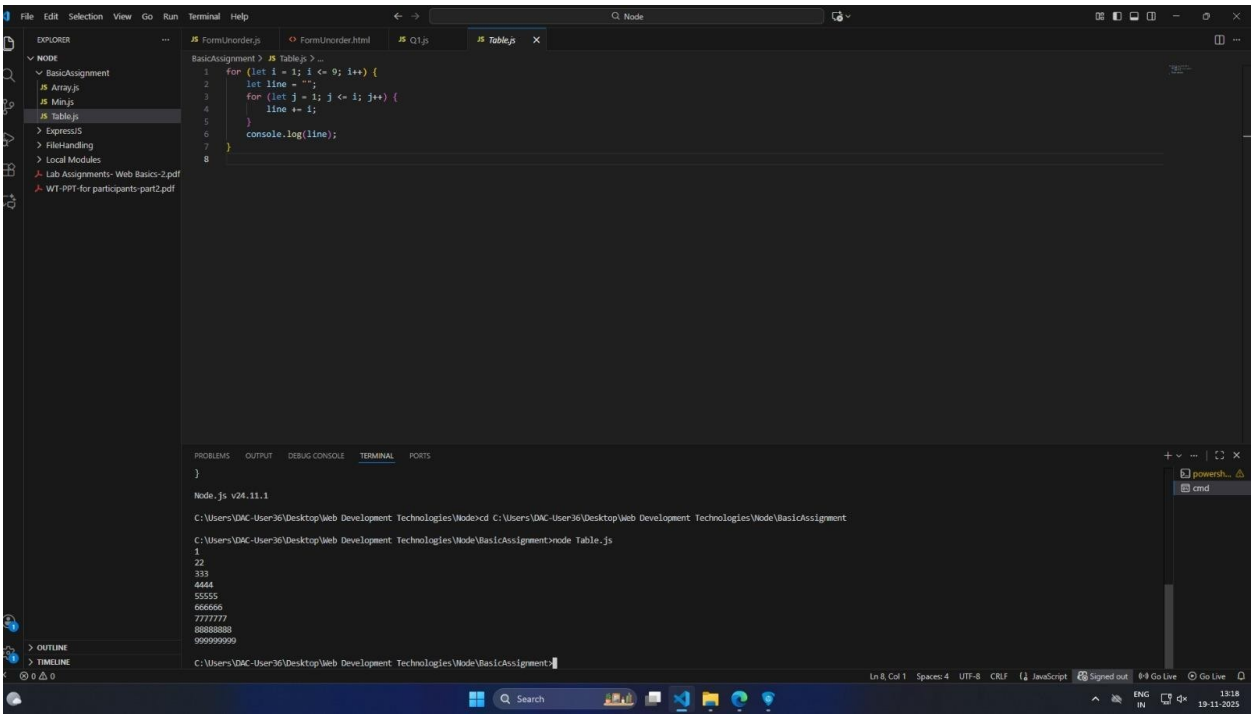
WPT in Assignment 2

Q1.

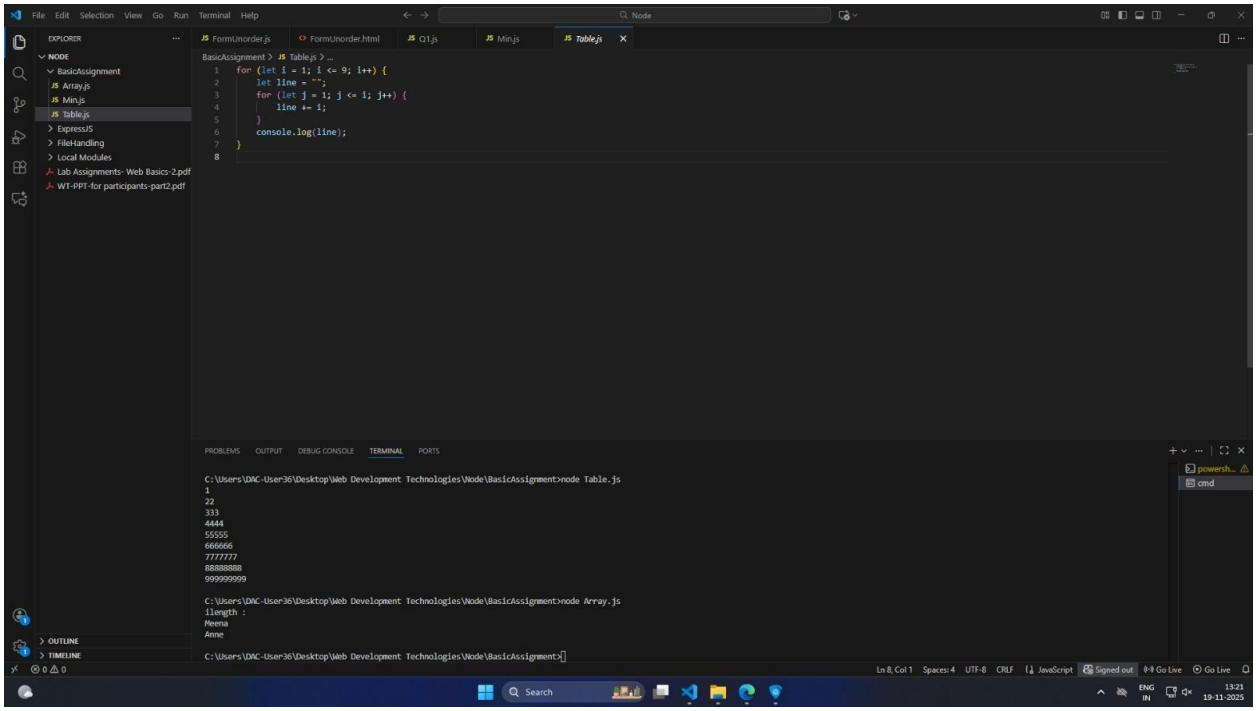
```
for (let i = 1; i <= 9; i++) {
let line = "";     for (let j = 1;
j <= i; j++) {         line += i;
    }
    console.log(line);
}
```



Q2.

```
for (let i = 1; i <= 9; i++) {
   let line = "";     for (let j =
1; j <= i; j++) {         line +=
i;
```

```
    }
    console.log(line);
}
```

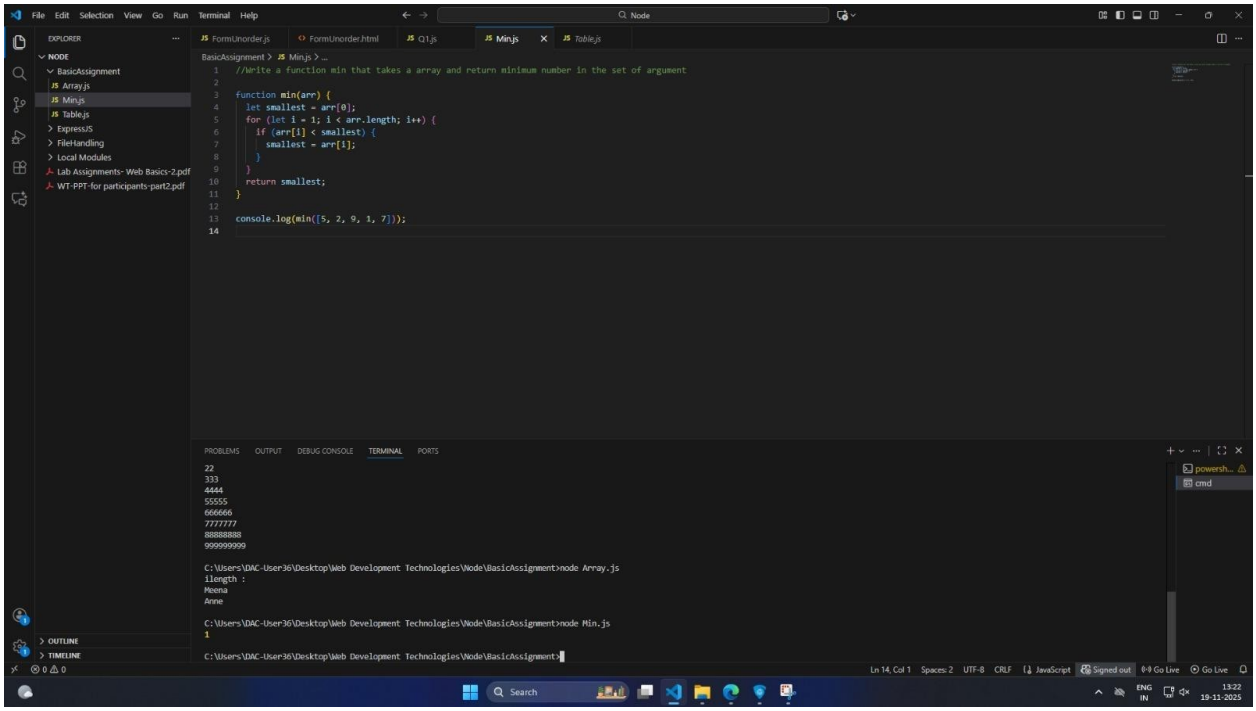


Q3.

//Write a function min that takes a array and return minimum number in the set of argument

```
function min(arr) {   let smallest =
arr[0];   for (let i = 1; i < arr.length;
i++)  {          if  (arr[i]  <  smallest)
{      smallest = arr[i];
    }
  }
  return smallest;
}
```

console.log(min([5, 2, 9, 1, 7]));



Local Modules :

Q1.

