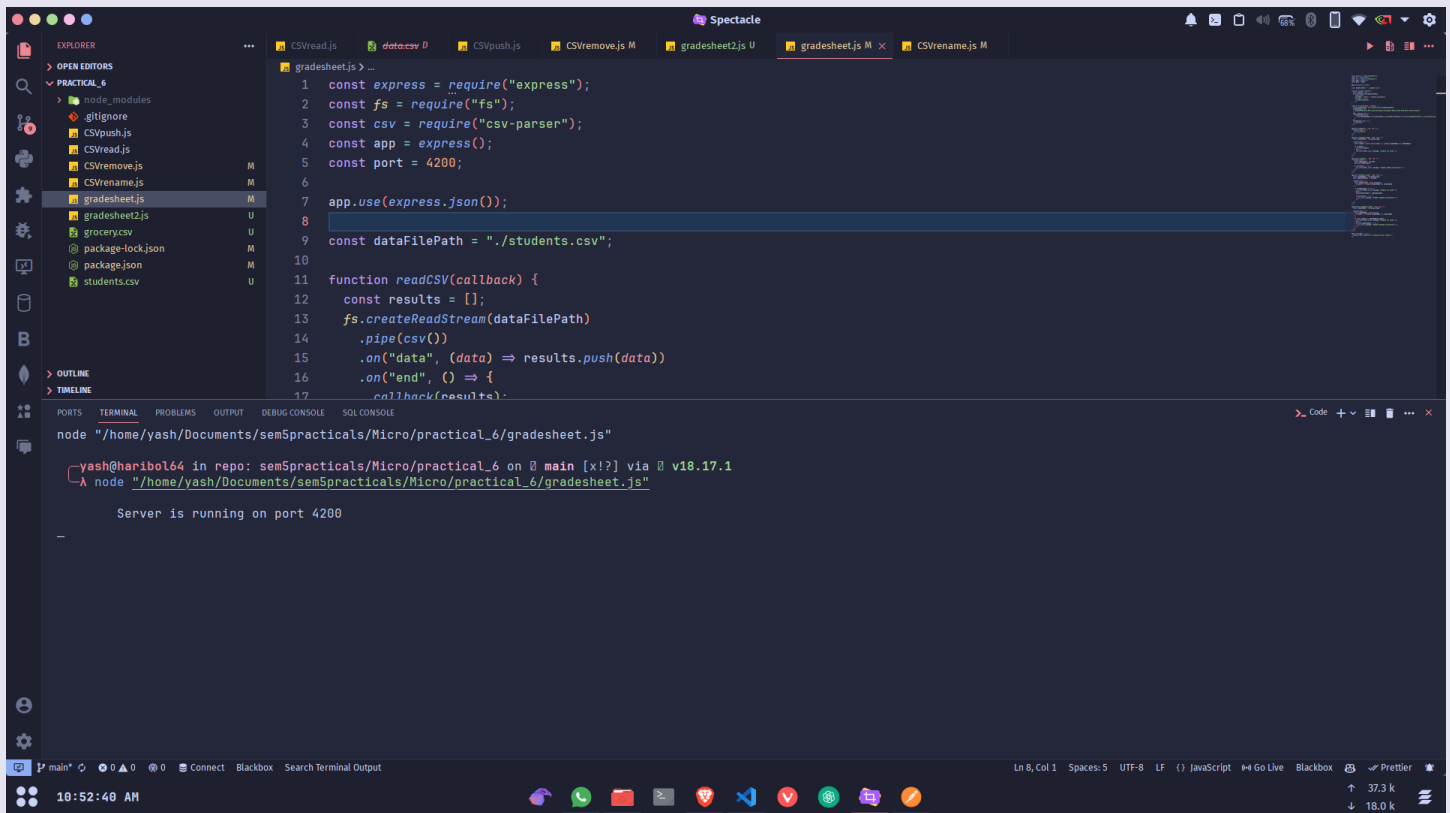
```
res.status(404).json({ message: "Student not found" });  
} else {  
  data[studentIndex] = updatedStudent;  
  
  writeCSV(data, () => {  
    res.json({ message: "Student updated successfully" });  
  });  
}  
});  
});  
  
app.delete("/students/:name", (req, res) => {  
  const studentName = req.params.name;  
  
  readCSV((data) => {  
    const updatedData = data.filter(  
      (student) => student.StudentName !== studentName  
    );  
  
    if (data.length === updatedData.length) {  
      res.status(404).json({ message: "Student not found" });  
    } else {  
      writeCSV(updatedData, () => {  
        res.json({ message: "Student deleted successfully" });  
      });  
    }  
  });  
});
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

```
});  
  
}  
  
});  
  
});  
  
app.listen(port, () => {  
  console.log(`\n\tServer is running on port ${port}`);  
});
```

Screenshots :

a. Running local server



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

b. GET all details

The screenshot displays a development environment with two main windows. On the left, Visual Studio Code shows a file named `students.csv` with the following content:

StudentName	Quiz_Marks	Mid-term_Marks	Assignment_Marks	Final_Exam_Marks	Total_Marks
John Smith	90	85	92	88	355
Alice Johnson	88	92	78	95	353
Bob Brown	76	81	89	70	316
Emily Davis	95	88	92	90	365

Below the editor, a terminal window shows the command `node /home/yash/Documents/sem5practicals/Micro/practical_6/gradesheet.js` being executed, with the output `Server is running on port 4200`.

On the right, the Spectacle REST client shows a GET request to `http://localhost:4200/students/`. The response is a JSON array of three student objects, with a status of 200 OK and a size of 789 B.