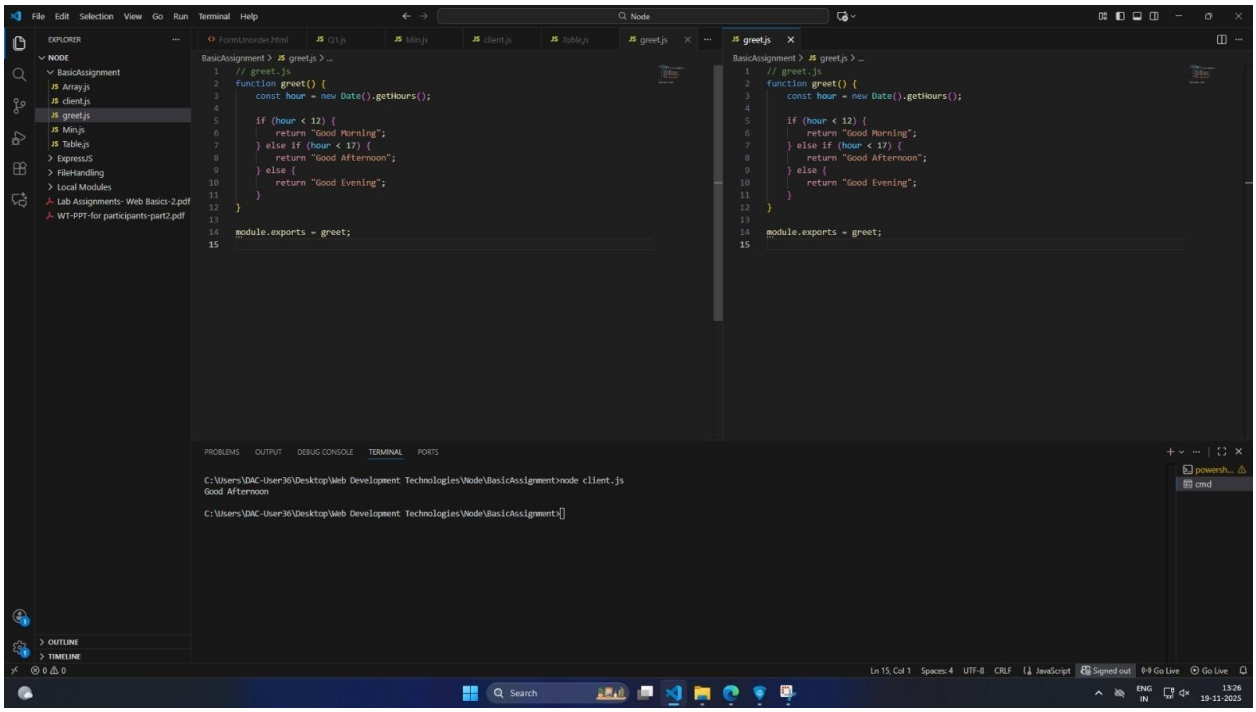
```
// greet.js function
greet() {
   const hour = new Date().getHours();

   if (hour < 12) {        return
"Good Morning";     } else if
(hour < 17) {        return
"Good Afternoon";     } else {

       return "Good Evening";
   }
}
```

module.exports = greet;

// client.js const greet =

require('./greet');

console.log(greet());



Q2.

// calc.js module.exports =

{

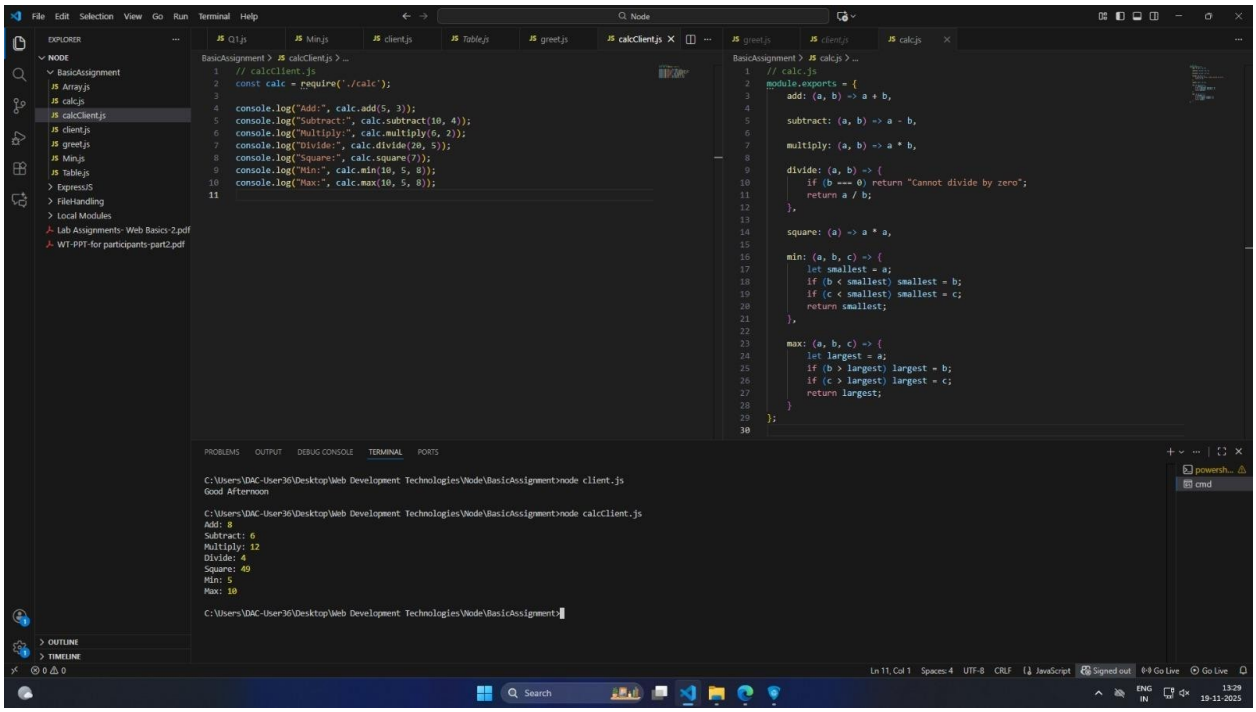
add: (a, b) => a + b,

subtract: (a, b) => a - b,

multiply: (a, b) => a * b,

divide: (a, b) => {

```
    if (b === 0) return "Cannot divide by zero";
return a / b;
  },

  square: (a) => a * a,

  min: (a, b, c) => {    let smallest
= a;    if (b < smallest) smallest =
b;    if (c < smallest) smallest = c;
return smallest;
  },

  max: (a, b, c) => {    let
largest = a;    if (b > largest)
largest = b;    if (c > largest)
largest = c;    return largest;
  }
};
// calcClient.js
const calc = require('./calc');

console.log("Add:", calc.add(5, 3)); console.log("Subtract:",
calc.subtract(10, 4)); console.log("Multiply:", calc.multiply(6,
2)); console.log("Divide:", calc.divide(20, 5));
console.log("Square:", calc.square(7)); console.log("Min:",
calc.min(10, 5, 8)); console.log("Max:", calc.max(10, 5, 8));
```



Q3.

// circle.js module.exports = { calcArea: (radius) => Math.PI * radius * radius,

```
calcCircumference: (radius) => 2 * Math.PI * radius,

calcDiameter: (radius) => 2 * radius
```

}; // rectangle.js module.exports = { calcArea: (length, breadth) => length * breadth,

```
calcPerimeter: (length, breadth) => 2 * (length + breadth)
```

}; // triangle.js module.exports = { isEquilateral: (a, b, c) => a === b && b === c,

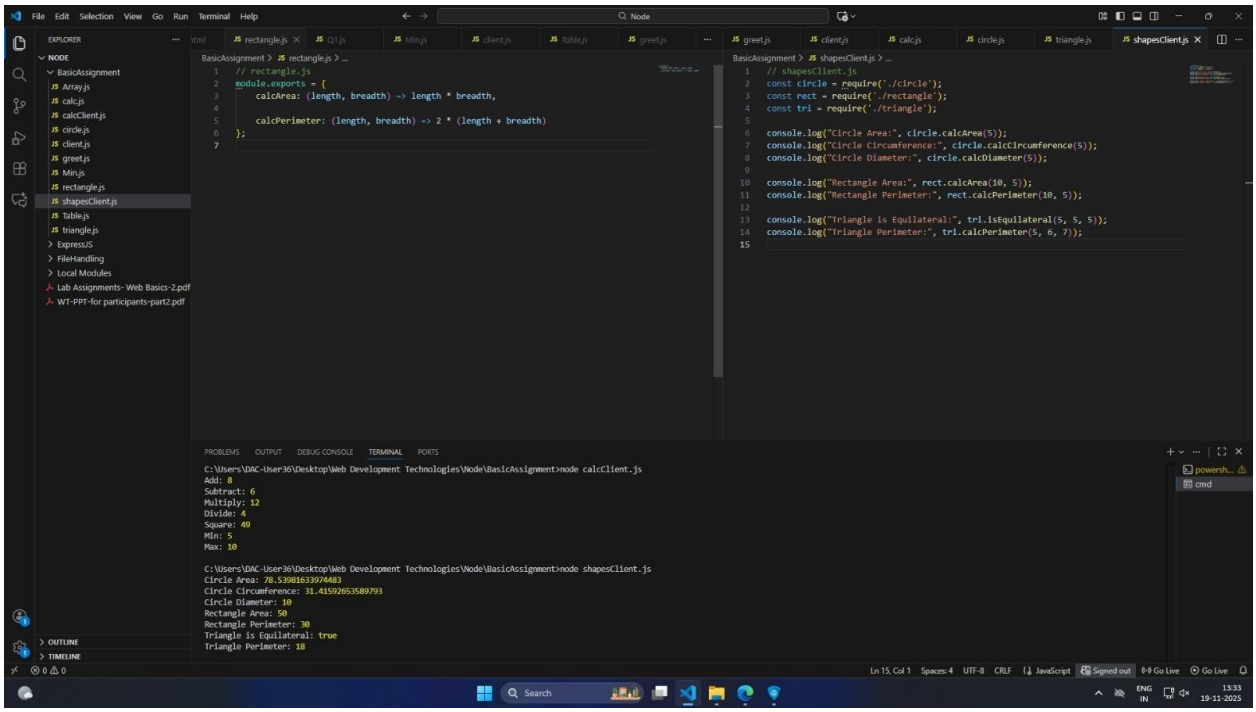
```
calcPerimeter: (a, b, c) => a + b + c
```

}; // shapesClient.js const circle = require('./circle'); const rect = require('./rectangle'); const tri = require('./triangle');

console.log("Circle Area:", circle.calcArea(5)); console.log("Circle Circumference:", circle.calcCircumference(5)); console.log("Circle Diameter:", circle.calcDiameter(5));

console.log("Rectangle Area:", rect.calcArea(10, 5)); console.log("Rectangle Perimeter:", rect.calcPerimeter(10, 5));

console.log("Triangle is Equilateral:", tri.isEquilateral(5, 5, 5)); console.log("Triangle Perimeter:", tri.calcPerimeter(5, 6, 7));



File Handling :

Q1.

// names.js const fs =

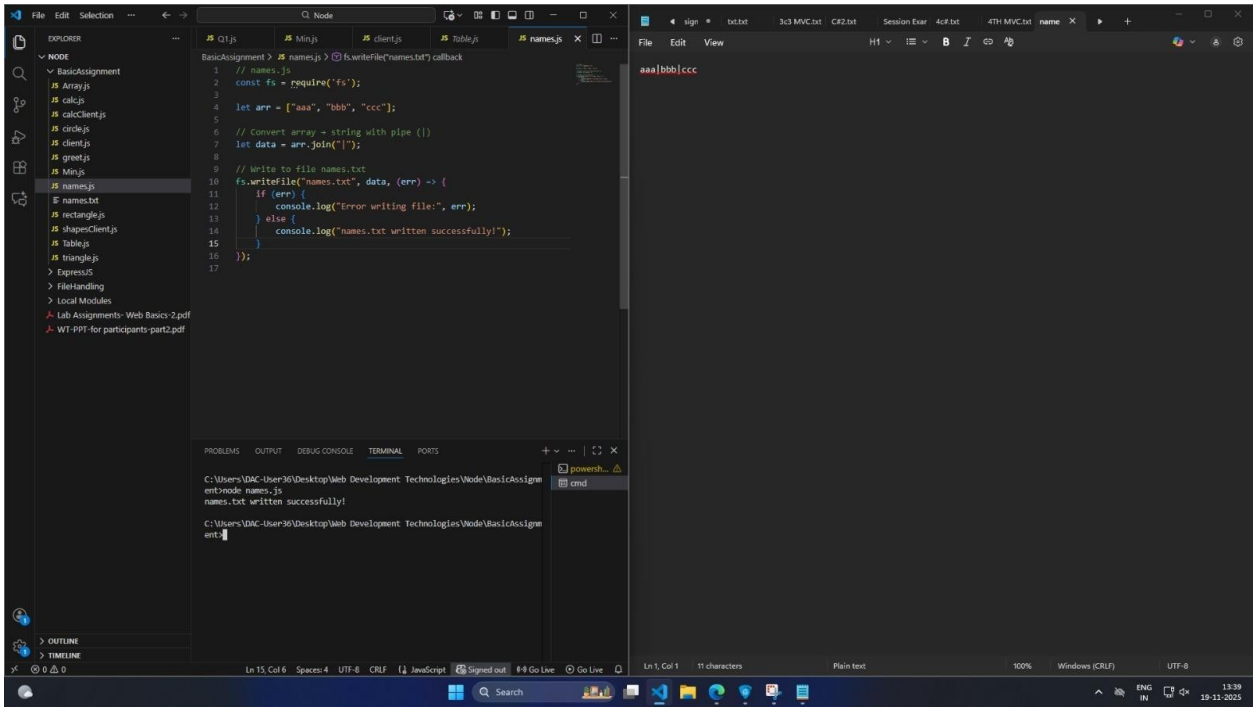
require('fs');

let arr = ["aaa", "bbb", "ccc"];

// Convert array → string with pipe (|)

let data = arr.join("|");

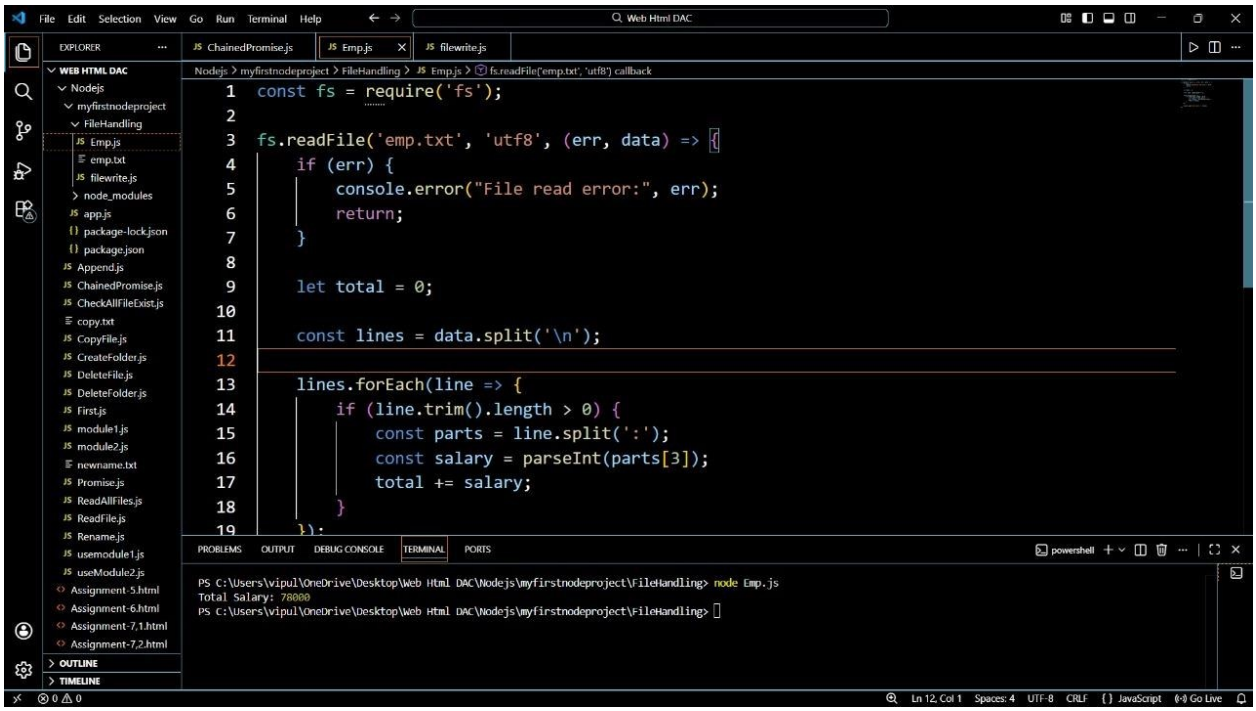
```
// Write to file names.txt fs.writeFile("names.txt",
data, (err) => {
   if (err) {
      console.log("Error writing file:", err);
   } else {
      console.log("names.txt written successfully!");
   }
});
```



Q2.

```
const fs = require('fs');


fs.readFile('emp.txt', 'utf8', (err, data) =>
{    if (err) {
      console.error("File read error:", err);
return;
```

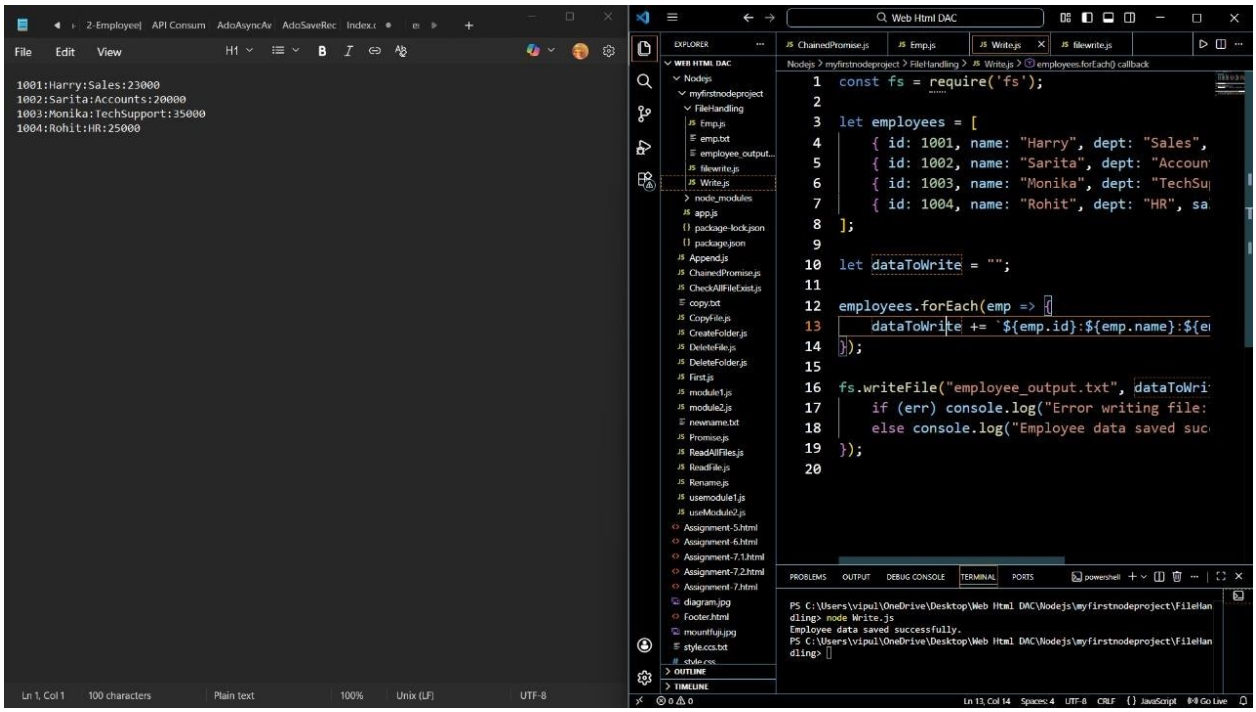
```
    }

    let total = 0;

    const lines = data.split('\n');

    lines.forEach(line =>
{       if (line.trim().length > 0)
{           const parts =
line.split(':');           const
salary = parseInt(parts[3]);
total += salary;
        }
    });

    console.log("Total Salary:", total);
});
```