```
[{"StudentName": "John Smith", "Quiz_Marks": "90", "Mid-term_Marks": "85", "Assignment_Marks": "92", "Final_Exam_Marks": "88", "Total_Marks": "355"}, {"StudentName": "Alice Johnson", "Quiz_Marks": "88", "Mid-term_Marks": "92", "Assignment_Marks": "78", "Final_Exam_Marks": "95", "Total_Marks": "353"}, {"StudentName": "Bob Brown", "Quiz_Marks": "76", "Mid-term_Marks": "81", "Assignment_Marks": "89", "Final_Exam_Marks": "70", "Total_Marks": "316"}]
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

c. GET details by name

The screenshot displays a development environment with Visual Studio Code on the left and a REST client (Spectacle) on the right.

Visual Studio Code:

- EXPLORER:** Shows the file structure of a project named 'practical_6'. The file 'students.csv' is selected.
- students.csv:** Contains the following data:

StudentName	Quiz_Marks	Mid-term_Marks
John Smith	90	85, 92, 88, 355
Alice Johnson	88	92, 78, 95, 353
Bob Brown	76	81, 89, 70, 316
Emily Davis	95	88, 92, 90, 365
- TERMINAL:** Shows the command to run the application: `node "/home/yash/Documents/sem5practicals/Micro/practical_6/gradesheet.js"`. The output is: `Server is running on port 4200`.

Spectacle (REST Client):

- URL:** `http://localhost:4200/students/Bob Brown`
- Method:** GET
- Body:** The response is a JSON object:

```
{  "StudentName": "Bob Brown",  "Quiz_Marks": "76",  "Mid-term_Marks": "81",  "Assignment_Marks": "89",  "Final_Exam_Marks": "70",  "Total_Marks": "316"}
```
- Status:** 200 OK, Time: 11 ms, Size: 371 B

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

d. Add details

The screenshot displays two applications side-by-side. On the left is Visual Studio Code, showing a project named 'practical_6' with a file explorer on the left. The 'students.csv' file is open in the editor, containing a CSV with student details. The terminal at the bottom shows the command to run a Node.js script, and the output indicates the server is running on port 4200.

On the right is the Postman application, showing a REST client setup for a POST request to 'http://localhost:4200/students/'. The request body is a JSON object representing a student's details, including name, quiz marks, mid-term marks, assignment marks, final exam marks, and total marks. The response shows a status of 201 Created with a message: 'Student added successfully'.

Visual Studio Code - students.csv

StudentName	Quiz_Marks	Mid-term_Marks
John Smith	90,85,92,88,355	
Alice Johnson	88,92,78,95,353	
Bob Brown	76,81,89,70,316	
Emily Davis	95,88,92,90,365	
Ellis Brown	76,82,89,71,318	

Postman - POST Request

URL: http://localhost:4200/students/

Method: POST

Body (JSON):

```
{  "StudentName": "Ellis Brown",  "Quiz_Marks": "76",  "Mid-term_Marks": "82",  "Assignment_Marks": "89",  "Final_Exam_Marks": "71",  "Total_Marks": "318"}
```

Postman - Response

Status: 201 Created | Time: 19 ms | Size: 280 B

Body (JSON):

```
{  "message": "Student added successfully"}
```


Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

e. Update details

The screenshot displays a development environment with two main windows. On the left, Visual Studio Code shows a file named `students.csv` with the following data:

StudentName	Quiz_Marks	Mid-term_Marks
John Smith	90,85,92,88,355	
Alice Johnson	88,92,78,95,353	
Bob Brown	76,81,89,70,316	
Emily Davis	95,88,92,90,365	
Ellis Brown	77,82,89,71,319	

Below the code editor, a terminal window shows the command `node "/home/yash/Documents/sem5practicals/Micro/practical_6/gradesheet.js"` being executed, with the output `Server is running on port 4200`.

On the right, the Spectacle REST client is configured with a PUT request to `http://localhost:4200/students/Ellis Brown`. The request body is a JSON object:

```
{  "StudentName": "Ellis Brown",  "Quiz_Marks": "77",  "Mid-term_Marks": "82",  "Assignment_Marks": "89",  "Final_Exam_Marks": "71",  "Total_Marks": "319"}
```

The response status is 200 OK, and the response body is:

```
{  "message": "Student updated successfully"}
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 6

f. Delete details

The screenshot displays a development environment with Visual Studio Code on the left and Postman on the right.

Visual Studio Code:

- The Explorer sidebar shows a project structure for 'practical_6' with files like `gradesheet.js`, `students.csv`, and various utility scripts.
- The Editor shows the `students.csv` file with the following data:

StudentName	Quiz_Marks	Mid-term_Marks
John Smith	90	85,92,88,355
Alice Johnson	88	92,78,95,353
Bob Brown	76	81,89,70,316
Emily Davis	95	88,92,90,365
- The Terminal shows the command `node "/home/yash/Documents/sem5practicals/Micro/practical_6/gradesheet.js"` being executed, with the output `Server is running on port 4200`.

Postman:

- The interface shows a DELETE request to `http://localhost:4200/students/Ellis Brown`.
- The request body is in JSON format:

```
{  "StudentName": "Ellis Brown",  "Quiz_Marks": "77",  "Mid-term_Marks": "82",  "Assignment_Marks": "89",  "Final_Exam_Marks": "71",  "Total_Marks": "319"}
```
- The response status is `200 OK` with a message: `"message": "Student deleted successfully"`.