Q3.

```
const fs = require('fs');


let employees = [
   { id: 1001, name: "Harry", dept: "Sales", salary: 23000 },
   { id: 1002, name: "Sarita", dept: "Accounts", salary: 20000 },
   { id: 1003, name: "Monika", dept: "TechSupport", salary: 35000 },
   { id: 1004, name: "Rohit", dept: "HR", salary: 25000 }
];


let dataToWrite = "";


employees.forEach(emp => {     dataToWrite += ${emp.id}:${emp.name}:$
{emp.dept}:${emp.salary}\n;
});
```

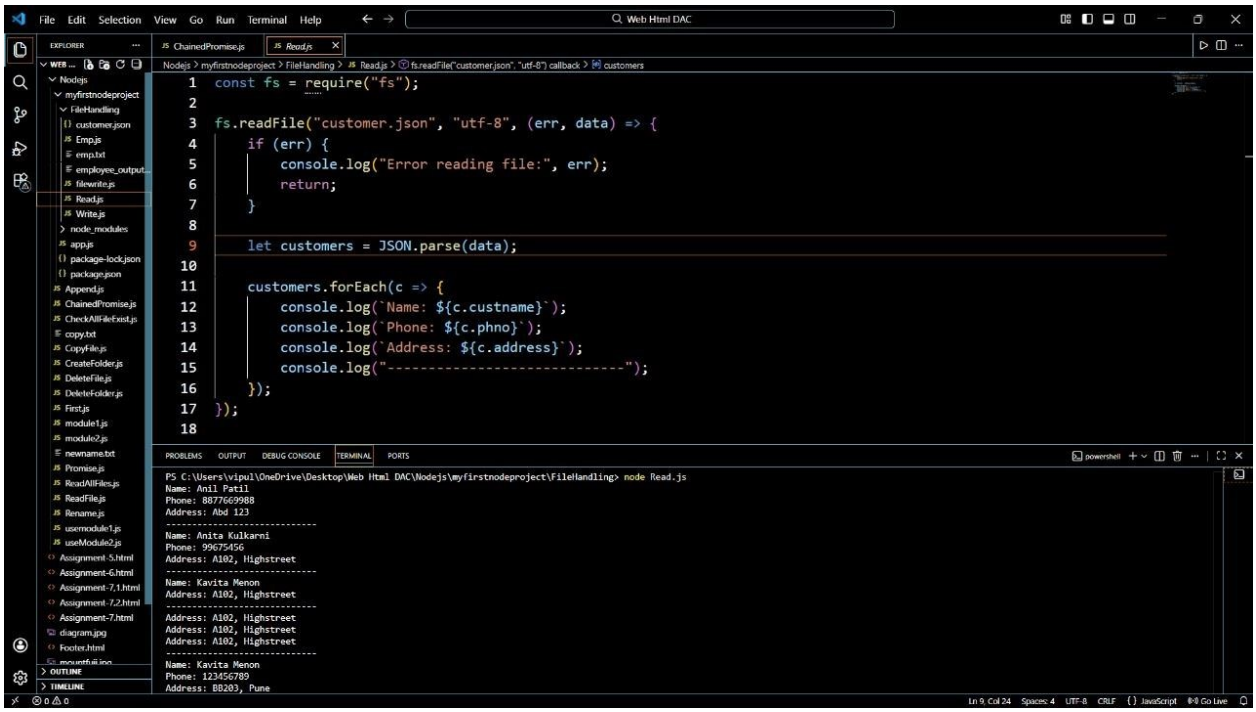
```
fs.writeFile("employee_output.txt", dataToWrite, (err) => {
if (err) console.log("Error writing file:", err);     else
console.log("Employee data saved successfully.");
});
```



Q4.

```
const fs = require("fs");


fs.readFile("customer.json", "utf-8", (err, data) =>{     if
(err) {
      console.log("Error reading file:", err);
return;
   }


   let customers = JSON.parse(data);
```

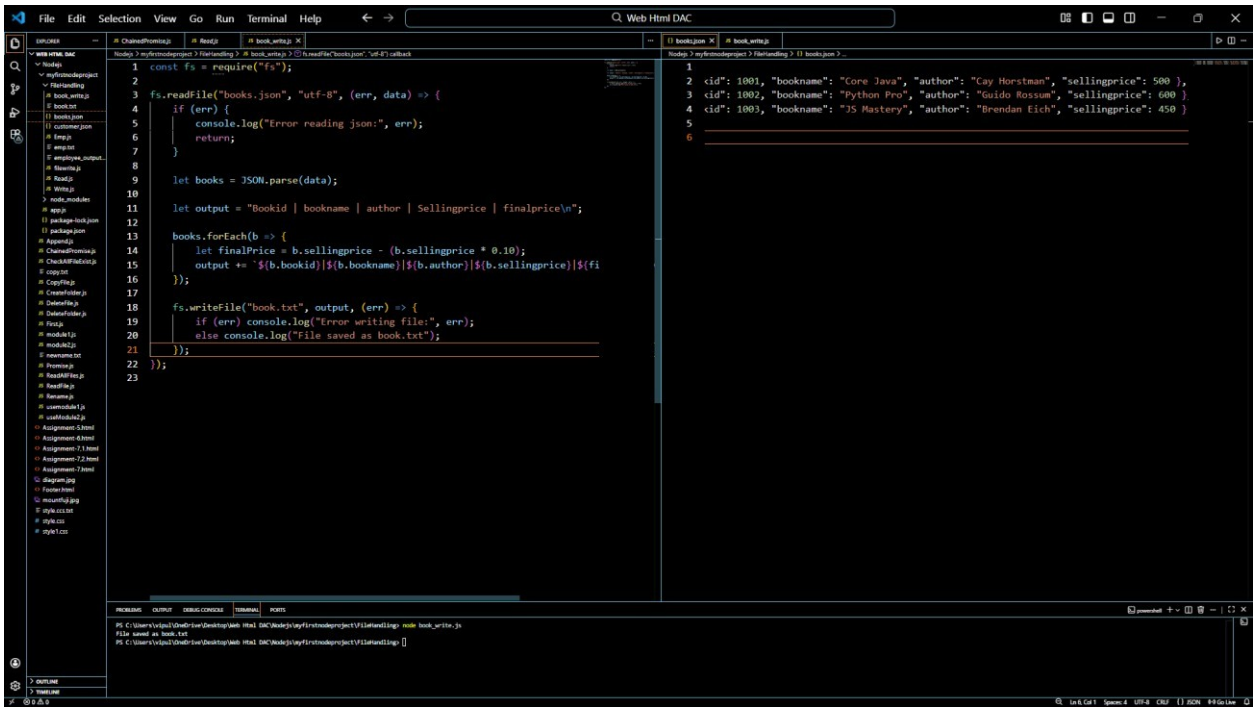
```
                customers.forEach(c       =>
{           console.log(Name: ${c.custname});
console.log(Phone:                ${c.phno});
console.log(Address:              ${c.address});
console.log("----------------------------");
   });
});
```



Q5.

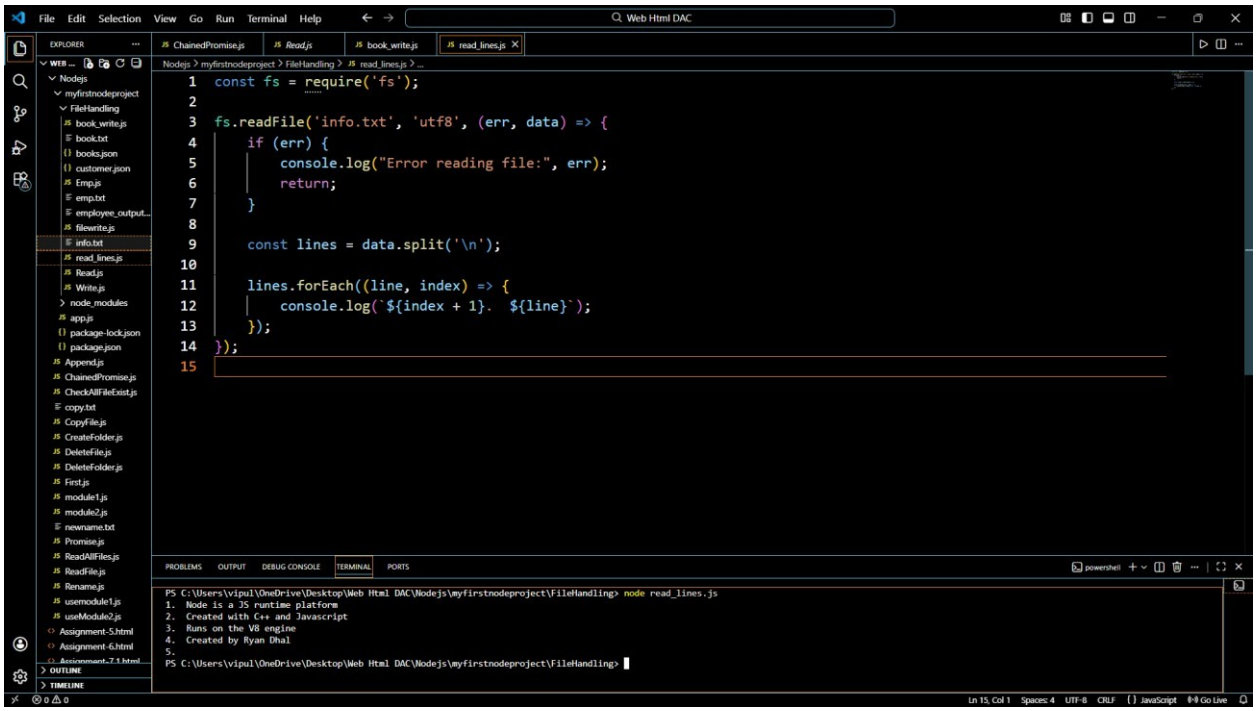
```
const fs = require("fs");


fs.readFile("books.json", "utf-8", (err, data) => {     if
(err) {
      console.log("Error reading json:", err);
return;
```

```
    }

    let books = JSON.parse(data);

    let output = "Bookid | bookname | author | Sellingprice | finalprice\n";

     books.forEach(b => {          let finalPrice = b.sellingprice -
(b.sellingprice  *  0.10);                output  +=  `${b.bookid}|$
{b.bookname}|${b.author}|${b.sellingprice}|${finalPrice}
\n`;

    });

    fs.writeFile("book.txt", output, (err) => {        if
(err) console.log("Error writing file:", err);
else console.log("File saved as book.txt");
    });
});
```



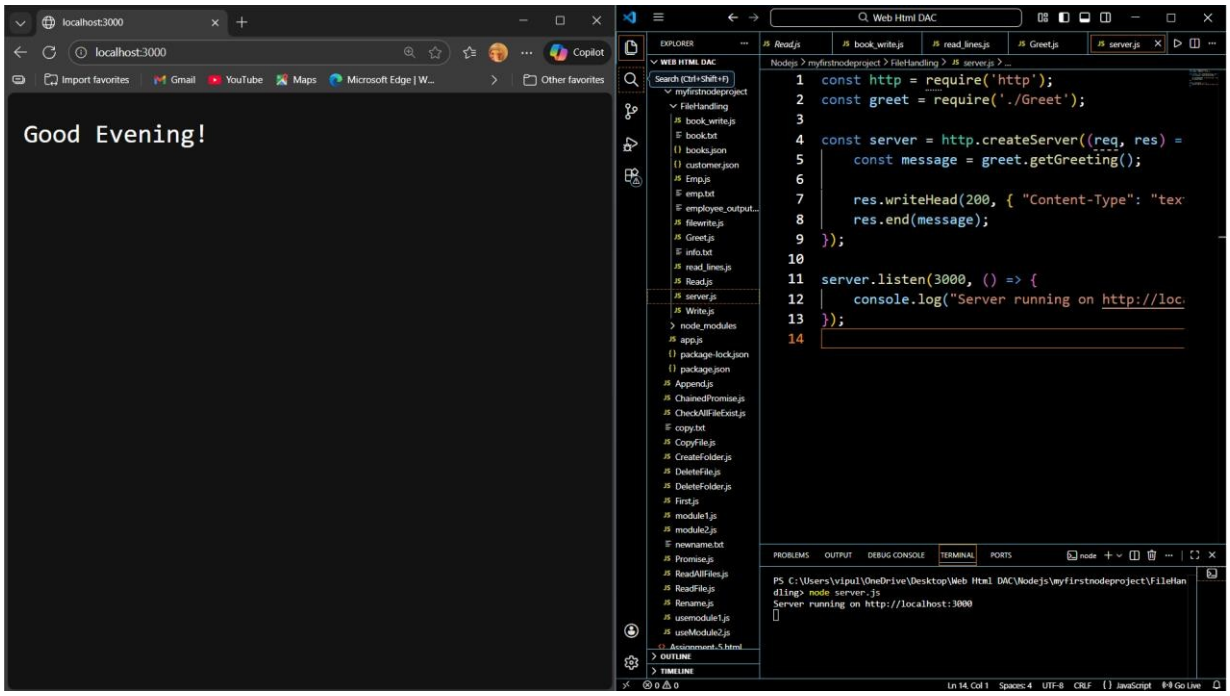
Q6.

```
const fs = require('fs');


fs.readFile('info.txt', 'utf8', (err, data) =>
{    if (err) {
       console.log("Error reading file:", err);
       return;
   }


   const lines = data.split('\n');


   lines.forEach((line, index) => {        console.log(`$
{index + 1}.  ${line}`);
   });
});
```



Node HTTP & the web Q1.

```
const http = require('http'); const
greet = require('./Greet');


const server = http.createServer((req, res) => {
   const message = greet.getGreeting();


   res.writeHead(200, { "Content-Type": "text/plain" });     res.end(message);
});


server.listen(3000, () => {     console.log("Server running on
http://localhost:3000");
});
```



EXPRESSJS :

Q1. const express = require("express"); const app = express(); const path = require("path"); app.get("/", (req, res) => { res.sendFile(path.join(__dirname, "form.html"));
});

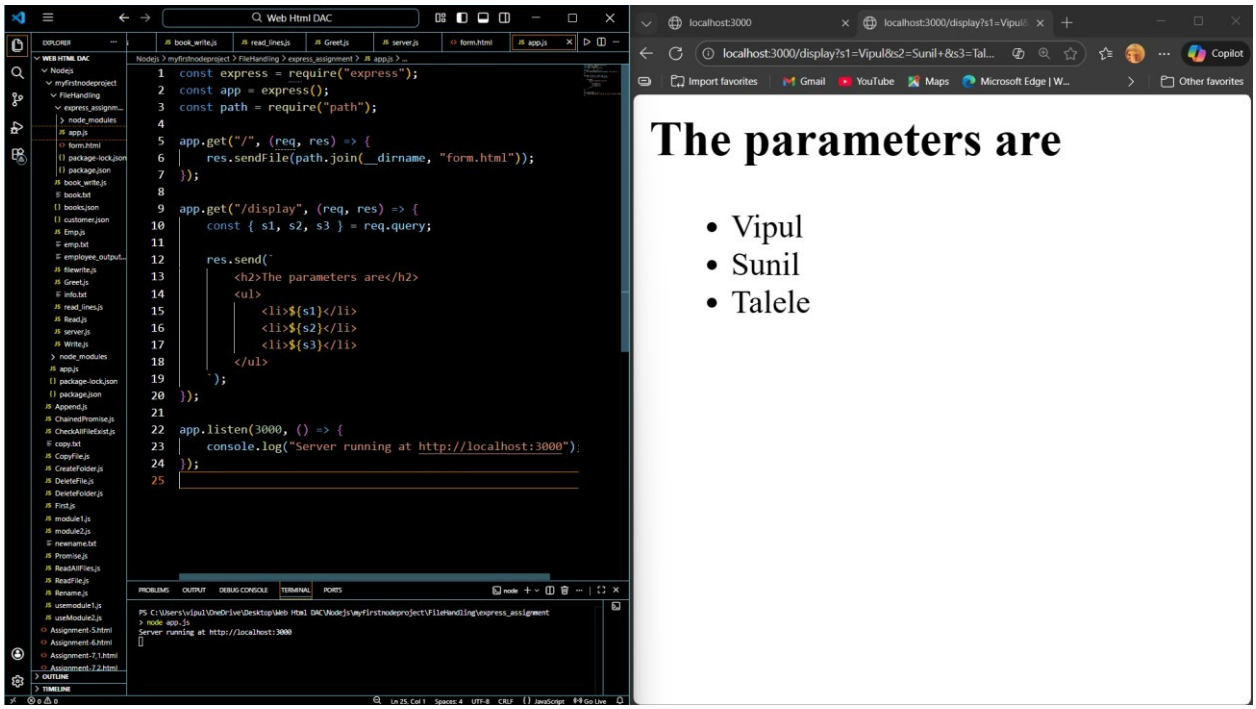
app.get("/display", (req, res) =>
{ const { s1, s2, s3 } = req.query;

res.send(`
<h2>The parameters are</h2>

```
    <ul>
      <li>${s1}</li>
      <li>${s2}</li>
      <li>${s3}</li>
    </ul>
  `);
});
```

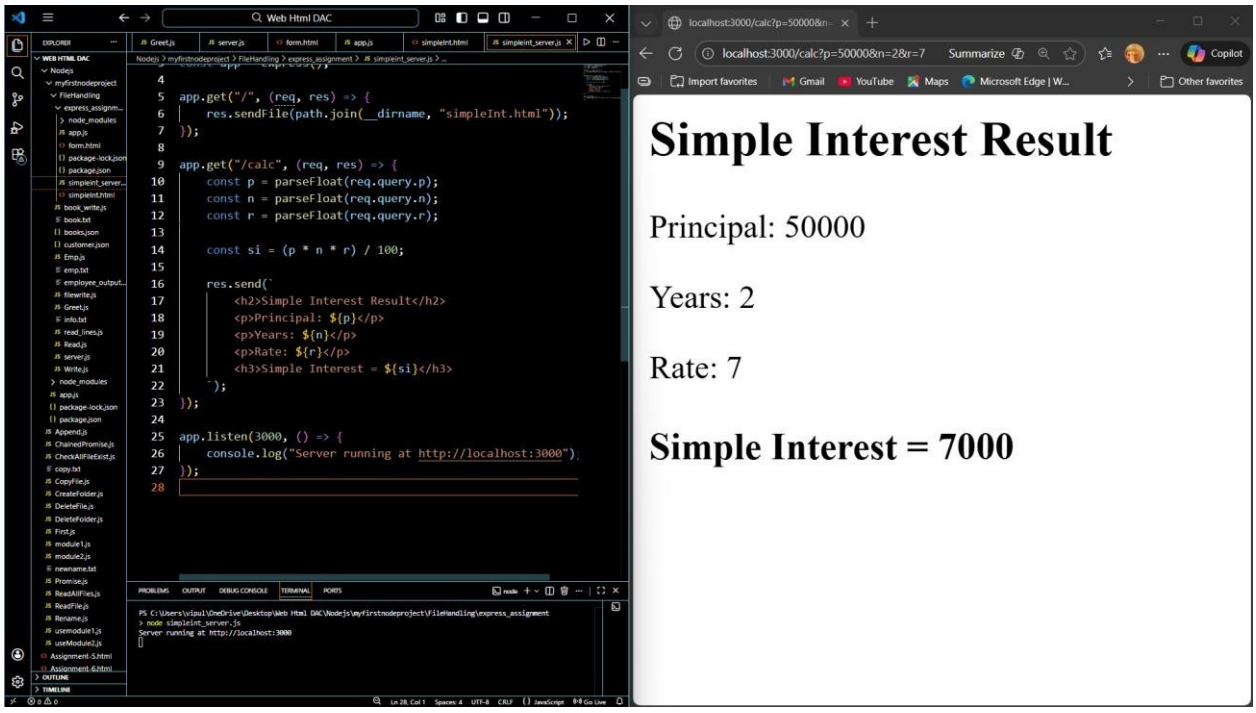
app.listen(3000, () => { console.log("Server running at http://localhost:3000");

});



Q2. const express = require("express"); const path =

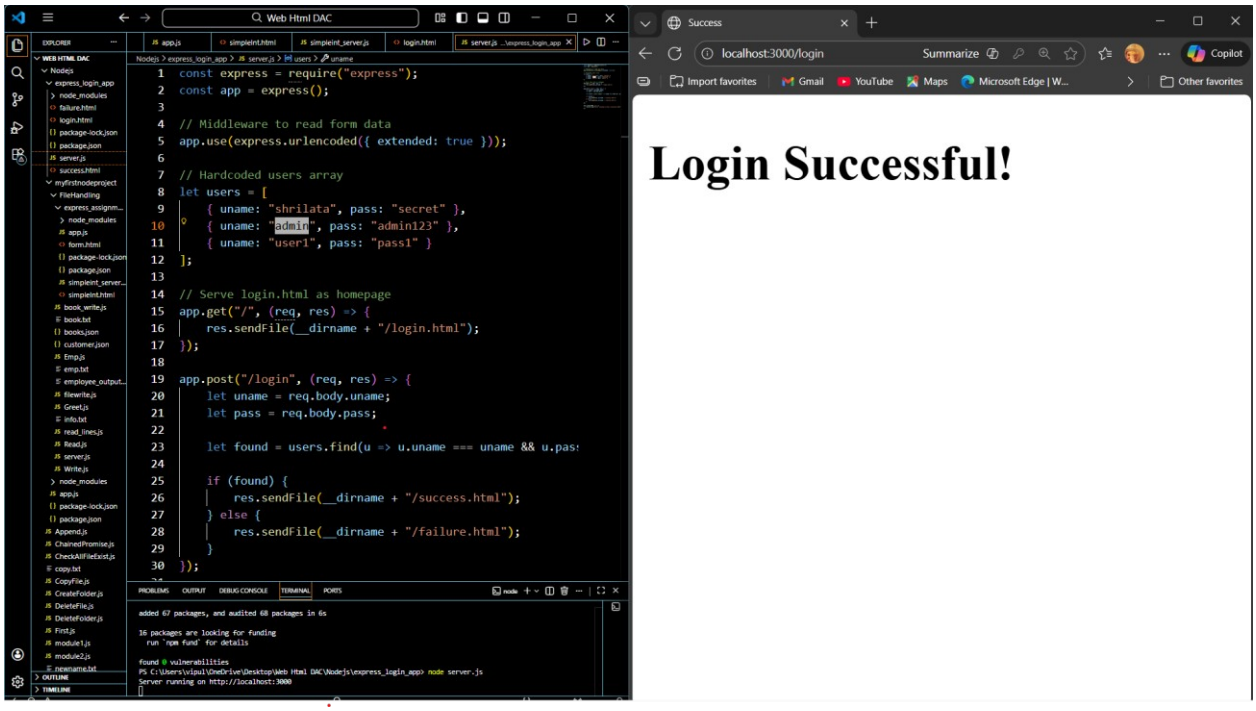
```
require("path");     const     app     =
express();

app.get("/", (req, res) =>{     res.sendFile(path.join(__dirname,
"simpleInt.html"));
});

app.get("/calc", (req, res) =>{     const
p = parseFloat(req.query.p);     const n
= parseFloat(req.query.n);     const r =
parseFloat(req.query.r);

   const si = (p * n * r) / 100;

   res.send(`
      <h2>Simple Interest Result</h2>
      <p>Principal: ${p}</p>
      <p>Years: ${n}</p>
      <p>Rate: ${r}</p>
      <h3>Simple Interest = ${si}</h3>
   `);
});

app.listen(3000, () =>{     console.log("Server running at
http://localhost:3000");
});
```



Q3.

```
const express = require("express"); const
app = express();

// Middleware to read form data
app.use(express.urlencoded({ extended: true }));

// Hardcoded users array let
users = [
  { uname: "shrilata", pass: "secret" },
  { uname: "admin", pass: "admin123" },
  { uname: "user1", pass: "pass1" }
];
```

```
// Serve login.html as homepage
app.get("/", (req, res) =>
{   res.sendFile(__dirname + "/login.html");
});

app.post("/login", (req, res) =>
{   let uname = req.body.uname;
let pass = req.body.pass;

  let found = users.find(u => u.uname === uname && u.pass === pass);

  if (found) {
    res.sendFile(__dirname + "/success.html");
  } else {
    res.sendFile(__dirname + "/failure.html");
  }
});

app.listen(3000, () => {   console.log("Server running on
http://localhost:3000");
});
```



Q4. const express = require("express"); const

bodyParser = require("body-parser"); const

path = require("path");

const app = express();

const PORT = 3000;

let users = [];

app.use(bodyParser.urlencoded({ extended: true })); app.use(express.static("public"));

app.post("/signup", (req, res) => { const

{ username, password } = req.body;

```
    const userObj = { username, password };     users.push(userObj);

    console.log("Current Users Array:", users);

    res.send("<h2>Signup Successful!</h2><p>User stored in array.</p>");
});

app.listen(PORT, () => {     console.log(`Server running at
http://localhost:${PORT}`);
});
```

Q5. const express = require("express"); const app = express();

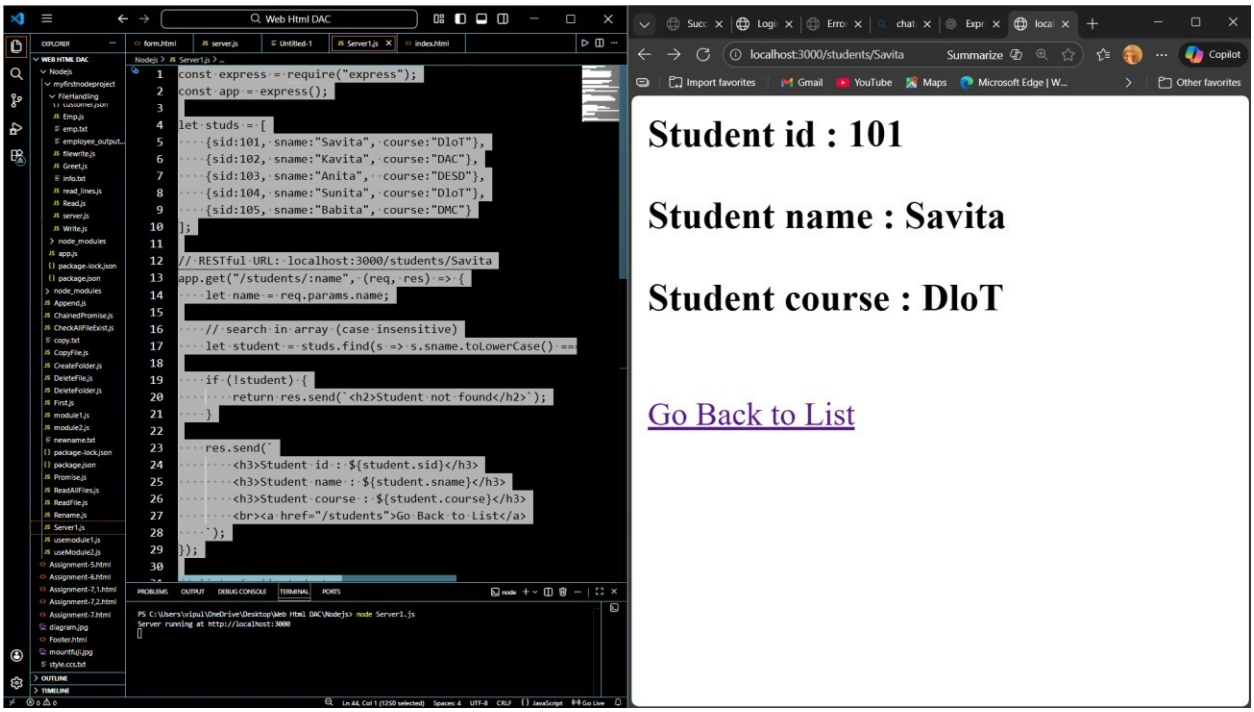
```
let studs = [
    {sid:101, sname:"Savita", course:"DIoT"},
    {sid:102, sname:"Kavita", course:"DAC"},
    {sid:103, sname:"Anita",  course:"DESD"},
    {sid:104, sname:"Sunita", course:"DIoT"},
    {sid:105, sname:"Babita", course:"DMC"}
];
```

```
// RESTful URL: localhost:3000/students/Savita
app.get("/students/:name", (req, res) => {       let
name = req.params.name;

  // search in array (case insensitive)     let student = studs.find(s =>
s.sname.toLowerCase() === name.toLowerCase());

  if (!student) {
     return res.send(`<h2>Student not found</h2>`);
  }

  res.send(`
     <h3>Student id : ${student.sid}</h3>
     <h3>Student name : ${student.sname}</h3>
     <h3>Student course : ${student.course}</h3>
     <br><a href="/students">Go Back to List</a>
  `);
});

// list of all students app.get("/students", (req, res) => {     let html =
"<h2>Student List</h2><ul>";     studs.forEach(s => {        html += `<li><a
href="/students/${s.sname}">${s.sname}</a></li>`;
  });
  html += "</ul>";
res.send(html);
});
```

app.listen(3000, () => { console.log("Server running at

http://localhost:3000");

});



Q6. const express = require("express"); const

students = require("./data/students"); const

app = express();

app.use(express.static("public"));

// API: All students

app.get("/students", (req, res) => { res.json(students);

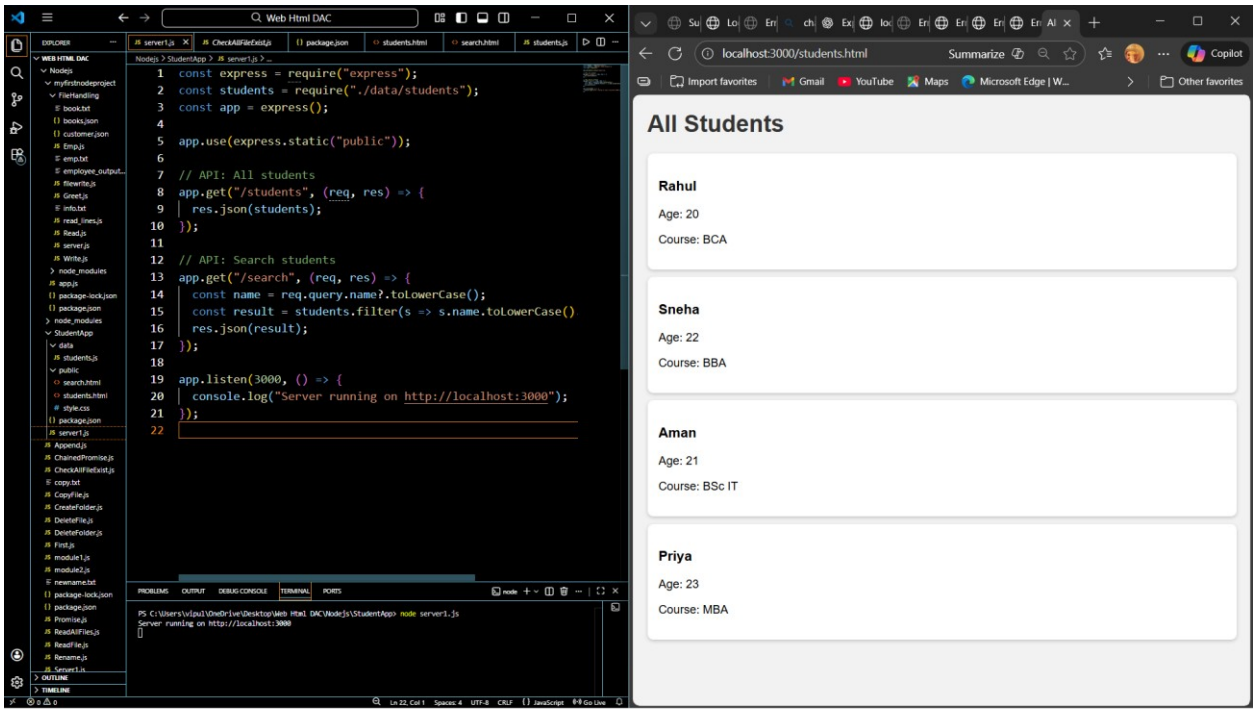
});

```
// API: Search students app.get("/search",
(req, res) => {   const name =
req.query.name?.toLowerCase();   const
result = students.filter(s =>
s.name.toLowerCase().includes(name));
res.json(result);
});

app.listen(3000, () => {   console.log("Server running on
http://localhost:3000");
});

{
  "name": "studentapp",
  "version": "1.0.0",
  "main": "server1.js",
  "scripts": {
    "start": "node server1.js"
  },
  "dependencies": {
    "express": "^4.18.2"
  }
}

<!DOCTYPE html>
<html>
```

```
<head>
   <title>All Students</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>

<h1>All Students</h1>

<div id="studentList"></div>

<script>
fetch("/students")
      .then(res => res.json())
      .then(data =>
{           document.getElementById("studentList").innerHTML =
             data.map(s => `
                <div class="card">
                   <h3>${s.name}</h3>
                   <p>Age: ${s.age}</p>
                   <p>Course: ${s.course}</p>
                </div>
             `).join("");
      });
</script>

</body>
```

```
</html>

<!DOCTYPE html>
<html>
<head>
   <title>Search Student</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>

<h1>Search Student</h1>

<input type="text" id="searchBox" placeholder="Enter student name">
<button onclick="searchStudent()">Search</button>

<div id="result"></div>

<script> function searchStudent() { let name =
document.getElementById("searchBox").value;

   fetch(`/search?name=${name}`)
      .then(res => res.json())
.then(data => { if (!
data.length) {
         document.getElementById("result").innerHTML = "<p>No student found</p>";
return;
```

```
        }

        document.getElementById("result").innerHTML =
data.map(s => `
            <div class="card">
              <h3>${s.name}</h3>
              <p>Age: ${s.age}</p>
              <p>Course: ${s.course}</p>
            </div>
          `).join("");
      });
}
</script>

</body>
</html>

module.exports = [
  { id: 1, name: "Rahul", age: 20, course: "BCA" },
  { id: 2, name: "Sneha", age: 22, course: "BBA" },
  { id: 3, name: "Aman", age: 21, course: "BSc IT" },
  { id: 4, name: "Priya", age: 23, course: "MBA" }
];
```



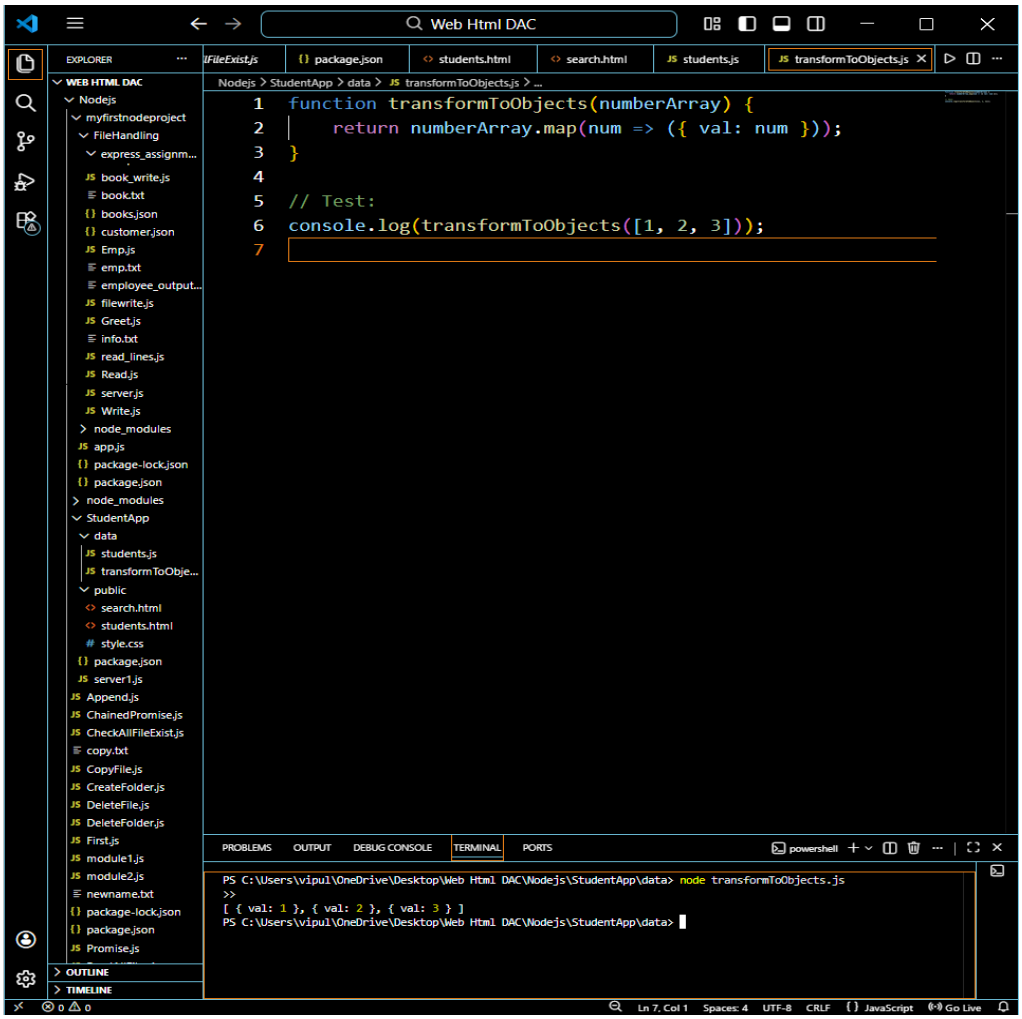
NEXT GEN JAVASCRIPT ASS :

Q1. function transformToObjects(numberArray)

{ return numberArray.map(num => ({ val: num }));

}

console.log(transformToObjects([1, 2, 3]));

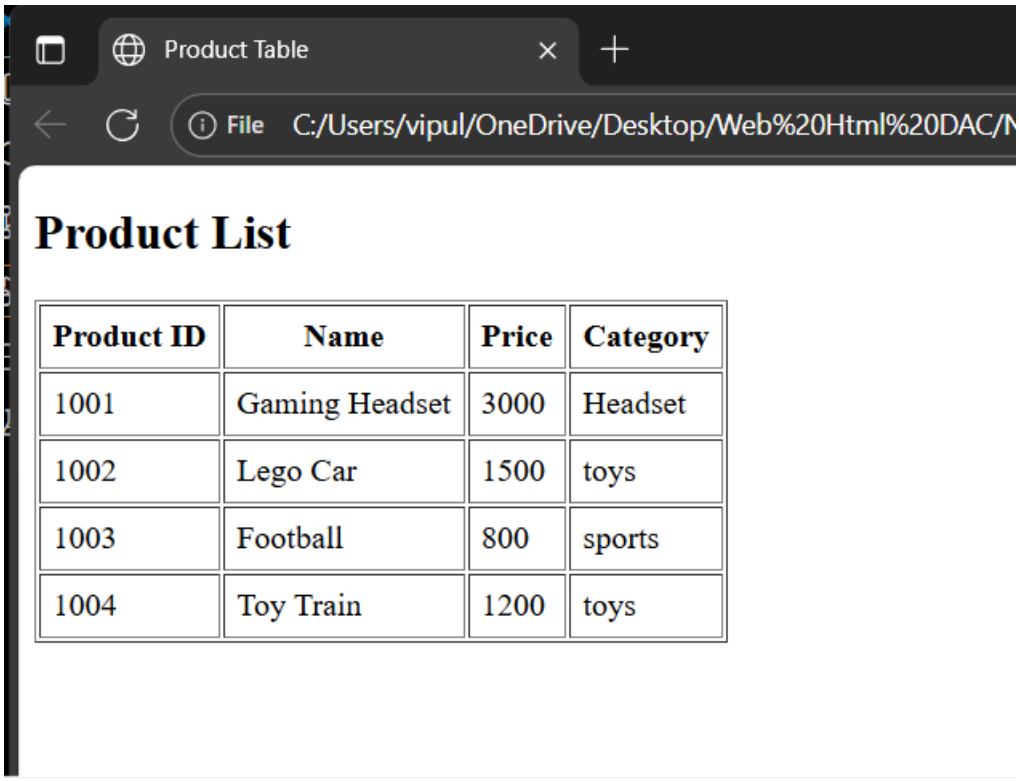


Q2.

```
<!DOCTYPE html>

<html>

<head>

   <title>Product Table</title>

</head>

<body>


<h2>Product List</h2>


<table border="1" cellpadding="6">
```

```
  <thead>
    <tr>
      <th>Product ID</th>
      <th>Name</th>
      <th>Price</th>
      <th>Category</th>
    </tr>
  </thead>
  <tbody id="productBody"></tbody>
</table>

<script>    let
productArr = [
    { pid: 1001, prodName: "Gaming Headset", price: 3000, category: "Headset" },
    { pid: 1002, prodName: "Lego Car", price: 1500, category: "toys" },
    { pid: 1003, prodName: "Football", price: 800, category: "sports" },
    { pid: 1004, prodName: "Toy Train", price: 1200, category: "toys" }
  ];

  let tableBody = document.getElementById("productBody");

  productArr.forEach(product =>
{       tableBody.innerHTML += `
        <tr>
          <td>${product.pid}</td>
          <td>${product.prodName}</td>
```

```
            <td>${product.price}</td>
            <td>${product.category}</td>
        </tr>
    `;
  });
</script>

</body>
</html>
```

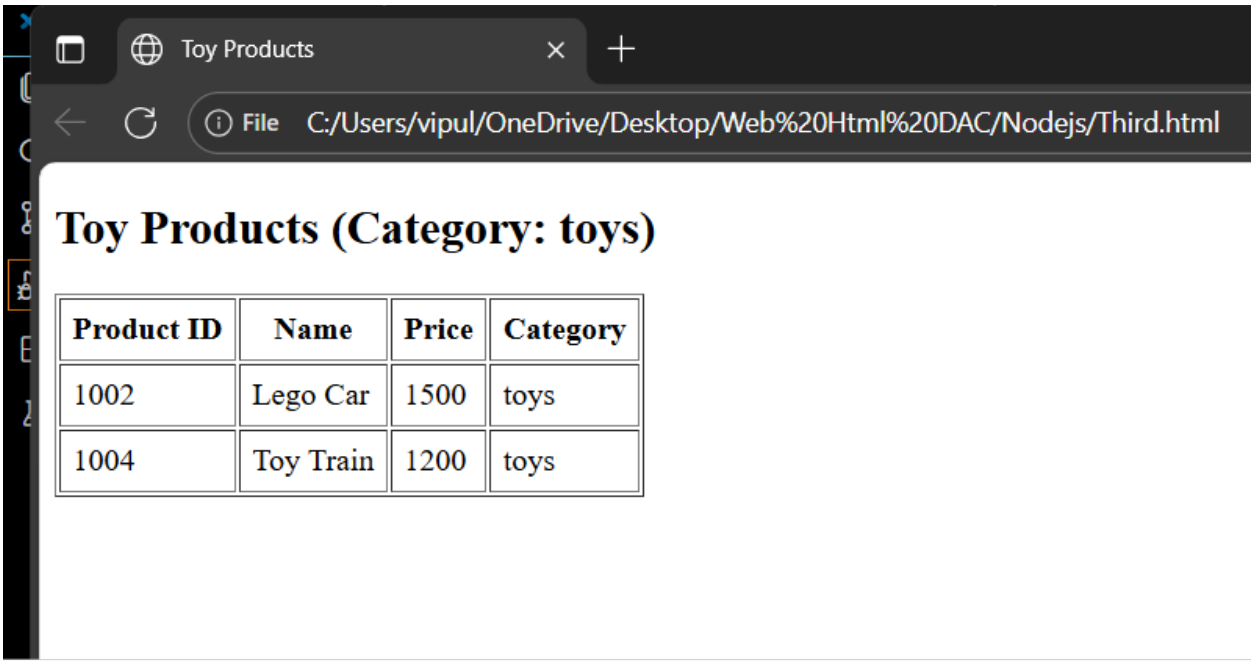


Q3.

```
<!DOCTYPE html>
```

```
<html>
<head>
  <title>Toy Products</title>
</head>
<body>

<h2>Toy Products (Category: toys)</h2>

<table border="1" cellpadding="6">
  <thead>
    <tr>
      <th>Product ID</th>
      <th>Name</th>
      <th>Price</th>
      <th>Category</th>
    </tr>
  </thead>
  <tbody id="toyBody"></tbody>
</table>

<script>

  let productArr = [
    { pid: 1001, prodName: "Gaming Headset", price: 3000, category: "Headset" },
    { pid: 1002, prodName: "Lego Car", price: 1500, category: "toys" },
    { pid: 1003, prodName: "Football", price: 800, category: "sports" },
```

```
        { pid: 1004, prodName: "Toy Train", price: 1200, category: "toys" }
    ];


    let ToyArray = productArr.filter(product => product.category === "toys");


    let toyBody = document.getElementById("toyBody");

    ToyArray.forEach(product =>
{       toyBody.innerHTML += `
          <tr>
             <td>${product.pid}</td>
             <td>${product.prodName}</td>
             <td>${product.price}</td>
             <td>${product.category}</td>
          </tr>
       `;
    });
</script>

</body>
</html>
```


Toy Products
File
C:/Users/vipul/OneDrive/Desktop/Web%20Html%20DAC/Nodejs/Third.html
Toy Products (Category: toys)
Product ID
Name
Price
Category
1002
Lego Car
1500
toys
1004
Toy Train
1200
toys